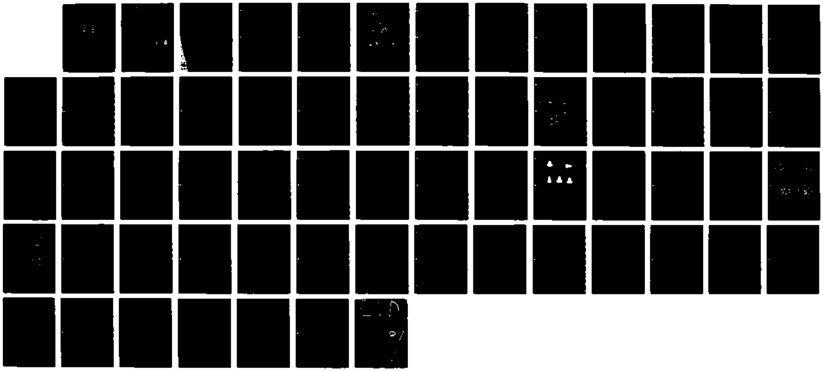
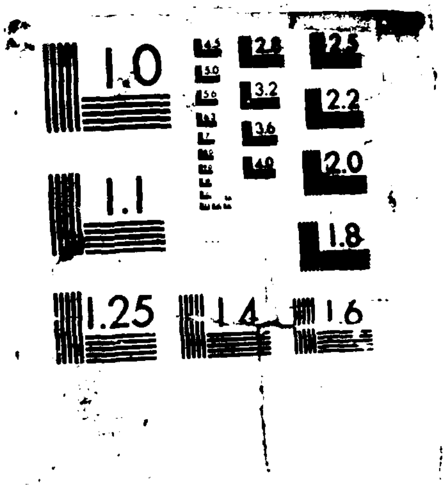
AD-A184 462   AN APPROACH TO OBJECT RECOGNITION: ALIGNING PICTORIAL   1/1
DESCRIPTIONS(U) MASSACHUSETTS INST OF TECH CAMBRIDGE
ARTIFICIAL INTELLIGENCE LAB S ULLMAN DEC 86 AI-M-931
UNCLASSIFIED   N00014-85-K-0214   F/G 12/9   NL





MICROCOPY RESOLUTION TEST CHART

12

REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS BEFORE COMPLETING FORM

1. REPORT NUMBER
AIM 931

2. GOVT ACCESSION NO.

3. RECIPIENT'S CATALOG NUMBER

4. TITLE (and Subtitle)
An Approach to Object Recognition: Aligning Pictorial Descriptions

5. TYPE OF REPORT & PERIOD COVERED
Memorandum

6. PERFORMING ORG. REPORT NUMBER

7. AUTHOR(s)
Shimon Ullman

8. CONTRACT OR GRANT NUMBER(s)
N00014-85-K-0214

9. PERFORMING ORGANIZATION NAME AND ADDRESS
Artificial Inteligence Laboratory
545 Technology Square
Cambridge, MA 02139

10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS

11. CONTROLLING OFFICE NAME AND ADDRESS
Advanced Research Projects Agency
1400 Wilson Blvd.
Arlington, VA 22209

12. REPORT DATE
December, 1986

13. NUMBER OF PAGES
57

14. MONITORING AGENCY NAME & ADDRESS(if different from Controlling Office)
Office of Naval Research
Information Systems
Arlington, VA 22217

15. SECURITY CLASS. (of this report)

15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Distribution is unlimited.

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

Distribution is unlimited

18. SUPPLEMENTARY NOTES

None

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Recognition
Vision
Object Recognition
Alignment

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

This paper examines the problem of shape-based object recognition, and proposes a new approach, the alignment of pictorial descriptions. The first part of the paper reviews general approaches to visual object recognition, and divides these approaches into three broad classes: invariant properties methods, object decomposition methods, and alignment methods.

The second part presents the alignment method. In this approach the recognition process is divided into two stages. The first determines the

(OVER)

Block 20 cont.

transfo:mation in space that is necessary to bring the viewed object into alignment with possible object-models. This stage can proceed on the basis of minimal information, such as the object's dominant orientation, or a small number of corresponding feature points in the object and model. The second stage determines the model that best matches the viewed object. At this stage, the search is over all the possible object-models, but not over their possible views, since the transformation has already been determined uniquely in the alignment stage.

The proposed alignment method also uses abstract description, but unlike structural description methods, it uses them pictorially, rather than in symbolic structural descriptions.

-A-

MASSACHUSETTS INSTITUTE OF TECHNOLOGY
ARTIFICIAL INTELLIGENCE LABORATORY

A.I. Memo No. 931 December, 1986

AN APPROACH TO OBJECT RECOGNITION: ALIGNING PICTORIAL DESCRIPTIONS

Shimon Ullman

ABSTRACT: This paper examines the problem of shape-based object recognition, and proposes a new approach, the alignment of pictorial descriptions. The first part of the paper reviews general approaches to visual object recognition, and divides these approaches into three broad classes: invariant properties methods, object decomposition methods, and alignment methods.

The second part presents the alignment method. In this approach the recognition process is divided into two stages. The first determines the transformation in space that is necessary to bring the viewed object into alignment with possible object-models. This stage can proceed on the basis of minimal information, such as the object's dominant orientation, or a small number of corresponding feature points in the object and model. The second stage determines the model that best matches the viewed object. At this stage, the search is over all the possible object-models, but not over their possible views, since the transformation has already been determined uniquely in the alignment stage.

The proposed alignment method also uses abstract description, but unlike structural description methods, it uses them pictorially, rather than in symbolic structural descriptions.

Acknowledgments. This report describes research done within the Artificial Intelligence Laboratory at the Massachusetts Institute of Technology. Support for the A.I. Laboratory's artificial intelligence research is provided in part by the Advanced Research Projects Agency of the Department of Defense under Office of Naval Research contract N0014-85-K-0214. Support was also provided by NSF Grant IST-8312240.

AN APPROACH TO OBJECT RECOGNITION: ALIGNING PICTORIAL DESCRIPTIONS

1. THE PROBLEM OF SHAPE-BASED RECOGNITION

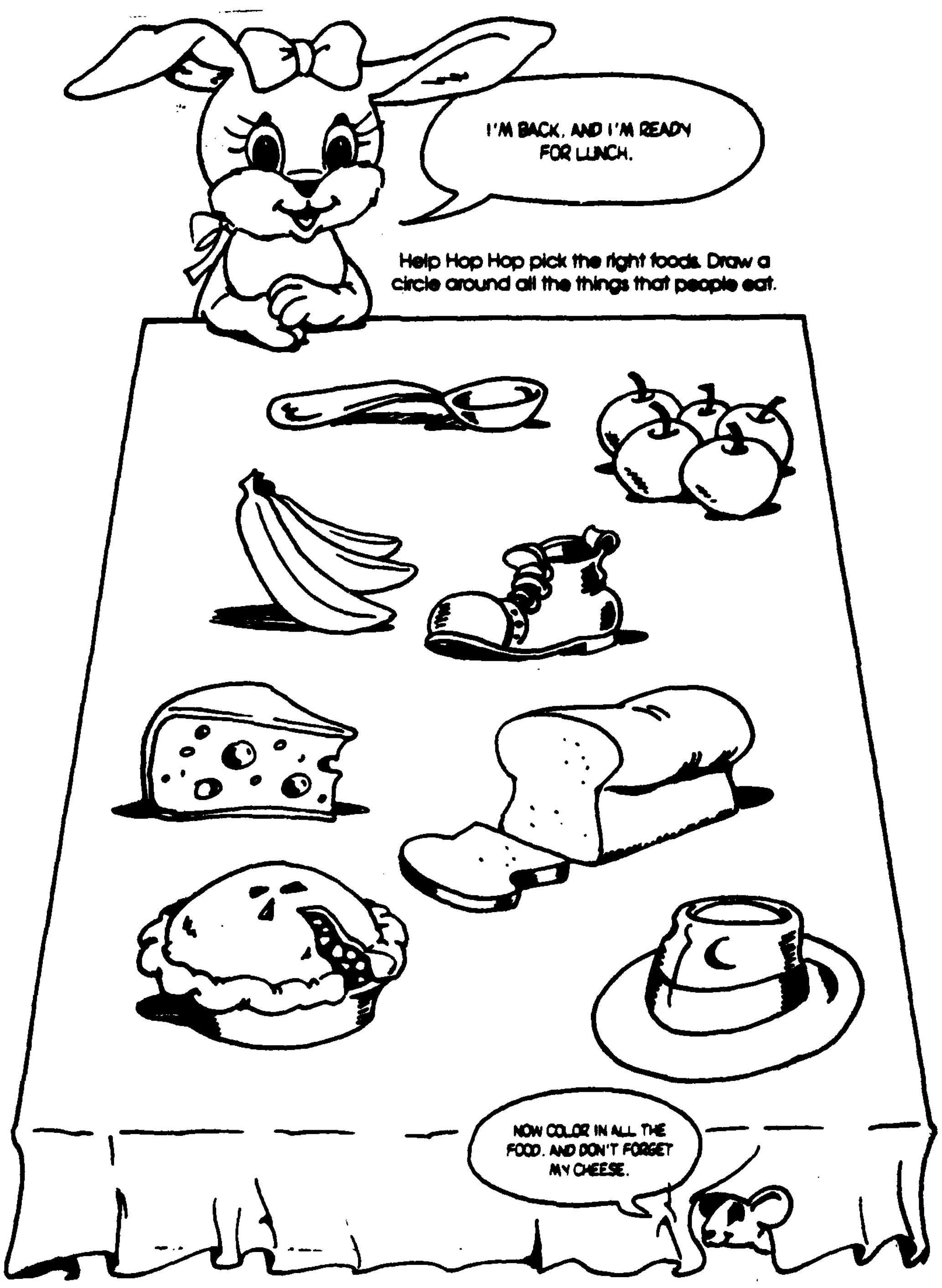
Object recognition is one of the most important, yet least understood, aspects of visual perception. In computational vision, the problem of recognizing objects from visual input has met so far with only limited success. Most of the work in computational vision in recent years has concentrated on the so called "lower level" parts of the visual process. These early processing stages are involved primarily in the extraction of physical properties of the visible environment, such as depth to different points in the visual field, surface orientation, contours of discontinuities, color, material properties, and motion.

The processes that are involved in the extraction of these properties, such as edge detection, stereo, and motion analysis, have been investigated extensively, and many algorithms have been developed to deal with these problems. Although the performance of current algorithms often cannot match that of the human visual system, it appears that the computation of a reasonably accurate low-level representation of the visible environment is becoming feasible.

In contrast, the recognition of common objects is still way beyond the capabilities of any computer vision scheme. The objects such as the ones depicted in Fig. 1 can be recognized immediately and effortlessly by young children, but not by any recognition scheme proposed so far. The difficulties involved in this problem do not seem to result from the lack of sufficient progress in our understanding of the early processing stages. That is, it does not seem that further improvements in edge detection or stereo computation would be of fundamental importance for making progress in the area of visual object recognition. What is required, instead, is a better understanding of the underlying issues: Why is object recognition difficult? Are there different types of object recognition processes? Can the overall task be broken down into more manageable problems? Different possible approaches to visual object recognition will then have to be identified, developed, and evaluated.

Goals of the Paper

This paper has two goals. The first is to examine critically the main classes of approaches that have been proposed for object recognition. The goal is not to give a comprehensive survey of the large number of different methods proposed in the literature (for recent reviews, see Binford 1982, Pinker 1984,



I'M BACK, AND I'M READY FOR LUNCH.
Help Hop Hop pick the right foods. Draw a circle around all the things that people eat.
NOW COLOR IN ALL THE FOOD. AND DON'T FORGET MY CHEESE.

Figure 1. Objects that can be recognized readily on the basis of the shape of their contours. Courtesy of Barron's Educational Series, Inc.

Besl & Jain 1985), but to classify the approaches into a smaller number of major classes. The different approaches are classified into three main classes: recognition by invariant properties, recognition by part decomposition, and alignment methods. Various combinations of the first two have been used extensively in the past, while the third, with a few exceptions, has not been used for object recognition.

The second goal of the paper is to present an approach that appears to offer a promising general strategy for a variety of object recognition problems. This approach belongs primarily to the class of alignment methods (although it uses some of the ideas of the other classes), and is termed the *alignment of pictorial descriptions.*

Scope of the Problem: Shape-Based Recognition

Visual object recognition is not a single problem. One reason for the diversity of approaches to the problem is that there are several different paths leading to visual object recognition.

We often recognize an object (a car, a familiar face, a printed character) visually on the basis of its characteristic shape. We may also use visual, but non-shape cues, such as color and texture. The recognition of a tree of a given type is based more on texture properties, branching pattern, and color, than on precise shape. Similarly, various material types, and different scene types such as "mountainous terrain" can be recognized visually, without relying on precise shape. Certain animals such as a tiger or a giraffe can sometimes be recognized on the basis of texture and color pattern rather than shape.

Objects can also be recognized visually solely on the basis of their location relative to other objects. For example, a door knob may have a non-standard shape, and still be recognized immediately as a door knob purely on the basis of its location relative to the door. Yet another possibility is to recognize objects visually on the basis of their characteristic motion, rather than specific shape. For example, a fly in the room may be perceived as a small dark blob, and still be recognized as a fly, on the basis of its characteristic erratic motion (Johansson 1973, Cutting 1977).

In all of the above examples, recognition can be said to be primarily visual, i.e., the recognition process proceeds primarily on the basis of the visual data. There are also situations in which the recognition process uses sources that are better classified as not primarily visual in nature. One example has to do with the use of prior knowledge and expectations (Potter 1975). For example, one may recognize a white object on one's desk as being a telephone even when the visual image does not contain enough detail for clear object recognition (because the viewing was too brief, or the illumination level too low, etc.). Finally, in some cases, visual recognition employs processes that may be described as reasoning. For example, the recognition of a fence surrounding a house may be based primarily not on the similarity of its shape to some typical fence, but derived f om the fact that its size and location with respect to the house are appropriate for serving a certain function.

These examples of different "paths" leading to visual object recognition are summarized in Table 1. The table is not intended to be complete, but to illustrate the point that visual object recognition includes a number of distinct processes that may be best addressed separately.

PATHS TO RECOGNITION

PRIMARILY VISUAL	VISUAL SUPPLEMENTED BY OTHER SOURCES
SHAPE	EXPECTATION PRIOR KNOWLEDGE
TEXTURE AND COLOR	
LOCATION	REASONING
CHARACTERISTIC MOTION	

Table 1. Different paths leading to visual recognition.

This paper is concerned with the problem of shape-based recognition. Most common objects, such as the ones in Fig. 1, can be recognized in isolation, without the use of context or expectations. For many objects color, texture, and motion play only a secondary role. In these cases, the objects are recognized by their shape properties. This is probably the most common and important aspect of visual recognition and therefore "object recognition" is often taken to mean the visual recognition of objects based on their shape properties. There are some difficulties in defining the term "shape" unequivocally, but such a precise definition will not be required in the ensuing discussion. The main point is that certain types of visual object recognition, e.g. on the basis of color or motion alone, will not be considered here.

Difficulties with the Definition of "Object Recognition"

Although we are constantly engaged in the process of object recognition, it is not easy to define the term "object recognition" in a simple, precise and uncontroversial manner. A first approximation may be something like: "given an image of an object, name the object." A closer scrutiny reveals, however, many problems with such a definition.

First, naming an object is not a necessary requirement for its recognition. Animals can recognize objects without naming them. One can recognize a particular face without naming it. While this objection is valid, it does not appear to be a particularly important one. We may replace "naming x" by "producing a response specific to x," and naming can be taken as a simple example of such a specific response.

More serious difficulties are associated with the term "object". Sometimes, we want to recognize an individual object, or a specific "token" (such as: "my car"), while in other cases recognition means identifying the object as a member of a certain class ("a truck"). Furthermore, an object may belong to a number of classes or categories simultaneously (e.g. my cat, a Siamese cat, a cat, an animal). Recognition would require in such cases a classification at the appropriate level, and what is considered appropriate may depend on the circumstances.

An image often contains multiple objects, and each object may contain a number of recognizable parts. Again, the problem of appropriateness arises. In a cat-image, one may also recognize an eye, a whisker, or a tail. If the question is, "what's in the image", different answers may be appropriate under different circumstances.

These and related issues pose important problems for the general theory of object recognition, but they will not be considered in this paper. For the purpose of the present discussion we will focus on the recognition of individual objects (although the methods discussed are often applicable to classes of objects as well). We will assume first that we are given an image of a single object, or that a region containing an object has been identified in the image. That is, we will not confront directly the segmentation problem. (The recognition scheme must take into account, however, the possibility that parts of the object may be occluded.) Given such a region (that will be called the "image of the object", or a "viewed object"), the problem is to identify (e.g., to name) the object that gave rise to the image in question.

Why Object Recognition is Difficult: The Regularity Problem

The process of object recognition requires the inversion of a complicated one-to-many mapping. The image cast by a single object will change when the object translates or rotates in space. It will also change with the illumination conditions: the level of illumination, the positions and distribution of the light sources, their spectral composition, etc. Formally, one can think of a mapping M that maps a given object 0_i to one of a large set of possible views $(V_{i_1}, \ldots, V_{i_{k_i}})$. Given a single view of the object, the problem is in a sense to invert M and recover the original object O_i. This problem is not limited to recognition based on visual sensory information. Objects can be recognized, for example, from their radar or infra-red images. In these cases, the mapping M depends on such properties as the object's density and temperature. As in vision, object recognition from such images involves the inversion of a one-to-many mapping.

The recognition problem is difficult because the set of possible views of a given object is large, and because different views of the same object can be widely dissimilar.

The problem of dissimilarity between different views of the same object is important in evaluating the main sources of difficulties in object recognition and it merits, therefore, a brief discussion.

One may argue that the notion of similarity between views depends on the particular similarity measure used. The dissimilarity problem may go away if we can define an appropriate similarity measure between object views. We may, therefore, try to define a similarity measure that would render all different views of the same object as closely similar to one another, and will assign a large measure of dissimilarity to views belonging to different objects. In fact, the entire process of object recognition can be thought of as providing such a similarity measure: two images of the same object that are taken, for instance, from widely different viewing positions, are judged ultimately to be closely related. The point is, however, that from the point of view of providing an explanation, assuming the existence of such a similarity measure would just bypass the problem. since the problem would then be to explain how such a similarity measure is defined and computed.

Certain similarity measures that can in fact be implemented directly by known mechanisms have been proposed in the past for the purpose of object recognition. In particular, mechanisms known as associative memories can store a large set of patterns $(P_1, P_2 \ldots P_n)$, and then, given an input pattern

Q, they can retrieve the pattern P_i which is most similar to Q (Kohonen 1978, Hopfield 1982, Huberman & Hogg 1984).

Have associative memories of this type solved the problem of object recognition? Discussions of associative memories sometimes suggest that they have. When the system has stored a representative view, or a few views, of each object, a new view would automatically retrieve the stored representation which most closely resembles it.

The problem is that the notion of similarity used in associative memories is a restricted one. The typical similarity measure used is the so-called "Hamming distance." This measure is defined for two binary vectors. Suppose that **u** and **v** are two binary vectors (i.e., two strings composed of 1's and 0's only), of length n. The Hamming distance between **u** and **v** is simply the number of coordinates in which they disagree. Suppose that we now wish to use such an associative memory to recognize objects, e.g. the letters in the alphabet. We first have to translate each image into a binary string in such a manner that all the A's will map onto vectors separated by a small Hamming distance, and at the same time the Hamming distance between a vector representing an A and a vector representing *any* other letter must be sufficently large. This coding problem, mapping views onto the appropriate vectors is, however, the crucial part that makes the recognition problem difficult.

The situation, then, is that certain similarity measures between input images, such as the Hamming distance and some variations of it, can be implemented directly by known mechanisms. In terms of these similarity measures, however, different views of the same object can be widely dissimilar. The problem remains, therefore, to find the processes by which the disparate views can be identified as representing the same object.

The Direct Approach

One extreme view to the problem of object recognition would be to store a sufficiently large number of different views associated with each object, and still use one of the simple similarity measures discussed above. This may be a feasible approach in some special applications where the total number of possible views is restricted. For the general problem of visual object recognition this direct approach is implausible for two reasons. First, the space of all possible views of all the objects to be recognized is likely to be prohibitively large. Second, objects can be recognized from novel views, whereas in the extreme direct approach, generalization to new views would be severely limited.

It will be possible to outperform the direct approach significantly when the set of views belonging to a given object is not arbitrary, but contains certain regularities. To recognize, for instance, triangles of any shape, position, and size, it is clearly not necessary to store in memory a large number of representative shapes. All of the shapes in this set have certain properties in common, and these regularities can be used to overcome the two limitations of the direct approach mentioned above. That is, it will become possible to limit the number of stored representations, and it will be possible to recognize novel shapes that are not similar in any simple direct measure to triangles seen before.

The conclusion is that finding regularities in the set of views that belong to a single object (or class or objects) is the key to visual object recognition. I will refer to this problem below as the "regularity problem" in object recognition. As we shall see, different approaches to object recognition can be classified into a small number of major classes according to their proposed solution to the regularity problem.

The problem of defining these regularities becomes difficult when we consider natural objects under various possible viewing conditions. For simple geometrical shapes, such as triangles, the set of transformations that a member in the family of views may undergo is well-defined and straightforward to characterize. For the family of views representing a 3-D object, the set of "allowable transformations" that the views may undergo cannot be defined easily, especially when the object can undergo non-rigid transformations. For example, what would be, the regularities in the transformations linking the different possible views of a rabbit?

Different approaches to visual object recognition differ in the type of regularities they propose to exploit. The proposal is not always made explicit, but any theory of object recognition that goes beyond the direct approach must make some assumptions regarding the expected regularities within a family of views that belong to the same object.

In the following sections, prevailing theories of object recognition are classified on the basis of their approach to the regularity problem. Three main classes of theories are distinguished: (i) invariant properties methods, (ii) parts decomposition methods, and (iii) alignment methods. Theories in the first class assume that certain simple properties remain invariant under the transformations that an object is allowed to make. This approach leads to the notion of invariances, feature spaces, clustering, and separation techniques. The second class relies on the decomposition of objects into parts. This leads into the notions of symbolic structural decriptions, feature hierarchies and syntactic pattern recognition. By and large, the first of these general approaches was the dominant one in the earlier days of pattern recognition and the second approach has become more popular in recent years. It will be argued that both of these approaches are insufficient for the general problem of shape-based visual recognition. A third approach, called the alignment method, will be presented and compared to the previous two.

The classification into invariant properties, part decomposition, and alignment methods, is a taxonomy of the underlying ideas, not of existing schemes. That is, a given scheme is not required to belong strictly to one of these classes, but may employ one or more of these ideas. The point is that the variety of methods used seem to rely on a small number of basic ideas for dealing with the regularity problem, and these ideas fall under the three mentioned categories.

The plan of the remainder of the paper is as follows. Sections 2, 3, and 4, describe the invariant properties, decomposition, and alignment approaches. The second part of the paper, Sections 5-7 discuss the recognition of objects using the alignment of pictorial descriptions.

2. INVARIANT PROPERTIES AND FEATURE SPACES

A common approach to object recognition has been to assume that objects have certain invariant properties that are common to all of their views. For example, in identifying different types of biological cells a "compactness measure", defined as the ratio between the cell's apparent area and its perimeter length, has been used as a useful characteristic. Cells that tend to be round and compact will have a high score on this measure, whereas long-and-narrow cells will have a low score.

Formally, a property of this type can be defined as a function from the set of object-views to the real numbers. It is also important that these properties be relatively simple to compute. Otherwise, in recognizing, for example, different instances of the letter "A", one may define a function whose value is 1 if the viewed object is the letter "A", and 0 otherwise. This function would be an invariant of the letter A, but the problem of computing this invariance would be, of course, equivalent to the original problem of recognizing the letter. The invariant properties approach must therefore prescribe, together with the set of invariant properties proposed, effective procedures for extracting these properties.

In an invariant properties scheme the overall recognition process is thus broken down into the extraction of a number of different properties followed by a final decision based on these properties, where each of these stages is relatively simple to compute.

The Domain of Binary Vectors

The invariant properties approach is illustrated schematically in Table 2 for the simplified domain of binary vectors. This domain does not incorporate many of the complexities of real objects, but, as discussed above in the context of associative memory, it is often a useful domain to illustrate in a simple and schematic manner some underlying principle.

Suppose that the set of "images" consists simply of binary vectors, i.e., strings of 0's and 1's, all six-elements long. As in visual object recognition, we assume that a given object may give rise to different sequences. In lack of any regularity in the set of "views" belonging to a given object there will not be a more efficient "object recognition" scheme in this domain than the direct approach. The set of 64 possible sequences may include, for example, eight different "objects", each one giving rise to eight different sequences. If no regularities can be found, recognition would require essentially storing all of the sequences in memory.

Object-1	Object-2
0 1 0 0 1 0	1 1 1 0 1 1
0 0 1 0 1 0	0 1 1 1 1 0
0 0 0 0 0 0	1 0 1 1 1 1
1 1 0 1 0 0	1 1 0 1 0 1

Table 2. Recognition by invariant properties in the domain of binary vectors. All the vectors representing object-1 have at most three 1's, those representing object-2 have four or more.

Table 2 shows a simpler case, in which only eight sequences are considered. Four of them (on the left) are classified as "object-1", the remaining four as "object-2". In this case, a simple property would suffice for distinguishing between the two objects: all the instances of object-1 include at most three 1's in them, and the sequences representing object-2 have four or more. This simplified example illustrates in a schematic manner the essence of recognition by invariant properties. Rather than storing a large number of representative shapes, recognition proceeds by computing a small number of simple functions (properties) of the viewed objects. These properties are supposed to be common to all of the views representing a given object (or class of objects), and to distinguish this class of views from other classes. The properties are also often global, as in the above example: they are not associated with any restricted part of the object, but depend for their computation on the object as a whole.

Feature Spaces and Separating Functions

In some approaches, a property defined for a given object (or class of objects) is not expected to remain entirely invariant, but to lie within some range. Properties of different objects may have partially overlapping ranges, but the hope is that by defining a number of different properties, it will become possible to define each object (or class) uniquely. This leads naturally to the concept of "feature spaces" which have been used extensively in pattern recognition. If n different properties are measured, each viewed object is characterized by a vector of n real numbers. It then becomes possible to represent a given view by a point in an n-dimensional space, R^n. The set of all the views induced by a given object define in this manner a subspace of R^n (e.g., Tou & Gonzalez 1974). This representation could become useful for identifying and classifying objects, provided that the subspaces have simple shapes. For example, suppose that each class to be recognized is contained within a sphere in R^n, and the spheres for the different classes are non-overlapping. Each class can then be represented simply by the center point and the radius of its sphere. A viewed object, including a novel view, would then be classified by determining the sphere in which the point lies in R^n.

Another common method of carving up the space R^n is by a set of linear separating functions. In the case of $n = 3$, for example, the three dimensional feature space is divided into subspaces using a set of 2-D planes. The main reason for using planar separating functions is to keep the computations involved manageable. When the shape of the subspaces does not permit the use of simple separation functions, the space can sometimes be "corrected", e.g. by re-scaling different axes.

Another approach that belongs to the general category of invariant-properties theories is Gibson's theory of high-order invariances (Gibson 1950, 1979). Gibson suggested that invariant properties of objects may be reflected in so-called "higher order" invariances in the optic array. Such invariances may be based, for example, on spatial and temporal gradients of texture density. A set of invariances may be "picked up", according to this view, by the visual system, and may be used to characterize objects and object classes.

How useful have invariant-properties methods been for approaching the problem of visual object recognition? The invariant properties approach, including the construction of feature spaces and their separation into sub-spaces, have probably been studied more extensively than any other method for object recognition. It has met with some success within certain limited domains: a number of industrial vision systems perform simple recognition of industrial parts based on the measurement of global properties such as area, elongation, perimeter length, and different moments (see a review in Bolles & Cain 1982). For the general problem of visual object recognition, however, this general approach does not seem to be very promising. In limited domains, such as the recognition of flat unoccluded parts lying parallel to the image plane, properties of this type may be sufficient to reliably characterize different objects. In more general visual recognition problems the usefulness of simple invariant properties appears doubtful. What simple invariances would distinguish, for example, a fox from a dog? It appears that a more precise description of shape, rather than a restricted set of basic invariances, would be necessary to recognize such objects. Even with simpler, man-made objects, it is not clear how a set of invariances would suffice to capture the regularities in the different views of an object or a class of objects. For example, it would be difficult to recognize the set of all motorcycles using primarily global properties such as apparent area, perimeter length, different moments, and the like.

In summary, the invariant properties approach offers one possible solution to the regularity problem of object recognition: performing the required many-to-one mapping in an efficient manner (compared with the direct approach). The use of invariant properties makes, however, certain assumptions about the regularities in the set of views that belong to the same object (or class of objects). When these assumptions are violated, the invariant properties approach is in trouble. In some cases, such as the artificial example in Table 2, simple invariant properties are indeed common to all the members of a given class. In other cases such invariances may not exist. The weakness of this approach is that in visual object recognition there is no particular reason to assume the existence of relatively simple properties that are preserved across the transformations that an object may undergo. It is not surprising, therefore, that, despite considerable effort, invariant properties of general applicability for visual object recognition proved difficult to find.

3. RECOGNITION USING OBJECT DECOMPOSITION

A second general approach to object recognition relies on the decomposition of objects into constituent parts. This approach clearly has some intuitive appeal. Many objects seem to contain natural parts: a face, for example, contains the eyes, nose, and mouth as distinct parts that can often be recognized on their own. These parts could be found first, and then the recognition of the entire object could use the identified parts.

The approach assumes that each object can be decomposed into a small set of generic components. The components are "generic" in the sense that all objects can be described as different combinations of these components. The decomposition must also be stable, that is, preserved across views. The recognition process locates the parts, classifies them into the different types of generic components and then describes the objects in terms of their constituent parts.

The crucial point in this approach is that *the many-to-one mapping implied by object recognition begins at the part level.* This can result in substantial savings compared with the direct approach. The basic idea is illustrated schematically in Table 3 for the simplified case of binary vectors.

The Domain of Binary Vectors

The domain in this example consists again of binary vectors six components long. There are 64 different vectors in this domain. It is assumed, however, that the first three components of each vector define a "part", and the last three another part. Each part can be of type P_1 or P_2. Table 3a shows how parts (vectors three-components long) are classified as either P_1 or P_2. It can be seen that a many-to-one reduction is achieved at the part level, since many different sub-sequences are all classified as different instances of the same part type (P_1 or P_2).

Table 3b gives the final classification of objects in terms of their parts. An object composed of either (P_1, P_2) or (P_2, P_2) is considered object-1, whereas (P_1, P_1), (P_2, P_1) are instances of object-2. These rules are sufficient to classify unambiguously each of the possible sixty-four vectors. Because of the decompostion into parts, it was possible to avoid the storage of all sixty-four six-long vectors; two reduced tables were sufficient to cover all of the possibilities. Substantial saving is obtained because an object whose first part is, e.g., P_1, and the second P_2, is classified as object-1 regardless of the details of the internal structure of each of the parts.

It should be noted that the part decomposition scheme and the invariant properties approach are not mutually exclusive, but can be combined. In the example given in Table 3, each of the two parts has been defined using an exhaustive list of its different instances. It is also possible to consider a situation in which a part is defined, for instance, by having at most two

P_1	P_2
0 0 0	0 0 1
1 0 0	0 1 1
0 1 0	0 1 1
1 1 0	1 1 1

a

Object-1	Object-2
P_1 P_2	P_1 P_1
P_2 P_2	P_2 P_1

b

Table 3. Recognition by part decomposition in the domain of binary vectors. Table *a* gives the classification of parts P_1 and P_2, *b* classifies two objects in terms of these parts.

1's in its sequence. An example along this line but less abstract is to use a description such as "a bushy tail" in the recognition of a squirrel. The idea behind such descriptions is to combine the advantages of part-decomposition with the use of invariant properties for classifying the constituent parts.

Following the initial classification of the individual components, there remains the problem of recognizing the object itself on the basis of the constituent components. In the language of the binary sequences example, the part classification stage results in a shorter sequence of part types, and a final classification must then be performed on the basis of the parts string. In the example above this final stage was achieved by an exhaustive listing (given in Table 3b). In more realistic object recognition problems other methods are usually employed.

Feature Hierarchies and Syntactic Pattern Recognition

There have been two main approaches to this second classification stage. One approach is to try to repeat the decomposition process: certain part subgroups, containing two or more parts, can be identified as new substructures, or higher-order parts. As in the process of parts identification, the assumption is that certain configurations can be classified independent of other parts and configurations, and that the internal structure of a configuration is immaterial as far as the recognition process is concerned.

An example of a simple part-hierarchy is to detect straight line segments as the most basic parts and then detect higher-level parts such as corners and vertices, based on the already-detected line segments. These parts can be combined in turn into higher-level structures. For example, certain configurations of lines and vertices can be combined to define triangles. Such approaches are known as "feature hierarchies". The simple basic level parts are termed "features" (a term also used in many other contexts) and higher level structures are constructed hierarchically (Selfridge 1959, Sutherland 1959, Barlow 1972, Milner 1974). This approach has been motivated in part by physiological findings (Hubel & Wiesel 1962, 1968) in the cat and monkey, that can be interpreted as the extraction by the visual cortex of elementary features such as oriented edge fragments and line segments.

A close relative of the feature-hierarchy approach is the syntactic pattern recognition method (Fu 1974). Here, too, the first stage consists of identifying simple parts in the input image, followed by the grouping of elementary parts into higher-order ones. The emphasis in the syntactic approach is on the construction of higher order parts using methods borrowed from the syntactic analysis of formal languages.

Structural Descriptions

A second approach to the transition from parts to objects can be viewed as a mixture of part-decomposition with the invariant-properties approach, where the invariant properties are defined using relations among parts. The underlying assumption is that it would be easier to capture object invariances at the level where parts have been identified. For example, the total number of parts of a given type may be an invariant of the object. A triangle, for instance, always has three lines, three vertices, and no free line terminators. This is in fact how perceptrons, which are simple parallel pattern recognition devices, have been used to recognize triangles independent of shape, location and size (Minsky & Papert 1969).

In other instances, simple relations between constituent parts would remain invariant under all object views. In the capital letter "A", for example, two of the line-segment parts meet at a vertex, and this property holds for most variations of the letter. Here, again, part decomposition is obtained first, and in the next stage simple invariances are defined in terms of the constituent parts. The invariances are expressed in terms of relations between two or more parts, such as "above", "to the left of", "longer then", "containing", etc. For 2-D applications, in which objects are restricted to move parallel to the image plane, simple relations such as distances and angles measured in the image would remain invariant (Bolles & Cain 1982, Grimson & Lozano-Perez 1984, Faugeras 1984). In the more general 3-D case, part decomposition schemes often try to employ relations that would remain invariant over a wide range of different viewing positions (Marr & Nishihara 1978, Biederman 1985, Lowe 1985).

Some of the experimental results concerning pattern recognition in animals may be interpreted in this context as indicating a certain deficiency, compared to humans, in this second stage of identifying invariant properties and relations among parts. It has been reported, for instance, that pigeons can recognize Charlie Brown pictures in a variety of positions, orientations and scales (Hernstein 1984). They do not distinguish, however, between a correct Charlie Brown figure, and a "jumbled up" version where the figure has been cut in half, and the two halves re-arranged. These facts are more consistent with recognition on the basis of a collection of local parts and features, rather than, for example, a direct comparison (e.g. by correlation) of complete figures. They also suggest a lack of sensitivity to the relations among different parts.

When augmented with descriptions of relations among parts, the object decompostion approach leads to the notion of structural description. Recognition using such structural decriptions has become in recent years a popular approach to visual object recognition.

An early example of a theory of this type applied to human vision is Milner's (1974) model of visual shape recognition. The main basic-level parts used in this theory are edges and line segments. This choice was motivated to a large degree by the classical physiological findings suggesting the detection of such elements in the image by the primary visual cortex. In a second level, invariant properties and relations are defined using primarily the total number of parts (e.g., the number of line segments of a given orientation) and length ratios of line pairs.

A recent example of a structural description recognition scheme is Biederman's (1985) theory of recognition by components (RBC). According to this scheme, objects are described in terms of a small set of primitive parts called "geons". These primitives are similar to the generalized cylinders used by Binford (1971), Marr & Nishihara (1978), Brooks (1981), and others. They include simple 3-D shapes such as boxes, cylinders, and wedges. More complex objects are described by decomposing them into their constituent geons, together with a description of the spatial relations between components. The number of primitive geons is assumed to be small (less than 50), and objects are typically composed of a small number of parts (less than 10).

In any scheme that relies on decomposition into parts it is crucial to devise a reliable and stable procedure for identifying part boundaries. Otherwise, the same object may give, under slightly different viewing conditions, different descriptions in terms of its constituent parts. In Biederman's scheme certain "non accidental" relationships between contours in the image are used to determine the part decomposition. These relations include, for example, the colinearity of points or lines, symmetry and skew symmetry, and parallelism of curve segments.

Another recent scheme employing part-decomposition is the "codon" scheme proposed by Hoffman and Richards (1986) for the description and recognition of contours. Contours are segmented at curvature minima [illegible] "transversality rule"). The resulting parts are then described in terms small "vocabulary" of shape primitives termed "codons".

The RBC and the codon schemes are complementary in that they emphasize different aspects of the problem. The codon scheme concentrates on the initial stages of analyzing image contours. Biederman's RBC scheme assumes that certain analysis of image contours has already been performed and then goes on to consider the description of complete objects.

Attempts have been made recently at combining these two levels of analysis into working systems that would actually recognize 3-D objects from their projections. An example is a recent scheme developed by Connell (1985). This scheme starts at the level of analyzing image contours. It first describes the contours in terms of constituent parts and their properties, using a representation scheme developed by Brady and his co-workers (Asada & Brady 1986). It then proceeds to generate higher-level constructs that eventually correspond to entire objects. The resulting description can become quite elaborate. Formally, it has a graph structure in which the nodes represent components and labelled arcs represent relations between parts. Recognition can proceed later by matching such graphs generated from the image with similar graph structures stored in memory.

Figure 2a shows an example of a contour image of an airplane, 2b shows the description generated by the system for a part of this figure (the right elevator).

The schemes mentioned above use primarily 1-D contour segments and 3-D volumes as their primitive shape parts. Other schemes use 2-d surface patches as their primitives (Dane & Bajcsy 1982, Potmesil 1983, Faugeras 1984, Brady *et al* 1985). There are significant differences between the various structural description schemes that have been proposed, but they all share a basic underlying idea: regularities in the families of views corresponding to an object (or class of objects) can be best captured by part-decomposition. Different schemes differ in the type of parts they use (contours, surface patches, primitive volumes, etc.), but they all attempt to employ simple parts, so that the identification of a part would be significantly simpler than the recognition of a complex object. The entire object is then recognized in a second stage in terms of the already classified parts.

For a variety of objects, the notion of part decomposition appears to be natural. A table, for instance, is often composed of a flat surface supported from below by four legs. Such a description appears much more natural than trying to characterize table-images in terms of simple properties such as total area, perimeter length, etc., as used in the invariant properties approach.

It is also true that, as argued by Hoffman (1983) and by Biederman (1985), human observers sometimes find it easy to identify the parts of an object even when the object is unfamiliar.

At the same time, it appears that for the purpose of visual object recognition the use of structural descriptions has at least two severe limitations. The first problem is that the decomposition into generic parts often falls considerably short of characterizing the object in question. For example, a dog, a fox, and a cat, (as well as several other animals) can probably have similar and perhaps identical decompositions into main parts. These animals are distinguishable not because each one has a different arrangement of parts, but because of differences in the detailed shape at particular locations (such as the snout). It may be argued, perhaps, that these animals are indistinguishable at the "basic level category" (Biederman 1985): they are first recognized perhaps as four-legged animals, and only a second recognition stage distinguishes among them. This possibility cannot be dismissed on the basis of current evidence, but at the same time, it is not clear that two such separate stages actually operate in this example. Moreover, the separation into two stages does not, by iteself, solve the problem: an explanation of how the objects are eventually recognized is still required.

A second limitation of the structural description approach is that many

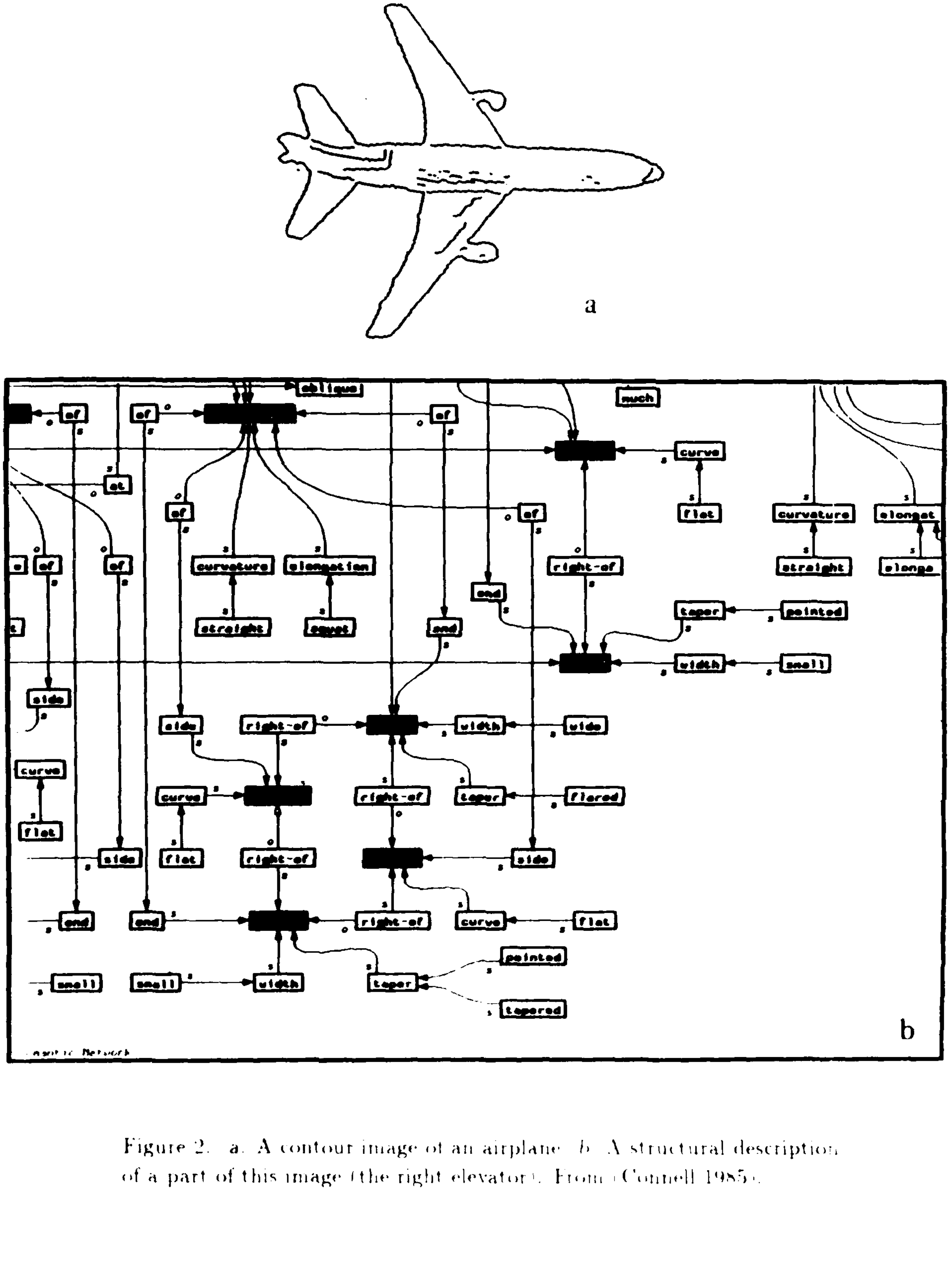


Figure 2. a. A contour image of an airplane. *b*. A structural description of a part of this image (the right elevator). From (Connell 1985).

objects do not decompose naturally into the union of clearly distinct parts. What, for example, are the decomposition of the shoe, loaf of bread, or rabbit shown in Fig. 1? It would be difficult to decompose these objects into parts that are sufficient to characterize the objects, and at the same time generic i.e., common to many other objects as well. A possible approach, illustrated by the aircraft example in Fig. 2, is to include in the description very simple generic parts, such as edges and line segments. The use of such parts causes, however, the resulting structural descriptions to be highly complex.

It seems, in conclusion, that for many objects the attempt to construct a structural description results in making strong commitments too early in the recognition process. The approach forces a categorization of shapes and relations into a small set of classes, and assumes that, as far as recognition is concerned, the internal structure (i.e., the details of the shapes and relations not captured by the structural descriptions) are immaterial.

The approach presented next (the alignment method) attempts to avoid these limitations. It preserves details of the viewed shape without enforcing a classification into predetermined categories of parts and spatial relations.

The alignment approach is not incompatible with the notion of part decomposition. Aspects of both approaches can, in fact, be incorporated in a single scheme. However, to keep the distinction between the approaches clear, the alignment approach will be presented first in a simple and "pure" form. Combinations with other schemes will be considered in a subsequent section.

4. THE ALIGNMENT APPROACH TO OBJECT RECOGNITION

To introduce the alignment approach, it is convenient to view visual recognition as a problem involving search in a large space: given a viewed object, a best match is sought in the space of all stored object-models and all of their possible views. If V denotes the viewed object, (M_i) are the different object-models stored in memory, and (T_{ij}) is the set of allowed transformations that can be applied to object-model M_i, then the goal of the search is to find a particular model and a particular transformation that will maximize some measure of fit F between the object and a model. That is, the search is for a maximum in $F(V,(M_i,T_{ij}))$ over all possible object-models M_i and their transformations T_{ij}.

The basic idea of the alignment approach is to decompose this search into two stages. First, determine the transformation between the viewed object and the object model. This is the *alignment* stage. Second, determine the object-model that best matches the viewed object. At this stage, the search is over all the possible object-models, but not over their possible views, since the transformation has already been determined uniquely in the alignment stage.

In terms of the maximization problem stated above, the idea is to determine for each potential object-model M_i a unique transformation T_{ij} that aligns M_i and V optimally. (It is also possible to transform the viewed object V rather than the model M_i; see section 6.) The search for a best match is now reduced to finding the maximum in $F(V, M_i)$ only; i.e., a search over the set of objects, but not over their different views.

A simple example may help to clarify the approach. The example is taken from the domain of character recognition. This is one of the only domains in which a rudimentary version of an alignment method has been attempted (Neisser 1966, Ch. 3), and it can be used to illustrate the differences between the alignment method and alternative approaches. It is, however, a somewhat special domain. Learning to recognize the letters in an alphabet is a difficult task that requires considerable training. It may require the use of some specialized skills that are not necessarily representative of object recognition in general. In sections 5 and 6 the application of a more general alignment approach to other objects will be considered.

Suppose that a character recognition system is required to recognize characters in the alphabet regardless of position, size, and orientation. A simple alignment scheme would proceed in the following manner. For each character, a single instance of the character would be stored in memory. Given an input character, the system will first go through an alignment phase. The goal of this stage is to "undo" the shift, scale, and rotation transformations. This may be accomplished by applying compensating transformations to the character. For example, to "undo" a possible shift, the center of mass of the input can be computed, and the character is then shifted, so that its center of mass always coincides with a fixed pre-determined location. In this manner, characters that differ in their position in the input image are "transformed back" to a canonical location. Similarly, scale can be compensated for by computing, for instance, the area of the character's convex hull. (The convex hull is the smallest convex envelope surrounding the character; see Preparata & Shamos 1985).

Orientation changes are more complicated to compensate for. (They are often more problematic in human perception as well (Neisser 1966, Rock 1973. Orientation can be determined for some letters on the basis of bilateral symmetry as in the case of (A, H, I, M, T, U, V, W, X, Y). Many characters have a line segment that, in the proper orientation, is oriented either vertically (B, D, E, F, H, I, K, L, N, P, R, T) or horizontally (A, E, F, F, H, I, J, L, T, Z), and these can be used to determine a small number of likely orientations.

The detection of bilateral symmetry and the orientation of the component line segments, together with the computation of center of mass and the convex hull area, would be performed during the alignment stage. After the shift, scale, and orientation have been compensated for, the "normalized" input is matched (possibly in parallel) against the stored representations of the different characters. Since the transformations have already been removed, the matching stage itself is expected to be relatively straightforward. At this stage, an associative memory-like mechanism may suffice to compare the tranformed input in parallel with the stored models. It should be noted, however, that even following the alignment the final matching cannot be as simple as, e.g., 2-D correlation between the contours. The difference between different characters, such as O and Q, may be a small but crucial contour element. Some parts of the model may therefore contribute more to the overall quality of the match than other parts.

The process of compensating for the transformations prior to comparing the viewed object with potential models is often referred to as a *normalization* stage. The use of such a normalization stage has been limited in the past to restricted applications, such as the domain of character recognition mentioned above.

The use of a normalization stage for more general object recognition suffers from two main shortcomings. First, normalization as used in the past has been usually restricted to changes in position, orientation, and scale, in the 2-D image plane. In contrast, the set of transformations that must be compensated for in 3-D object recognition is not limited to these transformations. When an object moves and rotates in 3-D space the transformations induced in the image are considerably more complicated. The second reason is that the methods used for normalization usually relied on global properties such as the object's apparent area, perimeter length, or center of mass. Such measures do not perform well in the face of occlusion, when only a part of the object is visible.

The alignment approach described in the next sections can be viewed as an extension to the simple notion of normalization. It has the same main goal, namely, compensating for the transformations separating the viewed object and potential object models prior to the matching stage. The main differences, detailed in Sections 5 and 6 below, are that: (1) the alignment process can compensate for a larger set of transformations, including rigid rotation in space as well as non-rigid transformations, (2) the proposed alignment method includes the use of abstract descriptions that are not usually incorporated in normalization schemes, and (3) the alignment process does not rely on global measures such as area or center of mass.

In the domain of character recognition, the normalization scheme outlined above would perform well provided that the set of allowable transformations is indeed limited to changes in position, scale, and orientation. If the input characters are allowed to change in a less restricted manner, so that additional distortions, changes in style, etc., are also permitted, these additional transformations should also be compensated for, as much as possible, during the alignment stage. Examples of a more extended set of transformations are illustrated in section 6.

The alignment approach can be contrasted with the two alternative approaches discussed above, the invariance properties and the part decomposition methods.

In the invariant properties approach characters are identified using properties that are supposed to be invariant with respect to position, size, orientation, style, etc. (Alt, 1962). The capital letter "A" may be identified, e.g., on the basis of relative width, height, and perimeter length, different moments, the fact that it contains a closed loop, etc.

A part decomposition approach may result in a structural description similar to the structure diagrammed in Figure 3. The character is decomposed into its main parts, and these parts and their relations are described in terms of a fixed "vocabulary" of part and relation types.

Variations and mixtures of these methods are, of course, possible. For example, if the orientation of the character can be determined in a preliminary stage, as in the alignment approach, then a structural description scheme would be able to utilize descriptions such as a "horizontal line segment" (for part-2 in Fig. 3). The invariant properties approach could also benefit from an alignment stage: measures such as overall line lengths, or orientation distribution of line segments could be utilized provided that the character has been aligned to a canonical size and orientation. The invariant properties method can be combined with part decomposition techniques and use, e.g., the number of line segments, vertices, and line terminators as invariants.

Although the domain of printed characters is a somewhat special one, the examples are nevertheless useful to illustrate in a schematic manner how the different approaches may be applied. Each of the above schemes has, in fact, been implemented in experimental character recognition schemes.

The Domain of Binary Vectors

The alignment approach can also be illustrated schematically using the domain of one-dimensional vectors used to illustrate the previous two approaches. For the current discussion, we shall assume that each vector can contain, in addition to $0's$ and $1's$, also a single occurence of the letter S,

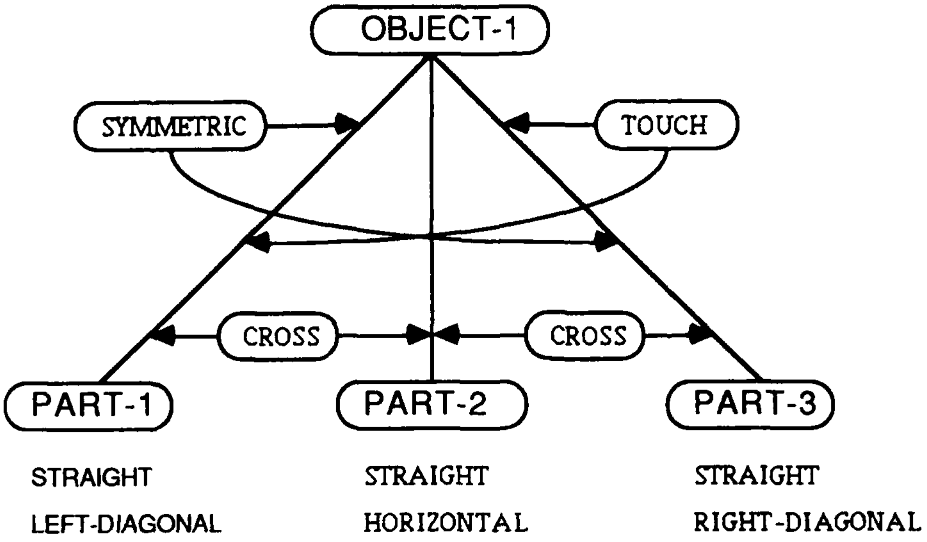


Figure 3. A simplified structural description of the letter "A".

which can be thought of as the "starting point" of the vector. Two vectors are considered in this example to represent the same object if they are identical when read, starting from the letter S, to the right, (and "wrapping around" if the letter S is not the first symbol in the vector). Thus, the vectors $V_1 = S100110$ and $V_2 = 0S10011$ would be considered members of the same class (see Table 4).

It would be easy in this domain to apply the alignment approach to "undo" the transformations and verify that V_1 and V_2 are, in fact, members of the same class. At the same time, the two vectors are separated by a relatively large Hamming distance and therefore the direct approach would not provide a useful comparison measure. Similarly, the invariant properties and part decomposition approaches would not be as effective and natural in this case as the alignment approach.

Which is the Correct Approach?

Object-1	Object-2
S 1 0 0 1 1 0	S 0 1 0 1 1 1
0 S 1 0 0 1 1	1 S 0 1 0 1 1
1 0 S 1 0 0 1	1 1 S 0 1 0 1
1 1 0 S 1 0 0	1 1 1 S 0 1 0
0 1 1 0 S 1 0	0 1 1 1 S 0 1
0 0 1 1 0 S 1	1 0 1 1 1 S 0
1 0 0 1 1 0 S	0 1 0 1 1 1 S

Table 4. *Recognition by alignment in the domain of binary vectors.*

Comparing this example to the examples in Tables 2 and 3, it can be concluded that there is no single best scheme that is appropriate for all cases. As the tables indicate, a given approach may be clearly superior for one set of conditions, but not for others. The different approaches represented by these tables should therefore not be classified as "correct" or "incorrect", but, rather, should be evaluated according to their usefulness in dealing with different types of object transformations.

This should not be surprising in view of the general discussion in Section 1. It was noted there that in the most general case, where different views of the same object (or class of objects) are distributed randomly in the space of views, a truly effective method that outperforms significantly the direct method would not be possible. To be effective, a recognition scheme must therefore exploit well the regularities inherent in a given domain. As shown by the schematic examples, different types of regularity would give rise naturally to different recognition schemes. The relevant question is, therefore, not which of the approaches discussed above is the correct one, but which would be useful for the purpose of shape-based visual object recognition.

In sections 2 and 3 it has been argued that the two most popular approaches, the invariant properties approach and the structural description approach, are insufficient for dealing with shape-based recognition. It seems to me that the alignment approach can provide an important, perhaps the main, missing ingredient.

The discussion above has introduced the general motivation behind the alignment approach. The next two sections illustrate the application of the method to the shape-based recognition of simple rigid objects (Section 5) as well as to more general non-rigid objects (Section 6).

5. THE ALIGNMENT APPROACH APPLIED TO SIMPLE OBJECTS

This section illustrates the application of the alignment approach to the recognition of simple objects. It discusses the problem of aligning objects in 3-D space using examples from a computer implementation by D. Huttenlocher that uses an alignment approach to recognize of objects of the type shown in Fig. 4.

The objects are flat machine parts that are allowed to translate, rotate in space, and change scale (as their distance from the camera changes).

Goal and restrictions

The goal of the recognition system is to demonstrate in a restricted and simplified application, how the alignment approach described above in general terms may be used for the recognition of objects. The domain of application of the current example is simplified in three respects. First, the objects considered are flat. It should be noted, however, that this is not a 2-D problem, since the objects are not restricted to move in the plane, but are allowed to move and rotate in 3-D space. The second restriction is that the transformations applied to the objects are limited to the class of rigid transformations, combined with changes of scale. Many real objects can undergo more complicated transformations, such as bending, stretching and other types of distortions. The class of allowable transformations is, however, less restrictive than many examples considered in the past. In various discussions of object recognition (Milner 1972, Baird 1984) the transformations that the recognition system is required to cope with are limited to changes in position, orientation, and scale. These are relatively simple transformations, that preserve the similarity of shapes. The transformations considered here are not similarity-preserving, because the objects are allowed to rotate in 3-D space.

The third simplification in these examples is that only a "pure alignment" approach is used. This means that the recognition scheme will not

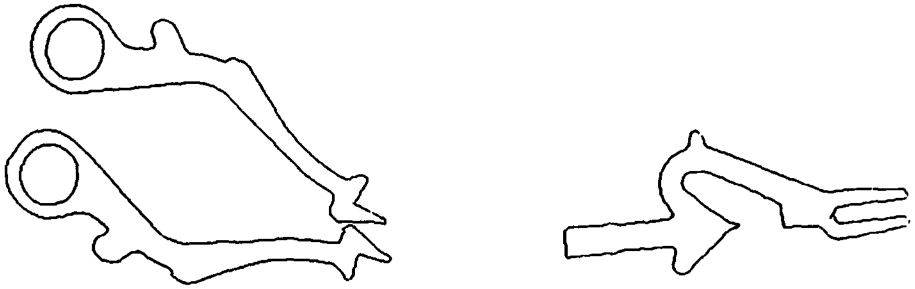


Figure 4. *Machine parts that were used in the recognition example.*

be combined with invariant properties or part decomposition methods, but will be used on its own. The alignment method described in this section uses the boundary and internal contours of objects as object models, without describing them further, or extracting invariant properties of objects or objects' parts. Such a simple description would be insufficient in more complicated situations, for example, when the objects contain parts that can move with respect to one another in a constrained manner. Useful combinations of the "pure" alignment scheme with certain aspects of other methods are possible, and will be discussed in section 6.

The Information Needed for Alignment: Three Points Suffice

In section 1 we have examined the "key problem" of visual recognition: defining regularities in the set of views that belong to the same object. That is, given two views, V_1 and V_2, that at the input level may be quite dissimilar (using simple distance metrics), the problem is to find methods for deciding whether or not the two views belong to the same object without necessarily storing both V_1 and V_2 separately in memory. The alignment method approaches this problem by noting that objects do not change in an arbitrary manner: the set of transformations applied to them is often restricted. The key point about these restrictions is that the transformation can be determined uniquely on the basis of very limited information.

Three-Point Alignment

To illustrate this concept, assume for the present that three dots, a red one, a green one, and a blue one, have been painted on every object in the collection the system is required to recognize. The exact location of the points on the object's surface is immaterial. They must only be visible, and must not be colinear. We will call these points, which are used in the alignment stage, the "anchor points" of the object.

For each object 0_i in the collection, the system stores an internal model, M_i, which is simply a picture of the object in a frontal view. This picture is an orthographic projection of the object on the image plane. It includes the projection of the object's boundary, as well as the position of the three anchor points (see Fig. 5). The real projection of objects on the retina or a camera's image plane is, of course, perspective rather than orthographic, but an orthographic projection combined with an admissible scale change provides a good approximation unless the projection center is very close to the viewed object.

We are now given a view of an unknown object, and the problem is to decide, for a given model M_i, whether or not V matches M_i (i.e. whether V is a possible view of M_i). To reach a decision, we can at first ignore the entire image of the object, and examine only the position of the three anchor points. Let (P_1,P_2,P_3) be the (3-D) coordinates of the three points in the model, and (p_1,p_2,p_3) their 2-D image coordinates.

The crucial point is that the model M_i and the view V can be aligned in a unique manner given only the coordinates (P_1,P_2,P_3) (known in the model) and p_1,p_2,p_3 (recovered from the image). In other words, the displacement D, the rotation in space R, and the scaling S, possibly relating M_i to V, are uniquely determined on the basis of the three corresponding points. These transformations are now applied to M_i. Following the transformations, M_i and V should be in complete registration (Figure 5). Unlike the original situation, M_i and V following the transformations are very similar in the Hamming or similar distance metrics. If V is not an instance of M_i, then M_i and V following the compensating transformations would still be out of register (Figure 6). The recognition process is decomposed in this manner into two stages: an initial alignment followed by a matching stage.

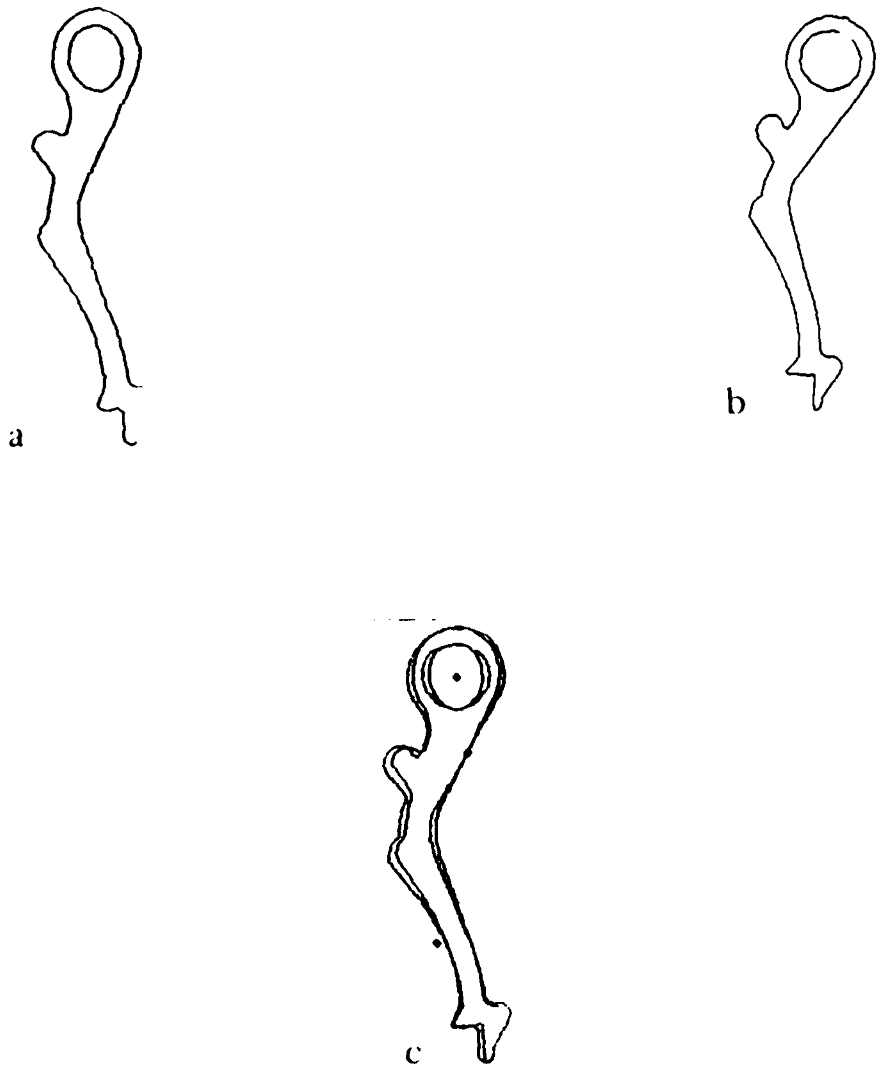


Figure 5. Matching an object with the correct model using alignment *a.* A part-image. *b.* A part-model. The image is both larger and slanted. *c.* The aligned model superimposed on the image. The points used for alignment are marked by black dots. Following the alignment, the model and image are in close agreement

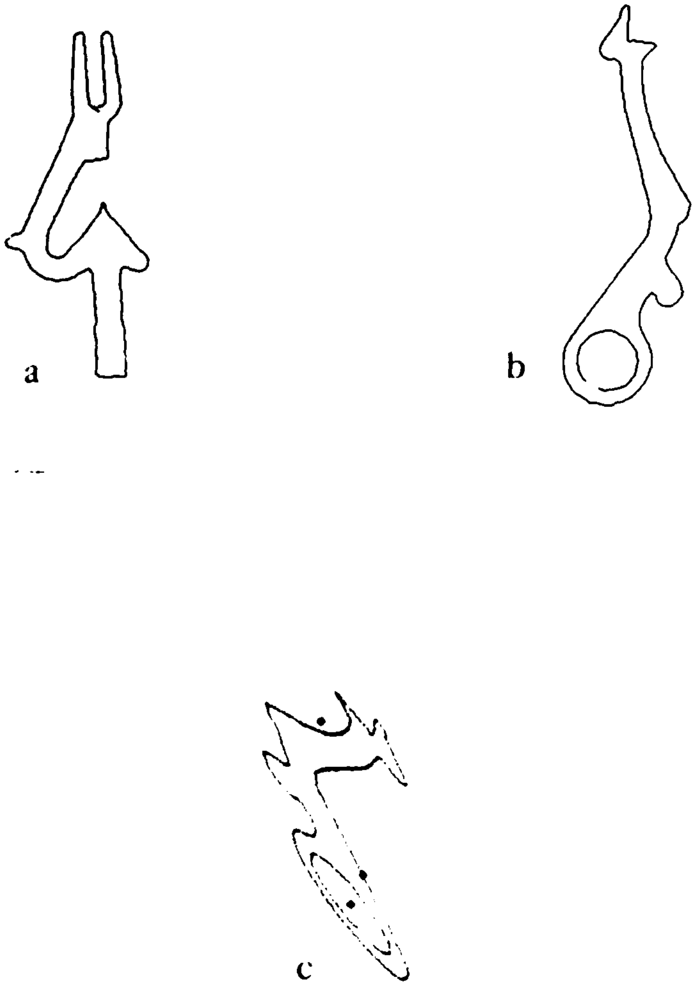


Figure 6. Attempting to match an object with an incorrect model using alignment. (a) A part image. (b) A part-model. (c) The model transformed in an attempt to align it with the image. The points used in the alignment are marked by black dots. Following the alignment, the model and object are still in disagreement.

The fact that three corresponding points are sufficient to "undo" the rotation, translation and scale is shown in Appendix 1. These transformations can be specified by six parameters: three for the rotation, two for the translation (under orthographic projection, absolute depth remains undetermined), and one for the scaling. Three points supply six equations (two for each point) and therefore the number of constraints matches the number of unknowns. This counting argument by itself is insufficient (a more complete proof is therefore given in the appendix), but it suggests why such a small number of points may be sufficient for recovering the transformation uniquely.

It is worth noting that, as shown in Appendix 1, the alignment stage does not require the extraction of 3-D information from the image: the 2-D coordinates of the points are sufficient. Three-dimensional information could be used, when available, to simplify the alignment stage somewhat, but the process can proceed in the absence of precise 3-D data.

The recognition system illustrated in this section did not in fact use colored points painted on the object. Instead, it identified a small number of salient points defined by the object's boundary. Such points included deep concavities, strong maxima in curvature, and the centers of closed or almost closed blobs. The anchor points identified and used by the recognition program are marked in Fig. 5 and 6. For more discussion on the extraction of alignment anchor points see (Huttenlocher & Ullman 1987).

Instead of the color of the points, the scheme uses simple labels to determine uniquely the correspondence between image-points and points in the model. A label of a point includes a point-type, such as blob-center, concavity, or curvature maximum, and may include a rough description of location. It is desirable, although not strictly necessary, to obtain a unique correspondence between object and model anchor points based on their associated lables. If this correspondence is not unique, a number of transformations will have to be evaluated, for the different possible transformations.

Following the alignment, a simple matching measure (similar to the Hamming distance) was sufficient in this application domain to unambiguously select the appropriate model. For more general recognition problems such a matching criterion may not be sufficient. More general considerations regarding the final matching and model selection are discussed in Section 7.

An alignment scheme somewhat similar to the three-point method has been used recently in Lowe's SCERPO system (Lowe 1985, 1986). SCERPO is one of the only systems in existence that attempts to use an alignment method for recognizing objects in 3-D space. The alignment procedure used in the system relies on perspective rather than orthographic projection. The alignment is not performed in a separate stage, it is intertwined with the recognition process. This is implemented as an iterative scheme, based on Newton's method. Another difference between the two alignment procedures is that SCERPO does not attempt to "label" the alignment features in a manner that will eliminate or reduce the required search between corresponding image and model features.

Alignment Using Simple Image Transformations

For the flat objects considered in this section, the alignment phase can be decomposed into a sequence of simple operations acting on the image. These operations include translation and rotation of the image, scaling along one axis, and a "shear" transformation: scaling along one axis by an amount that varies linearly with the distance from an orthogonal axis. The order of these operations and how they are used to bring the viewed object and model into correspondence is described in Appendix 2.

Orientation Alignment

Unique object-to-model alignment can be performed (for rigid transformations accompanied by scale changes) using three identifiable anchor points. The three-point scheme is only an example, other types of alignment schemes are also possible. In particular, if the object has a well-defined orientation, then this orientation can be used for alignment instead of the anchor points.

A number of properties can be used to define an overall orientation for an object, including overall elongation, bilateral symmetry or skew symmetry, oriented texture on the object surface, the existence of a flat or nearly flat side, the distribution of mass, and the existence of salient protrusions or nicks. The process of alignment by orientation is illustrated schematically in Fig. 7, and described in more detail in Appendix 3. The viewed object (or the model) is first rotated to align their orientations. Let this common orientation denote the y-direction. The object is next scaled along the x-direction so that the viewed object and the model match. A final scaling and shear (in the y direction) completes the alignment. (In Fig. 7 the final y-transformation is pure scaling, no shear was necessary.) The amount of scaling and shear can be deduced from any three locations along the object's boundary (for details, see Appendix 3).

Orientation alignment is simpler than the three-point scheme, since orientation is often easier to extract in a reliable and consistent manner, compared to the extraction of discrete identifiable points. The main disadvantage of the orientation scheme is in the case that the object or its image does not have a clearly defined orientation. There are indications that such cases can also cause difficulties for recognition by humans. That is, in the absence of

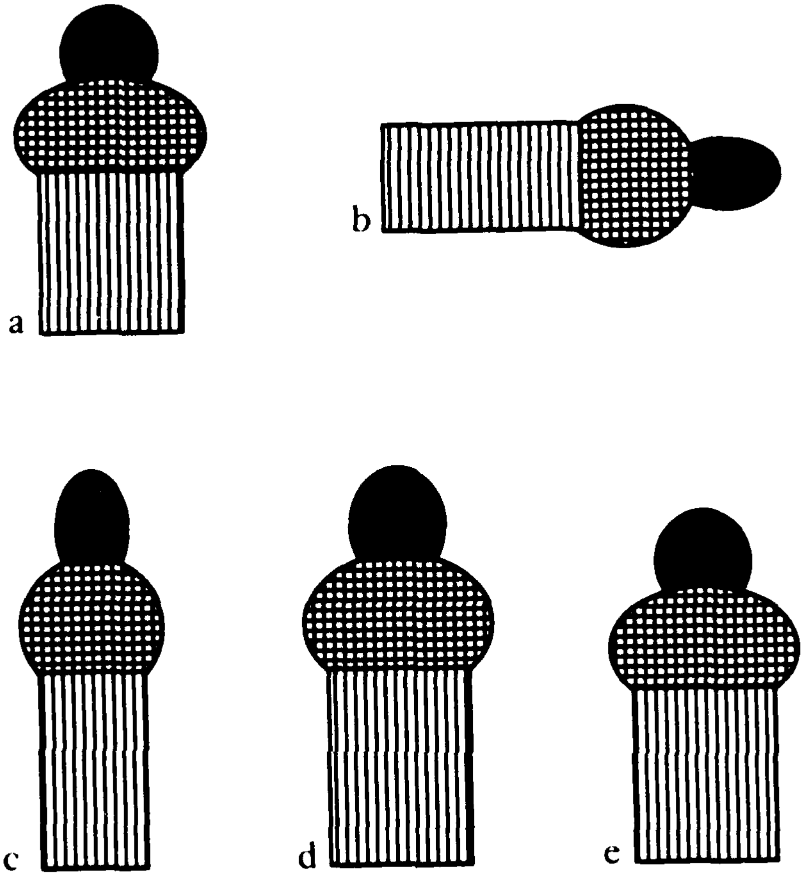


Figure 7. *Orientation alignment. An object (a) and its model (b).* The model is rotated *(c)*, scaled in x *(d)* and in y *(e)*. In this case no y-shear was necessary.

a well-defined orientation, human observers are more likely to fail to recognize shapes that are, in fact, identical (Rock, 1973). This suggests that the extraction of dominant orientation may play an important role in the recognition process used by the human visual system. It also appears that when the object lacks (perceptually) a preferred orientation, the human visual system may use instead an externally defined orientation (such as the direction of gravity, or the orientation of the page, in the case of printed pictures) to define orientation for alignment (*ibid.*). If orientation is indeed used by the human visual system for alignment purposes, there are some limitations on its use. In particular, complicated figures such as faces, are difficult to recognize when their orientation differs substantially from the familiar one. This appears to be a general limitation, not specific to faces alone.

Regardless of the exact form of the alignment scheme, the important point is that alignment can be performed using only very limited information. I shall refer to the information extracted from the image to align the viewed object with a candidate model as its "alignment key". We have seen above examples of two such alignment keys. One consists of a small set of identifiable anchor points (three, for the case of rigid motion and scale changes). The second consists of a dominant orientation, together with fragments of the object's boundary (Appendix 3).

The kind of information required for the alignment keys can, in principle, be extracted from the image in a bottom-up manner. The reason is that the alignment keys are defined by the object's bounding contour, a small number of salient points, dominant orientation, etc. This kind of information can usually be extracted by processes that do not require object-specific knowledge. Using such processes the alignment key can be extracted first, and the object can be aligned with a potential model, before the object's identity has been determined. After alignment has been performed, the viewed object and the model should be in close agreement (under ideal conditions they should match exactly) and therefore the task of determining the closest match becomes relatively straightforward. At this stage, a comparison (potentially in parallel) of the aligned object with all of the models using a simple comparison method (analogous to the Hamming distance) becomes feasible.

6. THE ALIGNMENT OF FLEXIBLE OBJECTS

In this section, the alignment scheme described above for flat objects transforming rigidly is extended to deal with non-flat objects that are allowed to transform in a non-rigid manner. The goal is not to discuss the problem of recognizing such objects in detail, but mainly to support the claim that alignment schemes can play a useful role in the recognition of large classes of objects.

Dealing with flexible objects is important for the purpose of object recognition for two reasons. First, many objects such as animals and faces can change in a non-rigid manner. Second, the differences between individual members of the same class of objects, such as two apples, can often be viewed as small non-rigid distortions. Dealing with flexible distortions may therefore provide a tool for handling classes of similar objects

It also appears that for recognition by the human visual system strict rigidity is not crucial. Objects can be recognized easily in a distorting mirror provided that the distortions are not too extreme. Object models can be constructed (e.g. from playdough) and distorted without affecting recognition severely. For animals such as the pigeon, rigid transformations play an even lesser role in recognition. Pigeons can learn to recognize a large variety of objects (including people, particular individuals, fish, and characters in the alphabet) from different views and in different contexts. In recognizing objects, they apparently do not distinguish, however, small non-rigid distortions from rigid transformations of simple 3-D objects (Hernstein 1984).

Treating flexible objects as locally rigid and planar

A straightforward generalization of the simple alignment scheme is to treat regions of the object as locally planar and rigid. This generalization requires two extensions of the simple scheme:

- *Use more than the minimum set of three anchor points.*
- Treat local regions of the object as semi-rigid.

The seoncd of these extensions can be implemented by a simple extension of the three-point alignment scheme outlined above (Huttenlocher & Ullman 1987). The extension is obtained by imposing a triangulation on the set of anchor points. Suppose, for example, that five anchor points have been selected for alignment. As in Section 5, these points may be curvature extrema, the extreme points of elongated parts, etc. The spatial arrangement of the points themselves (without the contours to which they belong) is shown in Figure 8a. In Figure 8b, a triangulation has been applied to the points. (A triangulation of a set of points means that the points are connected by non-intersecting lines in such a way that every region internal to the convex hull of the points is a triangle, see, e.g., Preparata & Shamos 1985.)

Each triangle is now aligned exactly as before, using its three vertices. This alignment induces a transformation to all the contours internal to the triangle. In this manner, the alignment of the anchor points defines a transformation for the entire object. (If the anchor points are all internal to the object, some pieces of its bounding contour will fall outside the triangulated area. These pieces can be treated separately, but this issue will not be discussed here.)

As before, the final stage consists of comparing the transformed object with each candidate model. Two examples of this alignment procedure are shown in Figure 9 and 10.

Figure 9 *a* and *b* shows two rabbits that are initially quite different. Figure *c* shows the first rabbit with the anchor points that have been selected and their triangulation. The corresponding anchor points in the second rabbit are shown in d. Figure e shows the superposition of the two rabbits following the

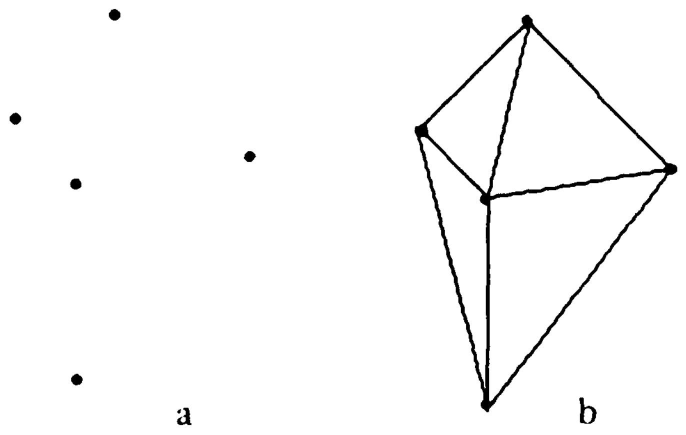


Figure 8. Triangulating a set of points. *a*. The original set. *b*. The set triangulated.

alignment. Figure j shows the superposition of the two rabbits without any alignment. It can be seen that the alignment is sufficient to bring the two figures into close agreement.

Figure 10 shows a similar sequence applied to two different objects, the car and the rocking-horse (a,b). Anchor points were selected manually on the two figures in an attempt to bring them to the closest possible match. Figure e shows the rocking horse following the alignment; it has been transformed to approximate the car figure as much as possible. Fig. f shows the aligned figures superimposed. Clearly, the agreement between the two figures is still poor.

The exapmles in Section 5 used flat objects in rigid transformations. In this section the objects were not necessarily flat, and the transformations were not assumed to be entirely rigid. There are also a number of intermediate cases that deserve special attention.

A common case is one where the objects are general 3-D objects, (rather than flat), but the transformations are assumed to be strictly rigid. There

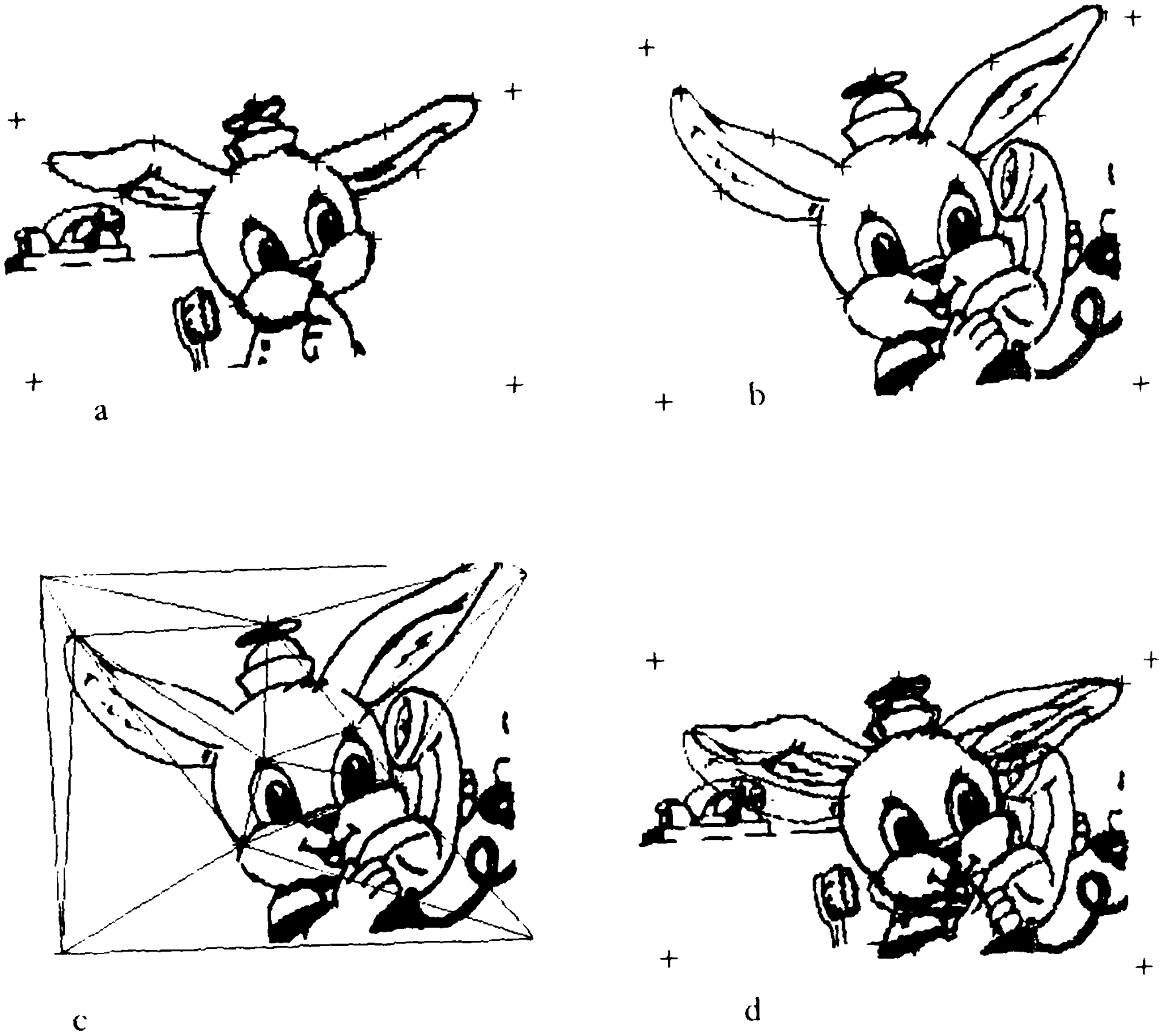


Figure 9. Matching an object to a model using flexible alignment. *a.* A rabbit-image. *b.* A rabbit-model. *c.* The model triangulated. *d.* The transformed model and the rabbit-image superimposed. Initially, the image and model are quite different. Following the alignment they are in close agreement.

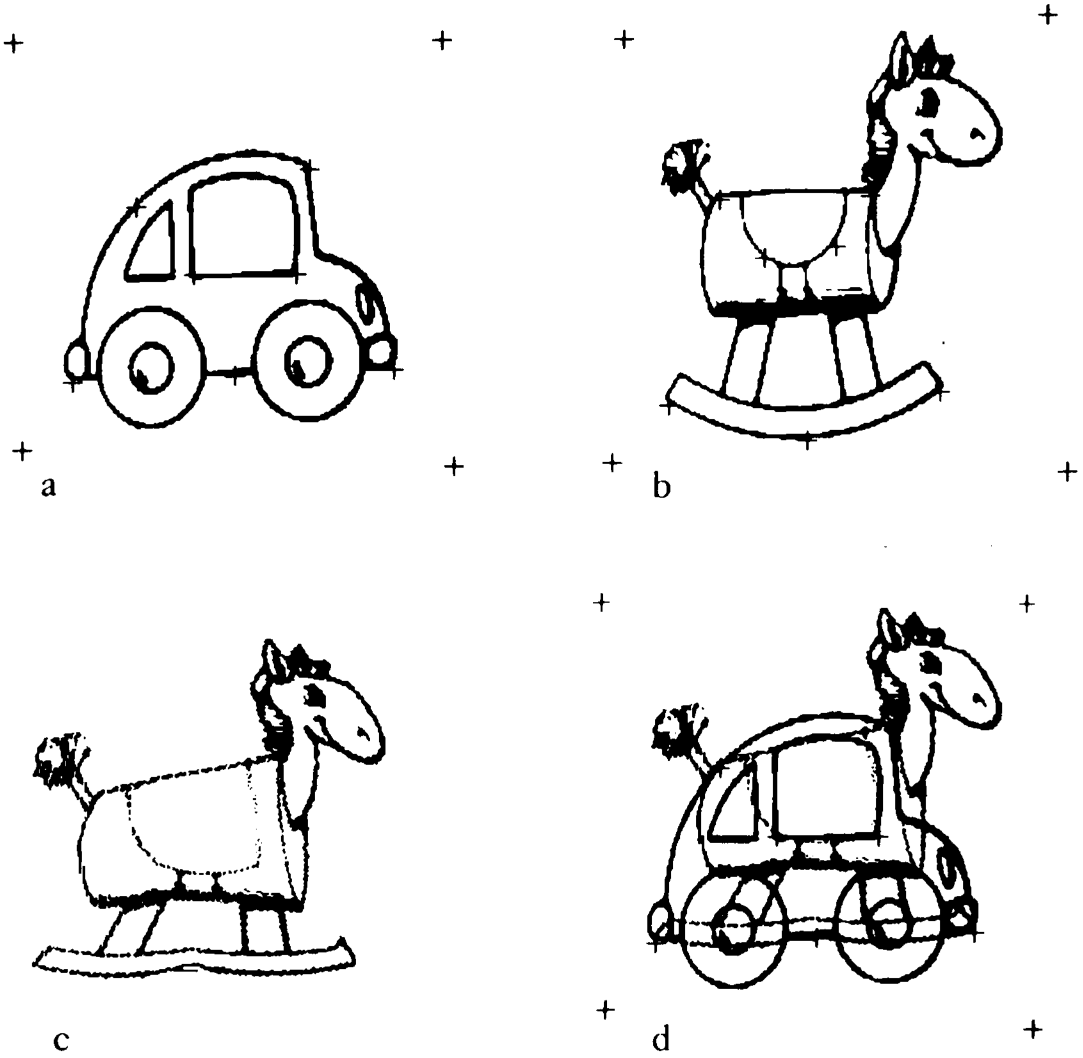


Figure 10. An attempt to match an object with an incorrect model using flexible alignment. *a.* A car-image. *b.* A horse-model. *c.* The model transformed to align it with the image. *d.* The transformed model and image superimposed. Because the model is an incorrect one, the agreement between the two figures following the alignment remains poor.

are two possible approaches to this problem within the general alignment framework. One is to maintain a single 3-D model for each object, and use the 3-D transformations recovered in the alignment stage to transform the model into alignment with the viewed object. The required transformation will be complicated from a computational standpoint. The computation will require, for instance, a process of "hidden line elimination", i.e., the computation of which object features are visible from a given viewing position (Lowe 1985, 1986).

An alternative possibility is to store a number of models corresponding to different viewing positions, and to use, for instance, an alignment procedure similar to the one outlined above for non-rigid objects. The required computations may be simplified, but at the expense of accuracy: it will become more difficult to ascertain whether two different views correspond to exactly the same 3-D object, or to slightly different shapes.

In the first of these alternatives the model is truly object centered and view independent (Marr & Nishihara 1978). In the second, the representation is view dependent, since a number of different models of the same object from different viewing positions will be used. (Perrett *et al* 1985). It is expected, however, to be view insensitive, since the differences between views are partially compensated by the alignment process.

As far as the human visual system is concerned, there are indications that observers can identify under certain situations views that correspond to the same 3-D object from widely disparate viewing positions (Shepard & Metzler 1971, Shepard & Cooper 1982). The process involved in these judgements appears to be slow, and may be restricted to relatively simple shapes. It is still unclear, therefore, whether this process is an integral part of ordinary object recognition, or a special process that is used for special purposes only.

A second intermediate case of interest concerns articulated objects, containing parts that can move with respect to one another in a constrained manner. Examples include objects with hinges and joints, such as a pair of scissors, a hand, a limb, etc. An object of this type has associated with it a set of allowable transformations that are less restricted than the rigid transformations discussed in Section 5, but more constrained than the non-rigid transformations discussed above. The problem of representing and matching such objects is a difficult one, and it will not be examined here. It appears, however, that the notion of an alignment scheme will be applicable to such objects as well. The transformations separating a model of an articulated object of this type and a particular view of it can still be determined on the basis of partial information. This information can be used, as before, to compensate for the transformations, and bring the viewed object and the model into alignment.

The discussion in Sections 5 and 6 suggests that if the alignment stage is performed properly, then, for both rigid and non-rigid objects, the match with the correct model would stand out as significantly better than with all other models. The alignment stage itself raises, however, a number of difficulties. For example, the anchor points, or other alignment keys, such as the orientation of the object, oŗ of its parts, must be extracted reliably from the viewed object and these points must be matched against the corresponding locations in the model.

These and related difficulties are examined in the next section.

Two Main Requirements from the Alignment Approach

Each of the approaches to visual object recognition makes a number of critical assumptions that give rise to certain difficulties when the approach is applied to large classes of natural objects.

The invariant properties approach assumes that simple invariant properties would be sufficient to characterize the different objects. But finding properties that are feasible to compute and at the same time powerful enough to characterize uniquely a large variety of objects did not prove successful, and the approach has been applied to limited domains only.

The structural description approach assumes the existence of categories for both parts and spatial relations that are sufficiently sensitive and stable (Marr & Nishihara 1978). They should be sufficiently sensitive to be able to make the required distinctions between objects, and at the same time stable enough to produce the same description for different instances of the same object (or class of objects). This proved to be difficult, especially for the categories of spatial relations. The structural description approach also faces some difficult computational problems – such as the reliable segmentation into parts and the computation of their spatial relations.

The use of alignment in the course of object recognition raises two main problems. The first is performing consistent alignment in a bottom-up manner. The second is a computational problem of transforming a large number of models.

Consistent Bottom-Up Alignment

For alignment to be successful, the information required for alignment (the alignment key) must be extracted reliably from the image. This stage is performed early in the recognition process, and therefore it must depend only on general image properties such as the saliency of some special points rather than on properties associated with specific objects. If alignment is performed, for example, on the basis of three points, then a small number of points, ideally always including the same three, must be extracted from the image in a reliable and consistent manner.

At least two factors can help this alignment process. The first is the use of object orientation. As mentioned in Section 5, when a dominant orientation for the object can be computed, it can substantially facilitate the alignment process. The alignment process can also be facilitated by using a number of different models for the same object. If the views become too disparate so that the use of the same alignment key becomes difficult, a new model can be added to the library of models.

It remains to be seen to what extent alignment keys can be extracted reliably from object images. The task appears, however, less demanding than the problem of decomposing an object in a consistent and reliable manner into all of its constituent parts. Alignment requires less information, and relatively stable-prominent properties can be used for the task, such as the most salient points associated with a given object, or its dominant orientation.

It is also possible that in the recognition of a specific object alignment may sometimes be obtained in more than a single stage. A matching may first be obtained with a general category, such as a face. This match may trigger routines for extracting features that can serve as useful additional anchor points, such as the eyes, even in cases where these features were not particularly salient in the image.

Transforming the Models

In matching a viewed object to a potential model using the alignment method, one of them (at least) must be transformed to compensate for the transformations between the two. It is possible to transform either the viewed object, or the stored model, (or both).

Applying the alignment transformations to the viewed object only has one important advantage: the transformation is applied only once. All the models remain unchanged. This can be accomplished provided that the various models are stored in memory in a common "canonical" form. Consider, for example, the case in which the viewed object is aligned to the model on the basis of three anchor points. For simplicity assume that each model has exactly three anchor points. An alignment transformation applied to the viewed object must bring the three points into alignment with the corresponding points in all of the potentially relevant models simultaneously. This implies that all of these models must be stored in a canonical form, in which the three anchor points are already in register.

A canonical form may be defined in an analogous manner also for alignment based on dominant orientation rather than anchor points. In either case, however, the use of canonical forms for the models also has its drawbacks. One complication arises if it is desired to recognize the same object using different alignment keys. Such a redundancy is useful. e.g., for dealing with occlusion. In a canonical form scheme, each model will have to be represented by multiple copies, one for each different alignment key.

An alternative would be to apply an alignment transformation separately to each of the potentially relevant models. In this case, the models need not be stored in any canonical form, since each one is transformed individually to align it with the viewed object. This also has the advantage that different transformations may be applied to different models. For example, the model of an object may include 3-D information that is not available from a particular single view of the object. This 3-D information could be used in transforming the model and predicting how it will appear from a different viewpoint. The model may also specify, for instance, that a certain point in the object can serve as a joint, where parts can change their relative orientation. For this model, but not for other ones, an attempt to align the model with the object may include bending around this known point. Such individual transformations add flexibility to the matching process, but at the cost of increased computational effort.

It is not clear at this stage which approach (transforming the viewed object or transforming the models) should be preferred. Two additional considerations are relevant in this regard. First, it is not necessary to adopt an extreme approach, a combination of the two is also possible. For example, oriented objects may be stored in a canonical orientation. The viewed object is then rotated once to bring its own orientation into alignment with the canonical orientation of the models. Following this common stage, an additional transformation, such as change of scale, may be applied to each model individually. More generally, the mixed approach is to apply to the viewed object all the alignment transformations that are common to all of the relevant models, yet allow the application of additional transformations to the different models. The second comment is that it may be possible to keep the transformations applied individually to the different models simple in nature, e.g. some scaling or stretching along one direction (as discussed in Appendix 2). When the discrepancy between a particular view of an object and its models already stored in memory becomes too large to be overcome using these restricted transformations, an additional model of the object can be added to the model library. The transformations that are applied individually to the different models may therefore be kept sufficiently restricted, *so that the computational load required for applying the transformations to* many objects in parallel may be kept within reasonable bounds.

Aligning Pictorial Descriptions

The last few sections have discussed the alignment approach in its "pure" form. The example used the unarticulated object boundaries, without defining parts in the object, and without using abstract descriptions for object parts, as done in the object decomposition approach.

It is also possible, however, to combine the main advantages of the part-decomposition approach with an alignment approach. The resulting scheme appears to be more suitable for dealing with the recognition of various objects that cannot be handled easily by either method alone.

Consider for instance the rooster sketch in figure 11. An internal model for this figure in a structural description method will contain a number of parts with their associated shape descriptions, and a description of the spatial relations among the various parts. A pure alignment method would keep a replica of the figure as an internal model. In evaluating the match between this model and a new viewed object, which is another possible instance of a rooster figure, the method will first try to align the model and the viewed object as precisely as possible. Clearly, however, *the details of the rooster's* crown have no particular importance in the normal process of recognizing such a figure. The part decomposition method seems to offer a more appropriate approach in this case. As mentioned in Section 3, the main step in this method is to start the many-to-one reduction at the part level. The details of the part depicting the crown will be ignored and replaced by a more abstract description, perhaps a "wiggly contour" of a certain type. The same kind of abstraction can be used in the alignment approach as well. One can imagine a "label" stating "wiggly line" being overlaid over the crown contour. This more abstract label is associated with a given location in the figure and it is shifted along with it in the coarse of the alignment process. When the aligned figure is then matched against the rooster model, the detailed internal contours of the crown in the aligned object and the model may not be in good agreement, but they will both have the same label in corresponding locations.

There are two differences in the manner that abstract descriptions are used in the alignment scheme compared with the structural description approach. First, in the alignment method abstract descriptions do not replace lower-level descriptions – they are added to them. A match may eventually occur at a low level (the actual contours may be in close agreement), or it may occur at a higher level (the corresponding abstract descriptions may match without a good match at the lower level). In the pure structual description

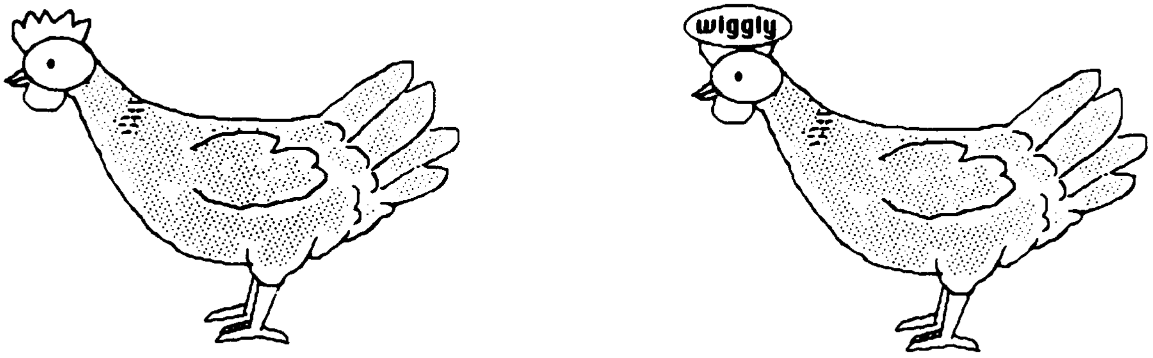


Figure 11. The use of an abstract label in a pictorial description.

method, without alignment, the low-level components such as boundary contours cannot be expected to match. The scheme must rely instead entirely on the correct categorization of parts; i.e., that different views of the same part will end up with the same abstract description. Unlike the part decomposition scheme, in the alignment scheme the part decomposition is therefore not required to be complete. Abstract labels may be associated with some locations, while other pieces of the object may remain unarticulated, not broken into parts, and not assigned to any category, or described by any abstract descriptors. Because of the alignment stage, which is not used in the structural description approach, these unarticulated parts are expected to produce (following the alignment) a good match with the stored model.

The second difference is that in the alignment method the description may be called "pictorial". It is much closer to the image compared with structural descriptions. In structural descriptions, spatial relations, like part shapes, are described using a limited set of categories such as "above" "in between" "near", etc. The position of part *A* may be described as "above *B* and near it, and to the left of *C*". This description is abstract in the sense that many different configurations in the input would fit a given category such as "above" or "left of". In the alignment approach, in contrast, spatial relations are not categorized. Instead, the actual position of parts and labels is preserved. The resulting description consequently has an image-like structure in which labels are associated with particular locations. Unlike part-decomposition schemes. descriptive labels are associated with specific locations, without requiring a precise delineation of part boundaries. In such a scheme it is natural to associate descriptions with locations such as the cheeks or forehead in a face image. These are well-defined locations, but not precisely delineated parts in the sense used in part-decomposition schemes.

The combined scheme, using alignment as well as abstract descriptions. can be described as the *"alignment of pictorial descriptions"*. This name implies three components. First, it is an alignment method. Second, it also uses (unlike the examples in Sections 5 and 6) abstract descriptions. Third, these descriptions are used pictorially: they are associated with specific locations, rather than being described by spatial relation categories. Such descriptions can be rotated, scaled, stretched, etc. prior to the matching stage.

The entire object recognition process is, in the alignment approach, less symbolic, more pictorial, and closer to the lower-level visual processes, than the structural description approach.

7. STEPS IN THE RECOGNITION PROCESS

The last section has advanced the notion of aligning pictorial descriptions as a general approach to the regularity problem in object recognition. This approach does not specify directly the processing stages that must take place in extracting the information from the image prior to the alignment stage, or the matching that takes place following it. These processing stages are required not only in the alignment scheme, but also in most other recognition schemes that have been proposed.

To put the alignment scheme in perspective, this final section will list briefly some of the major steps that are involved in the recognition process, and describe the problems that they raise.

Selection. By "Selection" I mean identifying in the image a region that is likely to contain an object of interest. A human observer rarely scans the entire scene in a systematic manner. Very often, objects of interest somehow attract our attention, and subsequent processing seems to be concentrated at these locations. Lowe (1986) has proposed a scheme in which feature configurations that have the least probability of arising by coincidence are examined first. (A similar notion has been suggested by Witkin & Tenenbaum 1983.) In human vision the initial selection appears to be based on simpler criteria The human visual system seems unable to extract relational properties among features in the early, pre-attentive, parallel stage (Treisman & Gelade 1980). Selection may be based instead on some measure of saliency defined by local differences in contrast, color, size, orientation, etc. (Mahoney 1986).

Segmentation. By "segmentation" in this context I mean the delineation of a sub-part of the image to which subsequent recognition processes will be applied. Segmentation schemes have been investigated extensively in the field of image processing, but their goals are usually more ambitious than what is required for recognition by alignment. For example, they often attempt to segment the entire image, as opposed to just the region of interest. Segmentation for recognition, applied to the region of interest only, can therefore be obtained by universal routines (Ullman 1984) that are spatially focused, rather than as a part of the base representations, where the computation is spatially uniform. For recognition by alignment, the main requirement from the segmentation stage is that the alignment key will be selected from a region that is likely to correspond to a single object. The exact delineation of the entire object is not of major importance at this stage.

Description. The next stage involves the extraction from the region of interest the information that will be used for matching the viewed object with stored object-models. Most recognition schemes propose that the viewed object is described for this purpose in some fashion, using 1-D contours (Baker 1977), 2-D surface patches (Dane & Bajcsy 1982, Potmesil 1983, Faugeras 1984, Brady *et al* 1985), or 3-D volumetric descriptions (Marr & Nishihara 1978, Biederman 1985).

An important decision at this stage is to what extent the description of the viewed object should rely on detailed 3-D information. Some recognition schemes (see Besl & Jain 1985) assume the availability of a detailed and precise depth map of the visible surfaces. Such information is not always available in the image, and from human vision it appears that recognition can often proceed in the absence of detailed 3-D information. It is desirable, therefore, for the recognition process not to depend critically on detailed 3-D information, although such information may be used when available.

If detailed 3-D information is not required, it appears that descriptions based on object contours are better suited for the recognition task than surface based, and to some extent volumetric, descriptions. At the same time, it is important not to identify object contours with intensity edges. Many intensity edges in the image are irrelevant for the purpose of recognition, and recognition can proceed in the lack of intensity edges altogether For example, objects can be recognized in random dot stereograms In this case object contours are defined, e.g., by discontinuities in depth and surface orientation, but not by intensity changes.

Alignment key extraction. The alignment key is used to bring the viewed object and internal models into alignment. As discussed in section 5, a number of different alignment procedures may be used, depending on some properties of the viewed object. For example, if it has a clearly defined orientation, then this orientation may be used for alignment. If the object is unoriented, the alignment key may be composed of salient points

Alignment. This stage brings the object into register with potentially matching objects. As suggested in Section 6, it may be possible to break down the alignment stage into two successive steps. In the first, which may be called "common alignment", the viewed object is brought into correspondence with a large number of models stored in memory in some canonical form. The second stage is composed of individual alignments: different models align themselves individually to the viewed object. A number of problems remain regarding the parallel execution of this stage. Can a large number of models be aligned simultaneously? If not, how can the load required by individual alignments be reduced?

Model filtering. Following alignment, the degree of match between the viewed object and different models must be assessed, and the best match selected. A number of different recognition schemes precede the final match with a process of model filtering. The goal of this stage is to use some simple criteria to "filter out" unlikely models, and obtain a smaller set of likely candidates. In other schemes this stage also includes rank-ordering of the models, so that matching with the more likely ones is attempted first.

It is not clear, however, that model filtering of this type can lead to significant savings in the required computations. If we start with a large number of models, it is probably unreasonable to expect that a simple filtering scheme would be sufficient to select a small number of candidates, since this will place the burden of the recognition process on the filtering stage. It seems, therefore, that the viewed object will have to be matched, perhaps in parallel, against a large number of object models.

Similarly, rank-ordering the models is not likely to result in substantial savings. In many instances the matching process will not result in a perfect match. We still wish to retrieve in these cases the best matching model. This means that the matching process will have to be fairly exhaustive, unless a perfect match is encountered It seems, in conclusion, that filtering and rank-ordering may help to limit the search in some specific sequential implementations, but in the more general case matching against a large number of object models is probably unavoidable.

Matching. Following the alignment stage, the correct model and the viewed object are expected to be in better agreement, but usually differences between them will still exist. A measure of the degree of match is therefore required to decide which of the models resembles the viewed object most closely. I will not attempt to define such a measure, but only define three general requirements for this measure.

First, as mentioned in Section 4, the contributions of different parts of the object to the match quality may carry different weights. Some parts may be small in size, but still be crucial for defining the object. In some cases it is also expected that the distinction between highly similar objects may require an additional separate stage. Two objects that differ only in small details would not be distingushed immediately, but would trigger a specialized routine (Ullman 1984) to distinguish between them.

Second, in aligning pictorial descriptions a match may be obtained at different levels, such as the underlying object contours, or the level of more abstract descriptors. The contributions of the different levels will have to be combined in an appropriate manner.

Finally, the decision regarding the best matching model will be affected by factors other than similarity of shape. The degree of match may have to take into account, for instance, the amount of distortion that was required to bring the viewed object and model into registration. As discussed in Section 1, the selection of the appropirate model may also be biased, for example, by prior expectation and by proximity to other objects in the scene.

APPENDIX 1
Three-Point Alignment: Uniqueness

We will consider a flat object-model F, which is simply a collection of points in the image plane $z = 0$. F then undergoes a transformation T in space, composed of a translation D, rotation R, and a scaling by a factor S $(S > 0)$. The new image of F is its orthographic projection F' on the image plane. The transformation T also induces a transformation T' in the image plane $T' : F \rightarrow F'$ that matches each point in F with its new location in the image. We will call T' the "image transformation" of F.

The alignment proposition:

Given the coordinates of three non-colinear points in the model and in the image, the image transformation is uniquely determined.

The proposition means that if we can identify three corresponding (non-colinear) points in the model and in the image of a flat object, then we can predict how the entire object will appear in the image following the transformation.

The transformation in space T is also determined uniquely, except for the residual ambiguities that are unavoidable in orthographic projection. That is, S is determined uniquely, D is determined up to translation in depth, and R is determined up to a possible reflection about the image plane.

Proof:

In an orthographic projection, the translation in (x, y) can be determined immediately from the image translation of one of the points. We can therefore assume without loss of generality that the object is fixed at one point, and undergoes a transformation T composed of a scaling S and a rotation R around that point. Let the coordinates of the three points before the transformation be $(0, P_1, P_2)$, and following the transformation $(0, Q_1, Q_2)$. Without loss of generality (P_1, P_2) are assumed to be in the image plane $z = 0$. The coordinate (Q_1, Q_2) in space are not known, we only observe the projection of (Q_1, Q_2) on the image plane.

We will assume that the transformation is not unique. That is, there is another transformation $\bar{T}$, composed of a scaling $\bar{S}$ and rotation $\bar{R}$, that transforms $(0, P_1, P_2)$ into $(0, \bar{Q}_1, \bar{Q}_2)$ in such a manner that the projection of $(0, Q_1, Q_2)$ and $(0, \bar{Q}_1, \bar{Q}_2)$ coincide. We wish to show that, except for the unavoidable reflection ambiguity, $T = \bar{T}$.

The transformation T can be represented by the matrix SR (the rotation matrix R multiplied at every coordinate by the scalar S), and $\bar{T}$ by $\bar{S}\bar{R}$. Let B represent the difference matrix $B = SR - \bar{S}\bar{R}$.

BP_1 has the form $(0,0.z_1)$ since SRP_1 and $\bar{S}\bar{R}P_1$ coincide in their x and y components. Similarly, $BP_2 = (0,0,z_2)$, and z_1, z_2, are not both 0. Assume $z_1 \neq 0$, and define a new point a: $a = \frac{z_2}{z_1}P_1 - P_2$. Since P_1, P_2, are non-colinear, $a \neq 0$, but $Ba = 0$ (where 0 here is the zero vector). It follows that $SRa = \bar{S}\bar{R}a$, and since $\|Ra\| = \|\bar{R}a\| = \|a\|$, it follows that $S = \bar{S}$.

The scale factor is therefore uniquely determined. To examine the rotation, let U denote $R - \bar{R}$. U maps P_1, P_2, and therefore the entire x, y plane onto the z axis. In particular, U maps $(0,0,1)$ and $(0,1,0)$ onto the z axis. This implies that U has the form:

$$U = \begin{pmatrix} 0 & 0 & u_{13} \\ 0 & 0 & u_{23} \\ u_{31} & u_{32} & u_{33} \end{pmatrix}$$

This implies that in the rotation matrices $R, \bar{R}$, $r_{11} = \bar{r}_{11}, r_{12} = \bar{r}_{12}, r_{21} = \bar{r}_{21}, r_{22} = \bar{r}_{22}$. Since in a rotation matrix $r_{33} = r_{11}r_{22} - r_{12}r_{21}$, it follows that $r_{33} = \bar{r}_{33}$. From this it follows that either $R = \bar{R}$, or that $\bar{R}$ has the form:

$$\bar{R} = \begin{pmatrix} r_{11} & r_{12} & -r_{13} \\ r_{21} & r_{22} & -r_{23} \\ -r_{31} & -r_{32} & r_{33} \end{pmatrix}$$

In this latter case $\bar{T}F$ is the mirror reflection of TF about the image plane (but the projection of TF and $\bar{T}F$ on the image plane coincide).

APPENDIX 2
Three-Point Alignment: Computation

The uniqueness proof in Appendix 1 is not a constructive one. In this appendix two methods are given for actually performing the alignment based on three corresponding points in the model and the object. The first method recovers the transformation in space (translation, rotation, and scale change) that brings the model into alignment with the viewed object. The second method specifies a sequence of simple image transformations (rotation, stretch, shear) that accomplish the same task.

The problem to solve is the following. We are given the 3-D coordinates of three non-colinear points. Without loss of generality we can assume that the three points lie initially in the plane $z = 0$ (the "image plane"). We are next given the image of the same points following a transformation T. This image is an orthographic projection on the image plane, and the transformation is composed of an unknown rotation R and scaling S. The objective is to determine R and S (subject to the limitation of R discussed in Appendix

1). As discussed above, the translation component of the transformation can be ignored, and we assume that the object is fixed at one point. It is also assumed that a correspondence between the points has been established.

Let the coordinates of the three model points be $(0,0,0), (x_1,0,0), (x_2, y_2, 0)$. (This means that the first point was chosen as the origin, and the second point was defined as lying on the x axis.) Let the position of the same points following the transformation (the "object points") be $(0,0,0), (\bar{x}_1, 0, \bar{z}_1), (\bar{x}_2, \bar{y}_2, \bar{z}_2)$. This means that the image of the object has been rotated so that the second point lies on the x axis. The z coordinates $\bar{z}_1, \bar{z}_2$ are, of course, unknown, as is the transformation relating the two sets of points. Let us assume first that no scaling has been involved. Since the necessary rotation about the z axis has already been performed, the model can be brought into alignment with the object points using two successive rotations: a rotation around the x axis by an angle θ, followed by a rotation about the y axis by an angle ϕ. The full rotation matrix composed of these two rotations is:

$$R = \begin{pmatrix} cos\phi & -sin\phi sin\theta & \sin\phi cos\theta \\ 0 & cos\theta & sin\theta \\ -sin\phi & -cos\phi sin\theta & cos\phi cos\theta \end{pmatrix}$$

If scaling is allowed as well then the relations between model and object points should satisfy:

$$\begin{aligned} S \cdot R(x_1, 0, 0) &= (\bar{x}_1, 0, \bar{z}_1) \qquad (1) \\ S \cdot R(x_2, y_2, 0) &= (\bar{x}_2, \bar{y}_2, \bar{z}_2) \end{aligned}$$

These equations simply relate the positions of the points before and after the transformation. Expanding (1) explicitly using the matrix R yields:

$$\begin{aligned} (i) \quad SX_1 cos\phi &= \bar{x}_1 \qquad (2) \\ (ii) \quad Sy_2 cos\theta &= \bar{y}_2 \\ (iii) \quad Sx_2 cos\phi - Sy_2 sin\phi sin\theta &= \bar{x}_2 \end{aligned}$$

From (i) and (ii) we can obtain expressions for $sin\phi$ and $sin\theta$ respectively, and substitute in (iii). This yields a quadratic equation in S^2 of the form:

$$AS^4 + BS^2 + C = 0 \qquad (3)$$

The coefficients A, B, C are all expressed in terms of observable quantities:

$$\begin{aligned} A &= x_1^2 y_2^2 \qquad (4) \\ B &= -(x_1^2 \bar{y}_2^2 + \bar{x}_1^2 y_2^2 + (x_2 \bar{x}_1 - \bar{x}_2)^2) \end{aligned}$$

$$C = \bar{x}_1^2 \bar{y}_2^2$$

There will be at most two solution for S^2 and since $S > 0$ at most two solutions for S itself. However, because of the uniqueness result for S, only one of the two will in fact solve the three equations in (2) above. Equations (i) and (ii) will determine $cos\phi$ and $cos\theta$ respectively. A freedom in the choice of the sign for $sin\phi$ will yield the two solutions for the rotation (R and $\tilde{R}$ in Appendix 1).

Alignment by image transformations

We have obtained above formulas for solving for the transformation parameters in space. When the model is entirely flat, it is possible to align the model with the viewed object using a sequence of simple image transformations: image rotation, stretch, and shear. In the terminology of Appendix 1, this method recovers the image transformation T' rather than the transformation T itself.

Assume first that the transformation is composed of rotation only. As before, the rotation of the model is broken down into a rotation in the image plane (about the z axis), followed by rotations about the x and y axes. The rotation about the x axis simply induces a stretch in the y direction ("y-scaling") by a factor $S_y = (\frac{\bar{y}_2}{y_2})$. The subsequent rotation about the y axis induces a transformation that can be expressed as x-scaling by a factor $S_x = (\frac{\bar{x}_1}{x_1})$, followed by a shear transformation of the form: $x \rightarrow x + \Delta y$. The value of Δ is given by: $\Delta = \frac{x_1\bar{x}_2 - x_2\bar{x}_1}{x_1\bar{y}_2}$

If scaling is added to the rotation, its effect can be subsumed by the y-scaling and x-scaling stages, and exactly the same sequence of image transformations would align the model with the viewed object. Given three model points and the three corresponding image points it is possible to apply in this manner a sequence of image transformations to the model to align it with the viewed object. In summary, the model can be aligned with the viewed object using the following sequence of operations: rotation in the image plane, y-scaling, x-scaling, and shear.

Unlike the first method, the image alignment will match any three model points with any three image points, even if the image set is not a possible projection of the model. At least one additional point will therefore be required to reject a model. Another difference between the two methods is that image alignment is applicable to planar models only. The first method is applicable to non-planar models as well. Based on three points the transformation (in space) is determined, and this transformation can then be applied to any 3-D model to determine its new position in space.

APPENDIX 3

Orientation Alignment

Appendix 1 and 2 have shown how a viewed object can be aligned with a potential model using three corresponding points. This is only one example of an alignment procedure; alternative procedures are also possible. In particular, if the viewed object has a prominent orientation, this orientation can be used for alignment. For a planar region, the orientation together with small pieces of the region's bounding contour are sufficient for alignment, without using any identifiable points in the object and model. This procedure also assumes that the occlusion is not too severe, as specified below.

The use of orientation means that it is possible to identify an orientation **u** in the image, which, following the alignment, should be parallel to a known direction **v** in the model. This information is more restricted than the use of an axis: an axis is a line whose position as well as orientation are known. Oriented texture on the object, for example, may specify an orientation in the image without specifying an axis location.

Given the orientation **u** in the image, the first step in the alignment is to rotate the model (or the image) until the direction of **u** is parallel to the desired direction **v**. The alignment can be completed by applying to the model the following sequence of operations:

x-scaling, i.e. $(x, y) \rightarrow (\gamma x, y)$

y-scaling, i.e. $(x, y) \rightarrow (x, \beta y)$

y-shear, i.e. $(x, y) \rightarrow (x, y + \alpha x)$

translation, i.e. $(x, y) \rightarrow (x + \Delta x, y + \Delta y)$

The amount of x-scaling can be determined directly. Scaling the model by γ in the x-direction should make the overall width of the model and the viewed object identical. (This assumes limited occlusion: the extrema of the viewed object in the x direction should be in view. If they are not in view, internal contours can be used instead.) The translation in the x direction is also immediately recoverable following this step.

The remaining parameters $(\alpha, \beta, \Delta y)$ can now be recovered from any three points along the object boundary. For each point on the viewed object's boundary, the corresponding point in the model is already known: it is a point with the same x-coordinate (since all of the transformations involving the x dimension have already been performed). Three boundary points will supply three simple linear equations in $(\alpha, \beta, \Delta y)$. The process can be further simplified by the appropriate selection of points. For example, from two points with the same x but different y coordinates the value of β can be recovered directly.

This orientation alignment uses no identifiable "anchor" points that are used in the-three-point alignment. It is also possible to use various intermediate approaches. For example, if an axis, rather than a dominant orientation, can be identified in the image, the alignment process can be facilitated, and can become more tolerant to occlusions.

Acknowledgements: Figure 1 by J. Schick is reproduced from "All Around the House" 1985, by W.H. Hooks, B.D. Boegehold, B. Brenner, J. Oppenheim, S.V. Reit, & J. Schick with the kind permisssion of Barron's Educational Series, Inc. I thank E. Hildreth, E. Grimson, D. Huttenlocher, J. Mahoney and W. Richards for valuable discussions and comments, and J. and T. Ullman for insightful comments on Fig. 1.

References

Alt, F.Z. 1962. Digital pattern recognition by moments. In: G.L. Fischer, D.K. Pollock, B. Raddack, & M.E. Stevens, (eds.), *Optical Character Recognition.* Washington: McGregor & Werner Inc.

Asada, H. & Brady, M. 1985. The curvature primal sketch. *IEEE PAMI* 8(1), 2-14.

Baird, H. S. 1984. *Model-Based Image Matching Using Locations.* Cambridge: MIT Press.

Barlow, H.H., 1972. Single units and sensation: A neuron doctrine for perceptual psychology? *Perception, I.*, 371-394.

Besl, P.J. & Jain, R. C. 1985. Three-dimensional object recognition. *Computing surveys*, 17(1), 75-145.

Biederman, I. 1985. Human image understanding: Recent research and a theory. *Comp. Vis. Graph. Im. Proc.*, 32, 29-73.

Binford, T.O. 1971. Visual perception by computer. Presented to the IEEE Conference on Systems and Control, Miami, De. 1971.

Binford, T.O. 1982. Survey of model-based image analysis systems. *Int. J. Robotics Research*, 1(1) 18-64.

Bolles, R.C. and Cain, R.A. 1982. Recognizing and locating partially visible objects: The local-feature-focus method. *Int. J. Robotics Research*, 1(3), 57-82.

Brady M., Ponce J., Yuille, A., & Asada, H. 1985. Describing surfaces. *A.I. Memo 882, The Artificial Intelligence Lab., M.I.T.*.

Brooks, R. 1981. Symbolic reasoning among 3-dimensional models and 2-dimensional images. *Artificial Intelligence,* 17, 285-349.

Connell, J.H. 1985. Learning shape descriptions: Generating and generalizing models of visual objects. *MIT Art. Int. Technical Report 853.*

Cutting, J.E. and Kozlowski, L.T., 1977. Recognizing Friends by Their Walk: Gait Perception Without Familiarity Cues, *Bull. Psychonometric Soc.*, Vol. 9, No. 5, 353-356.

Dane, C. & Bajcsy, R. 1982. An object-centered three-dimensional model builder. *Proc. 6th Int. Conf. Pat. Recog. Munich, West Germany, Oct 19-22 1982* 348-350.

Fu, K.S. 1974. *Syntactic Methods in Pattern Recognition* N.Y.: Academic Press.

Faugeras, O.D. 1984. New steps towards a flexible 3-D vision system for robotics. *Proc. 7th Int. Conf. Pat. Recog., Montreal, Canada, July 30-Aug 2, 1984* 796-805.

Gibson, J.J. 1950. *The perception of the visual world.* Boston: Houghton Mifflin.

Gibson, J.J. 1979. *The ecological approach to visual perception.* Boston: Houghton Mifflin.

Grimson, W.E.L. & Lozano-Perez, T. 1984. Model-based recognition and localization from sparse data. *Int. J. Robotics. Research.* 3(3), 3-35.

Hernstein, R.J. 1984. Objects, categories, and discriminative stimuli. In H.L. Roitblat, T.G. Bever, & H.S. Terrace (eds.), *Animal Cognition*, Hillsdale N.J.: Lawrence Erlbaum Assoc.

Hoffman, D. 1983. The interpretation of visual illusions. *Scien. Am.* 249 (6), 154-162.

Hoffman, D. & Richards, W. 1986. Parts of Recognition. In: A.P. Pentland (ed.), *From Pixels to Predicates*, Norwood N.J.: Ablex Publishing Corp.

Hopfield, J.J. 1982. Neural networks and physical systems with emergent collective computational abilities. *Proc. Nat. Acad. Sci. USA 79,* 2554-2558.

Hubel, D.G., & Wiesel, T.N., 1962. Receptive fields, binocular interaction, and functional architecture in the cat's visual cortex. *Journal of Physiology (London),* 160, 106-154.

Hubel, D.H., & Wiesel, T.N., 1968. Receptive fields and functional architecture of monkey striate cortex. *Journal of Physiology (London)*, 195, 215-243.

Huberman, B.A. & Hogg, T. 1984. Adaptation and self-repair in parallel computing structures. *Phys. Rev. Let.* 52 (12) 1048-1051.

Huttenlocher, D.P. & Ullman, S. 1987. Object recognition using alignment. A.I. Memo in preparation.

Johanson, G. 1973. Visual Perception of Biological Motion and a Model for Its Analysis, *Perception and Psychophysics*, Vol. 14, No. 2, 201-211.

Kohonen, T. 1978. *Associative Memory: A System Theoretic Approach.* Berlin: Springer Verlag.

Lowe, D.G. 1985. *Perceptual Organization and Visual Recognition.* Boston: Kluwer Academic Publishers.

Lowe, D. G. 1986. Three-dimensional object recognition from single two-dimensional images. *Robotics Research Technical Report 202, Courant Institute of Math. Sciences, N.Y. University.*

Mahoney, J.V. 1986. Image chunking: defining spatial building blocks for scene analysis. S.M. Thesis, Dept. of Electrical Engineering and Computer Science, M.I.T.

Marr, D. and Nishihara, H.K. 1978. Representation and recognition of the spatial organization of three-dimensional shapes. *Proc. Roy. Soc. B., 200,* 269-291.

Milner, P.M. 1974. A model for visual shape recognition. *Psychol. Rev. 81(6),* 521-535.

Minsky, M.and Papert, S. 1969. *Perceptrons.* Cambridge, MA and London: The M.I.T. Press.

Neisser, U. 1966. *Cognitive Psychology.* N.Y.: Appelton-Century-Crofts

Perrett, D.I., Smith, P.A.J., Potter, D.D., Mistlin, A.J., Head, A.S., Milner, A.D. & Jeeves, M.A. 1985. Visual cells in the temporal cortex sensitive to face view and gaze direction. *Proc. Roy. Soc. B,* 223, 293-317.

Pinker, S. 1984. Visual cognition: an introduction. *Cognition,* 18, 1-63.

Potmesil, M. 1983. Generating models of solid objects by matching 3D surface segments. *Proc. 8th Int. Joint Conf. Art. Intell. (Karlsruhe, West Germany, Aug. 8-12),* 1089-1093.

Potter, M.C. 1975. Meaning in visual search. *Science* 187, 965-966.

Preparata, F.P & Shamos, M.I. 1985. *Computational Geometry* N.Y.: Springer-Verlag.

Rock, I. 1973. *Orientation and Form* N.Y.: Academic Press.

Selfridge, O.G. 1959. Pandemonium: A paradigm for learning. In: *The Mechanisation of Thought Processes,* London: H.M. Stationary Office.

Shepard, R.N. & Cooper, L.A. 1982. *Mental Images and Their Transformations.* Cambridge, MA: MIT Press/Bradford Books.

Shepard, R.N. & Metzler, J. 1971. Mental rotation of three-dimensional objects. *Science,* 171, 701-703.

Sutherland, N.S. 1959. Stimulus analyzing mechanisms. In: *The Mechanisation of Thought Processes,* London: H.M. Stationary Office.

Tou, J.T. & Gonzalez, R.C. 1974. *Pattern Recognition Principles.* Reading, MA: Addison-Wesley.

Treisman, A. & Gelade, G. 1980. A feature integration theory of attention. *Cog. Psychol.* 12, 97-136.

Ullman, S. 1984. Visual routines. *Cognition,* 18, 97-159.

Witkin, A.P. & Tenenbaum, J.M. 1983. On the role of structure in vision. In: *Human and Machine Vision* , J. Beck, B. Hope & A. Rosenfeld (eds.), N.Y.: Academic Press, 481-543.

END

10-87

DTIC