Cognitive-Developmental Learning for a Humanoid Robot: A Caregiver's Gift

by

Artur Miguel Do Amaral Arsenio

Submitted to the Department of Electrical Engineering and Computer Science
in partial fulfillment of the requirements for the degree of

Doctor of Philosophy in Computer Science and Engineering

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

May 2004

Author .....
Department of Electrical Engineering and Computer Science
May 18, 2004

Certified by .....
Rodney A. Brooks
Fujitsu Professor of Computer Science and Engineering
Thesis Supervisor

Accepted by .....
Arthur C. Smith
Chairman, Department Committee on Graduate Students

Cognitive-Developmental Learning for a Humanoid Robot: A Caregiver's Gift

by

Artur Miguel Do Amaral Arsenio

Submitted to the Department of Electrical Engineering and Computer Science
on May 18, 2004, in partial fulfillment of the
requirements for the degree of
Doctor of Philosophy in Computer Science and Engineering

Abstract

The goal of this work is to build a cognitive system for the humanoid robot, Cog, that exploits human caregivers as catalysts to perceive and learn about actions, objects, scenes, people, and the robot itself. This thesis addresses a broad spectrum of machine learning problems across several categorization levels. Actions by embodied agents are used to automatically generate training data for the learning mechanisms, so that the robot develops categorization autonomously.

Taking inspiration from the human brain, a framework of algorithms and methodologies was implemented to emulate different cognitive capabilities on the humanoid robot Cog. This framework is effectively applied to a collection of AI, computer vision, and signal processing problems. Cognitive capabilities of the humanoid robot are developmentally created, starting from infant-like abilities for detecting, segmenting, and recognizing percepts over multiple sensing modalities. Human caregivers provide a helping hand for communicating such information to the robot. This is done by actions that create meaningful events (by changing the world in which the robot is situated) thus inducing the "compliant perception" of objects from these human-robot interactions. Self-exploration of the world extends the robot's knowledge concerning object properties.

This thesis argues for enculturating humanoid robots using infant development as a metaphor for building a humanoid robot's cognitive abilities. A human caregiver redesigns a humanoid's *brain* by teaching the humanoid robot as she would teach a child, using children's learning aids such as books, drawing boards, or other cognitive artifacts. Multi-modal object properties are learned using these tools and inserted into several recognition schemes, which are then applied to developmentally acquire new object representations. The humanoid robot therefore sees the world through the caregiver's eyes.

Building an artificial humanoid robot's *brain*, even at an infant's cognitive level, has been a long quest which still lies only in the realm of our imagination. Our efforts towards such a dimly imaginable task are developed according to two alternate and complementary views: cognitive and developmental.

Thesis Supervisor: Rodney A. Brooks
Title: Fujitsu Professor of Computer Science and Engineering

To Hugo,
my 2-year old son, who taught me so much
To Helga,
my wife, who taught me how to teach

To my caregivers: my father, Horacio
my mother, Benvinda
and to my brother, Luis

Acknowledgments

This is the acknowledgements section. You should replace this with your own acknowledgements.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

A learning machine must be programmed by experience.

(Wiener, 1948)

Intelligence is in the eye of the observer.

(Brooks et al., 1998)

The goal of this thesis is to build a cognitive system for a humanoid robot exploiting developmental learning: human caregivers are used as a catalyst to help a robot perceive and learn meaningful information. This strategy is shown to solve a broad spectrum of machine learning problems along a large categorical scope: actions, objects, scenes, people and the robot itself. This work is motivated by cognitive development of human infants, which is bootstrapped by the helping hand that human caregivers (and especially the infant's mother) provide to the infant.

Cognitive capabilities of the humanoid robot will be created developmentally, starting from an infant-like early ability to detect low-level features over multiple sensing modalities, such as skin-color or repetitive or abruptly varying world events from human-robot interactions, and moving developmentally towards robust perception and learning.

We argue for enculturated humanoid robots – introducing robots into our society and treating them as us – using child development as a metaphor for developmental learning of a humanoid robot. We exploit extensively childhood learning elements such as books (a child's learning aid) and other cognitive artifacts such as drawing boards. Multi-modal object properties are learned using these tools and inserted into

several recognition schemes, which are then applied to developmentally acquire new object representations. Throughout this thesis, the humanoid robot therefore sees the world through the caregiver's eyes.

1.1 Motivation

The working of the human mind has long puzzled and amazed the human kind, drawing theorists of every persuasion, and from many disciplines, to the barely imaginable task of trying to fill in the details of such scientific challenge:

> *Questions about the origins and development of human knowledge have been posed for millenia. What do newborn infants know about their surroundings, and what do they learn as they observe events, play with objects, or interact with people? Behind these questions are deeper ones: By what processes does knowledge grow, how does it change, and how variable are its developmental paths and endpoints?*
>
> *(Wilson and Keil, 1999)*

Classical artificial intelligence (AI) researchers adopted as an implicit and dominant hypothesis the claim of (Newell and Simon, 1961) that humans use symbolic systems to "think" (for a critic review see (Brooks, 1991a)). AI systems following such theory create explicit, internal representations and need to fully store the state of the outside world. Search strategies are used for problem-solving, applied mostly to narrow domains, such as playing chess (DeCoste, 1997).

Similarly to early AI, robotics has been applied mostly to narrow, goal-oriented specific applications. In addition, task execution often requires artificial, off-line engineering of the robot's physical environment, creating *idealized* working conditions. But moving towards intelligent robots will require general purpose robots capable of learning in any real-world environment.

Fundamental problems remain to be solved in several fields, such as

- ▷ Computer Vision: object segmentation and recognition; event/goal/action identification; scene recognition
- ▷ Robotics: robot tasking involving differentiated actions; learning and executing tasks without initial manual setups
- ▷ Cognitive Sciences and AI: understanding and creation of *grounded* knowledge structures, situated in the world
- ▷ Neurosciences: a more profound understanding of biological brains and their macro and micro structures,

just to name a few. Furthermore, these problems are mostly studied independently of each other, since they lie in different research domains. In this thesis I will concentrate on the problems in the first three fields, and in doing so I expect to better understand the last field.

1.1.1 Brains are Complex and Highly Interconnected

Although humans have specific brain processing areas, such as Broca's language area or the visual cortex, these areas are highly interconnected in complex patterns.

Classical AI has a tendency to overpass these complex aspects of human intelligence (Minsky and Papert, 1970). Throughout this thesis, I argue that several problems in different fields are closely tied. For instance, let us consider the computer vision problem of object recognition. In the summer of 1966, a vision project at the now extinct MIT AILab aimed at the implementation of object categorization – a problem which requires complex structures in the human brain – in two months (Papert, 1966). They were not successful, being the object recognition problem still under intensive research. A common definition for this problem, as presented in (Zhang and Faugeras, 1992), is:

Definition *We are given a database of object models and a view of the real world. For each object in the model database, the object recognition and localization problem consists in answering the following two questions:*

- ▷ *Is the object present in the observed scene?*
- ▷ *If present, what are the 3D pose parameters (translation and rotation parameters) with respect to the sensor coordinate system?*

If possible, the system should learn unknown objects from the observed data.

This definition relies on a very strong assumption: Objects are recognized by their appearance solely. I argue that this is not enough. All of the following are sometimes true:

- ▷ *Objects can be recognized by their appearance - color, luminance, shape, texture.*
- ▷ *Objects have other physical features, such as mass, of a dynamic nature.*
- ▷ *The dynamic behavior of an object varies depending on its actuation.*
- ▷ *Temporal information is necessary for identifying functional constraints. The object motion structure - the kinematics - should be taken into account.*
- ▷ *Objects are situated in the world, which may change an object's meaning depending on context.*
- ▷ *Objects have an underlying hierarchic tree structure - which contains information concerning objects that are reducible to other objects, i.e., that originated by assembling several objects.*
- ▷ *There is a set of actions that can be exerted on an object, and another set of actions that an object can be used for (e.g., a nail can be hammered, while a hammer is used for hammering). Therefore, a set of affordances (Adolph et al., 1993) also intrinsically define (albeit not uniquely) an object.*

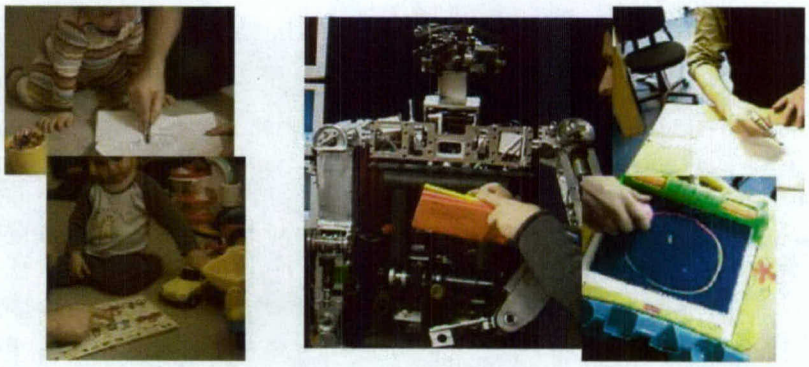


Figure 1-1: A caregiver as a child/humanoid robot's catalyst for learning.

The set of issues just stated requires the solution of problems in several fields.This thesis puts special emphasis in three specific problems: learning from demonstration, object recognition and the problem of robot tasking. Although I will not seek *perfect* solutions for these problems, I will have indeed to deal with them, due to the highly interconnected nature of this thesis work.

1.1.2 The Helping Hand – Humans as Caregivers

Teaching a visual system information concerning the surrounding world is a difficult task, which takes several years for a child, equipped with evolutionary mechanisms stored in its genes, to accomplish. Caregivers give a helping hand to facilitate infants' learning, changing interaction patterns according to the infants' performance. Hence, infants functional development occurs in simultaneous with the development of the caregivers' skills for socially interacting with infants (Newport, 1990).

However, developmental disorders such as autism (DSM-IV, 1994) severely damages the infants' social skills. Although autistic children often seem to have normal perceptual abilities, they do not recognize or respond to normal social cues (Baron-Cohen, 1995). This asocial behavior puts serious constrains on, and severely limits, the learning process in autistic children.

Our approach exploits therefore help from a human caregiver in a robot's learning loop to extract meaningful percepts from the world (see figures 1-1 and 1-2). Through social interactions of a robot with a caregiver, the latter facilitates the robot's perception and learning, in the same way as human caregivers facilitate a child's perception and learning during child development phases. Minsky seems to agree with such notion:

> *Isnt it curious that infants find social goals easier to accomplish than physical goals, while adults find the social goals more difficult? One way to explain this is to say that the presence of helpful people simplifies the infants social world – since because of them, simpler actions solve harder problems.* *(Minsky, 1985)*

Figure 1-2: Infant and caregiver social interactions in art.

1.1.3 Developmental Learning – Humanoids as Children

Infants develop both functionally and physically as they grow. Such development is very important for infants' learning (Newport, 1990; Elman, 1993; Elman et al., 1996). Turing, the creator of the famous Turing test to evaluate artificial intelligence of computers, suggested that, instead of producing programmes to simulate the adult mind, we should rather develop one which simulates the childs mind (Turing, 1950). He also suggested that an appropriate course of education would lead to the adult brain (Turing, 1950). Although teaching humanoid robots like children is not a new idea (Metta, 2000; Kozima and Yano, 2001; Lungarella and Berthouze, 2003), we apply such philosophy to solve a broad spectrum of research problems. In addition, this thesis approach also follows Turing's advice of giving educational courses to an artificial system, by utilizing in innovative ways a child's arsenal of educational tools and toys to boost learning capabilities.

Evidence suggests that infants have several preferences and capabilities shortly after birth (Bremner, 1994). Such predispositions may be innate or pre-acquired at the mother's womb. Inspired by infants' innate or pre-acquired capabilities, the

robot is assumed to be initially pre-programmed for the detection of real-world events both in time and frequency, and correlations among these events, no matter the sensing device from which they are perceived. In addition, the robot prefers salient visual stimuli (as do newborns (Banks and Ginsburg, 1985; Banks and Dannemiller, 1987)). These preferences correspond to the initial robot's capabilities (similar to the information stored on human *genes* – the **genotype**) programmed into the robot to process these events.

Starting from this set of premises, the robot should be able to incrementally build a knowledge database and extrapolate this knowledge to different problem domains (the social, emotional, cultural, developmental learning will set the basis for the **phenotype**). For instance, the robot learns the representation of a geometric shape from a book, and is thereafter able to identify animate gestures or world structures with such a shape. Or the robot learns from a human how to poke an object, and uses afterwards such knowledge to poke objects to extract their visual appearance.

1.1.4 Embodied Perception/Action: Putting Bodies Into Use

One approach to AI is the development of robots embodied and situated in the world (Brooks, 1999). Embodied and situated perception (Arsenio, 2002, 2003a; Brooks, 1999; Pfeifer and Scheier, 1926) consists of boosting the perceptual capabilities of an artificial creature by fully exploiting the concepts of an embodied agent situated in the world (Anderson, 2003). Active vision (Aloimonos et al., 1987; Bajcsy, 1988), contrary to passive vision, argues for the active control of the visual perception mechanism so that perception is facilitated. Percepts can indeed be acquired in a purposive way by the active control of a camera (Aloimonos et al., 1987). This approach has been successfully applied to several computer vision problems, such as stereo vision - by dynamically changing the baseline distance between the cameras or by active focus selection (Krotkov et al., 1990).

This thesis proposes embodiment and situatedness as a means to constrain complex problems. I argue, together with other researchers at the MIT CSAIL, for the control of not only the perception apparatus, but also for the manipulator control to achieve the same objective (Arsenio, 2003a; Fitzpatrick and Metta, 2002). We argue for solving a visual problem by not only actively controlling the perceptual mechanism, but also and foremost crucially changing the environment actively through experimental manipulation. We focus on a particular aspect of embodiment: embodied creatures act as a source of energy. They are able to actuate, to exert forces and cause motion over objects in the surrounding environment, in a meaningful way. However, this work intends to go further than embodied robots; humans are also embodied, and the robot can exploit others' bodies to constrain perceptual information. In addition, humans better and more easily execute general purpose tasks (Kemp, 1997).

Therefore, I intend to improve robot cognitive capabilities through interactions with a human "robotsitter" (or robot teacher), an embodied creature whose objective is teaching the robot. This active role of embodied creatures will also be exploited as a means to change scene context, facilitating perception.

1.1.5 Situatedness – Information Stored in the World

In classical AI, sensory perceptions are converted into representational states used within an internal model of the external world (which is often a computationally expensive three-dimensional representation of the external environment).

However, there is experimental evidence from psychology and the neurosciences that humans try to minimize their internal representations of the world. In one experiment, (Ballard et al., 1995) shows that humans do not maintain an internal model of the entire visible scene for the execution of a complex task (like building a copy of a display of blocks). Other experiments in the the area of change blindness corroborate such results (Rensink et al., 1997).

Hence, we will opt by a statistical approach which minimizes internal representations, by exploiting the availability of contextual information stored in the real world. Indeed, there are strong correlations between the environment and objects within it. There is increasing evidence from experiments in scene perception and visual search that the human visual system extensively applies these correlations for facilitating both object detection and recognition (Palmer, 1975; Biederman et al., 1982; Chun and Jiang, 1998). This suggests an early use of contextual information in human perception. Indeed, it seems that extensive visual processing of objects might occur only after a scene has been identified (Potter, 1975; Biederman, 1987; Rensink et al., 1997).

Therefore, objects in the world are situated (Brooks, 1999), in the sense that they usually appear in specific places under a determined context. Because of the strong dependence between scenes and objects found in it, knowledge of probable spatial locations for an object helps to find the same object in the scene, even if the object is not visible (e.g., if located inside a box). There is also evidence from experimental psychology that the attentional processes used to check the presence of a target object in a familiar scene may be similar to those involved in searching for a target object in a novel scene (Wolfe et al., 2002). But for an embodied creature, it is equally important to know where an object should be placed in an environment, so that it can be easily found later - if you place a book in the fridge, you will hardly find it later!

Object categorization may also change depending on the world context. Change the context, and the function, name or other attributes of an object may change (e.g., a rod is categorized as a pendulum when oscillating around a fixed point). This leads to scene-dependent object recognition, and scene recognition reframed as the detection of the probable configuration of objects. But most important, this means humans can transmit the right categorization for an object to an infant (or a humanoid robot) by controlling the world context. Such control of contextual information will be extensively applied to facilitate robot's perception.

1.1.6 Real World vs Virtual Worlds

The real world in which a robot operates is full of entropy. Indeed, it is not possible to just impose an upper bound on the complexity level of a scene. To make things

worse, sensors often perceive only poorly characterized or ambiguous information.

However, as demonstrated by several artistic movements such as the Impressionists, humans are particularly good at processing even very deteriorated signals to extract meaningful information. It seems humans apply stored knowledge to *reconstruct* low resolution images into higher resolution ones (Baker and Kanade, 2000) (see also (Torralba and Sinha, 2001) for another computational approach to the problem of recovering faces from noisy, very low resolution images). Therefore, this thesis will not put emphasis on improving or maximizing sensor resolution or quality. Instead, high resolution information from sensors will be discarded to decrease processing times. Sensors are intended therefore for filtering world data, and hence reducing complexity.

For infants, the caregiver biases the learning process by carefully structuring and controlling the surrounding environment. Similarly, help from a human actor will be applied to constrain the robot's environment complexity, facilitating both object and scene recognition. However, it should be emphasized that such help does not include constraining the world structure (for instance by removing environment cluttering or careful lighting). The focus will be placed on communicating information to a robot which boosts its perceptual skills, helping the visual system to filter out irrelevant information. Indeed, while teaching a toddler, parents do not remove the room's furniture or buy extra lights to just show the child a book! Help instead is given by facilitating the child's task of stimulus selection (for example, by pointing to or tapping an image in a book (Arsenio, 2004h)).

1.2 The Humanoid Form

Some countries like Japan, facing an aging population, are feeling the pressure for autonomous robotic devices able to ameliorate and assist the elderly. Besides the sector of service robots, industries are also investing in more autonomy, as exemplified by robotic prototypes for cleaning autonomously floors in manufacturing facilities, for automatic parts or oil pipe inspections. Security applications for robots are also in the rise, namely for surveillance and mine clearing. And robots are increasingly appearing at consumer homes through the toy market. Several biologically inspired robots (for now mainly at the aesthetical level) have been recently placed on the toy market, as is the case of Sony's "Aibo" or iRobot's "My Real Baby" robots.

The urban world is nowadays shaped to human-scale operation. And people are already used to interact socially with things that look familiar, such as people or animal like appearances. Advances in developmental psychology and the cognitive sciences has inspired roboticists to build intelligent systems having therefore a human-like appearance (Brooks et al., 1998; Brooks and Stein, 1994; Asada et al., 2001). The humanoid form (as shown in figure 1-3) is important for human-robot social interactions in a natural way. Two thorough reviews on socially interactive and developmental robots are presented in (Fong et al., 2003) and (Lungarella and Metta, 2003), respectively.

A sample of several humanoid research projects are shown in figure 1-3. A compar-

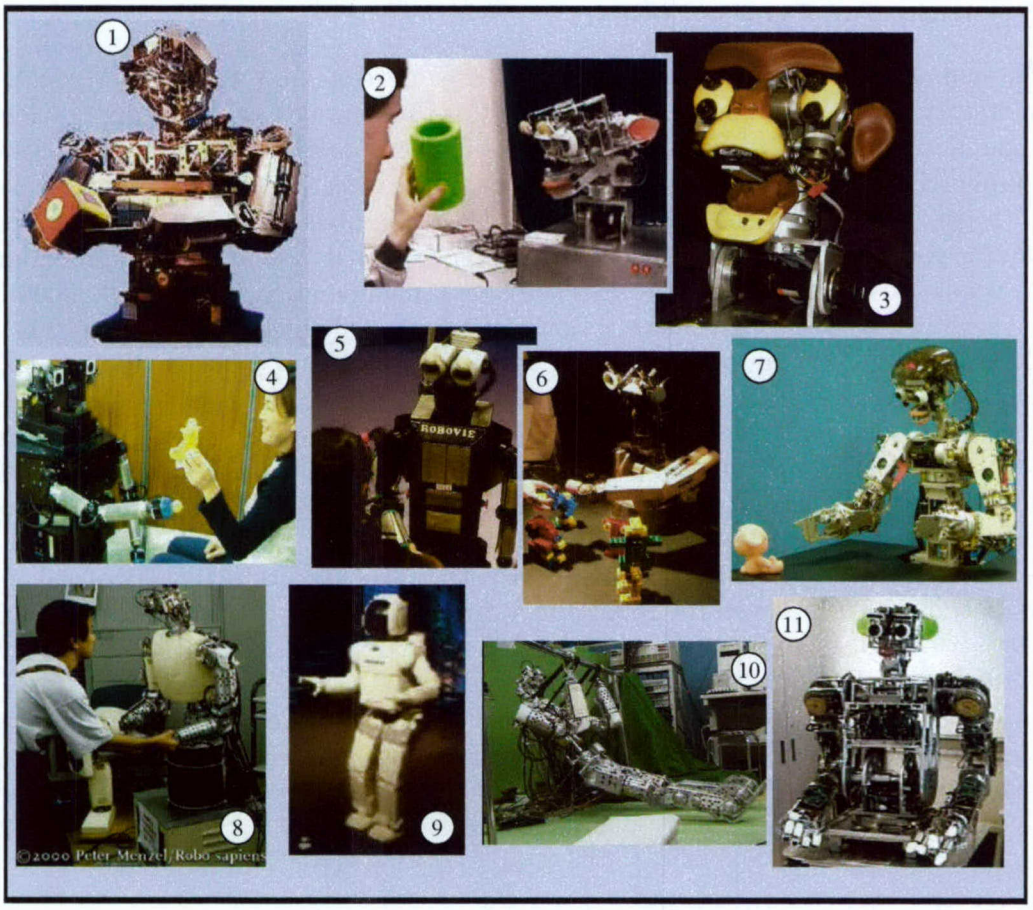


Figure 1-3: Several humanoid robot projects. Robots at the MIT CSAIL humanoid robotics group 1) The humanoid robot Cog; 2) Robot Kismet; and 3) The M2-M4 macaco robot, designed and built by the author. Other robots with the humanoid form: 4) Humanoid robot at Osaka University Asada's lab; 5) Robovie is an interaction-oriented robot developed at Intelligent Robotics and Communication Laboratories at ATR (Advanced Telecommunications Research Institute International); 6) Babybot, a humanoid robot developed at the LIRA-Lab; 7) Infanoid, an infant-like robot which has been developed at Communications Research Laboratory in Japan; 8) Humanoid Jack, at ETL in Japan; 9) Honda's Asimo robot; 10) and 11) Other humanoid platforms (see (Hashimoto, 1998) for a description of more humanoid robotics projects at Waseda University).

ative study of some of these robots over social-developmental capabilities is presented by (Nagai, 2004).

Cog, Kismet and Macaco Robots

The humanoid robot Cog (Brooks et al., 1998), developed at the MIT AI/CSAIL laboratory, is the experimental platform for this thesis work (see Appendix A). Past research works in Cog includes: robot arm control exploiting natural dynamics using neural oscillators (Williamson, 1999); creation of the basic structures required to give a "theory of mind" for the humanoid robot Cog, using mechanisms of shared attention (Scassellati, 2001); development of a deep perceptual system for Cog, grounding the acquisition of visual percepts in experimental manipulation (Fitzpatrick, 2003b); and sensorimotor learning via interaction with people and objects (Marjanović, 2003), which takes cognitive inspiration on the works in (Lakoff and Johnson, 1980; Lakoff, 1987; Johnson, 1987).

Kismet is a complex active vision robotic head capable of expressing several types of facial emotions for modulation of social interactions with caregivers. Kismet's computational system was designed by several elements of our group at the MIT AILab, and includes an attentional system (Breazeal and Scassellati, 1999) inspired from studies in experimental psychology (Wolfe, 1994); structures for turn taking (Breazeal and Fitzpatrick, 2000) between the robot and a caregiver, a motivational system (Ferrell, 1998); a babbling language (Varchavskaia et al., 2001); and recognition of affective communicative intent (Breazeal and Aryananda, 2000).

M2-M4 Macaco (Arsenio, 2003b,d) is a flexible active robotic head which can resemble aesthetically different robotic creatures, such as a chimpazee or a dog (see Appendix A). Macaco's brain includes processing modules for: object analysis, robot day and night navigation, and social interactions.

Other Humanoid Robotics Projects

A developmental model of joint attention, based on social interactions with a human caregiver, is described in (Nagai et al., 2003). Their model includes not only the development of the robot's sensing structures from an immature to a mature state, but also the simultaneous development of the caregiver's task evaluation criteria. They argue that the proposed developmental model can accelerate learning and improve the final task performance (Nagai, 2004).

Robovie is an interaction-oriented upper-torso humanoid robot assembled to a mobile platform. The robot head includes various sensorial modalities (includes visual, auditory and tactile sensors). The goal of this project is to develop a robot that communicates with humans and participates collaboratively in human society (Kanda et al., 2004; Ono et al., 2000).

Babybot is an eighteen degrees of freedom humanoid robot which includes an active vision head, a torso, one arm with one hand at its extremity. Babybot's sensors include a pair of cameras with space-variant resolution, two microphones and tactile sensors. The project's main goal is to investigate the functioning of the brain by

building physical models of the neural control and cognitive structures (Metta et al., 2000). They follow a developmental approach for building a sensorimotor system for a humanoid robot (Sandini et al., 1997; Metta et al., 2001). They also propose a biologically inspired functional model for the acquisition of visual, acoustic and multi-modal motor responses (Natale et al., 2002).

The Infanoid robot is an upper torso humanoid having roughly the same size and kinematic structure as a three-years-old child. This project aims at studying the mechanisms of social intelligence that enable a robot to communicate and socialize with humans (Kozima and Yano, 2001). The epigenetic principle is followed as a means to achieve social intelligence of robots (Kozima and Zlatev, 2000). Special emphasis is placed in investigating an early type of joint visual attention with a human caregiver.

The Humanoid robot Jack at the Electrotechnical Lab at Tsukuba/Japan got his name from Robin Williams's character in the movie "Jack". The character is the result of merging a 5-year-old boy's brain in the body of 40-year-old man (curiously, the same applies to Cog, since its body form is too big to be confounded for a child-like robot). This humanoid robot is applied as a research vehicle into complex human interactions, being user-friendly and design for safety interactions (part of the machinery is covered in thick padding). The humanoid robot DB, at ATR/Japan, is another platform built by SARCOS. This project is focused mainly on learning from imitation (Schaal, 1999; Nakanishi et al., 2003) and on-line learning of sensorimotor maps (Nakanishi et al., 2003; Gaskett and Cheng, 2003) for complex task execution (Schaal and Atkeson, 1994).

The Asimo robot is a very well designed mechanical platform built by Honda. However, up to now it has only very limited cognitive capabilities.

1.3 Road(Brain)Map

This thesis is organized according to functional structures of a biological human brain – the *brainmap*,

Chapter 2	**A Humanoid Brain**
Chapter 3	*Low-Level Visual Processing*
Chapter 4	*Primary Visual Association Area*
Chapter 5	*Visual Pathway – What*
Chapter 6	*Visual Pathway – Where*
Chapter 7	*Cross-Modal Data Association*
Chapter 8	*Memory and Acoustic Perception*
Chapter 9	*Sensory-Motor Area and Cerebellum*
Chapter 10	*Frontal Lobe*
Chapter 11	*Cognitive Development*
Chapter 12	**Toward Infant-like Humanoid Robots**

which corresponds to the following roadmap for the computational algorithms.

Chapter 2	Cognitive and developmental basis for the thesis framework
Chapter 3	Computational algorithms for low-level visual processing
Chapter 4	Learning structures for perceptual stimulis selection and grouping
Chapter 5	Visual object/face Recognition and the visual binding problem
Chapter 6	Learning about scenes, objects and people from contextual cues
Chapter 7	Cross-modal data processing from multiple senses
Chapter 8	Tracking and statistical storage of objects; sound recognition
Chapter 9	Sensory-motor structures for robot control
Chapter 10	Robot tasking grounded on perceived functional information
Chapter 11	Developmental learning for human-robot skill transfer
Chapter 12	Discussion, conclusions and future work

This thesis organization, according to the road(brain) map, is explained in greater depth in the following chapter.

Chapter 2

A Humanoid Brain

...many of the details of Cog's "neural" organization will parallel what is known about their counterparts in the human brain.

(Dennet, 1998)

A long term goal of Artificial Intelligence has been the understanding of the brain mechanisms of thought, and the emulation of such processes in an artificial manmade creature, such as a humanoid robot. This work presents our efforts at tackling this complex and challenging endeavor.

The thesis framework will be described from two different perspectives: cognitive and developmental. Each chapter will contain therefore an introduction documenting cognitive and developmental issues. This document is organized according to a cognitive mapping of the computational structures developed. On the other hand, the humanoid robot Cog will learn throughout this thesis as a child, and therefore is only able initially to perceive simple world events (such as a newborn during the autistic developmental phase (Arsenio, 2004a; Mahler, 1979)). Through the teachings of a human caregiver, the robot will then be able to learn developmentally about itself, objects, people with which it interacts and its surrounding world. Knowledge from the outside, unreachable world will also be learned by having a human caregiver teach the robot from books and other learning aids (Arsenio, 2004d).

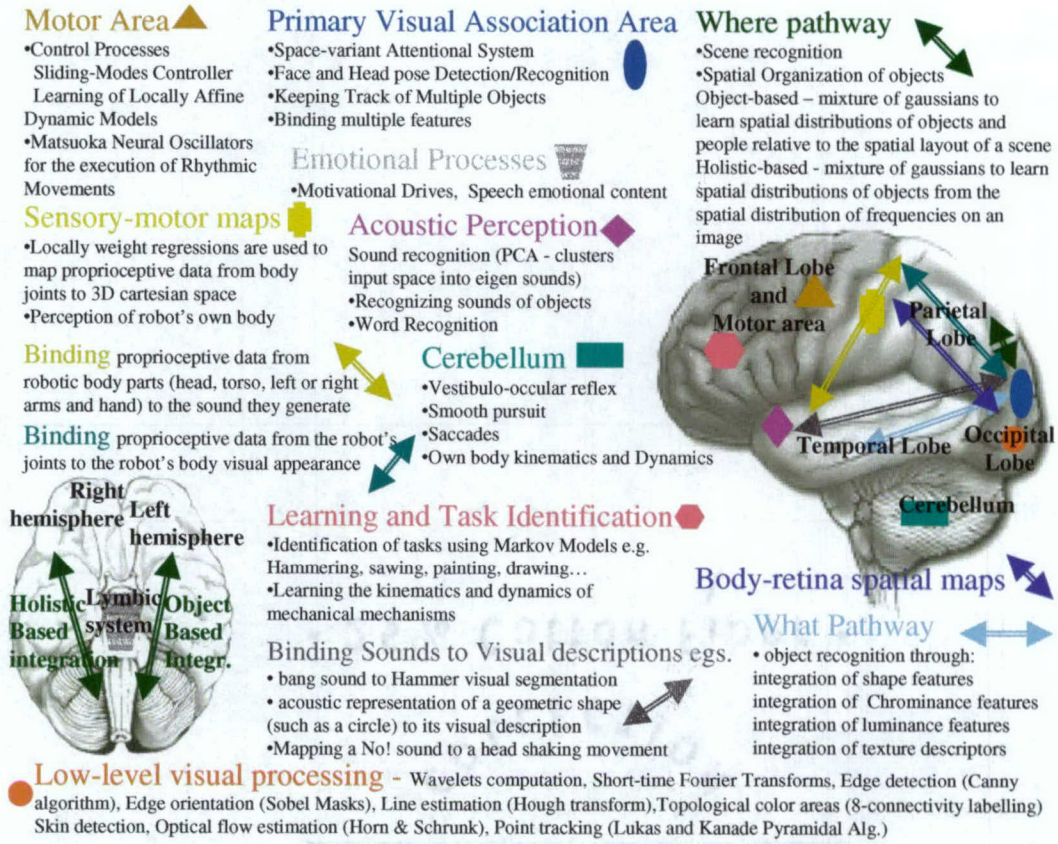


Figure 2-1: Cognitive modelling of a very simple brain.

2.1 Cognition

Several AI algorithms were implemented aiming at the emulation of different perceptual cognitive capabilities on the humanoid robot Cog (see figure 2-1). It is worthwhile stressing that this *simple artificial humanoid brain* is intended to emulate a real brain only at a higher level of abstraction. Furthermore, it is only a very simplistic approach to something as complex as the real brain of even simple creatures. Most functionalities still remain to be implemented, such as: language acquisition and synthesis; complex reasoning and planning; tactile perception and control; hand manipulation and high-level interpretation of human gestures, just to name a few.

Another point worth stressing is that although certain human brain areas are most responsible for specific cognitive processing, the brain is highly interconnected and a particular cognitive modality most often involves several brain structures. This complexity is reflected in the neural organization of the brain – it is estimated (Nieuwenhuys et al., 1980) that the human brain contains roughly 100 billion (10^{11}) neurons, with 1000-10000 synaptic connections for a "typical" neuron, which puts the number of synapses at the human brain in the order of quadrillions (10^{14}–10^{15}). On the

Low-Level Visual Processing	Chapter 3
Primary Visual Association Area	Chapter 4
What Pathway	Chapter 5
Where Pathway	Chapter 6
Binding Proprioceptive data to Sound, Binding Proprioceptive data to Vision, Binding Sounds to Visual Descriptions	Chapter 7
Memory, Acoustic recognition	Chapter 8
Motor Area, Sensory-Motor Maps, Body-Retina Spatial Maps, Cerebellum	Chapter 9
Task Identification, Emotional Processes (review)	Chapter 10
Cognitive enhancers and developmental learning of cognitive capabilities	Chapter 11

NOTE: Colors correspond to the ones in figure 2-1

humanoid robot, an example of such complexity results for building a description of a scene, which requires spatial information (the "where" pathway), recognition of specific world structures (the "what" pathway) and visual memory. Although most chapters will concentrate on a specific area of the humanoid robot's brain, functionalities often include other areas, and therefore it will be helpful to keep in mind a global perspective of the cognitive system. Hence, this work will consist on a large collection of distinct, but highly interconnected processes.

This thesis investigates therefore how to transfer some of the workings of such a complex system as the human brain to the humanoid robot Cog. The thesis will start by introducing a sensory modality important for human survival – vision – in chapter 3. Several low-level visual processing modules will be described for extracting a set of low level features: spectral, chrominance, luminance, and oriented lines. In addition, this chapter will describe a multi-scale approach for event detection from motion created by an agent's actions. Low-level features such as skin-color and optical flow are integrated by a logpolar attentional system for inferring stimulus saliency in the visual association area (chapter 4). Perceptual grouping of color and spectral regions also occurs chiefly in this area, together with depth inference.

Two parallel visual pathways in the human brain have been found to process information relative to objects/people identity (the "what" pathway, for which object and people recognition algorithms are described in chapter 5) and to objects/people location (the "where" pathway, for which map building and localization of objects, people and the robot itself are described in chapter 6).

The sensory information reaching the human brain is not processed solely based on data from the individual sensorial modalities, but it is also cross-correlated with data from other sensory modalities for extracting richer meanings. This processing in the human brain inspired the development of an approach for a humanoid robot to detect cross-correlations among data from different sensors, as described in chapter 7.

The effects of a visual experience are stored in the human brain for a posteriori use. Visual memory is considered to be divided into three differentiated systems (Palmer, 1999): Iconic Memory, Short Term Memory, and Long Term Memory (chapter 8). The algorithms developed in chapter 8 will have strong connections from and to the visual pathways (this strong connectivity is also pervasive in the human brain). But perception in the brain involves several sensory modalities. Processing of audio signals is one such modality which is also described in this chapter (sound recognition is often

a very important survival element on the animal kingdom).

Another perceptual modality is proprioception – position/velocity sensory data from the robot's joints (or from human bone articulations and muscles). These sensory messages terminate in the Parietal Lobe (on the somato-sensory cortex). This area has direct connections to the neighbor motor area, responsible for the high-level control of motor movements. Building sensory-motor maps (chapter 9) occurs early in childhood to enable reaching movements. In addition, networks for the generation of oscillatory motor rhythmics (networks of central pattern generators are also found in the human the spinal cord) are modelled by Matsuoka neural oscillators in chapter 9. The cerebellum plays a crucial role in the control of eye movements and gaze stabilization, also described in that chapter.

The frontal lobe is responsible for very complex functions in the human brain, and it has been shown that inputs from the limbic system play an essential role in modulating learning. Chapter 10 will only deal with the problem of task identification. Emotion systems implemented by our research group will also be reviewed.

Cognition in humans occurs developmentally. Inspired by human developmental mechanisms, chapter 11 presents cognitive enhancers and developmental learning of simple cognitive capabilities (although developmental learning was also exploited for cognitive elements from other chapters).

2.2 Developmental Perception and Learning

Robust perception and learning will follow the *Epigenetic Principle* (Zlatev and Balkenius, 2001) – as each stage progresses, it establishes the foundation for the next stages.

2.2.1 From Behaviorism to Scaffolding

Watson, the father of behaviorism, advocated that the frequency of occurrence of stimulus-response pairings, and not reinforcement signals, act directly to cause their learning (Watson, 1913). He rejected the idea that some mental representations of stimuli and responses needed to be stored in an animal mind until a reinforcement signal strengths an association between them.

Skinner argued against stimulus-response learning, which led him to develop the basic concept of operant conditioning. For Skinner the basic association in operant conditioning was instead between the operant response and the reinforcer. His Skinner-Box experimental apparatus improved considerably the individual learning trials of Watson.

Piaget gives equal roles to both nature (biological innate factors) and nurture (environmental factors) in child development (Piaget, 1952). In Piaget's theory, genes are the building blocks for development. He puts special emphasis on children's active participation in their own cognitive development.

Social-cultural-historical aspects are instead stressed by (Vygotsky, 1962, 1978; Bruner et al., 1966). They concentrate more on how adults help a child to develop

coherent knowledge concerning such aspects of the environment. Vygotsky's social-cultural-historical theory of cognitive development is typically described as learning by *"scaffolding"* (Vygotsky, 1978).

Learning by Scaffolding

New skills are usually socially transferred from caregivers to an infant through mimicry or imitation and contingency learning (Galef, 1988).

For a review of imitation in the fields of animal behavior and child development see (Dautenhahn and Nehaniv, 2001). Imitation in the field of robotics has also received considerable attention (Scassellati, 1998, 2001; Schaal, 1999).

In contingency learning, the simple contingent presence of the caregiver and the objects involved in the action provide the necessary cues for an infant to learn (Galef, 1988; Hauser, 1996; Dickinson, 2001). Whenever actions cause a change of state in the environment, infants detect a contingency – a relation between actions and the changes they produce in an environment – if they learn the relationship between the action and the environmental change (Leslie, 1982; Leslie and Keeble, 1987; Nadel et al., 1999). This way, infants can obtain positive or negative feedback rewards from their environments (and therefore they are able to prevent actions by which they receive a punishment). Contingency learning is also an essential ability for animals adaptation and survival. In the field of robot learning, it is often equated to reinforcement learning (Sutton and Barto, 1998) – for a review see (Kaelbling et al., 1996).

But mostly important to this thesis' work, caregivers also socially transfer abilities to an infant by means of scaffolding (Vygotsky, 1978). The term scaffolding refers to guidance provided by adults that helps a child to meet the demands of a complex task (Wood et al., 1976). The goal is to increase the chance of the child succeeding by making the task a little easier in some way. Examples of scaffolding includes the reduction of distractions and the description of a task's most important attributes, before the infant (or in our case, the robot) is cognitively apt to do it by himself (Wood et al., 1976).

This is the idea behind having a human caregiver facilitating a bit the chance of the robot to succeed, by:

- ▷ introducing repetitive patterns to the robot, as in chapter 3
- ▷ showing the features which compose the object, as shown in chapter 4 for perceptual grouping from human demonstration and further 3D reconstruction
- ▷ moving Cog's arm in front of a mirror to enable Cog's visual and acoustic perception of its own body (chapter 7)
- ▷ presenting toys that make repetitive noise, as in chapter 8
- ▷ presenting the robot objects from books, or from educational activities (chapter 11)

and by guiding in general all the learning tasks. As a matter of fact, all algorithms presented in this thesis make use within some degree of a "helping hand" provided by a human caregiver.

2.2.2 Mahler's Developmental Theory

This work draws inspiration not only from Vygotsky's learning by scaffolding developmental theory, but also and chiefly from Mahler's theory of child development (Mahler, 1979). Special emphasis is put on the child's *Separation and Individuation* developmental phase (Mahler, 1979) – during which the child eventually breaks the bound with his mother and embraces the external world. Mahler's autistic and Symbiotic developmental phases – characterized by the infant's simple learning mechanisms – antecede the Separation and Individuation phase (chapter 3). Mahler's theory has influences from movements such as the Ego's Developmental Psychology and Experimental Psychology, from psychologists such as Sigmund Freud, Piaget and others. According to her theory, the normal development of a child during the Separation and Individuation phase is divided into four sub-phases, following the *epigenetic principle*:

Differentiation (5-9 months) The first sub-phase, marked by a decrease of the infant's total dependency on his mother as the former crawls further away. The infant starts to realize his own individuality and separateness due to the development of the entire sensory apparatus and therefore a growing awareness.

Practicing (9,10-18 months) Sub-phase characterized by the child's active locomotion and exploration of his surroundings, together with the narcissist exploration of his own functions and body.

Re-approximation (15-24 months) Child has an egocentric view of the world during this phase, in which he also approximates again to his mother. World expands as new viewing angles are available from the child's erect walking.

Individuality and Object Constancy (24-36 months) Defined by the consolidation of individuality, and a clear separation between objects and itself. Towards the end, the child becomes aware of object constancy.

The child's *Separation and Individuation* phase (Mahler, 1979) is marked by the separation of the child from his mother as a different individual. However, the child still relies heavily on help provided by his mother to understand the world and even himself through this developmental phase (Gonzalez-mena and Widmeyer, 1997). Indeed, the child is part of a structured world that includes the immediate emotional, social and physical surroundings (Michel and Moore, 1995).

During this phase, the child learns to recognize itself as an individual, and its mirror image as belonging to itself. He learns also about the surrounding world structure - about probable places to find familiar objects (such as toys) or furniture items. In addition, he starts to identify scenes - such as his own bedroom and living-room. And

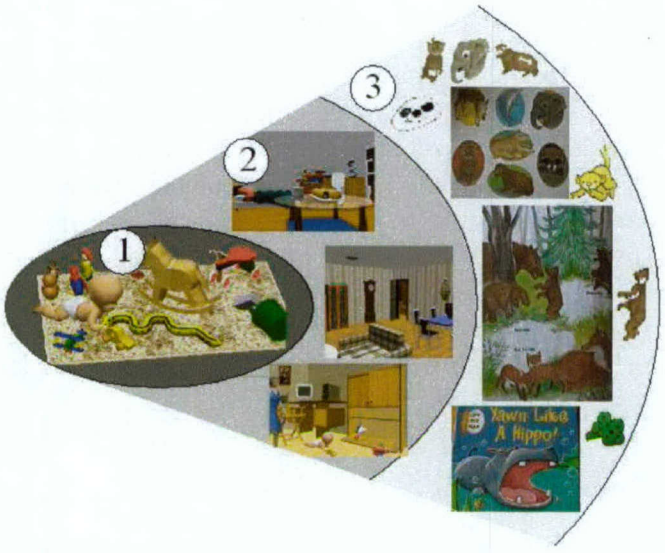


Figure 2-2: Developmental learning during the child's *Separation and Individuation* phase will be described along three different topological spaces: 1) the robot's personal space, consisting of itself and familiar, manipulable objects; 2) its living space, such as a bedroom or living room; and 3) its outside, unreachable world, such as the image of a bear in a forest.

children become increasingly aware (and curious) about the outside world (Lacerda et al., 2000). The implementation on the humanoid robot Cog of these cognitive milestones will be described throughout this document, placing special emphasis on developmental object perception (Johnson, 2002) during the Separation and Individuation stage.

The child's mother (or primary caretaker) plays an essential role (Gonzalez-mena and Widmeyer, 1997) in guiding the child through this learning process. With the goal of teaching a humanoid robot like a child, the child's mother role will be attributed to a human tutor/caregiver. Hence, a human-centered approach is presented to facilitate the robot's perception and learning, while showing the benefits that result from introducing humans in the robot's learning loop. Help from a human tutor will therefore be used to guide the robot through learning about its physical surroundings. In particular, this `"helping hand"` will assist the robot to correlate data among its own senses (chapter 7); to control and integrate situational cues from its surrounding world (chapters 6 and 8); and to learn about out-of-reach objects and the different representations in which they might appear (chapter 11). Special emphasis will therefore be placed on social learning across a child's physical topological spaces, as shown in figure 2-2.

2.2.3 Human-Robot Skill Transfer

This thesis will place a special emphasis on incremental learning. To accomplish this goal, a human tutor performs actions over objects while the robot *learns from the demonstration* the underlying object structure as well as the actions' goals.

This leads us to the *object/scene recognition* problem. Knowledge concerning an object will be organized according to multiple sensorial percepts. Objects will also be categorized according to their functional role (if any) and their situatedness in the world.

Learning *per se* is of diminished value without mechanisms to apply the learned knowledge. Hence, *robot tasking* will deal with mapping learned knowledge to perceived information (see figure 2-3).

Learning from Demonstration

We want the robot to learn autonomously learning information about unknown objects. With this in mind, our strategies will include simple actions such as grabbing or poking the object to learn its underlying structure. To support a mechanism for learning from human demonstration, we work on learning the actions that are applied to an object, or even between two actuators (such as clapping). Learning aids, such as books, will also be used as another source of information that can be transmitted to a robot through a human.

Object/Scene Segmentation and Recognition

Objects have complex uses, come in various colors, sound differently and appear in the world in ways constrained by the surrounding scene structure. Since these object properties are multi-modal, multiple sensorial input have to be processed. Each individual object property is important for recognition, but if more than one communication channel is available, such as the visual and auditory channels, both sources of information can be correlated for extracting richer pieces of information. For instance, cross-modal information can disambiguate object identity in cases where one perceptual modality alone could fail. This thesis demonstrates how to take advantage of multiple perceptual information.

Object and scene recognition constitutes a very important piece of the thesis framework core. Our approach is to cast the object recognition problem as an embodied and situated one. Through the action of an embodied agent on an object, it is possible to infer what pieces of an object move, how do they move, how to make them move, and why do they move. Therefore, the object recognition problem is the result of several highly interconnected problems.

An important concept in object recognition is object appearance. We humans often classify objects by their aesthetics. The visual appearance of an object is embedded in object templates (by templates we mean images with black background and the visual appearance of the object), which are segmented from an image. Embodied object segmentation may work in cases where disembodied approaches typically often

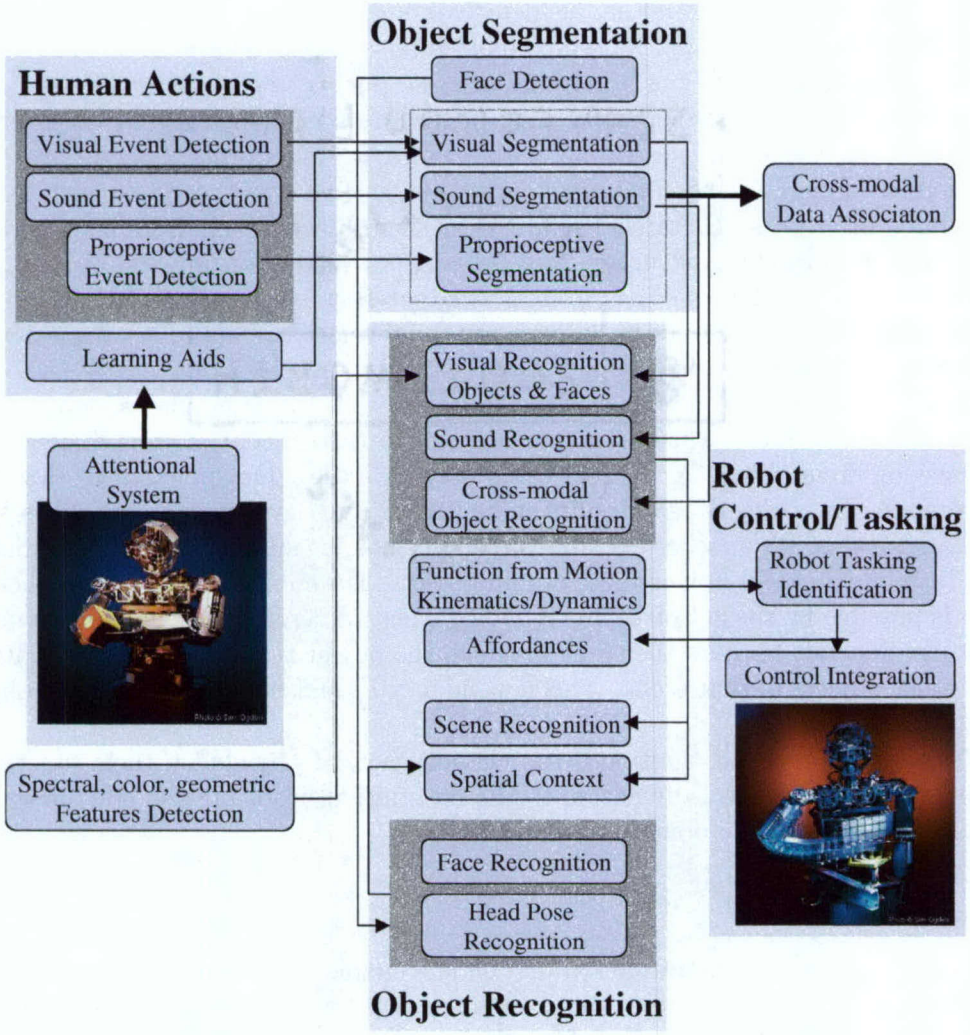


Figure 2-3: From human demonstration towards robot tasking. Four main groups of modules of the thesis developmental framework are shown, together with some other modules. The sequence includes learning from **Human Demonstration** – learning an object's underlying structure from demonstration. This is followed by **object segmentation**. **Object recognition** updates knowledge concerning an object. Finally, the **robot executes tasks** according to its embedded knowledge of the object.

fail. For instance, objects with a similar appearance as the background are distinguished. In addition, objects with multiple moving links can be segmented as such using the embodied approach, each link being assigned to a different template. Object templates constitute an important input for the object recognition scheme. The recognition mechanism should handle defective templates and classify them correctly as an object in the database, for which more robust templates are available.

The physical properties of a material are also put to test by embodied action on the object. Indeed, a strong strike may break an object (such as two *Lego* bricks assembled together) into individual parts, which are also objects themselves. In addition, a human teacher may demonstrate some tasks to the robot such as the assembling of objects. Hierarchal relations among objects result from such actions. Therefore, a hierarchical structure will be adopted for organizing the objects in a database. such a structure differs from the **object-files** – memory representations of objects used to keep track of basic perceptual information – approach suggested by Treisman (Treisman, 1988).

Another important component of my approach is object situatedness – objects are situated in the world – which is not orthogonal to the concept of embodiment. For instance, different objects in the real-world may have the same appearance. However, information concerning the scene surrounding an object (the context) often solves such ambiguous cases. As another example, an empty box is usually used to store smaller objects, while a closed box cannot contain objects. Disambiguation between the two cases is possible by the action of an embodied agent. Therefore, the categorization of an object depends both on the scene in which the object is inserted and its relations with other objects in that scene. This dependency should be embedded on an object recognition scheme.

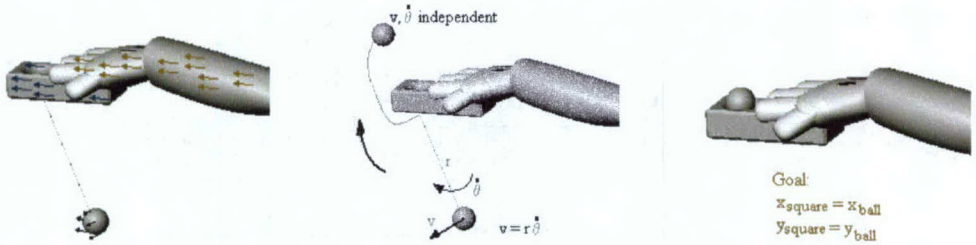
Functional object recognition - the classification of objects by their function - although adding an extra dimension to the recognition problem, also adds new constraints that narrow the domain of search.

Robot Tasking

The learning from demonstration strategy should enable the detection and recognition of a large array of complex dynamic systems, as well as the identification of several tasks. The range of applications for robot tasking were narrowed to very simple domains, because of mechanical constraints on Cog.

A Dynamic System Estimator will be used to estimate the desired accelerations of the robotic arm end-effector, while Robot Control is executed through an Adaptive Sliding Mode controller (Slotine and Weiping, 1991). A learning strategy will also be implemented to estimate the robotic arm nonlinear dynamics, which is modelled by a collection of locally linear models.

Finally, I will build a hybrid state machine for identifying **goals**. Actions are mapped to discrete states of this machine, while the kinematic/dynamic model is mapped to continuous states. Goals will correspond to stationary or oscillatory signals emitted by the discrete or continuous states. Figure 2-4 illustrates a goal corresponding to a stationary signal from a continuous state.



a) b) c)

Figure 2-4: a) The motion segmentation consists of segmenting the image optical flow into segments of coherent motion, which enables b) the estimation of the kinematic constraints among the several segments, which may vary for non-linear systems, as illustrated - the ball moves as a pendulum with three translational constraints in a region of the state-space, and as a free-moving object in another region. c) Task goal given by stationary state.

2.3 The Movie in a Robot's Brain

One important feature of this thesis work lies on the integration of several complex cognitive modules in a humanoid robot. The temporal collection and integration of all cognitive percepts in the human brain is often denoted *the movie in the brain* (Damasio, 1994, 1999). It is worth introducing here several snapshots of what is going on at Cog's *brain* at a given moment. Namely, how are memories stored? what information is interchanged between processing modules? In the following we try to answer these two questions[1].

2.3.1 What gets into the Robot's *Neurons*?

This thesis framework assumes no single, central control structure. Such assumption is supported by evidence from the cognitive sciences (mainly from brain-split patients at the level of the corpus callosum (Gazzaniga and LeDoux, 1978)). Cog's computational system comprises several dozens of processes running in parallel in 32 processors (see Appendix A) and interchanging messages continuously[2].

Storage at Low-level Visual Structures

Low level visual processing modules (chapter 3) do not really store any information relevant for learning. Processing outputs of low level features are basically sent to other processing units for further processing (see figure 2-5), as follows. Color, geometric features (oriented lines fitting contours), spectral components and motion are

[1]Integration of all the algorithms presented throughout this thesis will be described in full detail in this section.The reader should consult it whenever a need for better understanding of the global picture arises, i.e., how a given algorithm integrates within all the framework.

[2]Throughout this thesis, experiments are presented for different large groups of algorithms running together. This happens because the computational power, although big, was not enough to get all modules running together in real-time.

components required by other high-level processing structures.

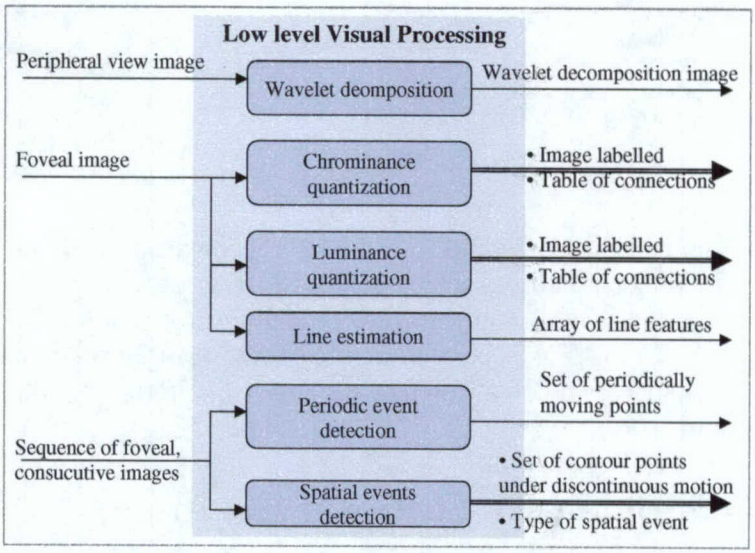


Figure 2-5: Processing at low-level visual structures. Each algorithm processes data independently. Output from these structures are the input for higher-level computations at the visual association area, visual pathways and other *brain* structures.

Storage in the Visual Association Area

The algorithms implemented to associate visual information do not perform any significant high-level learning. Instead, they act as filters, processing the information received and sending it to other modules (see figure 2-6). These algorithms correspond to active segmentation strategies, perceptual grouping from human cues and inference of 3D information for objects, as described in chapter 4. Some low-level storage is necessary, since individual points in the image tracked over a temporal sequence are stored for very short periods of time ($\sim 2-4$ seconds). But this storage is not relevant for learning percepts for future use. The perceptual grouping algorithm also keeps in storage the last stationary image detected by the robot.

Storage at the "What"-Visual Pathway

A very important percentage of the machine learning component is performed by these structures (see figure 2-7). Indeed, visual information is processed along the human brain's "what" visual pathway for categorical identification. Similarly, computational algorithms, as presented by chapter 5, process visual information to categorize world elements and for a posteriori recognition in incoming images. But in order to accomplish such identification, these algorithms need to store some sort of representation to a given object category, as follows.

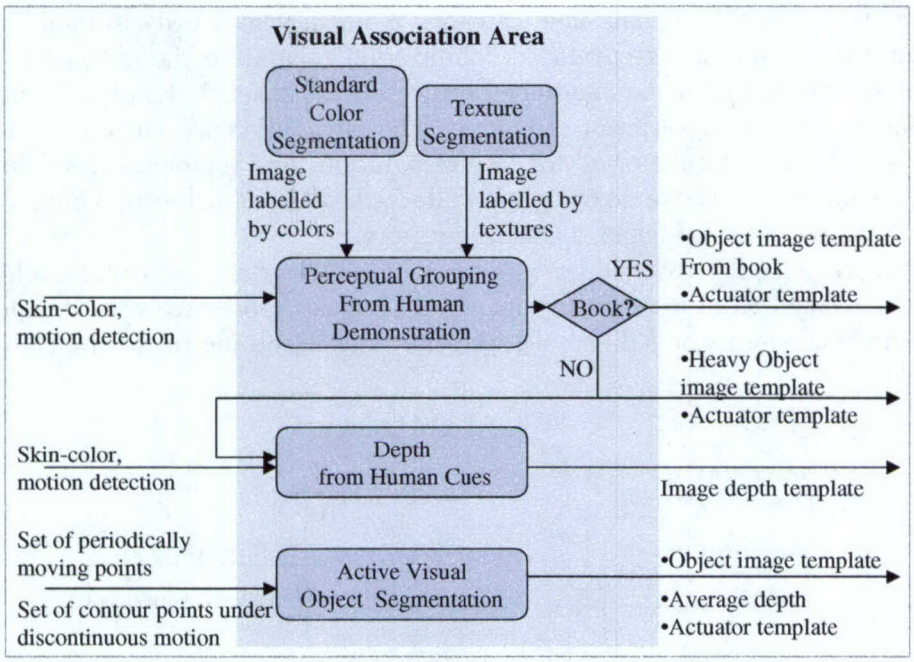


Figure 2-6: Processing at visual association structures includes mostly grouping of perceptual information and object segmentation. Since the perceptual grouping algorithm is applied both to detect heavy objects in a scene as well as to segment objects from books, both classes of categories are differentiated using the robot personal working space, as follows. Whenever the robot is gazing at his workbench upon stimuli receival and detection, it processes the segmentation as not coming from real objects (and therefore no depth measure is extracted). Otherwise, it is assumed that the robot is looking at a real object in a scene, and therefore the object shape is reconstructed. Although we tried to avoid human designed rules whenever possible, such a constraint was useful for this particular case.

Object recognition based on matching object templates needs to store percept representations extracted from several image templates of a given object category. Hence, the algorithm stores and updates 20 average color histograms (the aforementioned percept representations), which represent different object views. A new color histogram of an object template correctly matched to an average color histogram updates the later through an average process.

Face recognition based on matching face templates (from a face detector) stores a set of n face templates for each object category (whereas n maximum was set to 800). Therefore, a face database will consist of a collection of sets of faces. This modules stores as well the average face and the eigenfaces for each face category. For each face in the storage database, the coefficients originated by projecting such faces into the eigenfaces subspace are also saved. Whenever a batch of templates arrives for recognition, it is saved into the computer disk (if $n < 800$). The category

matched to the batch (or the new category if none is matched) is then updated by computing again the eigenvalue decomposition to update the average face and eigenfaces, together with the mentioned projection coefficients. Head pose inference stores exactly the same type of elements. The only difference lies on the type of images saved for each category – for face recognition, face templates from the same person are labelled to the same category, while for head face inference, templates from similar head poses are assigned the same category.

The more general *geometric-hashing based object recognition* algorithm, which detects and recognizes objects in any image of a scene, processes three independent inputs, and therefore stores different representations according to the input.

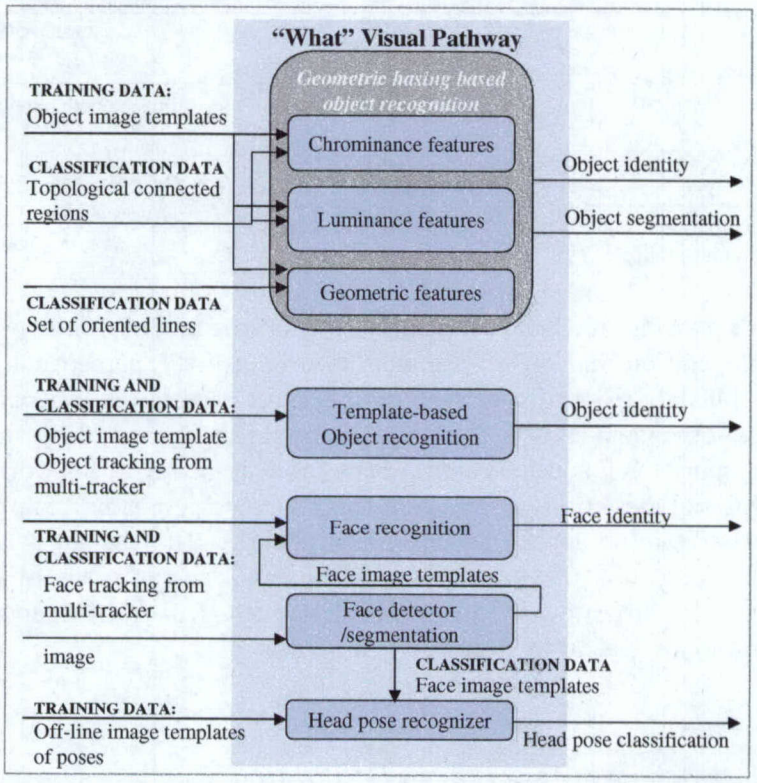


Figure 2-7: Computational processing at the "what" visual pathway. It includes computational structures for recognizing objects (both from templates and arbitrary scenes), faces and head poses. Training data for the machine learning algorithms is generated semi-autonomously (in case a person is generating percepts by acting on objects) or autonomously (if it is the robot acting by itself).

Storage at the "Where"-Visual Pathway

Algorithms for the visual localization of people, objects and the robot itself (robot localization by recognition of scenes) in the "where" pathway (chapter 6) have ex-

tensive connections with the computational structures in the "what" pathway (see figure 2-7).

Map Building from Human Contextual Cues builds an appearance mosaic image of all objects in a scene, together with a mosaic image of the scene depth. Hence, it needs to store the appearance image template of the last template acquired for a scene object (from visual association structures), together with the associated depth image template (computed from a low level visual processing module). In addition, these templates need to be related to each other. This is done by saving the egocentric coordinates of each template centroid (obtained by mapping retinal coordinates into $2D$ head gazing angles in the sensory-motor area).

This algorithm also stores, for each scene, links to all objects in such a scene. And for each object, links to all scenes where it might appear. Each link consists of an identifier code which uniquely indexes the object (or the scene) in the corresponding recognizer. This is done every time the algorithm assigns an object to a scene.

Scene recognition stores coefficients of principal component analysis (PCA) applied to the wavelet decomposition of input images from a scene. For each scene category, the algorithm stores a set of D-dimensional coefficients vector, each vector corresponding to an image of the scene. In addition, a set of n images ($n <= 800$) are also saved for each scene category. Each of such images result from wavelet decomposition, being the grouping of the wavelet coefficient images. Therefore, a database of scenes consists of a collection of sets. Each set contains n wavelet transform coefficient images and n vectors of coefficients from the PCA.

Whenever an image from a wide field view of the scene arrives, the output of the wavelet transform applied to it is saved into the computer disk (if $n < 800$). The category which is matched to (or the new category if not matched) is then updated by computing again the PCA, and hence estimating a new covariance matrix for the D-dimensional vectors of coefficients for such scene. Finally, the mixture of gaussians which models the scene distribution is optimized by running again the EM algorithm. Hence, the scene database also contains, for each scene category, the parameters of the mixture of gaussians: the number of gaussian clusters, and for each gaussian the average and covariance of the cluster, together with the gaussian's weight.

Object and people recognition from contextual features and scene recognition processing structures store similar data. The main differences is that categories correspond now to people or objects, instead of scenes. Hence, the algorithm stores for each category a set of PCA coefficients applied to the wavelet decomposition of input images of objects or people, and a set of n wavelet decomposition output images ($n <= 800$). The database also contains, for each category, the parameters of a mixture of gaussians, being each category cluster modelled by a product of gaussians.

Storage at *Memory* areas

The operational boundary between the memory algorithms (chapter 8) and the "where" visual pathway computational structures is not well defined (see figure 2-8). But the memory algorithms do not rely on visual contextual cues. These algorithms instead keep tracking of where objects are according to short or long term past experiences.

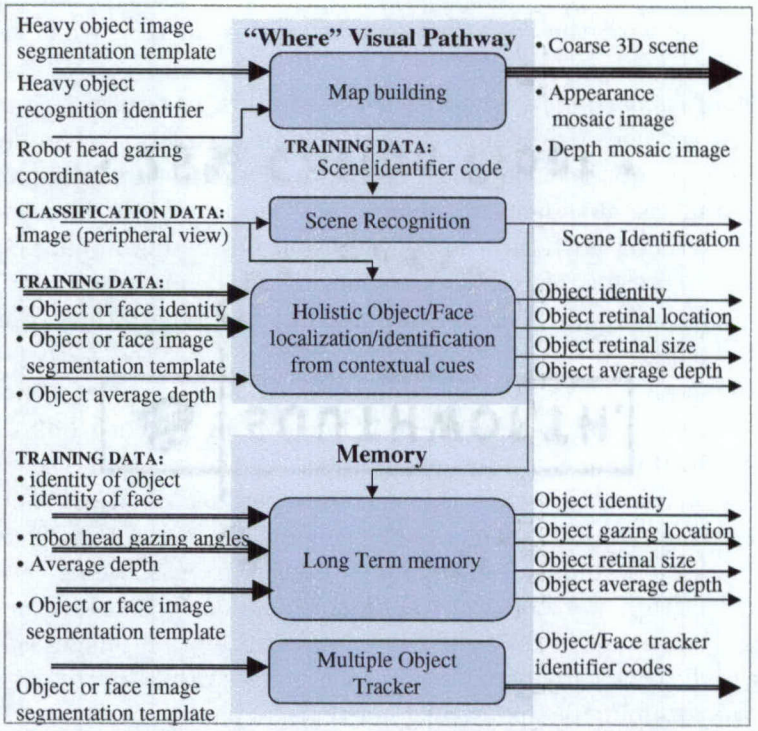


Figure 2-8: Computational processing at the "where" visual pathway and memory structures. Training data for the learning algorithms is generated semi-autonomously from human introduced cues and from data arriving from the "what" visual pathway.

The multiple object tracker algorithm plays a very important role in the process of automatic annotation of data for the machine learning algorithms. This algorithm stores solely short-term information, in the form of object/people trajectory tracking, to maintain information about objects and people identity as they move from place to place. Therefore, it keeps on *short-term memory*, for each object/people being tracked, the centroid position for the last 2 seconds, together with the current position of tracked object points.

Long term memory information concerning people and objects and relations among them is stored in egocentric coordinates, as follows. Probable object/people egocentric locations are modelled using a mixture of gaussians for each object/people category. Data from all previous locations of a category element, given as the robot's head gazing angles, the object/people size template, orientation and depth are stored, together with the parameters of the group of gaussians that model such distribution. This gives the robot a memory of where usually objects or people are expected to be found.

Since such memories are often influenced by contextual information given by the presence of other objects or people, inter-objects/people interference is accounted by modelling probabilities for finding elements of a category given the presence of

elements from other category. This requires the storage, for each pair of relations (for n categories of people and objects, there are $n^2 - n$ such possible pairs), of the parameters of the mixture of gaussians that model such interrelations. However, in order to place an upper bound on the necessary storage, such interrelations are determined solely for elements belonging to the same scene. One element of each scene from a special category (a fixed or heavy object) is selected as a gate, and for m scenes $2m$ interrelation links between these gates are also stored.

Storage at the Auditory Processing Area

Segmentation of periodic sounds (chapter 7) requires solely low-level, short-term memories of signal samples over a time window. The *sound recognition* algorithm (chapter 8) stores exactly the same type of percepts as the visual face recognition module. The only difference lies on the type of images saved for each category. For face recognition, face templates from the same person are labelled to the same category, while for sound recognition, image templates built from sound segmentations are assigned to the same category (see figure 2-9).

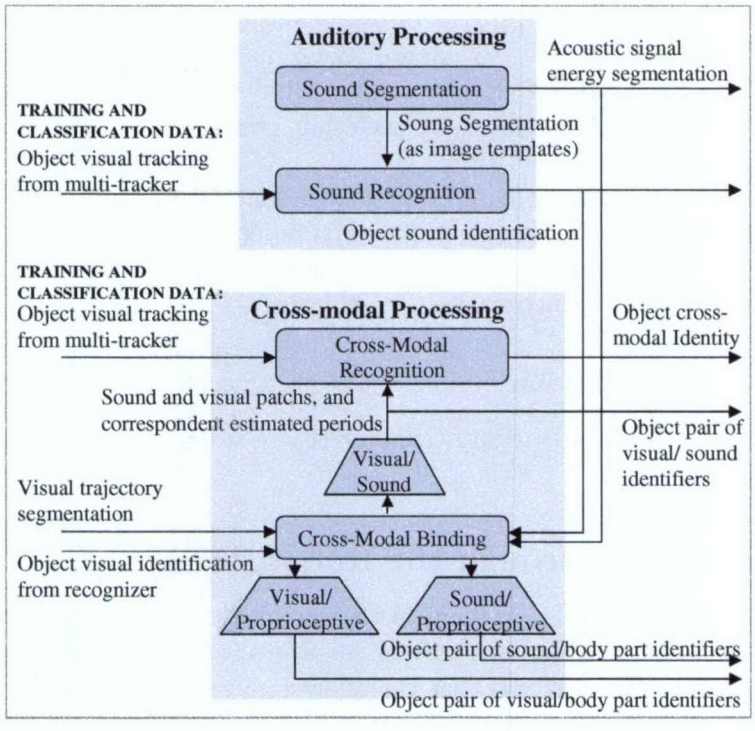


Figure 2-9: Auditory and cross-modal processing.

Storage at Cross-Modal Processing Structures

Information from multiple perceptual modalities (chapter 7) is integrated together as shown in figure 2-9.

The *Cross-modal association* algorithm stores the links between the cross-modal perceptual modalities, as follows. For a visual-sound binding, it stores the pair of identifier codes which uniquely indexes the object and sound categories in the corresponding recognizers. This is done every time the algorithm assigns the visual representation of an object to a sound. For a visual-proprioception or sound-proprioception binding, it stores the visual or sound identifier and a code which uniquely defines the body part matched.

Cross-modal object recognition requires the storage of the algorithm training data. Hence, $n \leq 800$ cross-modal features are stored for each object category, together with the parameters of a mixture of gaussians which models the data distribution. In addition, the identifier of the object category in the object recognizer is also stored for each cross-modal category, if available.

Storage in Sensory-Motor Areas and Cerebellum

A large amount of data is stored for learning kinematic and dynamic maps which are necessary to control the humanoid robot (chapter 9). *Sensory-motor mappings* include the head and arm forward/inverse kinematics maps, the visuo-motor maps and the Jacobians (see figure 2-10). Hence, these maps are built from the storage of thousands of training samples. Each sample includes the head/arm joints position and velocity, as well as the head/arm cartesian position and orientation, and the corresponding velocities, respectively.

A short history of errors is required by motor controllers (PD for eye control and sliding mode for arm control) for tracking error compensation. Periodic proprioceptive data is segmented and stored in disk, together with identifiers which uniquely define the joints which generated such data.

Storage in the Frontal Lobe

Markov chains for each task category (chapter 10)are stored by the task identification algorithm.

2.3.2 What goes through the Robot's *Sinapses*?

Perhaps the most challenging problem in the integration of complex systems is the interconnection of multiple modules – which messages need to be transmitted between the computational structures – so that the all system produces coherent behaviors.

Inputs from Low-level Visual Processing – Chapter 3 –

Signals to the Visual Association Area – Chapter 4 –: The attentional system integrates a set of low-level, pre-attentional features, arriving from low-level visual

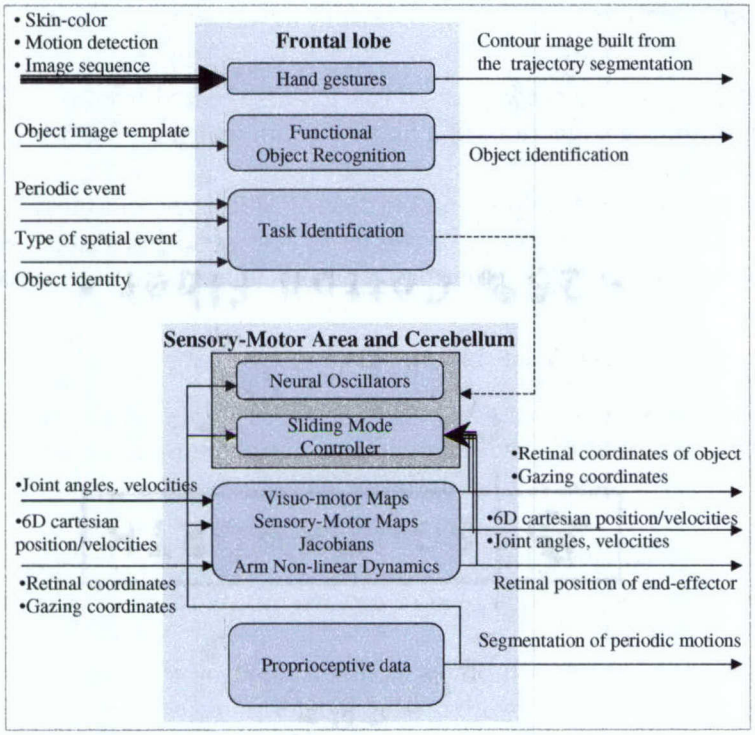


Figure 2-10: Computational algorithms corresponding to the Sensory-Motor areas, Cerebellum and Frontal lobe.

processing modules, into a saliency map.

Information concerning the image frequency spectrum, as computed by the wavelets transform, is the texture segmentation algorithm's input.

Events created by human actions (or by the robot itself) on objects at the low level visual processing modules trigger the transmission of these objects' moving points. Such data is received by active segmentation structures for object segmentation.

Signals to the "What"-Visual Pathway – Chapter 5 –: Topological connected color regions are sent into a general geometric hashing object recognition scheme for classification. These connected regions are represented by a template image, being each color region masked by an individual code, and by a binary table which stores which regions are connected to each other. A set of oriented lines in an image, which is represented by a multidimensional array of line center and end-points foveal retinal coordinates, and the line orientation angle, is also the classification input of the geometric hashing object recognition scheme based on geometric features.

Signals to the "Where"-Visual Pathway – Chapter 6 –: An holistic representation of a scene (also called the "image sketch") is computed from contextual cues. These cues are determined from the image's wavelets decomposition. They are then applied

to recognize and locate objects and people, and for scene recognition.

Signals to the Frontal Lobe – Chapter 10 –: Events created from periodic or discontinuous motions are the transitions of a markov chain which models the correspondent task being executed.

The image regions given by the intersection of Skin-tone and moving regions over a sequence of images are the input of the hand gestures algorithm, which segments arm/hand periodic trajectories, creating a contour image with such data.

Inputs from the Visual Association Area – Chapter 4 –

Signals to Low-level Visual Processing – Chapter 3 –: The detector of spatial events receives feedback information concerning object template masks, used to initialize the kalman-filtering based approach for updating the object's mask.

Feedback Signals to the Visual Association Area – Chapter 4 –: Some visual association computational algorithms send or receive inputs to/from other algorithms in the same area, respectively.

The boundaries of object templates as determined by the figure/ground segregation modules can be optimized by attracting a deformable contour, initialized to these initial boundaries and let converge to contours within the object template. The final boundaries are then filled to correct the object mask creating a new object template.

Perceptual grouping works by grouping colors of a stationary image. Such information is transmitted by a standard color segmentation algorithm (Comaniciu and Meer, 1997). Perceptual grouping also operates on textures computed for the stationary image. These textures are processed by the texture segmentation algorithm, which sends an image labelled by textures for the a posteriori perceptual grouping of these textures.

Signals to the "What"-Visual Pathway – Chapter 5 –: Object segmentations from both active segmentation and perceptual grouping algorithms are good perceptual elements to generate training data for visual object recognition algorithms. Color quantization of image templates (three channels image partitioned into 8^3 groups of colors) is both the classification and training input to the object recognition algorithm from object templates. Face recognition receives instead, after detection, face templates cropped from a scene image.

Training data for the general geometric hashing based object recognition scheme receives segmented object templates. It extracts chrominance, luminance and geometric features from such templates to train three recognition algorithms based on these three independent features.

Signals to the "Where"-Visual Pathway – Chapter 6 –: Images of objects and people segregated from the background are processed to extract the 2-dimensional retinal size of the template and its orientation. The object average depth is also receive from the

visual association area, computed from human introduced cues. These features are required to annotate training data for object and face localization from contextual features.

Object segmentations from both active segmentation and perceptual grouping algorithms are required for building descriptions of scenes (in the form of an appearance mosaic). Depth measures, in the form of depth templates (similar to disparity maps) are necessary as well to build both depth scene maps and 3D scene representations.

Signals to Memory Structures – Chapter 8 –: The multiple object tracker algorithm receives image templates from both object and face segmentation algorithms. If a new template matches the location of an object being tracked, such object template is updated and tracker points inside it are re-initialized. Otherwise, a new tracker identifier is also assigned to it.

Long-term memory structures, which store properties of objects and people, receive data as well from these segmentation algorithms, namely the template 2-dimensional retinal size and orientation, and the object depth.

Signals to the Cerebellum and Sensory-Motor Area – Chapter 9 –: The attentional system outputs a saliency map. The spatial coordinates corresponding to the maximum activation of this map follows to the eye control algorithms for the control of eye movements, if not inhibited by signals from other modules.

Signals to the Frontal Lobe – Chapter 10 –: Moving points initialized inside object templates are tracked, to extract functional constraints from their trajectories. This is used for identifying function from motion.

Inputs from the "What"-Visual Pathway – Chapter 5 –

Once the robot learns about object and people categories, their identification is used for further processing by other modules.

Signals to the Visual Association Area – Chapter 4 –: Image features detected from a scene by low-level visual processing structures (namely chrominance, luminance and geometric features), are matched to internal object representations of the geometric hashing based object recognition algorithms. After recognition, the matched scene features are grouped together. This additional grouping provides extra segmentation templates for the recognized object (for one example see figure 11-15). This closes one segmentation-recognition learning loop.

Feedback Signals to the "What"-Visual Pathway – Chapter 5 –: Feedback loops within the some processing areas, especially those responsible for machine learning, are pervasive, even if not explicitly. Indeed, upon receival of a new percept, recognition algorithms update the learning parameters for the category to which the new percept is matched.

Signals to the "Where"-Visual Pathway – Chapter 6 –: Both object and face recognition identifiers are required for annotation of training data of the object/face recognition algorithm from contextual features. Head pose information, estimated along the "what" visual pathway, would be extremely useful as an additional property for this "where" pathway algorithm. Although the extension is of trivial implementation – just increasing the dimension of the vector of object properties with this new input – this remains however relegated for future work.

Object identification is also requested to build scene descriptions.

Signals to Memory Structures – Chapter 8 –: The identity of both objects and faces is required to annotate training data for the object-based object/face recognition, which stores long-term information about object properties.

Signals to the Frontal Lobe – Chapter 10 –: Object identification codes are sent to the task identification algorithm.

Inputs from the "Where"-Visual Pathway – Chapter 6 –

Signals to the Visual Association Area – Chapter 4 –: Probable places in an image whether to find an object have a lot of potential to constrain the search space where to find an object, especially if estimations of its size and orientation are available, as is the case. This can be used to extract a very rough segmentation of the object, as shown by figure 6-8.

The algorithm that builds description of scenes sends wide-field-of-view images of a scene, annotated by the scene identifier, for low-level structures (wavelet decomposition module) for spectral analysis in order to build an holistic representation of the scene.

Signals to the "What"-Visual Pathway – Chapter 5 –: By constraining the space of possible locations of an object, the geometric hashing based recognition algorithm can be made more efficient.

As we have already stated, all the computational structures are highly interconnected. Hence, algorithms operating on contextual features output the identification of scenes, objects and faces. Hence, the boundary between the two visual pathways is somewhat blurry.

Feedback Signals to the "Where"-Visual Pathway – Chapter 6 –: Annotated data for scene recognition is created by the algorithm which builds description of scenes. Such algorithm attributes a scene identifier to wide-field-of-view images acquired for such scene.

Signals to the Cerebellum and Sensory-Motor Area – Chapter 9 –: Commands received for specifying desired head postures or head gazing configurations have priority over all the others.

Places constrained by environment structure to be of high probability for finding an object are good candidates for a robot to drop or store objects after manipulating them. Although the theoretical framework required for such physical object storage was developed in this thesis, experiments for testing such conjecture are yet to implement.

Inputs from Memory Structures – Chapter 8 –

Signals from memory structures – which keep track of object properties (such as location or depth) – are widely disseminated as inputs to several computational structures.

Signals to the "What"-Visual Pathway – Chapter 5 –: Both face and object recognition algorithms receive a signal from the multiple object tracker algorithm with a tracking code identifier, to relate the identity of segmentations arriving over a temporal window. This way, one is able to group a batch of samples belonging to the same category. Indeed, the multiple object tracking algorithm plays a very important role in the on-line creation of annotated data for the machine learning algorithms used for recognition.

Signals to Auditory Processing – Chapter 8 –: Visual object tracking output is used to annotate training data for sound recognition. Hence, an object identifier from the multi-tracker algorithm (immutable while the object is being tracked) is sent to this processing module for grouping sound segmentations into training data.

Signals to Cross-modal Processing – Chapter 7 –: Training data for cross-modal object recognition is also annotated by the output of visually tracking an object. In addition, the object tracking identification code is also required to store the various modal categories upon a successful binding.

Signals to the Cerebellum and Motor Area – Chapter 9 –: Modules which can suppress eye commands from the attentional system include the attentional tracker (Fitzpatrick, 2003a) and the multiple object tracking algorithm (chapter 8). Both send the position of a tracked object as a control command to the eyes.

Signals to the Frontal Lobe – Chapter 10 –: Object tracking identification codes are sent to the functional object recognition algorithm.

Inputs from Auditory Processing – Chapter 8 –

Signals to Cross-modal Processing – Chapter 7 –: Patches of sound signal, if locally periodic, are segmented and sent for cross-modal data association or recognition. In addition, the output of the sound recognizer is also sent to the cross-modal processing modules to keep track of modal categories during data association.

Feedback Signals to Auditory Processing – Chapter 8 –: Spectograms of the auditory signal are built by applying the Fourier transform to windows of the audio signal. This creates a two dimensional signal (frequency vs. temporal information) which is used for sound segmentation.

Sound templates, available from the sound segmentation output, are the training data of the sound recognition algorithm.

Inputs from Cross-Modal Processing – Chapter 7 –

Feedback signals to Cross-modal Processing – Chapter 7 –: Training data for cross-modal object recognition is created upon a visual-acoustic binding.

Inputs from Sensory-Motor Areas and Cerebellum – Chapter 9 –

Signals important for visual association processing: In order to actively segregate objects from their background or to group object features, it is necessary to stabilize the perceptual interface. This is done by the creation of visual stimulus which produce a saliency in the attentional system towards the desired object. The robot head is then driven towards such salient point and stabilized. However, this happens implicitly. No signal is explicitly sent from sensory-motor areas or cerebellum to the visual association area.

Signals to the "What"-Visual Pathway – Chapter 5 –: Proprioceptive data is applied for a very special type of object recognition – self-recognition.

Signals to the "Where"-Visual Pathway – Chapter 6 –: Construction of both scene appearance and depth image mosaics is implemented by mapping individual object templates in a scene to a referential frame. Such mapping requires transforming image pixels (from various head viewing angles) into head gazing coordinates.

Signals to Memory Structures – Chapter 8 –: The visual-retinal map outputs head gazing coordinates corresponding to foveal retinal coordinates. Such gazing coordinates are used to train long-term memories concerning object localization and identification.

Signals to Cross-modal Processing – Chapter 7 –: Proprioceptive data from all body joints is correlated with other perceptual modalities for self-recognition of visual images and sounds.

Feedback Signals to the Cerebellum and Motor Area – Chapter 9 –: Head forward / inverse kinematic maps are required to build arm forward/inverse kinematic maps, and hence this information has to be transmitted between the two modules for learning the latter map.

Inputs from the Frontal Lobe – Chapter 10 –

Signals to the Visual Association Area: The hand gestures recognition algorithm sends a binary contour image – built from the arm periodic trajectory – to the active segmentation algorithm, which fills such image, and applies such image to mask the image regions of a scene covered by such trajectory.

Signals to the "what"-Visual Pathway: Hand gestures are recognized by sending a contour image of the arm periodic trajectory to the geometric-hashing based object recognition algorithm. This recognition algorithm then fits the contour with a set of lines, and classifies such set, by comparing it with lines modelling geometric shapes previously learned.

Signals to the Cerebellum and Motor Area – Chapter 9 –: Description of periodic (feedback transition to self-state) and non-periodic transitions (transitions to other states) in a markov chain is what is need to integrate different control structures into a unified scheme. That remains to be done on-line. After learning of simple tasks such as poking and waving, such learning is then applied, off-line, to perform these simple tasks.

2.3.3 Discussion

Throughout this thesis, computational structures implemented for the humanoid robot Cog were mapped into the main functional areas of the human cortex and cerebellum. We did not intend to neglect the extremely important role of both other brain areas, such as basal ganglia or the limbic system, or the peripheral nervous system (such as the spinal cord). Indeed, we often refer to functions corresponding to such areas (for instance, the hippocampus memory functions, the central pattern generators at the spinal cord, or the thalamus important role for processing sensorimotor information, just to name a few cases). Our option was driven by the will to make the exposition clear and easy to understand. Indeed, mapping the functions of the brain is often inaccurate, since the brain is highly interconnected, being extremely difficult to map certain computational structures just to one specific brain area, as discussed in this manuscript. Therefore, such mapping should be viewed as a guiding element, but not in anyway fixed with strict boundaries.

Even though few signals are sent to visual structures from computational algorithms labelled to the frontal lobe, the feedback loop is strongly closed by robot actuation on objects which generates visual (and auditory) percepts. Indeed, we have constrained ourselves to only introduce in this section the most important communication channels internal to the robot from which transmission of information between the algorithms is possible. But of paramount importance are also the communication links established between the computational structures through the environment where the robot is situated, for which no explicit representation is available.

Although not explicitly indicated, some links are established from a series of links. For instance, the output of visual object recognition is sent for building descriptions

of scenes. A sound identifier is not sent. But if a cross-modal binding occurs, then the sound will be associated to the object which is linked to the scene. For instance, if the sound *"table"* is bound to the image of a table, it will make implicitly part of the scene description, since the sound is linked to a visual description of the object table in such a scene.

The algorithms are indeed highly interconnected. The cognitive organization is therefore somewhat fuzzy, and highly distributed.

Chapter 3

Low-level Visual Processing

The distinguished Gestalt psychologist Kurt Koffka... asked the single deceptively simple question "Why do things look as they do?"
...Because the world is the way it is.
...Because we have learned to see them that way.
...Because of the way in which each small piece of the visual field appears.
A behaviorist might have asked "What does vision enable us to do".

(Palmer, 1999)

Vision is perhaps the most important sense in human beings, and it is also a perceptual modality of crucial importance for building artificial creatures. In the human brain, visual processing starts early in the eye, and is then roughly organized according to functional processing areas in the occipital lobe. This chapter presents algorithms – approximately independent from each other – to model processing capabilities at low-level visual layers which receive inputs directly from the eye. The following chapters will then show how this low-level information is integrated by high-level processing modules.

Cognitive Issues Transformation of visual representations starts at the very basic input sensor – in the eye – by having light separated into gray-scale tones by rod receptors, and in the color visible spectrum by cones (Goldstein, 1996). The

brain's primary visual cortex receives then the visual input from the eye. These low-level visual brain structures change the visual input representation yielding more organized percepts (Goldstein, 1996). Such changes can be modelled through mathematical tools such as the Daubechies-Wavelet (Strang and Nguyen, 1996) or Fourier transforms (Oppenheim and Schafer, 1989) for spectral analysis. Multi-scale representations fine-tune such organization (Palmer, 1999).

Edge detection and spectral analysis are known to occur on the hypercolumns of $V1$ occipital brain's area. Cells at $V1$ have the smallest receptive fields (Palmer, 1999). Vision and action are also highly interconnected at low visual processing levels in the human brain (Iacoboni et al., 1999).

Developmental Issues During Mahler's autistic developmental phase (Mahler, 1979) (from birth to 4 weeks old), the newborn is most of the time in a sleeping state, awakening to eat or satisfy other necessities (Mahler, 1979; Muir and Slater, 2000). His motor skills consists mainly of primitive reflexes until the end of this phase (Michel and Moore, 1995). Towards the Symbiotic phase (until 4-5 months), the infant's attention is often drawn to objects under oscillatory motions, or to abrupt changes of motion, such as throwing an object. These two phases, which precede the Separation and Individuation phase, build the cognitive foundations for the healthy and marvellous cognitive growth which occurs thereafter.

3.1 Spectral Features

There is cognitive evidence for the existence of spatial frequency channels in the visual system (Goldstein, 1996). Spectral coefficients of an image are often used in computer vision for texture segregation or for coding contextual information of a scene (Goldstein, 1996; Palmer, 1999). But the selection of both spatial and frequency resolutions poses interesting problems, as stated by Heisenberg's Uncertainty Principle.

3.1.1 Heisenberg's Uncertainty Principle

A famous mathematical formula, the Heisenberg Principle of Uncertainty states that no filter can, with arbitrary accuracy, simultaneously locate a feature in terms of both its period and its time of appearance. We can only know time intervals in which certain bands of frequencies exist. We cannot know the precise frequencies existing at given time instants.

In order to gain more temporal precision, frequency information must be sacrificed. According to this principle, the narrower the processing window, the better the time resolution, but the poorer the frequency resolution. Wide processing windows have good frequency resolution but poor time resolution. On the other hand, narrow windows become blind to low frequencies, i.e., they imply a drop in frequency resolution.

3.1.2 Spectral Analysis: Fourier, Wavelets and Gabor Transforms

Several transforms are available in the literature for processing the spatial distribution of frequencies in a signal.

Short-Time Fourier Transform

The Fourier transform is one of the techniques often applied to analyze a signal into its frequency components (Oppenheim and Schafer, 1989). A variant of it is the Short-Time Fourier transform (STFT). This transform divides the signal into segments. The signal truncated by each of these segments is convolved by a window (Oppenheim and Schafer, 1989) – such as a rectangular or Hamming window – and the Fourier Transform is then applied to it. When applied over different window sizes, this transform obtains spatial information concerning the distribution of the spectral components – section 3.4 will further elaborate on this method to process 1-dimensional signals.

Wavelets

Another method widely used to extract the spatial distribution of spectral components is the Wavelet transform (Strang and Nguyen, 1996), which has been widely used for texture segmentation, image/video encoding and image reduction of noise, among other applications. Indeed, a distinct feature of wavelet encoding is that reconstruction of the original image is possible without loosing information (Strang and Nguyen, 1996).

The Continuous Wavelet Transform (CWT) consists in convolving a signal with a mathematical function (such as the Haar or Daubechies functions) at several scales (Strang and Nguyen, 1996). However, CWT is not well-suited for discrete computation, since in theory there are infinitely many scales and infinitely many translations to compute in order to achieve perfect reconstruction of the signal. In practice both have to be quantized, which leads to redundancy and errors in reconstruction.

The discrete implementation of the wavelet transform – Discrete Wavelet Transform (DWT) – overcomes the quantization problems and allows fast (linear time complexity) computation of the transform for digitized signals. Similarly to CWT, it also provides perfect signal reconstruction.

DWT produces sparse results, in the sense that its application for processing typical signals outputs many small or zero detail coefficients, which is a desired property for image compression. This tool for multi-resolution analysis is closely related to: Filter banks in signal processing (Strang and Nguyen, 1996); Pyramid algorithms (Burt and E.H, 1983) in image processing; Quadrature mirror filters in speech processing (Knutsson, 1982; Knutsson and Andersson, 2003); and Fractal theory (Massopust, 1994).

Gabor Filters

Gabor filters are yet another technique for spectral analysis. These filters provide the best possible tradeoff for Heisenberg's uncertainty principle, since they exactly meet the Heisenberg Uncertainty Limit. They offer the best compromise between spatial and frequency resolution. Their use for image analysis is also biologically motivated, since they model the response of the receptive fields of the orientation-selective cells in the human visual cortex (Daugman, 1985).

3.1.3 Comparative Analysis and Implementation

The STFT framework has an important pitfall, also known as the *windowing dilemma*. It is often difficult to select the width of the processing windows, the number of different widths to use, how or whether they should overlap, etc. In addition, and unlike the classical Fourier transform, the original signal cannot be numerically reconstructed from its STFT.

Wavelets have some drawbacks when compared to other alternatives, such as STFTs at multiple scales or Gabor filters. Compared to Gabor filters (Pichler et al., 1996; Grigorescu et al., 2002), wavelets' output at higher frequencies are generally not so smooth and orientation selectivity is poor (the maximum number of orientations is bounded by a small constant). This difference results from the encoding of redundant information by Gabor filters.

The Heisenberg's principle imposes a bound on how well a wavelet can detect a feature. Indeed, Wavelets are very narrow in their capabilities. They act as bandpass filters only, a given wavelet responds only to periodic variation in the vicinity of its center frequency. Gabor filters generalize wavelet filters by including low-pass and high-pass filtering operations.

However, wavelets are much faster to compute than Gabor filters and provide a more compact representation for the same number of orientations, which motivated their selection over Gabor filters in our work. Wavelet components are thus obtained by transforming input monochrome images using a Daubechies-4 (Daubechies, 1990) wavelet tree (other wavelet transforms are available in the research literature, such as the Haar wavelet transform (Strang and Nguyen, 1996)), along depth scales. Each time the transform is applied, four smaller (polarized) images are obtained, corresponding to the lower frequencies and higher vertical, horizontal and diagonal frequencies. Processing is applied iteratively, as shown by figure 3-1, using Mallat's algorithm (Mallat, 1989)

Because of signal polarity, wavelet components correspond to a compact representation of six orientations at each level. This somewhat models the visual cortex organization into orientation columns as observed in monkeys by (Hubel and Wiesel, 1977, 1979), with each column containing cells that respond best to a particular orientation.

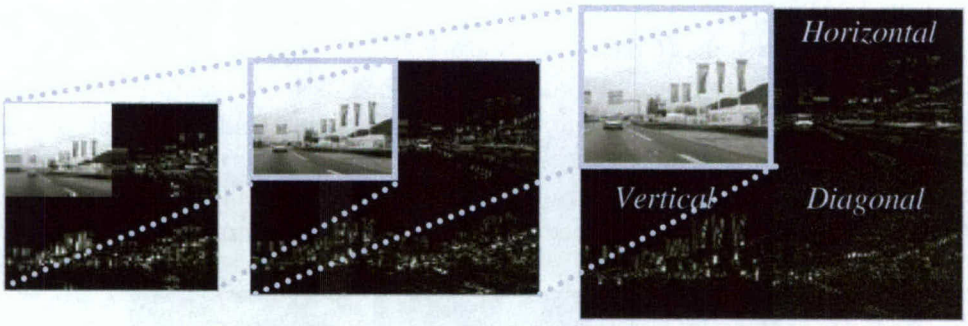


Figure 3-1: Wavelet decomposition of a scene. Wavelet transform is applied over three levels along the low-frequency branch of a tree ($T = 3$).

3.2 Chrominance/Luminance Attributes

Color is one of the most obvious and pervasive qualities in our surrounding world, having important functions for perceiving accurately forms, identifying objects and carrying out tasks important for our survival (Goldstein, 1996).

Two of the main functions of color vision are (Goldstein, 1996):

Creating perceptual segregation This is the process of estimating the boundaries of objects, which is a crucial ability for survival in many species (for instance, consider a monkey foraging for bananas on a forest: a color-blind monkey would have more difficulty in finding the fruit).

Signaling Color also has a signaling function. Signals can result from the detection of bright colors (e.g., stop at a red traffic light) or from the health index given by a person's skin color or a peacock's plumage.

The signaling function of color will later be exploited in chapter 4 as a pre-attentional feature for selection of salient stimuli. In addition, skin-color detection, which is described in detail by (Scassellati, 2001), will be used extensively for face and arm/hand/finger detection.

Perceptual segregation of color will be later applied for object recognition from local features (see chapter 5). One possible way of identifying an object in an image is through the reflectance properties of its surface - its *chrominance*, and its *luminance*. An object might be specified by a single color, or several groups of colors. Color (in RGB format, 8 bits per pixel) will be first decomposed into its normalized chrominance ($c_1 = (r-g+255)/2$, $c_2 = (r-b+255)/2$) and luminance $l = (r+b+g)/3$ components.

This color decomposition emulates the two distinct classes of photoreceptor cells in the human retina: rods and cones. Rods are more numerous, very sensitive to light and widely spread in the retina (except at its very center and at the blind spot where the the optic nerve gathers). These receptors are mainly used for vision at very low light levels. Cones, concentrated on the retinal center, are less abundant and much

less sensitive to light, are responsible for most of the visual experiences of color under normal lighting conditions (Goldstein, 1996).

Color histograms (Swain and Ballard, 1991) are used to create chrominance or luminance regions. Each chrominance/luminance region contains a set of similar colors within a given tolerance (denominated as color buckets). Geometric relationships among these regions are extracted by detecting spatially connected (using a 8-connectivity criterion) chrominance/luminance regions (figure 3-2).

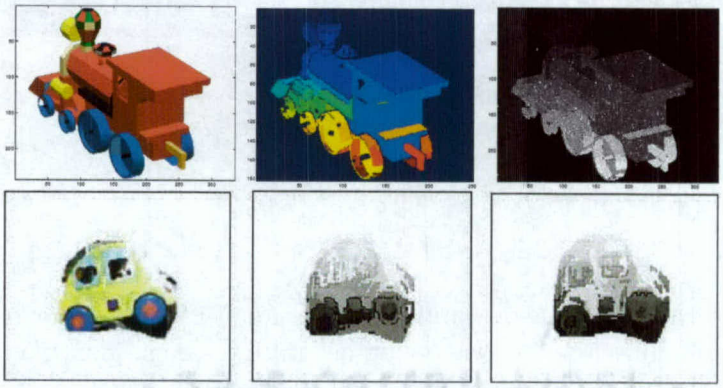


Figure 3-2: (left) Original template; (center) Topological Chrominance regions; (right) Luminance regions.

3.3 Shape - Geometric Features

Another potential description for an object is through its geometric shape. Classes of objects may be specified along a single eigendirection of object features (e.g. the class of squares, triangles, etc, independently of their color or luminance). Possible geometric descriptors are spectral components determined for specific scales and orientations, as described in section 3.1. But a different method for a more accurate estimation of geometric features from local features is hereby described. This method consists of estimating oriented lines as geometric features (pairs of these features will later be used in chapter 5 as the input for an object recognition algorithm based on local features), as follows:

1. An image of contours is detected using the Canny detector (Canny, 1986)

2. A Hough transform (Hough, 1962; Lim, 1990) is applied to fit lines to the contour image produced in (1) (see figure 3-3)

3. A Sobel mask (Jain, 1989) is applied at each contour point (contour points correspond to image pixels lying on the contour) to extract the phase of the image gradient at that point

4. All such phase measurements lying on a line are averaged to compute the phase of the line.

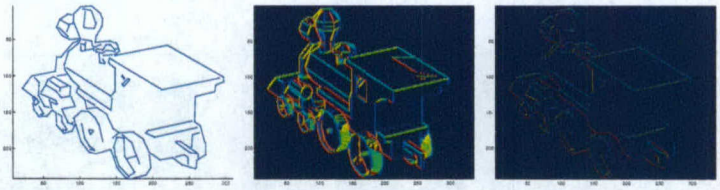


Figure 3-3: Shape features. (left) Lines fitted to the contour image; (center) Phase information from the Sobel mask; (right) All the phase values along a line are averaged to give the phase of the line.

3.4 Filtering Periodic Percepts

A human instructor may use different protocols to transmit information to a robot, as shown by Cog's algorithmic control structure in figure 3-4. The methodology developed for human-robot interactions is motivated by an infant's simple learning mechanisms in Mahler's autistic and symbiotic developmental phases (Mahler, 1979). Indeed, baby toys are often used in a repetitive manner – consider rattles, car/hammer toys, etc. This repetition can potentially aid a robot in perceiving these objects robustly. Playing with toys might also involve discontinuous motions. For instance, grabbing a rattle results in a sharp velocity discontinuity upon contact.

This motivated the design of algorithms which implement the detection of events with such characteristics. Moving image regions that change velocity either periodically, or abruptly under contact produce visual event candidates.

The detection of discontinuous motions is the topic of section 3.5, while this section focuses on the detection of periodic events, as follows: trackers are first initialized to moving image points, and tracked thereafter over a time interval; their trajectory is then evaluated using a Short Time Fourier transform (STFT), and tested for a strong periodicity.

3.4.1 Motion Detection

A motion mask is first derived through image differences over the spatially smoothed versions of two consecutive images:

$$\begin{aligned} I_0^{diff} &= I_0(x,y) * G(x,y) \\ I_k^{diff} &= I_k(x,y) * G(x,y) - I_{k-1}^{diff}(x,y), \quad k \geq 1 \end{aligned} \tag{3.1}$$

where $I_k(x,y)$ represents the image intensity at a point with coordinates (x,y) at time instant k, and $G(x,y)$ is a two dimensional gaussian (implementation was optimized

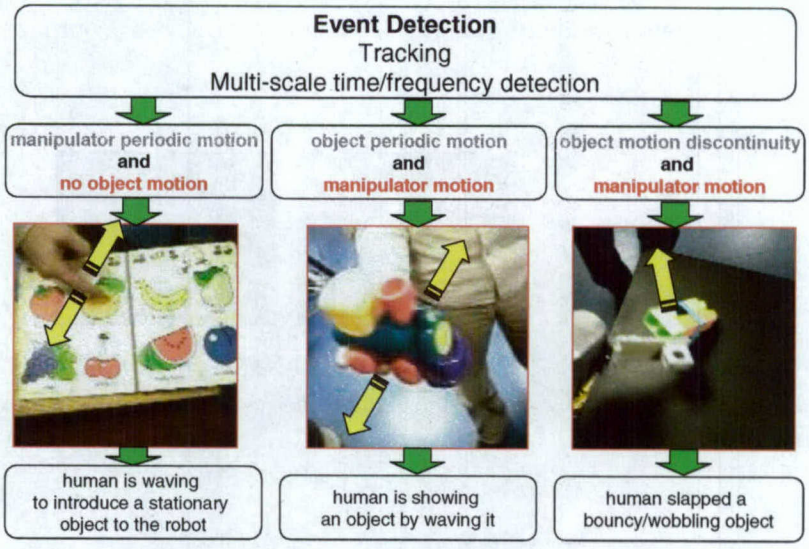


Figure 3-4: Images of objects are segmented differently according to scene context. The selection of the appropriate method is done automatically. After detecting an event and determining the trajectory of periodic points, the algorithm determines whether objects or actuators are present, and switches to the appropriate segmentation method.

for optimal performance). Sparse data often results from this process. Hence, non-convex polygons are placed around any motion found, as follows. The image is first partitioned into n overlapping windows. Then, to each of these elementary regions, a standard convex-hull approximation algorithm is applied if enough points are available - the convex optimization algorithm applies the standard Quicksort method for points sorting. The collection of all such polygons result in non-convex polygons that approximate the sparse data. The image moving region is given by such collection.

3.4.2 Tracking

A grid of points homogeneously sampled from the image are initialized in the moving region, and thereafter tracked over a time interval of approximately 2 seconds (65 frames). The choice of this time interval is related to the idea of natural kinds (Hendriks-Jansen, 1996), where perception is based on the range of frequencies which a human can easily produce and perceive. This was corroborated by an extensive number of experiments for different window sizes.

At each frame, each tracker's velocity is computed together with the tracker's location in the next frame. The motion trajectory for each tracker over this time interval was compared using four different methods. Two were based on the computation of the image optical flow field - the apparent motion of image brightness - and consisted of 1) the Horn and Schunk algorithm (Horn, 1986); and 2) Proesmans's algorithm - essentially a multiscale, anisotropic diffusion variant of Horn and Schunk's algorithm. The other two algorithms rely on discrete point tracking: 1) block matching; and 2)

the Lucas-Kanade pyramidal algorithm.

The evaluation methodology consisted of running several experiments. In each experiment, a 1-minute sequence of images of a human waving objects is saved into a video. The four algorithms are then applied to the sequence of images and the estimated trajectory of the object's centroid is evaluated empirically. The methods based on optical flow field are good for velocity estimation, but not so for position estimation due to dependence on a smoothing factor. The Lucas-Kanade algorithm achieved the best results, and was therefore selected for further use.

3.4.3 Multi-scale Periodic Detection

The Fourier transform does not explicitly encode time-varying properties of a signal. If a signal is locally periodic, i.e., its periodicity changes slowly with time, then this time-varying property may not be captured by the Fourier transform. But the encoding of such a property can be accomplished by applying the Short Time Fourier Transform over successive windows of the signal. Therefore, a STFT is applied to each tracker's motion sequence,

$$I(t, f_t) = \sum_{t'=0}^{N-1} i(t')h(t' - t)e^{-j2\pi f_t t'} \tag{3.2}$$

where h is usually a Hamming window, and N the number of frames (Oppenheim and Schafer, 1989). In this work a rectangular window was used. Although it spreads the width of the peaks of energy more than the Hamming window, it does not degrade overall performance (as corroborated from empirical evaluation over a large number of image sequences), and decreases computational times.

Periodicity is estimated from a periodogram determined for all signals from the energy of the STFTs over the frequency spectrum. These periodograms are filtered by a collection of narrow bandwidth band-pass filters. Periodicity is found if, compared to the maximum filter output, all remaining outputs are negligible. The periodic detection is applied at multiple time scales. If a strong periodicity is found, the points implicated are used as seeds for segmentation (this process will be described in chapter 4).

Previous research work in detecting rigid motions by application of a Fourier transform based approach was presented by (Polana and Nelson, 1997). (Seitz and Dyer, 1997) and (Cutler and Davis, 2000) propose alternative approaches. These works, as other approaches to solve this detection problem, are mostly focused on the classification of "biological motions" (Johansson, 1976) – the periodic motions performed by humans and animals in their gaits (such as running or walking). This thesis work is mostly interested in detecting periodic motions of a human arm while waving, shaking or tapping on objects. Examples of such situations are the use of tools in tasks (such as having a human sawing or hammering a piece of wood), or playing with toys (for instance, shaking a rattle or a musical instrument to produce sound).

Table 3.1: Categories of discrete, spatial events

Type of Interaction	**Contact** *eg. poking/grabbing an object, assembling it to another object.*	**Release** *eg. throwing or dropping an object, or disassembling it*
Explicit	•overlap of two entities •large a priori velocities	•two moving entities loose contact •large a priori velocities
Implicit	•abrupt grow of the actuator's motion area •large actuator velocity •abrupt velocity rise for previously stationary object 	•large initial velocity of ensemble •large a posteriori velocities for at least one of the entities •motion flow of assembled region separates into two disjoint regions 

3.5 Filtering Discontinuous Motions

Time information is lost by transforming a signal into the frequency domain, and neglecting the transform's phase information. It is not possible to detect time events from the energy of the Fourier transform of a signal. However, signals generated by most interesting moving objects and actors contain numerous transitory characteristics, which are often the most important part of the signal, and Fourier analysis is not suited to detect them. Actions that may generate these signals include, among others, throwing, grabbing, assembling or disassembling objects.

Therefore, it is imperative to detect events localized in time. With this in mind, the discontinuous motion induced on an object whenever a robot (or a human instructor) acts on it is used to facilitate the object's perception and to transmit relevant information to a robot, such as how to act on objects. We extend the work by (Fitzpatrick, 2003a) – who developed a framework for detecting a particular kind of events (implicit contacts) for the robotic task of poking objects – to the detection of more general events using a different approach. Hence, in order to detect discontinuous events, an algorithm was developed to identify interactions among multiple objects in the image:

1. A motion mask is first derived by subtracting gaussian filtered versions of successive images from each other and fitting non-convex polygons to any motion areas found, as presented in the previous section.

2. A region filling algorithm is applied to separate the mask into regions of disjoint polygons (using an 8-connectivity criterion).

3. Each of these regions is used to mask a contour image computed by a Canny edge detector.

4. The contour points are then tracked using the Lucas-Kanade pyramidal algorithm.

5. An affine model is built for each moving region from the 2-dimensional position p_i and velocity v_i vectors of the tracked points, as follows. Considering $V = (v_1, \ldots, v_n)$, $P = (p_1, \ldots, p_n)$ and $P_c = ([p_1^T \ \ 1], \ldots, [p_n^T \ \ 1])^T$, the affine model is given by

$$V = \beta P_c \tag{3.3}$$

where the $n \times (n+1)$ matrix β obtained using the standard method of the pseudo-inverse matrix, equivalent to a minimum least squares errors optimization. The model uncertainty is given by (3.4).

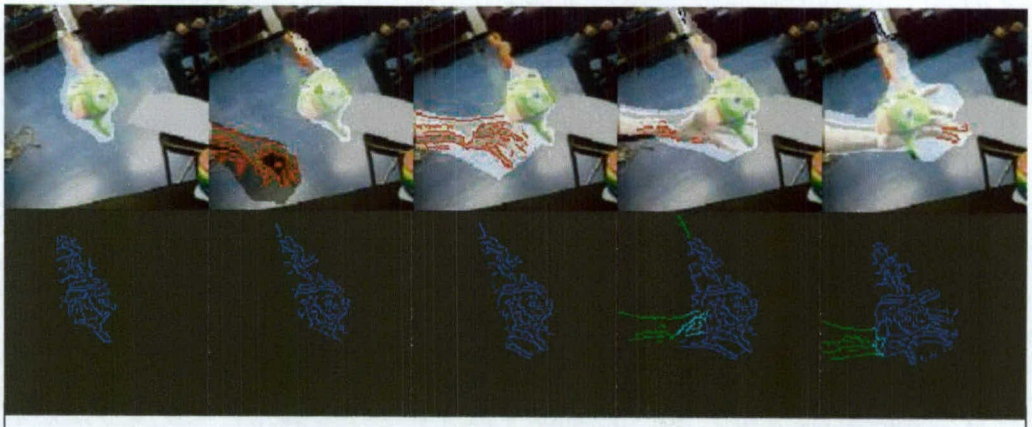
$$C = E[(v_i - \beta[p_i^T \ \ 1]^T)(v_i - \beta[p_i^T \ \ 1]^T))] \tag{3.4}$$

Outliers are removed using the covariance estimate for such model, by imposing an upper bound in the Mahalanobis distance $d = (v_i - \beta[p_i^T \ \ 1]^T)^T C(v_i - \beta[p_i^T \ \ 1]^T)$.

6. Objects' contour masks are updated through a Kalman filter (Anderson and Moore, 1979; Kailath, 1981) based approach.The model dynamics consists of a constant acceleration model with an associated uncertainty, the state space being given by the contour mask's centroid position and velocity. The filter model innovations are given by object image masks, which are weighted according to their centroid velocity (therefore, the corresponding uncertainty decreases with velocity). For slowly moving objects, the model prediction will have a large weight, and hence this is equivalent to average the innovation mask with previous masks. For fast moving objects, the innovation dominates the contour mask update.

Multiple image regions are tracked – a region is tracked if it is moving or it maps into an object in memory. An object is inserted into memory for a short period of time ($\sim$ 4 seconds) after the detection of actions exerted on it – a spatial event. Therefore, spatial events are defined according to the type of objects' motion discontinuity (Arsenio, 2003a), as presented in Table 3.1:

Implicit Contact This class of events is originated by such actions as grabbing a stationary object, or assembling it to a moving object. It corresponds also to the initial event of poking/slapping a stationary object. The velocity of the stationary object rises abruptly from zero under contact with the actor actuator (human arm/hand or robotic arm/grip) or moving object. This creates a sudden expansion of the optical flow (because both the actor and the object move instantaneously together).



Fish oscillating on a cable	*Hand/Fish motion do not match*	*Hand grabs fish*	*Hand/Fish move together*

Figure 3-5: Detection of a spatial event – grabbing an object – corresponding to an explicit contact. The algorithm detects two moving regions – the fish (already being tracked due to detection from a previous event), and the human arm – overlapping, with large a priori velocities. After contact, both stop moving, occurring therefore a sudden change in velocity for both. The algorithm's output is the type of event (an explicit contact) and image templates with the contours of both entities (the actuator and the object).

Implicit Release An object and the actor's actuator, or two objects assembled together, might be moving rigidly attached to each other. But without further information, the robot perceives them as a single object. The occurrence of a separation event upon the object (its release by the actuator) removes such ambiguity, either by throwing or dropping the object. This event may also result from a disassembling operation between two objects. For such cases, hierarchical information concerning object structure is available to an object recognition scheme.

Explicit Contact This event is triggered by two moving entities coming into contact, such as two objects crashing into each other or an actuator slapping or grabbing a moving object (see figure 3-5).

Explicit Release The detection of this event is similar to the previous one, differing on the trigger condition: two moving entities losing contact. The algorithmic conditions require that the entities' templates become disjoint over two consecutive frames, and the velocity of one of the entities change abruptly.

Figure 3-6: System running on the robot. Whenever a spatial event occurs (signaled by squares in the images), five images for five time instants in a neighborhood of the event are shown. Experiments are shown for the detection of a human a) dropping a ball b) bouncing a ball c) catching a flying object d) throwing a cube rattle. It also shows human performing a cycle of throws/catchs of e) a car and f) a bottle.

Results and Applications

This event detection strategy was used to detect events such as grabbing, dropping, poking, assembling, disassembling or throwing objects, and to segment objects from such events by application of an active segmentation algorithm, which will be described in chapter 4. Figure 3-6 shows several snapshots of running experiments for online detection by Cog. A human actor first captures Cog's attention to the desired object by creating a salient stimuli with it. This stimuli is then detected by an attentional system (which is described in the next chapter), and both cameras are verged to the object. The human actor then performs several actions on the object, each producing a discontinuous movement of the object. Different actions are separated by stationary states, in which no action occurs.

Chapter 4

Visual Association Area

Perception [is] the reception of form without matter.

Thus is it for a man to think whenever he will, but not so for him to perceive, because for that the presence of a sense-object is necessary... the sense-objects are among the particular and external things.

(*Aristotle; BC*, 350)

As posed by (*Aristotle; BC*, 350) for the case of human perception, for a robot to perceive a sense-object requires the presence of it. Ideally, we would like to learn models for these sense-objects even if we are not able to act on them. However, limited information is acquired in such situations, since most often objects are not irreducible, but instead are just the assembled result of simpler objects. An embodied approach permits disassembling an object, moving the links all together or one at a time to further identify kinematic constraints, or rotating an object for multi-view data.

Hence, this thesis follows an embodied approach for extracting objects properties (such as their visual appearance and form). We have seen in chapter 3 how the robot gets into physical contact with the sense-object, and in this chapter we will describe how object properties can be extracted from the world by processing such events.

Cognitive Issues The shape of an object, its global motion, or its surface boundaries can be perceived from information that is both temporally and spatially fragmentary (Shipley and Kellman, 1994).

A current issue in research on visual attention is whether the shape or boundaries of an object are segregated from their background pre-attentively, or else attention is first drawn to unstructured regions of the image. Evidence for the former is presented by (Driver et al., 1992), which motivated our development of preattentive segmentation schemes – attention is however useful to get a desired sense-object visible to the humanoid robot.

Luminance and color based figure-ground segregation seems to be correlated with activation in both visual low-level and visual association brain-areas, while motion based segregation appears mostly on extrastriate (visual association) areas (Spang et al., 2002). This suggests partly separate mechanisms for figure-ground segregation: color and luminance segregation may share some common neuronal mechanisms, while segregation not based on these low-level features takes place mainly on higher-level processing areas.

Developmental Issues Our approach detects physical interactions between an embodied creature and a sense-object – following (*Aristotle; BC*, 350) argument, and thereafter builds a visual appearance description of the object (as will be described in detail by sections 4.2 and 4.3) – together with its rough shape (section 4.4) – from basic visual percepts generated by the creature actions.

This developmental path is motivated by biological evidence that basic spatial vision and motion perception develops earlier than form perception (Kiorpes and Movshon, 2003).

We start by describing the integration of low level-features such as color and motion by a logpolar attentional system (section 4.1), which is used to select salient stimulus on the retina to which the robot's attention and computational resources are drawn.

4.1 A Logpolar Attentional System

Cones are densely packed over the fovea region at the center of the retina, where both spatial and color vision are most acute (except at the blind spot). The non-uniform distribution of cones on the retina results in more resources being allocated to process objects on the foveal view and less resources to process stimuli on the periphery. This fact led to the implementation of a space-variant attentional system (Scassellati, 2001; Metta, 2001) .

In order to select and integrate the information perceived from different sensors, a modified version of the logpolar attentional system proposed by (Metta, 2001) was developed to select relevant information, and to combine them in a saliency map (Arsenio, 2003d,c). Finally, this map is segmented to extract the region of stimuli saliency - the attentional focus (Wolfe, 1994; Wolfe et al., 2000).

There is cognitive evidence that the human visual system operates at different levels of resolution. Therefore, contextual priming is applied to modulate the attentional system at fine and coarse levels of resolution.

Logpolar vision - The human visual system has a retina with a non-uniform distribution of photo-receptors. This same idea is exploited by the log-polar transform, which maps a Cartesian geometry to a non-uniform geometry. We used space-variant images so that the central region of the image contains more information than the periphery, speeding up processing. The following basic feature detectors were used:

Color Processing - it includes (Metta, 2001) i) a general-purpose color segmentation algorithm based on histograms, and ii) a blob detector, based on region growing. Areas of uniform color according to hue and saturation are labelled and further processed to eliminate spurious results.

Skin Detection - based on the algorithm described in (Scassellati, 2001).

Optical Flow - optical flow is the apparent motion of image brightness (Horn, 1986). The optical flow constraint (Horn, 1986) assumes that i) brightness $I(x, y, t)$ smoothly depends on coordinates (x, y) on most of the image; ii) brightness at every point of an object does not change in time, and iii) higher order terms are discarded.

Edge Detection - it includes i) Gaussian filtering ii) Canny edge detector, and iii) selection of the image regions with stronger edges.

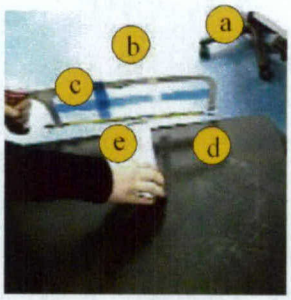
4.2 Active Figure/Ground Segregation

Object segmentation is a fundamental problem in computer vision, which will be dealt with by detecting and interpreting natural human/robot task behavior such as waving, shaking, poking, grabbing/dropping or throwing objects (Arsenio, 2004e). Object segmentation is truly a key ability worth investing effort in so that other capabilities, such as object/function recognition can be developed.

The number of visual segmentation techniques is vast. An active segmentation technique developed recently (Fitzpatrick, 2003b) relies on poking objects with a robot actuator. This strategy operates from first-person perspectives of the world: the robot watching its own motion. However, it is not suitable for segmenting objects based on external cues. This active strategy applies a passive segmentation technique – the maximum-flow/minimum-cut algorithm – developed by (Boykov and Kolmogorov, 2001).

Among previous object segmentation techniques a key one is the normalized cuts approach (Shi and Malik, 2000) based on spectral graph theory. This algorithm searches for partitions that maximize the ratio of affinities within a group to that across groups to achieve globally optimal segmentations (Shi and Malik, 2000). Although a good tool, it suffers from several problems which affect non-embodied techniques. Indeed, object segmentation on unstructured, non-static, noisy, low resolution and real-time images is a hard problem (see figure 4-1):

▷ objects may have similar color/texture as background



a. Other moving objects in the scene
b. Background saturation
c. background and part of the object have similar textures and color
c. Saw's parts with small, textureless surfaces
d. The human actor acting an object causes additional motion and shadows
e. Other moving object in contact with the target object, having similar color and texture

Figure 4-1: The results presented in this thesis were obtained with varying lighting conditions. The environment was not manipulated to improve naturally occurring elements, such as shadows or light saturation from a light source. Many of the experiments were taken while a human or the robot were performing an activity.

- ▷ multiple objects might be moving simultaneously in a scene
- ▷ necessary algorithm robustness to luminosity variations
- ▷ requirement of real-time, fast segmentations forces low resolution images (128×128 images)

Segmentations must also be robust to variations in world structure (like different levels of environmental cluttering). In addition, mobility constraints (such as segmenting heavy objects) poses additional difficulties, since motion cannot be used to facilitate the problem.

Distinguishing an object from its surroundings – the figure/ground segregation problem – will be dealt with by exploiting shared world perspectives between a cooperative human and a robot. Indeed, we argue for a visually embodied strategy for object segmentation, which is not limited to active robotic heads. Instead, embodiment of an agent is exploited by probing the world with a human/robot arm. A human teacher can facilitate a robot's perception by waving, poking or shaking an object (or tapping on it) in front of the robot, so that the motion of the object is used to segment it (or the actuator's motion). This strategy proves not only useful to segment movable objects, but also to segment object descriptions from books (Arsenio, 2004d), as well as large, stationary objects (such as a table) from monocular images.

We therefore use a number of segmentation strategies that are applied based on an embodied context. We achieve robustness through the combination of multiple computationally inexpensive methods. A potential weakness with our approach is the relatively crude model we use of a person – it is based only on motion of skin color patches.

4.2.1 Perceptual Organization of Object Features

The Gestalt school of visual perception, led by Max Wertheimer nearly a century ago, was the first to introduce formally the problem of visual grouping. This is often called

the "image segmentation" problem in computer vision, consisting on finding the best partition of an image (or sequence of images) into sets of features that correspond to objects or parts of them.

According to Gestalt psychologists, the whole is different than the sum of its parts – the whole is more structured than just a group of separate particles. The technique we suggest segregates objects from the background without processing local features such as *textons* (Malik et al., 1999). The proposed grouping paradigm differs from Gestalt grouping rules for perceptual organization. These rules specify how parts are grouped for forming wholes, and some of them (but not all) are indeed exploited by our grouping method: *Similarity* and *Proximity* rules are embedded in the color segmentation algorithm; moving periodic points in an image sequence are also grouped together.

Once periodic or discontinuous motion can be spatially detected and isolated from a majority of points generic in appearance, rather than drawn from the hand or finger, the active behavior guides the segmentation process according to the following steps,

1. The set of moving, non-skin points tracked over a time window is sparse. Hence, an affine flow-model is applied to the periodic flow data to recruit other points within uncertainty bounds (as it was done in section 3.5)

2. Clusters of points moving coherently are then covered by a non-convex polygon – approximated by the union of a collection of overlapping, locally convex polygons (Arsenio, 2003a), as presented in chapter 3.

4.2.2 Experimental Results

Figure 4-2 shows qualitative results – segmentation samples for a few objects tracked over time (see chapter 8) under a variety of perspective deformations – as well as quantitative results – error analysis shown as well in figure 4-2. Most experiments occurred in the presence of other scene objects and/or people moving in the background (they are ignored as long as their motion is non-periodic, as shown by the segmentation experiment displayed in figure 4-3). Segmentation performance varied between 10% and 30% depending on object characteristics (such as textures) and background, with a total error average of 15%. Under such errors templates seem to still preserve satisfactorily object shape and texture. Most experiments were performed while executing tasks, as shown by figure 4-4 for several tasks' experiments (swiping the ground, hammering, painting and filling).

Three different experiments are reported in figure 4-5, for the segmentation of objects from the discontinuous motion of moving edge points in the image. Figure 4-6 presents a quantitative analysis of segmentation quality from discontinuous events, together with qualitative segmentation results, built from a random sample of segmentations obtained through several experiments from a variety of actions either by a human or the robot. Besides object templates, the human or robot actuator's template is also extracted. Errors in the segmentation process varied from 3% to 121%.

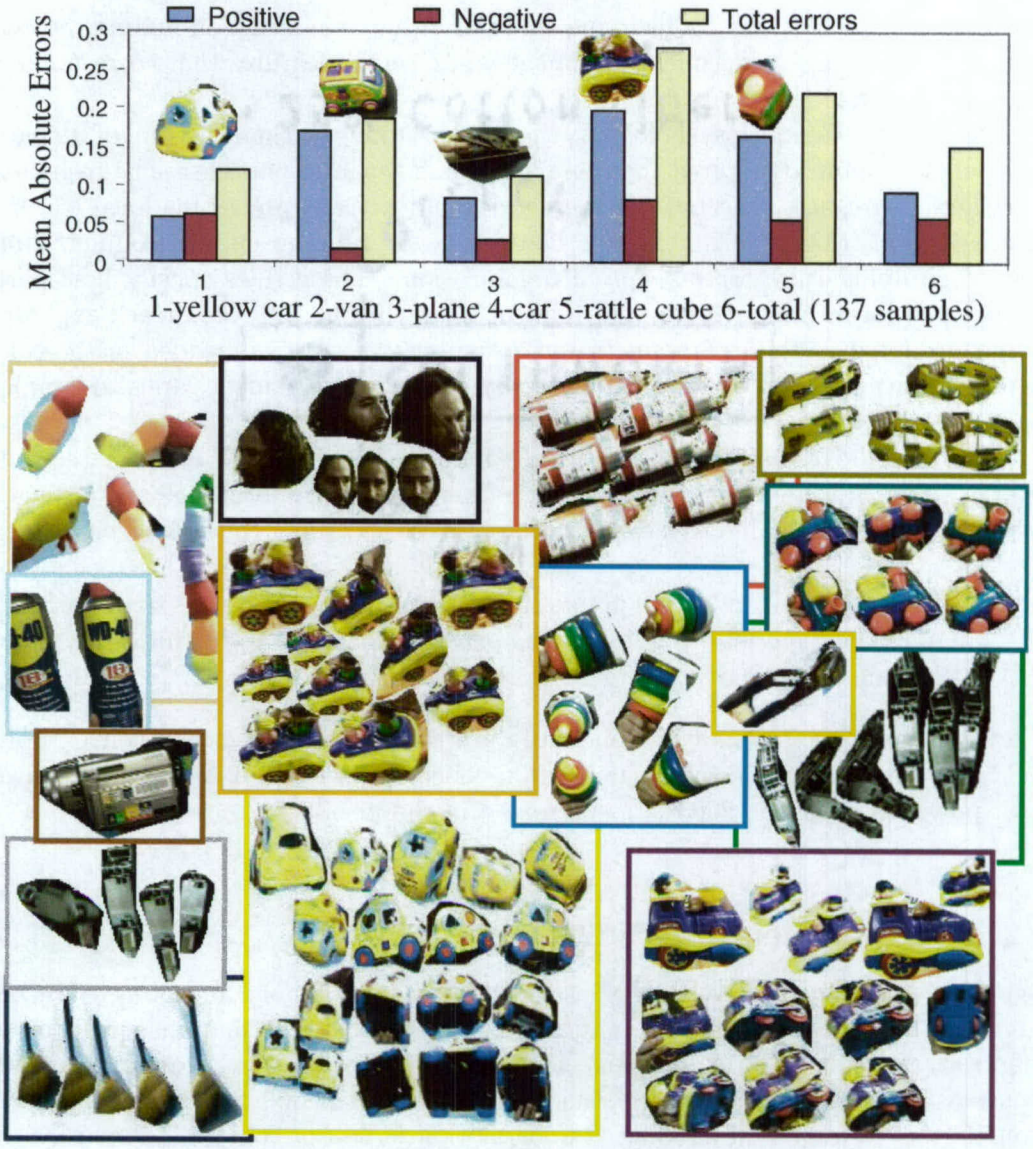
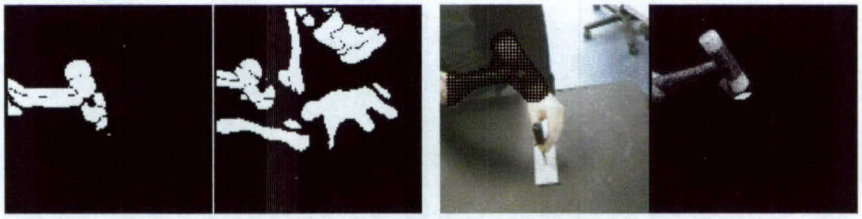


Figure 4-2: Statistical results for the segmentation of the objects shown. Errors are given by (template area - object's real visual appearance area)/(real area). Positive errors stand solely for templates with larger area than the real area, while negative errors stand for the inverse. Total errors stand for both errors (e.g., 0.15 corresponds to 15% error rate). Ground truth data for the object area was obtained by manual segmentation. Results shown in graph (6) correspond to data averaged from a larger set of objects in the database. Sample of objects segmented from oscillatory motions, under different views, are also shown. Several segmentations of the robots arm (and upper arm) are acquired, obtained from rhythmic motions of the robot's arm (and upper arm, respectively) in front of a mirror. Also shown are sample segmentations from a large corpora consisting of tens of thousands of segmentations computed by the real-time segmentation algorithm.



Motion of objects during experiment | *Grid of periodic moving points* | *Segmentation*

Figure 4-3: Segmentation results for objects from a grid of points moving periodically. Motion disturbances arise from other moving objects in the scene, shadows, lighting variations and human motion.

The latter error upper bound happens when background textures are segmented together with the object, creating an artificial membrane around the object boundary, with an area 121% bigger than the object real area (which is the case for the hammer segmentation in the aforementioned figure). The total mean error averages 48% with standard deviation in errors of 56%.

These results correspond to larger segmentation errors than the repetitive approach, since less information is available for segmentation – periodic events produce signals richer in information content than discontinuous events occurring abruptly over a very short period of time. As mentioned in chapter 3, shadows create sometimes fictitious events – such as a hammer beating its own shadow. But shadows introduce another difficulty during segmentation, since the shadow moves together with the object actuated and it is therefore associated with the object, introducing large segmentation errors (this explains the large errors on the experiment shown for segmenting a hammer).

This algorithm is much faster than the maximum-flow/minimum-cut algorithm (Boykov and Kolmogorov, 2001), and provides segmentations of similar quality to the active maximum-flow/minimum-cut approach presented by (Fitzpatrick, 2003b).

4.2.3 Deformable Contours

An interesting trade-off results from the procedure of both tracking and segmenting. Tracking through discrete methods over large regions (necessary for fast motions) requires larger correlation windows. For highly textured regions, points near an object boundary are correctly tracked. But whenever an object moves in a textureless background, background points near the object's boundary will be tracked with the object, therefore creating an artificial textureless membrane surrounding the object. Nonetheless, accurate segmentation is still possible by regularization using deformable contours (Arsenio, 2004f), initialized to the boundaries of the object templates, and allowed to contract to the real boundaries. For textureless backgrounds, the deformable contour is attracted to the closest edges that define the object's boundary.

Snakes (active contour models) are widely used deformable contours, that move

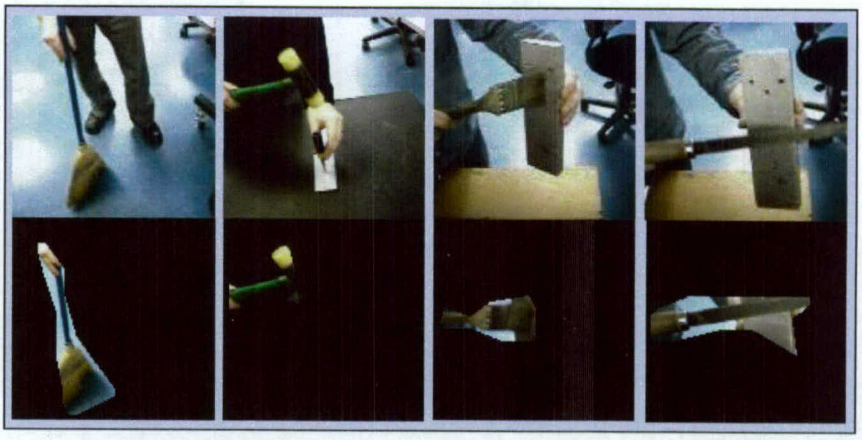


Figure 4-4: Segmentation results for objects involved in task execution. From left to right, human is: cleaning the ground with a swiping brush, hammering, painting and filling a metal piece. As shown in the top images, these cases offer challenging problems for segregation, such as light saturation, figure/background color similarity, shadows, etc. The algorithm extracts erroneously a portion of the background metal in one of the experiments (4^{th} image from left). However, only a small portion of the background object is extracted. Later in this chapter we will show hot to correct for these errors.

under the influence of image forces. We can define a deformable contour by constructing a suitable deformation internal energy $P_i(z)$, where z represents the contour points. The external forces on the contour result from an external potential $P_e(z)$. A snake is a deformable contour that minimizes the energy (Kass et al., 1987): $P(z) = P_e(z) + P_i(z)$, $z(s) = (x(s), y(s))$, with s being the distance along the contour. For a simple snake, the internal deformation energy is:

$$\int_0^T \frac{\alpha(s)}{2}|\dot{z}(s)|^2 + \frac{\beta(s)}{2}|\ddot{z}(s)|^2 ds \tag{4.1}$$

where the differentiation is in respect to s. This energy function models the deformation of a stretchy, flexible contour $z(s)$ and includes two physical parameter functions: $\alpha(s)$ controls the tension and $\beta(s)$ the rigidity of the contour. To attract the snake to edge points we specify the external potential $P_e(z)$,

$$P_e(z) = \int_0^T P_{image}(z(s))ds \tag{4.2}$$

where $P_{image}(x, y)$ is a scalar potential function defined over the image plane (Kass et al., 1987). The local minima of P_{image} are the snake attractors. Hence, the snake will have an affinity to intensity edges $|\nabla I(x, y)|$ computed by a Canny detector,

$$P_{image}(x, y) = -D(|\nabla I(x, y)|) \tag{4.3}$$

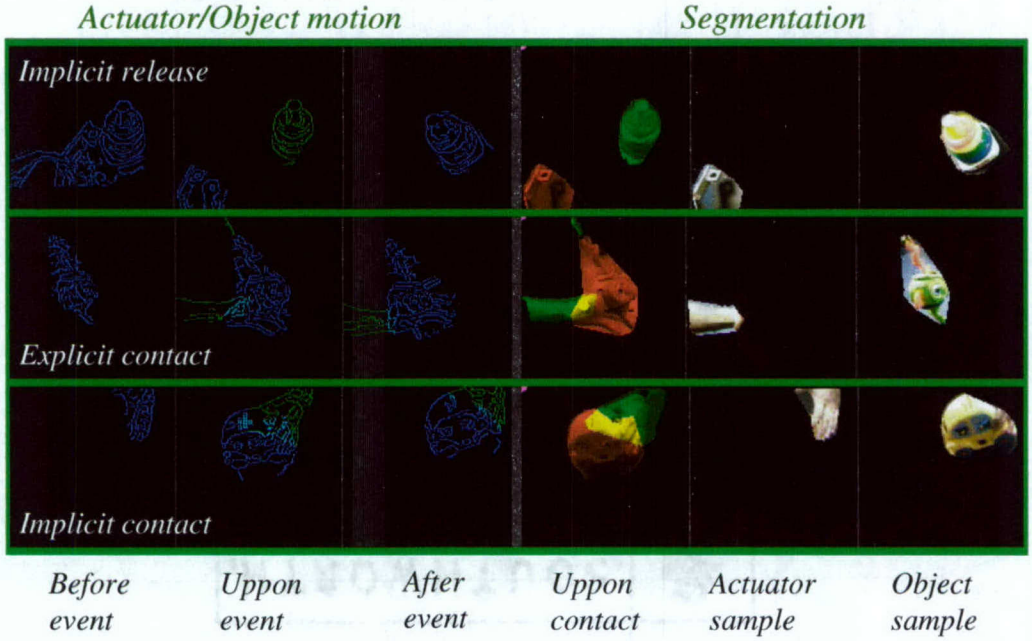


Figure 4-5: Detecting spatial discontinuous events. (top row) Implicit release: object and actuator move together initially, as a single entity, and separate suddenly – dropping event; (middle row) Human grabbing an oscillating fish – explicit contact; (bottom row) Human grabbing a stationary toy car – implicit contact. All these events enable segmentations for both object and actuator.

where D is the distance transform function, used to calculate the distance of image pixels to the deformable contour. This function labels every image pixel in the output image with a distance to the closest feature pixel. Feature pixels along the deformable contour correspond have a zero cost. This procedure converges in a small number of iterations, being the snake interpolated at each step to keep the distance between each point constant. The snake achieves high convergence rates towards the minimum, improving therefore segmentation results, as shown by figure 4-7.

Another potential function often used in the literature is given by equation 4.4,

$$P_{image}(x,y) = -G_\sigma \circ (|\nabla I(x,y)|) \tag{4.4}$$

which represents the convolution of the image gradient with a gaussian smoothing filter whose characteristic width, σ, controls the spatial extent of the attractive depression of P_{image}. However, not only did this convolution reveal itself to be very time expensive, but also low convergence rates of the deformable contour towards the minimum were attained.

For most of the experiments presented in this manuscript, the optimization of object boundaries using deformable contours was disabled in order to decrease segmentation processing times. Extensive segmentation evaluation both with and without

Errors								Total
Mean Error	0.24	0.03	0.25	1.21	0.14	0.29	0.17	0.48
Std	0.19	0.09	0.05	0.43	0.16	0.38	0.28	0.56
Mean abs error	0.29	0.07	0.25	1.21	0.20	0.37	0.30	0.52

Figure 4-6: (top) Statistical error analysis for the segmentation of the objects is shown in this table. Errors are given as in figure 4-2. (bottom) Sample segmentations from object's discontinuous motions actuated by humans and the robot. Not only the object's visual appearance is segmented from images, but also the robot's end-effector appearance.

the computation of optimal boundaries led us to believe that, although this optimization step improves segmentation quality, this improvement does not propagate later into relevant profits for object recognition efficiency.

4.3 Perceptual Grouping from Human Demonstration

We propose yet another human aided object segmentation algorithm to tackle the figure-ground problem which is especially well suited for segmentation of fixed or heavy objects in a scene, such as a table or a drawer, or objects drawn or printed in books (Arsenio, 2004d).

The problem to solve is formulated as follows:

> Given a monocular image which contains an object of interest, the problem consists in determining the clusters of features in the image which correspond to the correct representation of the apparent visual appearance of the object.

Objects might have multiple colors, as well as multiple textures. In addition, their shape might originate several groups of closed contours. In addition, this same

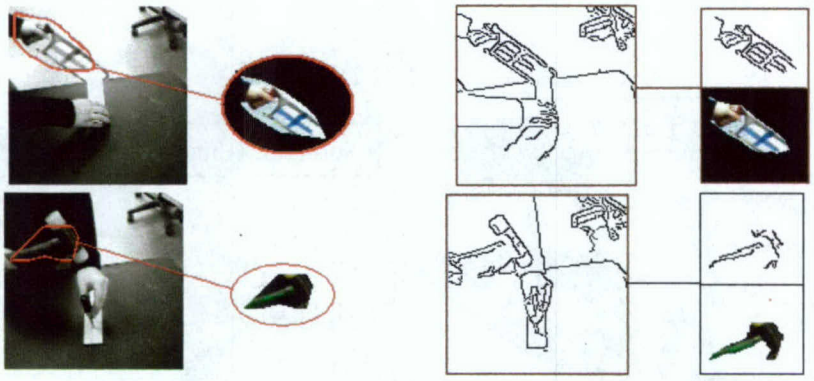


Figure 4-7: Deformable contours improve quality for object segmentations. The contour, which is initialized to the original template's boundaries, is tracked towards the nearest edges, removing the undesirable *membrane*.

richness in descriptive features usually applies for the object background as well (for non-trivial environments). Hence, to solve the aforementioned problem, one needs to:

- reject all clusters of features which belong to the object background
- group all clusters of features which make part of the object.

Our approach is to have a human actor to tell the robot, by repetitive gestures, which are the set of feature clusters which make part of an object, by pointing repetitively at them. This section deals with grouping color features into an unified object, while section 4.3.3 extends this human-robot interactive approach for grouping textures.

Embodiment of an agent is exploited through probing the world with a human arm. Near the object of interest, the human helper waves his arm, creating a salient stimuli in the robot's attentional system. The retinal location of the salient stimuli is thus near the object. Hence the robot moves its head to gaze at it, and becomes stationary thereafter. After saving a stationary image (no motion detected), a batch sequence of images is acquired to extract the human arm oscillating trajectory. Clusters of perceptual elements (for now colors) in the stationary image which intersect the human arm trajectory are grouped together to segment the visual appearance of the object from the background. The following algorithm implements the estimation process to solve this figure-ground separation problem (see figure 4-8):

1. A standard color segmentation (Comaniciu and Meer, 1997) algorithm is applied to a stationary image.

2. A human actor waves an arm on top of the target object.

3. The motion of skin-tone pixels is tracked over a time interval (by the Lucas-Kanade Pyramidal algorithm). The energy per frequency content – using Short-Time Fourier Transform (STFT) – is determined for each point's trajectory.

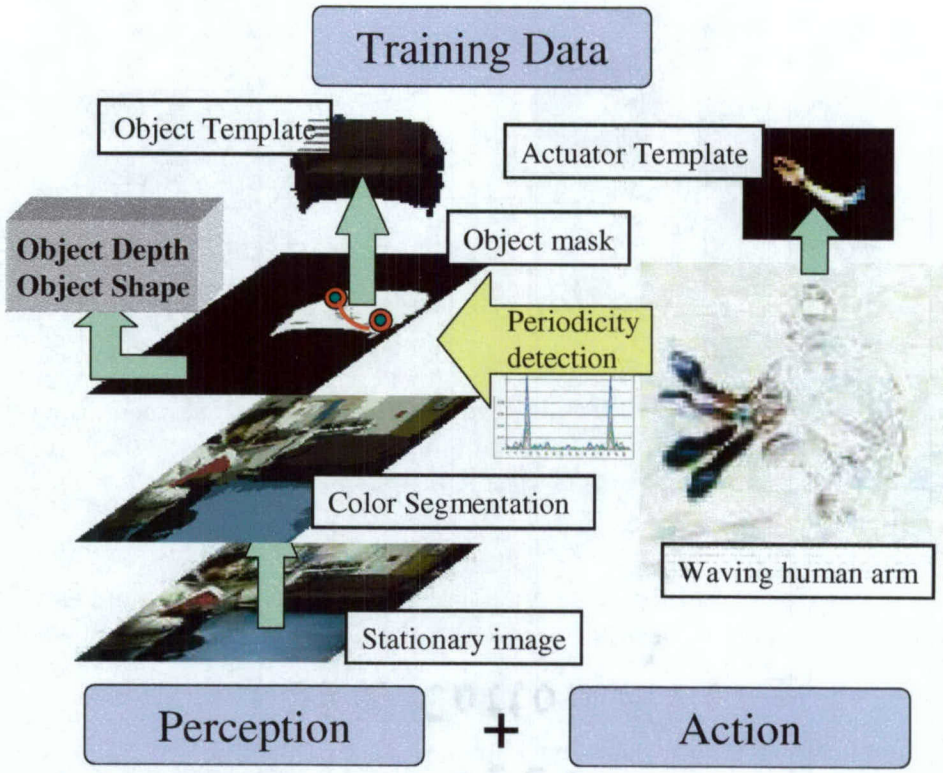


Figure 4-8: Algorithm for segmentation of heavy, stationary objects. A standard color segmentation algorithm computes a compact cover of color clusters for the image. A human actor *shows* the sofa to the robot, by waving on the objects' surface. The human actuator's periodic trajectory is used to extract the object's compact cover – the collection of color cluster sets which composes the object.

4. Periodic, skin-tone points are grouped together into the arm mask (Arsenio, 2004h).

5. The trajectory of the arm's endpoint describes an algebraic variety (Harris, 1994) – since it can be obtained from the zero locus of a collection of polynomials – over N^2 (N stands for natural integers). The target object's template is then given by the union of all bounded subsets (the color regions of the stationary image) which intersect this variety.

An affine flow-model is estimated (using a least squares minimization criterion, as described in section 3.5) from the optical flow data, and used to determine the trajectory of the arm/hand position over the temporal sequence. Periodic detection is then applied at multiple scales. Indeed, for an arm oscillating during a short period of time, the movement might not appear periodic at a coarser scale, but appear as such at a finer scale. If a strong periodicity is not found at a larger scale, the window size is halved and the procedure is repeated again. Periodicity is estimated from a

periodogram built for all signals from the energy of the STFTs over the frequency spectrum. These periodograms are processed by a collection of narrow bandwidth band-pass filters. Periodicity is found if, compared to the maximum filter output, all remaining outputs are negligible.

The algorithm consists of grouping together the colors that form an object. This grouping works by having periodic trajectory points being used as seed pixels. The algorithm fills the regions of the color segmented image whose pixel values are closer to the seed pixel values, using a 8-connectivity strategy. Therefore, points taken from waving are used to both select and group a set of segmented regions into the full object. This algorithm was used to segment both object descriptions from books and large, stationary objects (such as a table) from monocular images.

4.3.1 Perceptual Grouping Approaches

Previous research literature for figure/ground segregation is mainly divided in object segmentation from video (i.e., a sequence of images) – (Porikli, 2001) reports one such approach – and object segmentation from a single monocular image – which is perhaps best exemplified by the work at Berkeley University (Shi and Malik, 2000; Malik et al., 1999). Our approach does not fit exclusively in either of these: it segments an object from a single monocular image, using information provided by humans and extracted over a sequence of images.

Other research work for perceptual grouping includes optimal graph partitioning based approaches (Perona and Freeman, 1998; Shi and Malik, 2000) as well as variational approaches (Mumford and Shah, 1989). A perceptual grouping approach for image retrieval and classification is presented in (Iqbal and Aggarwal, 2001). A segmentation strategy which applies graph cuts to iteratively deform an active contour is presented in (Xu et al., 2003). All these strategies share a particular problem: they segment an image into groups of similar features, but they cannot say which collection of groups form a particular object.

The most related literature work to our approach applies data annotated off-line by humans (Martin et al., 2001). The authors report the construction of a database of natural images segmented manually by humans (each image segmented by three different people). A simple logistic regression (Ren and Malik, 2003) classifier is trained using as inputs classical *Gestalt* cues, including contour, texture, brightness and good continuation extracted from such annotated data. Classification consists in having this statistical algorithm estimating the most probable aggregation of regions for an image, from the knowledge inserted off-line by humans. (Martin et al., 2002) presents comparative results with other classifiers, such as classification trees, mixture of experts and support vector machines. The novelty of their approach is that they transform the segmentation problem into a recognition one.

Instead of using *off-line* knowledge, this thesis approach exploits *on-line* information introduced by a human helper, using such information as cues to agglomerate image regions into a coherent object. Hence, the robot is able to infer which collection of features groups form a particular object.

This is therefore an innovative approach.

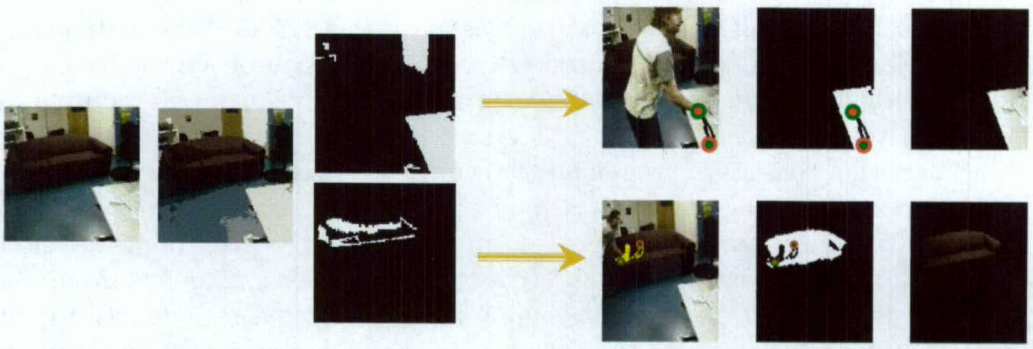


Figure 4-9: Segmentation of heavy, stationary objects. The arm trajectory links the objects to the corresponding color regions.

4.3.2 Experimental Results

Considering figure 4-9, both sofa and table segmentations are hard cases to solve. The clustering of regions by table-like color content produces two disjoint regions. One of them corresponds to the table, but it is not possible to infer which just from the color content. But a human teacher can *show* the table to the robot by waving on the table's surface. The arm trajectory then links the table to the correct region. For the sofa case, segmentation is hard because the sofa appearance consists of a collection of color regions. It is necessary additional information to group such regions without including the background. Once more, a human tutor *describes* the object, so that the arm trajectory groups several color regions into the same object - the sofa.

Figure 4-10 shows segmentations for a random sample of objects segmentations (furniture items), together with statistical results for such objects. Clusters grouped by a single trajectory might either form (e.g., table) or not form (e.g., black chair – a union of two disconnected regions) the smallest compact cover which contains the object (depending on intersecting or not all the clusters that form the object). After the detection of two or more temporally and spatially closed trajectories this problem vanishes – the black chair is grouped from two disconnect regions by merging temporally and spatially close segmentations. This last step for building templates from merging several segmentations is more robust to errors than extracting templates from a single event (such process is illustrated in figure 4-11).

Typical errors result from objects with similar color to their background, for which no perfect differentiation is possible, since the intersection of the object's compact cover of color regions with the object's complementary background is not empty. High color variability within an object create grouping difficulties (the compact cover contains too many sets – hard to group).

Visual Illusions: It is worth stressing that this grouping technique solves very easily the *figure and ground* illusion. This is usually experienced when gazing at the illustration of a white vase in a black background – the white vase (or the black faces)

Figure 4-10: Statistics for furniture items (random segmentation samples are also shown). Errors given by (template area - object's real area)/(real area). Positive/negative errors stand for templates with larger/smaller area than the real area. Total stands for both errors.

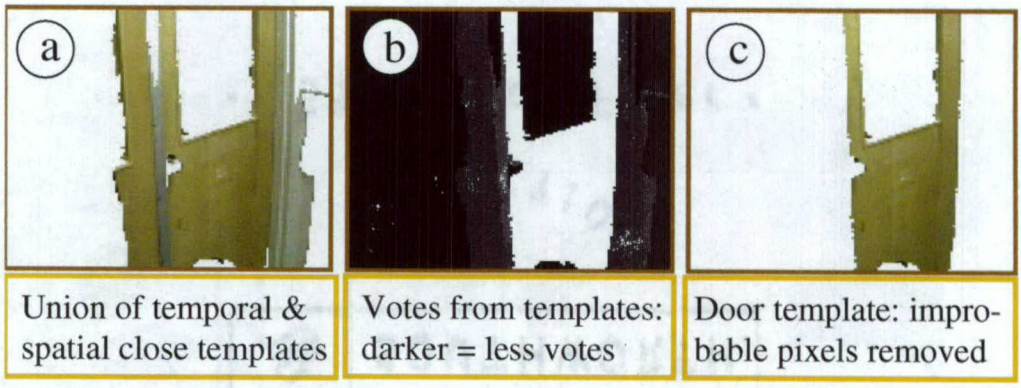


Figure 4-11: Merging temporally and spatially close templates. Example for segmenting a door a) Superposition of templates; b) Each template pixel places one vote in an accumulator map; c) final template results by removing pixels with the smaller number of votes.

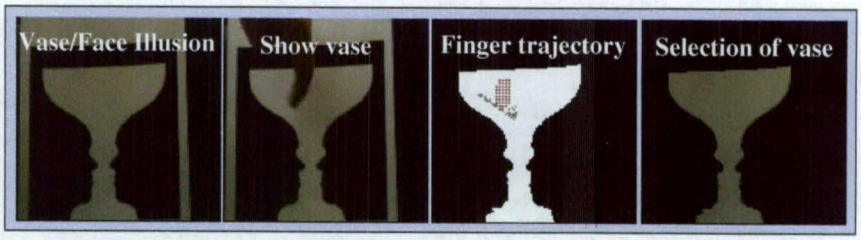


Figure 4-12: Solving the vase-figure illusion. Having a human tapping on the vase extracts the vase percept, instead of the faces percept. The third image from the left shows both sampled points of a human finger and the finger's endpoint trajectory.

is segregated just by having a human actor waving above it (see figure 4-12).

4.3.3 Perceptual Grouping of Spectral Cues

Another important property of objects is the texture of their surfaces – and texture has complementary properties to color. Texture is closely related to the distribution both in space and frequency of an object's appearance. Therefore, Gabor filters and Wavelets are tools often applied to solve the texture segmentation problem (Tuceryan and Jain, 1993) (see (Weldon et al., 1996) for Gabor based and (Laine and Fan, 1993) for wavelets based texture segmentation).

Texture Segmentation

Objects templates will be segmented into regions of similar texture using a standard texture segmentation approach, as follows. A Wavelet transform is initially applied to the image to be segmented. This is approximately equivalent to a family of Gabor

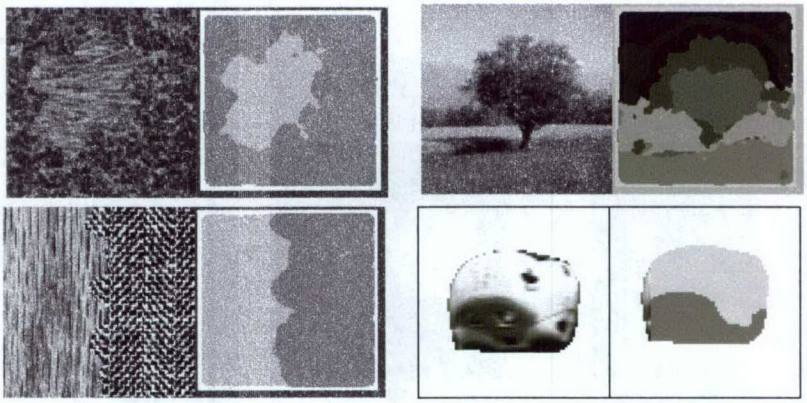


Figure 4-13: Texture segmentation. The number of texture clusters was determined automatically.

filters sampling the frequency domain in a log-polar manner. The original image is correlated with a Daubechies-4 filter using the Mallat pyramid algorithm (Mallat, 1989), at two levels ($N = 2$) of resolution (Strang and Nguyen, 1996), resulting in $n = 4 + 16 = 20$ coefficient images (see figure 4-13). All these coefficient images are then up-sampled to the original size, using a 5×5 gaussian window for interpolation. This way, more filtering is applied to the coefficients at the N^{th} level. For each pixel, the observation vector is then given by n wavelet coefficients. A mixture of gaussians is then applied to probabilistically model the input data by clusters. It is therefore necessary to learn the parameters for each cluster and the number of clusters. The former is estimated using the expectation-maximization (EM) algorithm (Gershenfeld, 1999). The latter uses an agglomerative clustering approach based on the Rissanen order identification criteria (Rissanen, 1983). The image is then segmented according to the cluster of gaussians: a point belongs to an image region if it occurs with maximum probability for the corresponding gaussian.

Grouping Textures

We can then apply a modified version of the perceptual grouping from human demonstration method to group perceptual (texture) cues, replacing the standard color segmentation algorithm applied to a stationary image by the texture segmentation algorithm. This approach is especially useful to segment objects with a homogeneous texture but heterogeneous colors (see figure 4-14).

4.3.4 Improving Texture Segmentation Accuracy

A more advanced and complex texture segmentation algorithm – the normalized cuts algorithm (Shi and Malik, 2000) – can be applied for improved performance, to divide the images into texture clusters which might then be grouped together through human-robot interactions. Therefore, we demonstrate in figures 4-15 and 4-16 how

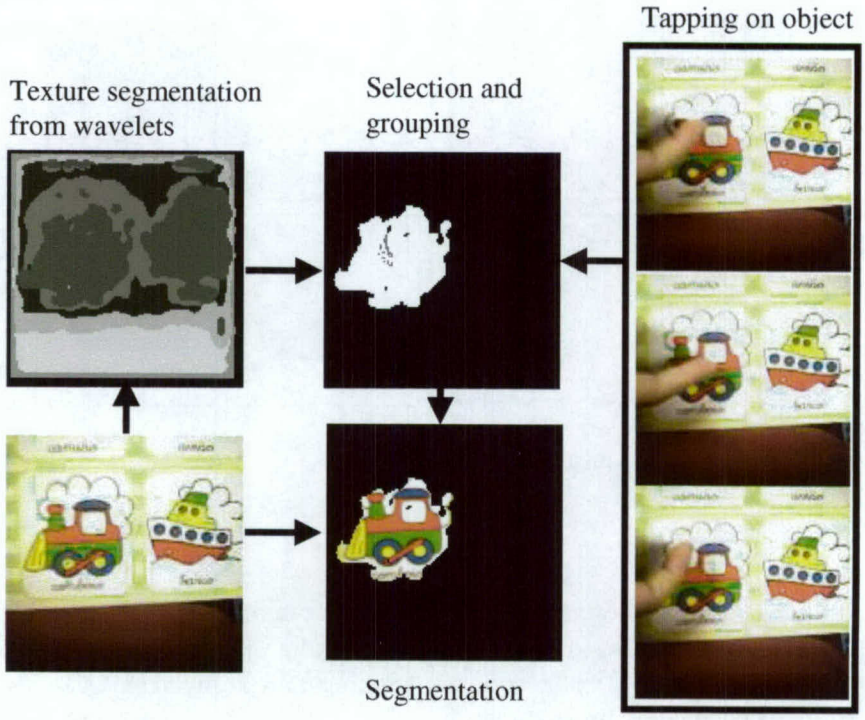


Figure 4-14: Texture segmentation of an image and grouping of texture clusters. The method is applied successfully for a case in which the grouping for color regions fails, due to color similarity with background (the white color).

to improve experimental segmentation results originally presented by (Shi and Malik, 2000), by grouping texture clusters into meaningful percepts of a church and the sky, and a mountain.

We have just seem in this section a strategy for a human to transmit to the robot information concerning objects appearance. This information was fundamentally described in a 2-dimensional space (retinal coordinates). We now move forward to demonstrate perceptual grouping of 3D information from human-robot interactions. The next section presents a new algorithm that extends the power of the perceptual grouping algorithm to make this possible.

4.4 Depth from Human Cues

Besides binocular cues, the human visual system also processes monocular data for depth inference, such as focus, perspective distortion, among others. Previous attempts have been made in exploring scene context for depth inference (Torralba and Oliva, 2001). However, these passive techniques make use of contextual clues already

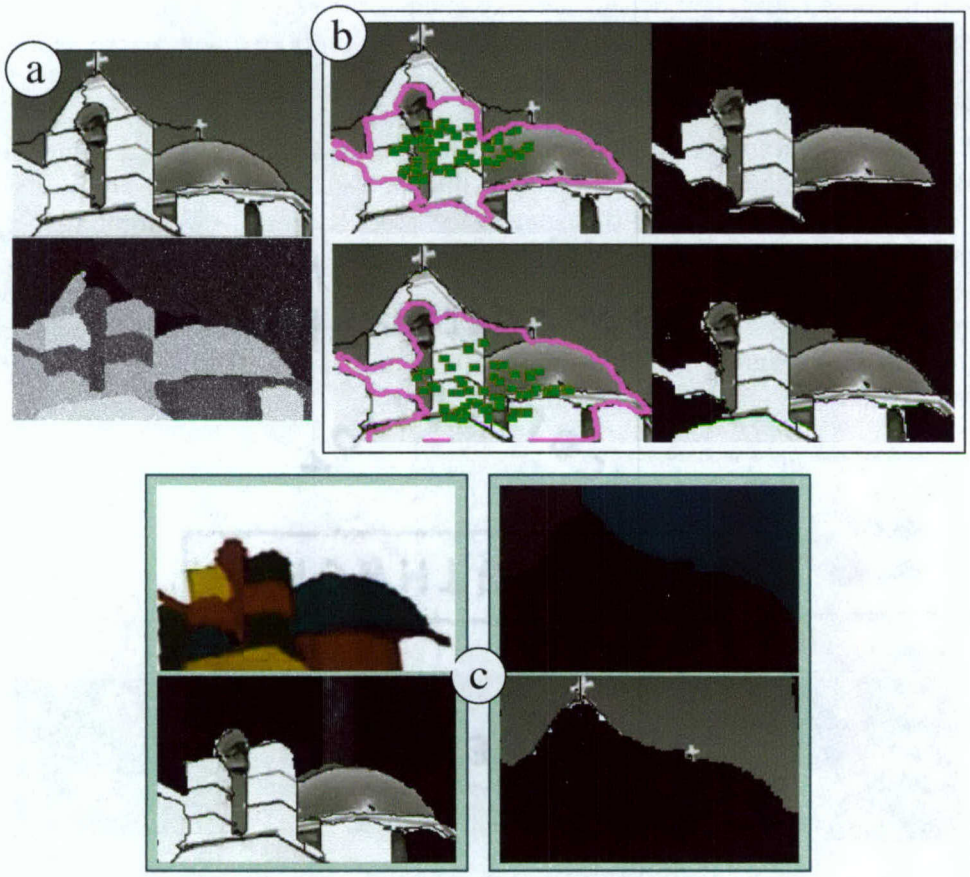


Figure 4-15: Improving normalized cuts' segmentation results. Texture clusters segmented from an image (a) by a normalized cuts algorithm (Shi and Malik, 2000) are grouped together (b) into a church, and into the church and the sky (c), by having a human showing the picture of the church to the robot and tapping on it.

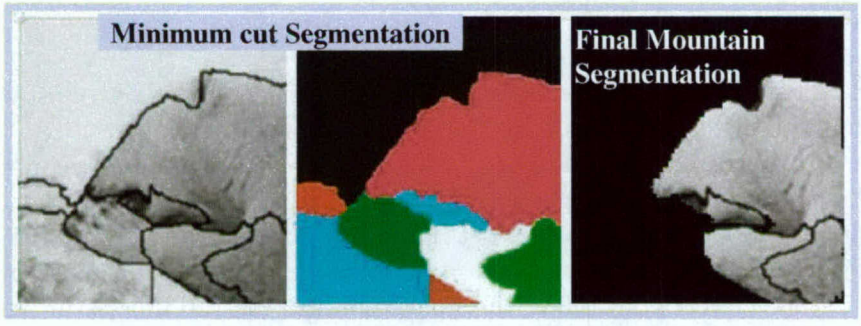


Figure 4-16: Texture clusters segmented from a nature scene (Shi and Malik, 2000) are grouped together into a mountain through human demonstration.

present in the scene. They do not actively change the context of the scene through manipulation to improve the robot's perception. We propose a new active, embodied approach that actively changes the context of a scene, extracting monocular depth measures.

The world structure is a rich source of information for a visual system – even without visual feedback, people expect to find books on shelves (because of world functional constraints stored in their memories – see chapter 8). We argue that world structural information should be exploited in an active manner (Arsenio, 2004g). For instance, there is a high probability of finding objects along the pointing direction of a human arm (Perrett et al., 1990). In addition, a human can be helpful for ambiguity removal: a human hand grabbing a Ferrari car implies that the latter is a toy car model, instead of a real car (see figure 4-17).

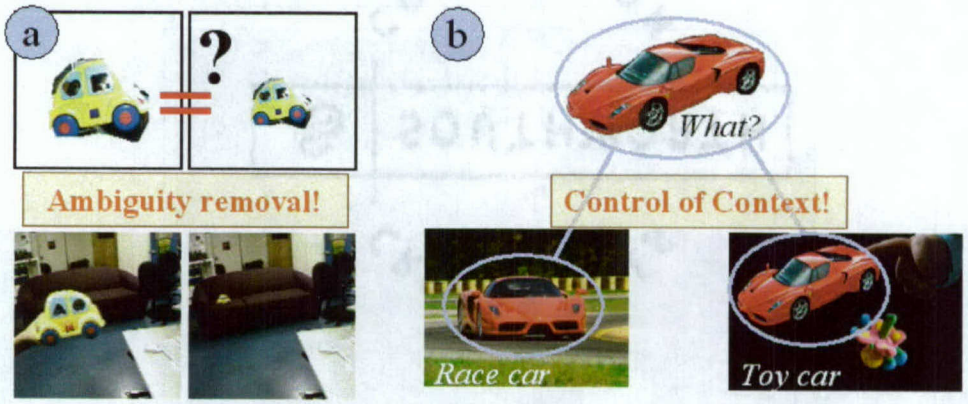


Figure 4-17: Control of contextual information in a scene. (a) Top: Two images: same object at different depths? or distinct size objects? Bottom: Contextual information removes ambiguity. The background image provides enough information for a human to infer that probably its the same object at different depths (b) Information from viewing a car without being in context may result in categorical ambiguity. If the car is viewed on a race track with trees, then it is probably a real race car. But for toy cars, a human can easily control context, introducing contextual references by manipulating the object.

Hence, the meaning of the image of an object depends usually on other visual information – the image context. a function of the surrounding context. But humans can control this image context to facilitate the acquisition of percepts from the robot's visual system. We propose a real-time algorithm to infer depth and build 3-dimensional coarse maps for objects through the analysis of cues provided by an interacting human. Depth information is extracted from monocular cues by having a human actor actively controlling the image context (as the images in figure 4-18 show). Transmission of world-structure to the perceptual system results therefore from the action of an embodied agent. A distinct monocular cue will be processed: the relative size of objects in a monocular image (also called familiar size by Gestalt psychologists). Special focus will be placed on using the human's arm diameter as a

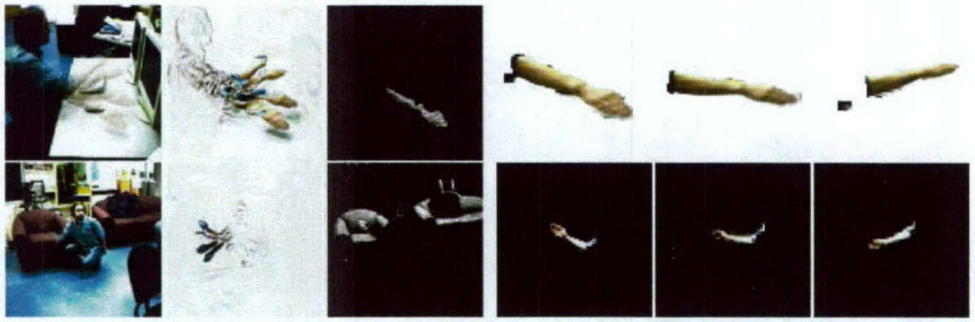


Figure 4-18: Human waving the arm to facilitate object segmentation. Upper row shows a sequence for which the skin-tone detector performs reasonably well under light saturation. Lower row shows background sofas with skin-like colors. The arm's reference size was manually measured as 5.83 pixels, while the estimated value was 5.70 pixels with standard deviation of 0.95 pixels.

reference measure for extracting relative depth information.

Therefore, the algorithm consists of automatically extracting the apparent size of objects and their depth as a function of human arm diameter. This diameter measure solves the image ambiguity between the depth and size of an object situated in the world (figure 4-17). Our technique relies on the assumption that the human actor waves his arm near to the object to be segmented. But it would make no sense for a human to show the features which compose an object to the robot, while staying far away from the object. Like children, one expects to find interesting objects from another person's gestures close to, or along the direction of, his gesturing arm (Perrett et al., 1990).

The human arm diameter (which is assumed to remain approximately constant for the same depth, except for degenerate cases) is extracted from periodic signals of a human hand as follows:

1. Detection of skin-tone pixels over a image sequence
2. A blob detector labels these pixels into regions
3. These regions are tracked over the image sequence, and all non-periodic blobs are filtered out
4. A region filling algorithm (8-connectivity) extracts a mask for the arm
5. A color histogram is built for the background image. Points in the arm's mask having a large frequency of occurrence in the histogram are labelled as background.
6. The smallest eigenvalue of the arm's mask gives an approximate measure of a fraction of the arm radius (templates shown in figure 4-18).

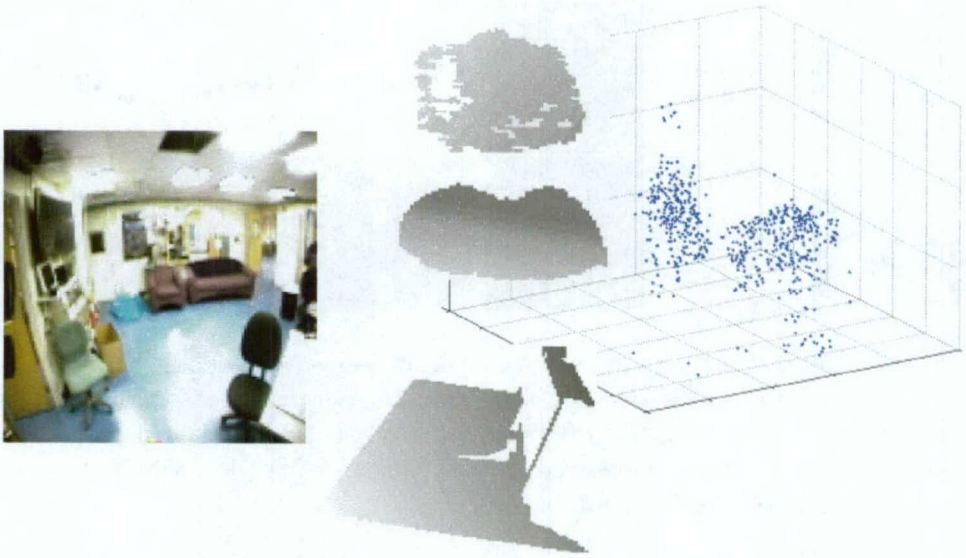


Figure 4-19: (left) An image of the lab. (right) Depth map (lighter=closer) for a table and a chair. Perpendicularity is preserved for the chair's disconnected regions (3D plot).

Once a reference measure is available, it provides a coarse depth estimation in retinal coordinates for each arm's trajectory point. The following factors affect the depth estimation process (these findings are supported by the error analysis presented in table 4.1 and object reconstruction results in figure 4-19):

Light sensitivity This is mainly a limitation of the skin-color algorithm. We noticed a variation in between $10 - 25\%$ on size diameter for variations in light intensity (no a-priori environment setup – the only requirement concerns object visibility). High levels of light exposure increase average errors.

Human arm diameter variability Variations along people diversity are negligible if the same person describes objects in a scene to the visual system, while depth is extracted relative to that person's arm diameter.

Background texture interference The algorithm that we propose minimizes this disturbance by background removal. But in a worst case scenario of saturated, skin-color backgrounds, the largest variability detected for the arm's diameter was 35% larger than its real size.

Hence, we argue that this technique provides coarse depth estimates, instead of precise, accurate ones, as reported by the experimental results in table 4.1. The average depth of an object can be estimated by averaging measures using a least squares minimization criterium – errors are even further reduced if large trajectories are available.

Errors in avg. depth	T	N	Mean error	Error Std	Mean abs error
big sofa	46	305	6.02	19.41	17.76
small sofa	28	326	-7.79	19.02	18.16
black chair	31	655	8.63	3.95	8.63
table	17	326	15.75	5.80	15.75
door	11	126	8.00	34.94	27.05
green chair	24	703	17.9	3.87	17.93

	S	T	N	Mean error
big sofa	H	7	146	27.96
	L	39	384	1.84
small sofa	H	25	369	-12.2
	L	3	158	33.4

S	Mean error	Error Std	Mean abs error
H	0.74	25.3	16.35
L	-0.98	10.6	7.44

Table 4.1: Depth estimation errors for objects from 5 scenes (as percentage of real size). T stands for number of templates, N for average number of trajectory points per template, S for light source, H and L for High/Low luminosity levels, respectively. Errors are given by (human arm diameter as projected in the retina - human arm diameter manually measured from the image real)/(manual image measurement of arm diameter). (left) Overall results – average absolute errors in extracting the human arm diameter are $\approx 20\%$. (right) Depth errors for different luminosity conditions are shown for the two sofas – top – and from all objects– bottom.

But since a collection of 3D trajectories (2-dimensional positions and depth) are available from temporally and spatially closed segmentations, it is also possible to determine a coarse estimate for the shape of an object from such data. A plane is fitted (in the least square sense) to the 3D data, for each connected region in the object's template – though hyper-planes of higher dimensions or even splines could be used. Outliers are removed by imposing upper bounds on the distance from each point to the plane (normalized by the data standard deviation). Such fitting depends significantly on the area covered by the arm's trajectory and the amount of available data. The estimation problem is ill-conditioned if not enough trajectory points are extracted along one object's eigendirection. Therefore, the fitting estimation depends on the human description of the object – accuracy increases with the area span by the human trajectories and the number of extracted trajectory points.

It should be emphasized that this thesis does not argue that this algorithm provides more accurate results than other stereo or monocular depth inference techniques. Indeed, there is a wide selection of algorithms available in the literature to infer depth or shape (Horn, 1986; Faugeras, 1993; Hartley and Zisserman, 2000):

- ▷ Monocular techniques for depth inference include, among others, Depth from Motion or Shading (Horn, 1986), Depth from Disparity (Faugeras, 1993), Depth from Focus (Krotkov et al., 1990) and Shape from Texture (Forsyth, 2001).

- ▷ Stereo techniques most often extract depth from a binocular system of cameras (Arsenio and Marques, 1997) or by integrating multiple simultaneous views from a configuration of several cameras. (Hartley and Zisserman, 2000; Faugeras et al., 2001).

Unlike some of these methods, the technique here proposed provides only coarse depth information. Its power relies in providing an additional cue for depth inference (a statistical scheme which augments the power of this algorithm by using cues from other scene objects besides the human arm will be proposed in chapter 8). In addition, the proposed algorithm has complementary properties to other depth inference algorithms, it does not require special hardware (low-cost cameras will suffice) and it also outputs object segmentations. And there are cases in which it can really provide more accurate results than standard depth inference techniques. Examples of such cases include textureless images, or low resolution representations (e.g., a foveated camera looking at a completely white table or a white wall, with no visible boundaries). Stereo, Depth from Motion and most algorithms will fail for these cases, but not this thesis approach.

Chapter 5

Visual Pathway – What

The perception [of the Taj Mahal] is much richer and more detailed than the image... The image, in contrast, is vague, ill-defined, and internal.
(Palmer, 1999)

This thesis work focuses on the development of a broad cognitive system for a humanoid robot to perceive actions (see previous chapter), scenes, objects, faces and the robot itself. This chapter describes methods for learning about two of these categories: objects and people's faces. Section 5.1 presents an object recognition algorithm to identify objects previously learned in scenes, and section 5.2 deals with object recognition when an object template for the object is available. Recognition of faces is the topic of section 5.3, while section 5.4 introduces head pose estimation from the face gazings.

Cognitive Issues Visual object recognition and categorization occurs in the human brain's temporal lobe, along what has become called the "What Visual Pathway". Of course, this lobe is highly interconnected with other brain areas, since recognition is often a function of such factors as context or location, among others. By the age of 4-6 months, infants are already able to recognize their mother and to differentiate between familiar faces and strangers.

Developmental Issues Object detection and segmentation are key abilities on top of which other capabilities might be developed, such as object recognition. Both object segmentation and recognition are intertwined problems which can then be cast under a developmental framework: models of objects are first learned from experimental human/robot manipulation, enabling their a-posteriori identification with or without the agents actuation – as put by (Dennet, 1998): "...where the inner alchimist can turn it into useful products."

5.1 Object Recognition from Visual Local Features

Recognition of objects has to occur over a variety of scene contexts. This led to the development of an object recognition scheme to recognize objects from color, luminance and shape cues, or from combinations of them. The object recognition algorithm consists therefore of three independent algorithms. Each recognizer operates along (nearly) orthogonal directions to the others over the input space. This approach offers the possibility of priming specific information, such as searching for a specific object feature (color, shape or luminance) independently of the others. For instance, the recognizer may focus the search on a specific color or sets of colors, or look into both desired shapes and luminance.

Color, luminance and shape will be used as object features. For each of them, a recognizer was developed over the affine or similarity topological spaces of the *visual stratum* (Faugeras, 1995).

5.1.1 Chrominance/Luminance Attributes

Traditional color histograms techniques (Swain and Ballard, 1991) compute the frequency of occurrence for each pixel color over the whole image. These algorithms will fail most of the time for recognizing an object in an image, because matching occurs globally instead of locally - they do not account for information concerning the number of chrominance/luminance clusters in the image, and the relative proximity between these clusters. Indeed, for the flags shown in figure 5-1, color histograms will not discriminate between the Czech Republic (left) and Costa Rica (right) flags, since they have the same color histograms (equal number of pixels of three different colors). However, these traditional approaches do not account for information concerning the number of color/luminance clusters in the image, and the relative proximity between these clusters.

The method presented here is more powerful than standard color histogram methods. Color histograms are used just to create chrominance/luminance regions (following the approach of chapter 3), while geometric relationships among these regions guide the recognition process.

The input space for the chrominance recognition algorithm is defined by color histograms over spatially connected (8-connectivity) chrominance regions. Each chrominance region contains a set of similar colors within a given tolerance (color buckets).

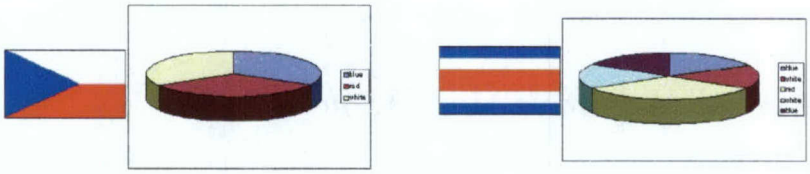


Figure 5-1: Region based Histograms versus Standard Histograms. Color histograms do not discriminate between the Czech Republic (left) and Costa Rica (right) flags. Indeed, the color histograms for both of them are identical. But the region based histogram for the Costa Rica flag is distinct.

For the luminance algorithm, the input state is defined by color histograms over connected regions of the image with similar levels of luminosity.

The input space consists of the chrominance/luminance values assigned to triples of connected regions, and the oriented ratio of these regions' areas. These features are invariant to affine transformations. The affine space preserves incidence and parallelism, allowing a large spectrum of (affine) transformations. This approach is view independent, unless the affine space is not a reliable approximation of the projective one (true for large perspective deformations or occlusions).

The search space is constrained by limiting the search to combinations of the five largest regions connected to a specific region. This significantly prunes the search space for highly connected regions by imposing bounds on the combinatorial explosion. In order to guarantee the consistency of the solutions - as well as to locate an object in the image - the center of mass of an object (which is invariant to affine transformations) is also stored, together with the coordinates of the regions' centroids. Affine models, which are computed for all hypothesized matches, are required to be coherent (under uncertainty accounted by the Mahalanobis distance) for all the features belonging to an object.

Notice that geometrical information is not exploited by these algorithms, since the goal is to make these chrominance/luminance recognition algorithms independent of such properties. However, shape properties are recognized independently, as described below.

5.1.2 Shape - Geometric Features

Another potential description for an object is through its geometric shape. Geometric features will consist of pairs of oriented lines (detected according to the chapter 3 method for estimating geometric features).

The input space consists of similarity invariants (the angle between the two edges and the ratio of their length). The search space is pruned by considering only pairs of oriented edge lines, with one of them intersecting a fixed neighborhood of any point of the other. Coherence is imposed by computing a similarity transformation for each hypothesized matching on the object, using both lines' midpoints and their point of intersection. These points define uniquely the transformation. Consistency

is enforced for all object transformations.

5.1.3 Nested, Adaptive Hash tables

Geometric hashing (Wolfson and Rigoutsos, 1997) is a rather useful technique for high-speed performance. In this method, invariants (or quasi-invariants) are computed from training data in model images, and then stored in hash tables. Recognition consists of accessing and counting the contents of hash buckets.

We applied Adaptive Hash tables (a hash table with variable-size buckets) to introduce hashing improvements. This technique, which consists of applying variable boundaries to the hash buckets, was motivated by the fact that test data points often fall near the boundary of a bucket, while training data points might have fallen on an adjacent bucket. Hence, whenever consistency fails on a bucket, if the distance between the test point and the average point of an adjacent bucket is smaller than the distance between the bucket centers, then a match is hypothesized for input features on the adjacent bucket. This considerably improved the matching results.

In addition, nested hash tables were used for classification. A training hash table is computed to store normalized values for all permutations over the space of the input features, along with references to the model and the set of input features that generated them. During classification, features are mapped into buckets of this fuzzy hash table. A second hash table is then built to store all coherent transformation models, and to detect peaks over the correspondent topological space.

5.1.4 The Visual Binding Problem

Visual binding consists of putting together all properties into an unified perceptual object. As stated in (Palmer, 1999):

> *"...without some mechanism to bind properties into objects properly, an observer would not be able to tell the difference between a display containing a red circle and a blue triangle and a display containing a blue circle and a red triangle.*

Binding is therefore a very important mechanism, and Treisman (Treisman, 1985) suggested a strategy of visual attention to model this mechanism: feature integration theory.

Feature Integration Theory

Evidence from cognitive perception suggests that feature integration during visual search is serial for conjunction features – features resulting from combining several properties, e.g., red vertical line (combining red color with vertical line) or triangle (combining three lines) – being approximately parallel for non-conjunction features (Wolfe, 1994; Palmer, 1999). Visual searches concerning elementary properties will be performed in *parallel*, as proposed by Treisman's model (Treisman and Gelade, 1980). Indeed, an elementary feature search needs to use just one identification channel,

while all the other channels are irrelevant for such task. This is in stark contrast with conjunction searches (e.g., dark green), where one needs to run all algorithms that recognize the properties being searched. Notice that elementary properties are searched in parallel because they do not depend (or the dependence is rather weak) on the number of distractors, as shown in figure 5-2.

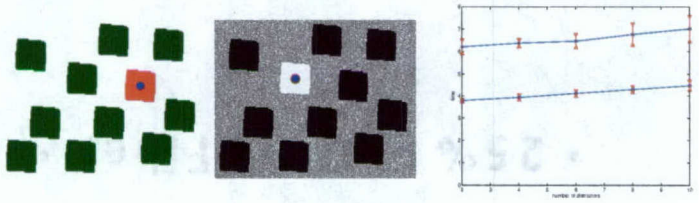


Figure 5-2: Pop-out in visual search. "Pop-out" is defined in the text. (left) Pop-out of a color property. The target is the red color, and green buckets are the distractors. (center) Pop-out of a luminance property. The white luminance level is the target. (right) Processing times for color – top line – and luminance – bottom line – properties.

Let us consider the following scenario. Given an image, find a target feature in the image for a given query. Target features can be described in terms of elementary low-level features such as color, luminance and shape, or by combinations of them (called features conjunction).

Visual Pop-Out: Feature integration theory predicts that whenever a target object can be discriminated from distractor objects by an elementary feature, visual pop-out should occur (Palmer, 1999).

Elementary features correspond to unit color or luminance regions, or an entity with a single line. For such searches, no hash table needs to be used. It is enough to compute the color/luminance regions or to detect the oriented lines on an image. Figure 5-2 shows results for such color/luminance searches, together with the increase in processing time with the number of distractors. Distractors are features in the field of view which are not the target of the visual search, and hence might not only be confounded by the target, but might also cause interference with the search efficiency. That is not the case for non-conjunction searches, which are approximately parallel since the (slightly increasing) step ratio for both lines is very small.

Conjunction Search: Treisman's theory predicts that pop-out should not occur for conjunction searches. Among conjunction searches it is important to distinguish between Within features conjunction and Between features conjunction. Within features conjunction searches (Wolfe, 1994) – which consist of a conjunction of properties within the same class (e.g., the yellow and green object) – are implemented by carrying out recognition tasks along orthogonal geometric-chrominance-luminance channels. This is in contrast with the method of hashing with rich features proposed by (Fitzpatrick, 2003b), which composes appearance (color between orientation regions) and geometric features (angle between a pair of oriented regions plus ratio of

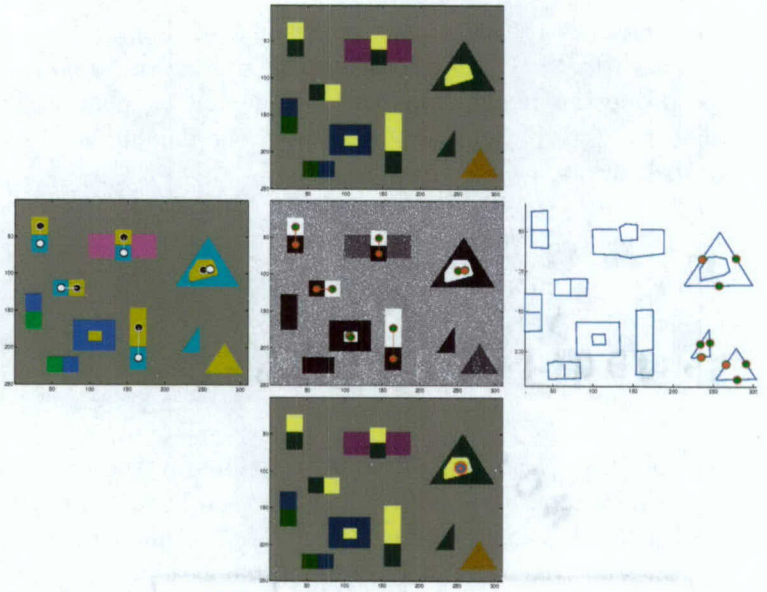


Figure 5-3: Conjunction searches. The image in the top row shows the original image. Middle row, from left to right, shows within features conjunction searches (conjunction of properties within the same class): normalized color buckets for the original image, with results for a yellow-green query (results are shown superimposed); luminance buckets of the original image, together with query results for a dark-light object; and search for triangles (conjunction of three oriented lines). Bottom row shows a between features conjunction search (conjunction of properties from different classes): target identification (a conjunction of features) among distracters.

these regions' sizes) into a generic input feature. Although recognition based solely on geometric features (and restrained to a similarity Topology) was also reported by (Fitzpatrick, 2003b), standard color histograms were applied for appearance based recognition (with all their limitations).

Lets then demonstrate the power of independent channels, by priming an identification task by a color cue (conjunction search of yellow and green identities), by a luminance cue (conjunction search of dark and light identities) and by a geometric cue (triangle-like identities). Results for such queries are shown in figure 5-3.

Between features conjunction searches (Wolfe, 1994) – which consist of a conjunction of properties from different classes (e.g., the green triangle) – are performed by intersecting the feature maps for all the channels. For the experiment in figure 5-3, a conjunction search for a light yellow - dark green triangle requires **feature integration** along all channels to locate and identify the target. Conjunctions of different properties require running algorithms over more channels - and hence a larger computational load.

The problem of recognizing objects in images can be framed as the visual search for:

- ▷ an elementary feature – if the object previously learned is only defined by a single feature,
- ▷ within features conjunction – in case the object is composed by many features of the same class, which is often the case
- ▷ between features conjunction – if the object is to be found in the image from the combination of features from different classes (such as color and shape).

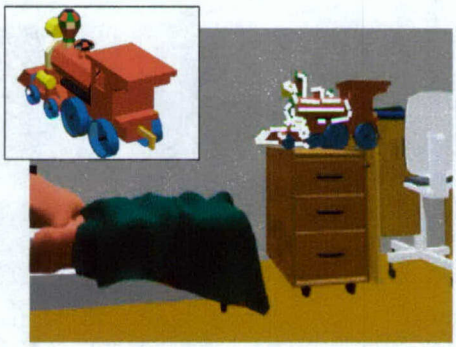


Figure 5-4: Object recognition and location. Train template appears under a perspective transformation in a bedroom scene - generated by the *DataBecker*© Software. Estimated lines are also shown. Scene lines matched to the object are outlined.

Figure 5-4 presents an experimental result for finding a toy train appearing under a large perspective transformation in a simulated bedroom scene. The train has a colorful appearance and its contours include several lines, hence this experiment corresponds to a conjunction search. Since only one of the three recognition channels is used, this experiment corresponds to a within features conjunction search.

5.2 Template Matching – Color Histograms

Whenever there is a priori information concerning the locus of image points whether to look for an object of interest (from prior segmentation, for instance), a simpler template matching approach suffices to solve the recognition problem for a wide set of object categories (assuming they look different).

A human-computer interactive approach was previously described to introduce the humanoid robot Cog to knowledge concerning the visual appearance of objects. Percepts extracted as object templates from the robot's surrounding world are then converted into an useful format through a template matching based object recognition scheme, which enables the robot to recognize object templates under different perspective views. This object recognition algorithm needs to cluster object templates by classes according to their identity. Such a task was implemented through color histograms – objects are classified based on the relative distribution of their color pixels. Since object's masks are available, external global features do not affect recognition,

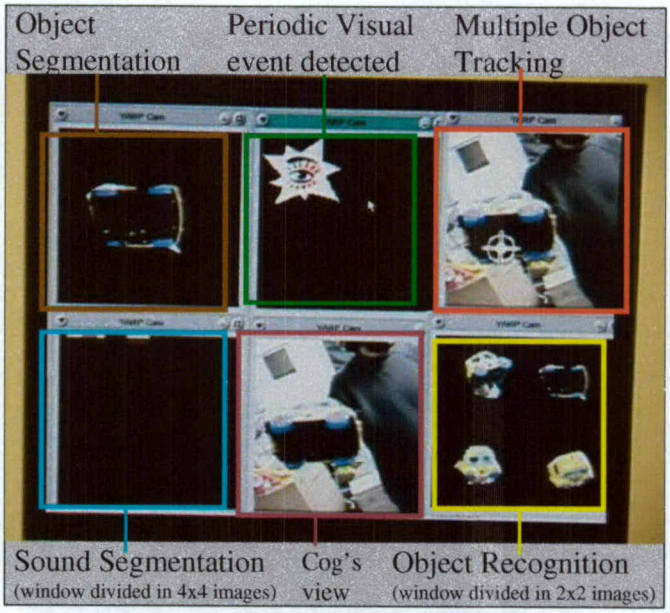


Figure 5-5: Visual segmentations are used to initialize a multi-target tracking algorithm, to keep track of the objects' positions. The object recognition algorithm, which matches templates based on color histograms, is shown running.

and hence color histograms are now appropriate. A multi-target tracking algorithm (which tracks *good features*(Shi and Tomasi, 1994) using a pyramidal implementation of the Lucas-Kanade algorithm, and described in detail in chapter 8) keeps track of object locations as the visual percepts change due to movement of the robot's active head. Ideally, a human actor should expose the robot to several views of the object being tracked (if the object appearance is view-dependent), in order to link them to the same object.

Recognition works as follows. Quantization of each of the three CYL color channels (see chapter 3) originates 8^3 groups G_i of similar colors. The number of image pixels n_{G_i} indexed to a group is stored as a percentage of the total number of pixels. The first 20 color histograms of an object category are saved into memory and updated thereafter. New object templates are classified according to their similarity with other object templates previously recognized for all object categories (see figure 5-5), by computing $p = \sum_{i=1}^{8^3} minimum\left(n_{G_i}, n_{G_i'}\right)$. If $p < 0.7$ for all of the 20 histograms in an object category, then the object does not belong to that category. If this happens for all categories, then it is a new object. If $p \geq 0.7$, then a match occurs, and the object is assigned to the category with maximum p.

Whenever an object is recognized into a given category, the average color histogram which originated a better match is updated. Given an average histogram which is the result of averaging m color histograms, the updating consists of computing the weighted average between this histogram (weight m) and the new color histograms (unit weight). This has the advantage that color histograms evolve as

more samples are obtained to represent different views of an object. In pseudo-code:

Given:

- ▷ a batch of n query images (from tracking) of object templates (no background)
- ▷ training data: 20 averaged color histograms in memory for each of the m object categories learned

Recognize the set of objects:

- ▷ for $k = 1, \ldots, n$,
  1. Compute color histograms for query image k (denoted G^k)
  2. set best=(-1,-1,-1)
  3. for $l = 1, \ldots, m$ (for each object category l in the database of m categories)
     (a) set $p_{max} = (-1, -1)$
     (b) for $j = 1, \ldots, 20$,
        i. compute the probability $p = \sum_{i=1}^{8^3} minimum\left(n_{G_i}, n_{G_i'^{lj}}\right)$, where G'^{lj} denotes the j average color histogram in category l
        ii. if $p \geq 0.7$
        - $p_{max} = maximum_p(p_l, (j, p)))$
     (c) if $p_{max} \neq (-1, -1)$ (a matching occurs for this category)
        - $best_k = maximum_p(best_k, (l, j, p))$, where j and p are the two elements of p_{max} (in that order)
- ▷ set category(1,...,m+1)=0
- ▷ for $k = 1, \ldots, n$,
  1. if $best_k \neq (-1, -1, -1)$ (a match occurred)
     - category(l)=category(l)+1, where l is the category given by the first element of $best_k$
  2. else (no match occurred)
     - category(m+1)=category(m+1)+1
- ▷ find max_{cat}, the index of the maximum value in category
- ▷ if $equal(max_{cat}, m + 1)$ (create new object category)
  1. Set $m = m + 1$, $minc = minimum(category(max_{cat}), 20)$
  2. initialize the average color histograms $G'^{m,\{1,\ldots,minc\}}_{1,\ldots,8^3} = G^{1,\ldots,minc}_{1,\ldots,8^3}$
  3. $histogramsavg_m(\{1, \ldots, minc\}) = 1$

  else (match - update object category max_{cat})
  - for $k = 1, \ldots, n$
    1. set j as the second element of $best_k$
    2. update (by weight average) the average histogram $G'^{max_{cat},j}$ with G^k

$$G'^{max_{cat},j}_{1,\ldots,8^3} = \frac{histogramsavg_m(j) \times G'^{max_{cat},j}_{1,\ldots,8^3} + G^k_{1,\ldots,8^3}}{histogramsavg_m(j) + 1}$$

    3. $histogramsavg_{max_{cat}}(j) = histogramsavg_{max_{cat}}(j) + 1$
- ▷ output identifier max_{cat}

Recog. objects	Errors %	Recog. objects	Errors %
Big sofa	**4.35** (23)	Green chair	**0.0** (4)
Small sofa	**18.2** (11)	Door	**3.57** (28)
Table	**5.56** (18)	Blue door	**14.29** (7)
Black chair	**0.0** (18)	Total	**5.5** (109)

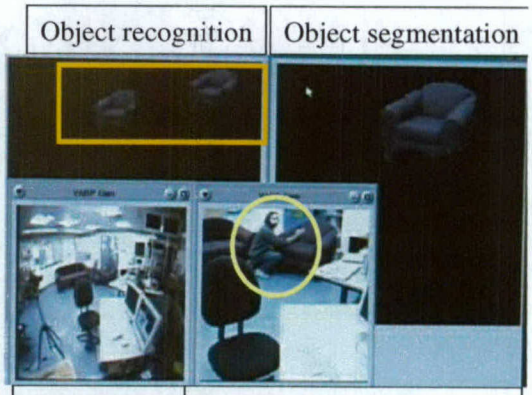


Table 5.1: (left) Recognition errors. It shows the number of matches evaluated from a total of 11 scenes (objects may be segmented and recognized more than once per scene). The number in parenthesis shows the total number of times a given object was recognized (correctly or incorrectly). Incorrect matches occurred due to color similarity among big/small sofas or between different objects. Missed matches result from drastic variations in light sources (right) sofa is segmented and recognized.

5.2.1 Other Object Recognition Approaches

Previous object recognition work using exclusively local (intrinsic, not contextual) object features (computed from object image templates) for performing object recognition tasks are mostly based in object-centered representations obtained from these templates. (Swain and Ballard, 1991) proposed an object representation based on its color histogram. Objects are then identified by matching two color histograms – one from a sample of the object, and the other from an image template. The fact that for many objects other properties besides color are important led (Schiele and Crowley, 2000) to generalize the color histogram approach to multidimensional receptive field histograms. These receptive fields are intended to capture local structure, shape or other local characteristics appropriate to describe the local appearance of an object.

A kalman filter based approach for object recognition was presented by (Rao and Ballard, 1997), and applied to explain neural responses of a monkey while viewing a natural scene. (Zhang and Malik, 2003) described a shape context approach to learn a discriminative classifier. (Papageorgiou and Poggio, 2000) approach used instead Haar Wavelets to train a support vector machine classifier, while (Freeman and Adelson, 1991) applied steerable filters.

Another set of methodologies apply the Karhunen-Loeve Transform (KLT) (Fukunaga, 1990) or Principal Component Analysis (PCA) for the calculation of eigenpictures to recognize faces (Turk and Pentland, 1991; Moghaddam and Pentland, 1995) and objects (Murase and Nayar, 1995; Ohba and Ikeuchi, 1996). The KLT approach yields a decomposition of a random signal by a set of orthogonal functions with un-

correlated coefficients. Its advantage is that a small number of coefficients is enough to represent each image, resulting in both efficient storage and classification. PCA reduces dimensionality by only using the KLT functions that account for the maximal variability of the signal described. Following these approaches, eigenfaces for face recognition will be the topic of section 5.3, while eigensounds for sound recognition will be later described in chapter 8.

The approach this manuscript describes for object recognition from image templates is similar to the one in (Swain and Ballard, 1991). Our contributions are therefore of different nature, consisting of: i) the automatic generation of training data (color histograms of the object templates) for object recognition, and ii) the integration of this recognition scheme to guide many other machine learning algorithms – such as object recognition and location from contextual cues which will be described in chapter 6.

5.2.2 Experimental Results

Table 5.1 presents quantitative performance statistics for this algorithm (which was extensively applied for building maps of scenes, a topic described in the next chapter). The quantitative evaluation was performed from data extracted while running online experiments on Cog. The batch number was reduced to $n = 1$ (object templates were classified one at a time), and the training data consisted of stored object template images annotated by the tracking algorithm. This table also shows the system running on the humanoid robot Cog, while recognizing previously learned objects. Incorrect matches occurred due to color similarity among different objects (such as a big and a small sofa). Errors arising from labelling an object in the database as a new object result are chiefly due to drastic variations in light sources. Qualitative results from an on-line experiment of several minutes for object segmentation, tracking and recognition of new objects on the humanoid robot are shown in figures 5-6 and 5-7.

Out of around 100 samples from on-line experiments, recognition accuracy average was 95%. The recognition accuracy depends however on the object being recognized. For instance, several experiments have shown the algorithm not capable of differentiating among different people's faces, although it differentiated correctly between faces and other objects. This demonstrates the need for an independent algorithm for face recognition, which is described next.

5.3 Face Detection/Recognition

Humans are especially good at recognizing faces, and this skill is rather robust, considering the large variability of face features due to viewing conditions, poses, emotional expressions, and visual distractions such as haircut variability or glasses, among others. Face identification plays a major social role to convey identity and emotion, and also to extract information concerning others intentions and living habits.

The problem of face identification is organized as shown in figure 5-8, which also shows the similar approach used to recognize objects for which visual templates are

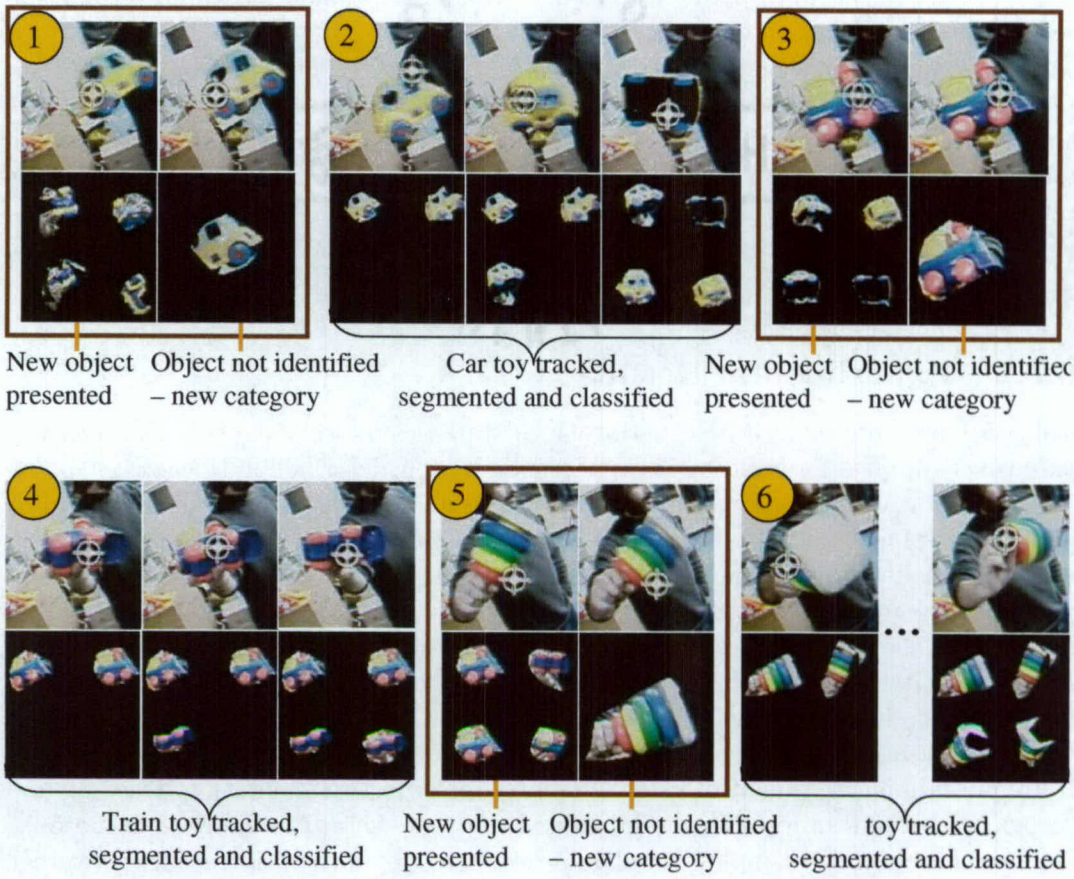


Figure 5-6: This sequence is from an on-line experiment of several minutes on the humanoid robot Cog. (1) A new, previously unknown object (a toy car) is presented. It is not recognized and a new category is created for it. (2) The robot is tracking a toy car (top row), and new template instances of it are being inserted into a database. A random set of templates from this database is shown on the bottom row. (3) A new object (a toy train) is presented. It was never seen before, so it is not recognized and a new category is created for it. (4) The toy train is tracked. (5) A new, unknown object presented, for which a new category is created on the object recognition database. (6) Templates for the new object are stored.

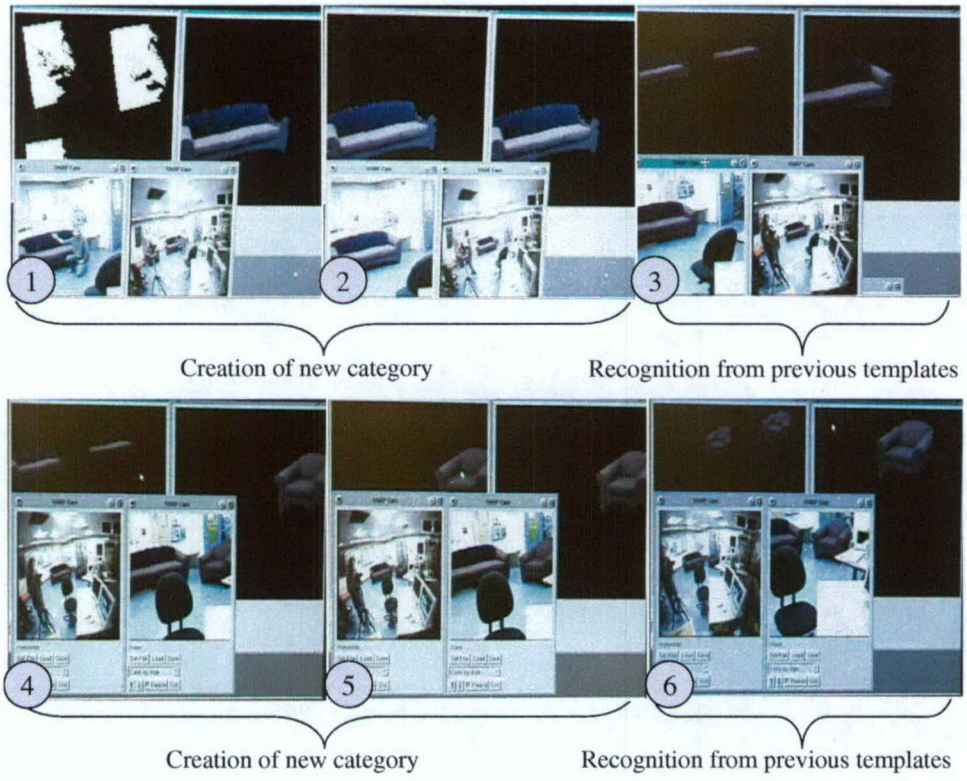


Figure 5-7: This illustrates a sequence from an on-line experiment of several minutes on the humanoid robot Cog. (1) The robot detects and segments a new object – a sofa; (2) New object is correctly assigned to a new category; (3) Object, not being tracked, is recognized from previous templates (as shown by the two sofa templates mapped to it); (4-5-6) Same sequence for a smaller sofa.

available. This approach generates training data automatically by detecting and tracking faces over time (tracker algorithm described in chapter 8). Batches of faces from one person (or batches of object's templates) are then inserted into the database. If more than 60% of this batch matches an individual (or object, respectively), it is recognized as such. Otherwise, a new entry is created on the database which corresponds to a new person (or a new object), and the learning algorithm is updated. Indeed, people often have available a few seconds of visual information concerning the visual appearance of other people before having to take a recognition decision, which is the motivation behind this approach.

Face Detection

Faces in cluttered scenes are located by a computationally efficient algorithm (developed by Paul Viola's group at MIT), which is applied to each video frame (acquired by a foveal camera). If a face is detected, the algorithm estimates a window con-

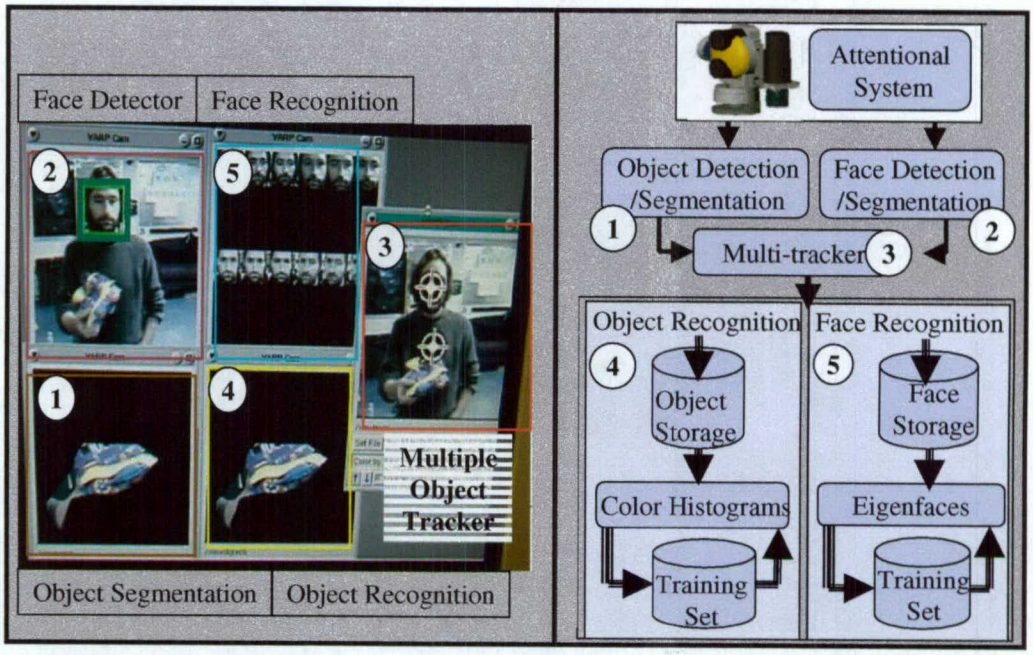


Figure 5-8: Approach for segmenting and recognizing faces and objects. Training data for object/face recognition is extracted by keeping objects and others faces in memory for a while, generating this way a collection of training samples consisting of multiple segmentations of objects and faces. (left) on-line experiment on Cog. 1) Object (train) segmentation, acquired by the active segmentation algorithm; 2) Face detection and segmentation using Jones and Viola algorithm; 3) Multiple object tracking algorithm, which is shown tracking simultaneously a face and the train; 4) Object Recognition window – this window shows samples of templates in the object database corresponding to the object recognized; 5) Face Recognition – this window shows 12 samples of templates in the face database corresponding to the face recognized. (right) schematic organization. 1) Object segmentation; 2) Face detection and segmentation; 3) Multiple object tracking; 4) Object Recognition; 5) Face Recognition.

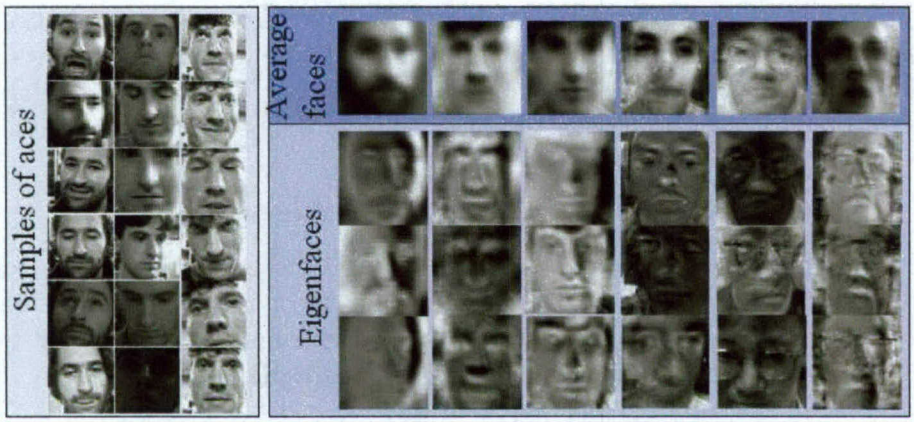


Figure 5-9: Six face image samples are shown for each of three people, to illustrate the face variability. The average face image for the six people on the database are shown, together with three eigenfaces for each one person.

taining that face, as shown in figure 5-8. The cropped image is then sent to the face recognition and gaze inference algorithms whenever a face is positively detected.

5.3.1 Face Recognition – Approaches

The scale, pose, illumination or facial expression can considerably change the geometry of a face. Attempts to model the latter based on a muscular point of view have been proposed (Sirovich and Kirby, 1987). Other methods include feature-based approaches in which geometric face features, such as eyebrow's thickness, face breadth, position and width of eyes, nose, and mouth, or invariant moments, are extracted to represent a face (Chellappa et al., 1995). However, feature extraction poses serious problems for such techniques (Chellappa et al., 1995).

Appearance-based approaches project face images into a linear subspace of reduced dimensions (Turk and Pentland, 1991). In order to efficiently describe a collection of face images, it is necessary to determine such a subspace – the set of directions corresponding to maximum face variability – using the standard procedure of Principal Component Analysis (PCA) on a set of training images. The corresponding eigenvectors are called eigenfaces – because they are face-like in appearance (see figure 5-9). Eigenfeatures, such as eigeneyes or eigenmouth for the detection of facial features (Belhumeur et al., 1997), is an alternative variant for the eigenfaces method.

Hence, the face recognition method we implemented – eigenfaces – is not new, having been the focus of extensive research. However, current methods assume off-line availability of training data. The novelty of our approach is the removal of this off-line constraint, as illustrated by figure 5-8. (Aryananda, 2002) presents as well an on-line training scheme for face recognition using eigenfaces. Her approach is to cluster the eigenfaces subspace into groups of coefficients from similar faces. However, this subspace is sparse, and clusters' boundaries are most often ill-defined. Instead

of relying in a clustering procedure, we rely on long interaction contacts between the robot and people – which are also useful for humans to develop individual relationships among us (Dautenhahn, 1995). Face constancy from these contacts is maintained by tracking people to gather a large set of faces which are labelled to the same person.

5.3.2 Face Recognition – Eigenfaces Algorithm

Our goal is not to develop a state-of-the-art, 99.99% effective recognition algorithm. Instead, we can live with much smaller recognition rates, as humans also often do.

The recognition steps are as follows (see (Turk and Pentland, 1991) for details). The training set of M face images from a person is represented as $\{\phi_1, \phi_2, \ldots \phi_M\}$ (see figure 5-9). The average face image of this set is defined by $\psi = 1/M \sum_{i=1}^{M} \phi_i$. The covariance matrix for the set of training faces is thus given by equation 5.1:

$$C_{\phi_n} = \frac{1}{M} \sum_{i=1}^{M} \Gamma_i \Gamma_i^T = AA^T \tag{5.1}$$

being $\Gamma_n = \phi_n - \psi$ the difference of each image from the mean, and $A = [\Gamma_1, \Gamma_2, \ldots, \Gamma_M]$. Let W and H be the image width and height dimensions, respectively, and $S = W \times H$ its size ($W = H = 128$). Determining the eigenvectors and eigenvalues of the S^2 size covariance matrix C is not tractable. However, since only a finite number of image vectors M are summed up, rank of C does not exceed $M - 1$. If the number of data points in the face image space is less than the dimension of the space $M < S^2$, there will only be $M - 1$ eigenvectors associated with non-zero eigenvalues, rather than S^2.

Let v_i be the eigenvectors of the $M \times M$ matrix $A^T A$. Simple mathematical manipulation shows that Av_i are the eigenvectors of $C = AA^T$. These vectors determine linear combinations of the M training set face images to form the eigenfaces μ_i:

$$\mu_i = \sum_{k=1}^{M} v_{ik} \Gamma_k, i = 1, \ldots, M \tag{5.2}$$

This way, processing is reduced from the order of the number of pixels S^2 in the images to the order of the number of images M in the training set. The number of basis functions can be further reduced from M to M' by selecting only the most meaningful M' eigenvectors (with largest associated eigenvalues) and ignoring all the others.

Classification of a face image ϕ consists of projecting it into eigenface components, by correlating the eigenvectors with it:

$$w_i = \mu_i(\phi - \psi), i = 1, \ldots, M' \tag{5.3}$$

for obtaining the coefficients w_i of this projection. The weights w_i form a vector $\Omega = \{w_1, w_2, \ldots, w_{M'}\}$ which represents how well each eigenface describes the input face image. A face is then classified by selection of the minimal L_2 distance to each object's coefficients in the database,

$$\varepsilon_\phi = \| \Omega - \Omega_k \| \tag{5.4}$$

where Ω_k describes the k^{th} face class in the database, and by selecting thereafter the class corresponding to the minimum distance. If ε_ϕ is below a threshold, then it corresponds to a new face. In pseudocode, the algorithm description is as follows:

Given:

- ▷ a batch of $n = 30$ query images (of the same face, as detected from tracking – chapter 8) of face templates (no background)
- ▷ training data: for each of the m categories learned, it stores in memory
  - an average face and the eigenfaces
  - a maximum of $h = 800$ face images, together with their projection coefficients into eigenfaces subspace ($\Omega'^{\{1,\ldots,h\},\{1,\ldots,m\}}$)

Recognize the set of faces:

- ▷ for $k = 1, \ldots, n$,
  1. set $best_k = (-1, MAX_{FLOAT})$
  2. for $l = 1, \ldots, m$ (for each face category l (a person) in the database of m categories)
     (a) project query face k into the category eigenfaces subspace, extracting solely the principal components M': $\Omega^{kl} = \{w_1^{kl}, w_2^{kl}, \ldots, w_{M'}^{kl}\}$
     (b) set $\varepsilon_\phi = MAX_{FLOAT}$
     (c) for $j = 1, \ldots, h_l$, ($h_l = h$ for category l)
        i. compute $\varepsilon_\phi^j = \| \Omega^{kl} - \Omega'^{jl} \|$
        ii. $\varepsilon_\phi = minimum(\varepsilon_\phi, \varepsilon_\phi^j)$
     (d) if $\varepsilon_\phi \geq th$ (a matching occurs for this category)
        – $best_k = minimum_{\varepsilon_\phi}(best_k, (l, \varepsilon_\phi))$,
- ▷ set category(1,...,m+1)=0
- ▷ for $k = 1, \ldots, n$,
  1. if $best_k \neq (-1, MAX_{FLOAT})$ (a match occurred)
     – category(l)=category(l)+1, where l is the category given by the first element of $best_k$
  2. else (no match occurred)
     – category(m+1)=category(m+1)+1
- ▷ find max_{cat}, the index of the maximum value in category
- ▷ if $equal(max_{cat}, m+1)$ (create new face category)
  1. Set $m = m + 1$
  2. compute the average face and the eigenfaces for this new category, together with the faces' projection coefficients into this eigenfaces subspace ($\Omega'^{\{1,\ldots,n\},max_{cat}}$), and store them
  3. set $h_m = n$ and store the n face images

  else (match - update face category max_{cat})
  - set $h_m = minimum(800, h_m + n)$ and store the additional n face images.
  - compute a new average face and the eigenfaces for this category, together with the new projection coefficients into this eigenfaces subspace for all h_m faces ($\Omega'^{\{1,\ldots,h_m\},max_{cat}}$), and store them

						
	**97**	1	6	3	4	0
	0	0	0	**44**	0	0
	2	1	5	0	**35**	0

Table 5.2: Confusion table for face recognition. Table show number of time the three faces in the vertical axis were matched to one of the six faces in the horizontal axis. The numbers on bold (97,44 and 35) correspond to correct matches, while the other correspond to false ones.

▷ output identifier max_{cat}

5.3.3 Experimental Results

The quantitative evaluation was performed off-line, from data extracted while running online experiments on Cog. The batch number was reduced to $n = 1$ (face images were classified one at a time), and the training data consisted of stored face template images annotated by the tracking algorithm. The training data set contains a lot of variation (see figure 5-9 for a few demonstrative samples). Validation data corresponds to a random 20% of all the data. The confusion table 5.2 shows results for recognizing three different people – the average recognition accuracy is 88.9%.

5.4 Head Pose Estimation

Head poses are estimated by applying the same eigenobjects based strategy used for face recognition (and therefore the eigenvectors are really eigenposes for this problem). The single algorithmic difference is that the input space, instead of including faces labelled by identity, consists of faces labelled by head gaze direction. But there is an important drawback between this algorithm and all the others: training data is segmented manually, off-line, for supervised learning.

The classification problem was separated into five category classes along two axis: left and right gaze, top and down gaze (although a finer resolution could be used – the algorithm could be extended to account for more classes), and front gaze. As shown in figure 5-10, head gazings vary considerably along a single direction.

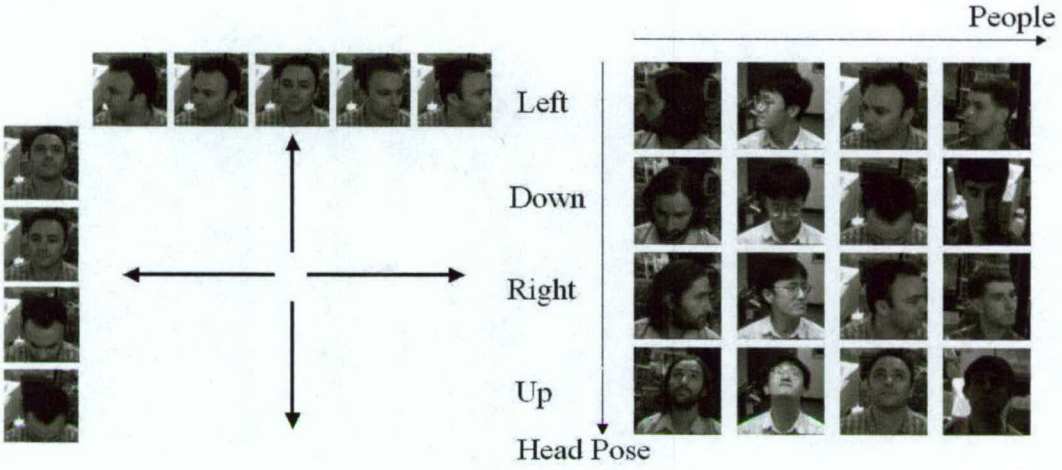


Figure 5-10: Head gaze directions are grouped into five classes along two directional axis. There is high variability along each axis, including partial occlusion of faces.

Experimental Results and Discussion: Validation data corresponds to a random set of 25% of all the data (which includes 300 images of the 5 postures from 6 people). An average recognition rate of 76.1% was achieved by using four eigenposes for each class. Experiments run with 4, 5 and 8 eigenposes did not reveal significant changes in performance. Errors were chiefly due to left and right gazings being assigned as front gazes, and vice-versa. High variability in people features due to different face appearance and expressions introduced additional errors.

It is worth stressing that it would be possible to extract additional information from the algorithm. During on-line classification, images with high probabilities assigned simultaneously to two classes along different axis (e.g., left and top) could be labelled as a new class (e.g., top-left).

This is a standard head pose inference algorithm. Since training data for this algorithm was obtained off-line, we do not claim any contribution from this algorithmic implementation. But head gaze pose estimation, together with object and face recognition, is important for further higher-level machine learning algorithms running in a humanoid robot. For instance, after identifying a face, one may try to associate with it probable places where the person often appears, or gaze directions that are associated with certain places (a person sitting in front of a monitor often looks towards the monitor). We will elaborate more on this in the next chapter.

Chapter 6

Visual Pathway – Where

... various hypothesis could explain how things look and feel. You might be sound asleep and dreaming, or a playful brain surgeon might be giving you these experiences... with wires running into your head from a large computer. *(Harman, 1974)*

Autonomous agents, such as robots and humans, are situated in a dynamic world, full of information stored in its own structure. For instance, the probability of a chair being located in front of a table is much bigger than that of being located on the ceiling. But a robot should also place an object where it can easily find it - if one places a book on the fridge, she will hardly find it later!

This dual perspective in object recognition is an important milestone for both children and humanoid robots - not only to be able to infer the presence of objects based on the scene context, but also to be able to determine where objects should be stored based on the probability of finding them in that place later on. This probability is directly related to the functional constraints imposed by the scene – the image context.

A statistical framework was developed to capture knowledge stored in the robot's surrounding world. This framework consists of: 1) grouping together templates of furniture items (acquired applying the segmentation by demonstration method) to form a global geometric description of a scene 2) learning 3D scenes from cues provided

by a human actor and localizing qualitatively (but with coarse measures) the robot on such built scenes; and 3) learning the probable spatial configuration of objects and people within a scene.

Cognitive Issues Spatial localization in the human brain is mainly located in the Hippocampus, while visual spatial localization is concentrated mostly along the "Where Visual Pathway" of the parietal lobe. Of course, building world constructs concerning the location of entities such as objects or people requires the identification of such entities (and hence neural connections to brain areas responsible for such function in humans), which was last chapter's topic.

Developmental Issues In children navigation capabilities evolve according to the epigenetic principle as they start to move around in their physical surroundings, learning about its structure. This occurs mainly during the practicing developmental sub-phase, and towards the re-approximation phase the child gets a completely new view of the world from erect walking. The child maintains an egocentric view of the world during these developmental stages. We will see next that Cog has as well an egocentric perspective of its surrounding world.

6.1 Map Building from Human Contextual Cues

Several techniques have been proposed in the literature for three-dimensional reconstruction of environments, ranging from passive sensing techniques to active sensing using laser range finders, or both (Sequeira, 1996). This thesis will focus on learning topological map representations (Chatila and Laumond, 1985) from cues provided by interactive humans. We try to minimize internal representations. Instead of precise $3D$ shape reconstructions for objects, we are interested in extracting coarse depth measures that creates an holistic 3D representation of a scene.

Our approach for map building relies on human control of contextual features (illustrated previously in figure 4-17). We have shown in chapter 4 how rough depth measures can be extracted by having a human introducing cues (by waving in front of, and close to objects, feeding the system with reference measures). We show here that it is also possible to reconstruct entire scenes, extending this depth estimation process for multiple objects. This is done by grouping the depth points for all objects together in the robot's egocentric coordinates.

An object's location $p = (\theta, \psi)$ in the active vision head's gazing angles (egocentric coordinates), together with the estimated depth and the objects's size and orientation, are grouped together for building scene maps. Each point in the object's template is converted to egocentric coordinates using a motor-retinal map (obtained by locally weighted regression, which is described in chapter 7).

A scene is defined as a collection of objects with an uncertain geometric configuration, each object being within a minimum distance from at least one other object in the scene. Figure 6-1 presents both appearance and coarse depth mosaic images, as well as 3D reconstruction data for a typical scene in the robot's lab. The geometry

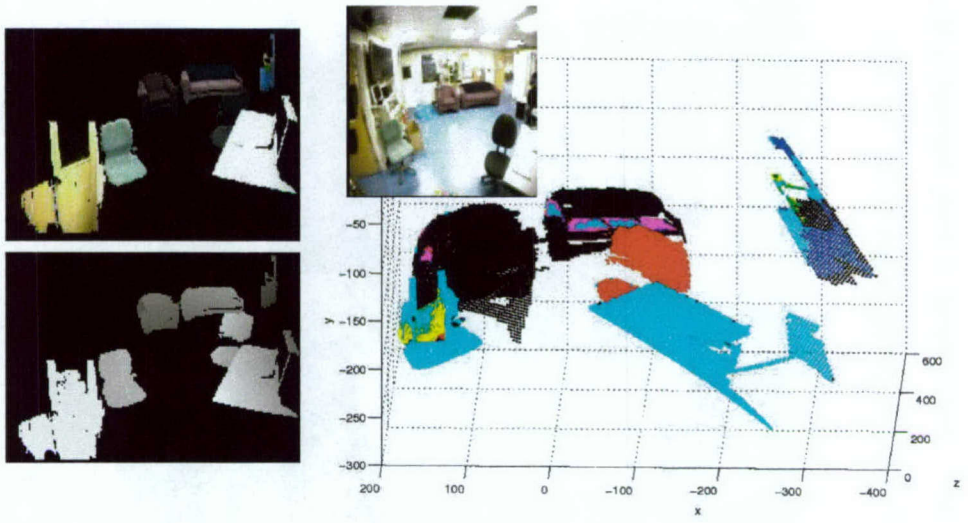


Figure 6-1: (left) Furniture image segmentations – on top – and depth map – bottom – for the scene shown; (right) Coarse 3D map of the same scene. Depth is represented on the axis pointing inside, while the two other axis correspond to egocentric gazing angles (and hence the spherical deformation).

of a scene was reconstructed from the egocentric coordinates of all points lying on the most recent object's template.

Figure 6-2 presents further scene reconstruction results without deformation for a typical scene on Cog's room, while figure 6-3 shows 3D plots for the same scene.

Figure 6-2: (left) Scene on Cog's room, showing stationary objects such as a sofa and a table. A toy car waved by a human is also shown. (center) coarse depth information (lighter corresponds to closer). Depth information on which object is modelled by planes. (right) coarse depth information for the stationary scene (toy car removed).

Scene reconstruction was evaluated from a set of 11 scenes built from human cues, with an average of 4.8 objects per scene (from a set of ten different furniture items). Seven such scenes were reconstructed with no object recognition error, and hence for such cases the scene organization was recovered without structural errors. An average of 0.45 object recognition errors occurred per scene.

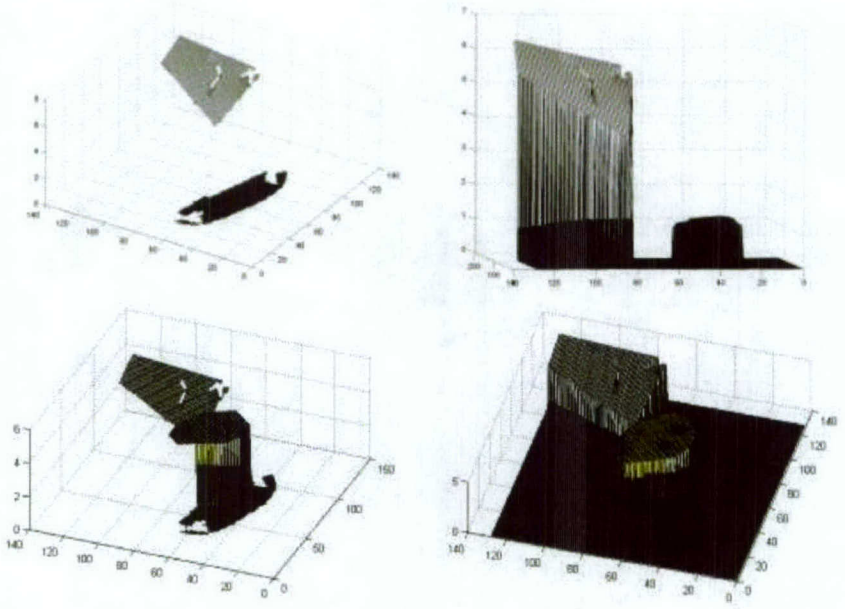


Figure 6-3: (top) Two different views of coarse 3D information (with color values rendered) for a stationary scene. (bottom) Two different views of coarse 3D information for the same stationary scene plus a movable toy car.

6.1.1 Relative Sizes Recovered from the Projective Space.

Consider again figures 6-2 and 6-1. It is easy to notice that the size of the objects on the estimated templates are not proportionally related to their true size. Indeed, in figure 6-2, the table appears about the same size as the sofa, while the car appears only slightly smaller than the other two. This is due to deformations introduced by the object's perspective projection into the retinal plane. But by using the arm diameter as a reference measure, it is possible to re-scale the templates proportionally so that they reflect the true proportions between these objects, as shown in figure 6-4.

6.2 Scene Recognition for Self-Localization

Given the image of an object, its meaning is often a function of the surrounding context. Context cues are useful to remove such ambiguity. Ideally, contextual features should incorporate the functional constraints faced by people, objects or even scenes (eg. people cannot fly and offices have doors). Therefore, functionality plays a more important role than more ambiguous and variable features (such as color, which selection might depend on human preferences). Functionality constraints have been previously exploited for multi-modal association (Arsenio and Fitzpatrick, 2003; Fitzpatrick and Arsenio, 2004) and for determining function from motion (Duric et al., 1995), just to name a few applications.

As such, texture properties seem appropriate as contextual features. Although

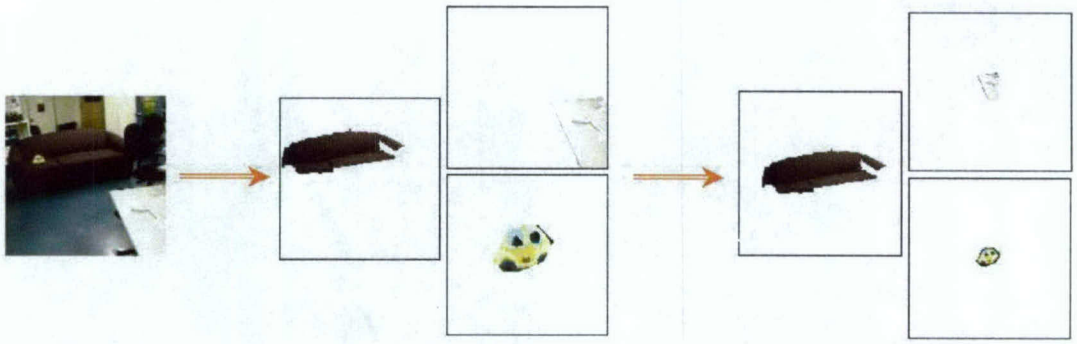


Figure 6-4: (left) Original scene, showing the toy car at the same depth as the sofa, and hence they correct relative size. (center) object templates (right) object templates after re-scaling.

environmental textures are also dependent on human selection, global features such as door placement, desks and shelf location, wall division or furniture geometry usually follow a predetermined pattern which presents low variability. Therefore, in order to incorporate such global constraints, features will be averaged to a low resolution spatial configuration.

Wavelets (Strang and Nguyen, 1996) were selected as contextual features. Processing is applied iteratively through the low frequency branch of the transform over $T = 5$ scales, while higher frequencies along the vertical, horizontal and diagonal orientations are stored (because of signal polarity, this corresponds to a compact representation of six orientations in three images). The input is thus represented by $v(x, y) = v(\vec{p}) = \{v_k(x, y), k = 1, \ldots, N\}$, with N=3T (N=15). Each wavelet component at the i^{th} level has dimensions $256/2^i \times 256/2^i$, and is down-sampled to a 8×8 image:

$$\bar{v}(x, y) = \sum_{i,j} v(i, j)h(i - x, j - y) \tag{6.1}$$

where h(x,y) is a Gaussian window. Thus, $\bar{v}(x, y)$ has dimension 960. Figure 6-5 shows image reconstructions from sets of features $\vec{p}$, which are also called image sketches or holistic representation (Oliva and Torralba, 2001) of a scene. This representation bypasses object identities, since the scene is represented as a single identity (Oliva and Torralba, 2001), holistically. (Oliva and Torralba, 2001) and (Torralba, 2003) apply Windowed Fourier Transforms (similar to STFTs) and Gabor filters, respectively, as contextual features. This manuscript proposes instead wavelets coefficients as contextual information.

Other contextual features can be found in the research literature. The approach presented by (Fu et al., 1994) assumes prior knowledge about regularities of a reduced world where the system is situated. (Moore et al., 1999) assumes as well a prior model, that of a particular fixed scene. The context to recognize objects is given both from this model and by the identification of human motion. In yet another approach presented by (Bobick and Pinhanez, 1995), visual routines are selected

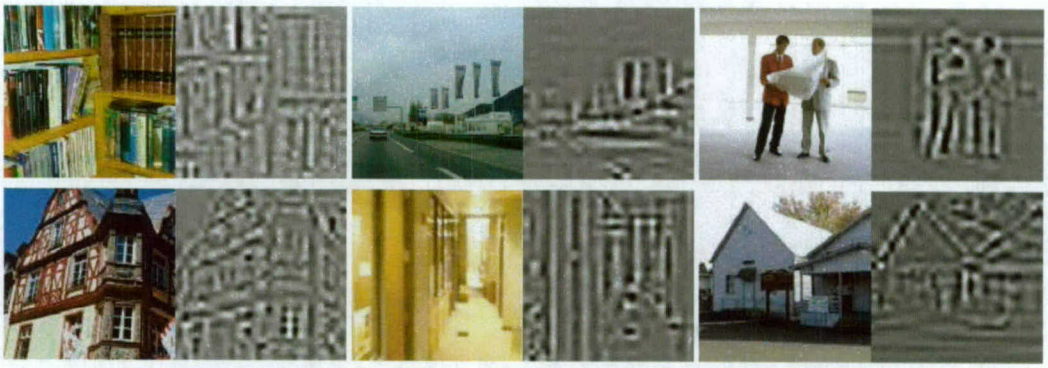


Figure 6-5: Reconstruction of the original image (by the inverse Wavelet transform). As suggested by (Torralba, 2003), this corresponds to an holistic representation of the scene. Instead of building the holistic representation using STFTs (Oliva and Torralba, 2001) or Gabor filters (as in (Torralba, 2003)), this thesis approach applies Wavelets decomposition. Original and reconstruction images are shown in pairs, with the original placed at the left side.

from contextual information. The context consists of a model of hand-written rules of a reduced world in which the vision system operates. Unlike these approaches, our system does not assumes any off-line information or constraint about the real-world scene. Such information is transmitted on-line by a human to the robot, as was described in the previous section.

Similarly to the approach in (Torralba, 2003), the dimensionality problem is reduced to become tractable by applying Principal Component Analysis (PCA). The image features $\bar{v}(\vec{p})$ are decomposed into basis functions provided by the PCA, encoding the main spectral characteristics of a scene with a coarse description of its spatial arrangement:

$$v(\vec{p}) = \sum_{i=1}^{D} c_i \varphi_k^i(\vec{p}) \tag{6.2}$$

where the functions $\varphi_k^i(\vec{p})$ are the eigenfunctions of the covariance operator given by $v_k(\vec{p})$. These functions incorporate both spatial and spectral information. The decomposition coefficients are obtained by projecting the image features $v_k(\vec{p})$ into the principal components:

$$c_i = \sum_{\vec{p},k} v_k(\vec{p}) \varphi_k^i(\vec{p}) \tag{6.3}$$

This is computed using a database of images automatically annotated by the robot. The vector $\vec{c} = \{c_i, i = 1, \ldots, D\}$ denotes the resulting D-dimensional input vector, with $D = E_m, 2 \leq D \leq Th_o$, where m denotes a class, Th_o an upper threshold and E_m denotes the number of eigenvalues within 5% of the maximum eigenvalue. The coefficients c_i are thereafter used as input context features. They can be viewed

as a scene's holistic representation since all the regions of the image contribute to all the coefficients, as objects are not encoded individually.

Mixture models are applied to find interesting places to put a bounded number of local kernels that can model large neighborhoods. In D-dimensions a mixture model is denoted by density factorization over multivariate Gaussians (spherical Gaussians were selected for faster processing times), for each object class n:

$$p(\vec{c}|o_n) = \sum_{m=1}^{M} p((\vec{c}|o_n), g_m) = \sum_{m=1}^{M} b_m p(\vec{c}|o_n, g_m)$$
$$p(\vec{c}|o_n, g_m) = G(\vec{c}, \vec{\mu}_{m,n}, C_{m,n}) = \frac{e^{-1/2(\vec{c}-\vec{\mu}_m)C_m^{-1}(\vec{c}-\vec{\mu}_m)}}{(2\pi)^{D/2}|C_m|^{1/2}}$$

where $|.|^{1/2}$ is the square root of the determinant, g_m refers to the m^{th} Gaussian with mean $\vec{\mu}_m$ and covariance matrix C_m, M is the number of Gaussian clusters, and $b_m = p_(g_m)$ are the weights of the local models. The estimation of the parameters will follow the EM algorithm (Gershenfeld, 1999):

E-step for k-iteration From the observed data $\vec{c}$, this step computes the a-posteriori probabilities $e_{m,n}^k(l)$ of the clusters:

$$e_{m,n}^k(l) = p(c_{m,n}|\vec{c}) = \frac{b_{m,n}^k G(\vec{c}, \vec{\mu}_{m,n}^k, C_{m,n}^k)}{\sum_{m=1}^{M} b_{m,n}^k G(\vec{c}, \vec{\mu}_{m,n}^k, C_{m,n}^k)} \tag{6.4}$$

M-step for k-iteration : cluster parameters are estimated according to the maximization of the joint likelihood of the L training data samples:

$$b_{m,n}^{k+1} = \frac{\sum_{l=1}^{L} e_{m,n}^k(l)}{L} \tag{6.5}$$

$$\mu_{m,n}^{k+1} = <\vec{c}>_m = \frac{\sum_{l=1}^{L} e_{m,n}^k(l)\vec{c}_l}{\sum_{l=1}^{L} e_{m,n}^k(l)} \tag{6.6}$$

$$C_{m,n}^{k+1} = \frac{\sum_{l=1}^{L} e_{m,n}^k(l)(\vec{c}_l - \vec{\mu}_{m,n}^{k+1})(\vec{c}_l - \vec{\mu}_{m,n}^{k+1})^T}{\sum_{l=1}^{L} e_{m,n}^k(l)} \tag{6.7}$$

All vectors are column vectors and $<>_m$ in (6.6) represents the weighted average with respect to the posterior probabilities of cluster m. The EM algorithm converges as soon as the cost gradient is small enough or a maximum number of iterations is reached. The probability density function (PDF) for an object n is then given by Bayes' rule:

$$p(o_n|\vec{c}) = p(\vec{c}|o_n)p(o_n)/p(\vec{c}) \tag{6.8}$$

where $p(\vec{c}) = p(\vec{c}|o_n)p(o_n) + p(\vec{c}|\neg o_n)p(\neg o_n)$. The same method applies for the out-of-class PDF $p(\vec{c}|\neg o_n)$ which represent the statistical feature distribution for the input data in which o_n is not present.

Finally, it is necessary to select the number M of gaussian clusters. This number can be selected as the one that maximizes the joint likelihood of the data. An agglomerative clustering approach based on the Rissanen Minimum Description Length (MDL) order identification criterion (Rissanen, 1983) was implemented to automatically estimate M. In summary:

Given:

- ▷ for al m scene categories learned, it has in memory for each scene l
  - a maximum of $h_l = 800$ wavelet coefficient images (see chapter 3), and h vectors with the components of the PCA applied to this set of images $(\vec{c}^{\{1,\ldots,h_l\},l})$
  - the parameters of the K mixture of gaussians: $e_{\{1,\ldots,K\},l}(\{1,\ldots,h_l\})$, $b_{\{1,\ldots,K\},l}$, $\mu_{\{1,\ldots,K\},l}$ and $C_{\{1,\ldots,K\},l}$.
- ▷ *Training Data:*
  - a batch of n wide-field of view scene images annotated to a scene l_q by the algorithm described in the previous section.
- ▷ *Classification Data:*
  - a query wide-field of view scene image

- TRAINING – Update scene category l with new images of the scene:
  - set $h_{l_q} = minimum(800, h_{l_q} + n)$ and store the additional n wavelet decomposition of scene images
  - apply PCA to all images in the category, and extract the new h_{l_q} coefficient vectors $(\vec{c}^{\{1,\ldots,h_{l_q}\},l_q})$ obtained from the PCA
  - apply EM to train the new mixture of gaussians, initializing the number of mixtures to a large value
  - after convergence, a new K is estimated (Rissanen method) for the number of gaussians, together with a new set of parameters for the mixture of gaussians
- CLASSIFICATION – Recognize query scene image:
  - for $l = 1, \ldots, m$ (for each scene category l in the database of m categories)
    - compute the probability $p(\vec{c}|o_l)$ of the query scene in the mixture of gaussians
  - $best = (-1, MAX_{FLOAT})$
  - for $l = 1, \ldots, m$
    1. compute $p(o_l|\vec{c})$
    2. $best = minimum_{p(o_l|\vec{c})}(best, (l, p(o_l|\vec{c})))$
  - set max_{cat} as the first element of $best$
  - output scene identifier max_{cat}

Figure 6-6 shows results for classifying two different scenes, which were built using the method described in the previous section. Each time a human presents a scene object to the robot, both foveal and wide field of view images are saved and automatically annotated for the corresponding scene.

Contextual cues are not only useful for scene classification, but also for the selection of an object's attentional focus, scale or orientation in an image.

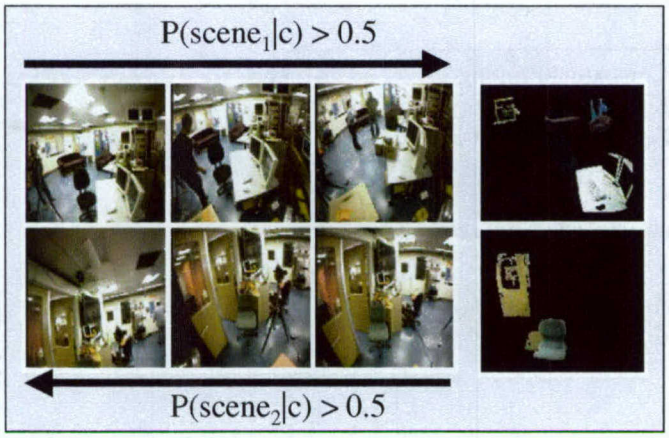


Figure 6-6: Test images (wide field of view) organized with respect to $p(o_n|\vec{c})$. Top row: $o_n = scene_1$, $p(scene_1|\vec{c}) > 0.5$; Bottom row: $o_n = scene_2$, $p(scene_2|\vec{c}) > 0.5$.

6.3 Localizing Others (People and Objects)

Children need to be able not only to built environment descriptions for safely locomoting, but also to learn the relative probability distribution of objects and people in a scene – for instance, books are often found on top of shelves. Hence, the structure of real world scenes is most often constrained by configurational rules similar to those that apply to an object. The scene context puts a very important constraint on the type of places in which a certain object might be found. In addition, contextual information can sometimes be more reliable for object recognition in an image of the real world than local object features.

From a humanoid point of view, contextual selection of the attentional focus is very important both to constrain the search space for locating objects (optimizes computational resources) and also to determine common places on a scene to drop or store objects such as tools or toys. Our approach is thus to enable individual object detection/prediction using the statistics of low-level features in real-world images, conditioned to the presence or absence of objects and their locations, sizes, depth and orientation.

A model for the contextual control of attentional focus (location and orientation), scale selection and depth inference is now presented (the algorithmic structure is illustrated in figure 6-7) which does not neglect dependency among the input state dimensions – see (Torralba, 2003) for scale selection independent from the attentional focus. The output space is defined by the 6-dimensional vector $\vec{x} = (\vec{p}, d, \vec{s}, \phi)$, where $\vec{p}$ is a 2-dimensional position vector, d is the object's depth, $\vec{s} = (w, h)$ is a vector containing the principal components of the ellipse that models the 2D size retinal size of the object, and ϕ is the orientation of such ellipse. Therefore, given the context $\vec{c}$, one needs to evaluate the PDF $p(\vec{x}|o_n, \vec{c})$ from a mixture of (spherical) Gaussians (Gershenfeld, 1999),

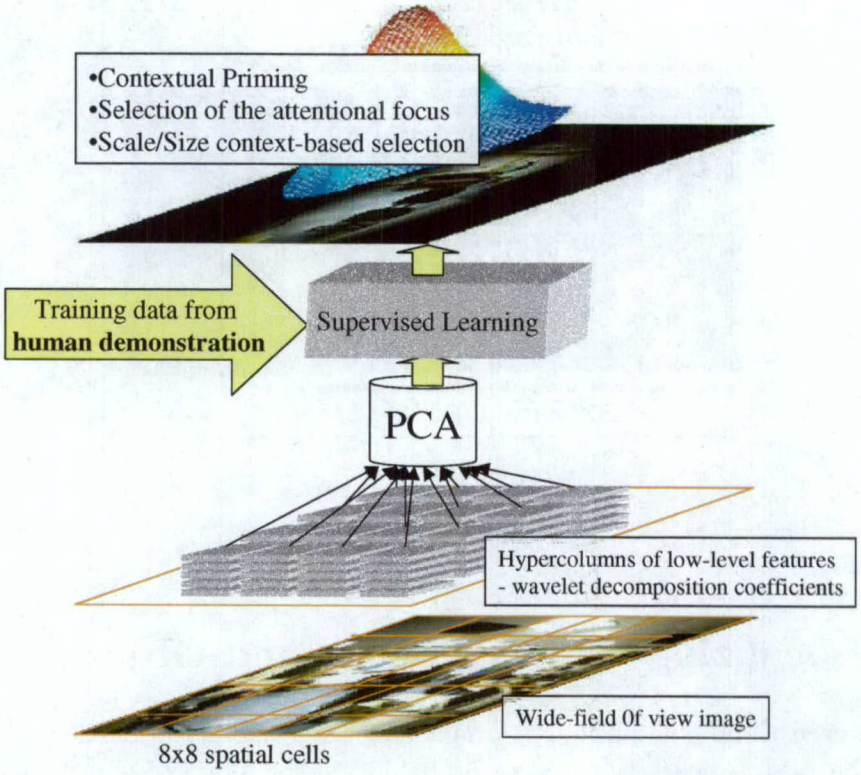


Figure 6-7: Algorithmic structure for learning to locate objects from contextual and human cues.

$$p(\vec{x}, \vec{c}|o_n) = \sum_{m=1}^{M} b_{m,n} G(\vec{x}, \vec{\eta}_{m,n}, X_{m,n}) G(\vec{c}, \vec{\mu}_{m,n}, C_{m,n}) \tag{6.9}$$

The mean of the new Gaussian $G(\vec{x}, \vec{\eta}_{m,n}, X_{m,n})$ is now a function: $\vec{\eta} = f(\vec{c}, \beta_{m,n})$, that depends on $\vec{c}$ and on a set of parameters $\beta_{m,n}$. A locally afine model was chosen for f, with $\{\beta_{m,n} = (\vec{a}_{m,n}, A_{i,n})$: $\vec{\eta}_{m,n} = \vec{a}_{m,n} + A^T \vec{c}\}$.

The learning equations become now (see (Gershenfeld, 1999) for a detailed description of the EM algorithm):

E-step for k-iteration From the observed data $\vec{c}$ and $\vec{x}$, this step computes the a-posteriori probabilities $e^k_{m,n}(l) = p(c_{m,n}|\vec{c}, \vec{x})$ of the clusters:

$$e^k_{m,n}(l) = \frac{b^k_{m,n} G(\vec{x}, \vec{\eta}^k_{m,n}, X^k_{m,n}) G(\vec{c}, \vec{\mu}^k_{m,n}, C^k_{m,n})}{\sum_{m=1}^{M} b^k_{m,n} G(\vec{x}, \vec{\eta}^k_{m,n}, X^k_{m,n}) G(\vec{c}, \vec{\mu}^k_{m,n}, C^k_{m,n})}$$

M-step for k-iteration : cluster parameters are estimated according to (where m

indexes the M clusters, and l indexes the number L of samples):

$$b_{m,n}^{k+1} = \frac{\sum_{l=1}^{L} e_{m,n}^{k}(l)}{L} \tag{6.10}$$

$$\mu_{m,n}^{k+1} = \frac{\sum_{l=1}^{L} e_{m,n}^{k}(l)\vec{c_l}}{\sum_{l=1}^{L} e_{m,n}^{k}(l)} \tag{6.11}$$

$$C_{m,n}^{k+1} =< (\vec{c} - \vec{\mu_{m,n}^{k+1}})(\vec{c} - \vec{\mu_{m,n}^{k+1}})^T >_m \tag{6.12}$$

$$A_{m,n}^{k+1} = (C_{m,n}^{k+1})^{-1} < (\vec{c} - \vec{\mu})(\vec{x} - \vec{\eta})^T >_m \tag{6.13}$$

$$a_{m,n}^{k+1} =< (\vec{x} - (A_{m,n}^{k+1})^T \vec{c}) >_m \tag{6.14}$$

$$X_{m,n}^{k+1} =< (\vec{x} - \vec{a_{m,n}^{k+1}} - (A_{m,n}^{k+1})^T \vec{c})(\vec{x} - \vec{a_{m,n}^{k+1}} - (A_{m,n}^{k+1})^T \vec{c})^T >_m \tag{6.15}$$

The conditional probability follows then from the joint PDF of the presence of an object o_n, at the spatial location p, with pose ϕ, size $\vec{s}$ and depth d, given a set of contextual image measurements $\vec{c}$

$$p(\vec{x}|o_n, \vec{c}) = \frac{\sum_{m=1}^{M} b_{m,n}^{k} G(\vec{x}, \vec{\eta_{m,n}^{k}}, X_{m,n}^{k}) G(\vec{c}, \vec{\mu_{m,n}^{k}}, C_{m,n}^{k})}{\sum_{m=1}^{M} b_{m,n}^{k} G(\vec{c}, \vec{\mu_{m,n}^{k}}, C_{m,n}^{k})}$$

Object detection and recognition requires the evaluation of this PDF at different locations in the parameter space. Measurements $\vec{c} = \vec{v}_{IO}$ are split into two sets

$$v_{IO} = (v_I, v_O) = (v_{B_{\vec{p},\epsilon}}, v_{\bar{B}_{\vec{p},\epsilon}}) \tag{6.16}$$

where $\bar{B}$ is the complement of B. If it is assumed that in the presence of an object, intrinsic object features ($\vec{c} \in B$) and context features ($\vec{c} \in \bar{B}$) are independent, then:

$$P(\vec{c}|\vec{x}, o_n) = p(\vec{v}_I|\vec{x}, o_n) p(\vec{v}_O|\vec{x}, o_n) \tag{6.17}$$

The two conditional PDFs obtained refer to different sets of features (Torralba, 2003):

Local information – $p(\vec{v}_I|\vec{x}, o_n)$ Corresponds to the object/face recognition schemes based on local features previously described in chapter 5. This corresponds to processing along the *what visual pathway*, on the left brain hemisphere, as illustrated in figure 2-1. Assumes $p(o_n|\vec{c}) \simeq p(o_n|\vec{v}_I)$, so that relevant features are inside the neighborhood B – local features which belong to the object and not to the background. The PDF has a high confidence, narrow high maxima for coherent measurements.

Contextual information – $p(\vec{v}_O|\vec{x}, o_n)$ Represents the object conditional PDF given a set of contextual features, which provides priors on the object presence,location, depth, size and orientation. Maps to processing on the right brain hemisphere, (figure 2-1).

Let us neglect therefore $\vec{v}_I$, being the contextual vector $\vec{c}$ given by the principal components of the wavelet decomposition. The effect of neglecting $\vec{v}_I$ is reduced by mapping the foveal camera (which grabs data for the object recognition scheme based on local features) into the image from the wide field of view camera, where the weight of the local features $\vec{v}_I$ is strongly attenuated (see figure 8-3 in the next chapter). The vector $\vec{p}$ is then given in the wide field of view retinal coordinates. The mixture of gaussians is used therefore to learn spatial distributions of objects from the spatial distribution of frequencies on an image.

Results and Discussion

Figure 6-8 presents results for selection of the attentional focus for objects from the low-level cues given by the distribution of frequencies computed by wavelet decomposition. Some furniture objects were not moved (such as the sofas), while others were moved in different degrees: the chair appeared in several positions during the experiment, and thus so did the chair's templates centroids, while the table and door suffered only mild displacements. Still, errors on the head gazing control added considerable location variability whenever a non-movable object (such as the sofa) was segmented and annotated. It demonstrates that, given an holistic characterization of a scene (by PCA on the image wavelet decomposition coefficients), one can estimate the appropriate places whether objects often appear, such as a chair in front of a table, even if no chair is visible at the time – which also informs that regions in front of tables are good candidates to place a chair. Object occlusions by people are not relevant, since local features are neglected, favoring contextual ones.

The same way we wish to be able to determine relations among objects (e.g., chairs are most probable in front of desks), it would also be extremely useful to extract relations among people as well as in between people and objects. For instance, people usually sit on chairs, and therefore might be expected to appear on places on top of chairs or in front of tables, as well as in walking places such as corridors, but not on a ceiling! Furthermore, objects or other people often appear along the gazing direction of a person, and so the feature vector should be extended to account for such relations.

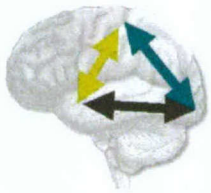
The relations just described do not involve people identity. But such information is useful to refine information about the surrounding world (for instance, Aaron sits at a desk next to Eduardo, or there is a high probability of finding a robotic head at Lijin's desk). All these relations can be put under a statistical framework, and learned using the same strategies applied for objects. Although this thesis developed the necessary structures (such as face recognition, head pose inference and the statistical framework for creating this relational links) to accomplish such tasks, experimental evaluation will have to be delayed for future work.

Figure 6-8: Localizing and recognizing objects from contextual cues (Top) Samples of scene images are shown on the first column. The next four columns show probable locations based on context for finding the door, the smaller sofa, the bigger sofa, the table and the chair, respectively. Notice that, even if the object is not visible or present, the system estimates the places at which there is a high probability of finding such an object. Two such examples are shown for the chair. It also shows that occlusion by humans does not significantly change the context. (Bottom) Results for the scene in Cog's lab for a different day. The training data did not contain any data samples acquired on this day.

Chapter 7

Cross-Modal Data Association

... the auditory system has a complex mechanism that has evolved to convert pressure changes in the air into electrical signals, and the visual system has evolved to convert electromagnetic energy into electrical signals. Although these differences exist, there are many similarities between these two senses. *(Goldstein, 1996)*



We wish to make a sweeping claim: all of the reflective traditions in human history – philosophy, science, psychoanalysis, religion, meditation – have challenged the naive sense of self.

(Varella et al., 1993)

To robots and young infants, the world is a puzzling place, a confusion of sights and sounds. But buried in the noise there are hints of regularity. Some of this is natural; for example, objects tend to go *thud* when they fall over and hit the ground. Some is due to the child; for example, if it shakes its limbs in joy or distress, and one of them happens to pass in front of its face, it will see a fleshy blob moving in a familiar rhythm. And some of the regularity is due to the efforts of a caregiver; consider an infant's mother trying to help her child learn and develop, perhaps by tapping a toy or a part of the child's body (such as its hand) while speaking its name, or making a toy's characteristic sound (such as the *bang-bang* of a hammer).

We seek to extract useful information exploiting such world regularities by applying repeated actions performed either by a caregiver or the robot itself[1]. Observation of infants shows that such actions happen frequently, and from a computational perspective they are ideal learning material since they are easy to identify and offer a wealth of redundancy (important for robustness). The information we seek from repeated actions are the characteristic appearances and sounds of the object, person, or robot involved, with context-dependent information such as the visual background or unrelated sounds stripped away. This allows the robot to generalize its experience beyond its immediate context and, for example, to later recognize the same object used in a different way.

Cognitive Issues Humans receive an enormous quantity of information from the world through their sensorial apparatus. Cues from the visual, auditory and somatosensory senses (as well from tactile and smell senses) are processed simultaneously, and the integration of all such percepts at the brain's cerebral cortex forms our view of the world.

However, humans' sensory modalities are not independent processes. Stimuli from one sensorial modality often influences the perception of stimuli in other modalities. Auditory processing in visual brain areas of early blind subjects suggests that brain areas usually involved in vision play a role in not only auditory selective attention, but also participate in processing changes on the auditory stimulus outside the focus of attention (Alho et al., 1993). Auditory illusions can be created from visual percepts as well – one such instance is the McGurk effect (Cohen and Massaro, 1990).

But audition can also cause illusory visual motion, as described by (Churchland et al., 1994). They report an experiment in which a fixed square and a dot (to its left) are presented to the observer. Without sound stimuli, no motion is perceived for blinking of the dot. Alternate perception of a tone in the left and right ears (left ear tone coinciding with the dot presentation), creates an illusory perception of oscillatory motion of the dot (while the square creates visual occlusions).

Developmental Issues: The development of intermodal perception in infants Infants are not born perceiving the world as an adult does; rather, their perceptual abilities develop over time. This process is of considerable interest to roboticists who seek hints on how to approach adult-level competence through incremental steps.

Vision and audition interact from birth (Wertheimer, 1961). Indeed, a ten-minute-old child turns his eyes toward an auditory signal. In the animal kingdom, studies with young owls have shown that development of sound localization has strong influences from the visual senses. Inducing visual errors from prisms worn over the eyes, owls adjusted their sound localization to match the visual bias (Knudsen and Knudsen, 1985).

Historically, the development of perception in infants has been described using two diametrically opposed classes of theory: integration and differentiation (Bahrick,

[1]This chapter presents collaborative work with Paul Fitzpatrick.

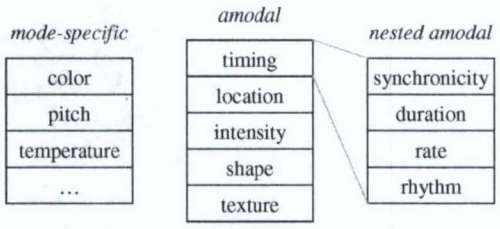


Figure 7-1: Features such as color and pitch are specific to a particular sense (sight and hearing respectively). But not all features are so specific. Several are *amodal* and can manifest themselves across multiple senses. For example, smooth and rough objects can generally be distinguished both by sight and touch. Timing is a particularly productive feature, giving rise to a set of *nested* amodal features.

2003). In a theory of integration, the infant learns to process its individual senses first, and then begins to relate them to each other. In a theory of differentiation, the infant is born with unified senses, which it learns to differentiate between over time. The weight of empirical evidence supports a more nuanced position (as is usually the case with such dichotomies). On the one hand, young infants can detect certain intersensory relationships very early (Lewkowicz and Turkewitz, 1980) – but on the other hand, there is a clear progression over time in the kinds of relations which can be perceived ((Lewkowicz, 2000) gives a timeline).

Time is a very basic property of events that gets encoded across the different senses but is unique to none of them. Consider a bouncing ball – the audible thud of the ball hitting the floor happens at the same time as a dramatic visual change in direction. Although the acoustic and visual aspects of the bounce may be very different in nature and hard to relate to each other, the time at which they make a gross change is comparable. The time of occurrence of an event is an *amodal* property – a property that is more or less independent of the sense with which it is perceived. Other such properties include intensity, shape, texture, and location; these contrast with properties that are relatively modality-specific such as color, pitch, and smell (Lewkowicz, 2003) (see figure 7-1).

Time can manifest itself in many forms, from simple synchronicity to complex rhythms. Lewkowicz proposes that the sensitivity of infants to temporal relationships across the senses develops in a progression of more complex forms, with each new form depending on earlier ones (Lewkowicz, 2000). In particular, Lewkowicz suggests that sensitivity to *synchronicity* comes first, then to *duration*, then to *rate*, then to *rhythm*. Each step relies on the previous one initially. For example, duration is first established as the time between the synchronous beginning and the synchronous end of an event as perceived in multiple senses, and only later does duration break free of its origins to become a temporal relation in its own right that doesn't necessarily require synchronicity.

(Bahrick, 2004) proposes that the perception of the same property across multiple senses (intersensory redundancy) can aid in the initial learning of skills which

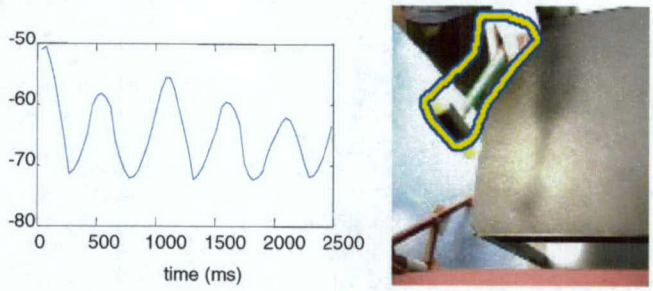


Figure 7-2: When watching a person using a hammer, the robot detects and group points moving in the image with similar periodicity (Arsenio et al., 2003) to find the overall trajectory of the hammer and separate it out from the background. The detected trajectory is shown on the left (for clarity, just the coordinate in the direction of maximum variation is plotted), and the detected object boundary is overlaid on the image on the right.

can then be applied even without that redundancy. Infants exposed to a complex rhythm tapped out by a hammer presented both visually and acoustically can then discriminate that rhythm in either modality alone (Bahrick and Lickliter, 2000) – but if the rhythm is initially presented in just one modality, it cannot be discriminated in either (for infants of a given age). The suggested explanation is that intersensory redundancy helps to direct attention towards amodal properties (in this case, rhythm) and away from mode-specific properties. In general, intersensory redundancy has a significant impact on attention, and can bias figure/ground judgements (Bahrick, 2004).

7.1 Detecting periodic percepts

We exploit repetition – rhythmic motion, repeated sounds – to achieve segmentation and recognition across multiple senses. We are interested in detecting conditions that repeat with some roughly constant rate, where that rate is consistent with what a human can easily produce and perceive. This is not a very well defined range, but we will consider anything above 10Hz to be too fast, and anything below 0.1Hz to be too slow. Repetitive signals in this range are considered to be *events* in our system. For example, waving a flag is an event, clapping is an event, walking is an event, but the vibration of a violin string is not an event (too fast), and neither is the daily rise and fall of the sun (too slow). Such a restriction is related to the idea of natural kinds (Hendriks-Jansen, 1996), where perception is based on the physical dimensions and practical interests of the observer.

To find periodicity in signals, the most obvious approach is to use some version of the Fourier transform. And indeed our experience is that use of the Short-Time Fourier Transform (STFT) demonstrates good performance when applied to the visual trajectory of periodically moving objects (Arsenio et al., 2003). For example, figure 7-2 shows a hammer segmented visually by tracking and grouping periodically

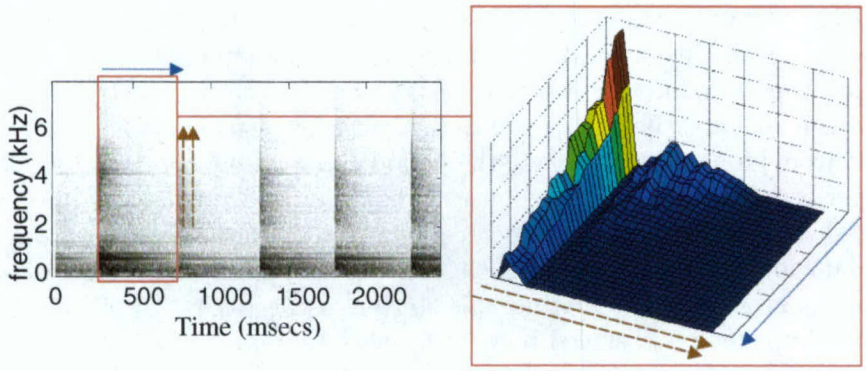


Figure 7-3: Extraction of an acoustic pattern from a periodic sound (a hammer banging). The algorithm for signal segmentation is applied to each normalized frequency band. The box on the right shows one complete segmented period of the signal. Time and frequency axes are labeled with single and double arrows respectively.

moving points. However, our experience from dozens of experiments also leads us to believe that this approach is not ideal for detecting periodicity of *acoustic* signals. And that might be the case even for visual signals, if these signals lack locally a constant period. This is corroborated by findings in (Polana and Nelson, 1997; Seitz and Dyer, 1997). Of course, acoustic signals have a rich structure around and above the kHz range, for which the Fourier transform and related transforms are very useful. But detecting gross repetition around the single Hz range is very different. The sound generated by a moving object can be quite complicated, since any constraints due to inertia or continuity are much weaker than for the physical trajectory of a mass moving through space. In our experiments, we find that acoustic signals may vary considerably in amplitude between repetitions, and that there is significant variability or drift in the length of the periods. These two properties combine to reduce the efficacy of Fourier analysis. This led us to the development of a more robust method for periodicity detection, which is now described. In the following discussion, the term *signal* refers to some sensor reading or derived measurement, as described at the end of this section. The term *period* is used strictly to describe event-scale repetition (in the Hz range), as opposed to acoustic-scale oscillation (in the kHz range).

Period estimation – For every sample of the signal, we determine how long it takes for the signal to return to the same value from the same direction (increasing or decreasing), if it ever does. For this comparison, signal values are quantized adaptively into discrete ranges. Intervals are computed in one pass using a look-up table that, as we scan through the signal, stores the time of the last occurrence of a value/direction pair. The next step is to find the most common interval using a histogram (which requires quantization of interval values), giving us an initial estimate $p_{estimate}$ for the event period. This is essentially the approach presented in (Arsenio and Fitzpatrick, 2003). We extended this method to explicitly take into account the possibility of drift and variability in the period, as follows.

Clustering – The previous procedure gives us an estimate $p_{estimate}$ of the event period. We now cluster samples in rising and falling intervals of the signal, using that estimate to limit the width of our clusters but not to constrain the distance between clusters. This is a good match with real signals we see that are generated from human action, where the periodicity is rarely very precise. Clustering is performed individually for each of the quantized ranges and directions (increasing or decreasing), and then combined afterwards. Starting from the first signal sample not assigned to a cluster, our algorithm runs iteratively until all samples are assigned, creating new clusters as necessary. A signal sample extracted at time t is assigned to a cluster with center c_i if $\| c_i - t \|_2 < p_{estimate}/2$. The cluster center is the average time coordinate of the samples assigned to it, weighted according to their values.

Merging – Clusters from different quantized ranges and directions are merged into a single cluster if $\| c_i - c_j \|_2 < p_{estimate}/2$ where c_i and c_j are the cluster centers.

Segmentation – We find the average interval between neighboring cluster centers for positive and negative derivatives, and break the signal into discrete periods based on these centers. Notice that we do not rely on an assumption of a *constant* period for segmenting the signal into repeating units. The average interval is the final estimate of the signal period.

The output of this entire process is an estimate of the period of the signal, a segmentation of the signal into repeating units, and a confidence value that reflects how periodic the signal really is. This value is given by the percentage of points indexed to $p_{estimate}$ in the histogram computed at the *period estimation* step, relative to the total number of points in such histogram.

The period estimation process is applied at multiple temporal scales. If a strong periodicity is not found at the default time scale, the time window is split in two and the procedure is repeated for each half. This constitutes a flexible compromise between both the time and frequency based views of a signal: a particular movement might not appear periodic when viewed over a long time interval, but may appear as such at a finer scale.

Figure 7-2 shows an example of using periodicity to visually segment a hammer as a human demonstrates the periodic task of hammering, while figure 7-3 shows segmentation of the sound of the hammer in the time-domain. For these examples and all other experiments described in this paper, our system tracks moving pixels in a sequence of images from one of the robot's cameras using a multiple tracking algorithm based on a pyramidal implementation of the Lukas-Kanade algorithm. A microphone array samples the sounds around the robot at 16kHz. The Fourier transform of this signal is taken with a window size of 512 samples and a repetition rate of 31.25Hz. The Fourier coefficients are grouped into a set of frequency bands for the purpose of further analysis, along with the overall energy.

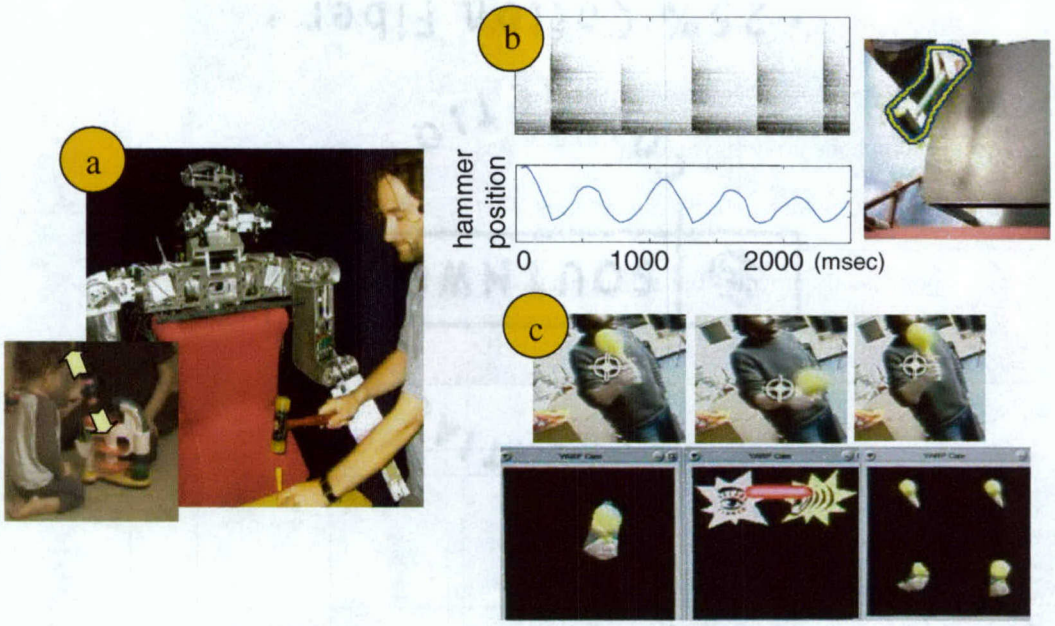


Figure 7-4: Binding vision to sound. a) A child and a human playing with a hammer, banging it on a table, producing a distinctive audio signal. b) A human hammering task, in which the human moves the hammer repetitively forward/backward producing sound in each direction, which is matched to the visual trajectory. The sound energy has one peak per visual period. c) top: tracking an oscillatory instrument – a castonete; down: image of object segmentation and display of a detected visual/sound matching.

7.2 Binding

Due to physical constraints, the set of sounds that can be generated by manipulating an object is often quite small. For toys which are suited to one specific kind of manipulation – as rattles encourage shaking – there is even more structure to the sound they generate (Fitzpatrick and Arsenio, 2004). When sound is produced through motion for such objects the audio signal is highly correlated both with the motion of the object and the tools' identity. Therefore, the spatial trajectory can be applied to extract visual and audio features – patches of pixels, and sound frequency bands – that are associated with the object (see figure 7-4), which enables the robot to map the visual appearance of objects manipulated by humans or itself to the sound they produce.

Figure 7-5 shows how the robot's perceptual state can be summarized – the icons shown here will be used throughout the remainder of this document. The robot can detect periodic events in any of the individual modalities (sight, hearing, proprioception). Any two events that occur in different modalities will be compared, and may be grouped together if there is evidence that they are causally related or *bound*. Such

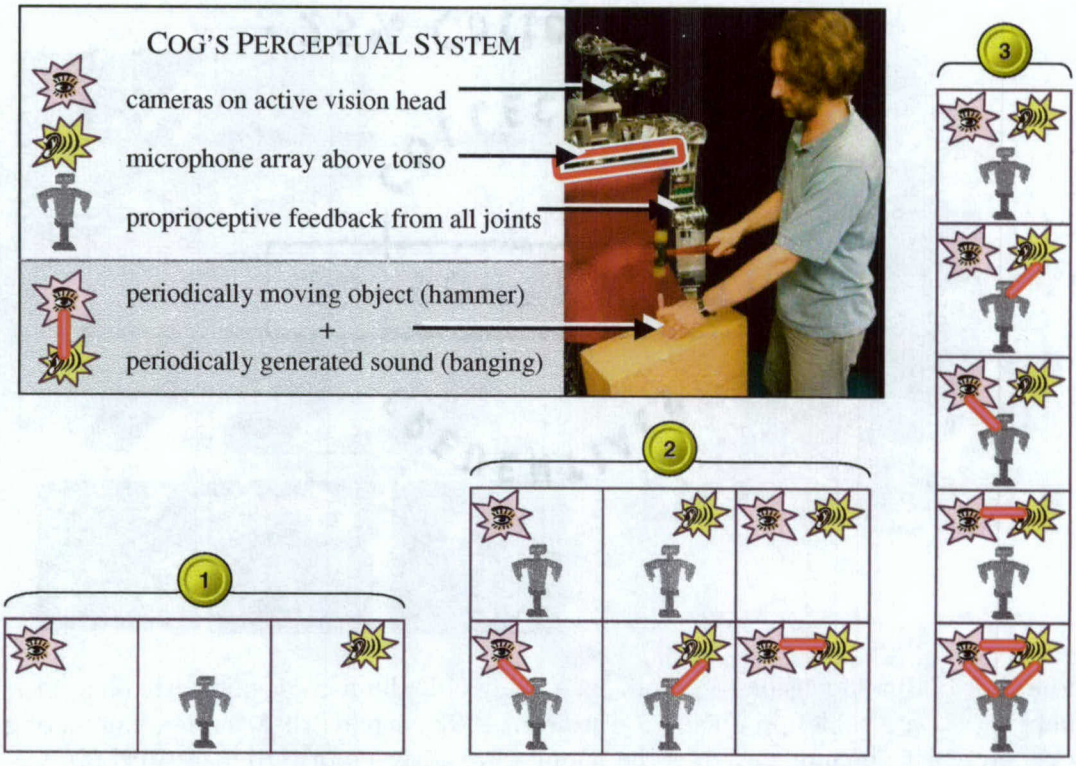


Figure 7-5: A summary of the possible perceptual states of our robot – the representation shown here will be used throughout the remainder of the thesis. Events in any one of the three modalities (sight, proprioception, or hearing) are indicated as in block **1**. When two events occur in different modalities, they may be independent (top of **2**) or bound (bottom of **2**). When events occur in three modalities, the possibilities are as shown in **3**.

relations are transitive: if events A and B are bound to each other, and B and C are bound to each other, then A and C will also be bound. This is important for consistent, unified perception of events. This kind of summarization ignores cases in which there are, for example, multiple visible objects moving periodically making different sounds. We return to this point later in this chapter.

7.2.1 Learning about Objects

Segmented features extracted from visual and acoustic segmentations (using the method presented in the last section) can serve as the basis for an object recognition system. Visual and acoustic cues are both individually important for recognizing objects, and can complement each other when, for example, the robot hears an object that is outside its view, or it sees an object at rest. But when both visual and acoustic cues are present, then we can do even better by looking at the relationship between the visual motion of an object and the sound it generates. Is there a loud bang at an extreme of the physical trajectory? If so we might be looking at a hammer. Are

the bangs at either extreme of the trajectory? Perhaps it is a bell. Such relational features can only be defined and factored into recognition if we can relate or *bind* visual and acoustic signals.

Several theoretical arguments support the idea of binding by temporal oscillatory signal correlations (von der Malsburg, 1995). From a practical perspective, repetitive synchronized events are ideal for learning since they provide large quantities of redundant data across multiple sensor modalities. In addition, as already mentioned, extra information is available in periodic or locally-periodic signals such as the period of the signal, and the phase relationship between signals from different senses – so for recognition purposes the whole is greater than the sum of its parts.

Therefore, a binding algorithm was developed to associate cross-modal, locally periodic signals, by which we mean signals that have locally consistent periodicity, but may experience global drift and variation in that rhythm over time. In our system, the detection of periodic cross-modal signals over an interval of seconds using the method described in the previous section is a necessary (but not sufficient) condition for a binding between these signals to take place. We now describe the extra constraints that must be met for binding to occur.

For concreteness assume that we are comparing a visual and acoustic signal. Signals are compared by matching the cluster centers determined as in the previous section. Each peak within a cluster from the visual signal is associated with a temporally close (within a maximum distance of half a visual period) peak from the acoustic signal, so that the sound peak has a positive phase lag relative to the visual peak. Binding occurs if the visual period matches the acoustic one, or if it matches half the acoustic period, within a tolerance of $60ms$. The reason for the second match is that often sound is generated at the fastest points of an object's trajectory, or the extremes of a trajectory, both of which occur twice for every single period of the trajectory. Typically there will be several redundant matches that lead to binding within a window of the sensor data for which several sound/visual peaks were detected. In (Arsenio and Fitzpatrick, 2003), we describe a more sophisticated binding method that can differentiate causally unconnected signals with periods that are similar just by coincidence, by looking for a drift in the phase between the acoustic and visual signal over time, but such nuances are less important in a benign developmental scenario supported by a caregiver.

Figure 7-6 shows an experiment in which a person shook a tambourine in front of the robot for a while. The robot detected the periodic motion of the tambourine, the rhythmic rise and fall of the jangling bells, and bound the two signals together in real-time.

7.2.2 Learning about People

In this section we do not wish to present any new algorithms, but rather show that the cross-modal binding method we developed for object perception also applies for perceiving people. Humans often use body motion and repetition to reinforce their actions and speech, especially with young infants. If we do the same in our interactions with Cog, then it can use those cues to link visual input with corresponding

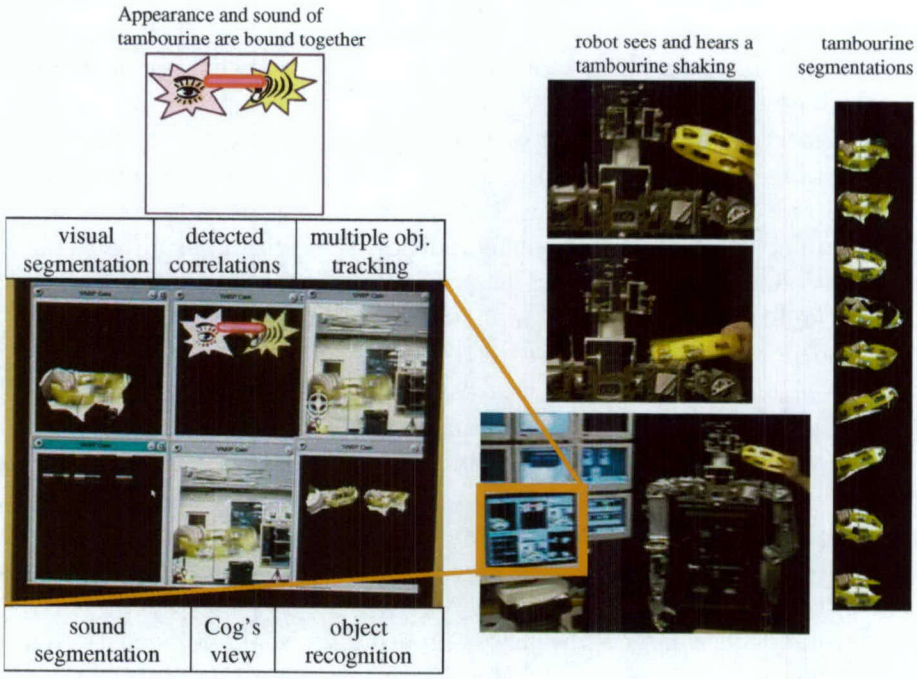


Figure 7-6: Here the robot is shown a tambourine in use. The robot detects that there is a periodically moving visual source, and a periodic sound source, and that the two sources are causally related and should be bound. All images in these figures are taken directly from recordings of real-time interactions, except for the summary box in the top-left (included since in some cases the recordings are of poor quality). The images on the far right show the visual segmentations recorded for the tambourine in the visual modality. The background behind the tambourine, a light wall with doors and windows, is correctly removed. Acoustic segmentations are also generated.

sounds. For example, figure 7-7 shows a person shaking their head while saying "no! no! no!" in time to his head motion. The figure shows that the robot extracts a good segmentation of the shaking head, and links it with the sound signal. Such actions appear to be understood by human infants at around 10-12 months (American Academy Of Pediatrics, 1998).

Sometimes a person's motion causes sound, just as an ordinary object's motion might. Figure 7-8 shows a person jumping up and down in front of Cog. Every time he lands on the floor, there is a loud bang, whose periodicity matches that of the tracked visual motion. We expect that there are many situations like this that the robot can extract information from, despite the fact that those situations were not considered during the design of the binding algorithms. The images in all these figures are taken from online experiments – no off-line processing is done.

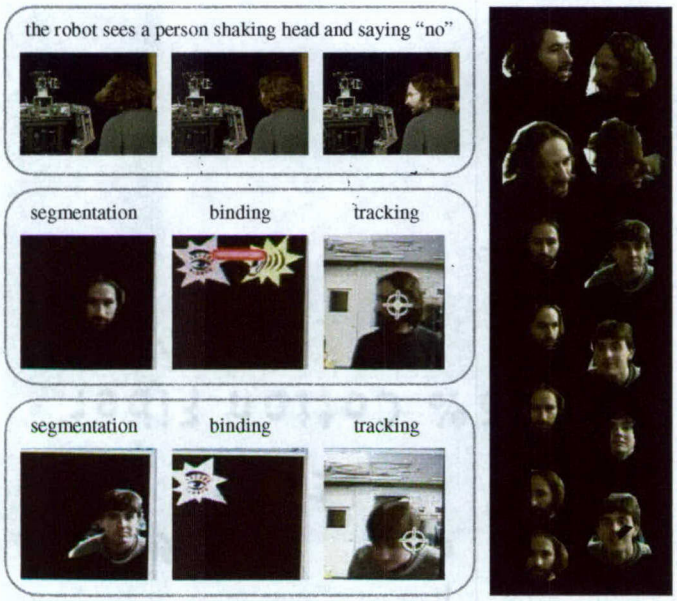


Figure 7-7: In this experiment, the robot sees people shaking their head. In the top row, the person says "no, no, no" in time with his head-shake. The middle row shows the recorded state of the robot during this event – it binds the visually tracked face with the sound spoken. The lower row shows the state during a control experiment, when a person is just nodding and not saying anything. Recorded segmentations for these experiments are shown on the right.

7.3 Cross-Modal Integration with Proprioception

Proprioception from the robot's joints is a sensorial modality very important to control the mechanical device, as well as to provide workspace information (such as the robot's gaze direction). But proprioceptive data is also very useful to infer identity about the robotic *self* (Fitzpatrick and Arsenio, 2004) (for instance, by having the robot recognize itself on a mirror). Children become able to self-recognize their image on a mirror during Mahler's developmental practicing sub-phase, which marks an important developmental step towards the child individuality. On a humanoid robot, large correlations of a particular robot's limb with data from other sensorial inputs indicates a link between such sensing modality to that moving body part (which generated a sound, or which corresponds to a given visual template, as shown in figure 7-9).

7.3.1 Self Recognition

So far we have considered only external events that do not involve the robot. In this section we turn to the robot's perception of its own body. Cog treats proprioceptive feedback from its joints as just another sensory modality in which periodic events may occur. These events can be bound to the visual appearance of its moving body

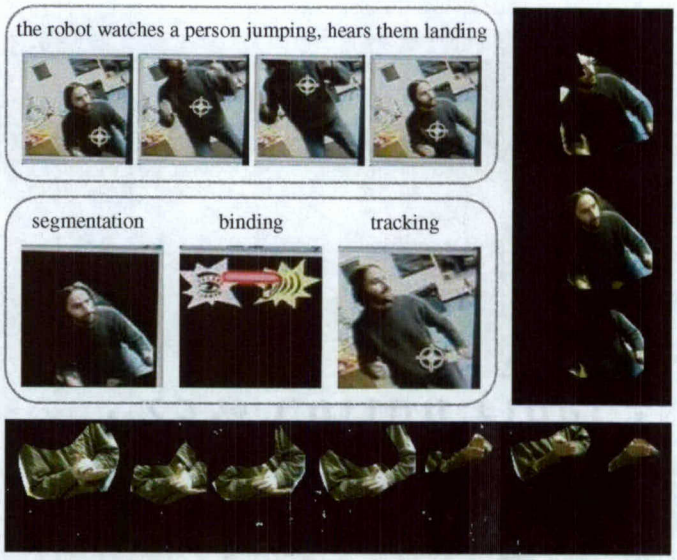


Figure 7-8: This figure shows the result of one human actor jumping up and down like crazy in front of the robot. The thud as he hit the floor was correctly bound with segmentations of his body (column on right). The bottom row shows segmentations from a similarly successful experiment where another human actor started applauding the robot.

part – assuming it is visible – and the sound that the part makes, if any (in fact Cog's arms are quite noisy, making an audible "whirr-whirr" when they move back and forth).

Figure 7-10 shows a basic binding experiment, in which a person moved Cog's arm while it is out of the robot's view. The sound of the arm and the robot's proprioceptive sense of the arm moving are bound together. This is an important step, since in the busy lab Cog inhabits, people walk into view all the time, and there are frequent loud noises from the neighboring machine shop. So cross-modal rhythm is an important cue for filtering out extraneous noise and events of lesser interest.

In figures 7-12 and 7-12, the situation is similar, with a person moving the robot's arm, but the robot is now looking at the arm. In this case we see our first example of a binding that spans three modalities: sight, hearing, and proprioception. The same is true in figure 7-13, where Cog shakes its own arm while watching it in a mirror.

An important milestone in child development is reached when the child recognizes itself as an individual, and identifies its mirror image as belonging to itself (Rochat and Striano, 2002). Self-recognition in a mirror is also the focus of extensive study in biology. Work on self-recognition in mirrors for chimpanzees (Gallup et al., 2002) suggests that animals other than humans can also achieve such competency, although the interpretation of such results requires care and remains controversial. Self-recognition is related to the notion of a theory-of-mind, where intents are assigned to other actors, perhaps by mapping them onto oneself, a topic of great interest in robotics (Kozima and Yano, 2001; Scassellati, 2001). Proprioceptive feedback provides very useful refer-

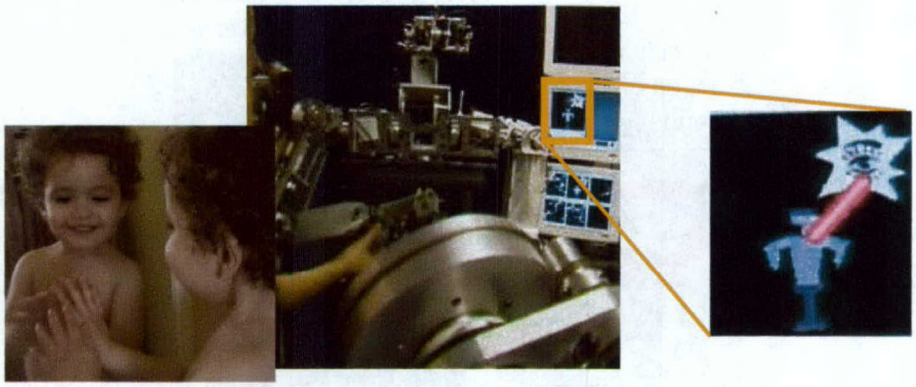


Figure 7-9: Child and robot looking at a mirror, associating their image to their body (image of robot/visual image association shown amplified for the robot).

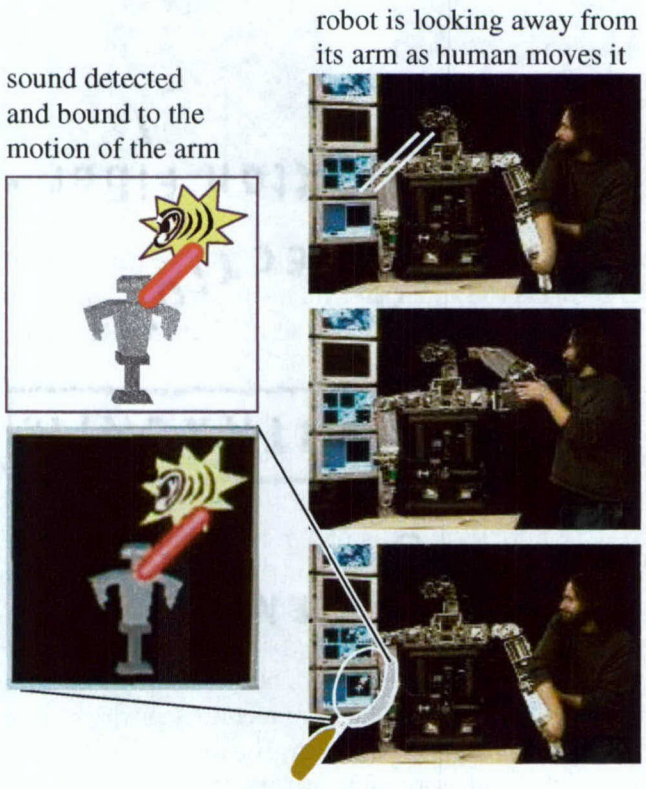


Figure 7-10: In this experiment, a person grabs Cog's arm and shakes it back and forth while the robot is looking away. The sound of the arm is detected, and found to be causally related to the proprioceptive feedback from the moving joints, and so the robot's internal sense of its arm moving is bound to the external sound of that motion.

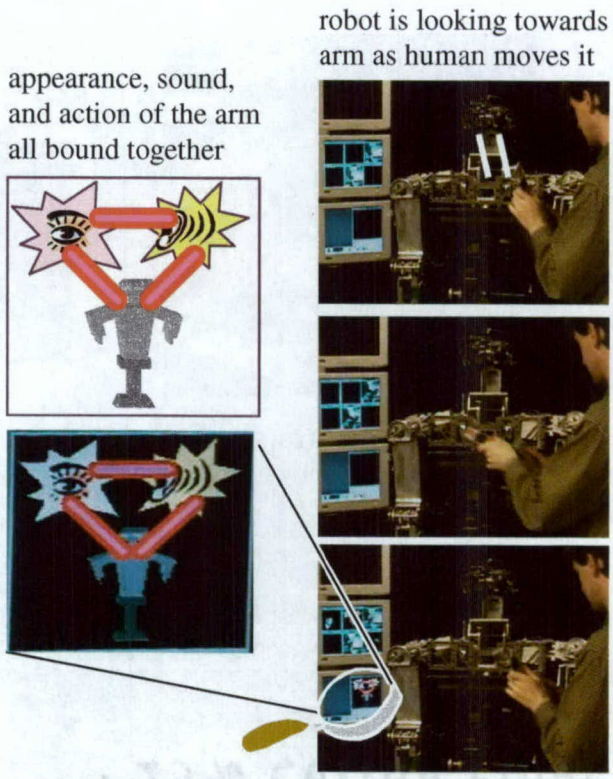


Figure 7-11: In this experiment, a person shakes Cog's arm in front of its face. What the robot hears and sees has the same rhythm as its own motion, so the robot's internal sense of its arm moving is bound to the sound of that motion and the appearance of the arm.

ence signals to identify appearances of the robot's body in different modalities. That is why we extended our binding algorithm to include proprioceptive data.

Children between 12 and 18 months of age become interested in and attracted to their reflection (American Academy Of Pediatrics, 1998). Such behavior requires the integration of visual cues from the mirror with proprioceptive cues from the child's body. As shown in figure 7-14, the binding algorithm was used not only to identify the robot's own acoustic rhythms, but also to identify visually the robot's mirror image (an important milestone in the development of a child's theory of mind (Baron-Cohen, 1995)). It is important to stress that we are dealing with the low-level *perceptual* challenges of a theory of mind approach, rather than the high-level *inferences* and mappings involved. Correlations of the kind we are making available could form a grounding for a theory of mind and body-mapping, but are not of themselves part of a theory of mind – for example, they are completely unrelated to the intent of the robot or the people around it, and intent is key to understanding others in terms of the self (Kozima and Zlatev, 2000; Kozima and Yano, 2001). Our hope is that the perceptual and cognitive research will ultimately merge and give a truly intentional

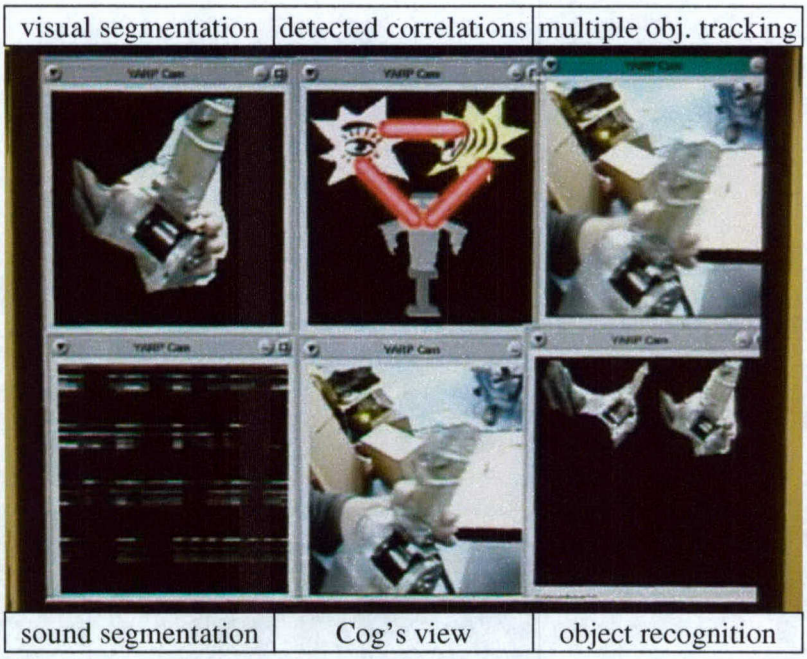


Figure 7-12: Real-time view of the robot's status during the experiment in figure 7-11. The robot is continually collecting visual and auditory segmentations, and checking for cross-model events. It also compares the current view with its database and performs object recognition to correlate with past experience (bottom right).

robot that understands others in terms of its own goals and body image – an image which could develop incrementally using cross-modal correlations of the kind explored in this paper.

7.4 Priming Other Senses for Attention

Human studies have shown that attention in one of the senses can be modified by input from the other senses. For example, (Bahrick, 2004) describes an experiment in which two movies of actions such as clapping hands are overlaid, and the sound corresponding to just one of the movies is played. Adult and infant attention is found to be directed to the matching action. In adults, there is a large reported difference between what is perceived when the sound is off (ghostly figures moving through each other) and when the sound is on (a strong sense of figure and background).

Matching with visual distraction

Lets consider the case of multiple objects moving in the robot's visual field, only one of which is generating sound. The robot uses the sound it hears to filter out uncorrelated moving objects and determine a candidate for cross-modal binding. This is a form

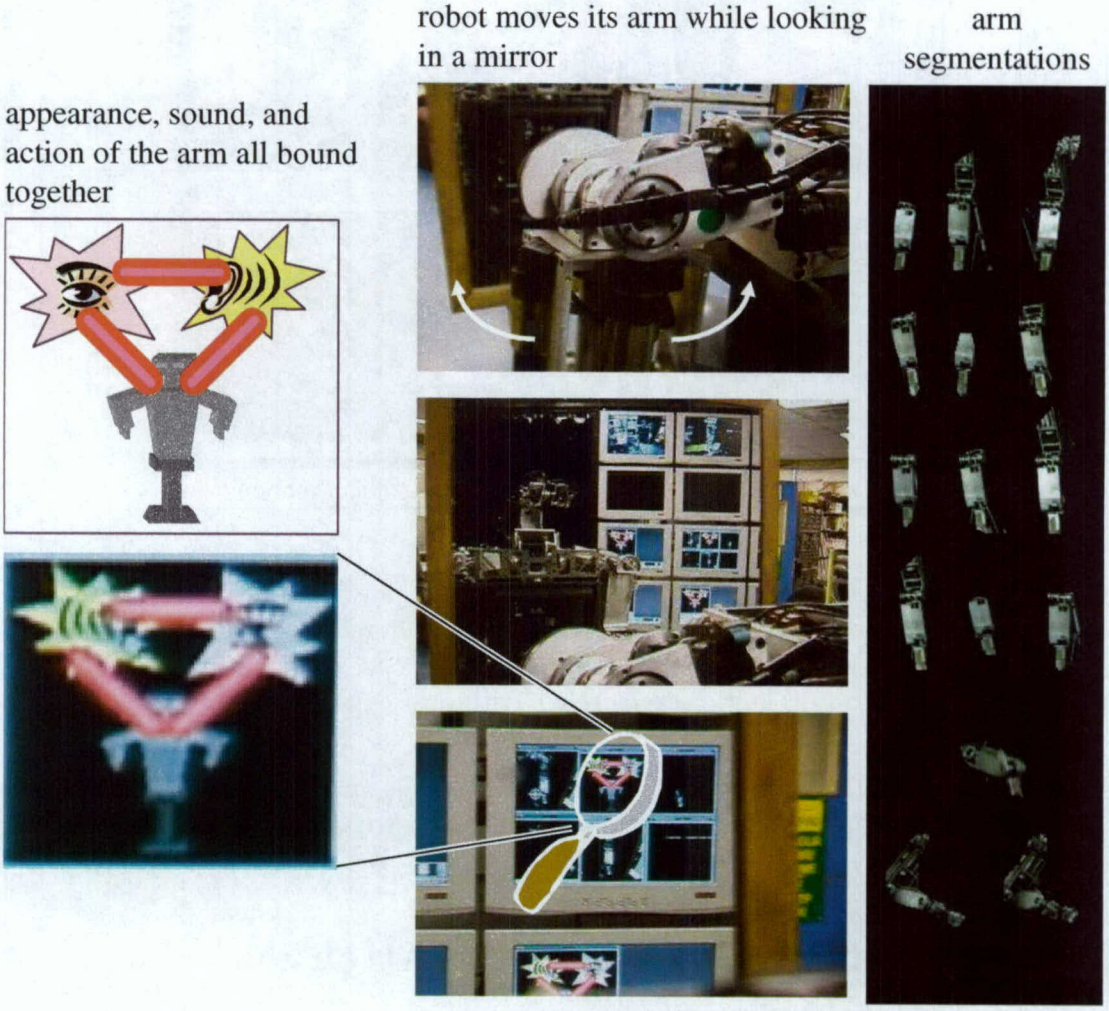


Figure 7-13: In this experiment, Cog is looking at itself in a mirror, while shaking its arm back and forth (the views on the center column are taken by a camera behind the robot's left shoulder, looking out with the robot towards the mirror). The reflected image of its arm is bound to the robot's sense of its own motion, and the sound of the motion. This binding is identical in kind to the binding that occurs if the robot sees and hears its own arm moving directly without a mirror. However, the appearance of the arm (right column) is from a quite different perspective than Cog's own view of its arm.

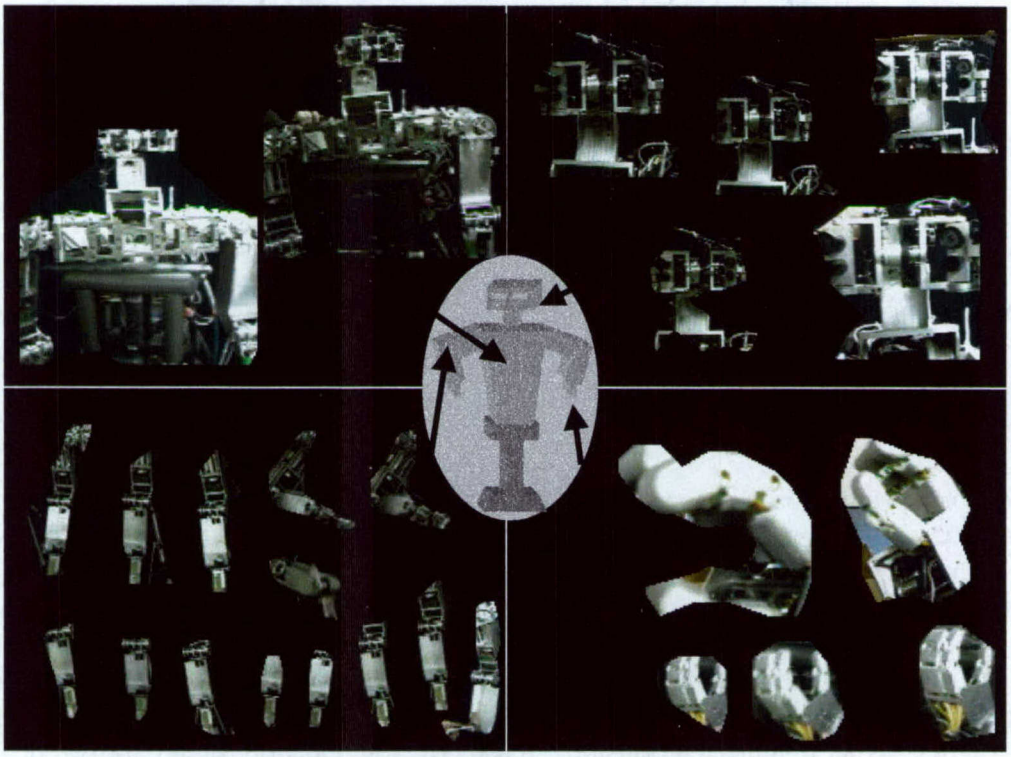


Figure 7-14: Results for mapping visual appearances of self to ones' own body. Cog can be shown different parts of its body simply by letting it see that part (in a mirror if necessary) and then shaking it, such as its (right) hand or (left) flipper. Notice that this works for the head, even though shaking the head also affects the cameras. The reason why this happens lies on the visual image stabilization by the vestibular ocular reflex in Cog's eye control – as will be described in chapter 9 – which reduces considerably background's motion relative to the head motion. In addition, the walls behind Cog's place (reflected as background in the mirror image) lack texture – contrary to the walls in front of it, which are highly cluttered– which might have also facilitated the figure/ground segregation task.

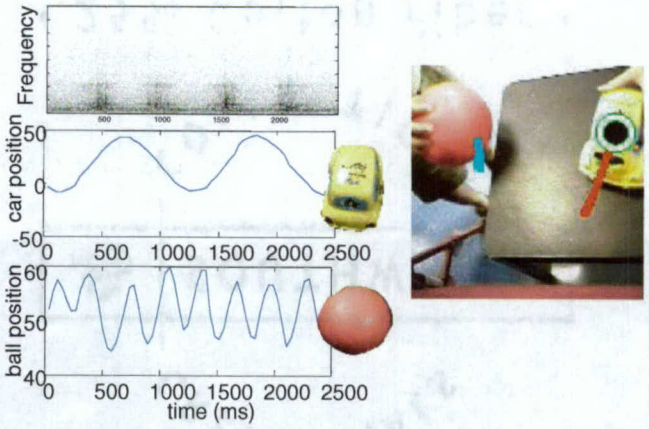


Figure 7-15: The top left image shows a car and a ball moving simultaneously, with their trajectories overlaid. The spectrogram during this event is shown to the right. Sound is only generated by the rolling car – the ball is silent. A circle is placed on the object (car) with which the sound is bound. The sound energy and the visual displacements of the objects are given.

of context priming, in which an external signal (the sound) directs attention towards one of a set of potential candidates.

Figure 7-15 shows measurements taken during an experiment with two objects moving visually, at different rates, with one - a toy car - generating a rolling sound, while the other - a ball - is moving silently. The acoustic signal is linked with the object that generated it (the car) using period matching. The movement of the ball is unrelated to the period of the sound, and so that object is rejected. In contrast, for the car there is a very definite relationship. In fact, the sound energy signal has two peaks per period of motion, since the sound of rolling is loudest during the two moments of high velocity motion between turning points in the car's trajectory. This is a common property of sounds generated by mechanical rubbing, so the binding algorithm takes this possibility into account by testing for the occurrence of frequencies at double the expected value.

Matching with acoustic distraction

Lets now consider the case of one object moving in the robot's field of view, and one 'off-stage', with both generating sound. Matching the right sound to the visible object is achieved by mapping the time history of each individual coefficient band of the audio spectrogram (see figure 7-16) to the visual trajectory of the object. We segment the sound of the object from the background by clustering the frequency bands with the same period (or half the period) as the visual target, and assign those bands to the object.

Within the framework being described, visual information is used to prune the range of frequency bands of the original sound - the coefficient bands of the audio

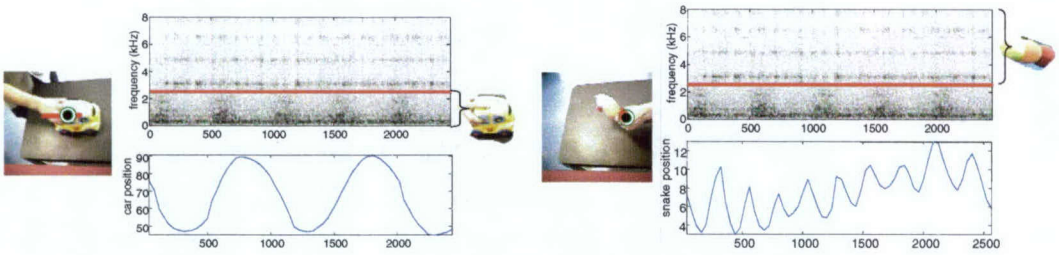


Figure 7-16: The two spectrogram/trajectory pairs shown are for a shaking toy car and snake rattle. The top pair occurs with only the car visible, and the lower pair occurs with only the snake visible. The line in each spectrogram represents the cutoff pitch frequency between the car and snake.

visual are segmented into clusters of bands that characterize the sound of an object. For the experiment shown in figure 7-16-left, the coefficients ranging below 2.6Hz are assigned to the object. Afterwards, a band-pass filter is applied to the audio-signal to filter out the other frequencies, resulting in the clear sound of the car with the sound of the rattle removed or highly attenuated. For the experiment shown in the right part of figure 7-16 the roles of the car and snake were switched. A low-pass filter with cut-off frequency at 2.6-2.8Hz is applied to the audio-signal to filter out the frequencies corresponding to the car, resulting in the snakes' sound.

Matching multiple sources

This experiment considers two objects moving in the robot's field of view and both generating sound, as presented in figure 7-17. Each frequency band is mapped to one of the visual trajectories if coherent with its periodicity. For each object, the lower and the higher coefficient band are labelled as the lower and higher cut-off frequencies, respectively, of a band-pass filter assigned to that object. The complex sound of both the moving car-toy and the cube-rattle are thus segmented into the characteristic sound of the car and sound of the rattle through band-pass filtering. Multiple bindings are thus created for multiple oscillating objects producing distinct sounds.

It is worth stressing that the real world is full of objects making all kinds of noise. However, the system is robust to such disturbances. During the experiments presented throughout this manuscript, people were speaking occasionally while interacting with the robot, while other people were making everyday sounds while working. If the distracting sound occurs at the same range of frequencies as the sound of the oscillating object, then a binding might just not occur for that specific time, but occur after a few seconds when the interference noise switches to other frequencies or disappears. Table 7.1 shows how well the methods described for binding sounds with objects work on a series of experiments.

Experiment	visual period found	sound period found	bind sound, vision	candidate binds	correct binds	incorrect binds
hammer	8	8	8	8	8	0
car and ball	14	6	6	15	5	1
plane & mouse/remote control	18	3	3	20	3	0
car (snake in backg'd)	5	1	1	20	1	0
snake (car in backg'd)	8	6	6	8	6	0
car & cube *car*	9	3	3	11	3	0
car & cube *cube*	10	8	8	11	8	0
car & snake *car*	8	0	0	8	0	0
car & snake *snake*	8	5	5	8	5	0

Table 7.1: Evaluation for four binding cases of cross-modal rhythms of increasing complexity. The simplest is when a single object (the hammer) is in view, engaged in a repetitive motion and a single repetitive sound source is also heard. This corresponds to a run of roughly 1 minute, for which binding is easy as shown by the data. The next case is when multiple moving objects are visible, but only one repeating sound is heard. Two experiments were made – a car and a ball visible and only the car generating sound, and a plane and other objects visible but only the plane generating sound. Since an object's sound is strongly affected by environment noise, highest confidence is required for this modality, which reduces the number of periodic detections, and consequently the number of bindings. The third case corresponds to two repeating sounds with different periods, and a single visible moving object (experiments for car with snake rattle in background and vice-versa). The car generates mainly low frequency sounds, but the rattle generates high frequency sounds with some weak low frequency components that cause interference with the detection of the car's sound. This is the reason for a weak percentage of bindings for the car. Finally, multiple sound and visual source can be bound together appropriately (two experiments: car and cube rattle; and car and snake rattle). Bindings occur more often for objects producing sounds with high frequency energies. Table data represents number of samples (e.g., 8 correct binds were obtained for the hammer, with no incorrect one).

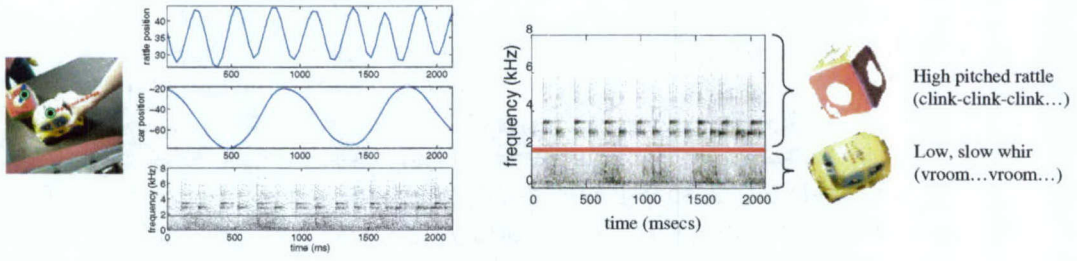


Figure 7-17: The car and the cube, both moving, both making noise. The line overlaid on the spectrogram (bottom) shows the cutoff determined automatically between the high-pitched bell in the cube and the low-patched rolling sound of the car. A spectrogram of the car alone can be seen in figure 7-15. The frequencies of both visual signals are half those of the audio signals.

7.5 Cross-Modal Object Segmentation/Recognition

Different objects have distinct acoustic-visual patterns which are a rich source of information for object recognition, if we can recover them. The relationship between object motion and the sound generated varies in an object-specific way. A hammer causes sound after striking an object. A toy truck causes sound while moving rapidly with wheels spinning; it is quiet when changing direction. A bell typically causes sound at either extreme of motion. All these statements are truly cross-modal in nature, and we explore here using such properties for recognition. Features extracted from the visual and acoustic segmentations are what is needed to build an object recognition system. Each type of feature is important for recognition when the other is absent. But when both visual and acoustic cues are present, then we can do even better by looking at the relationship between the visual motion of an object and the sound it generates. Is there a loud bang at an extreme of the physical trajectory? If so we might be looking at a hammer. Are the bangs soft relative to the visual trajectory? Perhaps it is a bell. Such relational features can only be defined and factored into recognition if we can relate or *bind* visual and acoustic signals. Therefore, the feature space for recognition consists of:

- ▷ Sound/Visual period ratios – the sound energy of a hammer peaks once per visual period, while the sound energy of a car peaks twice (for forward and backward movement).
- ▷ Visual/Sound peak energy ratios – the hammer upon impact creates high peaks of sound energy relative to the amplitude of the visual trajectory. Although such measure depends on the distance of the object to the robot, the energy of both acoustic and visual trajectory signals will generally decrease with depth (the sound energy disperses through the air and the visual trajectory reduces in apparent scale).

Human actions are therefore used to create associations along different sensor modalities, and objects can be recognized from the characteristics of such associa-

tions. Our approach can differentiate objects from both their visual and acoustic backgrounds by finding pixels and frequency bands (respectively) that are oscillating together. This is accomplished through dynamic programming, applied to match the sound energy to the visual trajectory signal. Formally, let $S = (S_1, \ldots, S_n)$ and $V = (V_1, \ldots, V_m)$ be sequences of sound and visual trajectory energies segmented from n and m periods of the sound and visual trajectory signals, respectively. Due to noise, n may be different to m. If the estimated sound period is half the visual one, then V corresponds to energies segmented with $2m$ half periods (given by the distance between maximum and minimum peaks). A matching path $P = (P_1, \ldots, P_l)$ defines an alignment between S and M, where $max(m, n) \leq l \leq m + n - 1$, and $P_k = (i, j)$, a match k between sound cluster j and visual cluster i. The matching constraints are imposed by:

The boundary conditions are $P_1 = (1, 1)$ and $P_l = (m, n)$.

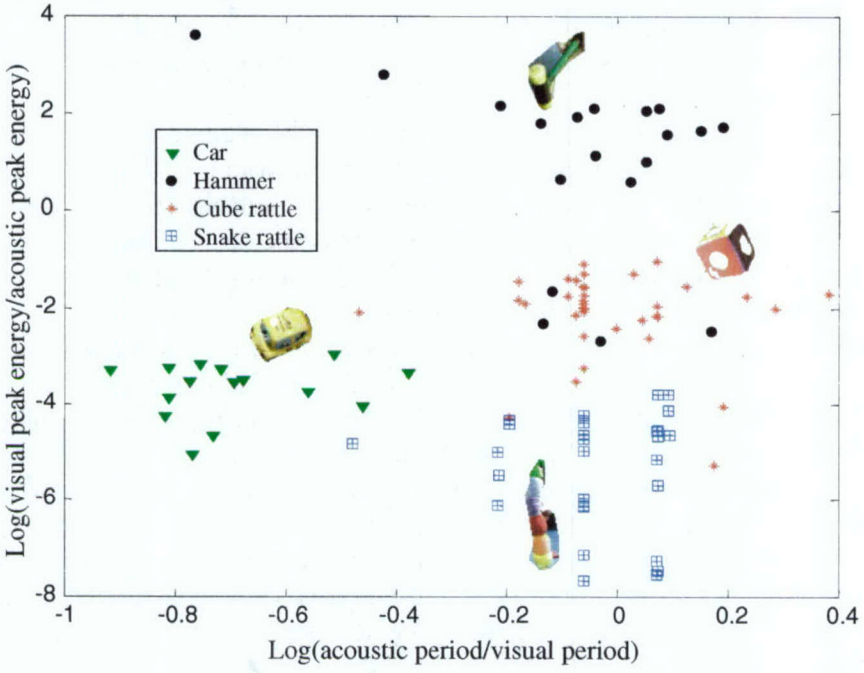
Temporal continuity satisfies $P_{k+1} \in \{(i + 1, j + 1), (i + 1, j), (i, j + 1)\}$. This restricts steps to adjacent elements of P.

The function cost $c_{i,j}$ is given by the square difference between V_i and S_j periods. The best matching path W can be found efficiently using dynamic programming, by incrementally building an $m \times n$ table caching the optimum cost at each table cell, together with the link corresponding to that optimum. The binding W will then result by tracing back through these links, as in the Viterbi algorithm.

Figure 7-18 shows cross-modal features for a set of four objects. It would be hard to cluster automatically such data into groups for classification. But as in the sound recognition algorithm, training data is automatically annotated by visual recognition and tracking. The classification scheme consists on applying a mixture of gaussians to model the distribution of the training data. The mixture parameters are learned by application of the iterative EM algorithm (Gershenfeld, 1999). After training, objects can be categorized from cross-modal cues alone. The system was evaluated quantitatively by selecting randomly 10% of the data for validation, and the remaining data for training. This process was randomly repeated fifteen times. The recognition rate averaged over all these runs were, by object category: 100% for both the car and the snake rattle, 86.7% for the cube rattle, and 83% for the hammer. The overall recognition rate was 92.1%. Such results demonstrate the potential for recognition using cross-modal cues.

7.6 Discussion

Most of us have had the experience of feeling a tool become an extension of ourselves as we use it (see (Stoytchev, 2003) for a literature review). Many of us have played with mirror-based games that distort or invert our view of our own arm, and found that we stop thinking of our own arm and quickly adopt the new distorted arm as our own. About the only form of distortion that can break this sense of ownership is a delay between our movement and the proxy-arm's movement. Such experiences



Confusion matrix	Car	Cube	Snake	Hammer
Car	**30**	0	0	0
Cube	0	**52**	7	1
Snake	0	0	**45**	0
Hammer	0	5	0	**25**

Figure 7-18: Object recognition from cross-modal clues. The feature space consists of period and peak energy ratios. The confusion matrix for a four-class recognition experiment is also shown. The period ratio is enough to separate well the cluster of the car object from all the others. Similarly, the snake rattle is very distinct, since it requires large visual trajectories for producing soft sounds. Errors for categorizing a hammer originated exclusively from erroneous matches with the cube rattle, because hammering is characterized by high energy ratios, and very soft bangs are hard to identify correctly. The cube rattle generates higher energy ratios than the snake rattle. False cube rattle recognitions resulted mostly from samples with low energy ratios being mistaken for the snake rattle.

argue for a sense of self that is very robust to every kind of transformation except latencies. Our work is an effort to build a perceptual system which, from the ground up, focuses on timing just as much as content. This is powerful because timing is truly cross-modal, and leaves its mark on all the robot's senses, no matter how they are processed and transformed.

We wish our system to be scalable, so that it can correlate and integrate multiple sensor modalities (currently sight, sound, and proprioception). To that end, we detect and cluster periodic signals within their individual modalities, and only then look for cross-modal relationships between such signals. This avoids a combinatorial explosion of comparisons, and means our system can be gracefully extended to deal with new sensor modalities in future (touch, smell, etc.).

Evidence from human perception strongly suggests that timing information can transfer between the senses in profound ways. For example, experiments show that if a short fragment of white noise is recorded and played repeatedly, a listener will be able to hear its periodicity. But as the fragment is made longer, at some point this ability is lost. But the repetition can be heard for far longer fragments if a light is flashed in synchrony with it (Bashford et al., 1993) – flashing the light actually changes how the noise sounds. More generally, there is evidence that the cues used to detect periodicity can be quite subtle and adaptive (Kaernbach, 1993), suggesting there is a lot of potential for progress in replicating this ability beyond the ideas already described.

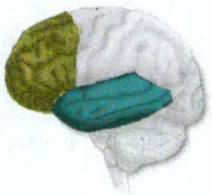
A lot about the world could be communicated to a humanoid robot through human demonstration. The robot's learning process will be facilitated by sending it repetitive information through this communication channel. If more than one communication channel is available, such as the visual and auditory channels, both sources of information can be correlated for extracting richer pieces of information. We demonstrated in this chapter a specific way to take advantage of correlating multiple perceptual channels at an early stage, rather than just by analyzing them separately - the whole is truly greater than the sum of the parts.

Chapter 8

Memory and Auditory Processing

[Observable in children] He that attentively considers the state of a child, at his first coming into the world, will have little reason to think him stored with plenty of ideas, that are to be the matter of his future knowledge. It is by degrees he comes to be furnished with them. And though the ideas of obvious and familiar qualities imprint themselves before the memory begins to keep a register of time or order, yet it is often so late before some unusual qualities come in the way, that there are few men that cannot recollect the beginning of their acquaintance with them.

(Locke, 1690)



Memory gives the ability to store and retrieve previously experienced stimuli when the latter is no longer present. This chapter deals with the problem of memorizing object locations, while chapter 5 dealt with a dual problem – memorizing identities.

Following a similar approach to the problem of recognizing faces, this chapter also presents processing of acoustic percepts, namely the segmentation of acoustic patterns and their a posteriori recognition.

Cognitive Issues Perceptions have lasting effects on the visual system. Indeed, the effects of a visual experience are not over as soon as the visual environment changes.

Visual memory is now considered standardly divided into three differentiated systems: Iconic Memory, Short Term Memory (STM - closely related to the so called working memory), and Long Term Memory (LTM). The Hippocampus plays a very important role in the human brain (Kandel et al., 1992) for storing and retrieving memories. Having complementary properties to vision perception, auditory processing for segmenting and recognizing sounds is mainly localized in the temporal lobe (together with visual object recognition and other categorization processes), having extensive connections to the brain's visual association area.

Developmental Issues Visual memory is a crucial aspect of learning, and children who have deficits in their visual memory skills have difficulties in reproducing a sequence of visual stimuli. In addition, keeping track of percepts creates links between temporally related data, which is what is need to generate training data for recognizing faces and objects, as introduced in chapter 5, and sounds.

Cross-modal developmental differentiation (Bahrick, 2003) enables an infant, starting from cross-modal, unified processing of the senses, to learn skills which can then be applied individually. Therefore, segmented percepts can be generated by cross-modal processing from which a recognizer for a perceptual modality, such as sound, can be trained. Following the integration approach, segmented percepts can also be segmented by processing the modality alone (such as extracting sound patterns from just acoustic periodicity).

Indeed, both approaches will be exploited hereafter to segment sound patterns, which are incrementally inserted into a sound recognition scheme: sound models of objects are first learned from experimental human/robot manipulation, enabling their a-posteriori identification with or without the agents actuation.

8.1 Auditory Perception: Acoustic Segmentation and Recognition

The repetitive nature of the sound generated by an object under periodic motion can be analyzed to extract an acoustic 'signature' for that object. We search for repetition in a set of frequency bands independently, then collect those frequency bands whose energies oscillate together with a similar period. Specifically, the acoustic signature for an object is obtained by applying the following steps:

1. The period of repetition for each frequency band is detected using the procedure developed in chapter 7.

2. A *period histogram* is constructed to accumulate votes for frequency bands having the same estimated period (or half the period – it is common to have sounds that occur once per repetition, for example at one endpoint of the trajectory, or twice per repetition, for example at two instants of maximum velocity). The histogram is smoothened by adding votes for each bin of the histogram to their immediate neighbors as well.

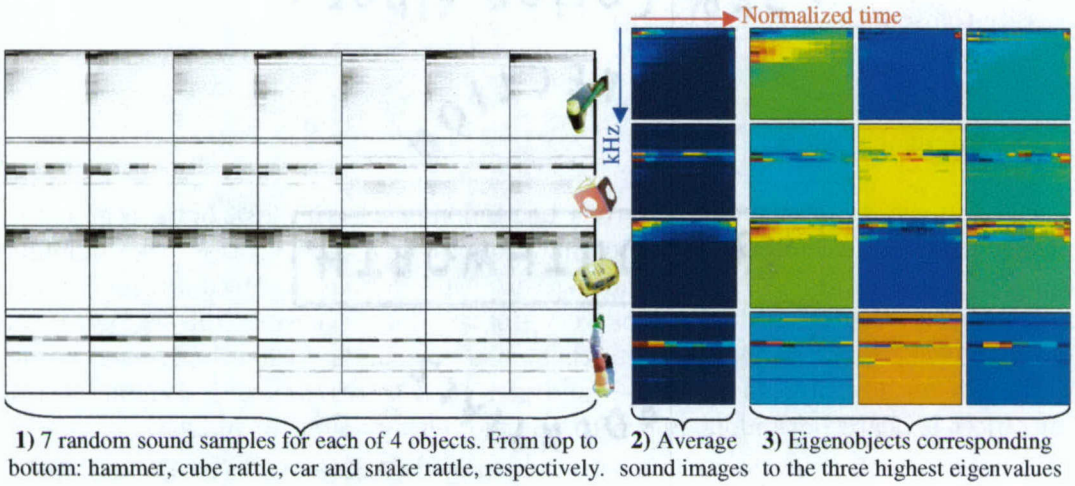


Figure 8-1: Sound segmentation and recognition. Acoustic signatures for four objects are shown along the rows. (1) Seven sound segmentation samples are shown for each object, from a total of 28 (car), 49 (cube rattle), 23 (snake rattle) and 34 (hammer) samples. (2) The average acoustic signature for each object is shown. The vertical axis corresponds to the frequency bands and the horizontal axis to time normalized by the period. (3) The eigensounds corresponding to the three highest eigenvalues are shown.

3. The maximum entry in the period histogram is selected as the *reference* period. All frequency bands corresponding to this maximum are collected and their responses over the reference period are stored in a database of acoustic signatures. Since the same objects can be shaken or waved at different velocities resulting in varying periodicity, it is important to normalize temporal information relative to the reference period.

A collection of annotated acoustic signatures for each object are used as input data (see figure 8-1) for a sound recognition algorithm by applying the eigenobjects method, which, as described in chapter 5, is also widely used for face recognition (Turk and Pentland, 1991). This method is a modified version of Principal Component Analysis. A sound image is represented as a linear combination of base sound signatures (or *eigensounds*). Only eigensounds corresponding to the three highest eigenvalues – which represent a large portion of the sound's energy – are retained. Classification consists of projecting novel sounds onto this space, determining the coefficients of this projection, computing the L_2 distance to each object's coefficients in the database, and selecting the class corresponding to the minimum distance.

Cross-modal information aids the acquisition and learning of unimodal percepts and consequent categorization in a child's early infancy. Similarly, visual data is employed here to guide the annotation of auditory data to implement a sound recognition algorithm. Training samples for the sound recognition algorithm are classified into different categories by the visual object recognition system or from information from the visual object tracking system. This enables the system, after training, to

classify the sounds of objects that are no longer visible.

Experimental Results

The system was evaluated quantitatively be randomly selecting 10% of the segmented data for validation, and the remaining data for training. This process was randomly repeated three times. It is worth noting that even samples received within a short time of each other often do not look very similar, due to noise in the segmentation process, background acoustic noise, other objects' sounds during experiments, and variability on how objects are moved and presented to the robot. For example, the car object is heard both alone and with a rattle (either visible or hidden).

The recognition rate for the three runs averaged to 82% (86.7%, 80% and 80%). Recognition rates by object category were: 67% for the car, 91.7% for the cube rattle, 77.8% for the snake rattle and 83.3% for the hammer. Most errors arise from mismatches between car and hammer sounds. Such errors could be avoided by extending our sound recognition method to use derived features such as the onset/decay rate of a sound, which is clearly distinct for the car and the hammer (the latter generates sounds with abrupt rises of energy and exponential decays, while sound energy from the toy car is much smoother). Instead, we have shown in the previous chapter that these differences can be captured by cross-modal features to correctly classify these objects.

8.2 Short-Term Memory

Keeping information in memory concerning objects with which the robot interacts can simplify other problems, such as for extracting segmentations from multiple object views for a posteriori face or object recognition, as demonstrated in chapter 5. This motivated an algorithm implementation for visually tracking multiple objects and faces, keeping them in a short-memory, limited size buffer (on the order of minutes). Objects and faces detected and segmented from the robot's surrounding world are inserted into memory, and tracked until they disappear of the visual field or they become stationary over a large time interval.

Good features to track (Shi and Tomasi, 1994) - strong corners of an object – are tracked using a Lucas-Kanade tracker. An entity externally identified corresponds either to a new object/face or to an object/face already being tracked. To make a decision, the centroid and variance in mass for each entity is tracked, and the algorithm maps the centroid of the new template to the gaussian given by such parameters. The weighted distance corresponds to a mahalanobis distance. A strong value for such distance indicates a high probability of matching, while weaker probabilities for all tracked objects suggests a new object. Figure 8-2 shows results for tracking. The algorithm runs and was tested on multiple targets on the visual field.

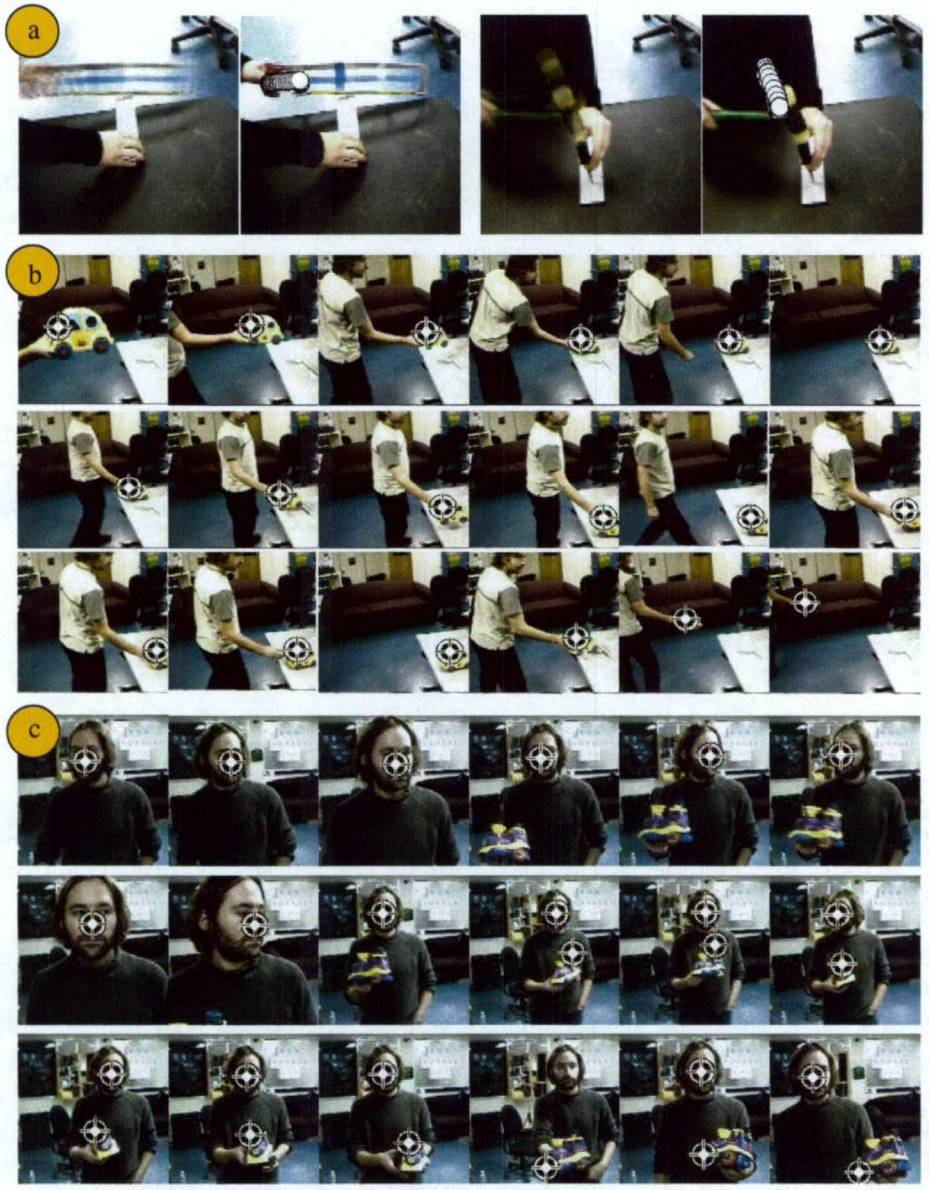


Figure 8-2: Multiple objects tracking in a short term memory. a) Tracking a hammer and a saw b) Tracking a toy car over a 2 minute period c) Tracking both a face and an object, for a sequence corresponding to the run in figure 5-8. The 5^{th} image from the left on the bottom row shows a recovery after losing track of a face on the 4^{th} image.

8.3 Long-Term Memory

Low-level attentive mechanisms and visual tracking maintain information in memory constrained by target visibility. Therefore, a statistical method for storing lasting information concerning the location of objects in the world was developed, which applies the same statistical framework as the visual-based holistic method presented in chapter 6. But in contrast to that, long-term memories will be vision-independent (indeed, humans are able to remember the location of objects without visual input).

The "context" of an object in a scene can be defined in terms of other previously recognized objects within the same scene. This context representation is object-centered since it requires object recognition as a first step. Training data consists of the data stored and automatically annotated while building scene descriptions from human cues. A scene is modelled as a collection of links, being each scene's object connected to any other object in the same scene. Only one object in a scene (which will hereafter be called a "gate") connects to all other objects in another scene, avoiding complexity explosion by imposing an upper bound on the number of links. Hence, each time an object is detected from local information or from human demonstration, the algorithm creates or updates connecting links.

Long term memory is stored in two different ways:

Probable Object/Face egocentric location This corresponds to determining

$$p(\vec{x_a}|o_a) = \sum_{m=1}^{M} b_m G(x_a, \mu_m, C_m) \tag{8.1}$$

from all previous object's a locations, with $c = x_a$ and $o = o_a$. This spatial distribution is modelled by a mixture of gaussians for each object, and trained using the expectation-maximization algorithm.

Inter-objects/faces correlations The presence of one object in a scene often determines the spatial configuration of nearby objects or faces by imposing functional constrains. For instance, chairs appear most often near tables, while toy objects are more common on top of the latter. Humans regularly use objects as cognitive enhancers – known as cognitive artifacts – that help them to retrieve their memories. Indeed, it is often hard for a scientist to find a research paper in his office *jungle*. But these same papers are easy to access in adequate folders after the office gets organized. Therefore, research papers are highly probable inside folders, and especially if those are marked with research labels. The same reasoning applies to people – one expects to find shoes on a student's feet at a research committee meeting.

The problem consists of estimating an object's vector $\vec{x}$ using other objects in a scene as contextual information. Each link from object a to object b, given x_a, is defined by the probability of finding object a at state x_a and object b with state $\vec{x}_b = (p_b, d_b, \vec{s}_b, \phi_b)$. In such an approach, the contextual feature vector is $\vec{c} = x_a$, $o_n = o_a$ and $\vec{x} = \vec{x}_b$. The vector p is the object's location in the robot's

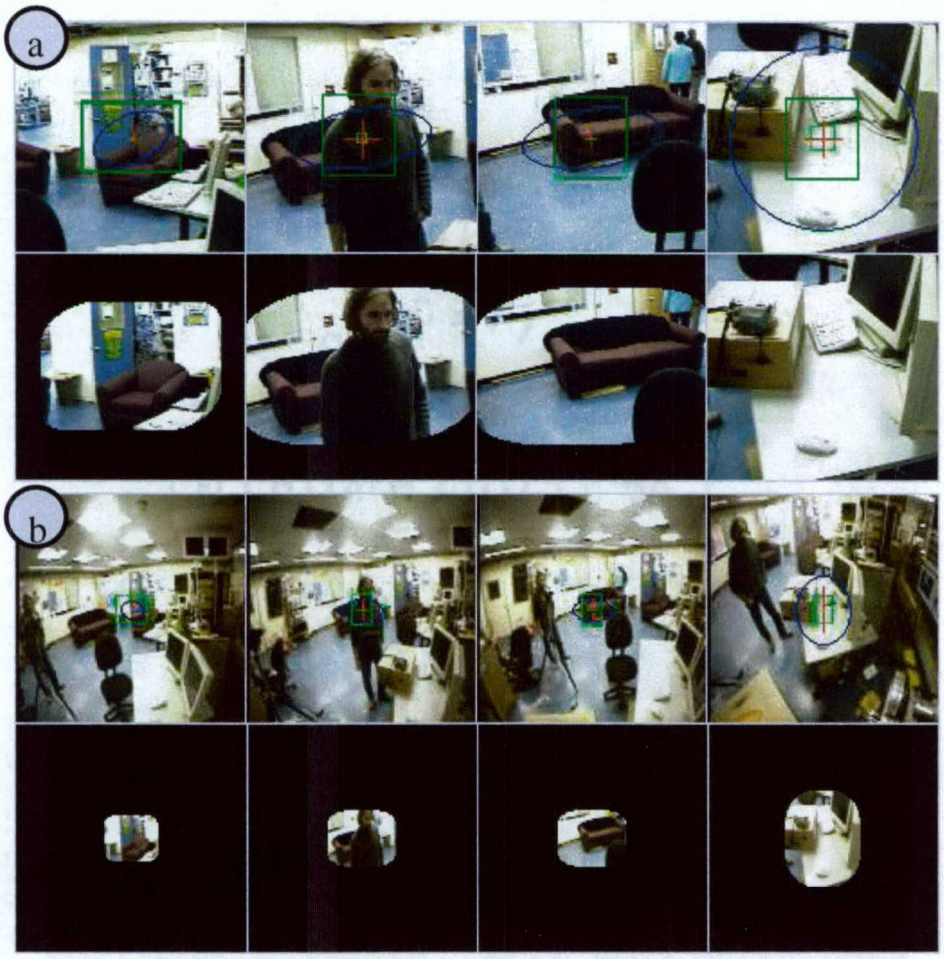


Figure 8-3: (a) Priming object locations. Sample regions (foveal coordinates) computed from predicted size (ellipse) and location (rectangle), plus uncertainties, for objects in a scene. (b) Mapping views. All the data concerning object properties, such as retinal size and location, is mapped onto wide field of view image coordinates, to train the contextual priming system described in chapter 6.

head gazing coordinates (the head joints-gazings mapping is estimated using a supervised learning technique which will be described in the next chapter). A cluster-weighted modelling technique (technique also used for learning locations from holistic features) is applied to learn the mixture of gaussians that model the spatial distributions of objects and people relative to the spatial layout of a scene.

Figure 8-3 presents results for probable object/face egocentric locations: the robotic head verges towards the predicted egocentric coordinates of several furniture objects (which corresponds to the image centers). Location uncertainty (squares) and predicted object size are shown in foveal retinal coordinates, and, through a mapping, in wide field of view coordinates as well.

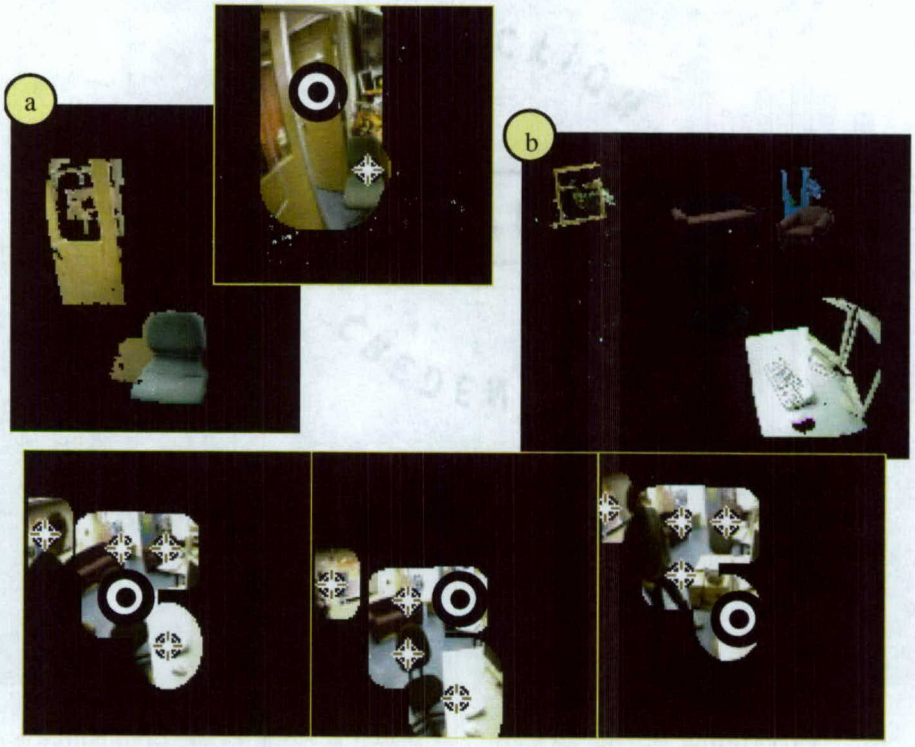


Figure 8-4: Predicting objects from other scene objects. a) Scene composed by two objects: a door and a chair (image appearance mosaic is shown in the left). Shows location predicted for the chair having found a door. The size of all objects, with associated uncertainty, is used to mask the original image. The large circles denote the reference object from which the system determines probable locations for other objects (smaller stars) from the statistical analysis of previously memorized locations b) Identical results for another scene (image appearance mosaic is shown in the right). It shows the prediction of the location of furniture items in a scene from a chair, a small sofa and a table (from the left).

Figure 8-4 shows results for selection of the attentional focus for objects given the state data $\vec{x}$ of another object. It is worth stressing that context priming prunes the set of candidate objects to match the primed object, and therefore reduces the computational resources required for object detection, which is only applied into the more likely spatial locations. Figure 8-5 presents further results for simultaneous selection of scale and attentional focus.

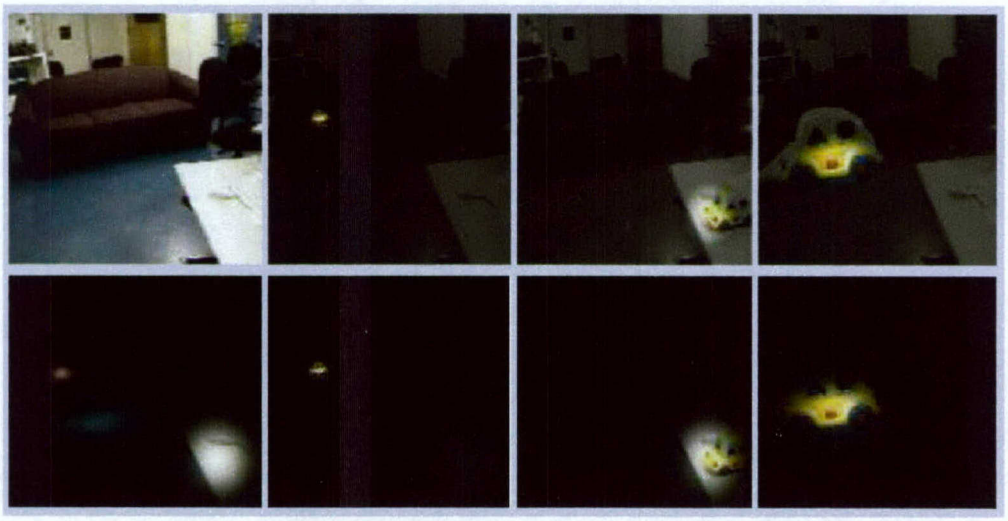


Figure 8-5: The left column shows regions of the image with high probability of finding the car (any size). The next three columns show the regions of higher probability to find the car at a given size scale (smaller, medium and large scales, counting from the left).

Chapter 9

Sensory-Motor Area and Cerebellum

Whereas Descartes projected sensory messages to the walls of the ventricle and Willis brought them to the thalamus, Swedenborg thought they terminated in the cerebral cortex, "the seat wherein sensation finally ceases,"... *(Gross, 1998)*

Sensory-motor integration and control occurs developmentally during childhood, as the child learns from simple perceptual stimulus to organize its neural control structure. Drawing on the teaching parallel between robots and children, a similar approach can guide the acquisition of control capabilities on a humanoid robot. An old version of the humanoid robot Cog (Brooks et al., 1998), is shown in figure 9-1 sawing a piece of wood using neural oscillators to control the rhythmic movements (Williamson, 1999). According to (Williamson, 1999), the robot did not know how to grab the saw or the underlying task sequence. The neural oscillator parameters needed to be inserted off-line, using a time-expensive trial-and-error approach. A new mathematical framework for the automatic selection of the neural oscillators' parameters was proposed in (Arsenio, 2000a,b,c, 2004c). But it remains necessary to recognize the object - a saw, identify it with the corresponding action, and learn the sequence of events and objects that characterize the task of sawing (described in chapter 10). Furthermore, a general strategy should apply to any object the robot interacts with, and for any task executed, such as hammering or painting.

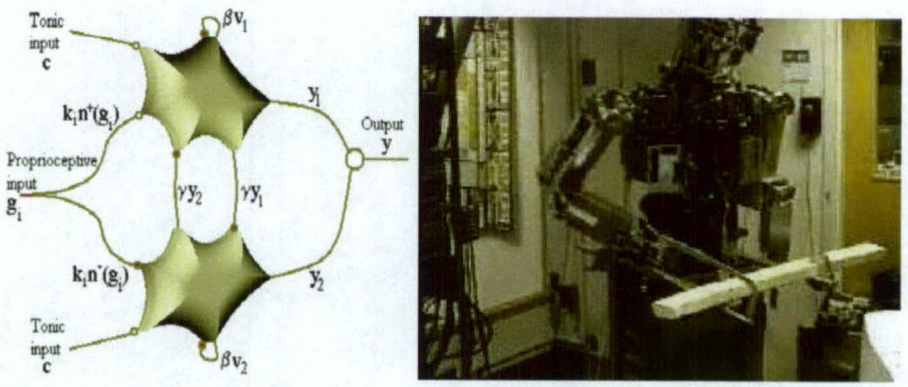


Figure 9-1: (left) Matsuoka neural oscillator, consisting on two mutually inhibiting neurons. (right) The humanoid robot Cog has two, six degree-of-freedom (6-dof) arms and a 7-d.o.f head. Each joint is driven by a series elastic actuator (Williamson, 1999). Hence, this compliant arm is designed for human/robot interaction.

Other research work (Schaal and Atkeson, 1994; Schaal et al., 2000) described robotic tasks such as a robot juggling or playing with a devil stick (hard tasks even for humans). However, the authors assumed off-line specialized knowledge of the task, and simplified the perception problems by engineering the experiments.

In this chapter, focus will be placed on the development of neural inspired controllers for robotic creatures which exploit and learn the natural dynamics of their own bodies and interacting environments.

Cognitive Issues

Neural oscillators & Biological Rhythmics. Autonomic oscillatory patterns of activity are widely spread in living beings. Several oscillatory or rhythmic movements or activities occur in most animals, such as respiration, heart beat, or homeostatic regulation. Almost all homeostatic functions of the body are regulated by circadian rhythms, and there is evidence for the existence of coordination between these rhythms (especially in the hypothalamus) (Kandel et al., 1992).

In primates, it is possible to distinguish two main oscillators. One possibly driven by the suprachiasmatic nucleus, which controls plasma growth hormone, slow-wave sleep and skin temperature. The other controls rapid eye movement (REM) sleep, body core temperature, plasma corticosteroids, and potassium excretion (Kandel et al., 1992). Other oscillatory activities include the locomotion of quadrupedes and the swimming of fish, or the flying of birds (Purves et al., 2001). Although locomotion is a voluntary movement in humans, once initiated the control is no more conscious. Certain reflexes, such as some spinal reflexes, also consist of rhythmic movements (Rossignol et al., 1996). Rhythmic scratching occurs after the animal having moved his limb to the starting posture, even in animals with the cervical cord damaged. Although these reflexes do not require input from higher-order cortical centers, they

depend on feedback from sensors, since properties of the reflex depend both on duration and intensity of the stimulus, (Kandel et al., 1992). Finally, innate Central Pattern Generators located in the spinal cord generate coordinated rhythmic patterns for the contraction of the several muscle groups involved in walking, (Purves et al., 2001).

These neural circuits are often modelled using a half-center model, consisting of motor neurons having mutually inhibitory synapses. Inspired by such biological systems, Matsuoka neural oscillators (Matsuoka, 1985, 1987) offer a plausible model for controlling rhythmic movements. The Matsuoka model consists of neurons connected in such a way that one neuron's excitation inhibits the other neuron's excitation, representing a kind of fatigue or adaptation effect (Kandel et al., 1992). These oscillators have entrainment properties with an external dynamic system to which they are coupled, being robust to disturbances and very stable. There is biological evidence for the existence of oscillation pattern generators in animals (Rossignol et al., 1996). Humans often exploit the natural dynamics of the body (Purves et al., 2001) to carry out tasks spending the minimum of energy, and this property is also fully exploited by such networks.

Developmental Issues. Human movements, as simple as reaching or more complex grasping manipulation, are learned developmentally over childhood, starting from the newborn inability to perform any coordinated movement. Such inability is mainly due to:

- ▷ Awkward postural control of trunk and limbs during the first trimester of life (Milani-Comparetti and Gidoni, 1967). Lack of knowledge of their dynamic properties, such as moments of inertia or stiffness.
- ▷ Most cortical projections to the spinal cord are not differentiated – neural organization of the control structure occurs after birth (Quartz and Sejnowski, 1997).

The newborn's movement repertoire includes muscular reflexes for which the feedback mechanisms are mainly located at the spinal cord. But around one week after birth infants try to goal-direct their arms towards objects (through small, rough, non-reflex movements). The also roughly orient their heads and eyes to track a salient, moving object (Trevarthen, 1980). Indeed, as introduced before, during the first month infants are pre-disposed to select salient visual stimulus from bright colorful objects or objects under abrupt or repetitive patterns (at suitable frequencies for interaction) of motion.

By the end of the first trimester of life, infants somehow are able to incorporate such rough perceptual capabilities into the motor system, being able to perform movements with discontinuous patterns of motions (such as poking by swiping their arm roughly towards an object), or oscillatory, jerky-like movements over long-periods of time. Since trajectory correction is still absent, these movements seem to be somewhat pre-programmed motions (Bower et al., 1970). The perceptual (visual or auditive) role is constrained to trigger the movement, but not to correct it. These earlier forms

prepare and enable adaptively more advanced forms of behavior to develop within the situated context they provide (Diamond, 1990).

The developmental work of this thesis for the robot's motor control is motivated by infants limited but effective early control ability. Starting from learning about objects and actions from simple perceptual stimulus, as described in chapter 3, the robot builds very simple models for the tasks of moving repetitively objects, such as waving a hammer for hammering, or for swiping motions such as poking an object without continuous close loop visual feedback. We will show in this chapter the implementation of controllers to achieve this motor capability, namely by developing a sliding mode controller for reaching motions and a neural oscillator's control of rhythmic motions. We will start, similarly as infants during their first two-three months of life, by building sensory-motor maps to learn kinematic (e.g., the limbs configuration) and dynamic properties of the robot's arms and head.

The next chapter will then demonstrate that the infants' mutual perceptual and control capabilities are what is need to enable the humanoid robot to generate by himself informative multi-modal percepts.

9.1 Learning Proprioceptive Maps

Controlling a robotic manipulator in the cartesian 3D space (e.g., to reach out for objects) requires learning its kinematic configuration – the mapping from joint space to cartesian space – as well as the inverse kinematics mapping. The problem of learning sensory-motor maps had been previously addressed at the Humanoid Robotics Group of the MIT CSAIL Laboratory, by building visual-servo mappings using the recognition of the robot's grip (Metta and Fitzpatrick, 2002) or else the recognition of the human hand (Marjanović et al., 1996) to guide the learning process.

Other techniques in the literature include motor-to-motor mappings (Metta et al., 1999). This work initially mapped two head/eye combined gazing parameters to an equal number of arm control parameters. More recently eye vergence was added to account for depth information (Metta et al., 2000). Another similar strategy (Gaskett and Cheng, 2003) is based on an on-line control learning scheme for reaching using a motor-motor mapping. It combines the reactive endpoint closed-loop visual control to move eyes, head, arm, and torso, based on both the robot's hand and target position, as well as a learned open-loop visual servo control which brings the hand into view by moving it closer to the visible target.

An alternative technique relies on the estimation of locally linear models for sensory-motor learning (Schaal and Atkeson, 1994), which is the grounding framework used for building sensory-motor maps on the humanoid robot Cog (see Appendix C).

9.1.1 Head Sensory-Motor Maps

In contrast with algorithms introduced so far for perceptual learning, sensory-motor maps are built using supervised learning techniques operating on off-line training data. Although such a constraint can be relaxed as described in Appendix C, the

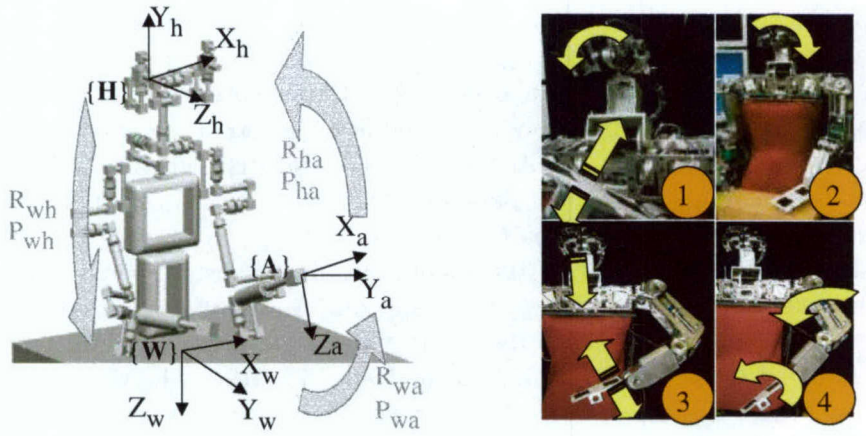


Figure 9-2: Learning proprioceptive maps. (left) {W}orld, {A}rm and {H}ead frames, and transformations among these frames (eg. R_{wa} stands for the orientation of frame {A} relative to {W}). (right) Motor behaviors for 1) calibrating the camera; 2) learning the head kinematics; 3) learning the arm kinematic models; and 4) learning the arm dynamics.

algorithms still require the availability of a minimum number of initial training points to avoid ill-conditioned solutions.

The *Intel Calibration Library* is used to both detect the corners of a calibration object inserted at the robot's end-effector, and to compute the intrinsic and extrinsic camera parameters. The calibration object consists of a grid of black and white squares. It is used not only for camera calibration but also to estimate the position and orientation of the robot's end-effector, to which this object is attached. Data for computing the camera intrinsic parameters is collected by moving both the arm gripper and camera (see figure 9-2-1) over a sequence of 20 images. Although the use of a calibration object could be dispensed in favor of the stereo visual localization of the grip from object recognition (after the acquisition of training grip templates), such approach would not be reliable for estimating the grip orientation due to the image's low resolution.

A calibrated camera (Appendix B) permits the extraction of the extrinsic parameters from the calibration object in the robot's arm grip. Inverting such transformation gives the transformation from the camera's focal center relative to the world referential on the grid (R_{wh}, P_{wh}). The calibration object on the arm gripper is kept within the camera's field of view by visually tracking it's centroid using an attentional tracker modulated by the attentional system (Fitzpatrick, 2003b). A PD controller controls the eyes (see this chapter's Section 9.3) in close-loop receiving as feedback input the centroid's retinal position. Inertial data is used to stabilize the robotic head and for VOR compensation.

Hence, moving 6 out of the 7 degree of freedom robotic head (just one of the two robot eyes is required) around a stationary arm gripper (see figure 9-2-2), enables the

estimation of the forward kinematics model using locally affine models. This model maps the head joints' configuration (given by 6 parameters) to the cartesian position and orientation of the camera's focal center relative to a world referential. Learning of these models is carried out by applying a modified version of memory-based Locally Weighted Regression (Schaal and Atkeson, 1994), which is reviewed in Appendix C, equivalent to linear receptive field networks.The input space corresponds therefore to the $m = 6$ head joint positions θ^h_j, for $j = 1, \ldots, p$, where p is the number of joint positions covered by moving the robot's head over several configurations, and θ^h is a 6-dimensional vector. The output space q^h is the $n = 6$ cartesian location of the head referential relative to the world referential, corresponding to the twist of (R_{wh}, P_{wh}): $q^h(1, ..., 3) = P_{wh}(1, ..., 3)$ and $\exp(\hat{w})=R_{wh}$, where $w = q^h(3, ..., 6)$ and the operator $\hat{}$ is defined from the cross-product operation $\hat{w}v=w \times v$ (Murray and Sastry, 1994). Training data for locally weighted regression consists of pairs $(x, y) = (\theta^h, q^h)$ generated by $q^h = F(\theta^h)$ from the set of postures resulting by moving the robotic head.

The head inverse kinematics mapping is also computed, by just switching the roles of θ^h and q^h, being the input now q and the robotic head joints θ^h the output. The problem now consists of estimating the non-linear function $\theta^h = F_i(q^h)$, solved also by fitting locally linear models to such function using locally weighted regression. In summary,

1. Calibrate camera
2. Move head keeping calibration grid stationary
3. Track calibration grid with the head, keeping it in the visual field of view
4. For each posture of the head (angle configuration), compute the location of the head referential (R_{wh}, P_{wh}). This vector defines the inverse of the transformation defined by the extrinsic parameters of the calibrated camera computed from the calibration grid
5. Collect the set of postural joint angles θ^h corresponding to this head posture
6. Learn head Forward Kinematics map (θ^h, q^h) by locally linear regression (Appendix C)
7. Learn head Inverse Kinematics map (q^h, θ^h) by locally linear regression.

9.1.2 Arm Sensory-Motor Maps

A calibrated camera and a head forward kinematics map are sufficient conditions for the estimation of both the arm forward and inverse kinematics. This is accomplished by moving the robotic arm over the joint configuration space, while visually tracking it's gripper (see figure 9-2-3) by the attentional tracker. The arm cartesian location (6 parameters) is determined from two transformations: one from the gripper to the camera's focal center (R_{ha}, P_{ha}), given by the camera's extrinsic parameters, and the other from the focal center to the world referential (R_{wh}, P_{wh}), given by the head forward kinematics. The arm location is hence given by the composition of the two

transformations: ($R_{wa} = R_{wh}R_{ha}, P_{wa} = P_{wh} + R_{wh}P_{ha}$). The corresponding twist q^a, together with the 6 arm joint angle measurements θ^a, generates training data to build locally affine models of $q^a = F(\theta^a)$, the arm forward kinematics, as well as of $\theta^a = F_i(q^a)$, the arm inverse kinematics. This is done using memory-based Locally Weighted Regression (Schaal and Atkeson, 1994) – off-line, batch minimum least squares over locally linear models (Apppendix C). In summary,

1. Compute head forward kinematics
2. Move head and arm over the robot working space
3. Track calibration grid with the head, keeping it in the visual field of view
4. For each posture of the head (angle configuration), compute
   - ▷ the location of the head referential (R_{wh}, P_{wh}) given by forward kinematics
   - ▷ the location of the arm relative to the head referential, given by the extrinsic parameters of the calibrated camera computed from the calibration grid at the arm's gripper
   - ▷ the location of the arm relative to the world referential, given by the composition of the two previous transformations
5. Collect the set of postural joint angles θ^a corresponding to this arm posture
6. Learn the arm Forward Kinematics map (θ^a, q^a) by locally linear regression
7. Learn the arm Inverse Kinematics map (q^a, θ^a) by locally linear regression.

Statistical results for building these maps are shown in Table 9.1. Errors in the arm kinematics estimation are larger than the head kinematics estimation (especially for orientation) because, unlike the head, the arm joints are actuated by serial elastic actuators, being therefore very compliant. In addition, force feedback from the strain gauges is very noisy, which decreases considerably the range of admissible values that the controller gains can take for a stable system. In addition, errors from head kinematics estimation will also propagate to the arm kinematics estimation. The alternative would be to map the cartesian location as an explicit function of both head and arm joints $q^a = F(\theta^a, \theta^h)$, but this would increase the dimension and hence processing times. In addition, locating the robotic arm on the retinal plane is straightforward from the approach here presented, as described next.

Proprioceptive-Visual Mappings

Detection of the robotic arm's end-effector on the retinal plane follows from both head and arm forward kinematics. Such maps predict the head/arm joint configurations (R_{wh}, P_{wh}) and (R_{wa}, P_{wa}), respectively, from which ($R_{ha} = -R_{wh}^T R_{wa}, P_{ha} = P_{wh} - R_{wh}^T P_{wa}$) follows. The gripper location in image coordinates is then just the perspective projection of P_{ha}, which is given by AP_{ha}, where A is the camera calibration matrix. Hence, for any configuration of both head and arm joint angles, this map gives the corresponding retinal position of the gripper referential.

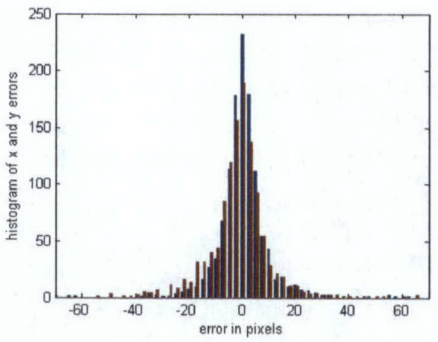


Figure 9-3: Error histogram for locating the arm's end-effector on the image, for various head/arm joint configurations.

Table 9.1 shows statistical data for such a mapping, while figure 9-3 presents an error histogram. From both table and figure it is seen that there is a very high probability to find the grip within the camera field of view by application of this map.

9.1.3 Jacobian Estimation

The velocity mapping between joint and world coordinates has a very important particularity: it is linear with respect to velocities, depending only on position data. Indeed,

$$\dot{q} = J(\theta)\dot{\theta} \tag{9.1}$$

Therefore, instead of having $y = F(x)$ as in Appendix C, now the non-linear input-output relation is more structured, given by $y = F(x^{nl})x^{l}$. Learning locally linear models from $y = F(x = [x^{nl^T}\, x^{l^T}]^T)$ enables learning maps which depend on all x for creating the receptive fields, hence ignoring the additional constraints available.

The approach described here to exploit the Jacobian structure consists of creating the receptive fields with x^{nl}, which models the non-linear dependence of the Jacobian on such a variable, but using only x^{l} to estimate locally linear models with no bias constant. This is equivalent to a zero order Taylor expansion of $F(x^{nl})$ into locally linear models of the form $y = F(x_0^{nl})x^{l}$. A first order Taylor expansion of $F(x^{nl})$ would lead to the non-linear models of the form $y = F(x_0^{nl})x^{l} + (\frac{\partial F}{\partial x^{nl}}|_{x^{nl}=x_0^{nl}} x^{nl})x^{l}$, which would have to be approximated again. The final form would include bias terms (requiring an affine model). But the linear property of such a mapping is important, and will be exploited in Section 9.3.

The linear weighted regression (LWR) is now computed, in pseudo-code (with $x^{nl} = \theta$, $x^{l} = \dot{\theta}$ and $y = \dot{q}$) as:

Given:

a

Mse	Head Map	Arm Kinematics
X (cm^2)	177	224
Y (cm^2)	75	173
Z (cm^2)	74	381
L_2 norm	326	778
Pitch	5.81	17.15
Yaw	5.75	62.78
roll	2.46	16.35
L_2 norm	14.02	96.29

b

mse	Inverse Arm Kinematics
J_1	11.6
J_2	4.5
J_3	17.41
J_4	10.17
J_5	9.32
J_6	2.63
L_2 norm	61.64

c

mse	Retinal plane (240×320)
x	117.73
y	215.78
L_2 norm	215.88

Table 9.1: Mean square errors for the sensorimotor maps. Experimental results in a set of 1300 validation points (50% of total). The other 50% are used for training. a) The ground truth data is obtained from visual information, starting by the identification of the corners of the calibration object, and then determining the extrinsic parameters using the calibration matrix to obtain the 6-dimensional location (in position and orientation) of the gripper referential. The error is given by the average sum (for all validation points) of the square of the difference between the 6-dimensional location predicted by the head/arm forward kinematics and the ground truth data. Since data is given in cm, the error appears in cm^2 for cartesian positions and in $degree^2$ for orientation angles. Hence, the mean absolute error for the L2 norm of the 3-dimensional position is $\sqrt{778} = 27.9cm$.
b) Inverse arm kinematics. The units are given in $degree^2$. The ground truth data corresponds to the value read by the potentiometers at each arm joint, and the prediction corresponds to the joint angle predicted by the inverse kinematics map. Joints $1, \ldots, 6$ correspond by increasing order to tilt (shoulder), roll (shoulder), pan (elbow), tilt (elbow), pan (gripper) and tilt (gripper). As can be seen from the table, there are significant errors (pan angle mean absolute error at the elbow joint is $\sqrt{17.41} = 4.17^o$). This is in part due to noise from the potentiometer, but especially because of very high noise from the strain gauges.
c) Mean square error (in $pixel^2$) for the retinal map. The ground truth data is the position of the gripper referential, given by a corner of the calibration object. The prediction is given by the proprioceptive-visual map, which maps head and arm joint angles to retinal coordinates of the grip referential. The mean absolute errors are $\sqrt{117.73} = 10.85pixels$ in x and $\sqrt{215.78} = 14.69pixels$ in y. These errors imply that, if the predictions do not center the object at the image center, at least the object will most probably appear within the field of view.

- ▷ a query point $x_q = (x_q^{nl}, x_q^l)$
- ▷ p training points $\{(x_i, y_i\}$ in memory, where x_i is an n-dimensional vector and y_i an m-dimensional vector

Compute prediction:

1. determine weight diagonal matrix W with $w_{i,i} = exp(-1/2(x_i^{nl} - x_q)^T D(x_i^{nl} - x_q))$
2. build matrix X from homogeneous vectors $\tilde{x}_i$, and output matrix Y such that
   $X = (x_1^l, x_2^l, \ldots, x_p^l)^T$,
   $Y = (y_1, y_2, \ldots, y_m)$
3. compute locally linear model

$$\beta = (X^T W X)^{-1} X^T W Y = P X^T W Y \tag{9.2}$$

4. the predicted output value is $\hat{y}_q = x_q^l \beta$.

9.2 Control Integration Grounded on Perception

Classical Control theory – such as PID controllers (Nise, 1995) or Optimal Controllers using dynamic programming optimization (Mosca, 1995) – has been around for about half-a-century, and has been used extensively for the control of robotic manipulators. In the last decade, Modern Control Theory techniques (such as H_2, H_∞ controllers, μ synthesis/analysys (Zhou and Doyle, 1998; J. Doyle, 1992; Vidyasagar, 1985) or non-convex optimization (Boyd et al., 1994)) have surged in the control of both linear and non-linear systems, including robotic manipulators.

However, these techniques rely heavily on the knowledge of the dynamics of the system to be manipulated, as well as on the quantification of the disturbances that may affect them. Although robots have already completed successfully a complex task, they were programmed to execute that single task. Different tasks require initial setups and possibly an entire new theoretical framework to execute the task.

In addition, these complex tasks often involve the control of a simple kinematic chain (such as inverted or double pendulums), for which not only the order of the system is pre-specified, but also the kinematics is always known. For systems with an unknown dynamics, most techniques rely on special markers on the links for learning visually the system dynamics.

The multi-scale time-frequency analysis developed in chapter 3 offers an elegant solution for integrating oscillatory and non-oscillatory tasks, or mixtures of both. Indeed, tasks can be communicated to the robot with simultaneous frequency and spatial desired contents. The integration of both rhythmic and non-oscillatory control movements is possible by such information stored in the task description.

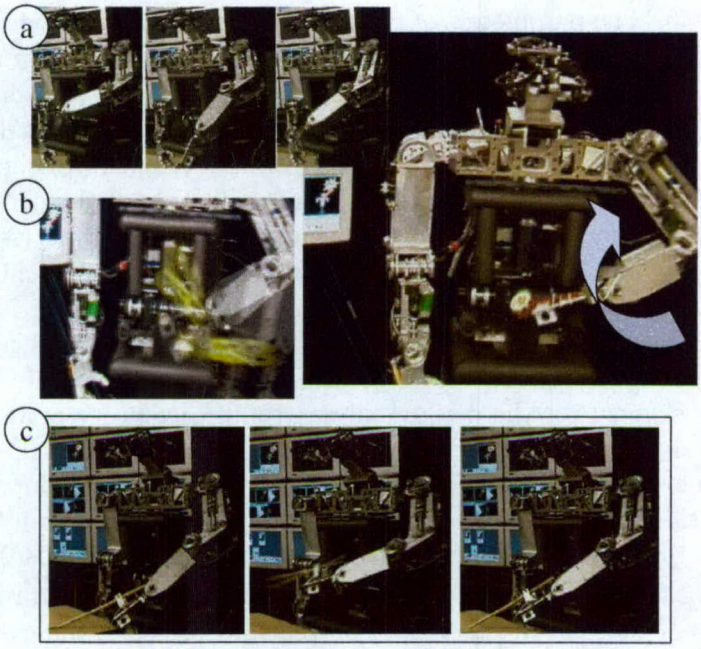


Figure 9-4: The humanoid robot Cog playing three musical instruments using neural oscillators. The latter receives proprioceptive feedback a) shaking a tic-tac drum b) shaking a tamborine c) drumming with a drum stick.

9.2.1 Control of Oscillatory Motions

The capability of the Matsuoka neural oscillator to entrain the frequency of the input signal or resonance modes of dynamical systems have been used for the generation of motion patterns to support legged locomotion (Taga et al., 1991; Taga, 1995), and for the control of robotic arms during the execution of different tasks (Williamson, 1998b, 1999), such as crank turn or drumming. The oscillator cancels the damping of the coupled dynamical system and entrains its resonance frequency. Thus, the neural oscillator adapts to the system dynamics to which is coupled, producing the minimum amount of energy required to control the system (Williamson, 1998b). In addition, multiple neural oscillators may be applied, and they may also be connected in networks to control Multiple-Input-Multiple-Output (MIMO) systems, so that complex behaviors may emerge (Taga et al., 1991).

Neural oscillators can adapt locally to the dynamics of the controlled system, being an important element of biological motor control systems. The tuning of mutual inhibiting neurons that model these sustained oscillations is difficult due to the highly non-linear dynamics, and thus current methods are based on trial and error or simulation procedures. To accomplish complex behaviors or execute hard tasks, networks of oscillators can be connected, making the tuning of the neural parameters even more difficult.

The Matsuoka neural oscillator consists of two neurons inhibiting each other mu-

tually (figure 9-1). The parameters of the nonlinear dynamics of the oscillator are the tonic c, gains γ, β, k_i, time constants τ_1, τ_2, an external input and four non-linearities. We proposed a mathematical analysis for multiple nonlinear oscillators connected to a (non)linear multi-variable dynamic system, by using multiple input describing functions (Arsenio, 2000c,a). As a result, the framework developed provides estimates for the frequency, amplitudes and/or parameters of oscillation of the controlled system, as well as an error bound on the estimates, using algebraic equations (Arsenio, 2000b). A time-domain stability analysis based on control theory for switching surfaces on piece-wise linear systems was initially introduced by (Williamson, 1999), and fully developed by (Arsenio, 2004c). Such a mathematical framework is fully detailed in Appendix D. Although this vast and original work on neural oscillators constitutes a very important element for the design and control of oscillatory motions, it includes much technical detail which is not essential for understanding this thesis core framework, and therefore was relegated to the end of the manuscript for interested readers. Such mathematical framework enables the design of networks of neural oscillators to control the rhythmic motions of an artificial creature (Arsenio, 2004c).

Strategies for the oscillatory control of movements of a humanoid robot are imperative, especially if they result on *natural* movements, which is the case of Matsuoka neural oscillators, since they track the natural frequency of the dynamic system to which they are coupled. As results in figure 9-4 show, playing musical instruments is an application where tuning of oscillations plays a rather important role. The wrong set of parameters result most often in no oscillations or else low-amplitude or low-frequency oscillations, which is an undesirable behavior (no rhythmic musical sound was produced during forced-response experiments to a sine wave generated input). But using the framework just referred, tuning is fast and effective, and rhythmic sound was produced.

Such production of sounds is closely associated to the neural oscillator's property of entraining the natural frequency of the dynamic system to which it is coupled. Figure 9-5 shows results for two different experiments consisting of having the robot shake a castonete and a rattle for producing musical sounds, with and without joint feedback activated. Clear rhythmic sounds are produced with feedback. But without it the robot's arm shakes loosely (no entrainment), and the rhythmic sound just does not come out.

9.2.2 Sliding Mode Control

The dynamics of an arm manipulator is strongly nonlinear, and its nonlinear dynamics poses challenging control problems. Especially for mechanisms with small gear transmission ratios or low-reduction cable-driven systems or direct-drive connections, nonlinear dynamic effects may not be neglected. The state space of Cog's $n = 6$-linked articulated manipulator is described by the n dimensional vectors θ and $\dot{\theta}$ of joint angles and velocities, respectively. Its actuator inputs τ consist of a n dimensional vector of torques applied at the joints. The nonlinear dynamics can be written as the system (Murray and Sastry, 1994):

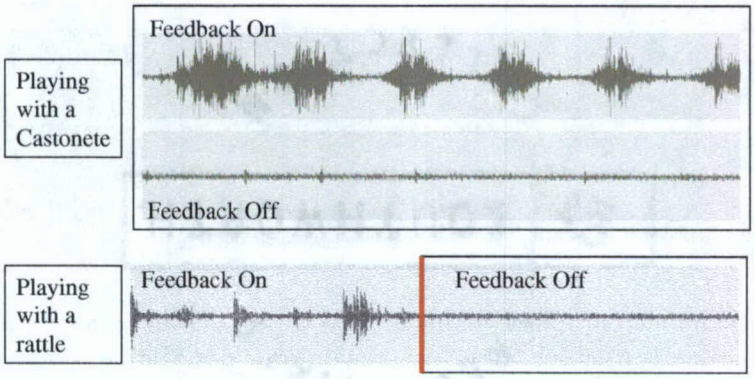


Figure 9-5: Rhythmic sounds produced by having the robot play two musical instruments – a castonete, and a rattle – with and without joint feedback. No entrainment occurs, and therefore rhythmic production of sounds is inhibited, without joint feedback.

$$H(\theta)\ddot{\theta} + C(\theta, \dot{\theta})\dot{\theta} + g(\theta) = \tau \tag{9.3}$$

where $H(\theta)$ is the manipulator inertia matrix (which is symmetric positive definite), $C(\theta, \dot{\theta})\dot{\theta}$ is the vector of centripetal and Coriolis torques, and $g(\theta)$ is the vector of gravitational torques. The feedback control problem for such a system is to determine the actuator inputs required to perform desired tasks from the measurements of the system state $(\dot{\theta}, \theta)$ of joint velocities and angles, in the presence of uncertainty.

As described in detail by (Slotine and Weiping, 1991; Murray and Sastry, 1994), control theories that are most often applied to such systems are Sliding Mode Controllers, PD and Computed Torque Controllers. The latter two are often used to reach desired positions. Sliding mode becomes rather useful to follow desired trajectories specified by $(\theta_d, \dot{\theta}_d, \ddot{\theta}_d)$, the position, velocity and acceleration for each manipulator joint, under model uncertainty.

A sliding mode control law (Slotine and Weiping, 1991) was implemented, given by equations 9.4 and 9.5, where s is a weighted sum of position $(\tilde{\theta} = \theta - \theta_d)$ and velocity errors.

$$\begin{aligned} \tau &= \hat{H}(\theta)\ddot{\theta}_r + \hat{C}(\theta, \dot{\theta})\dot{\theta}_r + \hat{g}(\theta) - K(\theta_d)sat(\Phi^{-1}s) \\ &= \hat{\tau} - K(\theta_d)sat(\Phi^{-1}s) \end{aligned} \tag{9.4}$$

$$s = \left(\frac{d}{dt} + \Lambda\right)^{m=2} \tilde{\theta} = \dot{\tilde{\theta}} - \Lambda\tilde{\theta} \tag{9.5}$$

The non-linear dynamics is learned adaptively on-line (see next Section). These non-parametric locally affine models are used to predict the feedforward term $\hat{\tau}$. The reference velocity is given by $\dot{\theta}_r = \dot{\theta} - \Lambda\tilde{\theta}$, and Λ is a symmetric positive definite matrix (assumption can be relaxed so that the matrix $-\Lambda$ is Hurwitz (Slotine and Weiping, 1991)). The matrix Φ defines the boundary layer thickness,

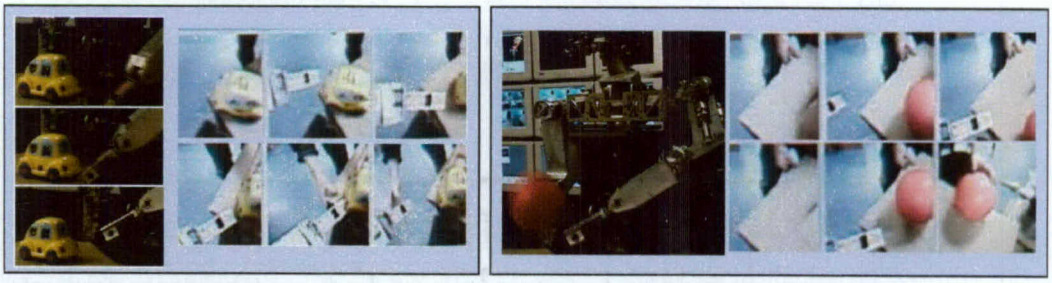


Figure 9-6: The humanoid robot poking a (left) stationary and (right) moving object. Cubic interpolation is used for trajectory generation, and Sliding Mode controller for feedback control.

$$\dot{\Phi} + \lambda\Phi = K(\theta_d) \tag{9.6}$$

leading to tracking to within a guaranteed precision $\varepsilon = \Phi/\lambda^{m=2}$. Three factors impose upper bounds on $\lambda \approx \lambda_R \approx \lambda_s \approx \lambda_D$: structural resonant nodes ($\lambda \leq \lambda_R$); time delays ($\lambda \leq \lambda_D = 1/T_D$), where T_D is the largest unmodeled time delay (which was set to one sampling interval - 5 ms); and sampling rate ($\lambda \leq \lambda_s = 1/5\nu_{sampling} = 40Hz$).

The Sliding Mode Controller was applied to control Cog's arms for simple movements, such as reaching for an object or poking it (see figure 9-6 for robot poking a stationary and a moving target).

9.3 Cerebellum

9.3.1 Eye Movements

The control of the eyes with visual feedback requires a camera model, which is simplified to a linear gain for velocity signals sent to the actuators. The robotic eyes are able to perform a set of basic movements made by awake, frontal eyed, foveal animals (Carpenter, 1988): saccade, smooth pursuit, vergence, and vestibulo-ocular-reflex. Saccadic eye-movements consist of very rapid ballistic motions that focus a target on the fovea (the central area of the visual field). Since the animal does not see well during these movements, these eye-movements are not controlled by visual feedback during their execution. In humans, saccades are extremely rapid, often up to 900 degrees per second. Smooth pursuit movements are slow and accurate movements of the eye, to keep the target on the fovea. Vergence movements adjust the eyes to view a moving target at different depths.

For vergence control stereo images from the wide-field of view cameras are correlated with each other in logpolar coordinates. One of the images is displaced by an horizontal shift in cartesian coordinates, transformed to log-polar coordinates, and then correlated with the other image. Disparity is estimated by the displacement corresponding to the maximum peak of the normalized correlation cost. This dispar-

ity value is fed into a velocity PD controller, which controls both eyes with opposite signs. In order to decrease processing times, images are matched at varying resolutions. Accurate disparity estimates are maintained by performing the correspondence at low resolution, and then a second higher resolution search in a neighborhood of the low resolution result.

Smooth pursuit movements adjust eye movements disregarding movement in depth, while vergence movements adjust the eyes for gazing at a target at different depths. There is neurophysiological evidence to corroborate performing vergence at lower rates than smooth pursuit. Hence, I gave less weight to the vergence signal. It was verified experimentally that this strategy improves the stability of the visual system. Saccades are ballistic (very fast) movements, executed without visual feedback. There may be so much blur in the images during a saccade that visual information would mostly be unreliable. Thereafter, visual feedback is enabled and the loop is closed using velocity control for smooth trajectories, unless the position error increases above a threshold. Input for the PD velocity controller is received from the attentional system.

Vestibular movements are automatic movements induced by the semicircular canals of the vestibular system. Acceleration measurements from both the semicircular canals and the otolith organs in the inner ear are integrated to measure head velocity. This measurement is used to compensate for fast head rotations and/or translations by counter-turning the eyes to stabilize the image on the retina, maintaining the direction of gaze. During head rotations of large amplitude, saccadic movements interrupt these movements. Two rotational velocity signals (yaw and pitch) extracted from the gyro are subtracted from the eyes' control velocity command, so that these compensate for movements of the head and body, keeping a salient stimulus in the cameras field of view. The roll position signal is compensated by a PID controller, which maintains the eyes baseline parallel to the ground.

9.3.2 Non-parametric Learning of Manipulator Dynamics

The dynamic model of the arm robotic manipulator is often unknown or else known with a large uncertainty. Standard controllers, such as the PID controller typically used in industry, do not require a dynamic model of the system being controlled. But performance is lost at high frequencies and instability might occur for very compliant systems. Non-parametric learning of the arm dynamics was hence implemented with a twofold goal:

- ▷ Compensate for unknown dynamics on the feedback controller
- ▷ Estimate tuning parameters of the neural oscillators

using an on-line, iterative version of Receptive Field Weighted Regression (Schaal and Atkeson, 1994) (see Appendix C), using as input space $(\theta, \dot{\theta}, \dot{\theta}_r, \ddot{\theta}_r)$ (since $\ddot{\theta}$ is unknown), and as output space the 6-dimensional vector τ of joint torques.

Compliance, Impedance, and Poli-articulated muscles

Joint compliance or impedance is easy to specify with joint feedback control using a diagonal matrix of gains $\tau = -K_p\theta$, but it is usually of reduced interest. On the other hand, manipulator compliance is often required in cartesian space, to accomplish a specific task such as insertion of a pin in a hole or safe human-robot interaction. Since the manipulator is controlled by torques, it becomes necessary to map the gain matrix K_p^c ($F = K_p^c q$) in cartesian space to K_p. Since the Jacobian enters in the transformation which maps cartesian gains K_p^c into joint gains K_p (Murray and Sastry, 1994), a linear model for it is required to determine the gain matrix in the joint space which corresponds to the desired compliance in the cartesian space. In humans, this is accomplish through poli-articulated muscles. A muscle actuates a group of joints, and each joint is actuated by several muscles. But such interconnectedness can be simulated by software for low frequency movements (high frequency operation requires real mechanical coupling).

Hence, compliance or impedance can be specified according to desired directions and values from knowledge of the arm manipulator's Jacobian and dynamic model. It is possible then to control Impedance or Compliance by varying such gains. Although this procedure was never tested on the humanoid robot Cog, it has a lot of potential for future developments for robot grasping and object manipulation.

9.4 Sensorimotor Integration

Figure 9-7 shows the humanoid robot Cog executing a task requiring the integration of both reaching movements of the arm and rhythmic motions, as well as the control of eye movements. The task consists of having the robot playing rhythmic sounds with a castonete by first reaching its gripper to the castonete and thereafter shaking it. Such actions enable Cog's perceptual system not only to extract visual descriptions of the castonete, but also the acoustic pattern that it produces. Percepts from these two different modalities are linked by correlating amodal features – timing and synchrony – through cross-modal processing, as described in chapter 7.

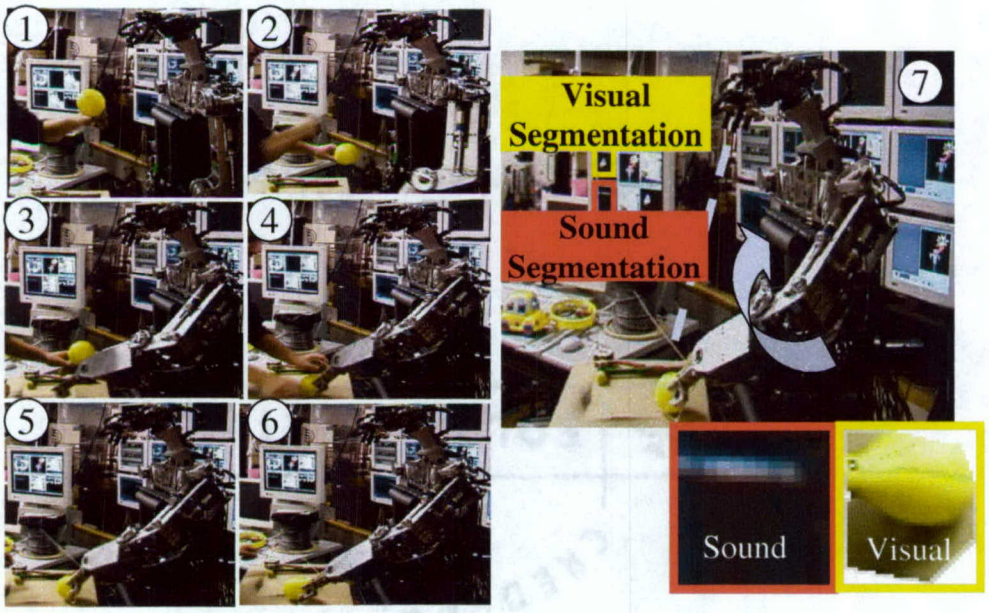


Figure 9-7: Reaching to and shaking a child's toy (a castonete). (1-2) A human actor attracts Cog's attention to the toy, by creating a salient stimulus to its attentional system. (3-4) The robot reaches to the object – feedback control applying a sliding mode controller. (5-6) Cog shakes the toy. Feedback proprioceptive signals are sent into a neural oscillator, which entrains the natural frequency of the dynamic system to which its coupled (the robot's arm), producing rhythmic sounds. (7) Visual and auditory segmentations by Cog's perceptual system. It shows two segmented images - one for a low resolution sound spectrogram over one period of oscillation, and the other for the toy's visual template extracted from the periodic movements of the toy.

Chapter 10

Frontal Lobe

If that [neural connection to emotional memories] is broken down, you are at the mercy of facts and logic, and that just is not enough.

(Damasio, 1994)

There is a large spectrum of applications for flexible robotic tool handling in industrial environments or for service robots. Children start learning and developing such competencies by playing with toys. Hence, toys are widely used throughout this work as learning aids. The operation of handling tools, such as hammers or swiping brushes, requires not only object location and recognition algorithms, but also algorithms to learn tasks executed with such tools. Therefore, task sequences will be learned on-line from the visual observation of human teachers.

Previous approaches for transferring skills from human to robots rely heavily on human gesture recognition, or haptic interfaces for detecting human motion (Dillmann et al., 1999; Aleotti et al., 2003). Environments are often over-simplified to facilitate the perception of the task sequence (Ikeuchi and Suehiro, 1994; Kuniyoshi et al., 1994). Other approaches based on human-robot interactions consist of visually identifying simple guiding actions (such as direction following, or collision), for which the structure and goal of the task are well known (??). Throughout this manuscript's work, task identification and object segmentation/recognition occurs while tasks are

being executed by a human teacher without any perceptual over-simplification or environmental setup.

Machine learning techniques have been applied to successfully solve a wide variety of Artificial Intelligence problems. In the humanoid robotics field, reinforcement or supervised learning techniques are often used to learn sensory-motor (kinematic) maps (?) or the model of a dynamic system. Among available learning techniques, neural networks, mixture of gaussians (?), Hidden Markov Models or Support Vector Machines (?) are most often used. The availability of a reward on reinforcement learning techniques is often sparse. On the other hand, supervised learning techniques often use training data manually annotated. This thesis proposes a strategy to overcome such limitations by on-line, automatic annotation of the training data through robot-human interactions or through active robotic manipulation. This learning strategy will be demonstrated for the problem of robot tasking. Tasks sequences are modelled as Markov Chains, being training data generated through the detection of events from interactions with a human instructor. Such information will also be applied to learn visual appearance templates for objects. These templates are the training data for object recognition.

These learning steps are performed in closed-loop. The action of the robot on an object creates an event which triggers the acquisition of further training data. For example, the event of a human poking a ball enables the robot to learn the *poking task* and the visual appearance of a ball for latter recognition. The robot becomes then able to poke other objects, such as a toy, which also enables learning the toy's visual appearance. Therefore, learning is implemented incrementally (similarly to what happens during child development phases).

Cognitive Issues (Lakoff, 1987) advocates that for an embodied system, task representations should be grounded in sensory-motor interactions with the environment.

Developmental Issues.

10.1 Task Identification

The world surrounding us is full of information. A critical capability to an intelligent system is filtering task relevant information. Figure 10-1 shows events detected from the demonstration of a repetitive task – hammering a nail.

The developed scheme poses robot tasking as learning the structure of an hybrid state machine. The discrete states of this machine correspond to events and the continuous states to the system dynamics. Remains a fundamental piece to be learn: the goal of the task. The goal might be verbally communicated to the robot (an elegant framework towards such interpersonal communication is described by (Fitzpatrick, 2003b)), which is out of the scope of this thesis, or can be inferred from the demonstration of a human caregiver. Indeed, both discrete and continuous states will be monitored for stationary or oscillatory conditions, which will translate into the task goal.

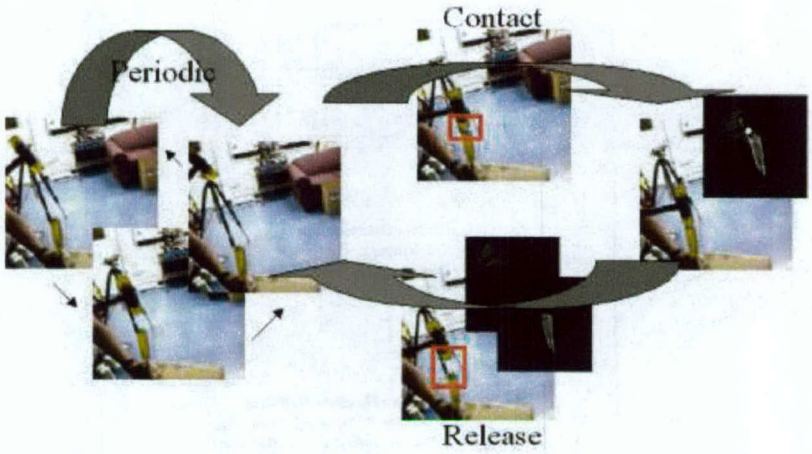


Figure 10-1: Hammering task. This task is both characterized by periodic motions, and also by spatial, discrete events. Through observation of the task being executed, the robot learns the sequence of events that compose a task, as well as the objects being acted on.

10.1.1 Tasks as Hybrid Markov Chains

Tasks are modelled through a finite Markov Decision Process (MDP), defined by five sets $< S, A, P, R, O >$. Actions correspond to discrete, stochastic state-transitions a $\in$ A={Periodicity, Contact, Release, Assembling, Invariant Set, Stationarity} from an environment's state $s_i \in S$ to the next state s_{i+1}, with probability $P^a_{s_i s_{i+1}} \in P$, where P is a set of transition probabilities $P^a_{ss'} = P_r\{s_{i+1} = s'|s, t\}$. Task learning consists therefore on determining the states that characterize a task and mapping such states with probabilities of taking each possible action.

State Space

States correspond to the continuous dynamics of a system, and are defined as a collection of objects $o \in O$ (O={hammer, nail, actuator,etc}) and a set of relations $r \in R$ between then (R={not assembled, assembled, in contact}.

Transitions

Sequence of images are analyzed at multiple time/frequency scales for the detection of periodic or discrete events caused by an agent's actions (Arsenio, 2003a). Transition statistics are obtained by executing a task several times. An action's frequency of occurrence from a given state gives the transition probability.

Goal Inference

The Markov chain jumps to a final state whenever the environment is stationary, or else whenever an invariant set is reached. Invariant sets are defined as a sequence of

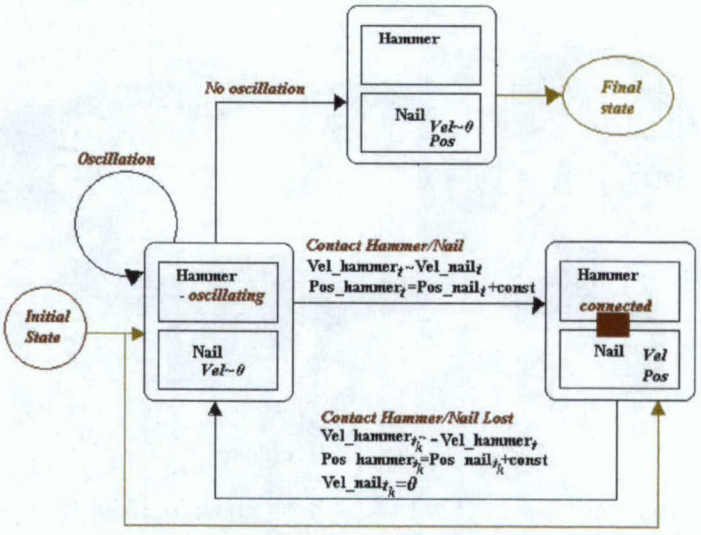


Figure 10-2: Hammering task as an hybrid state machine. The task goal is represented as the final state.

transitions which are repeatedly executed along the Markov chain. Invariant sets are detected by tracking sequences of actions and testing them for loops. All states that belong to an invariant set jump to a final state, as shown in figure 10-3.

10.1.2 Action Affordances

10.1.3 Applications

Most previous work in motion identification focused on human motion. Under this thesis framework, I expect to push these boundaries to arbitrary unknown systems, both with respect to object recognition and robot tasking.

Among simple tasks that this framework is designed to identify, it is worthy to refer

- ▷ Grabbing/Throwing - creation/removal of a kinematic constraint between an object and an actuator
- ▷ Assembling/Dissassembling - creation/removal of a kinematic constraint between two objects
- ▷ Slapping - position contact with discontinuous acceleration between an actuator and an object, without the creation or removal of a kinematic constraint (Impact - between two objects)
- ▷ Pulling/Pushing
- ▷ Control of an inverted pendulum

as well as simple oscillatory tasks:

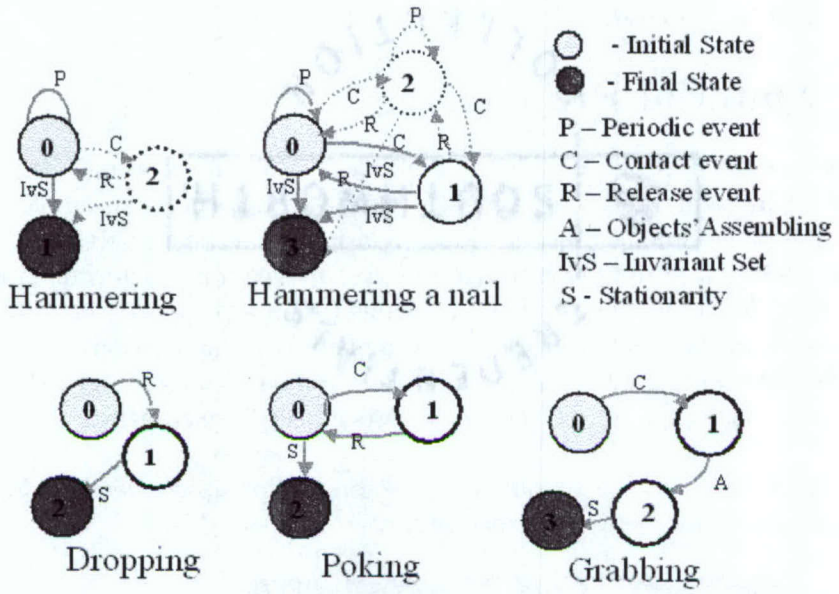


Figure 10-3: Hybrid Markov chains for different tasks. The chain for simply waving an hammer contains just two states. But an error occurred for one of the experiments in which an additional state, arising from a contact event with the hammer's own shadow, is created. Since tasks are executed from different locations and light conditions, this error is not statistically relevant. For the task of hammering a nail, contact between the hammer and a nail creates a transition from state 0 to state 1, while a release creates an opposite transition. An additional state is created whenever another actuator is holding the nail. The other graphs correspond to simple, non-oscillatory actions.

- ▷ Waving (frequency centered movement of an actuator)
- ▷ Sawing
- ▷ Juggling
- ▷ Cycle of Throws/Catchs
- ▷ Moving simple/multiple rotating cranes

The task repertoire should also include more complex tasks, that combine rhythmic and non rhythmic control movements, such as

- ▷ Painting a sheet of paper
- ▷ Hammering a nail

10.2 Functional Recognition

10.3 Material Properties

Perceiving mass relations is possible from interactions among objects. Contact events, as supported by figure 10-4, are especially useful to determine ensemble properties and mass relations among objects. Mass relations may be perceived as suggested in (Shelov, 1998), according to intuitive physics. Indeed, once determined the center of mass' velocities, the perceived mass relations (upon a perspective transformation) are obtained from the momentum conservation law (ignoring friction). Therefore, the mass of the car is perceived as much larger than the ball's mass - and indeed the ball has a smaller mass. Figure ?? shows the two object segmentations at the contact instant.

Needs more text - I need to elaborate also on bibliography. Needs more results.

Por aqui a dinamica em vez de na anterior

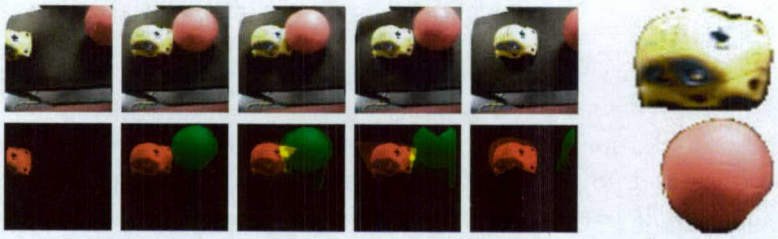


Figure 10-4: *A car impacts into a ball. Segmentations for both the car and the ball at the contact instant.*

10.3.1 Dynamic Properties

Video cameras are devices with finite resolution, and thus only discrete values in space are measured, although subpixel accuracy is possible. Because of the sampling rate (30Hz), only discrete values of the state variables are observable. Thus, what is observable is the discrete equivalent of the continuous system. Throughout this thesis only time-invariant systems will be considered.

For a continuous system with state variable $x(t)$ and input $u(t)$, the state equation is:

$$\dot{x}(t) = Ax(t) + Bu(t) \tag{10.1}$$

Solving this ordinary differential equation results the state transition equation:

$$x(t) = \phi(t - t_0)x(t_0) + \int_{t_0}^{t} \phi(t - \tau)Bu(\tau)d\tau \tag{10.2}$$

with $\phi(t) = e^{At}$. For a sampling interval $T = 33ms$, the discrete equivalent for such a system is given by equation 10.3:

$$x[(k+1)T] = \phi(T)x(kT) + \varphi(T)u(kT) \tag{10.3}$$

where $\varphi(T) = \int_0^T e^{A\sigma}Bd\sigma$. Equation 10.4 results from dropping the explicit dependence on T.

$$x_{k+1} = \phi x_k + \varphi u_k \tag{10.4}$$

The formalism needs also to deal with non-linear systems. With such in mind, the nonlinear system needs to be linearized locally. Given a non-linear system in state-space form:

$$\dot{x} = A(x) + Bu \tag{10.5}$$

or in general form:

$$H(q)\ddot{q} + C(q,\dot{q})\dot{q} + g(q) = ux = [\dot{q}q]^T \tag{10.6}$$

we linearize it around a point x_0, assuming small perturbations ϵ, i.e. $x = x_0 + \epsilon$. Expanding $A(x)$ in a Taylor series,

$$A(x) = A(x_0) + \frac{\partial A(x)}{\partial x}|_{x=x_0}x + \dots \tag{10.7}$$

eliminating the higher order terms, and replacing 10.7 into 10.5:

$$\dot{x} = A(x_0) + \frac{\partial A(x)}{\partial x}|_{x=x_0}x + Bu \Leftrightarrow \dot{x} = b_0 + A_{lin}x + Bu \tag{10.8}$$

The state transition equation, with $\phi = e^{A_{lin}t}$, is now given by:

$$x(t) = \phi(t-t_0)x(t_0) + \int_{t_0}^{t} \phi(t-\tau)Bu(\tau)d\tau + (\phi(t-t_0) - I)\,A^{-1}b_0 \tag{10.9}$$

and its discrete equivalent by:

$$x[(k+1)T] = \phi(T)x(kT) + \varphi(T)u(kT) + c \tag{10.10}$$

with $c = (\phi(T) - I)A^{-1}b_0$ and $\varphi = \int_0^T e^{A_{lin}\sigma}Bd\sigma$. A change of coordinates on the state variable x, $x = \tilde{x} + k$, with k a constant, eliminates the bias term b, resulting a linear system. However, this is only possible if the transition matrices are known, which is not the case. Hence, this change of coordinates is only possible after the learning process. As a consequence, locally affine models will be required, as given by equation 10.10. Thus, for a non-linear system, a collection of locally affine discrete equivalent models are fitted to the data:

$$x_{k+1} = \begin{bmatrix} \phi & \varphi \end{bmatrix} \begin{bmatrix} x_k \\ u_k \end{bmatrix} + c \Leftrightarrow \begin{bmatrix} x_k & u_k & I \end{bmatrix} \Leftrightarrow \begin{bmatrix} \phi \\ \varphi \\ c \end{bmatrix} = x_{k+1} \tag{10.11}$$

Image measurements provide x_{k+1}, x_k and u_k. The matrices ϕ and φ have to be estimated from the input $[\ x_k \quad u_k\ ]^T$ and output x_{k+1}, using locally affine models as described in Appendix ??. The continuous model matrices A_{lin} and B are then obtained using equation 10.12.

$$\begin{aligned} A_{lin} &= ln(\phi^{1/T}) = \tfrac{1}{T} ln(\phi) \\ B &= (\phi - I)^{-1} A\varphi \end{aligned} \tag{10.12}$$

It might be hard to measure the actuator value u from certain actions, like poking. It is still possible to recover some system structure from such cases. The action exerted on the system is dealt as a perturbation of the state, with $u_k = 0$ for all k, and the matrix ϕ is determined from the system natural response. Taking them $x_{k+1} = \phi x_k$, x_k as input and x_{k+1} as output, ϕ is learned using locally affine models.

Time response to a step input

The time response to a step input actuator command of a dynamical system is characterized by a set of parameters. The identification of these parameters adds extra insight into the type of motion of that system, enabling a coarse classification of their motion.

Single-Input-Single-Output (SISO) systems Suppose an object is composed of just one linkage element. The actuator signal may then be composed of just one entity (such like hand position) or multiple elements, such as velocity at instant k and $k-1$. For the former case, we have then a SISO system by selecting one of the state variables as output - for instance, the position of the link centroid. The local description of the state-space is then:

$$\begin{aligned} \dot{x} &= A_{lin} x + bu \\ y &= Cx \end{aligned} \tag{10.13}$$

being y the scalar output. Solving the differential equation, results a second-order system with transfer function of the form

$$G(s) = b \frac{w_n^2}{s^2 + 2\varsigma w_n + w_n^2} \tag{10.14}$$

Both damping ς and natural frequency w_n are determined from A, B and c. Other interesting measures are also extracted, such as:

- time to settle - $t_s = \frac{3}{w_n}$
- percentage of overshoot - $S = 100 e^{-\frac{\varsigma\pi}{\sqrt{1-\varsigma^2}}}$
- period of oscillations - $T_a = \frac{2\pi}{w_n \sqrt{1-\varsigma^2}}$

- rise time $t_c = T_a \frac{\pi - \arccos \varsigma}{2\pi}$

These measures are useful for dynamic system classification. For instance, a system with $\varsigma << 1$ presents strong oscillations, differing significantly of a system with $\varsigma = 1$, which will not oscillate.

Multiple-Input-Multiple-Output (MIMO) systems On the more general case of MIMO systems, we have, for each output and each input, a transfer function. Because of the interconnection, one output can not be described solely as a function of one input, independent of the others. Hence, the analyze just carried on for SISO systems does not bring additional insight. However, there is a measure that is of extreme importance both for MIMO and SISO systems, which is the H_∞ norm of a system, which will be further exploited in this thesis.

Euclidean vs Projective/Affine Spaces

As stated before, the affine space is used as an approximation to the projective space. The 3D Euclidean Space is topologically adequate for the description of a system's dynamics. The same does not hold for projective/affine spaces. The recovered dynamic in a projective space correspond to the perceived projective dynamics. While in most situations this results in useful information to describe actions that may be exerted on an object, this topological space is often not adequate for the control of complex dynamic systems.

10.4 Developmental Learning

Humans are pretty good in understanding visual properties of objects, even without acting on them. However, the competencies required for such perceptual capability are learned developmentally by linking action and perception. Actions are rather useful for an embodied actor, through the use of its own body, to generate autonomously cross-modal percepts (e.g., visual and auditory) for automatic object categorization. This is in contrast with non-embodied techniques such as standard supervised learning requiring manual segmentation of off-line data, imposing thus constraints on an agent's ability to learn.

10.4.1 Human-Robot Skill-Transfer

This manuscript describes an embodied approach for learning task models while simultaneously extracting information about objects. Through social interactions of a robot with an instructor (see figure 10-5), the latter facilitates robot's perception and learning, in the same way as human teachers facilitate children perception and learning during child development phases. The robot will then be able to further develop its action competencies, to learn more about objects, and to act on them using simple actions such as shaking (Arsenio, 2004b).

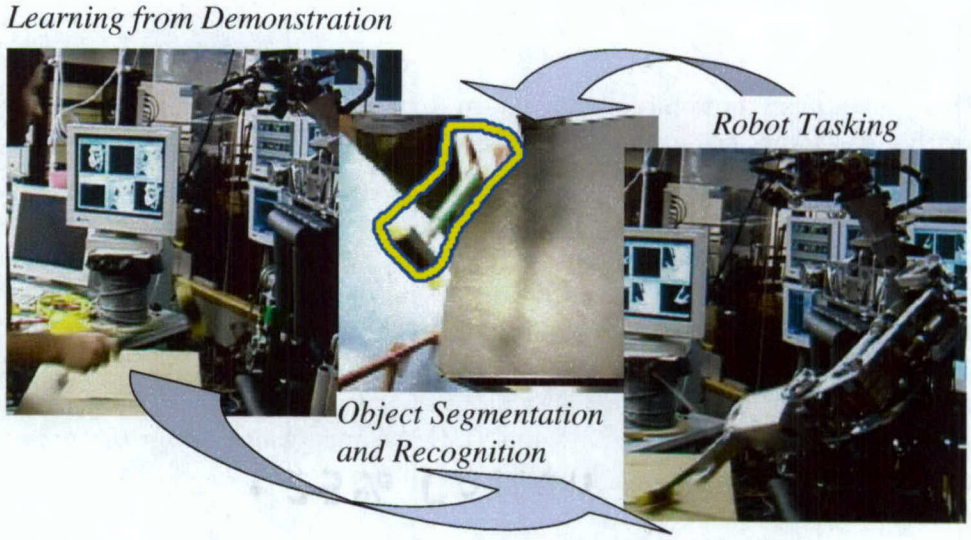


Figure 10-5: At initial stages, object properties (e.g. appearance, size or shape) are unknown to the robot. As a human teacher manipulates objects, the robot build object models and learn the teacher actions. Robot learning may then continue autonomously.

We now show how learning a very simple model (just a machine with one internal state) for the task of hammering on a table enables the robot to generate autonomously informative percepts by itself. Consider again figure 10-5. These images correspond to real experiments. We have shown before how object recognition and robot experimental manipulation evolve developmentally from human demonstration. By transferring the manipulation skill from human to robot, the latter can generate equally training data to the object recognition algorithm, as demonstrated by the experiment in figure 10-6. This figure shows that by having the robot hammering on a table, the perceptual system extracts visual templates of the object which is thereafter recognized as the same object previously segmented from human demonstration.

Figure 10-7 shows the robot executing a learned poking task, while extracting object segmentations for both the robot's grip and the poked objects from such action.

10.4.2 Human-Robot Cooperation

Input from one perceptual modality can also be useful to extract percepts from another perceptual modality. This is corroborated by an experiment (see figure 10-8) consisting of feeding the energy of the acoustic signal into the feedback loop of the neural oscillator, instead of proprioceptive signals. Therefore, the goal is to have the robot to play drums using sound feedback. The task rhythmic is imposed by a human actor, which cooperates with the robot for drumming with sticks. Since it is difficult for the neural oscillator to engage initially in a rhythmic pattern without a coherent source of repetitive sound, the human guides the process by providing such information. While executing the task, the robot is then able to learn the visual appearance

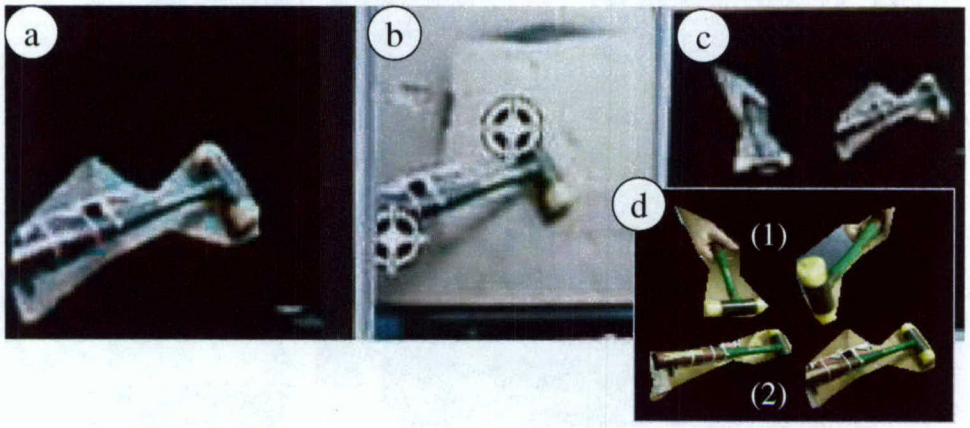


Figure 10-6: Human-robot skill transfer, from an on-line experiment. a) Hammer visual segmentation by having the robot hammering on a table. (b) Tracking multiple objects – the robot grip and the hammer – based on the Lucas-Kanade pyramidal tracker algorithm. (c) It shows two segmentations. One first obtained from human demonstration (on the left). The second (on the right), was segmented from robot actuation, and it was recognized as belonging to the same category as the first (otherwise it would not appear on the same window during the experiment). d) Several segmentations obtained by human demonstration and by the robot's experimental manipulation.

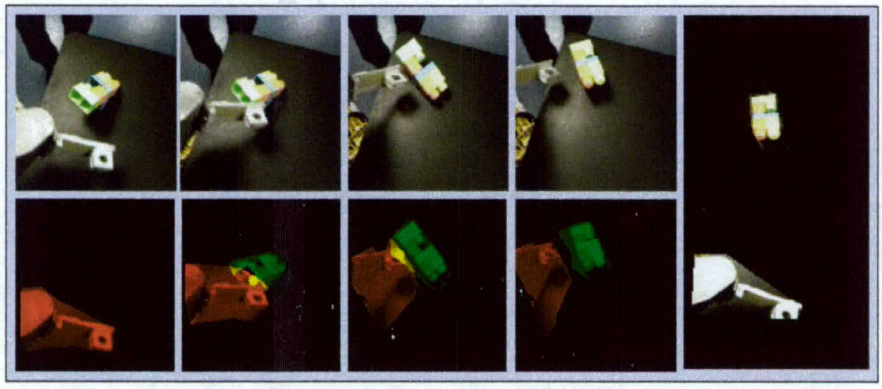


Figure 10-7: Top row: sequence of images of the robotic arm approaching, impacting and releasing and object. This exploratory mechanism enables the extraction of objects' appearance. Notice that two human legs are also moving in the background. Bottom row: tracked objects. A contact occurs at the 2^{nd} frame from the left, upon impact of the robot's arm with the object. Another event (a release) occurs on the 5^{th} frame. Segmentations for the robot's arm and the poked object are also shown.

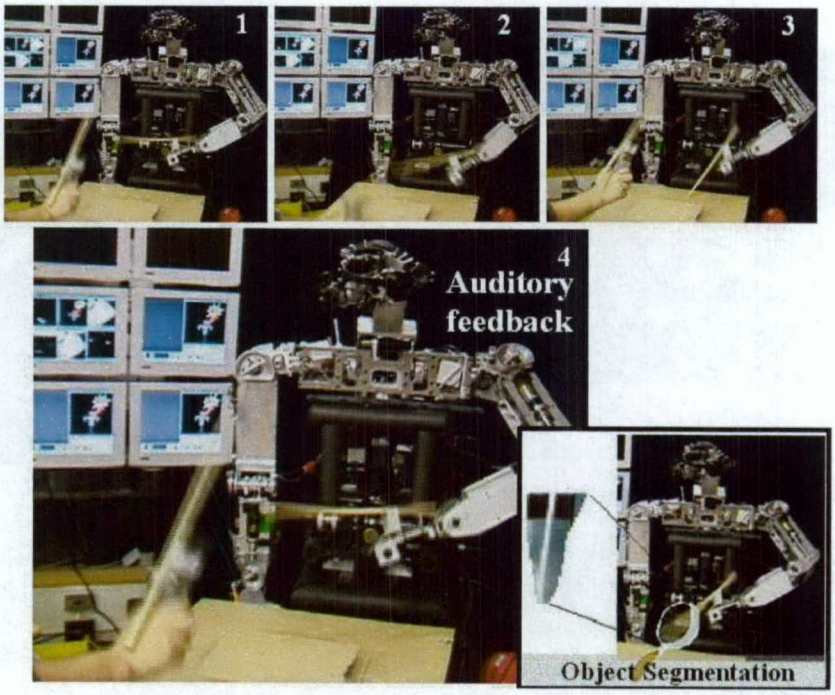


Figure 10-8: Human-Robot cooperation. Playing with a stick for drumming, entraining rhythms provided by a human actor, who drums together with the robot. The neural oscillator receives as feedback signal the acoustic energy. The robot is then able to extract a visual segmentation of the stick.

of the drumming stick (shown in figure 10-8), together with the sound it produces.

10.5 Emotions

A convincing intelligent robot is greatly gauged by its putative social skills.

Limbic system also

ref. to Macaco and ref. to Kismet

Chapter 11

Teaching Humanoid Robots like Children – Developmental Learning

Learning is a matter of extracting meaning from our interactions for use in the future.

... extract valuable information and put it in use in guiding your own action.
(Dennet, 1998)

For an autonomous robot to be capable of developing and adapting to its environment, it needs to be able to learn. The field of machine learning offers many powerful algorithms, but these require training data to operate. Infant development research suggests ways to acquire such training data from simple contexts, and use this experience to bootstrap to more complex contexts. We need to identify situations that enable the robot to temporarily reach beyond its current perceptual abilities, giving the opportunity for development to occur (Arsenio, 2002, 2003a; Fitzpatrick, 2003c). This led us to create children-like learning scenarios for teaching a humanoid robot. These learning experiments are used for the humanoid robot Cog (see figure 11-1) to learn about: object's multiple visual representations from books and other learning aids (section 11.1), education activities such as drawing and painting (section 11.2), auditory representations from all these activities (section 11.3) and musical instruments (section 11.4).

Figure 11-1: Teaching humanoid robots as if they were children, exploring the large arsenal of educational tool and activities widely available to human educators. Caregivers have available a rich set of tools to help a child to develop, such as (1) books; (2) drawing boards; (3) Toys or tools (e.g., hammer); (4) drawing material (e.g. pens or chalk pencils); (5) musical instruments; and (6) TV educational videos or programs, just to name a few. All these learning scenarios, and more, were implemented on the humanoid robot Cog, as shown: (a) an interactive reading scenario between the robot and a human *tutor*; (b) robot learning functional constraints between objects by observing a train moving on a railway track; (c) a human describes by gestures the boundaries of an object; (d) robot recognizing geometric shapes drawn by a human in a drawing board; (e) robot learns about rough textures from the sound they produce; (f) and (g) robot learns from educational activities played with a human, who paints with a ink can or draws with a pen, respectively.

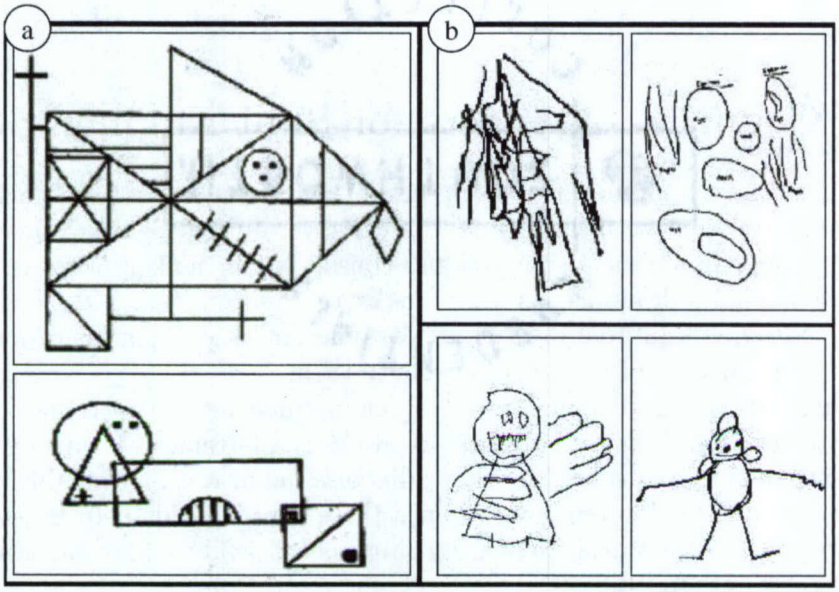


Figure 11-2: a) Rey-Osterrieth drawing tests of two models of different complexity. The child is asked, in separate tasks, to copy the figure while looking at it, and from memory 3 minutes after. Graded features during these task are deformations from figure scaling or stretching, lack of structure reproduction or structure repetition, among others. b) Drawing test of the human figure. Top images correspond to low graded drawings relative to the bottom ones. The pencil pressure, the trace precision, the figure localization on the sheet of paper, the size, are all graded features.

Developmental Issues: According to Mahler's theory, towards the second half of the child *Separation and Individuation* developmental phase, there is a clear separation between objects and the child itself. Play is a very important component of the child growth, which benefits a lot from the caregiver's guidance. The child learns from playing with musical instruments and other toys, such as trains and dolls. And books, so useful to teach adults, are also a very important educational tool for a caregiver to transmit visual information about objects to a child in a multitude of ways. Caregivers employ as well books and other learning aids to start teaching the child the sounds of words associated to object properties. The caregiver therefore facilitates the child learning process by means of scaffolding.

Through play activities such as drawing and painting, the child learns to associate different object representations. The drawing test of the human figure (Goodenough, 1926), as well as the Rey-Osterrieth complex figure test (Rey, 1959) are psychological tests to assess a child developmental stage based on the child drawings (see figure 11-2). Indeed, the level of detail in a drawing activity, together with representation of the figure elements, are closely associated to the child cognitive development. Therefore, it makes sense to use these same activities to guide the robot's learning

process.

11.1 Cognitive Artifacts for Skill Augmentation

Children's learning is often aided by the use of audiovisuals, and especially books, from social interactions with their mother or caregiver during the developmental subphases of re-approximation and individual consolidation, and afterwards. Indeed, humans often paint, draw or just read books to children during their childhood. Books are indeed a useful tool to teach robots different object representations and to communicate properties of unknown objects to them.

Learning aids are also often used by human caregivers to introduce the child to a diverse set of (in)animate objects, exposing the latter to an outside world of colors, forms, shapes and contrasts, that otherwise might not be available to a child (such as images of whales and cows). Since these learning aids help to expand the child's knowledge of the world, they are a potentially useful tool for introducing new informative percepts to a robot.

11.1.1 Teaching a Humanoid Robot from Books

The extraction of visual information from books poses serious challenges to current figure/ground segregation techniques. Current active vision techniques are not appropriate simply because the object image printed into a book page cannot be moved separately from the book page.

Hence, we have developed an innovative human aided object segmentation algorithm for extracting perceptual meaning from books. Indeed, a significant amount of contextual information may be extracted from a periodically moving actuator. This can be framed as the problem of estimating $p(o_n|v_{B_{\vec{p},\epsilon}}, act^{per}_{\vec{p},S})$, the probability of extracting object o_n from a book given a set of local, stationary features v on a neighborhood ball B of radius ϵ centered on location p, and a periodic actuator on such neighborhood with trajectory points in the set $S \subseteq B$ (see figure 11-3). Chapter 4's perceptual grouping from human demonstration algorithm implements this estimation process.

Clusters grouped by a single trajectory might either form or not the smallest compact cover which includes the full visual description of the object on the book (depending on intersecting or not all the clusters that form the object). After the detection of two or more temporally and spatially close trajectories this problem vanishes.

Experimental Results

This strategy relies heavily on human-robot interactions. Figure 11-4 shows qualitative experimental results for one such interaction in which a human introduces visual percepts to the robot from a fabric book, by tapping on relevant figures. These books correspond to the hardest cases to segment, since the fabric material is deformable,

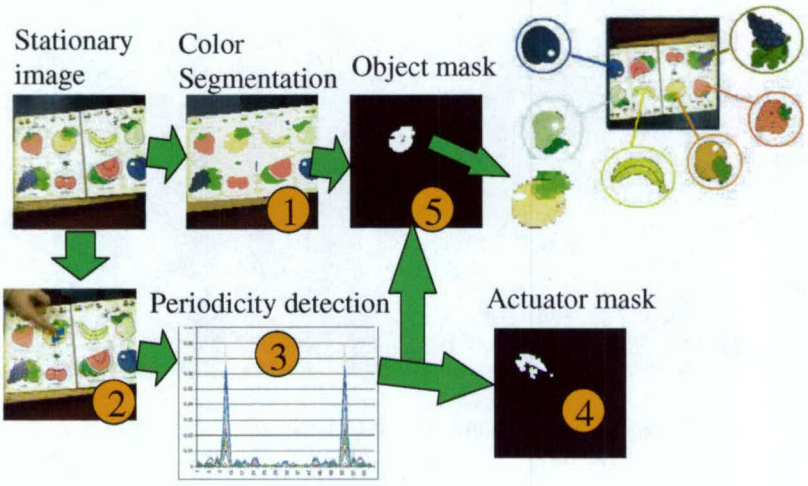


Figure 11-3: A standard color segmentation algorithm computes a compact cover for the image. The actuator's periodic trajectory is used to extract the object's compact cover – a collection of color cluster sets.

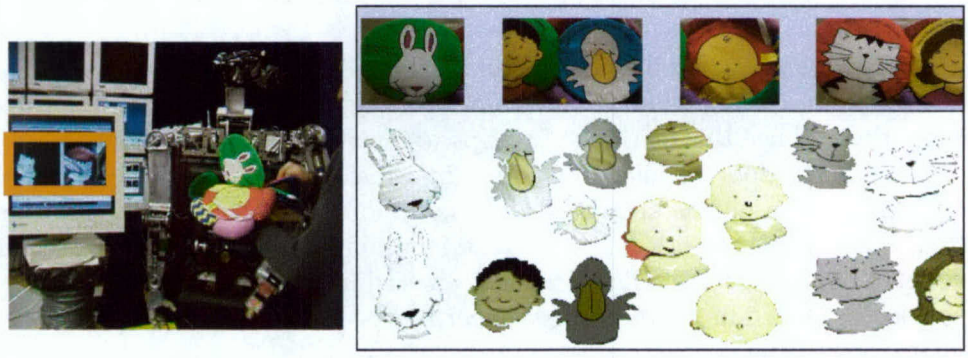


Figure 11-4: (left) Teaching the visual appearance of objects to a robot, by having a human showing a fabric book to the robot, as if it was an infant. (right) Illustration shows segmentation results – on the bottom – of book pages (on top)-from a book made of fabric. Multiple segmentations were acquired from all the book pages. It is worth stressing that segmentation on books made of fabric textile poses additional difficulties, since pages deform easily, creating perspective deformations, shadows and object occlusions.

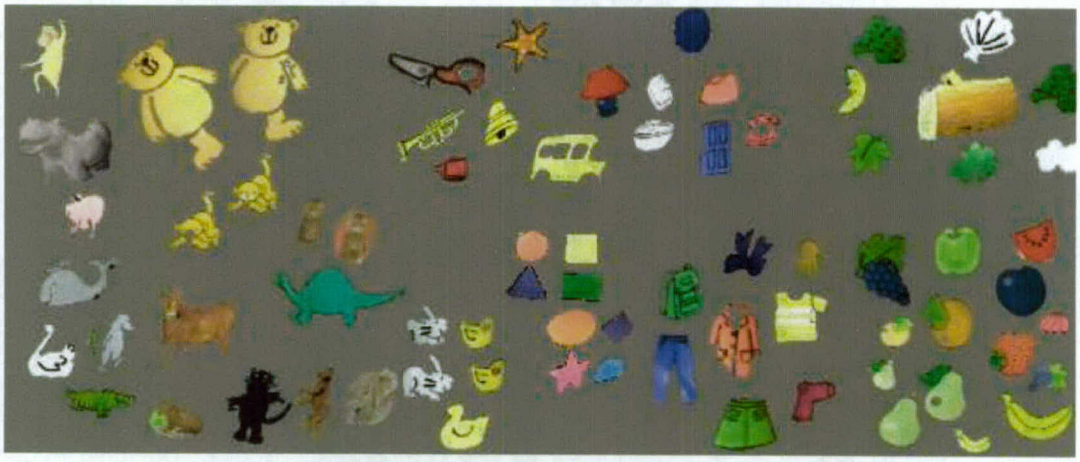


Figure 11-5: Templates for several categories of objects (for which a representative sample is shown), were extracted from dozens of books. No noticeable performance degradation was found from subjects not acquainted with the algorithm interacting with the robot.

often making the visual image appear distorted, occluded or with large portions of shadowy regions. Results for a large sample of animate and inanimate objects are shown in figure 11-5, for books made of paper. Segmentations were obtained for a large spectrum of categories: mammals, reptiles, birds, dinosaurs, plants, atmospheric - cloud - and astronomical - moon, clothes, fruits, geometric shapes, domestic utilities, furniture, among others.

It is indeed essential to have a human in the loop to introduce objects from a book to the robot (as a human caregiver does to a child), by tapping on their book's representations. Since this learning aid helps to expand the child's knowledge of the world, it is a useful tool for introducing new informative percepts to a robot (whales do not appear often on research labs!). To corroborate this argument, the human-centered segmentation algorithm was successfully applied to extract templates for a large variety of animals (including a whale), musical instruments, and other elements from books (as shown in figure 11-6 for paper books), under varying light conditions (no environment setup).

Typical errors might arise from with similar color patterns as the book's background page, for which no perfect differentiation is possible, since the intersection of the object's compact cover of color regions with the book's page cover (complementary to the object) is not empty (see figure 11-7). High color variability within an object create additional grouping difficulties (the object's compact cover of color clusters contains too many sets – harder to group).

Another experiment is presented in figure 11-8, which shows results for segmenting visual descriptions from a foam book, under varying luminosity conditions. The algorithm depends in some extent on luminosity conditions, since different locations for light sources project shades differently. In addition, they may also add or re-

Figure 11-6: Figure/ground segregation from paper books (books targeted for infants and toddlers). Templates for several categories of objects (for which a representative sample is shown), were extracted from dozens of books. The book pages shown were not recorded simultaneously as the segmentations, but they are useful to identify the background from which the object's template was extracted. (top) Clusters of colors were grouped together into an elephant, a piece of wood and a monkey. (middle) A bear and a cow are segmented from a book. These two animals are the union of a large set of local color clusters. (bottom) Segmentation of several elements from books.

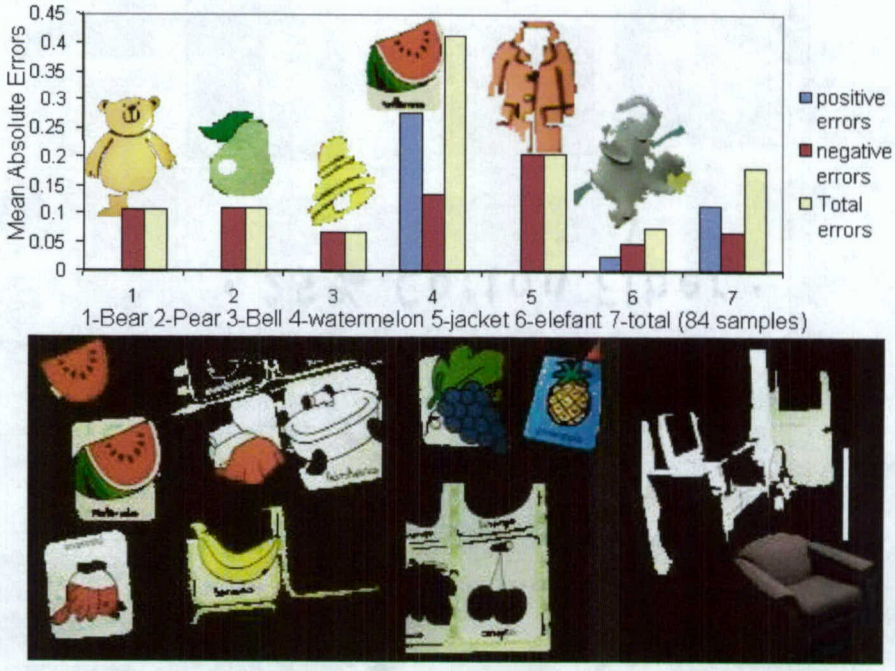


Figure 11-7: Statistical analysis. Errors are given by (template area - object's real visual appearance area)/(real area). Positive errors stand solely for templates with larger area than the real area, while negative errors stand for the inverse. Total errors stand for both errors (e.g., 0.2 corresponds to 20% error rate). The real area values were determined manually. (top) Statistical analysis for segmentation errors from books. (bottom) A representative set of templates illustrating sources of errors. The watermelon, banana, bed and white door have color clusters with identical color – white – to its background, for which no differentiation is possible, since the intersection of the object's compact cover of color regions with the background is not empty. High color variability of the sofa pixels create grouping difficulties (the compact cover contains too many sets – harder to group, unless object is well covered by human hand trajectories). The cherries reflect another source of errors - very small images of objects are hard to segment.

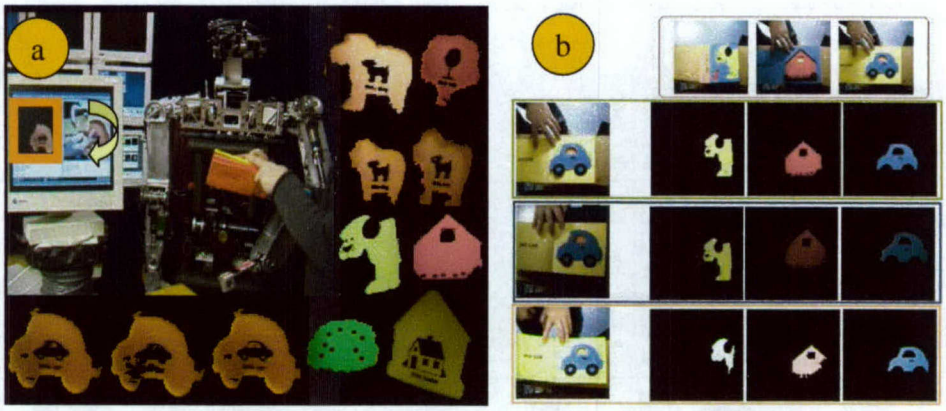


Figure 11-8: a) Human shows elements from a foam book to the robot, and all objects within it were segmented (partial view of yellow dog on cover, dog, car, tree and house in inside pages). Foam parts of the book might be removed, resulting therefore in two different collections of object segmentations b) Three experiments to text variability with light sources. Top images show pages of a book. The other rows show segmentation results for three different luminosity conditions (from top to bottom – middle, low and high luminosity).

move color clusters depending on light sources as well as on the book material. This happens because these variations also affects the performance of the standard color segmentation algorithm.

The qualitative analysis of the results is pretty satisfactory. Object templates with an associated segmentation error of the order of the total error (20% of error rate) are quite good representatives of the object category, retaining most features that uniquely characterize an object. This is exemplified by the various qualitative experimental results from several books, various book materials and varying lightning.

11.1.2 Matching Representations: Drawings, Pictures ...

Object descriptions may came in different formats - drawings, paintings, photos, etc. Therefore, we must be able to relate such variety in descriptions. The approach we follow is grounded on physical percepts, in stark contrast with traditional approaches, such as (Minsky, 1985) philosophy of multiple internal representations. The construction of common sense databases (Lenat, 1995), with an appropriate choice of system representation, offers as well an overlapping traditional view.

Hence, methods were developed to establish the link between an object representation in a book and *real* objects recognized from the surrounding world using the object recognition technique described in Section 5.1, as shown by figure 11-9. Except for a description contained in a book, which was previously segmented, the robot had no other knowledge concerning the visual appearance or shape of such object.

Additional possibilities include linking different object descriptions in a book,

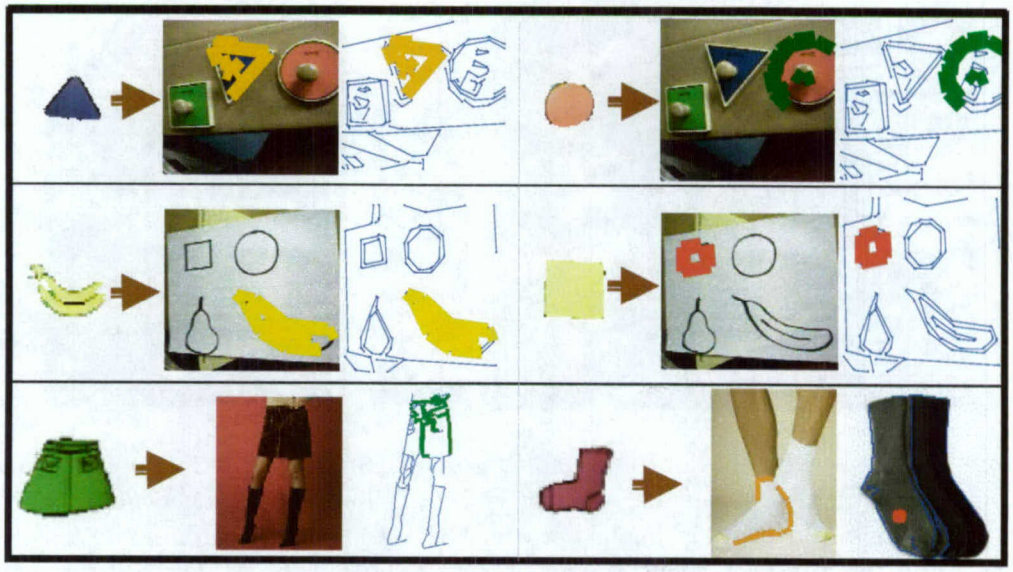


Figure 11-9: (Top) Geometric shapes recognized using the descriptions from a book triangle–left– and circle–right. The recognition from chrominance features is trivial – objects have a single, identical color (Middle) Recognition of geometric, manual drawings from the description of objects learned using books (Bottom) Object descriptions extracted from books are used to recognize the geometric shapes of pictures of objects in catalogues.

such as a drawing, as demonstrated also by results presented in figure 11-9. A sketch of an object contains salient features concerning its shape, and therefore there are advantages to learning, and linking, these different representations. These results demonstrated the advantages of object recognition over independent input features: the topological color regions of a square drawn in black ink are easily distinguished from a yellow square. But they share the same geometric contours.

This framework is also a useful tool for linking other object descriptions in a book, such as a photo, a painting, or a print (figure 11-9). Computer generated objects are yet another feasible description (Arsenio, 2004d).

11.1.3 On the Use of Other Learning Aids

An arsenal of educational tools are used by educators to teach children, helping them to develop. Examples of such tools are toys (such as drawing boards), TV programs (such as the popular *Sesame Street*), or educational videos (such as the video collection *Baby Einstein*).

The *Baby Einstein* collection includes videos to introduce infants and toddlers to colors, music, literature and art. Famous painters and their artistic creations are displayed to children on the *Baby Van Gogh* video, from the same collection. This inspired the design of an experiment in which Cog is introduced to art using an

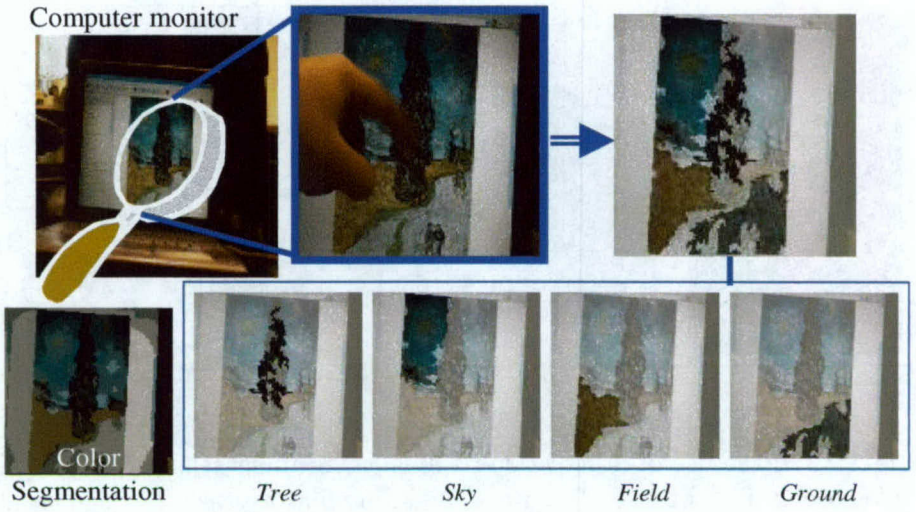


Figure 11-10: Segmenting elements from a painting by Vincent Van Gogh, *Road with Cypress and Star, 1890.* The image is displayed on a computer, from an internet website. A human then taps on several elements to segment them. The bottom row show the segmented elements individually segmented (darker on images): (from left to right) color segmentation of stationary image, and segmentations of the tree, sky, field and ground.

artificial display (the computer monitor). The image of a painting by Vincent Van Gogh, *Road with Cypress and Star, 1890* is displayed on a computer screen. Paintings are contextually different than pictures or photos, since the painter style changes the elements on the figure considerably. Van Gogh, a post-impressionist, painted with an aggressive use of brush strokes, as can be seen in his painting in figure 11-10. But individual painting elements can still be grouped together by having a human actor tapping on their representation in the computer screen to group them together. Figure 11-10 shows results of this experiment for several individual elements: a cypress tree, the sky (with a star), a field and the road.

But educational videos like *Baby Einstein* can have other utilities, such as to present oscillating objects in a display to facilitate their segmentation by the robot. Figure 11-11 shows other experiments using a television to display (which degrades considerably the image resolution) a Cog's video by (Williamson, 1999). Cog is sawing a piece of hood on such video, shaking the whole body in the process. Cog's image is segmented from the screen by applying chapter 4's active figure/ground segregation algorithm to periodically moving points. Indeed, due to the highly interconnectedness nature of this work, several of the implemented segmentation algorithms, together with the object recognition schemes of chapter 5, are extensively used to extract meaningful percepts from learning aids or, as we will see later, from educational and play activities. The same figure also shows active segmentation results for a synthetic ball oscillating on a synthetic background. Drawing boards are also very

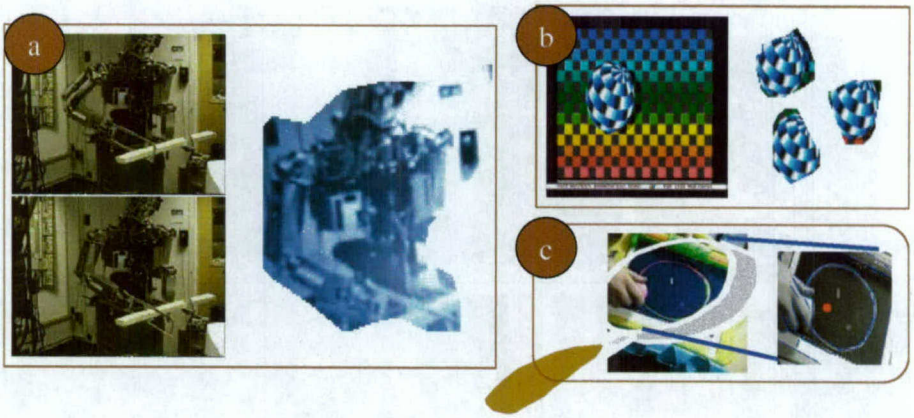


Figure 11-11: On the use of other learning aids. a) Segmentation of Cog's image by having Cog see a video of himself performing an oscillatory task; b) Segmentation of a synthetic ball moving periodically on a synthetic background. Both a) and b) result from applying chapter 4's active figure/ground segregation algorithm to periodically moving points in a display screen; c) Matching representations on a drawing board.

useful to design geometric shapes while interacting with a child, for which figure 11-11 shows a circle drawn being matched to a previously learned circle shape. Matching is implemented by the object recognition technique previously described in Section 5.1.

11.2 Educational and Play Learning Activities

A common pattern of early human-child interactive communication is through activities that stimulate the child's brain, such as drawing or painting. Children are able to extract information from such activities while they are being performed on-line. This capability motivated the implementation of three parallel processes which receive input data from three different sources: from an attentional tracker (Fitzpatrick, 2003b), which tracks the attentional focus and is attracted to a new salient stimulus; from a multi-target tracking algorithm, implemented to track simultaneously multiple targets, as described in chapter 8; and from an algorithm that selectively attends to the human actuator for the extraction of periodic signals from its trajectory. This algorithm operates at temporal, pyramidal levels with a maximum time scale of 16 seconds, according to the following steps:

1. A skin detector extracts skin-tone pixels over a sequence of images
2. A blob detector groups and labels the skin-tone pixels into connected regions
3. Non-periodic blobs are tracked over the time sequence are filtered out
4. A trajectory if formed from the oscillating blob's center of mass over the temporal sequence

Whenever a repetitive trajectory is detected from any of these parallel processes, it is partitioned into a collection of trajectories, each element being of such a collection described by the trajectory points between two zero velocity points with equal sign on a neighborhood (similarly to the partitioning process described in chapter 7 (Fitzpatrick and Arsenio, 2004)). The object recognition algorithm is then applied to extract correlations between these sensory signals perceived from the world and geometric shapes in such world, or in the robot database of previously recognized objects, as follows:

1. Each partition of the repetitive trajectory is mapped into a set of oriented lines by application of the Hough transform

2. By applying the recognition scheme previously described, trajectory lines are matched to oriented edge lines (from a Canny detector) in

   (a) a stationary background,

   (b) objects stored in the robot's object recognition database.

This way, the robot learns object properties not only through cross-modal data correlations, but also by correlating human gestures and information stored in the world structure (such as objects with a geometric shape) or in its own database.

11.2.1 Learning Hand Gestures

Standard hand gesture recognition algorithms require an annotated database of hand gestures, built off-line. Common approaches, such as Space-Time Gestures (Darrel and Pentland, 1993), rely on dynamic programming. (Cutler and Turk, 1998) developed a system for children to interact with lifelike characters and play virtual instruments by classifying optical flow measurements. Other classification techniques include state machines, dynamic time warping or HMMs. We follow a fundamentally different approach, being periodic hand trajectories mapped into geometric descriptions of these objects

1. Objects are recognized on-line, following the geometric hashing based recognition method. Similarity invariants are computed from such training data, and stored in hash tables

2. The trajectory of the periodic hand gestures projected into the retinal image defines a contour image

3. Oriented pairs of lines are fitted to such contour

4. Similarity invariants computed from these pairs are then matched to the similarity invariants defined by pairs of lines stored in the hash tables

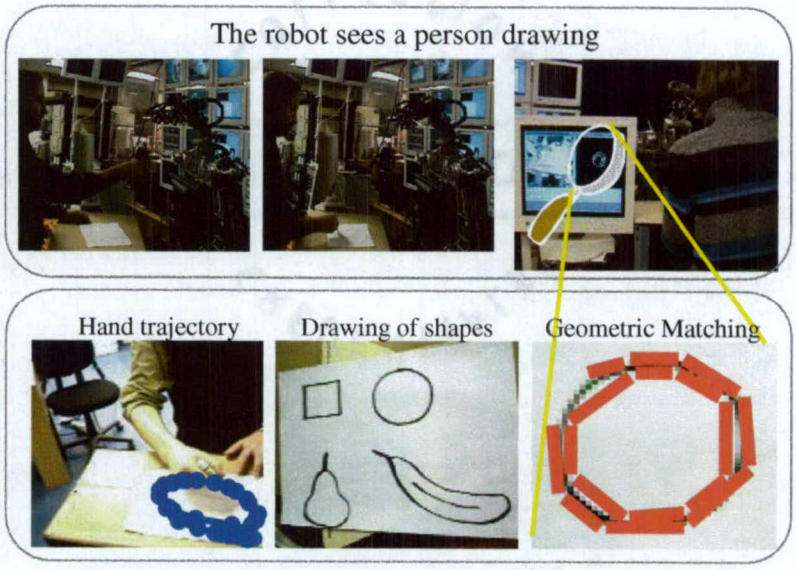


Figure 11-12: (Top) A human draws a circle on a sheet of paper with a pen. (Bottom) The hand circular trajectory is matched to another circle previously recognized and stored (see figure 11-9).

Figure 11-12 reports an experiment in which a human draws repetitively a geometric shape on a sheet of paper with a pen. The robot learns what was drawn by matching one period of the hand gesture to the previously learned shape (the hand gesture is recognized as circular). Hence, the geometry of periodic hand trajectories are on-line recognized to the geometry of objects in an object database, instead of being mapped to a database of annotated gestures.

11.2.2 Object Recognition from Hand Gestures

The problem of recognizing objects in a scene can be framed as the dual version of the hand gestures recognition problem. Instead of using previously learned object geometries to recognize hand gestures, hand gestures' trajectories are now applied to recover the geometric shape (defined by a set of lines) and appearance (given by an image template enclosing such lines) of a scene object (as seen by the robot):

1. The algorithm first detects oriented pairs of lines in a image of a world scene
2. The geometry of periodic hand gestures is used to build a contour image
3. The world image is masked by a dilated binary mask which encloses the arm trajectory on the contour image
4. Oriented pairs of lines fitted to the contour image are then matched to the pairs of lines on the world image through the object recognition procedure.

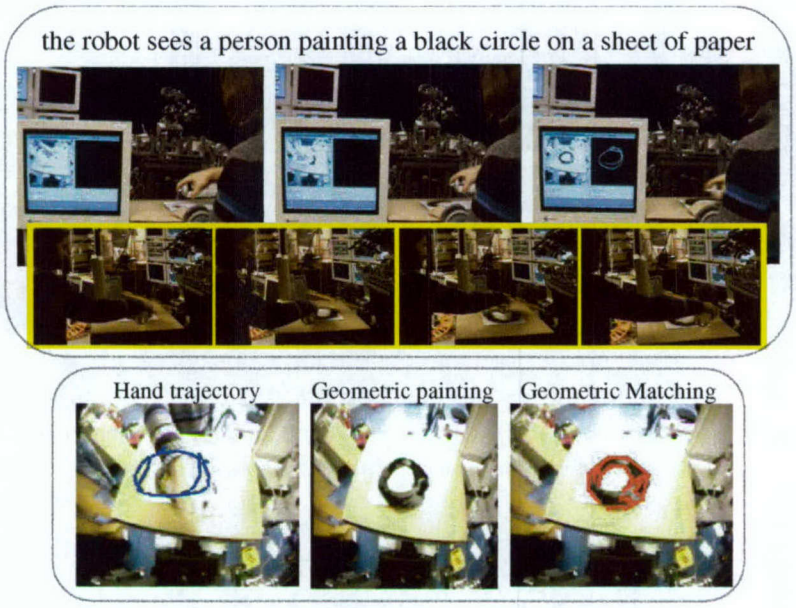


Figure 11-13: (Top) Human paints a black circle with a ink can on a sheet of paper (two different views of an experiment running on Cog are shown). The circle is painted multiple times. (Bottom) The 1^{st} image from the left displays the hand trajectory, the 2^{nd} image shows the geometric circle drawn, and the last shows edges of the painted circle which were matched to the hand trajectory.

Hence, visual geometries in a scene (such as circles) are recognized as such from hand gestures having the same geometry (as is the case of circular gestures). Figure 11-13 shows results for such task. The robot learns what was painted by matching the hand gesture to the shape defined by the ink on the paper. This algorithm is useful to identify shapes from drawing, painting or other educational activities.

Shape from Human Cues

This same framework is applied to extract as well object boundaries from human cues. Indeed, human manipulation provides the robot with extra perceptual information concerning objects, by actively describing (using human arm/hand/finger trajectories) object contours or the hollow parts of objects, such as a cup (figure 11-14). Tactile perception of objects from the robot grasping activities has been actively pursued (Rao et al., 1989). Although more precise, these techniques require hybrid position/force control of the robot's manipulator end-effector so as not to damage or break objects.

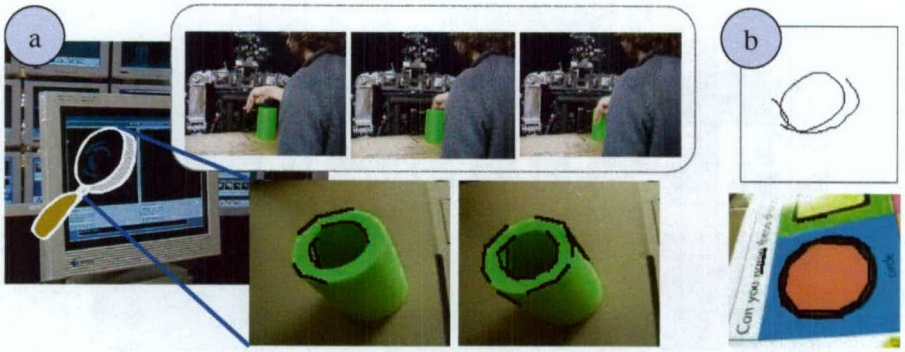


Figure 11-14: a) A human moves his hand around an object boundary (top). The contour image extracted from the trajectory is then matched to the object contours. b) Some procedure applied to a geometric shape on a book.)

11.2.3 Functional Constraints

Not only hand gestures can be used to detect interesting geometric shapes in the world as seen by the robot. For instance, certain toys, such as trains, move periodically on rail tracks, with a functional constraint fixed both in time and space. Therefore, one might obtain information concerning the rail tracks by observing the train's visual trajectory. To accomplish such goal, objects are visually tracked by an attentional tracker (Fitzpatrick, 2003b) which is modulated by the attentional system (chapter 4). The algorithm developed and here introduced starts by masking the input world image to regions inside the moving object's visual trajectory (or outside but near the boundary). Lines modelling the object's trajectory are then mapped into lines fitting the scene edges. The output is the geometry of the stationary object which is imposing the functional constraint on the moving object.

Figure 11-15 shows experimental results for the specific case of extracting templates for train rail tracks from the train's motion (which is constrained by the railway circular geometry). Three such experiments were taken over a total time of 20 minutes. We opted by a conservative algorithm: only 8 recognitions were extracted from all the experiments (out of 200). Two of them originated poor quality segmentations. But the algorithm is robust to errors, since no false results were reported. Extracting a larger number of templates only depends on running the experiments for longer – or letting Cog play for some more time. The same applied for the other algorithms described in this section: tested under similar conditions, they originated recognition results of comparable performance.

11.3 Learning the First Words

Speech is a very flexible communication protocol. Current speech work in robotics includes the development of vocabulary and/or grammar from experience (Steels, 1996; Roy, 1999). We are interested in importing for robots some of the special activities

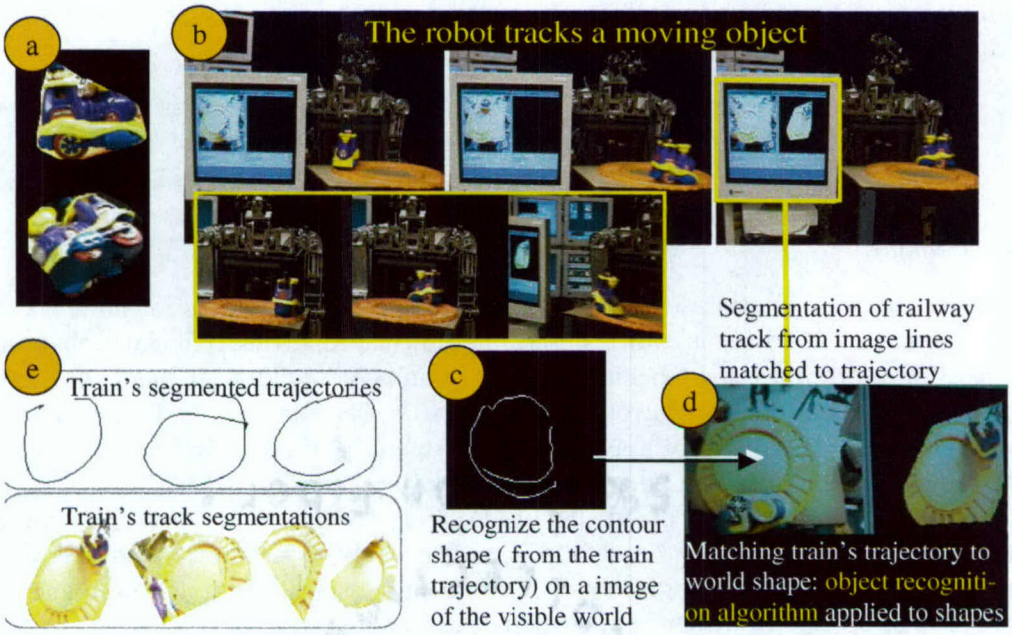


Figure 11-15: Mapping Object Trajectories to World Geometric Shapes. a) Segmentation templates for the train, obtained by applying the active segmentation algorithm of section 4.2 to the images of a train being waved by a human actor. The problem now faced is fundamentally different: we are interested in acquiring information about the railway tracks from the constrain that they impose on the train's motion. b) Two different views of a toy train moving periodically around a railway track (sequences of three images) c) Trajectory extracted from one period of the train's motion d) Cog's view of the train on the rail (left), and railway tracks segmentation recovered from the lines matched e) Some result samples for train trajectories and rail tracks' segmentations.

adults use to teach children (for instance, enunciating the name of objects in books). In addition, we want to borrow as well some special forms of speech that caregivers often apply to simplify the child perception and understanding of simple words. One example of such a special form is the "motherese" style of speech, implemented by our group for catalyzing robot Kismet's social interactions (Varchavskaia et al., 2001).

Experimental evidence (Brent and Siskind, 2001) supports the claim that isolated words in infant-directed speech facilitate word acquisition early in the infant's childhood, providing reliable information for such learning. Caregivers often try to facilitate the child's perception of utterances, by making them short, with long pauses between words (Werker et al., 1996), and often repeating such words several times, to stress the information to be transmitted. And there is evidence that caregivers use a significant number of isolated words for communicating with infants (Aslin et al., 1996; Brent and Siskind, 2001).

It is interesting that most of the speech input for these experiments comes from

having caregivers reading from books, caregivers speaking on videos or even both, as in the case reported by (Yu et al., 2003), and here transcripted:

> *Subjects were exposed to the language by video. In the video, a person was reading the picture book of "I went walking" [by Williams and Vivas] in Mandarin. The book is for 1-3 year old children, and the story is about a young child that goes for a walk and encounters several familiar friendly animals...*

Hence, books and other learning aids are really well suited tools for human caregivers to teach a child about words describing object properties. We claim that they are as well appropriate for teaching words to humanoid robots. It has been shown how a human caregiver can introduce a robot to a rich plethora of visual information concerning objects visual appearance and shape. But cognitive artifacts, which enhance perception, can also be applied to improve perception over other perceptual modalities, such as auditory processing. Indeed, chapters 8 and 7 introduced several contexts to generate training data for a sound recognition scheme. Therefore, auditory processing was also integrated with visual processing to extract the name and properties of objects from books and other learning aids.

The methodology is as follows. A caregiver taps either on top of objects in a book or in a computer display screen, while directing simultaneously verbal utterances towards the robot. The robot process the auditory signal for repetitive sounds. Hence, for the caregiver to be able to transmit the robot useful information, its direct speech should consist of repeated, isolated words, or else words which, although not isolated, appear locally periodic in time with a high frequency of occurrence. The sound of words describing specific object properties is isolated according to Section 7.1's segmentation method, and bound to the object's visual percept.

However, hand visual trajectory properties and sound properties are independent, since it is not the hand that generates sound while tapping on books, but the caregiver. Therefore, cross-modal events are associated together under a weak requirement: visual segmentations from periodic signals and sound segmentations are bound together if occurring temporally close, in contrast to chapter 7's strong additional requirement of matching frequencies of oscillation. Both visual and sound templates are then sent to the respective recognition modules, together with IDs identifying their cross-modal linkage. Obviously, no information is sent to the cross-modal recognizer, since there are no relevant cross-modal features extracted from this processing.

11.3.1 Results and Discussion

Figures 11-16 and 11-17 present results for sound segmentation and sound/visual linking from real-time, on-line experiments on the humanoid robot Cog. The three intra-category acoustic patterns are similar, but differ considerably between the two experiments (inter-category). Hence, such sound segmentations provide good data for the sound recognition algorithm (Section 8.1), as desired.

Additional experimental results for naming three different objects are shown in figure 11-18. The spectrograms obtained from sound segmentation differ as well

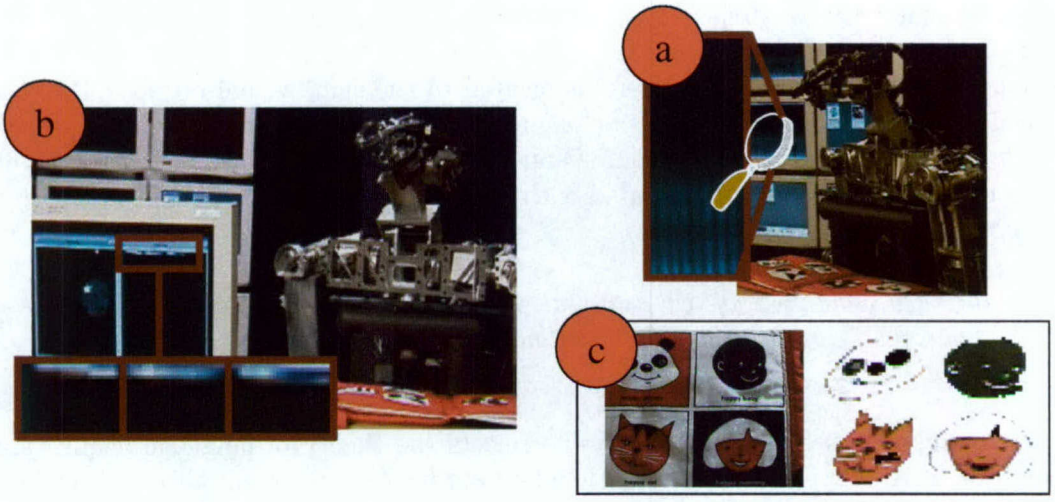


Figure 11-16: a) Spectrogram of environment sounds heard by Cog. During this experiment, periodic sounds of words were being pronounced by a human actor, explaining the oscillatory pattern in the image. b) Sound segmented and associated with an object. It shows low resolution sound spectrograms resulting from an averaging process (as in chapter 8). It is worth noticing the pattern similarity among the three samples corresponding to the same object property, as desired. c) Other visual segmentations extracted from the book's fabric pages.

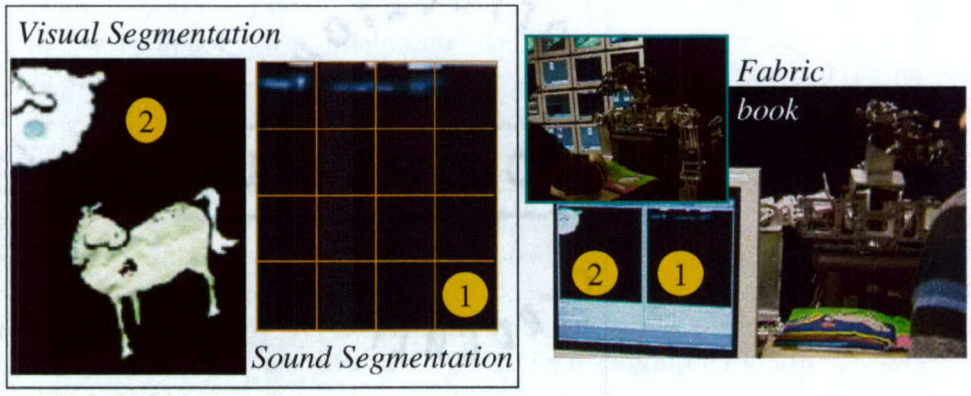


Figure 11-17: Another experiment for extracting a word for a white sheep (word *sheep*). (1) shows sound segmentations. Notice how different the sound pattern is from the one in figure 11-16. Such distinction is what is needed for reliable sound recognition (Section 8.1); (2) Visual segmentations of the sheep and a horse from this fabric book.

considerably for the words associated to the three objects. Over several experiments, we used many words often associated to object categories, such as *"dog"* for the dog's species, *"whoof-whoof"* for the dog's barking, or *"cão"*, which stands for dog in Portuguese. Since these words were associated to the same visual object – the dog visual template – they are dealt as belonging to the same category. Of course, we are left with Quines *"gavagai"* problem (Quine, 1960) – the utterance *"gavagai"* might have many meanings when correlated with a visual event – e.g., a rabbit passing by. As posed by (Tomasello, 1997):

> *the very same piece of real estate may be called: "the shore" (by a sailor), "the coast" (by a hiker), "the ground" (by a skydiver), and "the beach" (by a sunbather).*

Because of this ambiguity, Tomasello rejects the theory of language acquisition based on word mapped onto world. He adopts instead an experientialist and conceptualist view of language. He argues that human beings use linguistic symbols as a vehicle for inviting others to experience a shared perspective of the world.

An opposing view is proposed by (Markman, 1989), which advocates infants create word to meaning maps by basically applying a set of three constraints: 1) The whole-object hypothesis assumes that caregivers utterances directed at an object refer to the whole object and not a part of it; 2) the taxonomic hypothesis consists of grouping meanings according to "natural categories" instead of thematic relationships; 3) the mutual exclusivity principle assumes objects have only one word label – new words are mapped solely onto unnamed objects.

These three constraints should be viewed as biases on search for creating fast mappings from a few words, instead of fixed rules. Our work follows in some sense both approaches. As referred, we rely indeed on the creation of word to world mappings as the grounding process. But such constructs do not follow closely neither the mutual exclusivity nor the taxonomic assumption. In addition, they are also the result of both the robot and a caregiver sharing attention to a particular task, such as reading a picture book. This is in accordance to Tomasello, for whom language should be framed in the context of others joint referential activity, as is the case of shared attention.

Cross-modal association as described in this chapter does not impose rigid synchrony constraints over the visual and auditory percepts for matching. Such constraint is relaxed to events occurring within the same time window between 2-4 seconds. However, Tomasello argues against children using such simple word learning rules based on synchronous visual-acoustic cross modal events, or variants of it.

The approach for learning through activity described by (Fitzpatrick, 2003a) follows closely Tomasello theory, which argues for a "social-pragmatic" model of language acquisition. Fitzpatrick work could benefit from this thesis' approach for building early vocal vocabularies. Our work could also be extended to learn about actions, by correlating sounds of repetitive words to repetitive actions such as waving or tapping. But once more ambiguity would be pervasive (between action and object). Other improvements would arise from correlating acoustic discontinuous events with

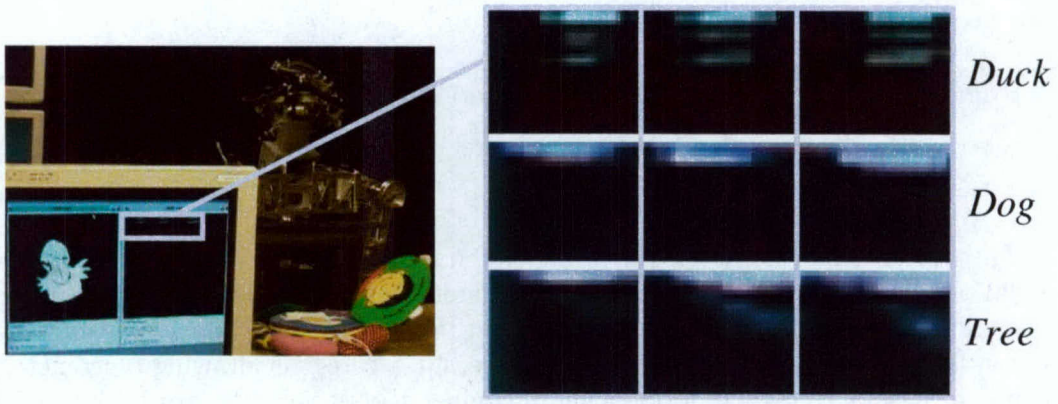


Figure 11-18: Sounds of words: caregiver pronounces the sounds *Duck, "Whoof-Whoof" (dog barking) and tree.* Isolated words are alternated with less frequent words, such as adjectives (e.g. green tree).

the visual discontinuous events as detected in chapter 3. This would be useful to learn vocabulary about actions such as throwing or poking, or about relationships such as an assemblage of objects. Indeed, many components will be needed to solve the complex language puzzle.

This thesis' approach for building a word vocabulary to a humanoid robot from learning aids is simple. Simple in the sense that it is focused solely on learning isolated words – the robot's first words. But such words are grounded in the world through rich experiences. Acoustic patterns associated to words are stored by the sound recognition algorithm, and a link is created to the visual recognizer, by storing a visual object identifier code (ID) on the sound recognizer, and the word sound ID on the visual recognizer. Due to ambiguity, an object in both sound and visual recognizers might have a set of template/word IDs associated to it. Therefore, the sound is grounded to a visual description of the object. Whenever such description feeds the object recognition scheme based on contextual cues, words become also grounded by contextual features of the environment because of transitivity.

11.3.2 Verbal Utterances, Gestures and Object Motions

It is hard for infants to build their early vocal vocabularies even after being exposed to one or a few more symbolic words. Caregivers or children body movements, such as hand movements, often transmit their referential intents in verbal utterances. This use of body movements is denoted by (Yu et al., 2003) as *Embodied Intention*, which it is claimed could play an important role in early language development.

Whenever infants successfully recognize, understand and use a body gesture before they can say the corresponding word, they are revealing that most of the structure required to learn such a word is already in place. Even if these children are not able to clearly articulate utterances (Acredolo et al., 1999), for such use of a body gesture

they need already to be able to:

- ▷ understand the concept of gesture categories
- ▷ recognize the equivalence between a caregiver's sound and the correspondent gesture.

These studies in child development motivated the application of the "weak" cross-modal association method to learn words which refer to both gestures or the geometric shape described by a (repetitive) gesture. And to learn gestures which transmit the referential intent of a verbal utterance. Hence, for a caregiver drawing a geometric shape, such as a triangular form, while including repetitively the word triangle in his speech, the robot extracts the word sound and links it to the caregiver's periodic gesture corresponding to such geometric shape.

11.4 The Robot's first Musical Tones

Experiments by (Hernandez-Reif and Bahrick, 2001) give evidence that an amodal relation (in this case texture, which is common to visual and tactile sensing) provides a basis for learning arbitrary relations between modality-specific properties (in this case the particular colored surface of a textured object). This motivated the development of a new strategy to extract image textures from visual-sound patterns, i.e., by processing acoustic *textures* (the sound signatures) between visual trajectory peaks. The original algorithm here proposed works by having a human probe the world for binding visual and acoustic textures, as follows:

1. Human plays rhythmic sounds on a textured surface.
2. Hand tracking of periodic gestures using the procedure applied in the previous section to learn from educational activities, that selectively attends to the human actuator for the extraction of periodic signals from its trajectory.
3. Tracking and mapping of the x and y human hand visual trajectories (horizontal and vertical directions in images, respectively) into coordinates along eigenvectors given by the Singular Value Decomposition, resulting in new axes x_1, y_1 (x_1 corresponding to the eigenvector along highest data variance). Three measures are then estimated:
   - The angle β of axis x_1 relative to x.
   - The visual trajectory period (after trajectory smoothing to reduce noise) by applying periodicity detection along the x_1 direction – as described in chapter 7.
   - The amplitude difference A_v between the maximum trajectory value along x_1 and the minimum value, over one visual period.

4. Partition of the acoustic signal according to the visual periodicity, and sound periodic detection applied over multiple scales on such a window (as described previously in this thesis). The goal is to estimate the spatial frequency F_s of the object's texture in the image (with the highest energy over the spectrum). This is done by computing the number n of acoustic periods during one (or half) visual period. The spatial frequency estimate is then given by $F_s = A_v/n$, which means that the sound peaks n times from a visual minimum location to a maximum (the human is producing sounds with n peaks of energy along the hand trajectory).

5. Spectral processing of a stationary image by applying to each image point a 1-dimensional STFT along the direction of maximum variation, with length given by the trajectory amplitude and window centered on that point, and storing for such point the energy of the F_s component of this transform. This energy image is converted to binary by a threshold given as a percentage of the maximum energy (or a lower bound, whichever is higher). The object is then segmented by applying this mask to the stationary image.

All these steps are applied in one experiment in figure 11-19. It shows a human hand playing rhythmic sounds with a textured metal piece (figure 11-19-a), producing sound chiefly along the vertical direction of motion. The x and y (horizontal and vertical directions in images, respectively) visual trajectories of the human hand are tracked during a period of approximately 4 seconds (128 frames). The axis x_1 is at an angle of $\beta = 100.92^o$ with the x axis for the experiment shown. Periodic detection along x_1 (after smoothing to reduce noise) estimates a visual period of 1.28 seconds. The visual trajectory's amplitude difference A_v is 78 pixels over one visual period (figure 11-19-b). Sound periodic detection is applied on the visually segmented acoustic signal over 2 scales of temporal resolution. For this experiment, the ratio between half the visual period and the sound period is $n \simeq 5$ (figure 11-19-c).

The sound presents therefore 5 peaks of energy along the hand trajectory, which corresponds to a frequency of $F_s = 16Hz$. The stationary image in figure 11-19-d is processed by selecting 16Hz components of the STFTs, resulting an energy image – middle – which masks the texture which produced such sound. Notice curiously that since this algorithm ran in parallel with other algorithms, the metal object's appearance was also obtained in parallel by the segmentation by demonstration strategy. Child toys like xylophones have a similar structure to the metal piece, which motivated this experiment.

It is also worth stressing however that this approach could also be applied by replacing sound with proprioceptive or tactile sensing, and the human action by robotic manipulation. Indeed, for the proprioceptive case, the visual system would have to detect visual trajectories of the robot finger, while detecting oscillations on the robot's finger joint over a period of the robot's finger visual trajectory.

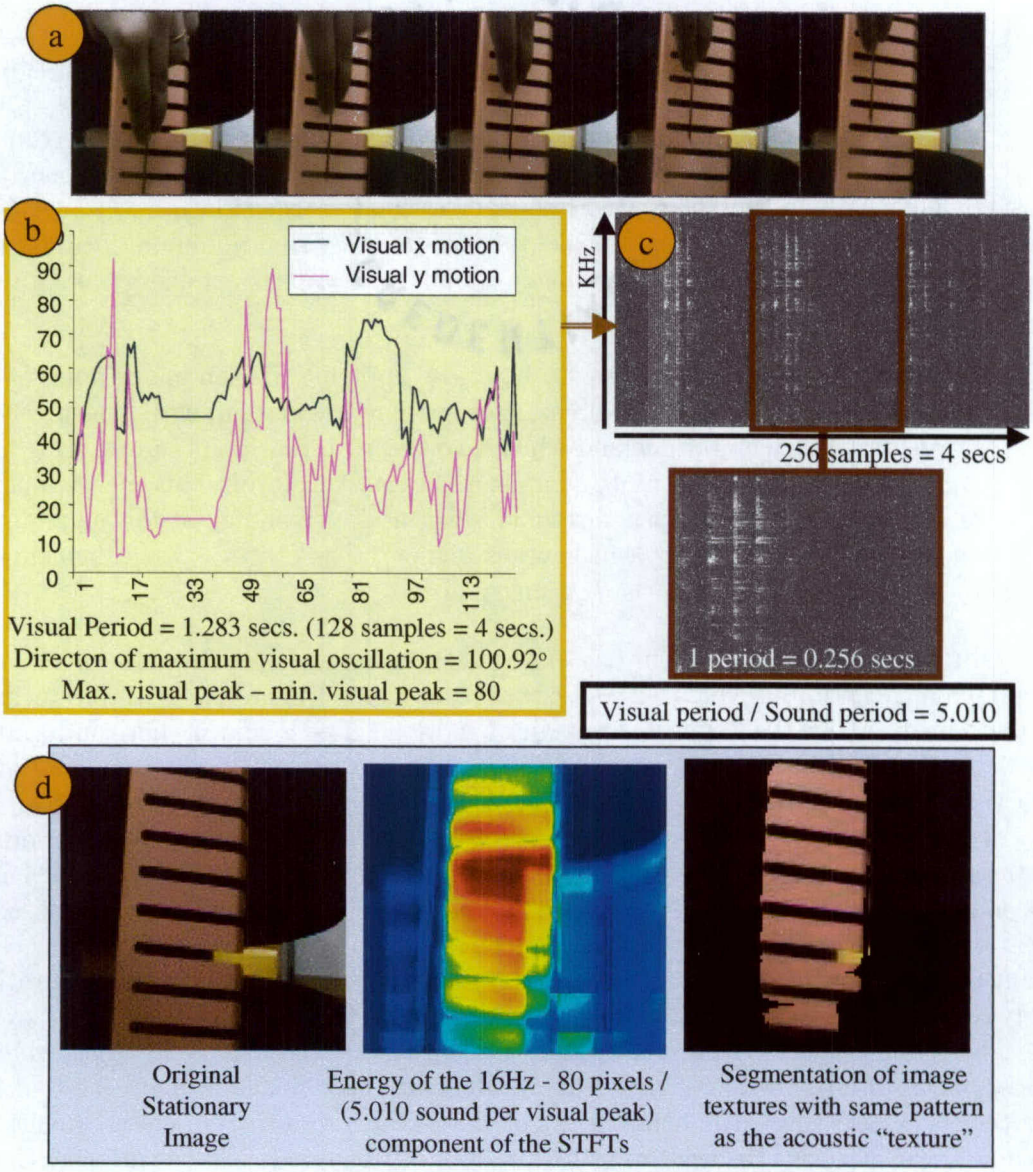


Figure 11-19: Matching visual/acoustic textures to visual textures. a) Sequence of images showing a human hand playing rhythmic sounds with a textured metal piece. Sound is only produced along one direction of motion. b) Horizontal and vertical visual trajectories of the human hand, during a time window of approximately 4 seconds (128 frames). The visual period is estimated as 1.28 seconds. The amplitude difference between the maximum and minimum trajectory values is 78 pixels. Maximum variation makes a 100.92^o angle with the horizontal. c) Ratio between half the visual period and the sound period is $\simeq 5$, which means that sound peaks five times from a visual minimum location to a maximum. d) Stationary image – left – is segmented using a mask – middle – computed from the $16Hz$ energy component of the STFTs applied at each point, selecting the relevant object's texture – right.

Chapter 12

Toward Infant-like Humanoid Robots

I believe myself and my children all to be mere machines.
But this is not how I treat them.

(Brooks, 2002)

If machines are to achieve human-level intelligence, the way humans interact with them has to change. We will have to stop treating *Machines* as machines. In other words, we need to change our current definition of machines. If we want them to become us (and as clever as, or more than us), we need to start programming machines, and treating them, first as if they were children and later like human adults.

Although there is much to do to develop cognitive capabilities for a humanoid robot, from a practical perspective a lot has already been accomplished. Lets consider again figures 5-8 and 7-12. Figure 7-12 shows a partial snapshot of the robot's state during one of the experiments described in this thesis. The robot's experience of an event is rich, with many visual and acoustic segmentations generated as the event continues, relevant prior segmentations recalled using object recognition, the relationship between data from different senses detected and stored, and objects tracked to be further used by statistical learning processes for object location/identification from contextual features. We believe that this kind of experience will form one important part of a perceptual toolbox for autonomous development, where many very good ideas have been hampered by the difficulties of robust perception, control and cognitive integration.

12.1 Motivation

The motivation behind this thesis work is the creation of a 2-year-old-infant-like humanoid robot. However, the technology necessary to achieve such a goal is not yet available, both at the computational hardware and algorithmic levels. The former limitation will probably disappear in the next few years, according to Moore's Law. This thesis aims at developing a core framework towards solving the latter technological limitation. We draw inspiration from children for such, taking very seriously the idea of teaching and treating humanoid robots as if they were children.

Playing with toys, reading books, and other educational and play activities are important for child development. The child's caregiver plays an essential role for guiding the child through the learning process. This thesis applies the same principles to teach a humanoid robot, importing both a child's plethora of learning aids and tools and the caregiver's facilitating role to develop cognitive capabilities for a humanoid robot.

The goal is to build Intelligent Creatures, instead of just an industrial automaton.

12.2 Main Contributions

This thesis main contributions to the field of artificial intelligence are separated into conceptual and technological ones.

12.2.1 Conceptual Contributions

The goal of creating a 2-year-old-infant-like humanoid robot requires innovative ideas, new ways to formulate and approach research problems. This is a very interdisciplinary area. Biology, Cognitive and Neurosciences, as well as Developmental Psychology, are good fields to look for inspiration. But extrapolating lessons from nature to a humanoid robot is a hard scientific problem. At the Humanoid Robotics Group of the MIT AILab, later the MIT CSAIL Living Machines Group, we have been trying to push for new concepts to discover what it takes to build an intelligent robot.

More specifically, this thesis' conceptual contributions include new approaches to tackle research issues using new original tools or else of-the-shelf tools to prove the concept behind the approach. Also included as conceptual innovation is the large scope of applications for a given technique, or the large range of problems it solves.

Development of an embodied and Situated Artificial System: A new Artificial Intelligence philosophy (Brooks, 1991b; Brooks et al., 1998) towards intelligent robots led the field to investigate many previously unexplored problems. Brooks suggested two important concepts - embodiment and situatedness - which were explored extensively on this thesis.

This thesis aims to show that both humans and robots - embodied creatures - can physically explore the environment. Furthermore, robots can boost their learning capabilities both by acting on the environment or by observing other person's actions.

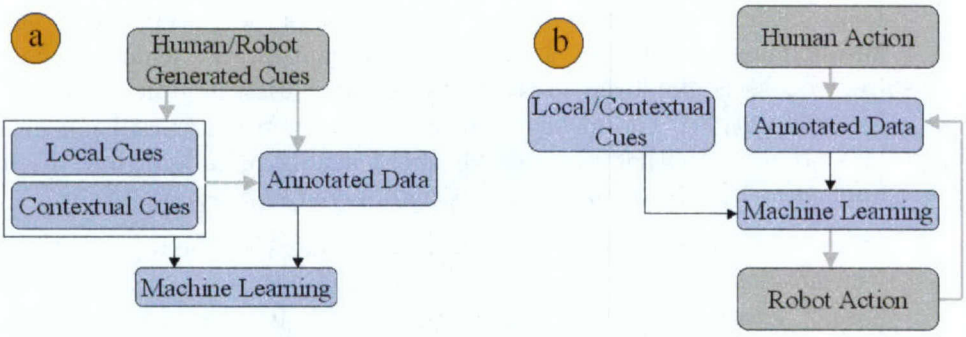


Figure 12-1: New paradigms for machine learning. Developmental learning occurs along two different paths. a) Learning paradigm for learning from books b) Learning paradigm for learning new actions for a robot. The lighter modules and darker arrows correspond to the standard supervised learning paradigm.

Integration of complex cognitive functions: The complex integration of cognitive functions in an embodied and situated artificial system was demonstrated along two alternative and complementary views: cognitive and developmental. These two views are highly interrelated, since complex cognitive functions are acquired and integrated developmentally.

Teaching a Robot like a Child: Teaching robots as if they were babies exploiting humans as caregivers has been already the focus of previous research work (Metta, 2000; Kozima and Yano, 2001; Breazeal, 2000). But this thesis explores a large range of applications in which the humanoid robot Cog is taught as a child by a human. Indeed, almost all the algorithms presented to emulate cognitive capabilities on the robot exploit human-robot interactions. In addition, such development is inspired both by Vygotsky and Margaret Mahler's developmental theories. Several developmental milestones predicted by Mahler's theory are implemented on Cog.

A different paradigm for machine learning: Annotated data is generated from human-robot interactions, which developmentally enable the robot to generate by itself such data or to automatically compute it. As illustrated by figure 12-1, developmental learning occurs along two different levels:

a) at the level of actively applying previous learned knowledge to get new percepts. This is the type of learning occurring for learning object properties from books or to learn about scenes. Standard techniques assume the off-line availability of training data, consisting of either image local or contextual features, together with the annotation of such cues. This data is used for off-line training of a classifier. Our approaches employ instead help from a human caregiver (or actions by the robot itself) for both the on-line generation and annotation of the training data, which is used to train a classifier.

Learning from books exemplifies this learning paradigm applied to extract autonomously local cues. The object visual appearance which is extracted from books is employed to train object recognition modules, without off-line annotation. The localization of objects from contextual information in a scene is another example for this learning paradigm, but this time applied for the autonomous acquisition of contextual cues.

b) at the level of learning how to do a task, and then being able to do it. New actions for a robot, such as waving an object, are first learned from a human actor, and then applied for the robot to generate learning percepts on its own.

Broad Learning Scope: Several learning tools, such as Receptive Field Linear Networks, Weighted Gaussian Clustering, Nearest Neighbor, Hybrid Markov Chains, Hash Tables or Principal Component Analysis, are extensively applied to acquire categorical information about actions, scenes, objects, people and the robot itself.

A new complex approach to object recognition: Objects may have various meanings in different contexts - a rod is labelled as a pendulum if oscillating with a fixed end-point. From a visual image, a large piece of fabric on the floor is most often a tapestry, while it is most likely a bed sheet if it is found on a bed. But if a person is able to feel the fabric's material or texture, or the sound that it makes (or not) when grasped with other materials, then he might determine easily the fabric's true function. Object recognition draws on many sensory modalities and the object's behavior, which inspired this thesis approach. Indeed, an object is recognized based on:

- ▷ local features such as its color or geometry
- ▷ the sound it produces or often associated to it
- ▷ being estimated (or not) as a face or the robot own body
- ▷ cross-modal features (using visual/sound patterns)
- ▷ probable locations where to find them based on memories of past locations (or based on other objects presence)
- ▷ contextual features, which are good indicators to identify the identity, location, size, angle and depth most probable for an object given the image of a scene
- ▷ typical scene in which they occur
- ▷ their functional role.

12.2.2 Technological Contributions

This thesis introduces new ideas to solve particular, well defined research problems in the fields of signal processing, computer vision, cross-modal perception and robot

control. As stated hereafter, some technological contributions consists of incremental improvements, while other propose completely new algorithms, or new approaches, to solve a research problem.

Time-Frequency Events Detection and Object Segmentation Events are detected and classified at multiple time-frequency resolution scales. Two different active object segmentation algorithms receive as input oscillatory points or points under discontinuous motions, and group such points into meaningful percepts of objects. Deformable contours were exploited for optimizing object boundary detection. This thesis contribution to the object segmentation problem is the large spectrum of events that enable the active figure/background segregation.

Perceptual Grouping Algorithm A new algorithm was proposed, based on human-robot interactions, to group the perceptual elements of objects into a unified percept. This algorithm is especially useful for segmenting objects that cannot be moved. Repetitive trajectories of a helping human arm/hand/finger are used to group regions of coherent features. The algorithm is shown to work using regions of coherent features of color clusters, texture clusters or regions segmented by applying the minimum-cut segmentation algorithm.

This innovative approach proved itself extremely useful to extract visual appearance of objects from books (a learning aid), or to segment heavy furniture objects in a scene. However, the range of potential applications is vast, including the segmentation of nearly any object.

Cross-modal Object Recognition Inputs from two different sensorial mechanisms - visual and auditive - are processed to enable the recognition of an object solely through the analysis of audio-vision correlations. This is an innovative object recognition approach, since no information concerning the visual appearance or the acoustic signature of an object is used. Instead, input features for this algorithm are features that only exist in the cross-modal sense, since they are ratios of modal properties (periods and energy peaks). Such properties are extracted from the temporal evolution of the modal signals following a Dynamic Programming approach. Sound decay ratios or object color have no effect on the final recognition result. A standard learning algorithm is then used for classification.

Priming for Attention Development of an algorithm for mapping multiple objects to multiple sounds heard, or for priming for specific information (for a sound or for a visual percept).

Self-Recognition Correlation with proprioceptive data from the robot joints is exploited to associate visual and acoustic patterns as being produced by the robot itself.

Appearance and Depth from Human Cues Algorithms are proposed to infer coarse 3D measures and to build scene descriptions containing both the visual appearance of objects on such scene from multiple viewpoints, as well as the depth at which they appear.

Scene Representation This algorithm exploits situatedness by building visual descriptions of scenes from furniture objects segmented or recognized from the robot's surrounding world. Scenes are built in egocentric coordinates, from groups of related objects (in the sense of proximity).

Coarse 3D Object Reconstruction A new embodied approach for 3D depth inference is proposed based on the concept of the familiar size of objects. The familiar, approximately constant size of the human arm if sufficient information to recover depth information for an object. A depth point is estimated as inversely proportional to the human arm diameter. Depth maps are obtained by having a human waving along the boundaries of (and close to) such object. This algorithm is not intended to be better than other stereo or monocular depth inference algorithms. It is just an alternative approach based on the familiar size of objects. But for textureless images, or for certain low resolution representations, stereo or depth from motion will fail – as will most algorithms. But not this thesis approach.

Map Building The map building approach results from merging the two previous algorithms. It is an innovative approach for building 3D maps of scenes, since it relies on humans to demonstrate to the robot the fundamental elements (such as furniture of a scene).

Situatedness – exploiting contextual information Objects, robots, humans are situated in the world. Strategies to exploit information stored in the world were developed based on contextual information. A novelty for these strategies is that training data is generated automatically from interactions with a human caregiver. Contrary to most approaches in the literature (Torralba, 2003; Oliva and Torralba, 2001), based on 2D Short-Time Fourier Transforms or Gabor filters, contextual information is extracted by Wavelet decomposition. Learning follows from the application of a standard classifier.

Robot Localization from Scene Recognition Images labelled in different scenes are automatically annotated to the corresponding scene, which trains a scene recognition algorithm based on the spectral-spatial image components. This is motivated by the fact that different scenes may have very different spectral distributions of frequencies. For instance, corridors have strong vertical edges corresponding to doors and other structures, and well defined horizontal edges at the end. High frequency components are frequent in cluttered offices with book shelves, spread all over the place.

Contextual Selection of Attentional Features Holistic representation of images is used for selection of the attentional focus, scale, size, orientation or depth of an object, or any combination of them. Such an algorithm is very useful, for instance, to identify probable locations of objects in a scene (e.g., chairs are probable in front of tables) even if they are occluded or not visible.

Object-base Selection of Attentional Features Algorithm for selection of object properties (the attentional focus, scale, size, orientation or depth of an object, or any combination of them), given other object properties. This algorithm memorizes object properties using familiar relations to other objects. This is a statistical framework for exploiting the familiar size notion using any object. Indeed, given certain properties for an object (such as its depth), it computes another objects properties (which also includes its depth). Therefore, it is a stochastic way of translating the arm reference cue as a measure of depth into depth relations between objects.

World Correlations A new approach is presented for detecting correlations among events, visual shapes and acoustic percepts perceived from the robot's surrounding world. Several interesting problems are solved by coupling this strategy to other algorithms. Hence, strategies based on object trajectories and hand gestures recognition, as well as on cross-modal data association, are used to teach a robot from several learning scenarios including educational or play activities such as drawing, painting, or playing musical instruments.

Hand gestures recognition Standard hand gesture recognition algorithms require an annotated database of hand gestures, built off-line. Different approaches, most often relying on dynamic programming techniques, have been proposed, such as Space-Time Warping or Space-Time Gestures (Darrel and Pentland, 1993). We follow a very different approach. Objects are recognized on-line, and geometric descriptions acquired from such objects are mapped into periodic hand trajectories. Hence, the geometry of periodic hand trajectories are on-line compared to the geometry of objects in an object database, instead of being mapped to a database of annotated gestures.

Object Recognition and Shape from Hand Gestures The problem of recognizing objects in a scene is also solved by using the dual version of the previous problem. The geometry of periodic hand gestures is used to build an image template, which is then mapped to geometric elements in the visual field. Hence, visual geometries in a scene (such as circles) are recognized as such from hand gestures having the same geometry (as is the case of circular gestures). This algorithm is especially useful to identify shapes from drawing, painting or other educational activities.

Shape from Functional Constraints This algorithm works by mapping periodic object trajectories acquired by an attentional tracker, to world objects having the some geometry, using object recognition operating on geometric features. This is especially useful to recognize world shapes by the constraint that they impose in other objects (e.g., the circular motion constraint which a circular railway track imposes on a train with wheels in such rail). An inter-object constraints are pervasive in the world (e.g., slide-and-crank mechanisms, car on tracks,...).

Shape from Acoustic/Visual Patterns A new approach segments the shape of an object by the sound produced by other objects acting on its visible surface (e.g. a stick scratching a rugged surface). This algorithm is useful to extract informative percepts from several musical instruments, such as a xylophone.

Neural Oscillators A new mathematical framework is described for 1) tuning neural oscillator parameters; 2) stability analysis of neural oscillators; and 3) estimation of errors bounds. This framework is applied for the execution of rhythmic motions by the humanoid robot.

Perception guided Control Integration Integration of rhythmic and non-periodic control structures for the control of arbitrary motions

12.3 Directions for Improvement

This thesis places special emphasis on the integration of cognitive functions in the humanoid robot Cog. However, new algorithms for a big portion of these functions have to be developed, for which this thesis proposes original approaches, as described in the previous section. This was motivated by the fact that for some of these problems, previous approaches required off-line annotated data. Hence, the removal of this off-line requirement led to the development of on-line approaches while introducing the human in the loop to facilitate data acquisition.

In addition, the robot is to learn from a plethora of learning aids and educational tools available for boosting children cognitive development. But algorithms were not readily available to achieve such goals, which motivated the development of original work to cope with such problems (e.g., extracting visual/auditory percepts from books).

Hence, in addition to concentrating on the integration of components, there was the real need of developing original work for many of these components. Due to the large scope of this work, there are future improvements that could effectively improve the individual algorithms' performance or the overall cognitive integration, as follows.

- The work on this thesis is being developed on low resolution (128×128) images due to real-time constraints. However, this resolution constraint poses severe problems for object recognition, since it bounds the scales at which objects can be recognized without actions being applied on them.

  Infants are as well born with low-acuity vision. But infants visual performance thereafter develops in accordance with their ability to process incoming stimulation (Johnson, 1993) (and this is the case as well for the motor system). The developmental process starts from simple systems that become increasingly more complex, significantly improving learning efficiency (Thelen and Smith, 1994). Therefore, it would be worthy to implement this autonomous gradual increase in image resolution for Cog to learn more efficiently.

- Integration of more depth inference methods for more accurate information concerning object and scene structure. Indeed, Shape from Motion, as described by the experiments in Appendix B, still needs to be integrated with the human-based depth inference method. Additional integration can result by integrating the stereo vision framework developed under the M2-M4 Macaco project (Arsenio, 2003c) (see also (Arsenio and Marques, 1997) for previous work on 3D reconstruction). Increased image resolution is required for this integration to take place.

- This work could also be extended to account for multi-link object recognition for carrying out the identification of possibly

  - parallel kinematic mechanisms (e.g., two pendulums balancing in a bar)
  - over constrained kinematic mechanisms (e.g., a slide and crank mechanism)
  - mechanisms constrained by pulleys, belts
  - chain of rotating cranes or pendulums
  - the different moving links of an object (e.g., car with wheels)

  or a combination of them. Such scheme could then be applied developmentally: for instance, by learning the kinematics of two rolling objects, such as two capstans connected by a cable, the robot might use such knowledge to determine a configuration of a capstan rolling around the other.

- Processing of tactile information, and cross-modal integration of this sensory modality.

- Integration of parsing structures for language processing. This thesis describes methods to learn about sounds and first words. A babbling language was previously developed at our laboratory (Varchavskaia et al., 2001), as well as a grounded language framework to learn about activities (Fitzpatrick, 2003b). The integration of these components would add flexibility for interacting with a robot. In addition, there is strong correlations between learning the execution of motor tasks and speech, which would be interesting to exploit.

- More work on learning by scaffolding from educational activities between a robot and a helping caregiver. Good examples of such future learning include games in which the caregiver position babies facing them in order to play "roll the ball" or "peek-a-boo" games (Hodapp et al., 1984). Another interesting example occurs when caregivers help toddlers solving their first puzzles by a priori orienting pieces in the right direction. In each case the children receives very useful insights that help them learn their roles in these social interactions, making future puzzles and games easier to solve.

  Solving puzzles is therefore an interesting problem involving object recognition, feature integration and object representation, for which this thesis framework could be extended. In addition, hand gestures recognition from non-repetitive

gestures (but without hand gesture annotation) would be of interest, since a lot of information can be conveyed to a robot by having a human gesturing.

- Applying task identification to more complex tasks. This is an perhaps the most important improvement, required to create a strong link between data learned by the robot about objects and actions, and the execution of complex tasks from these data.

- This thesis includes algorithms for face recognition and head pose inference. One of the original goals for such algorithms was to add them into the contextual framework to learn about objects. Indeed, similarly to what happens to objects, there are probable places where to find people (usually they do not hang from a ceiling up-side-down, but are common on side-walks!). But people identity also constrains the range of possible places to find people – The most probable place to find Rodney Brooks is at his office at MIT CSAIL headquarters, while his students are usually sitting down at their desks in the laboratory. Not least important, people often gaze at interesting information, and therefore the same statistical framework could be extended to find probable locations of objects based on who is in a given image, and where that person is gazing into – Rod's students are often looking at a computer monitor when sitting at their desks.

  Taking into account this information into the statistical framework is a very important step that remained to test because the robot ran out of time. The extended statistical framework could be used to incrementally gather information for shared attention mechanisms, offering both a developmental and statistical alternative to (Scassellati, 2001; Nagai, 2004).

- Incorporation of the motivational system developed under the M2-M4 project to modulate robot behaviors (Arsenio, 2003b,d) – see (Breazeal, 2000) for the motivational system implemented on the robot Kismet at our laboratory. The modelling of emotional states (Breazeal, 2000), and recognition of speech emotional tones (Breazeal and Aryananda, 2000), were worth integrating for more expressive feedback from the robot to the humans with which it interacts.

All these improvements would not have been enough to create a robot acting as a 2-year-old infant, but each would be an important step towards such goal.

12.4 The Future

Here are some problems that would be interesting to deal exploiting this thesis framework:

- An interesting possibility for future research is applying Genetic Algorithms (GAs) to evolve different kind of kinematic mechanisms. Indeed, from a database of mechanisms learned by the robot, the robot would evolve complex mechanisms using the learned ones as primitives. For example, two rotational bars

with a sliding mechanism could form a Slide-and-Crank mechanism when assembled.

- One topic that I find exciting is the one of not only self-reconfigurable robots, but also self-reconfigurable architectural artifacts, such as chairs or even more elaborated structures. Processing the kinematic structure of objects, together with object recognition schemes, is crucial to achieve such goal. It is theoretically possible that a structure could download a copy of its own software to the new structure, by the Recursion Theorem (Sipser, 1997).

It is always useful to analyze results under the perspective of new technology applied to incumbent markets, or else to new technological niches in the market.

- One such example is given by (Arsenio, 2003d) for the development of complex, small, light and flexible robotic heads. Application of such robotic technology in markets (especially the security market), is discussed in (Arsenio, 2003b). Development of pervasive robotic technology motivated the creation of a project for a start-up: `"Instinct Technologies"`, which competed at the 2002 MIT 1K competition.

- Teaching robots from books would be an interesting market application. The software required is computationally inexpensive, and it could easily be incorporated on a micro-chip. Integration of such technology with cross-modal association (which is also computationally inexpensive) would result in original and cool products. Just imagine a child showing a tree from a book to a baby robot (or a Sony AIBO robot), while saying tree. The posteriori visual perception of the tree would enact the production of the sound tree by the robot, or listening to the repetitive sound `tree` would trigger visual search behaviors for such object.

  Since most thesis algorithms rely on human-robot interactions, there is a lot more market potential behind this example.

Cog's life was full of broken *bones*: broken cables, motors, sensors, etc. This thesis presented Cog's last achievements (see Cog going away to a museum tour, for retirement, in figure 12-2). But Cog, and this thesis, will hopefully constitute an important step towards the goal of creating a 2-year-old-infant-like artificial creature.

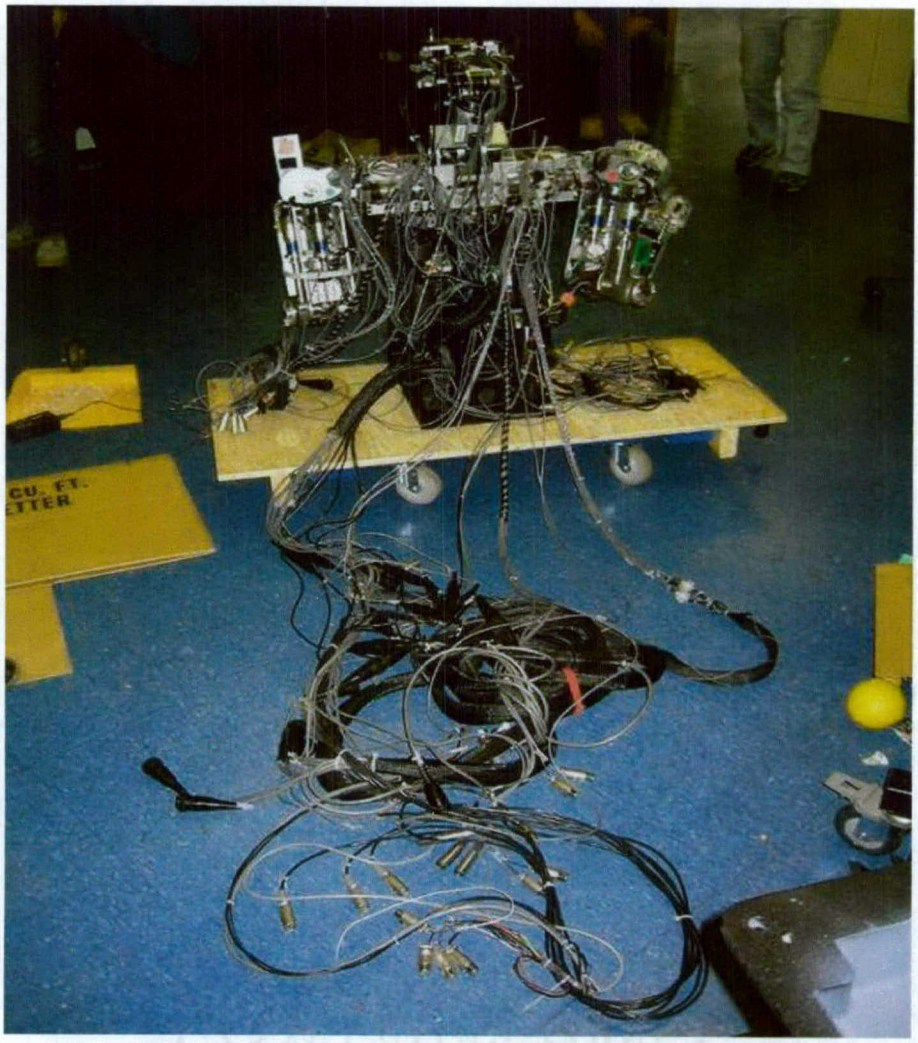


Figure 12-2: The *death* of a humanoid robot - Cog going away. An end to the interminable succession of motor breaks, broken cables and sensor failures. However, she/he will be missed.

Appendix A

Simple Bodies for Simple Brains

(And) of the soul the body form doth take;
For soul is form, and doth the body make.

(Aristotle; BC, 350)

This thesis follows the embodiment principle: that artificial brains require artificial bodies (Brooks and Stein, 1994) (or some `"kind"` of body). Therefore, we build real robotic systems to test our methodologies.

A.1 The Humanoid Robot Cog

My thesis work was developed on the humanoid robot Cog (Brooks and Stein, 1994), a complex humanoid robotic platform (see Figure A-1). The humanoid form is important for human-robot social interactions in a natural way.

A.1.1 Cog's Arm manipulators and Torso

The humanoid robot Cog has two six degree of freedom arms. Each joint is driven by a series elastic actuator (Williamson, 1995). This actuator consists of a motor serially connected to its load by a spring. This compliant arm is designed for human/robot interaction. Friction effects and gear backlash are minimized by the spring, which acts as a low pass filter. However, the spring also filters out high frequency commands,

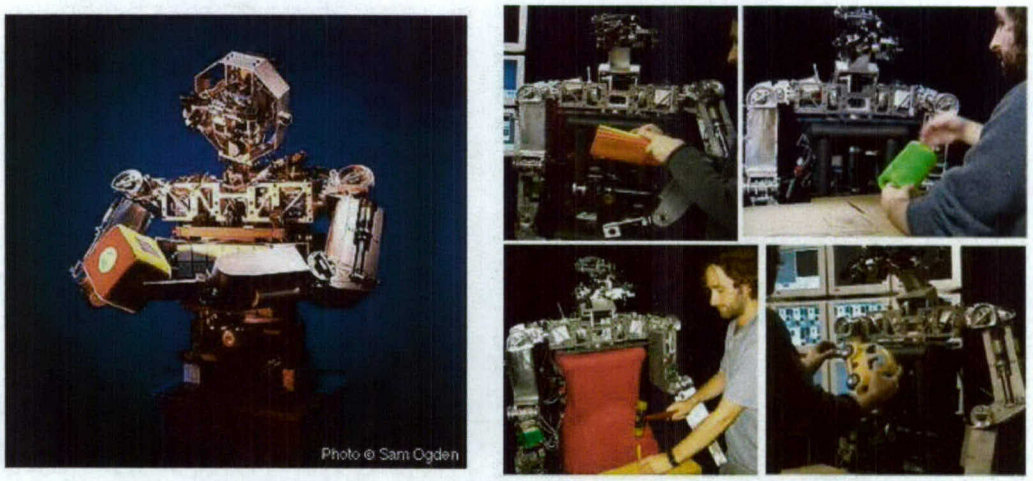


Figure A-1: The humanoid robot Cog. It is shown through several learning scenarios.

reducing significantly the control bandwidth. Hence, the arms are not reliable for the execution of precise motions, because of the low stiffness. While these problems will probably not constitute a problem for such tasks as throwing and poking, they definitely pose serious considerations for other tasks, such as controlling a pendulum or catching flying objects. Proprioceptive feedback available at each joint includes joint position from potentiometers and force (from the deformation of the elastic element as measured by strain gauges). Joint velocity at each joint is not measured, which poses a serious limitation, since it has to be estimated. Such estimation introduces significant noise.

Cog has also a three degree of freedom torso which was not actuated throughout this work. A microphone array containing sixteen sensors is placed across the upper torso for auditory processing.

A.1.2 Cog's Head

Cog's head is not equipped with series elastic actuators, since high bandwidth is essential for active vision. The robotic head has seven degrees of freedom (four in the neck, three for the eyes). Each eye has two cameras - one for the foveal vision, of higher resolution, and the other for a wide field of view. The eyes have a common actuated revolute tilt joint and independent pan actuated joints. An inertial sensor is placed in the head.

A.1.3 Cog's "neurons" and "synapses"

Cog's "brain" includes 32 processors, most of them Pentium III at 800MHz, to meet near real-time processing demands. The main operating system is QNX, a real-time operating system with a nice, transparent message-passing system, which runs on 28 of the processing units. The other four run Linux for auditory processing.

Cluster communication was chiefly implemented by Fitzpatrick (Fitzpatrick, 2003b) and Marjanovic (Marjanović, 2003). The following description follows closely the one presented by (Fitzpatrick, 2003b).

Three main communication entities were extensively applied:

- Synchronous synchronization using QNX message passing system. The server process registers an identification name in QNX (using the native service nameloc). Any client can send messages to the server, sending together the client process identification. The client can enter a waiting state until a reply occurs or keep running.

- Ports (Fitzpatrick, 2003b) are communication units which can be created in any number by all processes or threads. Ports can communicate with each other, and they shield communication complexity from their owner. The are ideal for implementing asynchronous communication between processes or threads. The client assigns the Port a name. This name is registered in a global namespace. The client can have a Port transmitting a piece of data, or reading data the Port has received either by polling, blocking, or callback. This approach is compatible with the existence of special memory areas managed by other entities. Ports use native QNX messaging for transport. The name server used is QNXs native nameloc service.

- For running or communicating with a non-QNX system, Ports use sockets. A simple socket-based wide nameloc service permits communicating with a non-QNX system (the Non-QNX/QNX issues are transparent to client code by default).

A.2 M2-M4 Macaco Project

A small number of software modules described on this thesis were initially developed under the M2-M4 Macaco robotic head project. I designed and built this active robotic head (at the mechanical, hardware and software level). M2-M4 Macaco is a two-in-one active vision robotic head designed to resemble two biological creatures. This flexible robotic head is described in detail in (Arsenio, 2003d,b,c).

The construction of both dog like and primate like robotic bodies at the MIT LegLab led to the creation of canine/primate like heads to fit these bodies. One goal of the M2-M4 Macaco project was, therefore, the creation of a flexible robotic head that could match both of these quadruped (M4) and biped (M2) robots. The replacement of a few number of M2-M4 Macaco aesthetic components allows for this metamorphosis, while all the remaining mechanical components are functional. The robotic head is easily assembled to a mobile platform, being the communications carried out through a serial port.

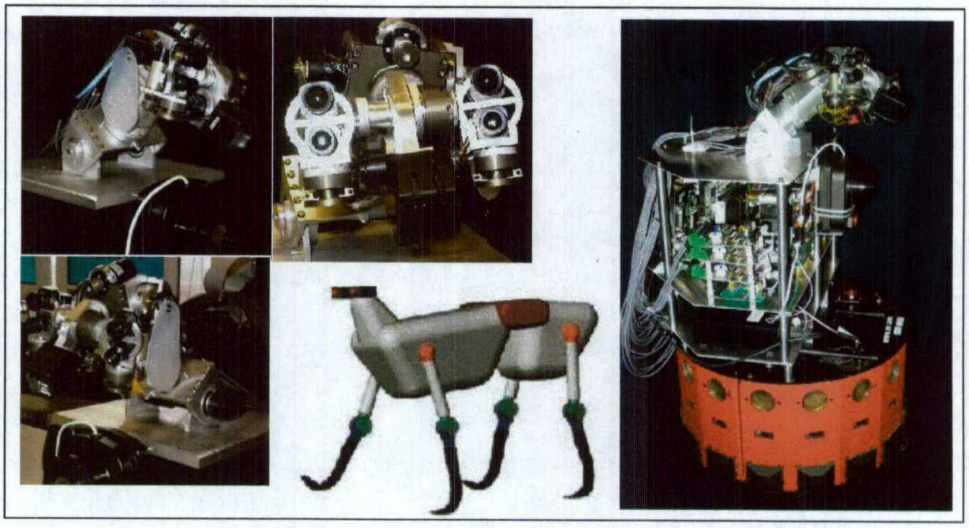


Figure A-2: The M4 Macaco robot, a biologically inspired active vision head designed to resemble a dog's head. It is also shown the original robotic body, resembling a dog, designed and built by the MIT AI LegLab. The head was latter assembled to a iRobot mobile platform, also shown.

A.2.1 Biological Inspired Design

A robotic mechanism designed to act as an intelligent creature should explore the mechanisms which evolution provided these creatures with, and incorporate them on the design. Hence, this robotic head was designed to resemble a biological creature, exploiting several features of an evolutionary design, but adding others (such as a thermal camera for human detection and night vision) for improved performance.

A.2.2 M4-Macaco

The M4-Macaco active vision head (shown in Figure A-2) resembles a dog-like head. The weight of the head, including motors, gears and gyro, is $\sim 3.55lbs$ ($\sim 1.6Kg$). Its height is approximately $7.5in$, and the distance between the eyes is $\sim 3.5in$. The head has seven degrees of freedom (four in the neck, three for the eyes), as shown in Figure A-3. It also includes two eyes, and two cheap CMOS miniature color board cameras (specially well-suited for dynamic scenes) at each eye - one for the foveal vision, of higher resolution ($31^o \times 23^o$), and the other for a wide field of view ($110^o \times 83^o$). The eyes have a common actuated revolute tilt joint and independent pan actuated joints. An inertial sensor was placed in the nose of the M4 head. All motors are controlled by small, compact and light JRKerr controllers, with amplifiers onboard. Motor commands are sent through two serial ports.

Macaco is a stiff mechanism. The three motors at the neck joints are in a direct-drive configuration, minimizing backlash and increasing stiffness. Although solely the Cog platform is used throughout this thesis, all the software was made compatible with software on the M2-M4 Macaco platform, and vice-versa.

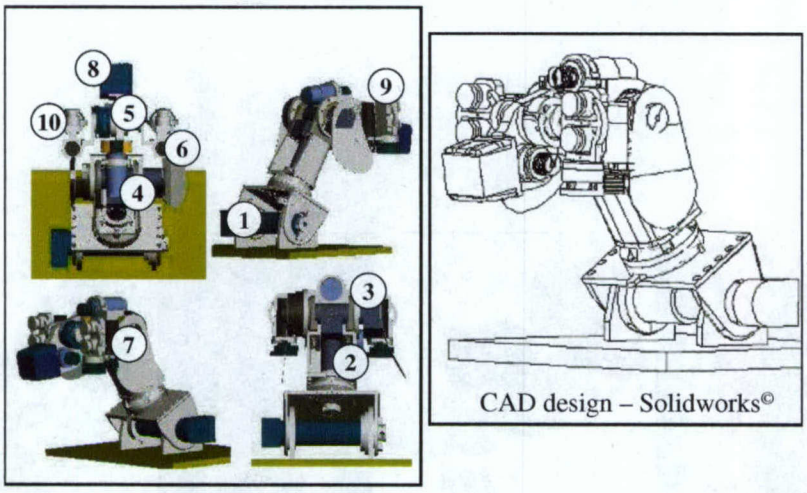


Figure A-3: M4 Macaco Computer Assisted Design (using the CAD tool *solidworks*©) showing motors and sensors. 1) Neck tilt motor (direct-drive actuation); 2) Neck pan motor (direct-drive); 3) Neck/Head Tilt motor (direct-drive); 4) Head Roll motor; 5) Eye Tilt motor; 6) and 7) Right and Left Eyes pan motors, respectively; 8) Inertial sensor; 9) and 10) Right and Left eyes, respectively. Each eye has two cameras. Cabling for the motors and sensors goes through the neck inside, near to motor (2).

A.2.3 M2-Macaco

The M2-Macaco active vision head (shown in Figure A-4) resembles a primate-like head. The number of sensors is identical to M4, although the inertial sensor was placed inside of the upper jaw of this head, instead of inside the nose. The motor configuration is the same as M2-Macaco, except for an additional motor moving M2-Macaco's lower jaw (offering a potential application for simulating speech synthesis – never used).

A.2.4 Macaco's `"neurons"` and `"synapses"`

Macaco's hardware consists of two PC104+ boards, AMD K6-II at 400MHz processor based, and seven small CPU boards with Pentium III at 800MHz, all modules connected by an Ethernet network. A total of nine framegrabbers are used for video acquisition, being each camera connected to several grabbers to reduce network latencies. The network server is connected to a notebook hard drive. Similarly to most of Cog's processors, OS-QNX runs on all processors, featuring high performance and real time transparent communications.

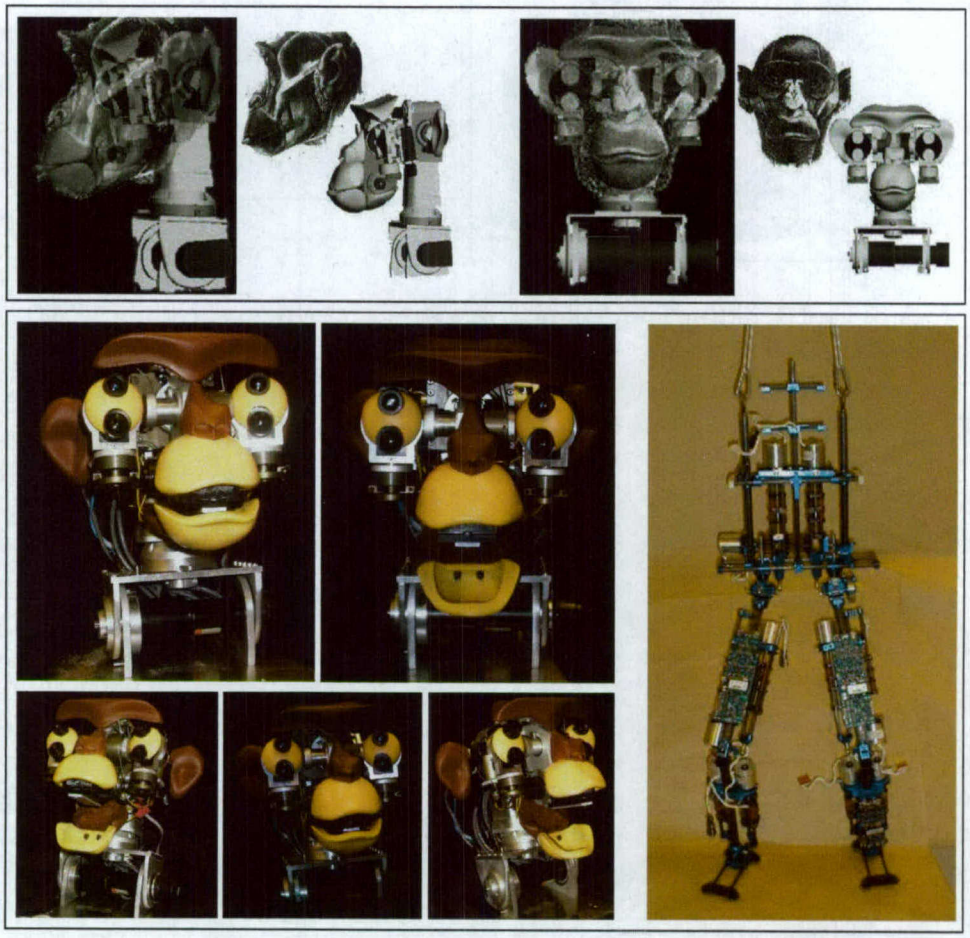


Figure A-4: M2-Macaco, a biological inspired robotic head designed to resemble a primate head. The M2 body, designed at the MIT AI LegLab, is also shown. The design of this head required some aesthetic parts described by complex surfaces. The CAD tool *Rhinoceros*© was used to create such surfaces, which were manufactured using the *Duraforming* manufacturing method.

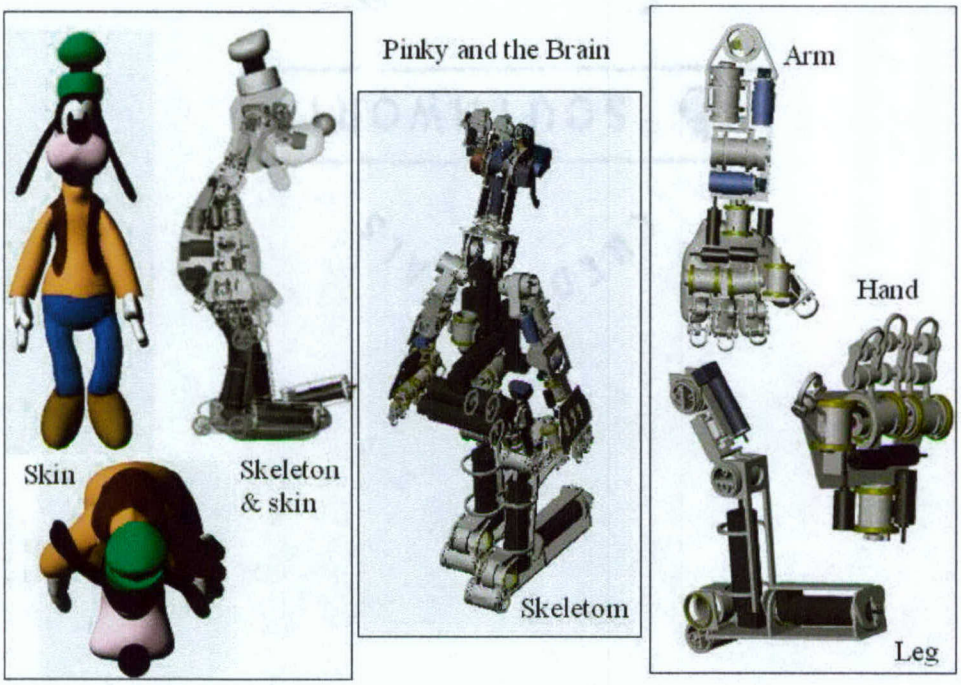


Figure A-5: Design of Pinky robot, and Deformable Skin. Pinky current design is in a very preliminary form. The motor selection is already done, together with the robot shape, bearings, and actuator design. This robot will include a new type of actuator, with variable stiffness.

A.3 Pinky and the Brain

Pinky (a mechanical humanoid robot) and the brain (a cart with his electronic hardware and processing units) that pinky pushes, is my view for building int the future a humanoid robots. The preliminary CAD mechanical design in *Solidworks*© is shown in Figure A-5, together with the design (using *Rhinoceros*©) of a silicone skin.

With the goal of having large computational power, Pinky will possibly have a cart that will contain the brain, and which Pinky will push as he walks, so that the overall system is completely autonomous. This is a biped robot that is designed to walk, and resembles goofy, although several dog features (pluto-like) were added. The robot was designed for human interaction. It has a big belly (which is appellative for humans and good to store electronic hardware), big feet (for a funny kind of walk and good static stability when up), goofy-like face features, big ears actuated each only by one motor to reduce complexity, mouth, four cameras in two eyes (this is another example of application of the M2-M4 flexible robotic heads), and an inertial sensor.

Appendix **B**

Camera Calibration and 3D Reconstruction

Camera calibration ought to estimate camera parameters, which can be grouped as:

intrinsic camera parameters specify the camera characteristics properties; these parameters are:

- focal length, that is, the distance between the camera lens and the image plane,
- location of the image center in pixel coordinates,
- pixel size,
- radial distortion, coefficient of the lens.

extrinsic camera parameters describe a spatial configuration of the camera relative to a world referential, given by a rotation matrix and a translation vector.

Camera calibration techniques can be classified roughly into two categories:

Photogrammetric calibration: Calibration is performed by observing a calibration object whose geometry in 3-D space is known with very good precision. These approaches require an expensive calibration apparatus, and an elaborate setup.

Self-calibration: Techniques in this category do not use any calibration object. If images are taken by the same camera with fixed internal parameters, correspondences between images are sufficient to recover both the internal and external parameters.

Calibration Constraints

Consider a collection of points $q \in \mathbb{R}^3$ in 3-dimensional Euclidean space. In homogeneous coordinates such points are given by $q^h = (q_1, q_2, q_3, 1)^T \in \mathbb{R}^4$. The perspective projection of such a point onto the 2-dimensional image plane is represented, in homogeneous coordinates, by $x\mathbb{R}^3$ satisfying, for image i,

$$\lambda_i x^h(i) = K_i g(i) q^h \tag{B.1}$$

where $\lambda_i \in \mathbb{R}$ is a scalar parameter and the non-singular matrix K_i – the calibration matrix – describes the geometry of the camera.

The Euclidean transformation $g \in SE(3)$ – SE(3) is the Special Euclidean Lie Group – is defined by a rotation $R \in SO(3)$ – SO(3) is the Lie group of orthogonal matrices with determinant one, denoted as the Special orthogonal group in $\mathbb{R}^3$ – and a translation T.

For a sequence of n images, (B.1) can be organized as (Ma et al., 2000),

$$\begin{pmatrix} x^h(1) & 0 & \dots & 0 \\ 0 & x^h(2) & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & x^h(n) \end{pmatrix} \begin{pmatrix} \lambda_1 \\ \lambda_2 \\ \vdots \\ \lambda_n \end{pmatrix} = \begin{pmatrix} M_1 \\ M_2 \\ \vdots \\ M_n \end{pmatrix} q^h \tag{B.2}$$

given by $X\vec{\lambda} = Mq^h$ in a compact notation. The motion matrix $M \in \mathbb{R}^{3n \times 4}$ is defined from the full rank matrices M_i, $rank(M_i = 3)$ for $i = 1, \dots, n$.

$$M_i \doteq (K_i R_i, K_i p_i) \in \mathbb{R}^{3\times 4}, \quad i = 1, \dots, n \tag{B.3}$$

If we denote the four columns of the matrix M by $\vec{m}_i \in \mathbb{R}^{3n}, i = 1, \dots, 4$, and the n columns of the matrix X by $\vec{X}_i \in \mathbb{R}^{3n}, i = 1, \dots, n$, then the coordinates $x^h(i)$ represent the same point tracked from different views only if they satisfy the following wedge product equation:

$$\vec{m}_1 \wedge \vec{m}_2 \wedge \vec{m}_3 \wedge \vec{m}_4 \wedge \vec{X}_1 \wedge \dots \wedge \vec{m}_n = 0 \tag{B.4}$$

This constraint is multi-linear in the measurements $x^h(i)$, and results from the fact that X is contained in the span of M. Constraints involving two images are called *bilinear* or *fundamental*, constraints involving three images are called trilinear, and so on.

B.1 Non-linear camera calibration

The technique we used for camera calibration using a calibration grid, together with the *C code* to implement it, are widely available from the Intel OpenCV library, and is described in detail by (Zhang, 1999). The algorithm consists of a closed-form solution, followed by a nonlinear refinement based on the maximum likelihood criterion, which is just briefly described here.

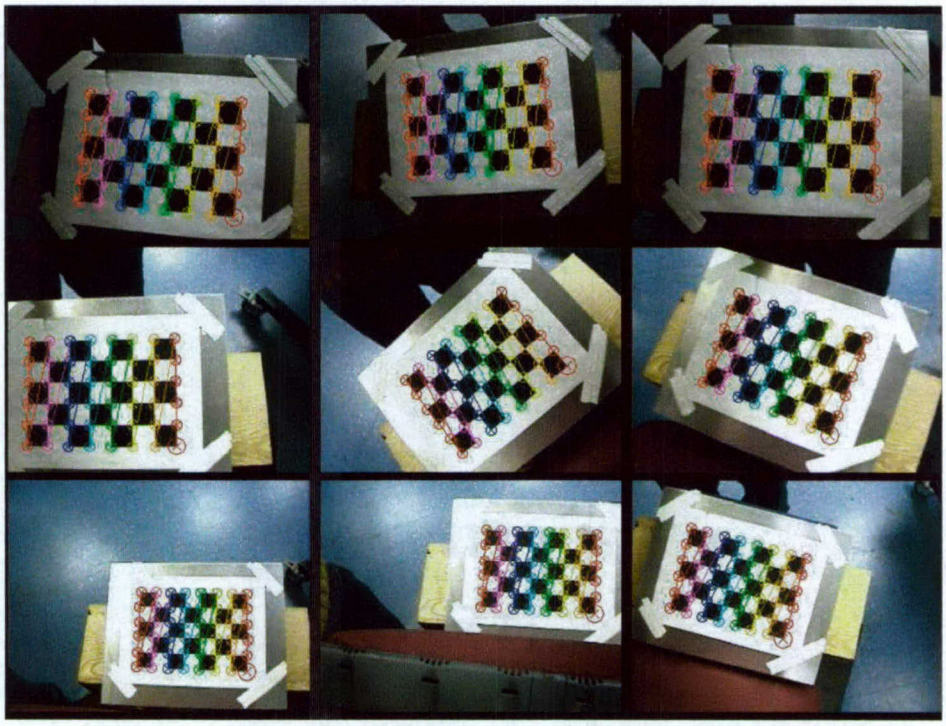


Figure B-1: Detection of corners for camera calibration, under various planar orientations.

However, since it requires the camera to observe a planar pattern shown at a few (at least two) different orientations, it goes somewhat against this thesis' philosophy. Indeed, this technique relies on an off-line procedure, using a metric calibration pattern, in which either the camera or this planar pattern must be moved manually (the motion needs not be known), or as we will see later, by the robot itself. This is very different from the way our visual system works. The selection was therefore motivated by this algorithm's robustness and the modelling of radial lens distortion.

The calibration procedure is as follows:

1. Take a sequence of images of a calibration planar pattern under different orientations by moving either the plane or the camera

2. Detect the feature points (corners) in the images, using the method described in (Shi and Tomasi, 1994) – see Figure B-1.

3. Estimate the five intrinsic parameters and all the extrinsic parameters as follows:

   (a) Find homography for all points on series of images.

   (b) Find intrinsic parameters; distortion is set to 0.

   (c) Find extrinsic parameters for each image of pattern.

4. Estimate the coefficients of the radial distortion

5. Refine all parameters – main optimization by minimizing error of projection points with all parameters.

B.1.1 Estimation of Planar Homography

Assume, without loss of generality, a plane at $Z = 0$ of the world coordinate system. Then (B.1) becomes, with $g(i) = (R(i), p(i))$ (using a pinhole camera model),

$$\lambda_i x^h(i) = K_i[\ r_1^i \quad r_2^i \quad r_3^i \quad t^i\ ]q^h = K_i[\ r_1^i \quad r_2^i \quad t^i\ ] \begin{bmatrix} X \\ Y \\ 1 \end{bmatrix}^T = H\tilde{q} \tag{B.5}$$

where r_j is the i^{th} column of the rotation matrix R ($r_3(i) = 0$), and H is an homography estimated between the pattern plane and the retinal plane (Faugeras, 1993; Zhang, 1999). Knowing that r_1 and r_2 are orthogonal, results, from B.5,

$$\begin{aligned} h_1^T A^{-T} A^{-1} h_2 &= h_1^T B h_2 = 0 \\ h_1^T A^{-T} A^{-1} h_1 &= h_2^T A^{-T} A^{-1} h_2 \end{aligned} \tag{B.6}$$

where the symmetric matrix B describes the image of the absolute conic. Denoting $b = []^T$, equation (B.6) becomes, in a compact form, for a sequence of n images,

$$Sb = 0 \tag{B.7}$$

with S a $2n \times 6 matrix$.

B.1.2 Optimization

The solution of equations in the form of (B.7) is well-known (and it will be widely used later), being given by:

- ▷ The eigenvector of $S^T S$ associated with the smallest eigenvalue, if T is square
- ▷ The right singular vector of S associated with the smallest singular value, otherwise.

Once b is known by solving (B.7), the intrinsic parameters of the calibration matrix K are uniquely extracted from linear equations (Zhang, 1999). Once K is known, the extrinsic parameters are also easily computed from linear equations (Zhang, 1999).

Cameras exhibit often significant lens distortion, especially radial distortion. The distortion is described by four coefficients, two of them radial distortion coefficients, and the other two tangential ones (Zhang, 1999). (Zhang, 1999) presents a solution based on linear least-squares for this problem. The algorithm implementation, as described in (Zhang, 1999), instead of solving alternatively the estimation of the camera parameters and estimation of distortion, is made to converge faster by a maximum likelihood estimation, using nonlinear minimization – the Levenberg-Marquardt algorithm.

B.1.3 Calibration with the robot's arm

The implementation of this framework on Cog was motivated by the need to extract not only the position of the arm's gripper, but also its orientation. A small calibration pattern was attached to the robot gripper, and the cameras were calibrated from moving, manually, Cog's arm and head through different orientations (see Figure B-2). Although not exploited in this thesis work by lack of time, it would be interesting to approach the problem of moving Cog's arm in order to reduce the covariance of the estimation. In other words, given a current estimation of the calibration matrix and of its associated uncertainty, the robot could be controlled in order to move its gripper towards an orientation which would reduce calibration uncertainty the most.

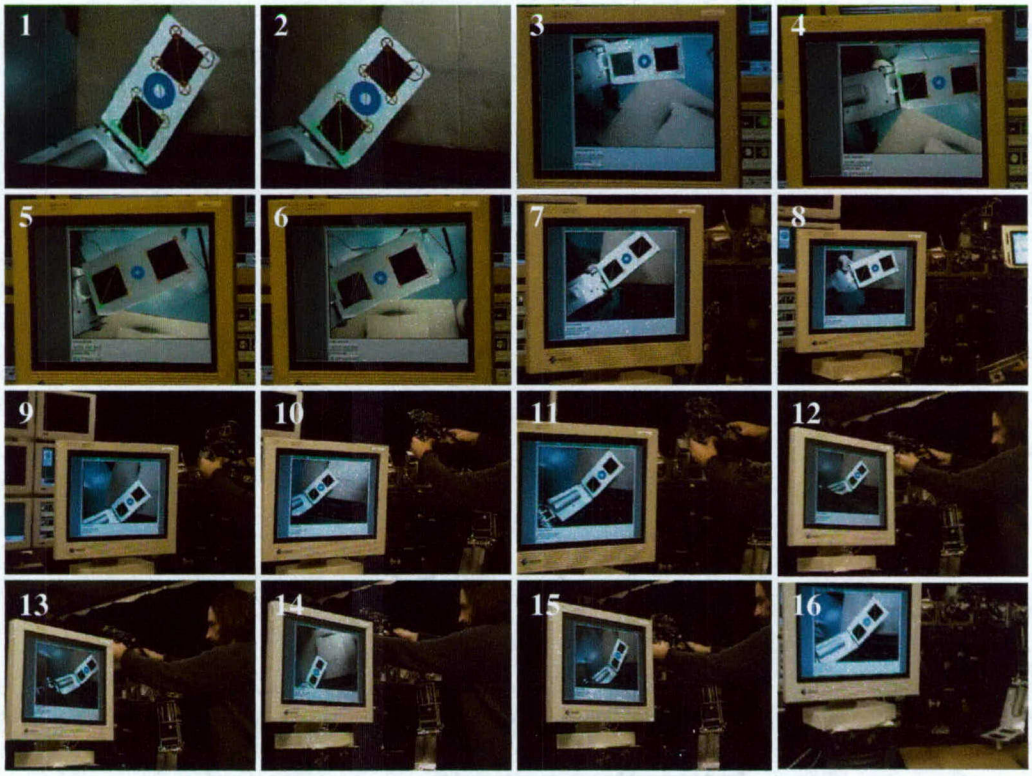


Figure B-2: Camera calibration with a pattern attached to the robot arm's gripper. It also shows the detection of corners for camera calibration, under various planar orientations.

The idea is therefore to implement an active vision strategy for camera calibration. Ultimately, the goal is for *active-self-calibration*: to self-calibrate the eyes without any calibration object, using just the robot gripper to control the motion perceived by the robot, in order to optimize the estimation of the calibration parameters. This embodied vision strategy would probably translate into a more flexible and reliable estimation than the alternative of just controlling the sensing apparatus. Self-calibration

of the cameras is therefore described in the next sections.

B.2 Fundamental Matrix

The fundamental matrix F establishes an homography (Faugeras et al., 2001) between two image planes. It is a simple proof to show that for a point projected into one image with coordinates m_1, all points m_2 belonging to the epipolar line of point m_1 in the second image have to satisfy the following equation (Faugeras et al., 1992):

$$m_2^T F m_1 = 0 \tag{B.8}$$

B.2.1 A Robust implementation of the Eight-point Algorithm

Let us first group the homogeneous vectors $m_2 = [\ x_2 \quad y_2 \quad 1\ ]$ and $m_1 = [\ x \quad y \quad 1\ ]$ as the 9×1 vector $a = [\ x m_2^T \quad y m_2^T \quad m_2^T\ ]^T$. The equation (B.8) is linear in the coefficients of the fundamental matrix F. Calling X the 9×1 vector built from the columns of the fundamental matrix: $X = [\ f_1^T \quad f_2^T \quad f_3^T\ ]^T$, (B.8) becomes

$$a^T X = 0 \tag{B.9}$$

A set of m point correspondences yields m equations like (B.10), which can be grouped together,

$$A_m X = 0 \tag{B.10}$$

where A_m is an $m \times 9$ matrix: $A_m = [\ a_1^T \quad \ldots \quad a_m^T\ ]^T$.

Since F is defined up to a scale factor, there are only 8 unknowns (and therefore we need $m \geq 8$).

If only 8 points are used ($m = 8$), and A_8 has rank 8, then the Eight-point Algorithm, as described in (Faugeras, 1993), can be applied to estimate F. However, the algorithm can be made more robust by using more points, if available (Faugeras, 1993). Therefore, the algorithm, as implemented, works as follows. If the first 8 columns of are linearly independent – tested by applying a singular value decomposition, and requiring 8 non-zero singular values – equation (B.10) can be written as

$$A'_m X' = -X_9 C_9 \tag{B.11}$$

where X' is the 8×1 vector containing the first 8 components of X, and C_9 is the ninth column vector of A_m, and A'_m is the $m \times 8$ matrix containing the first eight column vectors of A_m. Using the pseudo-inverse matrix, we obtain the elements of the fundamental matrix:

$$X' = -X_9 (A'^T_m A'_m)^{-1} A'^T_m C_9 \tag{B.12}$$

B.2.2 Singular Value Decomposition Method

The robust implementation of the eight-point algorithm solves the problem of estimating F when no noise is present. When noise is present, the problem can be reformulated as the estimation of the vector X which minimizes the norm of $A_m X$ subject to the constraint that their squared norm is 2 – so that $\|t\| = 1$ (Faugeras, 1993):

$$\begin{aligned} &\texttt{min}_X \|A_m X\|^2 \\ &\texttt{subject to } \|X\|^2 \;=\; 2 \end{aligned}$$

The solution is the eigenvector of norm $\sqrt{2}$ of the 9×9 matrix $A_m^T A_m$ corresponding to the smallest eigenvalue, and the solution is unique if the eigenspace associated with this eigenvalue is of dimension 1.

B.3 Self-Calibration

We describe here a technique proposed by (Mendonça and Cipolla, 1999), which is an extension of Hartley's self-calibration (Hartley, 1992) technique (which can only obtain two of the camera intrinsic parameters) for a sequence of images based on the properties of the essential matrix.

The essential matrix has the following two properties(Mendonça and Cipolla, 1999):

- ▷ one out of its three singular values is zero (the essential matrix has rank two)
- ▷ the other two singular values are identical

This matrix is obtained from the fundamental matrix by a a transformation, which depends on the intrinsic parameters of the camera (or pair of cameras) grabbing the two views:

$$E = K_2^T F_{12} K_1 \tag{B.13}$$

Hence, constraints on the essential matrix can be framed as constraints on these intrinsic parameters.

B.3.1 Algorithm

The calibration matrix K can be estimated automatically from the knowledge of the fundamental matrix, for certain type of motions – as analyzed rigorously by (Ma et al., 2000).

Let n be the number of images necessary for self-calibration of the camera. Each known intrinsic parameters imposes n constraints to the projective transformation applied to the camera to calibrate it, while a constant intrinsic parameter introduces

one less constraint. Denoting the number of known intrinsic parameters of each camera as n_k, the number of unknown constant intrinsic parameters as n_c, then n is given by

$$n \times n_k + (n-1) \times n_c \geq 8 \tag{B.14}$$

The projective space has 15 d.o.f. (each of the M_i matrices in (B.3) depends on 15 parameters) – the calibration matrix depends on 9 parameters, and the Euclidean transformation on 6 parameters: 3 for rotation (A rotation R is parameterized by a vector r of three parameters, and both are related by the Rodrigues formula (Murray and Sastry, 1994)) and 3 for translation. But since arbitrary scale can not be recovered from self-calibration (Ma et al., 2000), we can fix one of the calibration matrix parameters to 1, or else so that K has unit determinant. Hence, the 8 in equation (B.14) comes from $15 - 6 - 1 d.o.f.$. In addition, if we consider any rotation matrix R_o and let $B = KR_o^{-1}$, then from $BR_oq = BR_oRq_o + BR_op$ we can conclude that a point q, moving under (R, p) and viewed with a calibration matrix K is not distinguished from a point R_oq moving under $(R_oRR_o^T, R_op)$ with calibration matrix B. This means that K can only be computed up to an equivalence class of rotations, that $A \in SL(3)/SO(3)$. Both SL(3) and SO(3) are Lie Groups, being SL(3), the group of 3×3 matrices with unit determinant, denoted as the special linear group in $\mathbb{R}^3$.

Hence, we fix also three parameters so that K has the same structure for all images acquired by the camera. Let F_{ij} be the fundamental matrix relating consecutive images i and j of a sequence of images. Although not strictly necessary, we will assume $K_i = K_j = K$, i.e., the unknown intrinsic parameters are all constant – (Mendonça and Cipolla, 1999) presents the theoretical framework for varying intrinsic parameters. Hence,

$$K = \begin{bmatrix} \alpha_x & s & u_o \\ 0 & \epsilon\alpha_x & v_o \\ 0 & 0 & 1 \end{bmatrix} \tag{B.15}$$

is the structure selected for K, where α_x is the product of the focal length f and amplification factor, ϵ is the aspect ratio, $[u_o \;\; v_o]^T$ are the coordinates of the principal point and s is the skew.

Given a sequence of n images of corresponding points, the calibration matrix K is not affected by translation. In addition, it has a unique solution if and only if the rotation R spans at least two independent directions (according to the $Ma - Košecká - Sastry$ lemma in (Ma et al., 2000)).

Considering σ_1^{ij} and σ_2^{ij} to be the two non zero singular values of $K^TF_{ij}K$, in descending order, we will minimize a cost function C in the entries of K, for $i = 1, \ldots, n-1$ (which is a variant for the one suggested in (Mendonça and Cipolla, 1999):

$$C(K) = \left(\sum_{i,j}^{n} w_{ij} \frac{\sigma_1^{ij} - \sigma_2^{ij}}{\sigma_2^{ij}} \right)^2 \tag{B.16}$$

where w_{ij} is a confidence value in the estimation of the fundamental matrix F_{ij}, given by the number of points used in the computation of F_{ij}.

B.3.2 Nonlinear Numerical Optimization

This technique is numerically stable, robust and accurate. The derivatives of (B.16) can be computed accurately by finite differences, because the function that maps the entries of a matrix to its singular values is smooth, as stated by the Wielandt-Hoffman theorem for singular values (Mendonça and Cipolla, 1999).

The cost function was optimized using a multi-dimensional minimization technique: Powell's conjugate gradient based technique (Luenberger, 1991).

B.4 Depth from Motion exploiting the Essential Matrix

The essential matrix, having a calibrated camera and after computation of the fundamental matrix for two images, is then obtained by simply applying the transform given by (B.13). Two different methods were implemented to extract the Euclidean transformation (up to a scale factor) between the two cameras, which will be described next.

B.4.1 Recovering Translation

From the essential matrix, the translation t and rotation R of the camera's motion (or the dual problem of an object's motion with a stationary camera) between the acquisition of two images can be recovered as follows. Let e_i denote the column vectors of the essential matrix E. The equation $E = TR$ implies not only that t is orthogonal to to all c_i, but also that $c_i = t \wedge r_i$ for all i, and hence $t = \pm c_1 \wedge c_2$.

SVD Method

The computation of t can be made alternatively to take noise into account, because t is also the solution to the following problem (Faugeras, 1993):

$$\begin{aligned} &\texttt{min}_t \|E^T t\|^2 \\ &\texttt{subject to } \|t\|^2 = 1 \end{aligned}$$

The solution is the eigenvector of unit norm corresponding to the smallest eigenvalue.

B.4.2 Recovering Rotation

The rotation R is determined as a function of E and t from (see (Faugeras, 1993) for the details):

$$R = \frac{E^{*T} - TE}{t \cdot t} \tag{B.17}$$

where $E^{*T} = [\ c_2 \wedge c_3 \quad c_3 \wedge c_1 \quad c_1 \wedge c_2\ ]^T$.

Inverse Motions

If instead of a moving camera, one has a moving object, the same results apply by switching the roles of m_1 and m_2, or else by replacing on the end result R by R^T and t by $-R^T t$. We opted by the latter implementation.

B.4.3 Extracting 3D information

Given a calibrated camera with calibration matrix K, for each point M projected to (and tracked between) two images, with retinal coordinates m_1 and m_2, and normalized coordinates given by $u = K^{-1}m_1$ and $v = K^{-1}m_2$, the following constraint applies:

$$PM = b \tag{B.18}$$

with, being p_i the rows of the 4×3 matrix P and r_i^T the columns of R^T (or r_i the rows of R),

$$\begin{aligned} p_1 &= v_x * r_3 - r_1 \\ p_2 &= v_y * r_3 - r_2 \\ p_3 &= \begin{bmatrix} 1.0 & 0.0 & -u_x \end{bmatrix} \\ p_4 &= \begin{bmatrix} 0.0 & 1.0 & -u_y \end{bmatrix} \end{aligned}$$

$$b = \begin{bmatrix} t_x \\ t_y \\ 0 \\ 0 \end{bmatrix} - \begin{bmatrix} v_x \\ v_y \\ 0 \\ 0 \end{bmatrix} * t_z$$

and the 3-dimensional reconstruction M is given by solving equation (B.18) using the pseudo-inverse matrix.

Appendix C

Linear Receptive Field Networks

This appendix reviews learning algorithms as proposed by (Atkeson et al., 1997; Schaal et al., 2000), which were implemented in *C code*. The appendix also elaborates on some minor modifications introduced to speed up processing times. Two main techniques, equivalent to linear receptive field networks, are discussed: Locally Weighted Regression and Receptive Field Weighted Regression.

C.1 Locally Weighted Regression

Locally weighted regression (LWR), as proposed by (Schaal et al., 2000), is a batch update algorithm briefly reviewed here. Memory-based locally weighted regression is a simple, very data efficient *lazy* learning technique that stores all training data in memory. Predictions for new inputs are generated by efficient lookup and interpolation techniques. This kind of learning is especially suitable for forward or inverse transformations. Training of LWR only requires adding new training data to the memory, being therefore computationally inexpensive. This is at the cost of classification efficiency, which decreases with the size of the training data, constraining therefore the usefulness of this approach.

The goal is to fit a locally affine model $y = Ax + a$ to pairs of (x, y) data generated by $y = F(x) + \epsilon$, where x is a n-dimensional input and y a m-dimensional output, being ϵ a noise term with zero mean. However, from the first-order Taylor series expansion, such affine models only approximate the non-linear matrix A locally around an equilibrium point $y_0 = F(x_0)$, with $A = \frac{\partial F}{\partial x}|_{x=x_0}$ and $a = F(x_0)$. In order to approximate as best as possible F, it would be desirable to vary x_0 to a neighborhood of the point to be estimated, i.e., to estimate the region of validity in which a local

model is accurate. This is done by weighting training points (x, y) according to their distance to a query sample x_q, giving more weight to closer training points. This is equivalent to moving the equilibrium point x_0 towards the query point x_q.

Replacing the $n \times m$ matrix A^T by the $(n+1) \times m$ matrix β, whereβ results from appending the vector $a^T + x_q^T A^T$ to the last row of A^T, the problem becomes the estimation of a linear model given by β. The region of validity for each linear model, called a receptive field, can be given by a Gaussian kernel, as originally proposed (Atkeson et al., 1997):

$$w_k = exp(-1/2(x - c_k)^T D_k (x - c_k)) \tag{C.1}$$

where c_k is the center of the k^{th} linear model, and D_k corresponds to a positive-definite distance metric, which defines the shape and size of the this region of validity (diagonal D_k corresponding to spherical gaussians). Other kernel functions could be used. The estimation method consists of estimating the output y given a query point x_q using weighted least squares minimization. The linear weighted regression (LWR) is computed, in pseudo-code, as (Schaal et al., 2000):

Given:

- ▷ a query point x_q
- ▷ p training points $\{(x_i, y_i\}$ in memory, where x_i is an n-dimensional vector and y_i an m-dimensional vector

Compute prediction:

1. determine weight diagonal matrix W with $w_{i,i} = exp(-1/2(x_i - x_q)^T D(x_i - x_q))$
2. build matrix X from homogeneous vectors $\tilde{x}_i$, and output matrix Y such that
   $X = (\tilde{x}_1, \tilde{x}_2, \ldots, \tilde{x}_p)^T$,
   $\tilde{x}_i = [(x_i - x_q)^T 1]^T$
   $Y = (y_1, y_2, \ldots, y_m)$
3. compute locally linear model

$$\beta = (X^T W X)^{-1} X^T W Y = P X^T W Y \tag{C.2}$$

4. the predicted output value is $\hat{y}_q = \beta_{n+1}$.

The vector β_{n+1} denotes the $(n+1)^{th}$ row of the regression matrix A. The parameters of the distance metric D are optimized using off-line cross-validation. This is done by quantizing the diagonal values of D on bounded sets which contain all admissible values for D diagonals. For each set D^l of quantized values on such collection, compute the estimation error for x_j, $j = 1, \ldots, p$ by removing point x_j from the training data (becoming $p - 1$ the number of points in the training data), and use such point as x_q. Compute $y_j = y_q$ using LWR with $D = D^l$. Sum the L_2 norm of all p estimation errors $\|y_q - \hat{y}_q\|_2$ for the same D^l, obtaining e_l as the sum. For all l, select the one which corresponds to a minimum e_l, and make $D = D^l$.

C.1.1 Truncating Receptive Fields

High dimensional topological spaces require an enormous amount of training points in order to reasonably cover the space of all admissible values. Referring to matrix β in equation (C.2), its computation requires determining $P^{-1} = (X^T W X)$. Since X is of size $p \times (n+1)$, performance degrades rapidly as p increases for very large values. Such problem can however by solved easily by truncating the receptive fields so that they contain only a fixed number of d points. The procedure for such is linear in p, and consists of selecting as training points the d points that are closer to the query point (in the L_2 sense), ignoring all other points. One can then apply the LWR algorithm with $p = d$.

C.1.2 LWR by Recursive Least Squares

Another alternative to speed up classification consists of computing iteratively the result for matrix A in equation (C.2) by recursive least squares (Atkeson et al., 1997; Slotine and Weiping, 1991). Given a point (x_i, y_i), the incremental step for updating β is:

$$\beta^{k+1} = \beta^k + w_{i,i} P^{k+1} \tilde{x}_i e^T \tag{C.3}$$

$$\begin{aligned} P^{k+1} &= \frac{1}{\lambda}\left(P^k - \frac{P^k \tilde{x}_i \tilde{x}_i^T P^k}{\frac{\lambda}{w_{i,i}} + \tilde{x}_i^T P^k \tilde{x}_i}\right) \\ e &= y_i - \beta^{k^T} \tilde{x}_i \end{aligned}$$

C.2 Receptive Field Weighted Regression

Receptive Field Weighted Regression (RFWR) implements function approximation by incrementally building a set of receptive fields (Atkeson et al., 1997). New receptive fields might be created or destroyed at each iteration to achieve a compromise between computational efficiency and space coverage. The normalized weighted sum of the individual predictions $\hat{y}_k$ of all receptive fields gives the predicted value for a query point x_q:

$$\hat{y} = \frac{\sum_{k=1}^{K} w_k \hat{y}_k}{\sum_{k=1}^{K} w_k} \tag{C.4}$$

where the weights w_k is the activation strength of the receptive field k. These weights are computed from equation C.1. The distance metric D_k is generated by an upper triangular matrix M_k in order to ensure matrix D_k is positive definite. A parametric linear function models the input/output transfer function, for each receptive field (local polynomials of low order are often a choice). The RFWR algorithm, as originally proposed (Atkeson et al., 1997), requires three auxiliary computations.

Computing the Size and Shape of the Receptive Field

Similarly to the iterative computation of the gaussian size in LWR, a cost function J_1 is used together with leave-one-out validation:

$$J_1 = \frac{1}{W}\sum_{i=1}^{p} w_i \parallel y_i - \hat{y}_{i,-i} \parallel^2 = \frac{1}{W}\sum_{i=1}^{p} \frac{w_i \parallel y_i - \hat{y}_i \parallel^2}{(1 - w_i \tilde{x}_i^T P \tilde{x}_i)^2} \tag{C.5}$$

where P is obtained from equation C.3. A penalty term is introduced to avoid an ever increasing number of receptive fields:

$$J = J_1 + \gamma \sum_{i,j=1}^{n} D_{ij}^2 = \frac{1}{W}\sum_{i=1}^{p} \frac{w_i \parallel y_i - \hat{y}_i \parallel^2}{(1 - w_i \tilde{x}_i^T P \tilde{x}_i)^2} + \gamma \sum_{i,j=1}^{n} D_{ij}^2 \tag{C.6}$$

where the strength of the penalty is given by γ. Cost function J is then used to adjust M by gradient descent with learning rate α (Atkeson et al., 1997):

$$M^{n+1} = M^n - \alpha \frac{\partial J}{\partial M} \tag{C.7}$$

Replacing J in C.7 by C.6, and by applying a suitable approximation to the derivative, M (and therefore D) is obtained iteratively (see (Atkeson et al., 1997) for a detailed description of this procedure).

Creation of Receptive Fields

In cases in which the activation output of all receptive fields do not exceed a threshold w_{gen} for a new training sample (x, y), a new receptive field is created. Indeed, for such cases, there is not a locally linear model with a center sufficiently close to this new point, which requires a new receptive field to provide a better cover for the training data. The center for the new receptive field is given by $c = x$, and matrix M is set to a chosen default value ($M = M_{def}$), being all parameters zero except for P. A diagonal matrix is an appropriate initialization for P, being the diagonal elements given by $P_{ii} = 1/r_i^2$, with small coefficients r_i (e.g., 0.001). Such coefficients are updated by including the ridge parameters as adjustable terms in RFWR applying gradient descent in the cost C.6:

$$r^{n+1} = r^n - \alpha_r \frac{\partial J}{\partial r} \tag{C.8}$$

The change in r is added to P after each update of the latter (see (Atkeson et al., 1997) for details).

Removal of Receptive Fields

A receptive field is pruned

- for computational efficiency, if there is too much overlap with another receptive field (Atkeson et al., 1997)

- for excessive errors in w_{MSE} when compared to other units, where the bias-adjusted weighted mean squared error w_{MSE} of the linear model is given by:

$$w_{MSE} = \frac{E^n}{W^n} - \gamma \sum_{i,j=1}^{n} D_{ij}^2 \tag{C.9}$$

where E and W are given by the "memory traces" (Atkeson et al., 1997):

$$W^{n+1} = \lambda W^n + w \tag{C.10}$$
$$E^{n+1} = \lambda E^n + we^T e \tag{C.11}$$

The RFWR algorithm

To summarize, the RFWR pseudo-code is:

Given:

▷ Initialize the RFWR with no Receptive Field (RF)

▷ **For** every new training sample (x, y):

1. **For** k=1 to #RF:
   - calculate the activation from C.1
   - update receptive field parameters according to C.7, and C.8

   end;
2. **If** all RF activations do not exceed w_{gen}:
   - create a new RF with the new center $c = x$, and $M = M_{def}$

   end;
3. **If** two RFs have higher activation than w_{prune}:
   - erase the RF k for which the determinant of D_k is larger

   end;
4. calculate the $m = E\{wMSE\}$ and $\sigma_{std} = E\{(wMSE - m)^2\}^{0.5}$ of all RFs
5. **For** k=1 to #RF:
   - If $|w_{MSE}m| > \varphi\sigma_{std}$ (scalar φ is a positive outlier removal threshold)
     * re-initialize the receptive field with $M = \varepsilon M_{def}$

     end;

   end;

end;

Appendix D

Neural Oscillators for the Control of Rhythmic Movements

Matsuoka neural oscillators (Matsuoka, 1985, 1987) are an elegant solution for the control of rhythmic movements that exploit the linking between biomechanics and neuroscience. Having a highly nonlinear dynamics, their parameters are difficult to tune, and other available tuning methods are based on simulation programs (Williamson, 1999). This appendix describes in detail an analytical analysis of neural oscillators, both in isolated and coupled situations, by using multiple input describing functions that allow the designer to select the parameters using algebraic equations, therefore simplifying enormously the analysis of the system motion. Furthermore, robustness and stability issues are easily handled using this methodology. A complementary analysis on the time domain is also presented.

The non-linear equations describing neural oscillators depend on parameters that have to be tuned to mimic a desired biological oscillator. The frequency and amplitude of oscillations of the different physical variables controlled depend on the value of these parameters. Therefore, many problems need yet to be solved: determination of algebraic equations for parameter tuning, description of range of parameters corresponding to different modes of oscillation of the neural oscillator, and stability analysis of the results. Furthermore, parameter tuning and stability determination for MIMO systems connected to multiple neural oscillators is still lacking theoretical support. Under some filtering assumptions (Arsenio, 2000c), we propose the analysis of one oscillator connected to both linear and nonlinear systems, using a multiple-input describing function technique to model the nonlinearities, as will be described. Algebraic equations are used to (1) determine the frequency and amplitude of os-

cillation given oscillator parameters and a dynamical (non)linear system, and (2) determine the oscillator's parameters required to achieve a desired performance. The methodology is extrapolated in a natural way to multiple oscillators connected to MIMO (non)linear systems, and the number of equations to be solved increases linearly with the number of neural oscillators. This way, the design of a network of neural oscillators for the control of a complex dynamic system is simple.

The analysis hereafter presented consists of original work which 1) brings more insight into how neural oscillators - consisting on mutually inhibiting (Matsuoka, 1985) neurons - operate, 2) contributes to their understanding and range of applications, and consequently to the comprehension of the control of similar neural biological structures. Hence, this appendix introduces original work.

D.1 Matsuoka Neural Oscillators

The Matsuoka neural oscillator consists of two neurons inhibiting each other mutually. Each neuron i has two states variables, x_i and v_i, and they are equal for a zero external input g, as illustrated by a diagram of the oscillator in figure 9-1. The nonlinear dynamics of the neural oscillator follows the following equations:

$$\tau_1 \dot{x}_1 = c - x_1 - \beta v_1 - \gamma y_2 - \sum_i k_i n^+[g_i] \tag{D.1}$$

$$\tau_2 \dot{v}_1 = y_1 - v_1 \tag{D.2}$$

$$\tau_1 \dot{x}_2 = c - x_2 - \beta v_2 - \gamma y_1 - \sum_i k_i n^-[g_i] \tag{D.3}$$

$$\tau_2 \dot{v}_2 = y_2 - v_2 \tag{D.4}$$

where c is a positive tonic input, τ_1 and τ_2 are positive time constants, β, γ (usually both positives) and $k_i \geq 0$ are weights, and g_i is an external input to the oscillator. There are two types of nonlinearities: $n(u) = n^+(u) = max(u, 0)$, and $n^-(u) = -min(u, 0) = max(-u, 0)$, with u being the nonlinearity input. The output of each neuron is given by

$$y_i = n(x_i) = max(x_i, 0), i = 1, 2 \tag{D.5}$$

and the output of the oscillator is calculated from (D.6).

$$y_{out} = y_1 - y_2 \tag{D.6}$$

This work analysis Matsuoka neural oscillators connected to stable dynamical systems. However, these systems may be unstable if they can be stabilizable by a controller (as demonstrated by experiments carried out under this work for an inverted pendulum controlled by a LQR controller, and the input of the controller is connected to the neural oscillator).

D.1.1 Describing Functions

Frequency domain methods may not be applied directly to nonlinear systems because frequency response functions cannot be defined for these systems. However, the output of a nonlinear element to a bounded input signal may be approximated by a linear operation for specific forms of input signals. Driving the same linearity by inputs of different forms (or same form and different amplitudes) results in different linear approximations. These quasi-linear approximating functions (Gelb and Vander Velde, 1968; Slotine and Weiping, 1991), that describe approximately the transfer characteristics of the nonlinearity, are called describing functions (DF). Describing function analysis is only valid if the input signal form is similar to the real system signal, and if the linear element of the system has low-pass filter properties. This last requirement is due to the nature of describing functions, obtained by expanding the output signal in a Fourier series, and by neglecting harmonics different from the fundamental. Thus, given an odd periodic input *x(t)* (which implies no DC output terms) and a nonlinearity $n(x)$, the output $y(t) = n(x(t))$, after expanding in a Fourier series and removing harmonics different from the fundamental, is defined as:

$$y(t) = N_{a_i} cos(wt) + N_{a_r} sin(wt) \tag{D.7}$$

where $N_{a_r}(A, w) = \frac{1}{A} b_1$ and $N_{a_i}(A, w) = \frac{1}{A} a_1$. The Fourier coefficients $a_1(A, w)$ and $b_1(A, w)$ are determined by

$$a_1 = \frac{1}{\pi} \int_{-\pi}^{\pi} y(t) cos(wt) d(wt) \tag{D.8}$$

$$b_1 = \frac{1}{\pi} \int_{-\pi}^{\pi} y(t) sin(wt) d(wt) \tag{D.9}$$

For the analysis of neural oscillators, it is important to consider other class of input functions defined by $x(t) = B + A sin(wt)$, consisting of an odd periodic function plus a bias B. The quasi-linearization of the nonlinearity for this input is given by (with $j = \sqrt{-1}$):

$$N_a(A, B, w) = N_{a_r}(A, B, w) + N_{a_i}(A, B, w) j \tag{D.10}$$

$$N_b(A, B, w) = \frac{1}{2\pi B} \int_{-\pi}^{\pi} y(t) cos(wt) d(wt) \tag{D.11}$$

The describing function $N_a(A, B, w)$ is applied to the periodic term of the input, while $N_b(A, B, w)$, the approximating gain to the bias input, is applied to the bias term. Together, $N_a(A, B, w)$ and $N_b(A, B, w)$ define the Dual Input Describing Function (DIDF), which approximates the nonlinearity when the input is composed of a bias plus a sinusoidal signal. Considering the nonlinearity $[n]^+$, and expanding the output in a Fourier series, for an input $x = B + A sin(wt)$, and for $|B| \leq A$:

$$N_a(A,B) = \frac{1}{2} + \frac{1}{\pi}\left[\arcsin\frac{B}{A} + \frac{B}{A}\sqrt{1-\left(\frac{B}{A}\right)^2}\right] \tag{D.12}$$

$$N_b(A,B) = \frac{1}{2} + \frac{1}{\pi}\frac{A}{B}\left[\frac{B}{A}\arcsin\frac{B}{A} + \sqrt{1-\left(\frac{B}{A}\right)^2}\right] \tag{D.13}$$

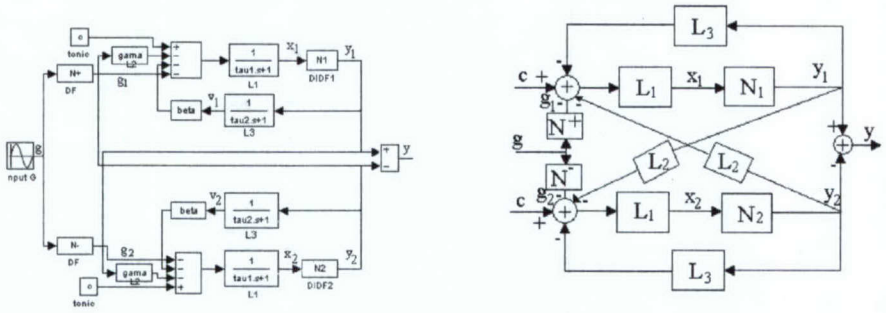
If $|B| > A$ then $N_a(A,B) = \begin{cases} 1\, for\, B > A \\ 0\, for B < -A \end{cases}$

However, for $|B| > A$, the output of the neural oscilator is zero, because x_1 and x_2 never change sign. Assuming that the forced input g of the neural oscilator has no DC component, a simple describing function is enough to approximate this nonlinearity transfer function, for an input $g = Dsin(wt)$:

$$N_a(D) = \frac{1}{2},\ N_b(D) = \frac{1}{\pi}.$$

There are three main methods of computing describing functions:

- ▷ Experimental evaluation, that requires specialized instrumentation to compute the describing function of a nonlinear element based on the response to harmonic excitation.
- ▷ Numerical integration may be used to evaluate the describing function of $y = n(x)$. This way, graphs or tables may be available for different input values. Recent design of rhythmic motions for neural oscillators (Williamson, 1999) applies this methodology. The main advantages of this method is that is relatively easy to determine stability and robustness properties using graphical tools for different input amplitude and frequency values. One pitfall is the limited application to single input systems, being not practical for analyzing MIMO systems. Furthermore, when the input possess a bias term, the plots will depend on three terms: A, B and w, which will increase considerably the number of operations. Indeed, same of the Matsuoka neural oscillators state variables have a bias term. However, the output of the oscillator does not present that term, which eliminates the necessity of determining B by this method when evaluating the describing function between the input and the output of the neural oscillator. But, of course, this means that the internal dynamics of the oscillator is lost, and the oscillator has to be analyzed as a black box. Another important drawback is the necessity of an interpolation procedure to determine the frequency and the amplitude of output oscillations (since there is only available discrete data), and the long time required to build even a sparse table.
- ▷ Analysis of the Matsuoka neural oscillator using analytical calculations, which was proposed in (Arsenio, 2000c,a,b, 2004c) and is presented here. This method may also be applied, in a straighforward manner, to other types of oscillators, such as the Van Der Pool oscillator. The neural oscillator is quasi-linearly



a) b)

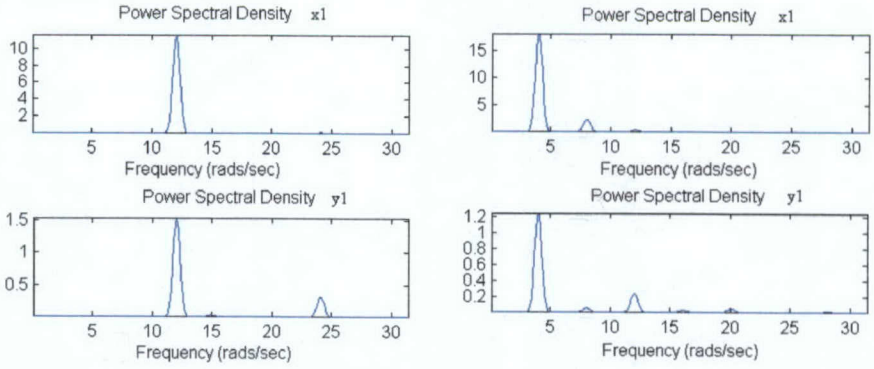
Figure D-1: a) Neural oscillator as a dynamical system. Nonlinearities are approximated by DFs, and analysis in the frequency domain is possible, by applying a Fourier transform. b) Block diagram of the oscillator.

approximated with simple and dual input describing functions, which provide a tool for determining the amplitude and frequency of the output using simple algebraic equations. Furthermore, this method allows not only the analysis of the oscillator internal states, but also a straighforward way for testing the filtering assumption for the linear element. Instead of determining graphically a describing function for all the oscillator, it is presented the determination of dual input describing functions for each of the nonlinearities. The analysis of coupled and/or MIMO systems is also possible using this methodology, which makes possible the analysis of multiple oscillators connected to high-order systems.

D.2 Design of Oscillators

In this section, an analytical procedure is described to determine the equations that govern the quasi-linearized system. Observing figure D-1-a, it is possible to verify the filtering hypothesis, i.e., that the output of a nonlinearity internal to the oscillator is filtered by two low-pass filters. Indeed, if the frequency of oscillation is larger than $\max(\frac{1}{\tau_1},\frac{1}{\tau_2})$ (the bandwidth of the low-pass filters L_1 and L_3, respectively), then higher harmonics are removed. This situation is illustrated in figure D-2-a. The outputs y_1 and g_1 of the nonlinearities N_1 and N^+, respectively, are both filtered, and no higher harmonics exist in x_1, although they do have a significant weight in y_1. On the other hand, if the bandwidth of the filter is smaller than the oscillation frequency, then the DF approximation introduces large errors, as attested in figure D-2-b, because x_1 contains higher harmonics, which violates the filtering assumption.

Exploiting symmetry between x_1 and x_2, and observing figure D-1, one notes that neither N_i, for $i = 1, 2$, nor N^+, N^- and L_2 depend on the frequency of oscillation, and therefore do not introduce any phase shift. Therefore, for the system to oscillate, the phase shift between x_1 and x_2 must be a multiple of π. Solving for the case of



a) b)

Figure D-2: a) Bandwidth $L_1 < w$. Although y_1 (not filtered) contains high harmonics, x_1 (filtered) only contains the fundamental. b) $L_1 > w$, so that x_1 now contains higher harmonics. System simulated using the MATLAB simulink control box.

no phase shift, there are no solutions that allow the system to oscillate (this solution originates $w_{n_{osc}}^2 < 0$, which is impossible). However, considering a phase shift of π between x_1 and x_2, results the equivalent approximate system (see figure D-3), solved in terms of x_1 (with $x_2 = 2B - x_1$, from $x_1 - B = -(x_2 - B)$):

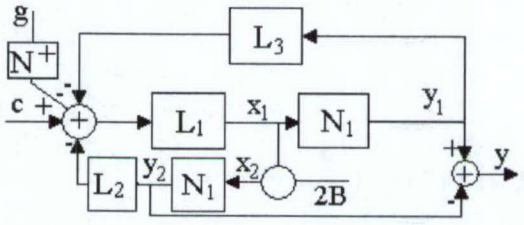


Figure D-3: Equivalent system when high harmonics are negligible, and thus $x_2 = 2B - x_1$.

$$x_1 = L_1 \frac{c - N^+ g - 2L_2 BN}{1 + L_1(L_3 - L_2)N} \tag{D.14}$$

This approximation is valid only if x_1 and x_2 are sinusoidal,i.e., if there are no harmonics higher than the fundamental.

If the input signa g contains a zero frequency component, then four DIDFs will be used to characterize the four nonlinearities and the bias component of x_1 and x_2 will differ by $|g| = D$, being the analysis similar.

D.2.1 Free Vibrations

The neural oscillator, given certain ranges of the parameters, oscillate without no forced input. Since the poles of L_1 and L_3 are stable, the open loop system has

no poles on the right half complex plane. Therefore, it is possible to determine the frequency of natural oscillation for the oscillator from the Nyquist criterion,

$$1 + L_1(L_3 - L_2)N_a = 0$$

Replacing expressions (see figure D-1),

$$\frac{1}{\tau_1 wj + 1}(\frac{\beta}{\tau_2 wj + 1} - \gamma)N_a = -1$$

and solving, results

$$\frac{j[-\gamma\tau_2 wp] + [(\beta - \gamma)p - \gamma(\tau_1 + \tau_2)w^2]}{(\tau_1 + \tau_2)^2 w^2 + p^2} = -\frac{1}{N_a} \tag{D.15}$$

with $p = 1 - \tau_1\tau_2 w^2$. Therefore, for (D.15) to be satisfied, it is necessary that the imaginary part of the left side is zero, which originates the following result for the Matsuoka neural oscillator: the natural frequency of oscillation of the Matsuoka oscillator is independent of the multiple nonlinearities contained on it. Indeed, the natural frequency w is obtained by:

$$-\gamma\tau_2 w(1 - \tau_1\tau_2 w^2) = 0 \;\equiv\; w_{n_{osc}} = \frac{1}{\sqrt{\tau_1\tau_2}} = \frac{1}{\sqrt{k}\tau_1} \tag{D.16}$$

considering $\tau_2 = k\tau_1$. Solving now for the real part of (D.15), and knowing that $w_{n_{osc}}^2 = \frac{1}{\tau_1\tau_2}$, results:

$$N_a = \frac{\tau_1 + \tau_2}{\gamma\tau_2} = \frac{1 + k}{\gamma k} \tag{D.17}$$

Remains to solve for the input bias. However, there exists an artificial input c, and thus the following equation must be satisfied.

$$B(1 + L_1(jo)(L_3(J0) - L_2(j0))N_b) = L_1(j0)(c - 2BL_2(j0)N_b)$$

The bias gain of the describing function is therefore:

$$N_b = \frac{c/B - 1}{\beta + \gamma} \tag{D.18}$$

If $B = 0$, then $N_b B = \frac{1}{2}A$, and thus $\frac{c}{\beta+\gamma} = \frac{1}{2}A$.

Thus, (D.12), (D.13), (D.16), (D.17) and (D.18) are used to determine analytically w, N_a, N_b, the bias B and amplitude A of the input sinusoid x_1 and x_2. From

$$y = x_1 - x_2 = 2N_a(A, B, w)Asin(t)$$

results that the output amplitude is $A_y = 2N_a A$, and all the system is solved, requiring the solution of five equations, being only one required to be solved by a numerical algorithm. Figure D-4-a plots the estimates A, B and A_y versus the measured using simulations, varying the tonic input c. As was expected, for $c = 0$ no oscillations occur, and the oscillator converge to the origin, the unique equilibrium point. For $c > 0$, the amplitude of the oscillations increase with c. As expected, $w_{n_{osc}} \propto 1/\tau_1$, as shown in figure D-4-b.

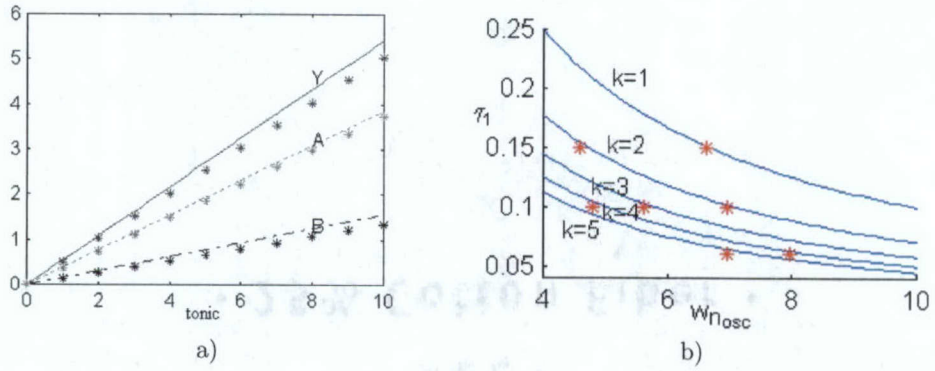


Figure D-4: a) Variation of x_1 amplitude - A, bias B and output amplitude A_y with the tonic input c. The stars represent the measured values using the simulink box for the simulations. This measurements are subject to a gain error because of no low-pass filtering. b) Determination of τ_1 as a function of $w_{n_{osc}}$.

D.2.2 Forced Vibrations

The response of linear systems to a forced sinusoidal input is the sum of the free vibration response (with no input), and the forced response (considering zero initial conditions). However, for nonlinear systems superposition does not holds. Even when approximating the nonlinearities by describing functions, these depend on the amplitude of the input signal, and the nonlinearity output will be a complex function of A, B and C, where A and C are the amplitude of the forced response and free vibration sinusoids, respectively, for a nonlinearity input $x = B + Csin(w_{n_{osc}}t) + Asin(wt + \phi)$, where the forced input is $g = Dsin(wt)$. Therefore, the analysis of the system would require a tri-input describing function. This would be complex, and it would consist of transfer functions for A, B and C. However, some assumptions may be considered to simplify the analysis. For $w \neq w_{n_{osc}}$, the free response decays to zero with time, and therefore, after a transient, only the force response remains. Since the interest is the determination of limit cycles parameters and oscillations stability, the analysis may be carried out neglecting the transient (especially because the oscillator converges rapidly). Thus, considering (D.14), the forced response to a sinusoidal input is given by

$$A = \frac{-L_1 \frac{1}{2}}{1 + L_1(L_3 - L_2)N_a} D = L(jw)D \tag{D.19}$$

where L(jw) is described by

$$L(jw) = \frac{1}{2}\frac{1}{\tau_1 wj + 1 + (\frac{\beta}{\tau_2 wj+1} - \gamma)N_a}$$

Since the system oscillates at w, there will be a phase shift and a gain between x_1 and g (and therefore between y and g). Thus, (D.20) will reflect the gain between the two signals:

$$A = \sqrt{imag(L(jw))^2 + real(L(jw))^2 D} \tag{D.20}$$

Simplifying (D.20), results finally:

$$N_a^2 a_l + N_a b_l + c_l l - \frac{D^2}{4A^2} = 0 \tag{D.21}$$

$$\begin{aligned} a_l &= (-\gamma + \tfrac{\beta}{\tau_2^2 w^2+1})^2 + \tfrac{\beta^2 \tau_2^2 w^2}{(\tau_2^2 w^2+1)^2} \\ b_l &= -2\gamma + 2\beta \tfrac{1-\tau_2 w}{\tau_2^2 w^2+1} \\ c_l &= \tau_1^2 w^2 + 1 \end{aligned}$$

Furthermore, the bias has to be maintained over a complete cycle, requiring that:

$$B = \frac{(c - 2BN_b L_2(j0) - \frac{1}{\pi} D) L_1(j0)}{1 + L_1(j0)(L_3(j0) - L_2(j0))N_b} = \frac{c - \frac{1}{\pi} D}{1 + (\beta + \gamma) N_b} \tag{D.22}$$

The four unknowns A, B, N_a and N_b are the solutions of the four equations (D.12), (D.13), (D.21) and (D.22). The phase shift ϕ between y and g is determined by:

$$\phi = \pi - atan2[imag(2N_A L(jw)), real(2N_A L(jw))] \tag{D.23}$$

Figure D-5-a shows results varying the amplitude. As expected, A_y is barely affected with the input amplitude. The variation of the output amplitude with input frequency and amplitude is plotted in figure D-5-b. The results match simulation measurements (table D.1), although errors are introduced at the output signal because the filtering assumption is not satisfied.

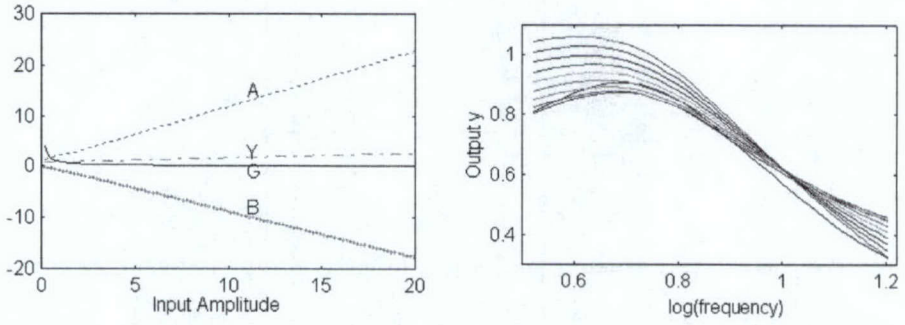
Figure D-6 shows the variation of gain and phase, respectively, with input frequency and amplitude.

D.2.3 Entraining Oscillators

The neural oscillator's output can be connected to the input of a second order system, and vice-versa. As corroborated by experimental data, shown in figure D-7-a, the frequency of oscillation w is never smaller than the natural frequency $w_{n_{osc}}$ of the oscillator. Furthermore, the oscillator entrains the natural frequency of the system w_n, for $w_n \geq w_{n_{osc}}$. This way, the system is driven in resonance, which corresponds to a small amount of actuator energy required to drive the system. In addition,

D	w=4	w=7	w=10	w=12	w=14
1.5	.97	1.03	.85	.7	.6
	−.45	−.48	−.32	−.17	−.08
	.52	.55	.53	.53	.51
	.35	.37	.35	.35	.34
	161	146	118	115	96
4.5	2.33	2.21	1.86	1.65	1.45
	−1.77	−1.66	−1.32	−1.11	−.95
	.56	.56	.56	.54	.5
	.12	.12	.12	.12	.11
	160	144	122	111	104

Table D.1: Table for simulation measurements using MATLAB. The vector shown corresponds to $[A, B, A_y, Gain, Phase]^T$.



a) b)

Figure D-5: a) Oscillation estimates varying input amplitude. The numerical equation was solved for all the points using an iterative algorithm to determine close initial conditions so that the error of the numerical calculation is small. b) Variation of the output amplitude with input frequency and amplitude.

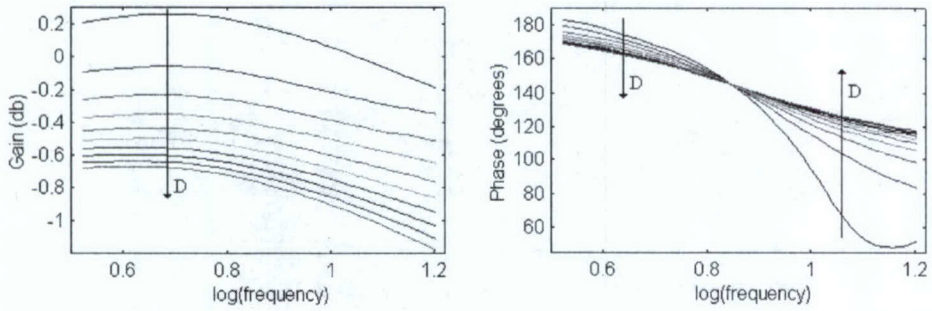
$w = w_n$ verifies the filtering condition, since the oscillator's output frequency is equal to the filter bandwidth, therefore suppressing the higher harmonics of y. The internal oscillator dynamics filtering assumption is not satisfied only for a very small range of frequencies $w_{n_{osc}} \leq w \leq 1/\tau_1$.

Let us consider, without loss of generalization, a second order system $m\ddot{\theta}+b\dot{\theta}+k\theta = ky$, with natural frequency $w_n = \sqrt{k/m}$, and its transfer function,

$$\Theta(jw) = G(jw)Y(jw) = \frac{k}{k - mw^2 + bwj}Y(jw) \tag{D.24}$$

After connecting this system to an oscillator, the close-loop equation is (D.25) for a sinusoidal signal,

$$1 + L_1(L_3 - L_2 + G)N_a = 0 \tag{D.25}$$



a) b)

Figure D-6: Variation of the gain a) and phase shift b) with input frequency and amplitude.

obtained by using (D.52) and

$$y = 2N_a \frac{-L_1 \frac{1}{2}}{1 + L_1(L_3 - L_2)N_a}\theta = 2N_a L(jw, N_a)\theta = . \tag{D.26}$$

Taking the imaginary parts of (D.25),

$$\frac{1}{\tau_1 wj + 1}(\frac{\beta}{\tau_2 wj + 1} - \gamma + \frac{k}{k - mw^2 + bwj})N_a = 0 \tag{D.27}$$

it is possible to obtain w, since N_a is real, and therefore it disappears from the equation. Not only the imaginary part must be zero for the system to oscillate, but also the real part:

$$real(1 + L_1(L_3 - L_2 + G)N_a) = 0 \tag{D.28}$$

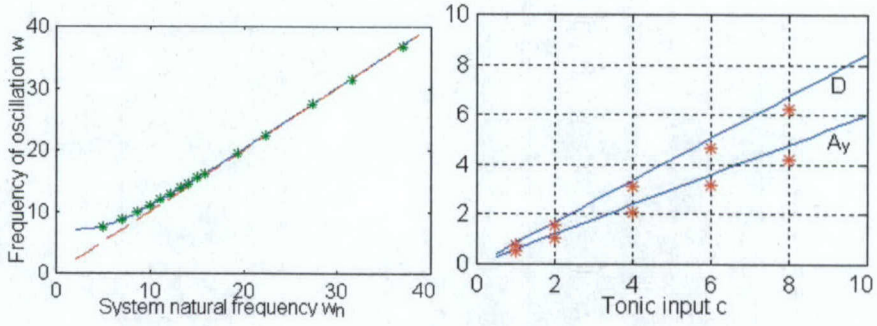
From (D.26), it is possible to extract another relation:

$$D = |G|2N_a A = \frac{k}{\sqrt{((k - mw^2)^2 + b^2 w^2}} 2NaA \tag{D.29}$$

Solving the close-loop system for a DC signal, results (D.22), where D (the amplitude of θ) is given by (D.29). Therefore, there are six equations to solve for six unknowns. If w, A, B, D, Na and N_b are considered as the unknowns, then they are obtained from (D.12), (D.13),(D.22),(D.27), (D.28) and (D.29), as plotted in figure D-8. The phase shift is

$$\phi = atan2(bw, k - mw^2). \tag{D.30}$$

If the designer wants to specify a given D, then the equations may be solved for the other five unknowns and for a parameter, such as the tonic necessary to maintain such oscillation, as was shown in figure D-7-b. The same reasoning applies to other parameters.



a) b)

Figure D-7: a) Frequency of oscillation for the close-loop system as the system natural frequency varies. The oscillator entrains w_n for a large range of frequencies. For very high or small values of w_n the oscillator frequency does not entrains w_n. b) Variation of the amplitudes D and $A_{y_{osc}}$ with the tonic c ($k = 30$, $m = 0.4$, $b = 2$). Gain $D = 0.8434$, while the measured gain is $D = 0.778$, and the gain $A_y = 0.525$, while the measured is 0.6. The gain error is because $w < 1/\tau_1$.

D.2.4 Connecting a Nonlinear System

The neural oscillator is now connected to a non-linear second order system, with linear part $L_1(jw)$ and nonlinear part $N_1(jw)$. As a result, seven equations need to be solved, being the additional equation the one corresponding to the describing function of the additional nonlinearity. Thus, the close-loop system contains five nonlinear elements for this case. Five of the previous equations remain the same, namely D.12, D.13, D.25, D.28 and D.22. However, the system transfer function G is,

$$G(jw) = \frac{L_1(jw)}{1 + L_1(jw)N_1(jw)},$$

and the equation (D.29) is replaced by (D.31).

$$D = |G|2N_a A = 2\left|\frac{L_1(jw)}{1 + L_1(jw)N_1(jw)}\right| NaA. \tag{D.31}$$

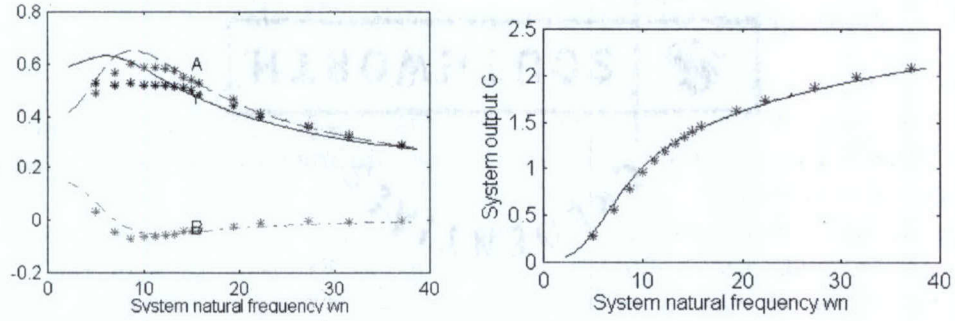
For a simple pendulum, the differential equation is,

$$\ddot{\theta} + k_1\dot{\theta} + \frac{k}{ml}\theta = \frac{k}{ml}y - gsin(\theta)$$

where g is the gravity acceleration, and $k_1 = \frac{b}{ml}$. The linear element is given by

$$L_1(jw) = \frac{1}{k/(ml) - w^2 + k_1 wj}.$$

The nonlinear element is the product of a periodic signal with a gain. The Fourier series expansion gives



a) b)

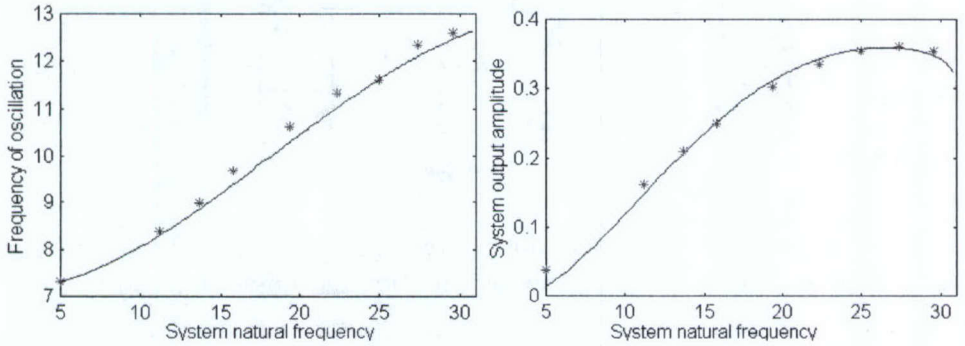
Figure D-8: a) Estimated and measured values, varying w_n, for A, B and A_y and b) for the system output. This estimate is very precise for $w > 1/\tau_1$, because the system acts as a low-pass filter.

$$N_1 = 2J(D)/D \tag{D.32}$$

where $J(D)$ is the Bessel function for first order arguments, (Gelb and Vander Velde, 1968). Simplifying for the transfer function G, (D.31) originates (D.33).

$$D = |G|2N_a A = 2\left|\frac{k/(ml)}{-w^2 + \frac{k}{ml} + k_1 wj + 2g\frac{J(D)}{D}}\right| NaA. \tag{D.33}$$

Therefore, seven equations are used to solve for the same number of unknowns (the additional unknown is N_1, the describing function of the system nonlinearity). Experimental results are shown in figure D-9.



a) b)

Figure D-9: Estimated (-) and measured (*) values for a) w and b) D varying k (nonlinear system).

D.3 Analysis of Multivariable Systems

Lets consider the analysis of a MIMO close-loop system, composed of n oscillators, which are coupled through the dynamical system $L(jw)$, but have no connections among them. However, the oscillators are coupled through the natural dynamics of the controlled system.

For MIMO systems, one has to deal with transfer matrices. The open loop transfer matrix for a multivariable system controlled by oscillators is given by the product of the matrices $N(jw)L(jw)$, where $N(A, jw) = diag(N_i)(A, jw)$, for $i = 1, \cdots n$, represents here the dynamics of all the oscillator, and $L(jw)$ represents the transfer matrix of a MIMO system. $N(A, jw)$ and $L(jw)$ are n by n matrices, with n being the number of inputs. Denote P_u the number of unstable poles of the open loop system, obtained from the open loop characteristic equation. Applying the Nyquist criterion for MIMO systems, the system presents oscillations if

$$|I + N(jw)L(jw)| = -P_u. \tag{D.34}$$

For an n^{th} input/output system $L(jw)$ connected to n oscillators without mutual interconnections (but coupled through the system dynamics), the application of the Nyquist criterion is applied to check for the existence of oscillations using (D.34).

D.3.1 Networks of Multiple Neural Oscillators

Let us analyze two oscillators connected to a 4^{th} order system (for example, the linear part of a two-joint arm), consisting of a dynamical model of two masses connected by three springs and two dampers, as shown in figure D-10. The dynamical model is,

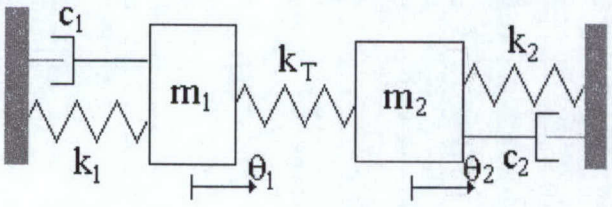


Figure D-10: Dynamical model composed of two masses connected by three springs and two dampers. There are two modes of vibration for masses m_1 and m_2.

$$\begin{aligned} m_1\ddot{\theta_1} + c_1\dot{\theta_1} + (k_1 + k_T)\theta_1 - k_T\theta_2 &= k_1 y_1 \\ m_2\ddot{\theta_2} + c_2\dot{\theta_2} + (k_2 + k_T)\theta_2 - k_T\theta_1 &= k_2 y_2 \end{aligned} \tag{D.35}$$

Without damping and oscillators connected, the natural modes of free vibration correspond to the solution of the equation:

$$\begin{bmatrix} w^2 + \frac{k_1+k_T}{m_1} & -\frac{k_T}{m_1} \\ -\frac{k_T}{m_2} & w^2 + \frac{k_2+k_T}{m_2} \end{bmatrix} \begin{bmatrix} \theta_1 \\ \theta_2 \end{bmatrix} = 0 \tag{D.36}$$

The two natural frequencies result from the fact that the determinant of the matrix in (D.39) must be zero, and thus

$$w^4 - w^2 \left(\frac{k_1 + k_T}{m_1} + \frac{k_2 + k_T}{m_2} \right) + \frac{k_1 k_2 + (k_1 + k_2) k_T}{m_1 m_2} = 0. \tag{D.37}$$

This equation leads to two values for w^2. Considering $\theta_i = A_{\theta_i} sin(wt)$, for $i = 1, 2$, the amplitude ratio is,

$$A_{\theta_1} / A_{\theta_2} = k_T / (-m_1 w^2 + k_1 + k_T) \tag{D.38}$$

The natural modes of motion are obtained by replacing in (D.38) each value for the natural frequency, and using the initial conditions.

Network Analysis 1. When only one oscillator is driving this dynamical system (e.g., $y2 = 0$), the amplitudes of oscillation will not depend on initial conditions, because the neural oscillator fixes the oscillations to a certain amplitude. Thus, considering now the damping (the oscillator is going to compensate for this damping), results for a sinusoidal signal:

$$\begin{bmatrix} P'_{11} - \frac{k_1}{m_1} L^1(jw, N_a^1) & -\frac{k_T}{m_1} \\ -\frac{k_T}{m_2} & P'_{22} \end{bmatrix} \begin{bmatrix} \theta_1 \\ \theta_2 \end{bmatrix} = P\Theta = 0 \tag{D.39}$$

where matrix $P = [P_{11} P_{12}; P_{21} P_{22}]$ and the elements P'_{ii}, for $i = 1, 2$, are given by:

$$P'_{ii} = jw \frac{c_i}{m_i} - w^2 + \frac{k_i + k_T}{m_i}$$

and $L^1(jw)$ is given by L(jw) in (D.26). The Nyquist criterion for MIMO systems states that the determinant of matrix P must cancel for free oscillations to occur. Therefore, both real and imaginary parts of the determinant must cancel, which introduces two more conditions for the imaginary and real parts of (D.40):

$$\left(P'_{11} - \frac{k_1}{m_1} L^1(jw, N_a^1) \right) P'_{22} - k_T^2 / (m_1 m_2) = 0 \tag{D.40}$$

Two other conditions result from the ratio amplitude: $P_{11} A_{\theta_1} + P_{12} A_{\theta_2} = 0$ - the other row is linearly dependent due to (D.40). Therefore, the imaginary and real parts of (D.41) have to cancel,

$$A_{\theta_1} \left(\frac{c_1}{m_1} wj - w^2 + \frac{k_1 + k_T}{m_1} - \frac{k_1}{m_1} L^1 \right) - A_{\theta_2} \frac{k_T}{m_1} = 0 \tag{D.41}$$

where $A_{\theta_1} = |\theta_1|$ and $A_{\theta_2} = |\theta_2|e^{j\varphi_2} = |\theta_2|cos(\varphi_2) + j|\theta_2|sin(\varphi_2) = A_{\theta_{2_{real}}} + jA_{\theta_{2_{imag}}}$. This results because of the phase-shift φ_2 between the system outputs. From (D.53), there is one more condition for the gains:

$$|A_{\theta_1}\left(\frac{c_1}{m_1}wj - w^2 + \frac{k_1 + k_T}{m_1}\right) - A_{\theta_2}\frac{k_T}{m_1}| = \frac{k_1}{m_1}2N_a^1 A_1 \tag{D.42}$$

One additional constraint results from the bias input to the oscillator internal state variables. The condition is therefore (i_c being the tonic of oscillator i, and A_{θ_i} the amplitude of the signal A_i):

$$B_1\left(1 + (\beta + \gamma)N_b^1\right) - 1_c + |A_{\theta_1}|/\pi = 0 \tag{D.43}$$

Finally, there are two more conditions given by (D.12) and (D.13) for N_a^1 and N_b^1, respectively. Therefore, eight equations are available to solve for eight incognits: A_1, B_1, A_{θ_1}, N_a^1, N_b^1, $|A_{\theta_2}|$, φ_2 and w. For the close-loop system, there is only one resonance mode, whatever the initial conditions, as shown in table D.2, since with $N_2 = 0$ results $1 + N_1 L_{11} = 0$, where N_1 is the DF of the oscillator and $w_{n_2}^2 = (k_2 + k_T)/m_2$ is the natural frequency of

$$L_{11} = \frac{k_1}{m_1}\frac{P_{22}}{\left(P_{11} - \frac{k_1}{m_1}L^1(jw, N_a^1)\right)P_{22} - k_T^2/(m_1 m_2)}.$$

	w	$\lvert A_{\theta_1}\rvert$	$\lvert A_{\theta_2}\rvert$	φ_2
Estimated	8.81	0.32	0.276	-130.6^o
Simulated	9.1	0.305	0.25	-131^o

Table D.2: Estimation versus measurement from simulation using MATLAB Simulink, with, for i=1,2, $c_{osc_i} = 1$, $\tau_{osc_i,1} = 0.1$, $\tau_{osc_i,2} = 0.2$, $\beta_{osc_i}\gamma_{osc_i} = 2$ (the oscillator parameters are equal for all the experiments in this manuscript), $m_i = 1$, $c_i = 3$, $k_T = 30$ and $k_1 = k_2 = 25$. The linear system has two natural frequencies: $w_{n_1} = \sqrt{k_1/m_1} = 5$, and $w_{n_2} = \sqrt{(k_2 + k_T)/m_2} = 9.22$. The manual measurements are subject to errors.

Network Analysis 2. Lets extend the analysis to n oscillators connected to a $(2n)^{th}$ order system (each oscillator has only one input, and no connections among them). If the outputs of the coupled linear system would oscillate at n natural frequencies, then the analysis of the oscillator would have to consider these frequencies (but considering that the principle of sobreposition does not holds). For example, for $n = 2$, it would be necessary a two-sinusoidal describing function to describe the input non-linearities, and a tri-input (one bias and two sinusoids) describing function to describe the internal nonlinearities of the neural oscillator. As a consequence, this would increase considerably the complexity of the closed-loop system, but could be implemented using a variation of the method of the describing function matrix,

(Mees, 1972), which determines higher order describing functions that approximate the nonlinearity to higher harmonics or to a vector input.

However, the neural oscillators, for a high enough k_T, only tracks one natural mode (see table D.3). Thus, the natural frequency at which the system oscillates will only depend on the initial conditions and system parameters. Intuitively, a high k_T implies that the masses are stiffly connected, so that they tend to oscillate together, on phase. For a small k_T, the neural oscillator may oscillate at several frequencies, but this corresponds to the case where the dynamics matrix P in (D.39) is approximately diagonal, meaning that the coupling between the neural oscillators through the natural dynamic may be neglected. Thus, the plant dynamics may be considered in such cases as n second-order systems, each one connected to one oscillator. In such a situation (for a 2^{th} order system), $w_{n_i} = (k_i + k_T)/m_i$, for $i = 1, 2$ (see table D.4).

As a matter of fact, outputs containing more than one resonance mode only occur for a small range of oscillators parameters in a MIMO system (and only for $w_{n_i} \approx w_{n_j}, \forall i, j$). Indeed, observations varying the parameters revealed that usually each oscillator tracks one of the n frequencies, as shown in figure D-11 for a 12^{th} dimensional system with two oscillators. It was also verified experimentally that, if the oscillator's input is a sum of n signals at n different frequencies, but the power spectrum at one frequency is significantly larger than at the others, then the oscillators' outputs oscillate only at one frequency and the describing function analysis still holds.

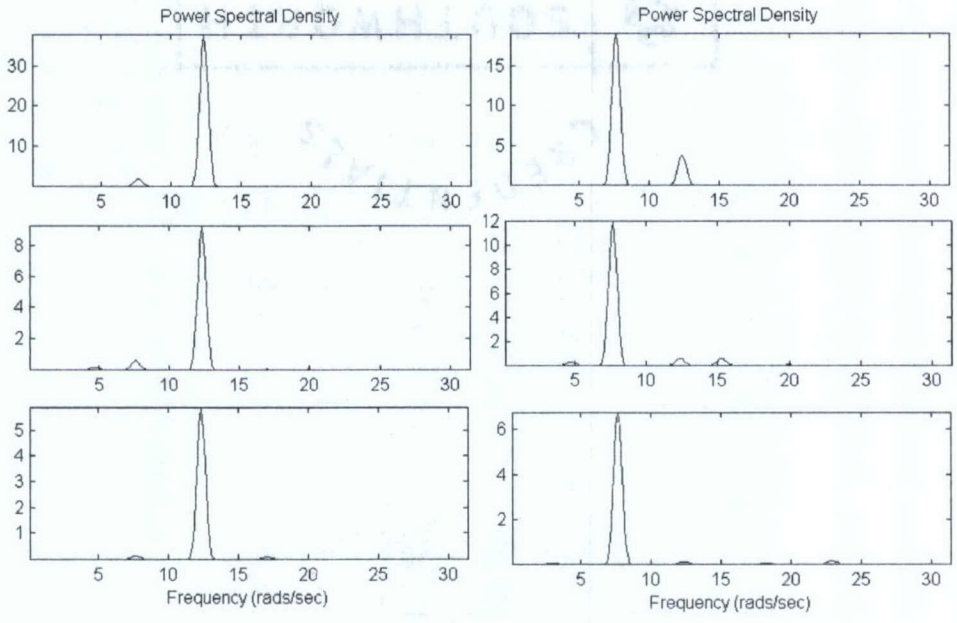


Figure D-11: Spectrum for the signals a) θ_1, x_1^1, y^1 and b) θ_2, x_2^1, y^2, respectively (where y^i stands for the output of oscillator i). The experiment was carried out for a non-symmetric MIMO system, and corresponds to a *worst case* analysis.

Simple describing functions are assumed for the nonlinear elements at oscillator's inputs and two input DFs for oscillator's internal nonlinearities, which is valid for high-order SISO systems, and for MIMO systems under the assumption that all the signals oscillate at the same frequency, which results from the fact that the outputs only track one resonance mode (demonstrated in Case 3).

Lets now consider $n = 2$, with both oscillators, θ_1 and θ_2 oscillating at the same frequency $w_1 = w_2 = w$. For sinusoidal signals, (D.39) still holds, but P_{22} is replaced by $P_{22} - \frac{k_2}{m_2} L^2(jw, N_a^2)$, where N_a^2 is the DF of the 2^{nd} oscillator for a sinusoidal signal.

The first two conditions are imposed by the Nyquist criterion for MIMO systems, and described by equation (D.40), for both real and imaginary parts (but with the new P_{22}). Two other conditions result from the ratio amplitude, and are given by (D.41), for both complex parts. From (D.53), two more conditions result: one is given by (D.42), and the other results from (D.44).

$$|A_{\theta_2}\left(\frac{c_2}{m_2}wj - w^2 + \frac{k_2 + k_T}{m_2}\right) - A_{\theta_1}\frac{k_T}{m_2}| = \frac{k_2}{m_2}2N_a^2 A_2 \tag{D.44}$$

Two additional constraints result from the bias input to the oscillators internal state variables x_1^i and x_2^i, for $i = 1, 2$ (for two oscillators). One is imposed by (D.43), and the other by (D.45).

$$B_2\left(1 + (\beta + \gamma)N_b^2\right) - 2_c + |A_{\theta_2}|/\pi = 0. \tag{D.45}$$

Finally, there are four more conditions given by (D.12) and (D.13) for N_a^1 and N_b^1, respectively, and for N_a^2 and N_b^2. Therefore, twelve equations are available to solve for twelve unknowns: A_1, B_1, A_{θ_1}, N_a^1, N_b^1, A_2, B_2, $|A_{\theta_2}|$, φ_2, N_a^2, N_b^2 and w, as supported by the experimental results shown in tables D.3 and D.4.

w_1	w_2	A_{θ_1}	A_{θ_2}	A_1	B_1	A_2	B_2
15.15	15.15	2.64	1.60	0.94	-0.39	0.61	-0.13
15.13	15.13	2.55	1.53	0.85	-0.35	0.59	-0.09

Table D.3: Experiment for $k_T = 300$ (high relative to k_i), and $k_1 = 100$, $k_2 = 400$. Estimation (1^{th} row) and measurement using MATLAB Simulink (2^{nd} row). The phase-shift between A_{θ_2} and A_{θ_1} is $\varphi_2 = 0^o$.

	w_{n_1}	w_{n_2}	$w_{1_{measured}}$	$w_{2_{measured}}$
$k_T = 3$	10.15	22.08	10.94	20.32
$k_T = 30$	11.4	20.74	11.9	20.99

Table D.4: Simulation measurements for $k_1 = 100$, $k_2 = 400$ and two relative small values of k_T. Each oscillator output oscillates at one different natural frequency, but only at that frequency. In this case there is no dependence on initial conditions.

Network Analysis 3 For $n = 2$ oscillators, one useful alternative analysis results because of total symmetry, i.e., the parameters of both oscillators are equal, as well the system parameters: $k_1 = k_2 = k$, $c_1 = c_2 = c$, and $m_1 = m_2 = m$. For free vibrations, the natural frequencies and corresponding natural modes are given by,

$$\begin{array}{ll} w_{n_1}^2 = k/m & w_{n_2}^2 = (k + 2k_T)/m \\ A_{\theta_1}/A_{\theta_2} = 1 & A_{\theta_1}/A_{\theta_2} = -1 \end{array}$$

Therefore, the imaginary parts of P must cancel for stable oscillations, because of the Nyquist condition,

$$w\frac{c}{m} + \frac{k}{m} imag(L) = 0 \tag{D.46}$$

and thus the condition for the determinant to cancel includes only real terms,

$$real(P_{11})real(P_{22}) - \frac{k_T^2}{m_1 m_2} = 0 \tag{D.47}$$

Experimental results are shown in table D.5 and figure D-12. Even without symmetry, the natural frequencies and natural modes for the system in free vibrations may be very useful as rude estimations, since the oscillator tracks these frequencies. Thus, using these values as the natural frequencies, it is possible to use the forced response results to estimate the amplitude of the oscillator's output y_{osc}. Indeed, as shown in figure D-6-a, this amplitude remains approximately constant with relation to input's amplitude variations A_θ. Thus, given the frequency, one may determine a rough value for the amplitude of y_{osc}.

	mode 1			*mode 2*		
k	w	A_{θ_1}	A_{θ_2}	w	A_{θ_1}	A_{θ_2}
50 E	8.41	1.0	1.01	10.88	0.84	-0.839
50 M	8.57	0.84	0.84	10.63	0.83	-0.83
100 E	10.83	1.608	1.608	13.04	1.26	1.26
100 M	10.88	1.5	1.5	13.08	1.29	-1.29
200 E	14.67	1.80	1.801	16.46	1.42	-1.42
300 E	17.73	2.26	2.26	19.27	2.05	-2.05
300 M	17.59	2.29	2.29	19.11	2.07	-2.07
400 E	20.34	2.23	2.23	0.574	-0.163	0.574
600 E	24.76	2.518	2.518	25.92	2.63	-2.63
900 E	30.21	2.64	2.64	31.17	2.62	-2.62
900 M	30.20	2.6	2.6	31.01	2.55	-2.55

Table D.5: Estimations (E) and measurements (M) with $k_1 = k_2 = k$ and $k_T = 30$, for several k.

Due to these simplifications, the oscillators' outputs y_{osc_i}, for $i = 1, \cdots, n$, as well as the frequency of oscillation, may be estimated (but not accurately) a priori.

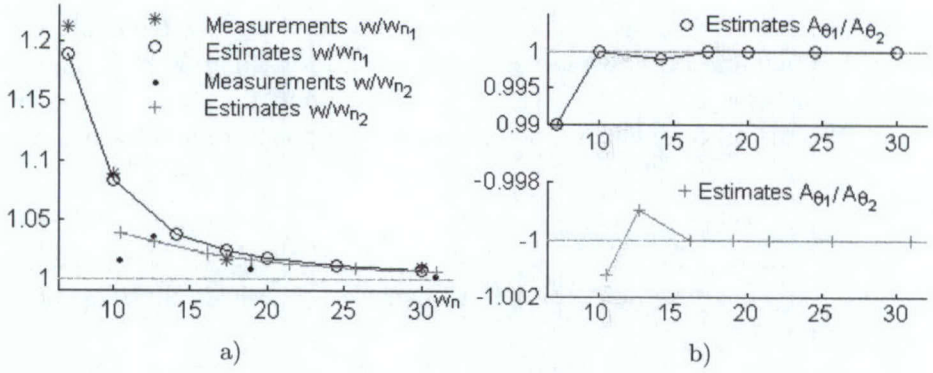


Figure D-12: a) Ratio between the two oscillatory frequencies and corresponding natural frequencies, for two different initial conditions. The horizontal axis represents the natural frequencies of the dynamic system. b) Amplitude ratio $A_{\theta_1}/A_{\theta_2}$ plotted versus the system natural modes.

Knowing the inputs to the controlled system and using (D.53), it is possible to get a rough estimate for the amplitude of oscillation of θ_i, for $i = 1, \cdots, n$. In addition, this simplified procedure may also be used to obtain an initial guess as initial conditions for the numerical solutions of the analitic equations.

Network Analysis 4 It is worth noticing that, for $n = 3$, corresponding to three neural oscillators connected to a 6^{th} order system, there are six more variables relative to $n = 2$, i.e., 18 unknowns. Two additional equations are obtained in a similar way to (D.41), but applied to the second row of the 3×3 matrix P in (D.39), for both complex parts. Other two conditions result from applying (D.44) to the third row of P, and from applying (D.45) to the third oscillator. The last two constraints are given by (D.12) and (D.13) for N_a^3 and N_b^3, respectively. Therefore, six additional constraints are available for the additional six unknows: A_3, B_3, $|A_{\theta_3}|$, φ_3, N_a^3 and N_b^3 (φ_3 is the phase-shift between θ_1 and θ_3). For a dynamical system given by the plant:

$$
\begin{aligned}
m_1\ddot{\theta_1} + c_1\dot{\theta_1} + (k_1 + k_T)\theta_1 - k_T\theta_2 &= k_1 y_1 \\
m_2\ddot{\theta_2} + c_2\dot{\theta_2} + (k_2 + k_T + k_{T_2})\theta_2 - k_T\theta_1 - k_{T_2}\theta_3 &= k_2 y_2 \\
m_3\ddot{\theta_3} + c_3\dot{\theta_3} + (k_3 + k_{T_2})\theta_3 - k_{T_2}\theta_2 &= k_3 y_3
\end{aligned}
$$

the system oscillates at one of the three plant resonance modes, as illustrates table D.6.

	w	A_{θ_1}	A_{θ_2}	A_{θ_3}	φ_2	φ_3
Est.	10.827	1.507	1.507	1.508	0.001	-0.006
Meas.	10.821	1.5	1.5	1.5	0.0	0.0

Table D.6: Estimations and simulation measurements with $k_i = k_T = 100$, $m_i = 1$ and $c_i = 3$, for $i = 1, \ldots 3$. The estimation error is very small.

Therefore, generalizing for a system with n oscillators coupled to a $2n$ dimensional linear system, the closed-loop system is $6n$ dimensional, and has $4n$ nonlinearities.

As verified until here, the number of equations necessary to solve the system is $6n$: 2 to cancel the imaginary and real parts of the determinant of the n^{th} square matrix P for a sinusoidal input, $2(n-1)$ to cancel the complex parts of (D.41), for the $n-1$ linearly independent equations:

$$A_{\theta_i}\left(P'_{i,i} - \frac{k_i}{m_i}L^i\right) + \sum_{l=1,l\neq i}^{n} A_{\theta_l}P_{i,l} = 0, \quad for\ i = 1,\ldots,n-1,$$

n to relate the amplitudes of the oscillators outputs with its inputs,

$$\left|P'_{ii} + \sum_{l=1,l\neq i}^{n} A_{\theta_l}P_{i,l}\right| = \frac{k_i}{m_i}N_a^i A_i, \quad for\ i = 1,\ldots,n-1,$$

n for the close-loop system to maintain the same bias B_i, and $2n$ given by the describing functions for a sinusoidal input and for the bias. If the oscillators are coupled to a nonlinear system which has m nonlinearities, then the total number of nonlinearities in the close-loop system is $4n+m$, and the number of variables and equations is $6n+m$. The additional variables are $N_{D_{\theta_i}}$, the gain of the describing function, and the additional equations are the describing function equations (Arsenio, 2000c).

It is also worth demonstrating, for the same n oscillators, that the amplitude of oscillation of the natural modes do not depend on initial conditions. Initial conditions only affect the mode to which the neural oscillator is attracted. For k_T negligible, the result is immediate, since the system is decoupled and thus the oscillators outputs y_i oscillates only at w_{n_i}, for i=1,...,n. For a significant k_T, the neural oscillators are coupled through the system natural dynamics. For linear systems, it is well known that for stable oscillations, $|\underline{G}_{CloseLoop}(jw)| = 0$, and thus this condition is independent of $\underline{A_\theta} = [\ A_{\theta_1}\ \ \ldots\ \ A_{\theta_n}\ ]^T$. There are $2n$ unknowns ($\underline{A_\theta}; \varphi_i, for\ i = 1,\ldots,n; w$), and only two conditions for the complex parts of the determinant. One is used to determine the natural frequencies, and the other the amplitudes ratio (or natural modes $\underline{A_\theta}^i$), and thus the system output is (depending on the initial conditions):

$$\underline{\theta} = \underline{A_\theta}^1 e^{jw_{n_1}t} + \ldots + \underline{A_\theta}^n e^{j(w_{n_n}t+\varphi_n)}.$$

For nonlinear systems, $|\underline{G}_{CloseLoop}(\underline{A_\theta}, jw)| = 0$, and thus the determinant is now dependent on $\underline{A_\theta}$. Therefore, the complex parts of the rows of $\underline{G}_{CloseLoop}(\underline{A_\theta}, jw)\underline{A_\theta}$ must all cancel for the system to oscillate, which imposes $2n$ conditions for $2n$ unknowns. Therefore, the amplitudes $\underline{A_\theta}$ are well determined, and their value corresponds to the natural mode. Thus, the system oscillates at a unique frequency. The mode entrained may depend on the initial conditions or not, depending on the parameters, but the amplitude of oscillation is fixed, independent of initial conditions.

D.3.2 Networks of Interconnected Oscillators

Assume connections between a network of oscillators coupled to a system. The oscillators may oscillate at different frequencies, and not being entrained with the system

and with each other, and even beating may occur. An analysis similar to the previous subsection could even be applied for $w_i \neq w_j$, for $i \neq j$, maintaining the same assumptions concerning the power spectrum of oscillator's inputs.

When the oscillators are entrained with each other by input connections to all the coupled system' states, the analysis is similar to the one already presented for MIMO systems, but constrained to the fact that all the oscillators oscillate at the same frequency. The close-loop equation, for $n = 2$, is thus described (with $w_i = w_j$) by:

$$\begin{aligned} P_{ii} &= jw_i \tfrac{c_i}{m_i} - w_i^2 + \tfrac{k_i + k_T}{m_i} - \tfrac{k_i}{m_i} L^i(jw_i, N_a^i) \\ P_{ij} &= -k_T/m_i - \tfrac{K_i}{m_i} L^j(jw_j) \end{aligned}$$

The analysis now is similar, resulting twelve equations for twelve unknowns. Thus, using the set of algebraic equations proposed in this section, networks of oscillators can be automatically tuned.

D.4 Stability and Errors Bounds in the Frequency Domain

The first step in the error analysis is to check for *nondegeneracy*, i.e., nonzero degree relative to Ω, (Bergen et al., 1982). The conditions for a system to be stable are derived next. For a close-loop SISO system to oscillate it is necessary that $1 + N(D, w)G(jw) = 0$. Expressing this relation in terms of its real and imaginary parts, $U(D, w) + jV(D, w) = 0$.

Allow small perturbations in the limit cycle amplitude, rate of change of amplitude, and frequency, and introduce therefore,

$$D \to D + \Delta D, \quad w \to w + \Delta w + j\Delta\sigma \tag{D.48}$$

knowing that the perturbation in the rate of change of amplitude has been associated to the frequency term, so that $\Delta\sigma = -\dot{D}/D$. By substitution,

$$\frac{\partial U}{\partial D}\Delta D + \frac{\partial U}{\partial w}(\Delta w + j\Delta\sigma) + j\left(\frac{\partial V}{\partial D}\Delta D + \frac{\partial V}{\partial w}(\Delta w + j\Delta\sigma)\right) = 0$$

Solve for real and imaginary parts, eliminating Δw,

$$\left(\tfrac{\partial V}{\partial w}\tfrac{\partial U}{\partial D} - \tfrac{\partial U}{\partial w}\tfrac{\partial V}{\partial D}\right)\Delta D = \left[\left(\tfrac{\partial U}{\partial w}\right)^2 + \left(\tfrac{\partial V}{\partial w}\right)^2\right]\Delta\sigma. \tag{D.49}$$

For a stable limit cycle, $\Delta D/\Delta\sigma > 0$, and therefore

$$\frac{\partial V}{\partial w}\frac{\partial U}{\partial D} - \frac{\partial U}{\partial w}\frac{\partial V}{\partial D} > 0. \tag{D.50}$$

From (D.50), the degree is given by

$$d\left((1+NG)\ \ or\ \ N+\frac{1}{G}\ \ or\ \ 1/N+G,\Omega,0\right)=\sum_i sgn(detJ_i) \tag{D.51}$$

where the summation is over all DF solutions in Ω, and J_i is given by $J_i=\left[\begin{array}{cc|cc}\frac{\partial U}{\partial w} & \frac{\partial U}{\partial d_1} & ; & \frac{\partial V}{\partial w} & \frac{\partial V}{\partial d_1}\end{array}\right]$.

Example 1 Consider the 2^{th} order system,

$$\Theta(jw)=G(jw)Y(jw)=k/(k-mw^2+bwj)Y(jw) \tag{D.52}$$

Using the algebraic equations determined in (Arsenio, 2000c,b), it is very fast and straightforward to plot a graph as illustrated in figure D-13. It should be stressed here that the graph is build using analytical equations, and not simulation results. This way, stability properties may be inferred with high precision, because the equations may be solved for as many values as the desired. As easily seen in figure D-13, at the intersection with the origin, equation (D.50) is satisfied, because

$$\partial V/\partial w>0, \partial U/\partial w>0, \partial U/\partial D>0 \, and \, \partial V/\partial D<0.$$

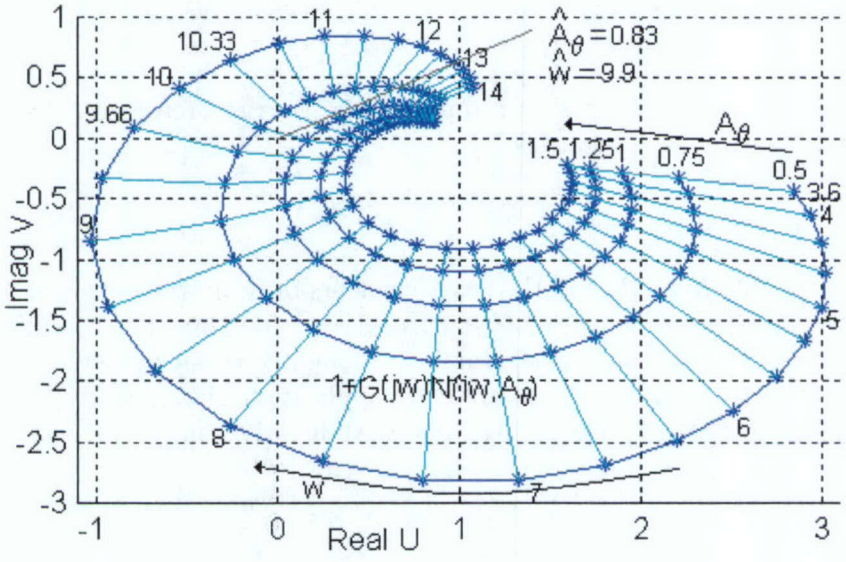


Figure D-13: Graph of 1+N(D,jw)L(jw) in the complex plane. After drawing for different amplitude and frequency values, the intersection with the origin defines the frequency and amplitude of oscillation. This graph is very useful to infer stability. Parameters: $k=30$, $m=0.4$, $b=2$. Oscillator parameters, for all the results in this manuscript: $\beta=\gamma=2$, $\tau_1=0.1$, $\tau_2=0.2$, $c=1$.

Stability may also be demonstrated by decomposing the system $1+N(D,jw)G(jw) = 0 \longrightarrow 1/G(jw)+N(D,jw) = 0$. Thus, for $\hat{w} = 9.9$, $U = N_u+(1/G)_u$, and $V = N_v+(1/G)_v$ (a plot of N is shown ahead in figure D-17), the same result as before is obtained, i.e., the limit cycle is stable.

$$\begin{array}{l} \frac{\partial(1/G)_u}{\partial w} = -2mw/k = -0.26 < 0, \frac{\partial N_u}{\partial w} < 0, \frac{\partial N_u}{\partial D} \approx 0 \\ \frac{\partial(1/G)_v}{\partial w} = b/k = 0.067 > 0, \frac{\partial N_v}{\partial w} \approx 0, \frac{\partial N_v}{\partial D} > 0. \end{array}$$

Stability analysis can also be evaluated using the limit cycle criterion (Slotine and Weiping, 1991) (the proof can be found in (Gelb and Vander Velde, 1968)):

> **Limit Cycle Criterion**: *Each intersection point of the curve of the stable linear system G(jw) and the curve -1/N(D) corresponds to a limit cycle. If points near the intersection and along the increasing-D side of the curve -1/N(D) are not encircled by G(jw), then the corresponding limit cycle is stable. Otherwise, the limit cycle it is unstable.*

The intersection of both plots at the same frequency determines the frequency and amplitude of oscillations for the limit cycle, as shown in figure D-14.

Example 2 Lets analyze the case of one oscillator ($y_2 = 0$) connected to the following 4^{th} order system:

$$\begin{array}{l} m_1\ddot{\theta_1} + c_1\dot{\theta_1} + (k_1 + k_T)\theta_1 - k_T\theta_2 = k_1 y_1 \\ m_2\ddot{\theta_2} + c_2\dot{\theta_2} + (k_2 + k_T)\theta_2 - k_T\theta_1 = k_2 y_2 \end{array} \tag{D.53}$$

The transfer matrix G results from applying the Fourier transform. The Nyquist criterion for a system to oscillate is

$$|I + N_{osc}(D,jw)G(jw)| = 0 \tag{D.54}$$

where $N_{osc_1} = N\Lambda$, $\Lambda_{11} = 1$, and the other elements of Λ are zero. This is equivalent to $|1 + N(D,jw)G_{11}| = 0 \Longleftrightarrow |1/N(D,jw) + G_{11}| = 0$, and thus it is possible to build the graph in figure D-14, and check stability through the limit cycle criterion.
As before, stability may also be inferred by application of condition (D.50), being $U = (1/N)_u+G_{11_u}$ and $V = (1/N)_v+G_{11_v}$ obtained directly from figure D-14, resulting:

$$\frac{\partial(\frac{1}{N})_u}{\partial w}, \frac{\partial G_{11_u}}{\partial w} < 0, \frac{\partial(\frac{1}{N})_v}{\partial w}, \frac{\partial(\frac{1}{N})_u}{\partial D}, \frac{\partial(\frac{1}{N})_v}{\partial D} > 0, \frac{\partial G_{11_v}}{\partial w} \approx 0.$$

D.4.1 Multiple Oscillators

For a system to oscillate, the Nyquist criterion for MIMO systems requires that $|G^{cl}| = 0$, (Zhou and Doyle, 1998). However, the oscillator dynamics is non-linear, and the plant may or may not be linear. Thus, the transfer matrix is not independent

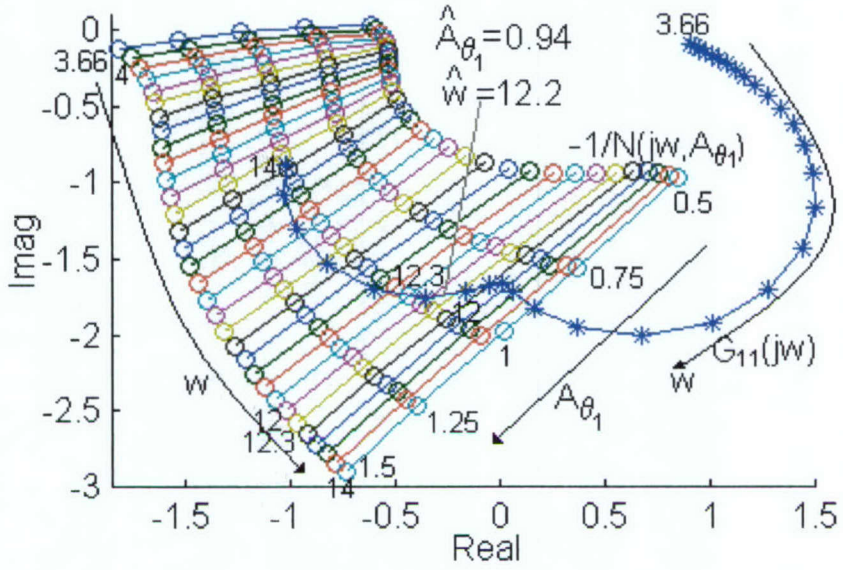


Figure D-14: Graphs of -1/N(D,jw) and $G_{11}(jw)$ in the complex plane. Parameters used: $m_i = 1$, $c_i = 3$, $k_i = 100$, for i=1,2, and $k_T = 30$. The estimated oscillation frequency is $\hat{w} = 12.2$, with amplitude $\hat{D} = \hat{A}_{\theta_1} = 0.94$. After drawing -1/N(D,jw) for different amplitude and frequency values, and $G_{11}(jw)$ for different frequency values, the intersection of both curves at the same frequency corresponds to the frequency and amplitude of oscillation, and the limit cycle is stable.

of the n amplitudes of oscillation, and therefore $n-1$ more constraints are required for oscillations, which are applied to the $n-1$ rows of G^{cl} (all rows could be used, instead of the determinant, since the determinant constraint introduces a singularity in the matrix, (Arsenio, 2000a)):

$$G^{cl}_{i,1} D_{\theta_1} + \sum_{k=2}^{n-1} G^{cl}_{i,k} D_{\theta_k} (cos(\varphi_k) + j sin(\varphi_k)) = 0 \tag{D.55}$$

Expressing (D.54) and (D.55) in terms of its complex parts, results $\sum_{i=1}^{n} U_i + jV_i = 0$. Allow small perturbations in the limit cycle rates of change of amplitude, amplitude ($D_i \rightarrow D_i + \Delta D_i$), phase-shift ($\varphi_i \rightarrow \varphi_i + \Delta\varphi_i + i\sum_{j=1}^{n} \Delta\sigma_{D_j}$) and frequency ($w \rightarrow w + \Delta w + i\sum_{j=1}^{n} \Delta\sigma_{D_j}$). Therefore, $2n$ equations are available, for both complex parts to cancel. The frequency w is unknown, there are n unknowns for the variation in amplitudes ΔD_i, $n-1$ unknowns for the phase-shift variation $\Delta\varphi_i$, $i > 1$ between the first and the i^{th} outputs, and n for the rate of change σ_{D_j}, in a total of $3n$ unknowns. Similarly to the previous section, n constraints are used to eliminate the frequency w and the phase-shifts from the remaining n equations. These equations are then combined to define a transformation $J = M^{-1}N$, described by (D.56). For a locally stable limit cycle, J must be positive definite.

$$[\Delta\sigma_{D_1} \ldots \Delta\sigma_{D_n}]^T = J_{n\times n} [\Delta D_1 \ldots \Delta D_n]^T \tag{D.56}$$

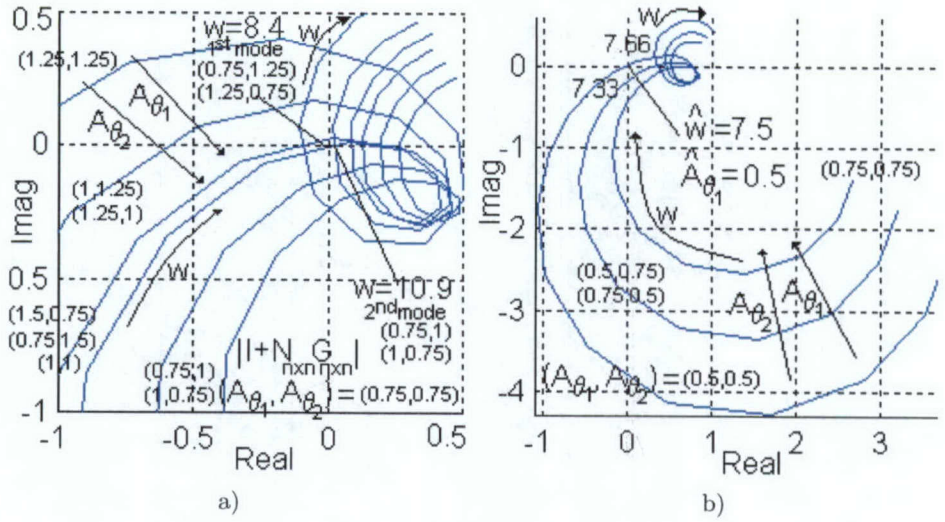


Figure D-15: a) Graph for system with two stable oscillation modes, where $k_1 = k_2 = 50$, and $k_T = 30$. b) Graph for system with one stable limit cycle, where $k_1 = k_2 = 25$, and $k_T = 60$.

Example 3 Lets analyze the stability of two oscillators connected to a 4^{th} order system described by (D.53). In figure D-15-a it is plotted the graph corresponding to

$$|L(jw) + 1/N(D_1, D_2, jw)| \;=\; U_1 + iV_1$$

where $N(D_1, D_2, jw) = [N^1(N(D_1, jw)\ 0\,;\ 0\ N^2(N(D_2, jw)]$ and N^i is the describing function of each oscillator. The intersection with the origin occurs for two frequencies, $w_1 = 8.5$ and $w_2 = 10.8$, corresponding to two different limit modes of oscillation (on phase and in phase opposition, respectively). Since this condition is not enough to infer stability, there are several values of (D_1, D_2) which cancel the determinant at these frequencies. However, the second condition given by (D.55) is used to impose additional constraints.

The same reasoning applies for the system in figure D-15-b. However, for the parameters selected there are now no intersection with the origin for the 2^{th} resonance mode (high frequency mode). Despite the low resolution data in the figure, small amplitude values will not cause high frequency oscillations, because the oscillators do not entrain this plant natural mode for small input amplitudes.

D.4.2 Error Bounds

Being DFs an approximate method, it is crucial an error analysis of the describing functions, to check for error bounds. The goal of this analysis is to find a set Ω of

frequencies and amplitudes within which the true values are contained. A detailed analysis for single input single output (SISO) systems with simple nonlinear elements is presented in (Bergen et al., 1982). However, the nonlinearity represented by the overall dynamics of the neural oscillator is very complex.

For a π-symmetric signal $x(wt + 2\pi) = -x(t)$ with period $2\pi/w$, define $K = \{0, 1, 3, \cdots\}$ and $K^* = \{3, 5, \cdots\}$:

$$x = imag\left(\sum_{k\in K} d_k e^{jkwt}\right), \quad x^* = imag\left(\sum_{k\in K^*} d_k e^{jkwt}\right)$$

where $d_1 = D$, and x^* is the part of the solution neglected by the estimate of the DF. The estimate is reliable only if x^* is small. Since the filtering assumption is the requirement for a small x^*, the performance of the linear part of the system in filtering unwanted harmonics is given by:

$$\rho(w) = \sqrt{\sum_{k\in K^*} |G(jkw)|^2} \tag{D.57}$$

The function used to find the DF output error is

$$p(D) = \sqrt{||n(Dsin(wt))||_2^2 - |DN_d|^2} \tag{D.58}$$

where n is the nonlinearity (accounting for the dynamics of all the oscillator), N_d the describing function of the nonlinearity, and $||\cdot||_2$ is the L_2 norm on $[0, 2\pi/w]$. The function p may be replaced by an upper bound αD if n has finite gain α, $|n(x)| \leq \alpha|x|$, with some loss of accuracy in the eventual error estimate. For the neural oscillator, N_d was computed from the solutions of the algebraic equations, and then $p(D)$ was determined using *Matlab*.

The function that measures the error introduced by neglecting higher harmonics at the input of n is given by (where $||\cdot||_\infty$ is the L_∞ norm), (Bergen et al., 1982),

$$q(D, \epsilon) = sup_{||x^*||_\infty \leq \epsilon} ||n(Dsin(wt) + x^*) - n(Dsin(wt))||_2.$$

The smaller the value of q, the better the eventual error estimate. An upper bound was used for q. As described in (Bergen et al., 1982), one has to find an upper bound on the higher harmonic error $\epsilon > 0$, and then finding the set Ω. Inequality (D.59) has to be satisfied,

$$\rho(w) min\left\{q(D, \epsilon) + p(D), r(D, \epsilon)\right\} \leq \epsilon \tag{D.59}$$

where $r(D, \epsilon) = \sqrt{2} sup_{|y_1| \leq d_1 + \epsilon} |n(y)|$. A closed-bounded set Ω, which contains $(\hat{w}, \hat{D})$ - the estimated frequency and system output amplitude, is found by all points in the neighborhood that satisfy

$$\eta = \left| N(D) + \frac{1}{G(jw)} \right| / \sigma(w, D) \leq 1 \tag{D.60}$$

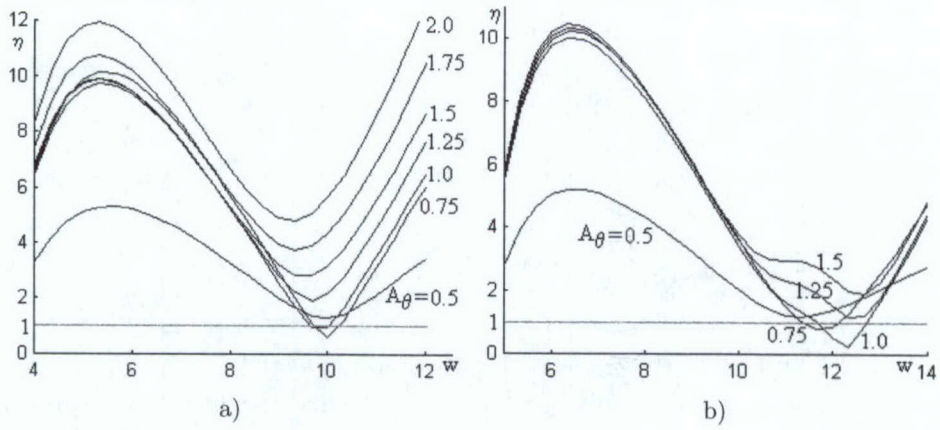


Figure D-16: Estimation of Ω using a grid of points, and checking intersection with unity, for (left) $w = 10$, and (right) $w = 12.9$.

where $\sigma(w, D) = q(D, \epsilon(w, D))/D$. The tightest bounds correspond to small values of ϵ. There are several ways for finding Ω, as described in (Bergen et al., 1982). One can surround $(\hat{w}, \hat{D})$ by a grid of points, find ϵ at each point and calculate the ratio η. The boundary of Ω is given by the points corresponding to a unit ratio. A method that improves the accuracy of the results is poleshifting, (Bergen et al., 1982). Only two equations have to be changed, by adding an extra term:

$$\tilde{q} = \sup_{\|x^*\|_\infty \leq \epsilon} \| \, n(Dsin(wt) + x^*) - n(Dsin(wt)) - \phi x^* \, \|_2$$
$$\tilde{\rho}(w, \phi) = \sum_{k \in K^*} \left| \frac{G(jkw)}{1 + \phi G(jkw)} \right|^2$$

It is now possible to choose ϕ to minimize $\tilde{\epsilon}$. The poleshift factor ϕ is arbitrary. One can determine the optimal value through an optimization process for the best bounds, or through a worst-case analysis, by selecting $\phi = 0$ (value used for the experiments in this section), or even $\phi = 1/2(\alpha_1 + \alpha_2)$, if the nonlinearity has slope bounds α_i.

Example 4 Lets determine the error bounds for the systems in examples 1 and 2. As shown in figure D-16-a, the measured values for the frequency of oscillation are $w = 9.88$ and for the oscillator input amplitude $D = 0.85$, and the predicted ones $\hat{w} = 9.9$ and $\hat{D} = 0.83$, with $\Omega = \{w \in [9.6, 10.2], D \in [0.75, 1]\}$. Note that the discretization of D is very small (scale is 0.25), and thus a higher resolution for D would improve the error bounds, making other curves crossing the unity, giving a more precise boundary. Figure D-16-b shows the error bounds for the system introduced in example 2, which are $\Omega = \{w \in [11.4, 13], D \in [0.75, 1.0]\}$.

Another way of checking error bounds is through the use of error discs, (Bergen et al., 1982). For which frequency, draw a disc of radius σ, and draw circles on the $-1/G$ locus as in figures D-17 and D-18. The points where the tangent of the circles intersect the $N(D, w)$ locus at the same frequency give the range for w and D.

Example 5. The graphical method for error bounds based in the use of error discs is shown in figures D-17 and D-18. This method allows to visualize the variation of the uncertainty with frequency.

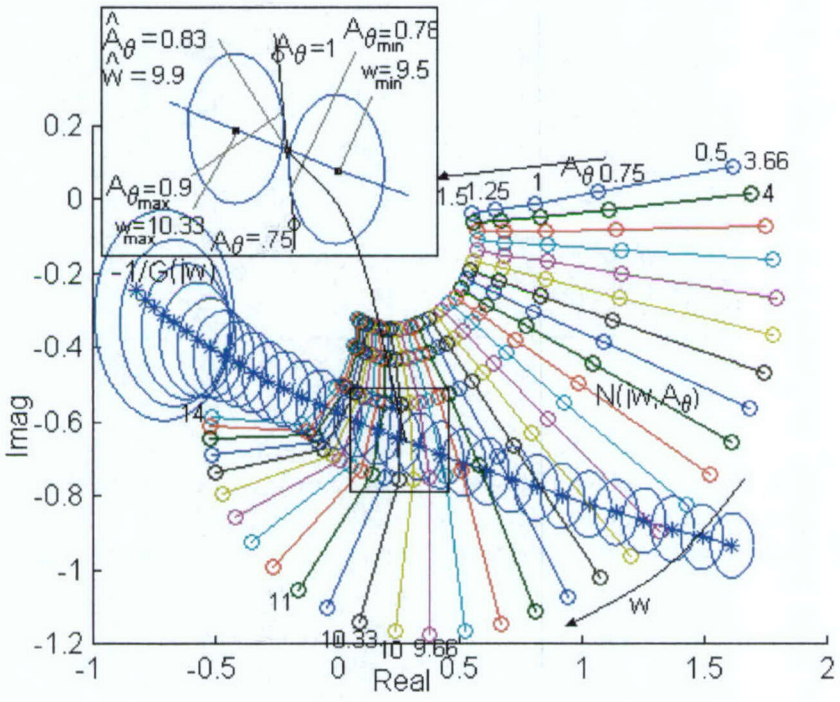


Figure D-17: Error bounds for coupled system with $w_n = 8.66$. The error bounds in frequency and amplitude are $\Omega = \{w = [9.6, 10.2], D = [0.78, 0.9]\}$, being the estimated values $\hat{w} = 9.9$ and $\hat{D} = 0.83$. The real values are $w = 9.88$ and $D = 0.85$.

The degree given by (D.51) can also be determined by inspection. Considering the $-1/G$ loci as a spring, and put pegs at the points correspondent to w_{min}, w_{max}, D_{min} and D_{max}. Now pull the string from the pegs. The degree is nonzero if the strings still cross, otherwise the degree is zero and the analysis has failed, (Bergen et al., 1982).

D.4.3 Extension to Multivariable Systems

Elements of the transfer matrices for MIMO systems are complex transfer functions. Singular value decomposition (*SVD*) (Zhou and Doyle, 1998) provides the tools for a notion of the *size* of $\underline{G}(jw)$ versus frequency, equivalent to a MIMO gain. Direction information can also be extracted from the SVD. Indeed, for inputs along U_i (columns of left singular vector of $\underline{G}$), the outputs along V_i will be amplified by a gain σ_i. Thus, σ_{max} of $\underline{G}(jw)$ define maximum amplification of a unit sinusoidal input at frequency w, and the corresponding singular vector the direction for maximum amplification. One useful norm for such stable systems is the H_∞ norm, defined by

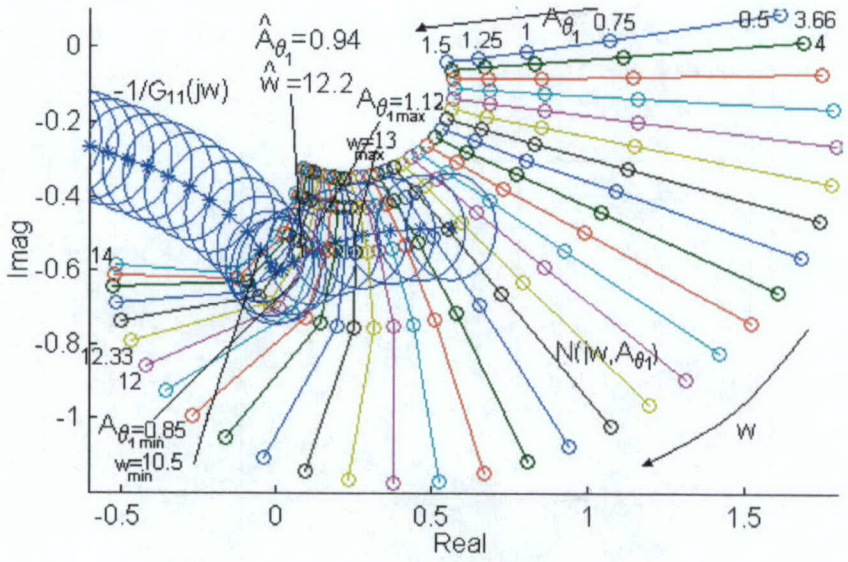


Figure D-18: Error bounds for coupled system with $w_n = 12.24$. The error bounds in frequency and amplitude are $\Omega = \{w = [10.5, 13], D = [0.85, 1.12]\}$, being the estimated values $\hat{w} = 12.2$ and $\hat{D} = 0.94$. The real values are $w = 12.9$ and $D = 1$.

$$\|\underline{G}\|_\infty = sup_w \sigma_{max}\left[\underline{G}(jw)\right] \tag{D.61}$$

For the derivation of the inequalities, consider $X = X_1 + X^*$ as the input vector for the oscillators, $X = -Gn(X)$, and that $X^* = P^*X$ contains only harmonics whose indexes are in K^*. Thus,

$$X^* = -G(jw)P^*n(X_1 + X^*) \tag{D.62}$$

$$X_1 = -G(jw)P_1n(X_1 + X^*) \tag{D.63}$$

where n is a column vector whose elements are the dynamics of each neural oscillator. Writing (D.62) as $X^* = F(w, D_1, X^*)$, it will be necessary to find an inequality that guarantees $||X^*||_2 \leq \epsilon \Rightarrow ||F||_2 \leq \epsilon$, in such a way that F maps $B(0, \epsilon)$ to itself, using the L_2 norm for vectors. Thus,

$$F(w, D_1, X^*) = -Imag\left(\sum_{k\in K^*} G(jkw)c_k e^{j(kwt+\varphi_k)}\right)$$

for $n(D_1 sin(wt) + X^*) = Imag\left(\sum_{k\in K} c_k e^{j(kwt+\varphi_k)}\right)$. By application of Schwartz's inequality,

$$||F||_2^2 \leq \left[\sum_{k\in K^*} \sigma_{max}^2(G(jkw)\right] \sum_{k\in K^*} ||c_k||_2^2 = \rho^2 ||P^*n||_2^2 \tag{D.64}$$

This corresponds to a worst case analysis, and therefore it is independent of D_i and φ_i. The computation of $\rho(w)$ requires only a finite number of terms if G is a stable strictly proper transfer matrix. From (D.63) and (D.65) results the phasor equation (D.67),

$$-E(D_1, X^*)D_1 sin(wt) = P_1 \left[N(X_1 + X^*) - N(X_1)\right] \tag{D.65}$$

$$D_1 = G(jw)E(D_1, X^*)D_1 - G(jw)N(D_1)D_1 \tag{D.66}$$

where $D_1 = [d_{x1}\ d_{x2}e^{j\varphi_2}\ \cdots\ d_{xn}e^{j\varphi_n}]^T$, and N is a diagonal matrix which elements are the DFs of each neural oscillator. For an invertible matrix G results

$$\left[G^{-1}(jw) + N(D_1) - E(D_1, X^*)\right] D_1 = \underline{0}. \tag{D.67}$$

To estimate a bound, lets apply the Schwartz Inequality

$$||P^*n(X)||_2^2 \leq ||P^*(n(X) - n(X_1))||_2^2 + \| P^*n(X_1) \|_2^2$$

so that the DF output error is

$$p(D_1) = \| P^*n(X_1) \|_2^2 = ||n(D_1 sin(wt)||_2^2 - ||D_1 N(D_1)||_2^2.$$

This last relation results from the Pythagorean equality on the Hilbert space of square-integrable periodic functions. From (D.64), one obtains the inequality (D.68):

$$||F||_2 \leq \rho(w)(q(D_1, \epsilon) + p(D_1)) \leq \epsilon \tag{D.68}$$

The function that measures the error at the output of n is given by

$$q = sup_{||X^*||_2 \leq \epsilon} ||n(D_1 sin(wt) + X^*(t)) - n(D_1 sin(wt))||_2.$$

Finally, the resulting condition, using (D.67), is

$$\eta = ||[G^{-1} + N(D_1)]D_1||_2 / q(D_1, \epsilon) \leq 1 \tag{D.69}$$

where ϵ is determined using $q(D_1, \epsilon) = ||E(D_1, X^*)D_1||_2$ and equation (D.68). A closed-bounded set Ω, which contains the estimated frequency, system output amplitudes and phase-shifts - $(\hat{w}, \sum_{i=1}^{n} \hat{D}_i, \sum_{i=2}^{n} \hat{\varphi}_i)$, is found by all points in the neighborhood that satisfy the inequality (D.69). A better bound could be achieved through pole shifting, similarly to the SISO case.

Of course, other norms could be used for $\rho(w)$ in equation (D.57), such as the H_∞ norm, given by (D.61). This norm would be conservative for some cases, but not when equal to $\sigma_{max}(G(jkw)$, for any k. This would be equivalent to $||F||_2/\epsilon = \gamma \leq 1$,

where γ is the L_2 gain of ΔL, with $\Delta = P^* n(X)$ and $L = \sum_{k \in K^*} G(jkw)$ (which could be represented as an interconnection diagram for the error dynamics, (Zhou and Doyle, 1998)). In addition, the L_∞ norm could be used instead of the L_2 norm to find the error bounds.

The singular value decomposition may also be very useful in other situations, such as the analysis of neural oscillators inter-connected in networks. Indeed, when a neural oscillator has more than one input (from other oscillators or from other plant state-variables), it is often difficult to infer which connections should be made to get a desired performance. Thus, inputs which drive the system along the direction of a minimum singular will have a small effect in driving the plant, and may even be negligible for very small σ_{min}. On the other hand, the direction associated to σ_{max} is useful to not only infer maximum propagated errors, but also to determine the optimal inter-connection of networks, so that the network outputs will drive the system with the maximum amplitude, along the maximum singular direction.

For the results thereafter presented the neural oscillators are connected to a MIMO system with transfer matrix G, without mutual connections among the oscillators, i.e., the oscillators are coupled through the dynamics of the MIMO system (N, the approximated transfer matrix for the neural oscillators is diagonal, containing each element the approximated dynamics of each neural oscillator). If there were mutual connections, the computation of the error bounds p and q would be harder. In addition, it is assumed that all neural oscillators oscillate at only one resonance mode (indeed, a large spectrum of frequencies would imply the use of a Describing Function Matrix, (Gelb and Vander Velde, 1968)).

Example 6 For the system in figure D-15 of example 3, figure D-19 shows the graphs obtained from (D.69) varying D_1, D_2, φ and W. As shown, there are two resonance modes at $w = 8.6$ and at $w = 10.9$ with a well defined error interval of frequencies at $[8.3, 9]$ and $[10.2, 11.2]$. Figure D-19-b shows the plot for the system in figure D-15-b of example 3. The system does not oscillate at the second resonance mode, and therefore the error intervals are only determined for the first mode, i.e., $w \in [7, 8]$, with $\hat{w} = 7.5$.

The graphic method of error discs (example 5) might also be extended to two oscillators connected to a MIMO system. However, instead of error discs, the method would now consist of error spheres, and, for higher dimensions, hyperspheres.

D.5 Time-Domain Parameter Analysis

Matsuoka neural oscillators nonlinearities are all linear by parts, (Williamson, 1999). For example, the $max(x, 0)$ nonlinearity has a unity gain when the input is non-negative and zero otherwise. All the nonlinearities of this oscillator may thus be decomposed into regions of operation, and analyzed with linear tools in that regions. Since the oscillator nonlinearities are all continuous, the system is well defined at the boundary of these regions (although the derivatives are not). A time domain analysis is presented for a piece-linear model of the dynamical system, which will

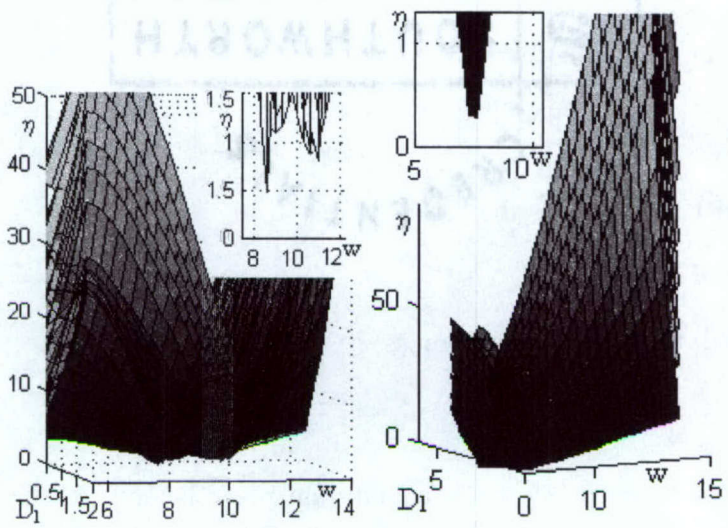


Figure D-19: Error estimation for a MIMO system with the parameters used in example 3. Plots for one of the necessary conditions a) $k_1 = k_2 = 50$, and $k_T = 30$. b) $k_1 = k_2 = 25$, and $k_T = 60$.

bring more insight to variation of oscillator's oscillations with its parameters, and to stability issues. The time-domain description allows a better comprehension of the neural oscillator, being possible the determination of the range of values for which the neural oscillator converges: to a stable equilibrium point, to a stable limit cycle or to a stable limit set. An alternative mathematical formalism of the time-domain analysis for the study of the parameters was presented by (Matsuoka, 1985).

D.5.1 Free Vibrations

The piece-linear dynamic equations of one oscillator for free vibrations, i.e., without an applied input, are,

$$\begin{bmatrix} \dot{x}_1 \\ \dot{v}_1 \\ \dot{x}_2 \\ \dot{v}_2 \end{bmatrix} = \begin{bmatrix} -\frac{1}{\tau_1} & -\frac{\beta}{\tau_1} & -\frac{\gamma}{\tau_1}u_2^x & 0 \\ \frac{1}{\tau_2}u_1^x & -\frac{1}{\tau_2} & 0 & 0 \\ -\frac{\gamma}{\tau_1}u_1^x & 0 & -\frac{1}{\tau_1} & -\frac{\beta}{\tau_1} \\ 0 & 0 & \frac{1}{\tau_2}u_2^x & -\frac{1}{\tau_2} \end{bmatrix} \begin{bmatrix} x_1 \\ v_1 \\ x_2 \\ v_2 \end{bmatrix}$$

$$+ \begin{bmatrix} \frac{c}{\tau_1} & 0 & \frac{c}{\tau_1} & 0 \end{bmatrix}^T \Longleftrightarrow \dot{X} = A_{ij}X + B \tag{D.70}$$

where u_i^x, for i=1,2, is the unit input function relative to x_i, i.e., it cancels for negative values and is equal to unity for positive values. To check for stability of the equilibrium points, results:

$$\begin{vmatrix} \lambda+\frac{1}{\tau_1} & \frac{\beta}{\tau_1} & \frac{\gamma}{\tau_1}u_2^x & 0 \\ \frac{1}{\tau_2}u_1^x & \lambda+\frac{1}{\tau_2} & 0 & 0 \\ \frac{\gamma}{\tau_1}u_1^x & 0 & \lambda+\frac{1}{\tau_1} & \frac{\beta}{\tau_1} \\ 0 & 0 & -\frac{1}{\tau_2}u_2^x & \lambda+\frac{1}{\tau_2} \end{vmatrix} = 0 \tag{D.71}$$

Lets consider the four possible cases:

- $u_1^x = 1 \, and \, u_2^x = 1$

From (D.70), the equilibrium point is,

$$x_1^* = x_2^* = v_1^* = v_2^* = \frac{c}{\beta+\gamma+1} \tag{D.72}$$

For $c \geq 0$ and $\beta + \gamma \geq -1$ the equilibrium point belongs to the region of the state space considered, and therefore this is an equilibrium point for all the system. If c and $\beta + \gamma + 1$ have opposite signs, then the equilibrium is located in the third quadrant and it is not an equilibrium point of the overall system (these points will be thereafter called *virtual* equilibrium points). The stability of the equilibrium point is determined by (D.71), being the eigenvalues given by (D.73),

$$\begin{aligned} \lambda_{1,2} &= -\tfrac{1}{2}\epsilon_1 \pm \tfrac{1}{2}\sqrt{\epsilon_1^2 - 4\tfrac{\beta-\gamma+1}{\tau_1\tau_2}} \\ \lambda_{3,4} &= -\tfrac{1}{2}\epsilon_2 \pm \tfrac{1}{2}\sqrt{\epsilon_2^2 - 4\tfrac{\beta+\gamma+1}{\tau_1\tau_2}} \end{aligned} \tag{D.73}$$

where $\epsilon_1 = \frac{1-\gamma}{\tau_1} + \frac{1}{\tau_2}$ and $\epsilon_2 = \frac{1+\gamma}{\tau_1} + \frac{1}{\tau_2}$. The corresponding eigenvectors are,

$$\begin{aligned} v_{1,2} &= [\alpha_{1,2} \;\; 1 \;\; \alpha_{1,2} \;\; 1]^T \quad & \alpha_{1,2} &= \tau_2\lambda_{1,2} + 1 \\ v_{3,4} &= [-\alpha_{3,4} \;\; -1 \;\; \alpha_{3,4} \;\; 1]^T \quad & \alpha_{3,4} &= \tau_2\lambda_{3,4} + 1 \end{aligned} \tag{D.74}$$

The first two eigenvalues are in the left half of the complex plane if $\beta > \gamma - 1$ and $\gamma < 1 + \frac{\tau_1}{\tau_2}$, and the other two if $\beta > -\gamma - 1$ and $\gamma > -1 - \frac{\tau_1}{\tau_2}$. Therefore, this point is asymptotically stable if $\beta > max(-\gamma-1, \gamma-1)$ and $-1-\frac{\tau_1}{\tau_2} < \gamma < +1+\frac{\tau_1}{\tau_2}$. The point would have all manifolds unstable if $\beta < min(-\gamma-1, \gamma-1)$ and $+1+\frac{\tau_1}{\tau_2} < \gamma < -1-\frac{\tau_1}{\tau_2}$. Since the time constants τ_1 and τ_2 are both positive, this last condition is impossible. Therefore the system' stability depends only on the values of γ and β, and may correspond to a stable or a saddle equilibrium point. A saddle point with two unstable and two stable manifolds (one of the essential conditions for free oscillations for this oscillator) if:

$$\begin{aligned} &\beta > max(-\gamma - 1, \gamma - 1) \\ &\gamma < -1 - \tfrac{\tau_1}{\tau_2} \; or \; \gamma > +1 + \tfrac{\tau_1}{\tau_2}. \end{aligned}$$

For $\tau_1 = 0.1$, $\tau_2 = 0.2$, $\gamma = \beta = 2$, the equilibrium point is a four dimensional saddle point, with two directions converging and the other two diverging, as shown in figure D-20-b.

- $u_1^x = 1 \, and \, u_2^x = 0$

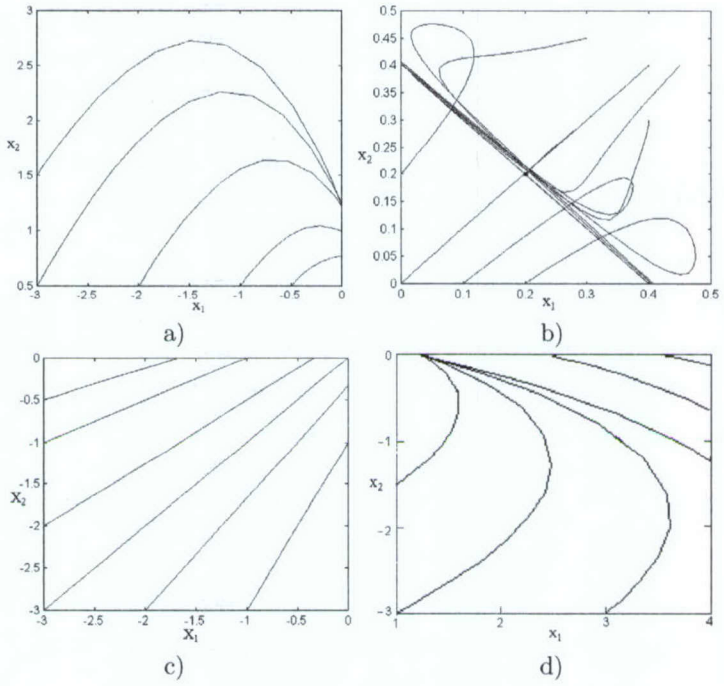


Figure D-20: Plot of the neural oscillator by linear parts: a) $x_1 < 0,\ x_2 \geq 0$ b) $x_1, x_2 \geq 0$ c) $x_1, x_2 < 0$ d) $x_1 \geq 0,\ x_2 < 0$. The parameters used were: $\tau_1 = 0.1$, $\tau_2 = 0.2$, $\beta = \gamma = 2$, and $c = 1$.

The equilibrium point in this case is:

$$x_1^* = v_1^* = \tfrac{c}{\beta+1}, \quad x_2^* = c\tfrac{\beta-\gamma+1}{\beta+1}, \quad v_2^* = 0 \tag{D.75}$$

The stability of the equilibrium point is determined by the eigenvalues:

$$\begin{aligned} \lambda_1 &= -\tfrac{1}{\tau_1}, \qquad \lambda_2 = -\tfrac{1}{\tau_2} \\ \lambda_{3,4} &= -\frac{\left(\frac{1}{\tau_1}+\frac{1}{\tau_2}\right)}{2} \pm \frac{\sqrt{\left(\frac{1}{\tau_1}+\frac{1}{\tau_2}\right)^2 - 4\frac{\beta+1}{\tau_1\tau_2}}}{2} \end{aligned} \tag{D.76}$$

and the corresponding eigenvectors,

$$\begin{aligned} v_1 &= \begin{bmatrix} 0 & 0 & -\frac{\beta\tau_2}{\tau_2-\tau_1} & 1 \end{bmatrix}^T \\ v_2 &= \begin{bmatrix} 0 & 0 & 1 & 0 \end{bmatrix}^T \\ v_{3,4} &= \begin{bmatrix} -\frac{\tau_1\lambda)3,4+1}{\gamma} & \alpha_{3,4} & 1 & 0 \end{bmatrix}^T \end{aligned} \tag{D.77}$$

$$\alpha_{3,4} = \frac{1}{\gamma\beta\tau_2}(-\tau_1^2\lambda_{3,4} - \frac{3\tau_1}{2} + \frac{\tau_2}{2} \pm \frac{\sqrt{(\tau_1-\tau_2)^2 - 4\tau_1\tau_2\beta}}{2} - \tau_1\beta)$$

This equilibrium is stable unless $\beta < -1$, value for which it is unstable. However, only if β and γ are within a certain range of values the eq. is on the fourth quadrant. For other values, this becomes a *virtual* equilibrium point, since it is not an equilibrium point for the overall system. For $\tau_1 = 0.1$, $\tau_2 = 0.2$, $\gamma = \beta = 2$, the equilibrium point is stable and coincident with the equilibrium point for $x_1 \geq 0$ *and* $x_2 \geq 0$, as illustrated in figure D-20-d.

- $u_1^x = 0$ *and* $u_2^x = 1$

Since the equations are symmetric, this case is similar to the previous, and thus the same analysis holds interchanging x_1 and x_2. The eigenvalues and eigenvectors are the same, and the state-space trajectories for this region are illustrated in figure D-20-a.

- $u_1^x = 0$ *and* $u_2^x = 0$

The equilibrium point of the piece-linear dynamic equations is

$$x_1^* = x_2^* = c, \quad v_1^* = v_2^* = 0 \tag{D.78}$$

The stability of this equilibrium point is determined by the eigenvalues,

$$\lambda_{1,2} = -\frac{1}{\tau_1}, \quad \lambda_{3,4} = -\frac{1}{\tau_2} \tag{D.79}$$

and the associated eigenvectors are,

$$\begin{aligned} v_1 &= [1 \;\; 0 \;\; 0 \;\; 0]^T, & v_2 &= [0 \;\; 0 \;\; 1 \;\; 0]^T \\ v_3 &= \left[\tfrac{\beta\tau_2}{\tau_1-\tau_2} \;\; 1 \;\; 0 \;\; 0\right]^T, & v_4 &= \left[0 \;\; 0 \;\; \tfrac{\beta\tau_2}{\tau_1-\tau_2} \;\; 1\right]^T \end{aligned} \tag{D.80}$$

This equilibrium is always stable. However, only a negative tonic would locate the equilibrium on the third quadrant. Therefore, this is a *virtual* equilibrium point, as illustrated in figure D-20-c.

From the exposed, for a zero tonic input (which is always non-negative), the equilibrium point is $(0, 0, 0, 0)$, and therefore all the trajectories will converge to this equilibrium point. Indeed, even if the initial conditions are in the first quadrant, as soon as x_1 or x_2 changes sign, the trajectory will converge asymptotically to the equilibrium, and the system does not oscillate. Therefore, from the previous conditions, for free oscillations $k = \frac{\tau_2}{\tau_1}$, β and γ must satisfy (D.81).

$$\begin{aligned} &\beta > max(-\gamma - 1, \gamma - 1, -1) \\ &\gamma < -1 - 1/k \;\; or \;\; \gamma > +1 + 1/k \end{aligned} \tag{D.81}$$

However, for $\gamma < 0$, $x_1 = x_2$, i.e., the states oscillate on phase, and therefore the output is zero. Thus, (D.81) is simplified to (D.82), which is the same result obtained by Matsuoka, (Matsuoka, 1985), using a different methodology.

$$\beta > \gamma - 1, \quad \gamma > +1 + 1/k \tag{D.82}$$

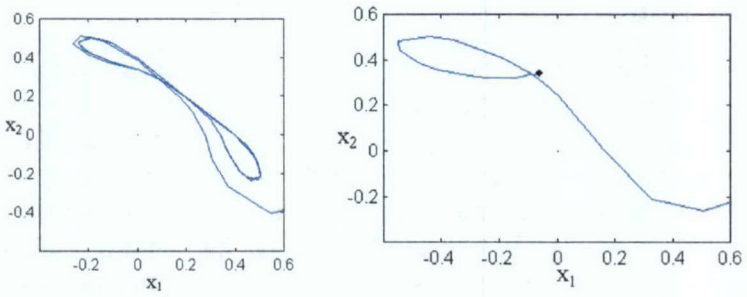


Figure D-21: (left) Simulation for free vibrations, using the MATLAB Simulink Control Box. (right) Simulation for a constant input. The neural oscillator does not oscillates, and converges to the stable equilibrium point.

D.5.2 Forced Vibrations

Generally, the oscillator has a non-zero input g. The main changes on the previous analysis is that now there are two more conditions to be tested, $g > 0$ and $g < 0$, which implies that the system becomes piece-wise linear in eight regions. A constant input $g = D > 0$ (if $D < 0$, the analysis is the same, interchanging indices 1 and 2) is going to change the location of the equilibrium point as follows:

- $x_1 \geq 0 \, and \, x_2 \geq 0$

$$x_1^* = v_1^* = \frac{c}{\beta+\gamma+1} - \frac{1+\beta}{(1+\beta)^2-\gamma^2} D$$
$$x_2^* = v_2^* = \frac{c}{\beta+\gamma+1} + \frac{\gamma}{(1+\beta)^2-\gamma^2} D$$

- $x_1 \geq 0 \, and \, x_2 < 0$

$$x_1^* = v_1^* = \frac{c-D}{\beta+1}, \quad x_2^* = \frac{c(\beta+1-\gamma)+\gamma D}{\beta+1}, \quad v_2^* = 0$$

- $x_1 < 0 \, and \, x_2 \geq 0$

$$x_2^* = v_2^* = \frac{c}{\beta+1}, \quad x_1^* = c\frac{\beta+1-\gamma}{\beta+1} - D, \quad v_1^* = 0$$

- $x_1 < 0 \, and \, x_2 < 0$

$$x_1^* = c - D, \quad x_2^* = c, \quad v_1^* = v_2^* = 0$$

The system's oscillation depends on having three *virtual* attractors in the first quadrant and one real repulsor there. For a constant positive input $D < c\frac{\beta+1-\gamma}{\beta+1}$, or for a constant negative input $D < -c\frac{\beta+1-\gamma}{\gamma}$, the stable *virtual* equilibrium point in region $x_1 < 0 \, and \, x_2 \geq 0$ becomes a true one, while the unstable equilibrium changes quadrant and therefore becomes *virtual*. Therefore, the system converges now to the stable equilibrium and therefore there are no oscillations, as shown in figure D-21-b. Figure D-21-a shows the state-space for free vibrations, for an experiment using the same parameters as in figure D-20.

D.5.3 Transients

The Matsuoka neural oscillator is very robust to perturbations, (Williamson, 1998a). The oscillator usually converges very fast, being often one time period enough for the transient to disappear. However, the duration of the transient depends on the eigenvalues of the dynamics at each region. By tuning γ, β, τ_1 and τ_2, it is possible to design the system with very fast transients and with a desired frequency bandwidth.

When the amplitude of the input signal decreases, for a certain range of input amplitude value the oscillator output is oscillating at two frequencies, corresponding to the input frequency and to the oscillator's free vibration frequency $w_{n_{osc}}$. If the input amplitude is increased/decreased from these range of values, the oscillator spectrum will be concentrated on only one frequency: input frequency w or $w_{n_{osc}}$, respectively. This may occur after a transient in which the oscillator output spectrum power is concentrated on two frequencies, as shown in figure D-22-a.

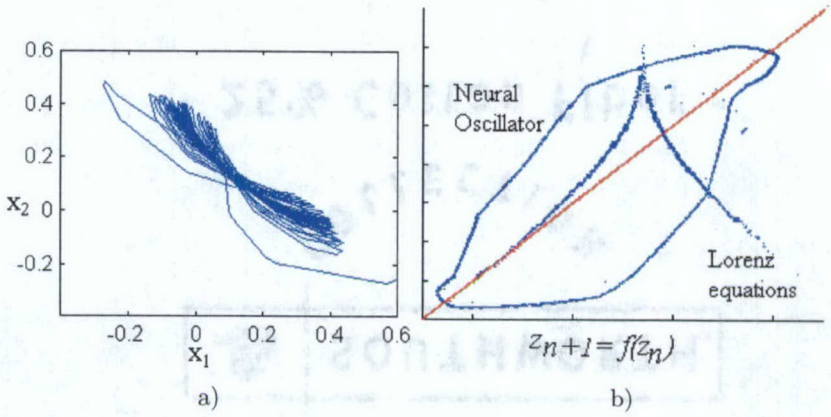


Figure D-22: a) Transients in oscillations. The oscillator initially oscillates at low frequency, but converges slowly to a higher frequency limit cycle. b) Lorenz maps for the neural oscillator and Lorenz equations.

Figure D-22-b shows the *Lorenz map*, (Strogatz, 1994), for the neural oscillator versus Lorenz equations (Strogatz, 1994). In the graph, z_n is the local maximum of $z(t)$ (Lorenz equations) or of the oscillator output y_{osc}. The function $z_{n+1} = f(z_n)$ is called the Lorenz map. If $|f'(z)| > 1$ everywhere, then if any limit cycle exist, they are necessarily unstable. Thus, observing figure D-22-b, contrary to Lorenz equations, the neural oscillator does not present Chaotic behavior during the transients.

D.5.4 Contraction Analysis

Contraction analysis, (Lohmiller and Slotine, 1998), is a method inspired from fluid mechanics and differential theory, that analyzes convergence between two neighboring trajectories by considering the local flow at a given point. Following the definition presented in (Lohmiller and Slotine, 1998), given $\dot{x} = f(x,t)$, the region of the state-space where the Jacobian $\partial f/\partial x$ is uniformly negative definite,

$$\exists \beta > 0, \forall t \geq 0, \frac{1}{2}\left(\frac{\partial f}{\partial x} + \frac{\partial f^T}{\partial x}\right) \leq -\alpha I < 0 \tag{D.83}$$

is denominated a contraction region. However, partial derivatives do not exist on regions boundaries. Therefore, contraction analysis is applied to each linear region of the neural oscillator. Applying (D.83) to (D.70), for each of the four piece-wise linear regions, results that the dynamics do not contract in any of these regions (indeed, there is at least one positive eigenvalue of the matrix defined by (D.83), in each linear region). This is because, for the parameters necessary for oscillations, given by (D.82), the dynamics in any of the three quadrants where $x_1 < 0$ or $x_2 < 0$ (or both), converges to a *virtual* stable equilibrium. Thus, these regions are not contracting – the *virtual* equilibrium points are not contained in these regions. For both $x_1, x_2 \geq 0$, the saddle equilibrium contains two unstable manifolds and two stable. Since the states have to transverse this region in both directions (see figure D-21-a), there is no trajectory to which all points converge.

Volume Contraction

The Matsuoka neural oscillator is dissipative, which means that volumes defined by the state space variables contract in time, although not all the states contract, as just referred. Lets select an arbitrary surface $S(t)$ of volume $V(t)$ in phase space, (Strogatz, 1994). Considering f the instantaneous velocity of points on S (the initial conditions for trajectories), and n the outward normal on S, in time dt the volume expands $(fndt)dA$, and thus $\dot{V} = \int_S fndA$. Using the divergence theorem, results $\dot{V} = \int_V \nabla f dV$. Lets consider first the oscillator uncoupled, as described by (D.70):

$$\begin{array}{l}\nabla \dot{f} = \frac{\partial}{\partial x_1} 1/\tau_1 (c - x_1 - \beta v_1 - \gamma max(x_2, 0) - \sum_i k_i n^+(u_i)) \\ + \frac{\partial}{\partial v_1} 1/\tau_2 (-\beta v_1 + max(x_1, 0)) + \\ \frac{\partial}{\partial x_2} 1/\tau_1 (c - x_2 - \beta v_2 - \gamma max(x_1, 0) - \sum_i k_i n^-(u_i)) \\ + \frac{\partial}{\partial v_2} 1/\tau_2 (-\beta v_2 + max(x_2, 0)) = -2/tau_1 - 2/tau_2 < 0\end{array}$$

Therefore, since the divergence is constant, $\dot{V} = -2(1/\tau_1 + 1/\tau_2)V$. Thus, volumes in phase space shrink exponentially fast to a limiting set of zero volume, (Strogatz, 1994), and the rate of convergence only depends on the positive time constants τ_1 and τ_2.

For an oscillator coupled to a 2^{nd} order system, the state space is now six-dimensional, being the two additional states θ_1 and θ_2, such that

$\dot{\theta_1} = \theta_2, \quad \dot{\theta_2} = -k/m\theta_1 - b/m\theta_2 + k/m[y_1 - y_2]$

resulting $\nabla f = -2/\tau_1 - 2/\tau_2 - b/m$, which is negative, since both the mass and the damping are positive. Therefore, volume contraction occurs. Since the Poincare-Bendixon theorem does not applies for systems with more than two dimensions, contraction analysis is a useful tool to infer volume convergence, and therefore contraction to a limit set.

Considering a multivariable input multivariable output (MIMO) closed-loop system consisting of two oscillators (with only one input for each oscillator), connected to a stable 4^{th} order system,

$$\begin{aligned}
\dot{\phi}_1 &= \phi_2 \\
\dot{\phi}_2 &= \tfrac{1}{m_1}\left(-c_1\phi_2 - (k_1+k_T)\phi_1 + k_2\phi_3 + k_1(y_1^1 - y_2^1)\right) \\
\dot{\phi}_3 &= \phi_4 \\
\dot{\phi}_4 &= \tfrac{1}{m_2}\left(-c_2\phi_4 - (k_2+k_T)\phi_3 + k_1\phi_1 + k_2(y_1^2 - y_2^2)\right)
\end{aligned}$$

results $\nabla f = -4/\tau_1 - 4/\tau_2 - c_1/m_1 - c_2/m_2 < 0$. Therefore, the volume of the MIMO close-loop system also contracts to a limit set. Since the volume contraction occurs $\forall \beta, \gamma$, even for unstable oscillations the volume still contracts. Indeed, there are eigenvectors in this 4^{th} dimensional space along which the state converges to zero, and faster than the eigendirections along which the state may diverge.

D.5.5 Stability Analysis on a Piece-Wise Linear System

Lets first investigate the operation of the neural oscillator in the 1^{st} state-space quadrant. As described by (D.74), there are two eigenvectors ($v_{1,2}$) in which the 1^{st} and 3^{rd} elements are equal, as well as the 2^{nd} and the 4^{th}, and other two eigenvectors ($v_{3,4}$) in which the 1^{st} and 3^{rd} elements are symmetric, as well as the 2^{nd} and the 4^{th} elements. If there is an invariant set on this region, it must occur along $v_{3,4}$, since oscillations do not occur along $v_{1,2}$, because the states along these eigen-directions would oscillate in phase. Indeed, there are no invariant sets along $v_{1,2}$, which are the stable manifolds of the saddle point. Considering directions along $v_{3,4}$, and constrained to the fact that the saddle equilibrium point given by (D.72) is a solution in the state space, lets consider the set $S_1 \bigcap S_2$,

$$\begin{aligned}
S_1 &= \left\{x_1 \geq 0, x_2 \geq 0 : x_1 + x_2 = \tfrac{2c}{\beta+1+\gamma}\right\} \\
S_2 &= \left\{v_1, v_2 : v_1 + v_2 = \tfrac{2c}{\beta+1+\gamma}\right\}
\end{aligned}$$

and apply to this set the local invariant set theorem (or La Salle theorem), (Slotine and Weiping, 1991). This set is invariant for the dynamic system given by (D.70), in $\Omega_l = \{x_1 \geq 0, x_2 \geq 0\}$, if every system trajectory which starts from a point in this set remains in this set for all future time, (Slotine and Weiping, 1991). For a proof, lets determine $\frac{\partial S_i}{\partial t}$, for i=1,2:

$$\dot{S}_1 = \frac{1+\gamma}{-\tau_1}\left(x_1 + x_2 - \frac{2c}{\beta+1+\gamma}\right) - \frac{\beta}{\tau_1}\left(v_1 + v_2 - \frac{2c}{\beta+1+\gamma}\right)$$

$$\dot{S}_2 = \frac{1}{\tau_2}\left[x_1 + x_2 - \frac{2c}{\beta+1+\gamma} - \left(v_1 + v_2 - \frac{2c}{\beta+1+\gamma}\right)\right]$$

Writing the equations in matrix notation, (D.84), it is easily concluded that the derivative is zero on the set. Thus, $S_1 \bigcap S_2$ is an invariant set.

$$\begin{bmatrix} \dot{S}_1 \\ \dot{S}_2 \end{bmatrix} = \begin{bmatrix} -\frac{1+\gamma}{\tau_1} & -\frac{\beta}{\tau_1} \\ \frac{1}{\tau_2} & -\frac{1}{\tau_2} \end{bmatrix} \begin{bmatrix} S_1 \\ S_2 \end{bmatrix} = Q \begin{bmatrix} S_1 \\ S_2 \end{bmatrix} \tag{D.84}$$

Is this invariant set attractive? Lets consider a Lyapunov function (which represents a measure of the *distance* to the invariant set, (Slotine and Weiping, 1991)) and its time derivative,

$$V = \frac{1}{2}(S_1^2 + S_2^2)$$

$$\dot{V} = S_1\dot{S}_1 + S_2\dot{S}_2 = \begin{bmatrix} S_1 & S_2 \end{bmatrix} Q \begin{bmatrix} S_1 \\ S_2 \end{bmatrix}.$$

The matrix Q is negative definite for β and γ satisfying (D.82), since all eigenvalues of Q are negative for $\beta > -1 - \gamma$ and $\gamma > -1 - 1/k$ (which once more demonstrates that oscillations are impossible for $\gamma < 0$, or $x_1 = x_2$). Thus, $\frac{\partial V}{\partial t}$ is zero only on the invariant set, and thus it is negative semi-definite. Therefore, applying the local invariant set theorem, (Slotine and Weiping, 1991), results that every solution in Ω_l tend to this invariant set as $t \to \infty$.

Matsuoka proofed in (Matsuoka, 1985) that the output of the neural oscillator is bounded without any input. Williamson, in (Williamson, 1999), extends the analysis for bounded inputs, and also demonstrates that for oscillators connected to LTI dynamic systems, the close-loop system variables are bounded. The output boundness, the existence of only unstable fixed points (for a certain range of the parameters), and the uniqueness of the solutions, by checking the Lipschitz condition, (Williamson, 1999) (since the oscillator is linear by parts, it is locally Lipschitz in each linear region), imply that the oscillator has oscillatory solutions (not necessarily periodic). Furthermore, Williamson, using a method suggested in (Gonçalves, 1999) based on the linearized *Poincaré map*, (Strogatz, 1994), also showed the local stability of a candidate limit cycle, by imposing conditions for the *Floquet multipliers*, (Strogatz, 1994).

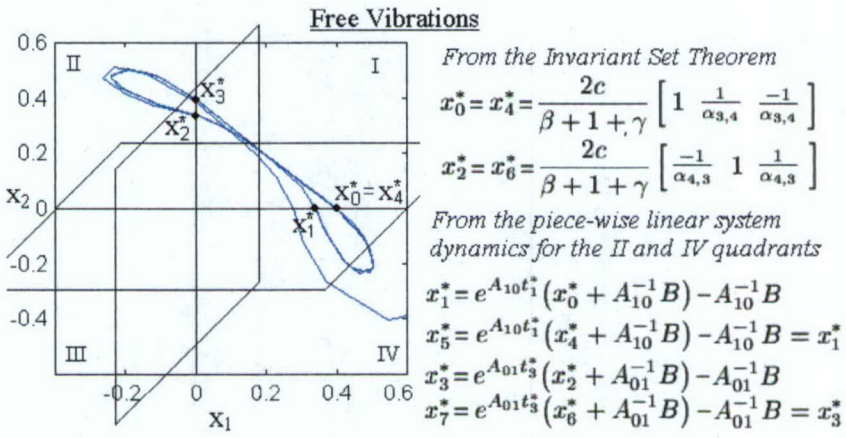


Figure D-23: Two surface sections transverse the flow of the neural oscillator at $x_1 = 0$ and at $x_2 = 0$.

The *Poincaré* map maps the n dimensional system $\dot{x} = f(x,t)$ into the $n-1$

discrete system $x_{k+1} = p(x_k)$, by intersecting the flow with a $n-1$ dimensional hypersurface transverse to the flow, (Strogatz, 1994). Thus, it is possible to translate the problem of close orbits to one of fixed points of a mapping, as shown in Figure D-23.

Bibliography

Acredolo, L. P., Goodwyn, S. W., Horobin, K. D., and Emmons, Y. D. (1999). The signs and sounds of early language development. In Tamis-LeMonda, C. and Balter, L., editors, *Child psychology: A handbook of contemporary issues*, pages 116–142. Psychology Press, New York.

Adolph, K. E., Eppler, M. A., and Gibson, E. J. (1993). Development of perception of affordances. *Advances in Infancy Research*, 8:51–98.

Aleotti, J., Caselli, S., and Reggiani, M. (2003). Toward programming of assembly tasks by demonstration in virtual environments. In *IEEE International Workshop on Human-Robot Interactive Communication*, San Francisco, USA.

Alho, K., Kujala, T., Paavilainen, P., Summala, H., and Naatanen, R. (1993). Auditory processing in visual brain areas of the early blind: evidence from event-related potentials. *Electroencephalogr Clin Neurophysiol*, 86(6):418–27.

Aloimonos, J., Weiss, I., and Bandopadhay, A. (1987). Active vision. *Int. Journal on Computer Vision*, 2:333–356.

American Academy Of Pediatrics (1998). *Caring for Your Baby and Young Child: Birth to Age 5.* Bantham.

Anderson, B. and Moore, J. (1979). *Optimal Filtering.* Prentice-Hall.

Anderson, M. (2003). Embodied cognition: A field guide. *Artificial Intelligence*, pages 91–130.

Aristotle; BC (350). *De Anima (On the Soul). 1986 English Translation.* Penguin Classics.

Arsenio, A. (2000a). Neural oscillator networks for rhythmic control of animats. In *From Animals to Animats 6.* MIT-Press.

Arsenio, A. (2000b). On stability and error bounds of describing functions for oscillatory control of movements. In *IEEE International Conference on Intelligent Robots and Systems*, Takamatsu, Japan.

Arsenio, A. (2000c). Tuning of neural oscillators for the design of rhythmic motions. In *IEEE International Conference on Robotics and Automation.*

Arsenio, A. (2002). *Boosting Vision through Embodiment and Situatedness.* MIT CSAIL research abstracts.

Arsenio, A. (2003a). Embodied vision - perceiving objects from actions. *IEEE International Workshop on Human-Robot Interactive Communication.*

Arsenio, A. (2003b). Towards pervasive robotics. In *IEEE International Joint Conference on Artificial Intelligence*, Acapulco, Mexico.

Arsenio, A. (2004a). *Children, Humanoid Robots and Caregivers.* Fourth International Workshop on Epigenetic Robotics.

Arsenio, A. (2004b). *Developmental Learning on a Humanoid Robot.* IEEE International Joint Conference on Neural Networks, Budapest.

Arsenio, A. (2004c). *On Stability and Tuning of Neural Oscillators: Application to Rhythmic Control of a Humanoid Robot.* International Joint Conference on Neural Networks.

Arsenio, A. (2004d). Teaching a humanoid robot from books. In *International Symposium on Robotics.*

Arsenio, A. and Fitzpatrick, P. (2003). Exploiting cross-modal rhythm for robot perception of objects. In *Proceedings of the Second International Conference on Computational Intelligence, Robotics, and Autonomous Systems - Special Session in Humanoid Robotics*, Singapore.

Arsenio, A., Fitzpatrick, P., Kemp, C. C., and Metta, G. (2003). The whole world in your hand: Active and interactive segmentation. In *Third International Workshop on Epigenetic Robotics.*

Arsenio, A. M. (2003c). Active vision for sociable, mobile robots. In *Proceedings of the Second International Conference on Computational Intelligence, Robotics, and Autonomous Systems - Special session in Robots with a Vision*, Singapore.

Arsenio, A. M. (2003d). A robot in a box. In *International Conference on Advanced Robotics.*

Arsenio, A. M. (2004e). An embodied approach to perceptual grouping. In *IEEE CVPR Workshop on Perceptual Organization in Computer Vision.*

Arsenio, A. M. (2004f). Figure/ground segregation from human cues. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS-04).*

Arsenio, A. M. (2004g). Map building from human-computer interactions. In *IEEE CVPR Workshop on Real-time Vision for Human Computer Interaction.*

Arsenio, A. M. (2004h). Towards and embodied and situated ai. In *Interbational FLAIRS conference*. Nominated for Best Paper Award.

Arsenio, A. M. and Marques, J. S. (1997). Performance analysis and characterization of matching algorithms. In *International Symposium on Intelligent Robotic Systems SIRS'97*, Sweden.

Aryananda, L. (2002). Recognizing and remembering individuals: Online and unsupervised face recognition for humanoid robot. In *Proceedings of the International IEEE/RSJ Conference on Intelligent Robots and Systems*.

Asada, M., MacDorman, K. F., Ishiguro, H., and Kuniyoshi, Y. (2001). Cognitive developmental robotics as a new paradigm for the design of humanoid robots. *Robotics and Autonomous Systems*, 37:185–193.

Aslin, R., Woodward, J., LaMendola, N., and Bever, T. (1996). Models of word segmentation in fluent maternal speech to infants. In Morgan, J. and Demuth, K., editors, *Signal to Syntax: Bootstrapping From Speech to Grammar in Early Acquisition*. Lawrence Erlbaum Associates: Mahwah, NJ.

Atkeson, C. G., Moore, A. W., and Schaal, S. (1997). Locally weighted learning. *Artificial Intelligence Review*, 11:11–73.

Bahrick, L. E. (2003). Development of intermodal perception. In Nadel, L., editor, *Encyclopedia of Cognitive Science*, volume 2, pages 614–617. Nature Publishing Group, London.

Bahrick, L. E. (2004). The development of perception in a multimodal environment. In Bremner, G. and Slater, A., editors, *Theories of infant development*, pages 90–120. Blackwell Publishing, Malden, MA.

Bahrick, L. E. and Lickliter, R. (2000). Intersensory redundancy guides attentional selectivity and perceptual learning in infancy. *Developmental Psychology*, 36:190–201.

Bajcsy, R. (1988). Active perception. *Proceedings of the IEEE*, 76(8):996–1005.

Baker, S. and Kanade, T. (2000). Hallucinating faces. In *Fourth International Conference on Automatic Face and Gesture Recognition*.

Ballard, D., Hayhoe, M., and Pelz, J. (1995). Memory representations in natural tasks. *Journal of Cognitive Neuroscience*, 7(1):66–80.

Banks, M. S. and Dannemiller, J. L. (1987). Infant visual psychophysics. In Salapatek, P. and Cohen, L., editors, *Handbook of Infant Perception*, pages 115–184. New York: Academic Press.

Banks, M. S. and Ginsburg, A. P. (1985). Infant visual preferences: a review and new theoretical treatment. In Reese, H. W., editor, *Advances in Child Development and Behavior*, volume 19, pages 207–246. New York: Academic Press.

Baron-Cohen, S. (1995). *Mindblindness: An Essay on Autism and Theory of Mind.* MIT Press.

Bashford, J. A., Brubaker, B. S., and Warren, R. M. (1993). Cross-modal enhancement of repetition detection for very long period recycling frozen noise. *Journal of the Acoustical Soc. of Am.*, 93(4):2315.

Belhumeur, P., Hespanha, J., and Kriegman, D. (1997). Eigenfaces vs. fisherfaces: Recognition using class specific linear projection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(7):711–720.

Bergen, A., Chua, L., Mees, A., and Szeto, E. (1982). Error bounds for general describing function problems. *IEEE Transactions on Circuits and systems.*

Biederman, I. (1987). Recognition-by-components: A theory of human image interpretation. *Psychological Review*, 94:115–148.

Biederman, I., Mezzanotte, R., and Rabinowitz, J. (1982). Scene perception: detecting and judging objects undergoing relational violations. *Cognitive Psychology*, 14:143–177.

Bobick, A. and Pinhanez, C. (1995). Using approximate models as source of contextual information for vision processing. In *Proceedings of the ICCV'95 Workshop on Context-Based Vision*, pages 13–21, Cambridge, MA.

Bower, T. G., Broughton, J. M., and Moore, M. K. (1970). The cordination of visual and tactual input in infants. *Perception and Psychophysics*, 8:51–53.

Boyd, S., ElGhaoui, L., Feron, E., and Balarkishnan, V. (1994). *Linear Matrix Inequality.* SIAM Press.

Boykov, Y. and Kolmogorov, V. (2001). An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision. In *Energy Minimization Methods in Computer Vision and Pattern Recognition*, pages 359–374.

Breazeal, C. (2000). *Sociable Machines: Expressive Social Exchange Between Humans and Robots.* PhD thesis, MIT, Cambridge, MA.

Breazeal, C. and Aryananda, L. (2000). Recognition of affective communicative intent in robot-directed speech. In *Proceedings of the International IEEE/RSJ Conference on Humanoid Robotics*, Cambridge, MA.

Breazeal, C. and Fitzpatrick, P. (2000). That certain look: Social amplification of animate vision. In *Proceedings AAAI Fall Symposium, Socially Intelligent Agents - The Human in the Loop*, North Falmouth, MA, USA.

Breazeal, C. and Scassellati, B. (1999). A context-dependent attention system for a social robot. In *Proceedings of the Sixteenth International Joint Conference on Artificial Intelligence*, pages 1146–1151, Stockholm, Sweden.

Bremner, J. G. (1994). *Infancy.* Blackwell.

Brent, M. and Siskind, J. (2001). The role of exposure to isolated words in early vocabulary development. *Cognition*, 81:B33–B44.

Brooks, R. (1999). *Cambrian Intelligence.* MIT Press.

Brooks, R. (2002). *Flesh and machines: how robots will change us.* Pantheon Books.

Brooks, R. A. (1991a). Intelligence without reason. In *Proceedings of the 1991 International Joint Conference on Artificial Intelligence*, pages 569–595.

Brooks, R. A. (1991b). Intelligence without representation. *Artificial Intelligence Journal*, 47:139–160. originally appeared as MIT AI Memo 899 in May 1986.

Brooks, R. A., Breazeal, C., Irie, R., Kemp, C. C., Marjanović, M., Scassellati, B., and Williamson, M. M. (1998). Alternative essences of intelligence. In *Proceedings of the American Association of Artificial Intelligence*, pages 961–968.

Brooks, R. A. and Stein, L. A. (1994). Building brains for bodies. *Autonomous Robots*, 1(1):7–25.

Bruner, J. S., Olver, R. R., and Greenfield, P. M. (1966). *Studies in Cognitive Growth.* New York: Wiley.

Burt, P. and E.H, E. H. A. (1983). The laplacian pyramid as a compact image code. *IEEE Trans. COMM*, 31:532–540.

Canny, J. F. (1986). A computational approach to edge detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 8(6):34–43.

Carpenter, R. (1988). *Movements of the Eyes.* Pion Limited, London.

Chatila, R. and Laumond, J. (1985). *Position referencing and consistent world modelling for mobile robots.* IEEE International Conference on Robotics and Automation.

Chellappa, R., Wilson, C., and Sirohey, S. (1995). Human and machine recognition of faces: A survey. In *Proceedings of the IEEE*, volume 83, pages 705–740.

Chun, M. and Jiang, Y. (1998). Contextual cueing: Implicit learning and memory of visual context guides spatial attention. *Cognitive Psychology*, 36:28–71.

Churchland, P., Ramachandran, V., and Sejnowski, T. (1994). *A Critique of Pure Vision, in C. Koch and J. Davis eds, 'Large-Scale Neuronal Theories of the Brain'.* MIT Press.

Cohen, M. and Massaro, D. (1990). Synthesis of visible speech. *Behaviour Research Methods, Intruments and Computers*, 22(2):pp. 260–263.

Comaniciu, D. and Meer, P. (1997). Robust analysis of feature spaces: Color image segmentation. In *IEEE Conference on Computer Vision and Pattern Recognition*, San Juan, Puerto Rico.

Cutler, R. and Davis, L. (2000). Robust real-time periodic motion detection, analysis, and applications. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(8):781–796.

Cutler, R. and Turk, M. (1998). View-based interpretation of real-time optical flow for gesture recognition. In *Int. Conference on Automatic Face and Gesture Recognition.*

Damasio, A. R. (1994). *Descartes Error. Emotion, Reason, and the Brain.* Bard.

Damasio, A. R. (1999). *The Feeling of What Happens: Body and Emotion in the Making of Consciousness.* Harcourt.

Darrel, T. and Pentland, A. (1993). Space-time gestures. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 335–340, New York, NY.

Daubechies, I. (1990). The wavelet transform, time-frequency localization and signal analysis. *IEEE Transactions in Informatation Theory*, 36.

Daugman, J. (1985). Uncertainty relation for resolution in space, spatial frequency, and orientation optimized two-dimensional visual cortical filters. *Journal of the Optical Society of America*, 2(7):1160–1168.

Dautenhahn, K. (1995). Getting to know each other–artificial social intelligence for autonomous robots. *Robotics and Autonomous Systems*, 16(2–4):333–356.

Dautenhahn, K. and Nehaniv, C. L., editors (2001). *Imitation in Animals and Artifacts.* MIT Press.

DeCoste, D. (1997). The future of chess-playing technologies and the significance of kasparov versus deep blue. In *Proceedings of the AAAI-97 Workshop on Deep Blue vs Kasparov: The Significance for Artificial Intelligence.*

Dennet, D. (1998). *Brainchildren.* MIT Press.

Diamond, A. (1990). Developmental time course in human infants and infant monkeys, and the neural bases of inhibitory control in reaching. In *The Development and Neural Bases of Higher Cognitive Functions*, volume 608, pages 637–676. New York Academy of Sciences.

Dickinson, A. (2001). Causal learning: Association versus computation. *Current Directions in Psychological Science*, 10(4):127–132.

Dillmann, R., Rogalla, O., Ehrenmann, M., ollner, R., and Bordegoni, M. (1999). Learning robot behaviour and skills based on human demonstration and advice: The machine learning paradigm. In *9th International Symposium of Robotics Research.*

Driver, J., G., B., and Rafal, R. (1992). Preserved figure-ground segregation and symmetry perception in visual neglect. *Nature*, 360(6399):73–5.

DSM-IV (1994). Diagnostic and statistical manual of mental disorders. American Psychiatric Association, Washington DC.

Duric, Z., Fayman, J., and Rivlin, E. (1995). Recognizing functionality. In *Proc. International Symposium on Computer Vision.*

Elman, J. L. (1993). Learning and development in neural networks: The importance of starting small. *Cognition*, 48:71–99.

Elman, J. L., Bates, E. A., Johnson, M. H., Karmiloff-Smith, A., Parisi, D., and Plunkett, K. (1996). *Rethinking Innateness: A connectionist perspective on development.* MIT Press.

Faugeras, O. (1993). *Three - Dimensional Computer Vision: A Geometric Viewpoint.* MIT Press.

Faugeras, O. (1995). Stratification of three-dimensional vision: projective, affine, and metric representations. *Optical Society of America*, 12(3).

Faugeras, O., Luong, Q., and Maybank, S. (1992). Camera self-calibration: Theory and experiments. In *Proceedings of the European Conference on Computer Vision*, pages 321–334.

Faugeras, O., Luong, Q., and Papadopoulo, T. (2001). *The Geometry of Multiple Images: The Laws That Govern the Formation of Multiple Images of a Scene and Some of Their Applications.* MIT Press.

Ferrell, C. (1998). A motivational system for regulating human-robot interaction. In *AAAI-98.*

Fitzpatrick, P. (2003a). First contact: Segmenting unfamiliar objects by poking them. Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems.

Fitzpatrick, P. (2003b). *From First Contact to Close Encounters: A Developmentally Deep Perceptual System for a Humanoid Robot.* PhD thesis, MIT, Cambridge, MA.

Fitzpatrick, P. (2003c). Perception and perspective in robotics. In *Proceedings of the 25th Annual Conference of the Cognitive Science Society*, Boston.

Fitzpatrick, P. and Arsenio, A. (2004). *Feel the beat: using cross-modal rhythm to integrate robot perception.* Fourth International Workshop on Epigenetic Robotics.

Fitzpatrick, P. and Metta, G. (2002). Towards manipulation-driven vision. In *IEEE/RSJ Conference on Intelligent Robots and Systems*, volume 1, pages 43–48.

Fong, T., Nourbakhsh, I., and Dautenhahn, K. (2003). A survey of socially interactive robots. *Robotics and Autonomous Systems*, 42:143–166.

Forsyth, D. (2001). Shape from texture and integrability. In *International Conference on Computer Vision*, Vancouver, BC.

Freeman, W. T. and Adelson, E. H. (1991). The design and use of steerable filters. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13(9):891–906.

Fu, D. D., Hammond, K. J., and Swain, M. J. (1994). Vision and navigation in man-made environments: Looking for syrup in all the right places. In *Proceedings of CVPR Workshop on Visual Behaviors*, pages 20–26, Seattle, Washington. IEEE Press.

Fukunaga, K. (1990). Introduction to statistical pattern recognition. In *Computer Science and Scientific Computing*. Academic Press, New York.

Galef, B. G. (1988). Imitation in animals: History, definitions, and interpretation of data from the psychological laboratory. In Zentall, T. and Galef, B. G., editors, *Social learning: Psychological and biological perspectives*, pages 3–28. Lawrence Erlbaum Associates, Hillsdale, NJ.

Gallup, G., Anderson, J. R., and Shillito, D. J. (2002). The mirror test. In Bekoff, M., Allen, C., and Burghardt, G. M., editors, *The Cognitive Animal: Empirical and Theoretical Perspectives on Animal Cognition*, pages 325–33. Bradford Books.

Gaskett, C. and Cheng, G. (2003). Online learning of a motor map for humanoid robot reaching. In *International Conference on Computational Intelligence, Robotics and Autonomous Systems (CIRAS 2003)*, Singapore.

Gazzaniga, M. S. and LeDoux, J. E. (1978). *The Integrated Mind*. New York. Plenum Press.

Gelb, A. and Vander Velde, W. (1968). *Multiple-Input Describing Functions and Nonlinear System Design*. McGraw-Hill.

Gershenfeld, N. (1999). *The nature of mathematical modeling*. Cambridge university press.

Goldstein, E. (1996). *Sensation & Perception*. Brooks/Cole.

Gonçalves, J. (1999). *Analysis of Switching Systems*. PhD thesis, MIT EECS department.

Gonzalez-mena, J. and Widmeyer, D. (1997). *Infants, Toddlers, and Caregivers*. Mountain View.

Goodenough, F. (1926). *Measurement of Intelligence by Drawings*. New York: World Book Co.

Grigorescu, S. E., Petkov, N., and Kruizinga, P. (2002). Comparison of texture features based on gabor filters. *IEEE Transactions on Iimage Processing*, 11(10).

Gross, C. (1998). *Brain, Vision, Memory: Tales in the History of Neuroscience.* MIT Press.

Harman, G. (1974). *Handbook of perception*, volume 1, chapter Epistemology, pages 410–455. E. C. Carterette & M. P. Friedman (Eds). New York: Academic.

Harris, J. (1994). *Algebraic Geometry: A First Course (Graduate Texts in Mathematics, 133).* Springer-Verlag.

Hartley, R. (1992). Estimation of relative camera positions for uncalibrated cameras. In *Proceedings of* 2^{nd} *European Conference on Computer Vision*, number 588, page 579587. Lecture Notes in Computer Science.

Hartley, R. and Zisserman, A. (2000). *Multiple View Geometry in Computer Vision.* Cambridge University Press.

Hashimoto, S. (1998). Humanoid robots in Waseda University - Hadaly-2 and WABIAN. In *IARP First International Workshop on Humanoid and Human Friendly Robotics*, Tsukuba, Japan.

Hauser, M. and Carey, S. (1998). Building a cognitive creature from a set of primitives: Evolutionary and developmental insights. In Cummins, D. D. and Allen, C., editors, *The Evolution of Mind.* Oxford University Press, New York.

Hauser, M., Kralik, J., Botto-Mahan, C., Garrett, M., and Oser, J. (1995). Self-recognition in primates: Phylogeny and the salience of species-typical features. *Proc. Natl. Acad. Sci.*, 92:10811–10814.

Hauser, M. D. (1992). Costs of deception: Cheaters are punished in rhesus monkeys. *Proc. Natl. Acad. Sci.*, 89:12137–12139.

Hauser, M. D. (1996). *Evolution of Communication.* MIT Press.

Hendriks-Jansen, H. (1996). *Catching Ourselves in the Act.* MIT Press, Cambridge, Massachusetts.

Hernandez-Reif, M. and Bahrick, L. E. (2001). The development of visual-tactual perception of objects: Amodal relations provide the basis for learning arbitrary relations. *Infancy*, 2(1):51–72.

Hodapp, R. M., Goldfield, E. C., and Boyatzis, C. J. (1984). The use and effectiveness of maternal scaffolding in mother-infant games. *Child Development*, 55:772–781.

Horn, B. K. P. (1986). *Robot Vision.* MIT Press.

Hough, P. (1962). Methods and means for recognising complex patterns. U.S. Patent 3 069 654.

Hubel, D. H. and Wiesel, T. N. (1977). Functional architecture of macaque monkey visual cortex. *Proceedings Royal Society of London*, 198(3):1–59.

Hubel, D. H. and Wiesel, T. N. (1979). Brain mechanisms of vision. *Scientific American.*

Iacoboni, M., Woods, P., Brass, M., Bekkering, H., Mazziotta, C., and Rizzolatti, G. (1999). Cortical mechanisms of human imitation. *Science*, 286:25262528.

Ikeuchi, K. and Suehiro, T. (1994). Towards an assembly plan from observation, part i: Task recognition with polyhedral objects. *IEEE Transactions on Robotics and Automation*, 3(10).

Iqbal, Q. and Aggarwal, J. K. (2001). Perceptual grouping for image retrieval and classification. In *Third IEEE Computer Society Workshop on Perceptual Organization in Computer Vision (POCV01)*, page 19.119.4, Vancouver, BC.

J. Doyle, B. Francis, A. T. (1992). *Feedback Control Theory.* Macmillan.

Jain, A. K. (1989). *Fundamentals of Digital Image Processing.* Prentice Hall.

Johansson, G. (1976). Visual motion perception. *Scientific American*, (232):75–88.

Johnson, M. (1987). *The Body in the Mind.* University of Chicago Press, Chicago, IL.

Johnson, M. H. (1993). Constraints on cortical plasticity. In Johnson, M. H., editor, *Brain Development and Cognition: A Reader*, pages 703–721. Blackwell, Oxford.

Johnson, S. (2002). *Development of object perception*, pages 392–399. Nadel, L and Goldstone, R., (Eds.) Encyc. Cognitive Science. Macmillan, London.

Kaelbling, L. P., Littman, M. L., and Moore, A. W. (1996). Reinforcement learning: A survey. *Journal of Artificial Intelligence Research*, 4:237–285.

Kaelbling, L. P., Oates, T., Hernandez, N., and Finney, S. (2001). Learning in worlds with objects. In *AAAI Spring Symposium.*

Kaernbach, C. (1993). Temporal and spectral basis of the features perceived in repeated noise. *Journal of the Acoustical Soc. of Am.*, 94(1):91–97.

Kailath, T. (1980). *Linear Systems.* Prentice–Hall.

Kailath, T. (1981). *Lectures on Wiener and Kalman Filtering.* Springer-Verlag.

Kanda, T., Ishiguro, H., Imai, M., and Ono, T. (2004). Development and evaluation of interactive humanoid robots. In *Proceedings of IEEE (Special issue on Human Interactive Robot for Psychological Enrichment).*

Kandel, E. R., Schwartz, J. H., and Jessell, T. M., editors (1992). Appleton and Lange, 3rd edition.

Kass, M., Witkin, A., and Terzopoulos, D. (1987). Snakes: Active contour models. *International Journal of Computer Vision*, 1:21–31.

Kemp, C. (1997). *Research Abstracts*, chapter A Platform for Visual Learning. MIT Artificial Intelligence Laboratory.

Kiorpes, L. and Movshon, J. (2003). Differential development of form and motion perception in monkeys. *Journal of Vision*, 3(9).

Knudsen, E. I. and Knudsen, P. F. (1985). Vision guides the adjustment of auditory localization in young barn owls. *Science*, 230:545–548.

Knutsson, H. (1982). *Filtering and Reconstruction in Image Processing.* PhD thesis, Linkping University. Diss. No. 88, Sweden.

Knutsson, H. and Andersson, M. (2003). What's so good about quadrature filters? In *IEEE International Conference on Image Processing (ICIP 2003)*, Barcelona, Spain.

Kozima, H. and Yano, H. (2001). A robot that learns to communicate with human caregivers. In *Proceedings of the First International Workshop on Epigenetic Robotics.*

Kozima, H. and Zlatev, J. (2000). An epigenetic approach to human-robot communication. In *IEEE International Workshop on Robot and Human Communication (ROMAN00)*, Osaka, Japan.

Krotkov, E., Henriksen, K., and Kories, R. (1990). Stereo ranging from verging cameras. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12(12):1200–1205.

Kuniyoshi, Y., Inaba, M., and Inoue, H. (1994). Learning by watching: Extracting reusable task knowledge from visual observation of human performance. *IEEE Transactions on Robotics and Automation*, 6(10).

Lacerda, F., Hofsten, C., and Heimann, M. (2000). *Emerging Cognitive Abilities in Early Infancy.* Erlbaum.

Laine, A. and Fan, J. (1993). An adaptive approach for texture segmentation by multi-channel wavelet frames. Technical Report TR-93-025, Center for Computer Vision and Visualization, University of Florida, Gainesville, FL.

Lakoff, G. (1987). *Women, Fire, and Dangerous Things: What Categories Reveal about the Mind.* University of Chicago Press, Chicago, Illinois.

Lakoff, G. and Johnson, M. (1980). *etaphors We Live By.* University of Chicago Press, Chicago, IL.

Lenat, D. B. (1995). CYC: A large-scale investment in knowledge infrastructure. *Communications of the ACM*, 38(11):33–38.

Leslie, A. M. (1982). The perception of causality in infants. *Perception*, 11:173–186.

Leslie, A. M. and Keeble, S. (1987). Do six-month old infants perceive causality? *Cognition*, 25:265–188.

Lewkowicz, D. J. (2000). The development of intersensory temporal perception: an epigenetic systems/limitations view. *Psych. Bull.*, 126:281–308.

Lewkowicz, D. J. (2003). Learning and discrimination of audiovisual events in human infants: The hierarchical relation between intersensory temporal synchrony and rhythmic pattern cues. *Developmental Psychology*, 39(5):795–804.

Lewkowicz, D. J. and Turkewitz, G. (1980). Cross-modal equivalence in early infancy: Auditory- visual intensity matching. *Developmental Psychology*, 16:597–607.

Lim, J. S. (1990). *Two-Dimensional Signal and Image Processing.* Prentice Hall, Upper Saddle River, New Jersey 07458.

Locke, J. (1690). An essay concerning human understanding. Of ideas. Book Two.

Lohmiller, W. and Slotine, J.-J. E. (1998). On contraction analysis for nonlinear systems. *Automatica*, 34(6).

Luenberger, D. (1991). *Linear and Nonlinear Programming.* Addison-Wesley.

Lungarella, M. and Berthouze, L. (2003). Learning to bounce: first lessons from a bouncing robot. In *Proceedings of the 2nd International Symposium on Adaptive Motion in Animals and Machines.*

Lungarella, M. and Metta, G. (2003). Beyond gazing, pointing, and reaching: A survey of developmental robotics. In *Proceedings of the 3rd International Workshop on Epigenetic Robotics*, pages 81–89.

Ma, Y., Soatto, S., Kosecka, J., and Sastry, S. S. (2000). Euclidean reconstruction and reprojection up to subgroups. In *The special issue of International Journal of Computer Vision for David Marr's prize papers*, volume 38, pages 217–227.

Mahler, M. (1979). *The Selected Papers of Margaret S. Mahler : Separation-Individuation*, volume II. NY.

Malik, J., Belongie, S., Shi, J., , and Leung, T. (1999). Textons, contours and regions: Cue integration in image segmentation. In *IEEE International Conference on Computer Vision*, Corfu, Greece.

Mallat, S. (1989). A theory for multiresolution signal decomposition: the wavelet representation. *IEEE Transactions in Pattern Analysis and Machine Intelligence*, 11(7):674–693.

Marjanović, M. J. (2003). *Teaching an Old Robot New Tricks: Learning Novel Tasks via Interaction with People and Things.* PhD thesis, Massachusetts Institute of Technology, Department of Electrical Engineering and Computer Science, Cambridge, MA.

Marjanović, M. J., Scassellati, B., and Williamson, M. M. (1996). Self-taught visually-guided pointing for a humanoid robot. In *From Animals to Animats: Proceedings of 1996 Society of Adaptive Behavior*, pages 35–44, Cape Cod, Massachusetts.

Markman, E. M. (1989). *Categorization and naming in children: problems of induction.* MIT Press, Cambridge, Massachusetts.

Martin, D., Fowlkes, C., and Malik, J. (2002). Learning to detect natural image boundaries using brightness and texture. In *Neural Information Processing Systems.*

Martin, D., Fowlkes, C., Tal, D., and Malik, J. (2001). A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics. In *International Conference on Computer Vision*, Vancouver, BC.

Massopust, P. R. (1994). *Fractal Functions, Fractal Surfaces, and Wavelets.* Academic Press, San Diego, CA.

Matsuoka, K. (1985). Sustained oscillations generated by mutually inhibiting neurons with adaption. *Biological Cybernetics*, 52:367–376.

Matsuoka, K. (1987). Mechanisms of frequency and pattern control in neural rhythm generators. *Biological Cybernetics*, 56:345–353.

Mees, A. (1972). The describing function matrix. *J. Inst. Maths Appplics*, 10:49–57.

Mendonça, P. R. S. and Cipolla, R. (1999). A simple technique for self-calibration. In *Proceedings of the International Conference on Computer Vision and Pattern Recognition*, volume 1, pages 500–505, Fort Collings, Colorado.

Metta, G. (2000). *Babybot: a study into sensorimotor development.* PhD thesis, LIRA-Lab, DIST.

Metta, G. (2001). An attentional system for a humanoid robot exploiting space variant vision. In *IEEE-RAS International Conference on Humanoid Robots.*

Metta, G. and Fitzpatrick, P. (2002). Better vision through experimental manipulation. In *EPSRC/BBSRC International Workshop on Biologically-Inspired Robotics: The Legacy of W. Grey Walter*, Bristol, UK.

Metta, G., Panerai, F., Manzotti, R., and Sandini, G. (2000). Babybot: an artificial developing robotic agent. In *From Animals to Animats: Sixth International Conference on the Simulation of Adaptive Behavior (SAB 2000).*

Metta, G., Sandini, G., and Konczak, J. (1999). A developmental approach to visually-guided reaching in artificial systems. *Neural Networks*, 12:1413–1427.

Metta, G., Sandini, G., Natale, L., and Panerai, F. (2001). Development and robotics. In *Proceedings of the IEEE-RAS International Conference on Humanoid Robots*, pages 33–42.

Michel, G. and Moore, C. (1995). *Developmental Psychobiology: An Interdisciplinary Science.* MITPress.

Milani-Comparetti, A. and Gidoni, E. (1967). Routine developmental examination in normal and retarded children. *Developmental Medicine and Child Neurology*, 9:631–638.

Minsky, M. (1985). *The Society of Mind.* Simon and Schuster, New York.

Minsky, M. and Papert, S. (1970). Draft of a proposal to arpa for research on artificial intelligence at mit, 1970-71.

Moghaddam, B. and Pentland, A. (1995). Maximum likelihood detection of faces and hands. In *International Workshop on Automatic Face and Gesture Recognition*, pages 122–128.

Moore, D. J., Essa, I. A., and Hayes, M. H. (1999). Exploiting human actions and object context for recognition tasks. In *Proceedings of the IEEE International Conference on Image Processing*, volume 1, pages 80–86, Corfu, Greece.

Mosca, E. (1995). *Optimal, Predictive, and Adaptive Control.* Prentice-Hall.

Muir, D. and Slater, A. (2000). *Infant Development: The essential readings.* Essential readings in Developmental Psychology.

Mumford, D. and Shah, J. (1989). Optimal approximations by piecewise smooth functions, and associated variational problems. *Communications on Pure and Applied Mathematics*, page 577684.

Murase, H. and Nayar, S. (1995). Visual learning and recognition of 3d objects from appearance. *International Journal of Computer Vision*, 14:524.

Murray, L. and Sastry, S. (1994). *Robotic Manipulation.* CRC Press.

Nadel, J., Carchon, I., Kervella, C., Marcelli, D., and Reserbat-Plantey, D. (1999). Expectancies for social contingency in 2-month olds. *Developmental Science*, 2(2):164–173.

Nagai, Y. (2004). *Understanding the Development of Joint Attention from a Viewpoint of Cognitive Developmental Robotics.* PhD thesis, Osaka University.

Nagai, Y., Hosoda, K., Morita, A., and Asada, M. (2003). Emergence of joint attention based on visual attention and self learning. In *Proceedings of the 2nd International Symposium on Adaptive Motion of Animals and Machines, Vol.CD-ROM, SaA-II-3*, pages 961–968.

Nakanishi, J., Morimoto, J., Endo, G., Cheng, G., Schaal, S., and Kawato, M. (2003). Learning from demonstration and adaptation of biped locomotion with dynamical movement primitives. In *Workshop on Robot Programming by Demonstration, IEEE/RSJ International Conference on Intelligent Robots and Systems*, Las Vegas, NV.

Natale, L., Metta, G., and Sandini, G. (2002). Development of auditory-evoked reflexes: Visuo-acoustic cues integration in a binocular head. *Robotics and Autonomous Systems*, 39(2):87–106.

Newell, A. and Simon, H. (1961). Gps, a program that simulates thought. In Billing, H., editor, *Lernende Automaten*, pages 109–124. R. Oldenbourg, Munich, Germany. Reprinted in (Feigenbaum and Feldman, 1963, pp.279–293).

Newport, E. L. (1990). Maturational constraints on language learning. *Cognitive Science*, 14:11–28.

Nieuwenhuys, R., Voogd, J., and Huijzen, C. V. (1980). *The Human Central Nervous System: A Synopsis and Atlas.* Springer-Verlag.

Nise, N. (1995). *Control Systems Engineering.* Addison-Wesley.

Oates, T. (1999). Identifying distinctive subsequences in multivariate time series by clustering. In *Knowledge Discovery and Data Mining*, pages 322–326.

Oates, T., Eyler-Walker, Z., and Cohen, P. (2000). Toward natural language interfaces for robotic agents: Grounding linguistic meaning in sensors. In *Proceedings of the 4th International Conference on Autonomous Agents*, pages 227–228.

Oates, T., Jensen, D., and Cohen, P. (1998). Discovering rules for clustering and predicting asynchronous events. In *Predicting the Future: AI Approaches to Time-Series Problems*, pages 73–79. AAAI Press.

Ohba, K. and Ikeuchi, K. (1996). Recognition of the multi specularity objects for bin-picking task. In *IEEE International Conference on Intelligent Robots and Systems (IROS)*, page 14401447, Osaka, Japan.

Oliva, A. and Torralba, A. (2001). Modeling the shape of the scene: a holistic representation of the spatial envelope. *International Journal of Computer Vision*, pages 145–175.

Ono, T., Imai, M., and Nakatsu, R. (2000). Reading a robots mind: A model of utterance understanding based on the theory of mind mechanism. *Advanced Robotics*, 13(4):311–326.

Oppenheim, A. V. and Schafer, R. W. (1989). *Discrete-Time Signal Processing.* Prentice Hall.

Palmer, S. (1999). *Vision Science.* MIT Press.

Palmer, S. E. (1975). The effects of contextual scenes on the identification of objects. *Memory and Cognition*, 3:519–526.

Papageorgiou, C. and Poggio, T. (2000). A trainable system for object detection. *International Journal of Computer Vision*, 38(1):15–33.

Papert, S. (1966). *The summer vision project.* Memo AIM-100, MIT AI Lab.

Pepperberg, I. (1990). Referential mapping: A technique for attaching functional significance to the innovative utterances of an african grey parrot. *Applied Psycholinguistics*, 11:23–44.

Perona, P. and Freeman, W. T. (1998). A factorization approach to grouping. In *European conference in computer vision*, page 655670.

Perrett, D. I., Mistlin, A. J., Harries, M. H., and Chitty, A. J. (1990). Understanding the visual appearance and consequence of hand action. In *Vision and action: the control of grasping*, pages 163–180. Ablex, Norwood, NJ.

Pfeifer, R. and Scheier, C. (1926). *Understanding Intelligence.* MIT Press, Cambridge, MA.

Piaget, J. (1952). *The Origins of Intelligence in Children.* Norton, New York, NY.

Pichler, O., Teuner, A., and Hosticka, B. (1996). A comparison of texture feature-extraction using adaptive gabor filtering, pyramidal and tree-structured wavelet transforms. *Pattern Recognition*, 29(5):733–742.

Polana, R. and Nelson, R. (1997). Detection and recognition of periodic, non-rigid motion. *International Journal of Computer Vision*, 23(3):261–282.

Porikli, F. M. (2001). Object segmentation of color video sequences. In *International Conference on Computer Analysis of Images and Pattern (CAIP).* Springer.

Potter, M. C. (1975). Meaning in visual search. *Science*, 187:965–966.

Purves, D., Augustine, G. J., Fitzpatrick, D., Katz, L. C., Lamantia, A.-S., McNamara, J. O., and Williams, S. M. (2001). Sinauer Associates Inc.

Quartz, S. R. and Sejnowski, T. J. (1997). The neural basis of cognitive development: A constructivist manifesto. *Behavioral and Brain Sciences*, 20:537–596.

Quine, O. (1960). *Word and object.* Harvard University Press, Cambridge, Massachusetts.

Rao, K., Medioni, G., Liu, H., and G.A., B. (1989). Shape description and grasping for robot hand-eye coordination. *IEEE Control Systems Magazine*, 9(2):22–29.

Rao, R. and Ballard, D. (1997). Dynamic model of visual recognition predicts neural response properties in the visual cortex. *Neural Computation*, 9(4):721–763.

Ren, X. and Malik, J. (2003). Learning a classification model for segmentation. In *International Conference on Computer Vision and Pattern Recognition.*

Rensink, R., O'Regan, J., and Clark, J. (1997). To see or not to see: The need for attention to perceive changes in scenes. *Psychological Science*, 8:368–373.

Rey, A. (1959). *Test de copie et de reproduction de mmoire de figures go-metriques complexes.* Paris: Les Editions du Centre de Psychologie Applique.

Rissanen, J. (1983). A universal prior for integers and estimation by minimum description length. *Annals of Statistics*, 11:417–431.

Rochat, P. and Striano, T. (2002). Who's in the mirror? self-other discrimination in specular images by four- and nine-month-old infants. *Child Development*, 73(1):35–46.

Rosenschein, S. J. and Kaelbling, L. P. (1986). The synthesis of machines with provable epistemic properties. In Halpern, J., editor, *Proceedings of the Conference on Theoretical Aspects of Reasoning about Knowledge*, pages 83–98, Los Altos, California. Morgan Kaufmann Publishers.

Rossignol, S., Chau, C., Brustein, E., Belanger, M., and Drew, T. (1996). Locomotor capacities after complete and partial lesions of the spinal cord. *Acta Neurobiologiae Experimentalis*, 56:449–463.

Roy, D. (1999). *Learning Words from Sights and Sounds: A Computational Model.* PhD thesis, MIT.

Sandini, G., Metta, G., and Konczak, J. (1997). Human sensori-motor development and artificial systems. In *Proceedings of the Int. Symp. on Artificial Intelligence, Robotics, and Intellectual Human Activity Support for Applications*, pages 303–314.

Scassellati, B. (1998). Imitation and mechanisms of shared attention: A developmental structure for building social skills. Technical Report Technical Report 98-1-005, University of Aizu, Aizu-Wakamatsu, Japan.

Scassellati, B. (2001). *Foundations for a Theory of Mind for a Humanoid Robot.* PhD thesis, MIT Department of Electrical Engineering and Computer Science.

Schaal, S. (1999). Is imitation learning the route to humanoid robots? *Trends in Cognitive Sciences*, 3(6):233–242.

Schaal, S. and Atkeson, C. (1994). Robot juggling: Implementation of memory based learning. *IEEE Control systems Magazine*, 14(1):57–71.

Schaal, S., Atkeson, C. G., and Vijayakumar, S. (2000). Real-time robot learning with locally weighted statistical learning. In *International Conference on Robotics and Automation*, San Francisco, CA.

Schiele, B. and Crowley, J. L. (2000). Recognition without correspondence using multidimensional receptive field histograms. *International Journal of Computer Vision*, 36(1):31–50.

Seitz, S. M. and Dyer, C. (1997). View-invariant analysis of cyclic motion. *International Journal of Computer Vision*, 25(3):1–23.

Sequeira, V. (1996). *Active Range Sensing for Three-Dimensional Environment Reconstruction.* PhD thesis, Department of Electrical and Computer Engineering, IST/UTL.

Shelov, S. (1998). *Your Baby's First Year.* The American Academy of Pediatrics. Bantam Books.

Shi, J. and Malik, J. (2000). Normalized cuts and image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, (22):888–905.

Shi, J. and Tomasi, C. (1994). Good features to track. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 593 – 600.

Shipley, T. and Kellman, P. (1994). Spatiotemporal boundary formation: boundary, form, and motion perception from transformations of surface elements. *Journal of Experimental Psychology*, 123(1):3–20.

Sipser, M. (1997). *Introduction to the Theory of Computation.* PWS Publishing Company.

Sirovich, L. and Kirby, M. (1987). Low dimensional procedure for the characterization of human faces. *Journal of the Optical Society of America*, 4(3):519–524.

Slotine, J. and Weiping, L. (1991). *Applied Nonlinear Control.* Englewood Cliffs, NJ (Eds). Prentice-Hall.

Spang, K., Brandt, S., Morgan, M., Diehl, V., Terwey, B., and Fahle, M. (2002). Areas involved in figure-ground segregation based on luminance, colour, motion, and stereoscopic depth visualized with fmri. *Journal of Vision*, 2(7).

Steels, L. (1996). Emergent adaptive lexicons. In *Proceedings of the fourth international conference on simulation of adaptive behavior*, pages 562–567, Cape Cod, MA.

Stoytchev, A. (2003). Computational model for an extendable robot body schema. Technical report, Georgia Institute of Technology, College of Computing. GIT-CC-03-44.

Strang, G. and Nguyen, T. (1996). *Wavelets and Filter Banks.* Wellesley-Cambridge Press.

Strogatz, S. (1994). *Nonlinear Dynamics and Chaos.* Addison-Wesley.

Sutton, R. S. and Barto, A. G. (1998). *Reinforcement Learning: An Introduction.* MIT Press, Cambridge, MA.

Swain, M. and Ballard, D. (1991). Colour indexing. *International Journal of Computer Vision*, 7(1):11–32.

Taga, G. (1995). A model of neuro-musculo-skeletal system for human locomotion. *Biological Cybernetics.*

Taga, G., Yamaguchi, Y., and Shimizu, H. (1991). Self-organized control of bipedal locomotion by neural oscillators in unpredictable environment. *Biological Cybernetics*, 65:147–159.

Thelen, E. and Smith, L. (1994). *A Dynamic Systems Approach to the Development of Cognition and Action.* MIT Press, Cambridge, MA.

Tomasello, M. (1997). The pragmatics of word learning. *Japanese Journal of Cognitive Science*, (4):59–74.

Torralba, A. (2003). Contextual priming for object detection. *International Journal of Computer Vision*, pages 153–167.

Torralba, A. and Oliva, A. (2001). *Global depth perception from familiar scene structure.* MIT AI-Memo 2001-036, CBCL Memo 213.

Torralba, A. and Sinha, P. (2001). Detecting faces in impoverished images. Technical Report AI Memo 2001-028, CBCL Memo 208, MIT.

Treisman, A. (1985). Preattentive processing in vision. *Computer Vision, Graphics, and Image Processing*, 31:156–177.

Treisman, A. (1988). Features and objects: The fourteenth bartlett memorial lecture. *Quarterly Journal of Experimental Psychology: Human Experimental Psychology*, 2(40):2001–237.

Treisman, A. and Gelade, G. (1980). A feature integration theory of attention. *Cognitive Psychology*, pages 97–136.

Trevarthen, C. (1980). Neurological development and the growth of psychological functions. In Sants, J., editor, *Developmental Psychology and Society.* MacMillans, London.

Tuceryan, M. and Jain, A. (1993). Texture analysis. In *Handbook of Pattern Recognition and Computer Vision*, pages 235–276.

Turing, A. M. (1950). Computing machinery and intelligence. *Mind*, 49:433–460.

Turk, M. and Pentland, A. (1991). Eigenfaces for recognition. *Journal of Cognitive Neuroscience*, 3(1).

Varchavskaia, P., Fitzpatrick, P., and Breazeal, C. (2001). Characterizing and processing robot-directed speech. In *Proceedings of the International IEEE/RSJ Conference on Humanoid Robotics*, Tokyo, Japan.

Varella, F., Thompson, E., and Rosch, E. (1993). *The Embodied Mind: Cognitive Science and Human Experience.* MIT Press.

Vidyasagar, M. (1985). *Control system synthesis: a factorization approach.* MIT Press.

von der Malsburg, C. (1995). Binding in models of perception and brain function. *Current Opinion in Neurobiology*, 5:520–526.

Vygotsky, L. (1962). *Thought and language.* MIT Press, Cambridge, MA.

Vygotsky, L. (1978). *Mind in society: The development of higher psychological processes.* Harvard University Press, Cambridge, MA.

Watson, J. (1913). Psychology as the behaviorist views it. *Psychological Review*, 20:158–177.

Weldon, T. P., Higgins, W. E., and Dunn, D. F. (1996). Gabor filter design for multiple texture segmentation. *Optical Engineering*, 35(10):2852–2863.

Werker, J., Lloyd, V., Pegg, J., and Polka, L. (1996). Putting the baby in the bootstraps: Toward a more complete understanding of the role of the input in infant speech processing. In Morgan, J. and Demuth, K., editors, *Signal to Syntax: Bootstrapping From Speech to Grammar in Early Acquisition*, pages 427–447. Lawrence Erlbaum Associates: Mahwah, NJ.

Wertheimer, M. (1961). Psychomotor coordination of auditory and visual space at birth. *Science*, 134:1692.

Wiener, N. (1948). *Cybernetics: or Control and Communication in the Animal and the Machine.* MIT Press.

Williamson, M. (1995). Series elastic actuators. Master's thesis, Massachusetts Institute of Technology, Cambridge, Massachusetts, USA.

Williamson, M. (1998a). Neural control of rhythmic arm movements. *Neural Networks*, 11(7-8):1379–1394.

Williamson, M. (1999). *Robot Arm Control Exploiting Natural Dynamics.* PhD thesis, Massachusetts Institute of Technology, Cambridge, Massachusetts, USA.

Williamson, M. M. (1998b). Exploiting natural dynamics in robot control. In *Fourteenth European Meeting on Cybernetics and Systems Research (EMCSR '98)*, Vienna, Austria.

Wilson, R. and Keil, F., editors (1999). *The MIT Encyclopedia of the Cognitive Sciences (MITECS)*. Bradford Books.

Wolfe, J., Klempen, N., and Dahlen, K. (2000). Post-attentive vision. *ournal of Experimental Psychology: Human Perception and Performance*, (26):693–716.

Wolfe, J. M. (1994). Guided search 2.0: A revised model of visual search. *Psychonomic Bulletin & Review*, 1(2):202–238.

Wolfe, J. M., Oliva, A., Butcher, S., and Arsenio, H. C. (2002). An unbinding problem: the desintegration of visible, previously attended objects does not attract attention. *Journal of Vision*, 2(3):256–271.

Wolfson, H. and Rigoutsos, I. (1997). Geometric hashing: an overview. *IEEE Computational Science and Engineering*, 4:10–21.

Wood, D., Bruner, J., and Ross, G. (1976). The role of tutoring in problem-solving. *Journal of Child Psychology and Psychiatry*, 17:89–100.

Xu, N., Bansal, R., and Ahuja, N. (2003). Object segmentation using graph cuts based active contours. In *International Conference on Computer Vision and Pattern Recognition*.

Yu, C., Ballard, D., and Aslin, R. (2003). The role of embodied intention in early lexical acquisition. In *25th Annual Meeting of Cognitive Science Society (CogSci 2003)*, Boston, MA.

Zhang, H. and Malik, J. (2003). Learning a discriminative classifier using shape context distance. In *International Conference on Computer Vision and Pattern Recognition*.

Zhang, Z. (1999). Flexible camera calibration by viewing a plane from unknown orientations. In *International Conference on Computer Vision (ICCV'99)*, pages 666–673, Corfu, Greece.

Zhang, Z. and Faugeras, O. (1992). *3D Dynamic Scene Analysis*. Springer-Verlag.

Zhou, K. and Doyle, J. (1998). *Essentials of Robust Control*. Prentice-Hall.

Zlatev, J. and Balkenius, C. (2001). Introduction: Why "epigentic robotics"? In *International Workshop on Epigentic Robotics*, pages 1–4.