Boston University Computer Science Technical Report 2003-007, April 10, 2003. Submitted to IEEE PAMI.

The Specialized Mappings Architecture

Rómer Rosales
Probabilistic and Statistical Inference Group
Dept. of Electrical and Computer Engineering
University of Toronto
Toronto, Ontario CANADA
romer@psi.toronto.edu

Stan Sclaroff
Image and Video Computing Group
Dept. of Computer Science
Boston University
Boston, MA 02215 USA
sclaroff@cs.bu.edu

Abstract

A probabilistic, nonlinear supervised learning model is proposed: the Specialized Mappings Architecture (SMA). The SMA employs a set of several forward mapping functions that are estimated automatically from training data. Each specialized function maps certain domains of the input space (e.g., image features) onto the output space (e.g., articulated body parameters). The SMA can model ambiguous, one-to-many mappings that may yield multiple valid output hypotheses. Once learned, the mapping functions generate a set of output hypotheses for a given input via a statistical inference procedure. The SMA inference procedure incorporates an inverse mapping or feedback function in evaluating the likelihood of each of the hypothesis. Possible feedback functions include computer graphics rendering routines that can generate images for given hypotheses. The SMA employs a variant of the Expectation-Maximization algorithm for simultaneous learning of the specialized domains along with the mapping functions, and approximate strategies for inference. The framework is demonstrated in a computer vision system that can estimate the articulated pose parameters of a human's body or hands, given silhouettes from a single image. The accuracy and stability of the SMA are also tested using synthetic images of human bodies and hands, where ground truth is known.

Keywords: Supervised learning, statistical inference, mixture models, Expectation Maximization algorithm, articulated structure estimation, human body pose, hand shape.

Report Documentation Page

Form Approved
OMB No. 0704-0188

Public reporting burden for the collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington VA 22202-4302. Respondents should be aware that notwithstanding any other provision of law, no person shall be subject to a penalty for failing to comply with a collection of information if it does not display a currently valid OMB control number.

1. REPORT DATE
10 APR 2003

2. REPORT TYPE

3. DATES COVERED
00-04-2003 to 00-04-2003

4. TITLE AND SUBTITLE
The Specialized Mappings Architecture

5a. CONTRACT NUMBER

5b. GRANT NUMBER

5c. PROGRAM ELEMENT NUMBER

6. AUTHOR(S)

5d. PROJECT NUMBER

5e. TASK NUMBER

5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Boston University ,Computer Science Department,111 Cummington Street,Boston,MA,02215

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)

10. SPONSOR/MONITOR'S ACRONYM(S)

11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release; distribution unlimited

13. SUPPLEMENTARY NOTES
The original document contains color images.

14. ABSTRACT

15. SUBJECT TERMS

16. SECURITY CLASSIFICATION OF:

a. REPORT	b. ABSTRACT	c. THIS PAGE	17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
unclassified	**unclassified**	**unclassified**		**35**	

1 Introduction

A fundamental task for vision systems is to infer the state of the world given some form of visual observations. From a computational perspective, this often involves facing an ill-posed problem: information is lost via projection of the three-dimensional world into a two-dimensional image. As a result, it is often the case that multiple valid interpretations of an image are possible. Solving an ill-posed problem requires additional information, usually provided as a model of the underlying process. In their day to day life, humans are surprisingly adept at interpreting the visual world, despite the ill-posed nature of the problem. For example, humans can easily estimate the articulated pose and motion of people in a scene, given only relatively low-resolution, monocular images of the world, e.g., from a photograph or a video. It is believed that humans employ extensive prior knowledge about human body structure and motion in this task [22]. Assuming this, in this paper we will consider how a computer might learn the underlying *knowledge* in the form of a probabilistic model, and thereby infer pose from a single image.

Let us consider an example pose inference task: given only a person's silhouette, estimate that person's articulated body pose. To be concrete, let us define articulated pose in terms of: (a) the 2D locations of the person's joints in the image, or (b) the 3D locations of the person's joints in Euclidean space. Imagine drawing marks on the silhouette image that approximately label the joints: left elbow, right elbow, left knee, right knee, and so on. Also consider a plausible 3D pose interpretation for this silhouette. While this inference task seems relatively simple for a human to perform, the task is quite challenging, using either representation (a) or (b), for current computer vision systems.

For purposes of computation, the above inference task can be defined as follows: given an observation vector of cues $\mathbf{x} \in \Re^c$ that were extracted from an image of a person, infer the articulated pose parameter vector $\mathbf{h} \in \Re^t$. Assume these input and output spaces $\Re^c$ and $\Re^t$ are continuous. In a generic machine learning framework, inference might be achieved via a mapping function $\phi : \Re^c \rightarrow \Re^t$ that for a given input (cues) computes the correct output (a single pose, or more generally a pose probability distribution). While the apparent simplicity of this strategy is alluring, it leaves a number of nettlesome open issues: how to select the appropriate functional form for this mapping, how to estimate (learn) this function from data, and how to perform inference.

The functional form required for this mapping ϕ may not be simple, because the mapping from cues to articulated poses is generally ambiguous (one-to-many). In fact no single function can perform this mapping. An example is illustrated in Fig. 1 (R_1 and R_2). The arm locations cannot be uniquely inferred given the silhouette $\mathbf{x}$; therefore, $\mathbf{a}$–$\mathbf{h}$ are all possible pose hypotheses. Note also that pose $\mathbf{c}$ is the reflection of $\mathbf{a}$: the

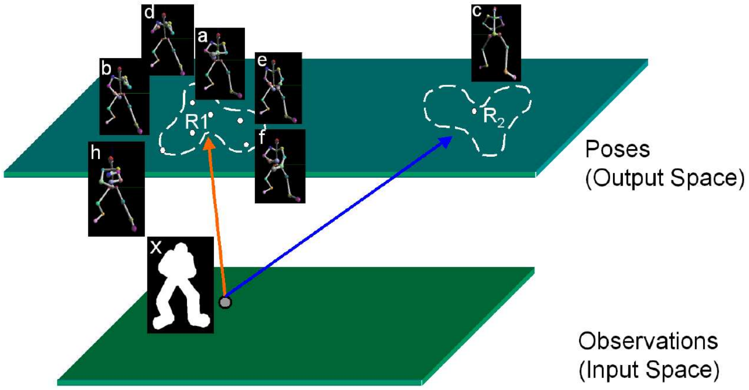

Figure 1: Example ambiguity in mapping body silhouette cues in $\Re^c$ to articulated body poses in $\Re^t$. Given silhouette $\mathbf{x}$, poses $\mathbf{a}$–$\mathbf{h}$ are all valid hypotheses. In general, entire regions in $\Re^t$ may contain valid poses.

camera looks at the back rather than at the front of the body. In practice, there may be entire regions in $\Re^t$ that contain valid poses associated with the silhouette, as shown in Fig. 1. Thus, there might be an infinite number of valid poses for a particular input. Moreover, the regions of valid poses need not be connected. For instance, different regions in $\Re^t$ may correspond to ranges of valid poses, *e.g.*, some viewed from the front and others from behind. Such ambiguities are not particular to human body pose; for instance, analogous inference problems exist in estimating hand pose from image features, as will be seen later.

Let us now consider the inverse problem: given an articulated pose vector $\mathbf{a}$, generate its silhouette cues $\mathbf{x}$. With a good computer graphics model of the human body, one can easily render the silhouette $\mathbf{x}$. Thus, we can easily compute the inverse mapping $\zeta : \Re^t \rightarrow \Re^c$. Many real world problems share the property that their inverse problem is simpler, e.g., speech recognition. In fact, this property is a key part of our problem definition and it will play an important role in developing the framework presented in this paper.

We now have a notion of the input and output spaces, the forward and inverse relationships associated with them, and a few basic difficulties that can arise in the context of our example application. The mapping of inputs (cues) to outputs (poses) is ambiguous and one-to-many; this precludes the use of supervised learning methods that fit a single function to the data, e.g., most neural networks, support vector machines, least squares estimation, boosting, etc. On the other hand, we have access to the *inverse* map $\zeta : \Re^t \rightarrow \Re^c$, which we can exploit in formulating a solution to the learning problem.

In this paper, we describe a probabilistic, nonlinear supervised learning framework: the Specialized Mappings Architecture (SMA). The SMA employs a set of M mapping functions $\phi_k : \Re^c \rightarrow \Re^t$, where each specialized function maps certain sub-domains of the input space (cues) onto the output space (poses). The sub-domains of ϕ_k need not be connected regions in the input or output spaces. The SMA mapping functions are estimated automatically from training data, via a supervised learning procedure. A variant of

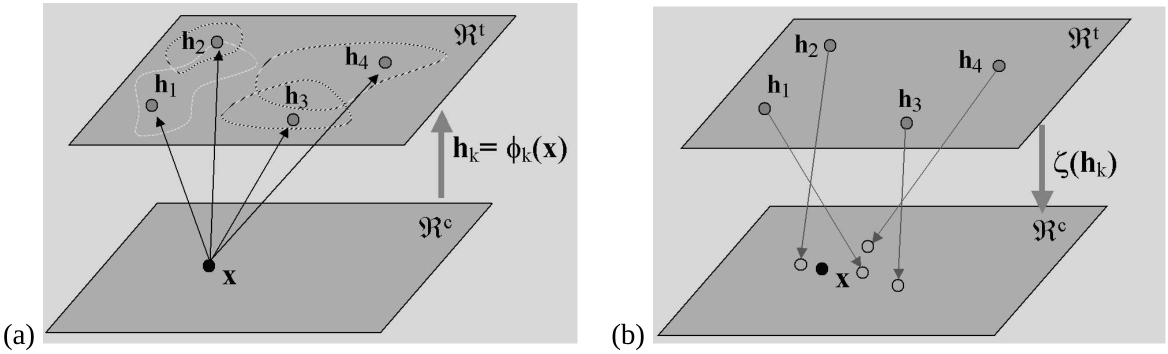

Figure 2: Inference in the specialized mappings architecture. (a) Given an input vector $\mathbf{x}$, the mapping functions ϕ_k generate a set of hypotheses. (b) The inverse mapping function ζ is employed in evaluating the likelihood of each hypothesis.

the Expectation-Maximization algorithm is used for simultaneous learning of the specialized domains along with the mapping functions. Once the SMA model is learned, approximation strategies, based on sampling, make the SMA's inference tractable and fast. The basic concepts of SMA inference are illustrated in Fig. 1. For a given input $\mathbf{x}$, the mapping functions generate a set of output hypotheses. The SMA inference procedure then exploits an inverse mapping ζ in evaluating the likelihood of each hypothesis.

An important advantage of the SMA is that it can model ambiguous, one-to-many mappings that may yield multiple valid output hypotheses. Unlike other learning approaches that employ a set of mapping functions (*e.g.*, [12, 16, 24]), the SMA incorporates an inverse mapping ζ in probabilistic inference. The framework is evaluated in a computer vision system that can estimate the articulated pose parameters of a human body or human hands, given real image silhouettes. The accuracy and stability of the SMA are also tested using synthetic images of human bodies and hands, where ground truth is known.

2 Related Work

In computer vision, recovery of articulated body pose from images is often formulated as a *tracking* problem. Usually, link-joint models comprised of 2D or 3D geometric primitives are designed beforehand to roughly match the specific morphology of the target in question [7, 10, 13, 27, 31, 36, 38]. Mesh models have also been used as an alternative to link-joint models [15]. At each frame, these models are fitted to the image to minimize some cost function that favors the overlap of the model and associated image regions (or motion). Despite their descriptive power, this family of approaches has a number of critical drawbacks. Generally, a non-linear optimization problem must be solved at every frame. Careful manual placement of the model on the first frame in a video sequence is also required. Moreover, tracking in subsequent frames tends to be sensitive to errors in initialization and numerical drift; as a result, these systems cannot recover from tracking errors in the middle of a sequence.

To address these weaknesses, specialized dynamical models have been proposed [20, 27, 29]. These methods learn a prior distribution over some specific motion class, such as walking. This prior is used to predict and hopefully improve the pose estimates in future frames. However, this strong prior substantially limits the generality of the motions that can be tracked; a prior for a given class of motions is generally useless when used for tracking objects undergoing a different class of motion, e.g., walking vs. dancing.

Other methods for constrained tracking include [4, 5], where a subspace of allowable motions is learned from a set of examples. These examples and the model (usually linear) are hoped to be sufficient to span the set of possible motions to be seen during tracking. Thus, pose inference involves finding a linear projection of the observed data onto the motion subspace. This subspace approach enforces a strong prior; as mentioned previously, this limits the generalization of the model to classes of motions not seen in the training set. Furthermore, articulated motion is generally non-linear, and cannot be easily explained as a linear projection.

In our approach we avoid matching image features (e.g., image regions, points, or articulated models) from frame to frame. Therefore, we do not refer to our approach as *tracking*, per se. This is in direct contrast with the techniques mentioned above. A number of other approaches also depart from the aforementioned tracking paradigm. We summarize these next.

In [18] a statistical approach is employed in reconstructing the 3D motions of a human figure. The approach employs a Gaussian probability model for short human motion sequences. It is assumed that 2D tracking of the joint positions in the image is given; therefore, this assumption implicitly incurs the restrictions found in all tracking approaches.

In [39] dynamic programming is used to calculate the best global matching of image points to predefined body joints, given a learned probability density function of the position and velocity of body features. Although not explicitly mentioned by the authors, the probability function is defined by a triangulated acyclic graph. Thus, inference is feasible due to the running intersection property [23, 30]. Still, in this approach, the image points and model initialization must be provided by hand or through some other method.

In [6], the manifold of human body dynamics is modelled via a hidden Markov model with an entropic prior. Once the states are inferred from observations, a quadratic cost function is used to generate a continuous path in configuration space, *i.e.,* body pose space.

In all of the non-tracking approaches mentioned [6, 18, 39] models of *motion* were estimated from data. Although the approach presented in this paper can be used to model dynamics, we argue that when general human motion dynamics are to be learned, the amount of training data, model complexity, and

number of training examples	N
training set	$\mathcal{Z} = \{\mathbf{z}_1, ..., \mathbf{z}_N\}$
training example (input,output) pair	$\mathbf{z}_i = (\upsilon_i, \psi_i)$
input (feature) training vector	$\upsilon_i \in \Re^c$
output (pose) training vector	$\psi_i \in \Re^t$
feedback (rendering) function	$\zeta : \Re^t \rightarrow \Re^c$
number of mapping functions	M
k-th input-output mapping function	$\phi_k : \Re^c \rightarrow \Re^t$
mapping function parameter vector	θ_k
output (pose) hypothesis	$\mathbf{h} = \phi_k(\mathbf{x}, \theta_k), \mathbf{h} \in \Re^t, \mathbf{x} \in \Re^c$
most likely output hypothesis	$\mathbf{h}^*$
SMA mapping functions	$\Phi = \{\phi_1, \ldots, \phi_M\}$
discrete set of labels for mapping functions	$\mathcal{C} = \{1, \ldots, M\}$
hidden random variables assigning mapping functions to training samples	$\mathbf{y} = (y_1, \ldots, y_N), y_i \in \mathcal{C}$
prior probability that mapping function ϕ_k will be used	$\lambda_k = P(y = k)$
prior probability on mapping functions	$\lambda = (\lambda_1, \ldots, \lambda_M)$
SMA parameters (to be learned)	$\theta = (\theta_1, \ldots, \theta_M, \lambda)$
responsibility of k-th mapping function for $\mathbf{z}_i$	$\tilde{P}(y_i = k)$

Table 1: Some mathematical symbols used in the SMA formulation.

computational resources required are impractical. As a consequence, models with unacceptably large priors towards specific motions are generated. Although by not modelling the dynamics we may be ignoring information that could be used to further constrain the inference process, there are some benefits. For instance, a model for inferring body pose that does not consider dynamics provides invariance with respect to speed (*i.e.,* sampling differences) and direction in which motions are performed. This happens simply because this model treats configurations as temporally independent of each other. Other approaches that use a single image include [3, 14, 25, 28, 40]; however, most of these methods also require that projected joint locations be given as input. In our approach this is not necessary.

Our approach maps visual features to likely body configurations. Following a machine learning paradigm, stochastic functions that map visual features to pose parameters are approximated from training data. A unique aspect of our approach is the combined use of (1) these mapping functions with (2) the inverse mapping function ζ. After multiple poses have been inferred from just the visual cues, ζ transforms these pose configurations back to the visual cue (observation) space. In this space, we can then automatically choose among a set of reconstruction hypotheses. This is a fully probabilistic inference process. Our approach avoids the need for manual initialization or tracking; it thereby avoids the consequent disadvantages of tracking. Remarkably, relatively few computations are required for inference. We will now formalize and explain the SMA in detail.

3 Probabilistic Model

In the SMA, a set of mapping functions is estimated from training data, via a supervised learning procedure. Let $\mathcal{Z} = \{\mathbf{z}_1, ..., \mathbf{z}_N\}$ be an observed training set of input-output pairs $\mathbf{z}_i = (\upsilon_i, \psi_i)$. Each $\upsilon_i \in \Re^c$ is an input (feature) vector, and each $\psi_i \in \Re^t$ is its corresponding output (pose) vector. A summary of mathematical symbols used in the SMA formulation is provided in Table 1.

We will approach our forward problem as one of hidden variable density estimation. We begin by introducing the unobserved random variable $\mathbf{y} = (y_1, \ldots, y_N)$. In our model any y_i has as its domain the discrete set $\mathcal{C} = \{1, \ldots, M\}$ of labels for the specialized mapping functions, and can be thought of as the function number used to map the i-th training pair, $\mathbf{z}_i$. Thus M is the number of specialized mapping functions. Our model uses parameters $\theta = (\theta_1, \ldots, \theta_M, \lambda)$, where θ_k represents the parameters of the k-th mapping function, and $\lambda = (\lambda_1, \ldots, \lambda_M)$, where λ_k represents $P(y = k)$, the prior probability that the mapping function with label k will be used to map an input-output pair.

Taking a maximum-likelihood viewpoint, we are interested in finding the optimal parameter settings for our model; thus, we seek to maximize the joint log-probability:

$$\theta^* = \arg\max_\theta \log p(\mathcal{Z}|\theta). \tag{1}$$

Assuming independence of observations given θ, and using Bayes' rule we obtain:

$$\begin{aligned}
\theta^* &= \arg\max_\theta \sum_i \log p(\mathbf{z}_i|\theta) & (2)\\
&= \arg\max_\theta \sum_i \log \sum_k p(\mathbf{z}_i|y_i = k, \theta) P(y_i = k|\theta) & (3)\\
&= \arg\max_\theta \sum_i \log \sum_k p(\psi_i|\upsilon_i, y_i = k, \theta) P(y_i = k|\theta) p(\upsilon_i), & (4)
\end{aligned}$$

where we used the independence assumption $p(\upsilon|\theta) = p(\upsilon)$. Note that because the inputs, υ_i do not depend on the model parameters we can ignore their distribution when finding the optimal parameter settings.

Due to the sum of terms inside the logarithm of Eq. 4, this optimization is generally intractable. However, a variety of practical approximate optimization methods exist, for example, methods that are based on alternating minimizations [8]. An Expectation Maximization (EM) [9, 26] method is described in Sec. 4.

3.1 Choice of a Likelihood Function

Note that the above formulation is general. In particular, the form of the probability $p(\psi_i|\upsilon_i, y_i = k, \theta)$ was not specified. A key question in instantiating the specialized mapping architecture is: what form should be used for $p(\psi|\upsilon, y, \theta)$? This is the probability that output ψ was generated by the mapping function y, given the input υ and model parameters θ. In this work we analyze the following possible cases:

1. A Gaussian joint distribution of input-output vectors:

$$p(\upsilon, \psi|y, \theta) = \mathcal{N}((\upsilon, \psi); \mu_y, \Sigma_y), \tag{5}$$

2. A Gaussian distribution, whose mean is the output of the y-th mapping function:

$$p(\psi|\upsilon, y, \theta) = \mathcal{N}(\psi; \phi_y(\upsilon, \theta), \Sigma_y). \tag{6}$$

One way to interpret (2) is that the error in estimating ψ, given we know what mapping function to use, is Gaussian distributed. These are the two forms tested in our experiments; however, the SMA formulation is general, and can accept other forms for the likelihood function.

4 Learning

As explained above, an approximation method must be used in learning the SMA parameters. We will employ an Expectation Maximization (EM) approach. EM provides a general framework for solving the maximum likelihood parameter estimation problem in statistical models with hidden variables, like Eq. 4. Since the EM algorithm is well known [9, 2, 26], we will only provide derivations specific to the SMA.

Note that the unobserved random variables y_i are assumed independent, given $\mathbf{z}_i$. Thus, the E-step reduces to computing the posterior probabilities for each y_i given the model parameters and observed data:

$$\tilde{P}^{(t)}(y_i = k) = \lambda_k p(\psi_i|\upsilon_i, y_i = k, \theta^{(t-1)}) / \sum_{j \in \mathcal{C}} \lambda_j p(\psi_i|\upsilon_i, y_i = j, \theta^{(t-1)}). \tag{7}$$

Stated differently, this step estimates the responsibility of each mapping function, ϕ_k for each data point, $\mathbf{z}_i$.

The M-step consists of finding $\theta^{(t+1)} = \arg\max_\theta E_{\tilde{P}^{(t)}}[\log p(\mathcal{Z}, \mathbf{y}|\theta)]$. In both of our cases we can show that this is equivalent to finding:

$$\theta^{(t+1)} = \arg\max_\theta \sum_i \sum_{k \in \mathcal{C}} \tilde{P}^{(t)}(y_i = k)[\log p(\mathbf{z}_i|y_i = k, \theta) + \log P(y_i = k|\theta)]. \tag{8}$$

It is important to mention that this is valid if $p(\mathbf{z}_i|\theta)$ depends on y_i and not on y_j, for any $j \neq i$. Note that for the distributions discussed above, this is true. We present solutions for the cases described above.

4.1 Case (1)

In this case we have:

$$p(\upsilon,\psi|y,\theta)=\mathcal{N}(\mu_y,\Sigma_y)=\mathcal{N}(\left[\begin{array}{c}\mu_\upsilon\\ \mu_\psi\end{array}\right],\left[\begin{array}{c}\Sigma_{\upsilon\upsilon}\Sigma_{\upsilon\psi}\\ \Sigma_{\upsilon\psi}^\top\Sigma_{\psi\psi}\end{array}\right])_y. \tag{9}$$

In this case, we can show that the SMA parameter learning problem is reduced to a mixture of Gaussian estimation, for which it is straightforward to estimate θ using EM. Moreover, the Bayesian estimate of ψ given an observed υ is also Gaussian:

$$p(\psi|\upsilon,y,\theta)=\mathcal{N}(\mu_\psi+\Sigma_{\upsilon\psi}^\top\Sigma_{\upsilon\upsilon}^{-1}(\upsilon-\mu_\upsilon),\Sigma_{\psi\psi}-\Sigma_{\upsilon\psi}^\top\Sigma_{\upsilon\upsilon}^{-1}\Sigma_{\upsilon\psi})_y. \tag{10}$$

Therefore in case (1), each specialized function ϕ_k is just the mean of the conditional distribution

$$\phi_k(\upsilon,\theta)=(\mu_\psi+\Sigma_{\upsilon\psi}^\top\Sigma_{\upsilon\upsilon}^{-1}(\upsilon-\mu_\upsilon))_{y=k}. \tag{11}$$

The confidence of the estimate is given by the covariance $\Sigma_k=(\Sigma_{\psi\psi}-\Sigma_{\upsilon\psi}^\top\Sigma_{\upsilon\upsilon}^{-1}\Sigma_{\upsilon\psi})_{y=k}$. However, this expression does not depend on the input, a sometimes undesirable consequence of the given model. Thus, each function ϕ_k is linear in the input vector from $\Re^c$.

4.2 Case (2)

In this case we have:

$$\begin{aligned}\frac{\partial E}{\partial\lambda_k} &= \sum_i\tilde{P}^{(t)}(y_i=k)\frac{\partial}{\partial\lambda_k}\log P(y_i=k|\theta) &(12)\\ \frac{\partial E}{\partial\Sigma_k} &= \sum_i\tilde{P}^{(t)}(y_i=k)\frac{\partial}{\partial\Sigma_k}\log p(\psi_i|y_i=k,\upsilon_i,\theta_k) &(13)\\ \frac{\partial E}{\partial\theta_k} &= \sum_i\tilde{P}^{(t)}(y_i=k)[(\frac{\partial}{\partial\theta_k}\phi_k(\upsilon_i,\theta_k))^\top\Sigma_k^{-1}(\psi_i-\phi_k(\upsilon_i,\theta_k))], &(14)\end{aligned}$$

where E is the cost function that we would like to maximize in Eq. 8.

This gives the following update rules for λ_k and Σ_k, where Lagrange multipliers were used to incorporate the constraint that the sum of the λ_k's is 1:

$$\begin{aligned}\lambda_k^{(t+1)} &= \frac{1}{N}\sum_i\tilde{P}^{(t)}(y_i=k) &(15)\\ \Sigma_k^{(t+1)} &= \frac{\sum_i\tilde{P}^{(t)}(y_i=k)(\psi_i-\phi_k(\upsilon_i,\theta_k))(\psi_i-\phi_k(\upsilon_i,\theta_k))^\top}{\sum_i\tilde{P}^{(t)}(y_i=k)} &(16)\end{aligned}$$

To keep the formulation general, we have not yet defined the form of the specialized functions ϕ_k. Whether or not we can find a closed form solution for the update of θ_k depends on the form of ϕ_k. For example if ϕ_k is a non-linear function, we may have to use iterative optimization to find $\theta_k^{(t)}$. If ϕ_k yields a quadratic form, then a closed form update exists.

4.3 Stochastic Learning

The aforementioned optimization equations can be used to find a local minimum given the initial parameter values. In order to improve this process, and avoid some of the local minima that inevitably arise, we use an annealing schedule on the $\tilde{P}^{(t)}$ probabilities during the M-step. In this way, we redefine:

$$\tilde{P}^{(t)}(y_i = j) \leftarrow \frac{e^{\log(\tilde{P}^{(t)}(y_i=j))/T(t)}}{\sum_{k\in\mathcal{C}} e^{\log(\tilde{P}^{(t)}(y_i=k))/T(t)}}. \tag{17}$$

In our experiments, the temperature parameter T decays exponentially. This step not only helps in avoiding local minima, but it also creates two desirable effects. It forces $\tilde{P}^{(t)}(y_i = j)$ to be binary (either 1 or 0) at low temperatures; as a consequence each point will tend to be mapped by only one specialized function at the end of optimization. Moreover, it makes $\tilde{P}^{(t)}(y_i = k)$ ($k = 1, 2, ..., M$) be fairly uniform at high temperatures, making the optimization less dependent on initialization.

5 Inference

Learning yields a set of specialized functions that map the input space to the output space. As a result of the divide and conquer strategy employed in learning, each of the specialized functions maps different parts of the input space with different levels of accuracy. The mapping behavior of each function is described probabilistically. We can now formulate inference in terms of maximum a posteriori (MAP) estimation. In inference, we want to find the most likely output hypothesis $\mathbf{h} \in \Re^t$ for a given observation $\mathbf{x} \in \Re^c$:

$$\mathbf{h}^* = \arg\max_{\mathbf{h}} p(\mathbf{h}|\mathbf{x}) = \arg\max_{\mathbf{h}} \sum_y p(\mathbf{h}|\mathbf{x}, y)P(y). \tag{18}$$

Any further treatment depends on the properties of the probability distributions involved.

In both Cases (1) and (2) considered in previous sections, we can write $p(\mathbf{h}|\mathbf{x}, y) = \mathcal{N}(\mathbf{h}; \phi_y(\mathbf{x}), \Sigma_y)$. Thus, in either case we have a mixture of Gaussians:

$$\mathbf{h}^* = \arg\max_{\mathbf{h}} \sum_y \mathcal{N}(\mathbf{h}; \phi_y(\mathbf{x}), \Sigma_y)P(y). \tag{19}$$

Eq. 19 is the result of using standard (MAP) inference given our learned model. However, we have yet to make use of the inverse (rendering) function $\zeta : \Re^t \rightarrow \Re^c$ in our framework.

5.1 Maximum A Posteriori Estimation Using the Inverse Mapping Function ζ

The mixing factors in Eq. 19, $\lambda_y = P(y)$, do not depend on the input $\mathbf{x}$, which is consistent with our conditional independence assumption $P(y|\mathbf{x}) = P(y)$ in the forward model. This differs from the Mixture of Experts (ME) formulation [21, 24], which does not assume $P(y|\mathbf{x}) = P(y)$; instead, $P(y|\mathbf{x})$ is assumed to take a certain form, embodied by a set of *gating networks*. In [21, 24] the gating networks were modeled by a *logit* linear model, learned from data. In the SMA, an entirely different approach can be used due to the availability of the rendering function ζ, which we call the inverse function or inverse map.

This inverse map ζ can be obtained via computer graphics rendering. For instance, in human pose estimation, ζ could render an articulated, computer graphics model given pose parameters $\mathbf{h}$. If computer graphics rendering is unavailable or too slow, an approximate inverse map $\hat{\zeta}$ may be obtained via supervised learning over a training set of input-output pairs, $\mathcal{Z}$. For example, $\hat{\zeta}$ could employ a multi-layer neural network, support vector machine, etc. Note that the inverse mapping is assumed to be a function, i.e., one-to-one or many-to-one; thus, functional forms for $\hat{\zeta}$ are acceptable.

Given an inverse map ζ, it is possible to derive an expression for the probability of the observed input $\mathbf{x}$, given the output hypothesis $\mathbf{h}$. For instance, we could employ the Gaussian model[1]:

$$p(\mathbf{x}|\mathbf{h}) = \mathcal{N}(\mathbf{x}; \zeta(\mathbf{h}), \Sigma_\zeta), \tag{20}$$

where Σ_ζ is estimated for a given SMA using a training set. While this is one example of a model for $p(\mathbf{x}|\mathbf{h})$ that incorporates knowledge of ζ, indeed others are possible. Once we have a model for $p(\mathbf{x}|\mathbf{h})$, then finding an optimal $\mathbf{h}^*$ given an input $\mathbf{x}$ can be formulated as a continuous optimization problem

$$\begin{aligned}
\mathbf{h}^* &= \arg\max_{\mathbf{h}} p(\mathbf{h}|\mathbf{x}) && (21)\\
&= \arg\max_{\mathbf{h}} \frac{p(\mathbf{x}|\mathbf{h})p(\mathbf{h})}{p(\mathbf{x})} && (22)\\
&= \arg\max_{\mathbf{h}} \frac{p(\mathbf{x}|\mathbf{h}) \int\int p(\mathbf{h}, \mathbf{x}, y) d\mathbf{x} dy}{p(\mathbf{x})} && (23)
\end{aligned}$$

via Bayes' rule, and marginalizing over $\mathbf{x}$ and y.

Since $\mathbf{x}$ is observed, say $\mathbf{x} = \mathbf{x}_o$, $\int p(\mathbf{h}, \mathbf{x}, y) d\mathbf{x} = \delta(\mathbf{x} - \mathbf{x}_o) p(\mathbf{h}, \mathbf{x}, y)$ we can rewrite Eq. 23:

$$\begin{aligned}
\mathbf{h}^* &= \arg\max_{\mathbf{h}} \frac{p(\mathbf{x}|\mathbf{h}) \int\int p(\mathbf{h}, y|\mathbf{x}) p(\mathbf{x}) d\mathbf{x} dy}{p(\mathbf{x})} && (24)\\
&= \arg\max_{\mathbf{h}} \frac{p(\mathbf{x}|\mathbf{h}) p(\mathbf{x}) \int p(\mathbf{h}, y|\mathbf{x}) dy}{p(\mathbf{x})} && (25)\\
&= \arg\max_{\mathbf{h}} p(\mathbf{x}|\mathbf{h}) \sum_y p(\mathbf{h}|\mathbf{x}, y) P(y), && (26)
\end{aligned}$$

where we assume $p(\mathbf{h}, \mathbf{x}, y)$ factorizes into $p(\mathbf{h}|\mathbf{x}, y)p(\mathbf{x})P(y)$, and $P(y|\mathbf{x}) = P(y)$ as before.

Unfortunately, finding the maximum of Eq. 26 is generally infeasible [35]. In the following sections, we describe approximation algorithms for obtaining good estimates of $\mathbf{h}^*$.

[1] Note that the distribution $p(\mathbf{x}|\mathbf{h})$ does not have to be consistent with the forward model $p(\mathbf{h}|\mathbf{x})$, *i.e.,* related by Bayes rule in this case. Indeed the key insight is that they represent different probabilistic models that can be used alternatingly.

5.2 Non-deterministic Approximate Inference: Multiple Samples (MS)

Let us assume that we can approximate $\sum_y p(\mathbf{h}|\mathbf{x}, y)P(y)$ by a set of samples generated according to $p(\mathbf{h}|\mathbf{x}, y)P(y)$ and a kernel function $K(\mathbf{h}, \mathbf{h}_s)$, such that $K(\mathbf{h}, \mathbf{h}_s) \geq 0$ and $\int K(\mathbf{h}, \mathbf{h}_s)d\mathbf{h} = 1$ for any given $\mathbf{h}_s$. Given a set of samples $\mathcal{H}_{Spl} = \{\mathbf{h}_s\}_{s=1...S}$, we can construct the approximation $\sum_y p(\mathbf{h}|\mathbf{x}, y)P(y) \approx \frac{1}{S}\sum_{s=1}^{S} K(\mathbf{h}, \mathbf{h}_s)$. We now consider two simple forms for the kernel function K.

If we use a Dirac delta function kernel centered at each sample $K(\mathbf{h}, \mathbf{h}_s) = \delta(\mathbf{h} - \mathbf{h}_s)$, then we have: $\mathbf{h}^* \approx \arg\max_{\mathbf{h}} p(\mathbf{x}|\mathbf{h})\frac{1}{S}\sum_{s=1}^{S}\delta(\mathbf{h} - \mathbf{h}_s)$. This can be reduced to an equivalent discrete optimization problem where the goal is to find the most likely sample s^*:

$$s^* = \arg\max_s p(\mathbf{x}|\mathbf{h}_s) = \arg\min_s (\mathbf{x} - \zeta(\mathbf{h}_s))^\top \Sigma_\zeta (\mathbf{x} - \zeta(\mathbf{h}_s)), \tag{27}$$

by using the Gaussian form of $p(\mathbf{x}|\mathbf{h})$ as given in Eq. 20.

If instead we use Gaussian kernels centered at each sample $K(\mathbf{h}, \mathbf{h}_s) = \mathcal{N}(\mathbf{h}; \mathbf{h}_s, \Sigma_{Spl})$, then we have: $\mathbf{h}^* \approx \arg\max_{\mathbf{h}} p(\mathbf{x}|\mathbf{h})\frac{1}{S}\sum_{s=1}^{S}\mathcal{N}(\mathbf{h}; \mathbf{h}_s, \Sigma_{Spl})$. This approximation is harder to use in practice. Unlike the Dirac delta kernel approximation, the Gaussian approximation cannot be reduced to an equivalent discrete optimization since there is no guarantee that the optimal $\mathbf{h}$ for this form is among the samples in general.

5.3 Deterministic Approximate Inference: Mean Output (MO)

The structure of inference in the SMA, as well as the form of $p(\mathbf{h}|\mathbf{x}, y)$ employed, make it possible to construct a deterministic approximation to Eq. 26. The basic intuition is straightforward. For a given $\mathbf{x}$, we *ask* each specialized function ϕ_k to give its most likely estimate for $\mathbf{h}^*$. We then evaluate the probability of each function's estimate via the distribution $p(\mathbf{x}|\mathbf{h})$. This approximation is good in practice, as will be demonstrated in the experiments.

To justify this deterministic approximation, we note that the probability of the mean is maximal in a Gaussian distribution; *i.e.*, it is the most-likely value. Formally, in both Case (1) and Case (2) described earlier, $p(E[\mathbf{h}|\mathbf{x}, y, \theta]) \geq p(\mathbf{h}'|\mathbf{x}, y, \theta)$, for any $\mathbf{h}'$. Consider again the set of samples $\mathcal{H}_{Spl} = \{\mathbf{h}_s\}_{s=1...S}$ generated in the MS approximation. We can build a set of samples $\mathcal{H}_\phi = \{\mathbf{h}_k^\phi\}_{k=1...M}$ that has the property:

$$\forall y, \max_k p(\mathbf{h}_k^\phi|\mathbf{x}, y) \geq \max_s p(\mathbf{h}_s|\mathbf{x}, y) \tag{28}$$

simply by setting $\mathbf{h}_k^\phi = \phi_k(\mathbf{x}, \theta)$.

This insight leads to a deterministic approximation for inference, the *Mean Output* solution (MO). This approximate solution relies on the observation that by considering the means $\phi_s(\mathbf{x})$, we would be considering the most likely output of each specialized function, given the input. The smaller the overlap among the distributions associated with each specialized function, the better the accuracy of this approximation.

In MO approximate inference, the expression to be minimized is the same as that used in Eq. 27, except for the use of the M means instead of the S samples:

$$k^* = \arg\max_{k \in \mathcal{C}} p(\mathbf{x}|\mathbf{h}_k^\phi) = \arg\min_{k \in \mathcal{C}} (\mathbf{x} - \zeta(\mathbf{h}_k^\phi))^\top \Sigma_\zeta (\mathbf{x} - \zeta(\mathbf{h}_k^\phi)). \tag{29}$$

This generally requires substantially less computation than would be required in the MS approach.

5.4 Bayesian Inference

Note that in some applications, instead of a *point* estimate the most likely output $\mathbf{h}^*$, it may be desirable to employ an approximation to the full posterior distribution $p(\mathbf{h}|\mathbf{x})$. We can show that for the two kernel functions, K given in Sec. 5.2 we can respectively obtain

$$p(\mathbf{h}|\mathbf{x}) \propto \frac{1}{S} \sum_{s=1}^{S} \mathcal{N}(\mathbf{x}; \zeta(\mathbf{h}_s), \Sigma_\zeta), \tag{30}$$

$$p(\mathbf{h}|\mathbf{x}) \propto \frac{1}{S} \mathcal{N}(\mathbf{x}; \zeta(\mathbf{h}), \Sigma_\zeta) \sum_{s=1}^{S} \mathcal{N}(\mathbf{h}; \mathbf{h}_s, \Sigma_{Spl}). \tag{31}$$

These approximations can be useful in algorithms that carry a *distribution* over the possible state $\mathbf{h}$. For example, in the context of dynamic probabilistic models, such as Markov models, one would like to fuse past pose estimates with new observations, i.e., to obtain distributions of the $p(\mathbf{h}_t|\mathbf{x}_t, \mathbf{h}_{t-1})$.

6 Example Application: Articulated Pose from Visual Features

The SMA formulation is rather general, and could be applied in a number of supervised learning problems for which the output-to-input (feedback) map is relatively easy to compute. To demonstrate and test our framework, we have developed a system that uses the SMA to infer articulated pose from low-level visual features. In particular, we focussed on pose estimation of the human hand and body from an image silhouette. In this class of computer vision applications, ground truth datasets for use in training can be obtained via motion capture gloves or body suits, and computer graphics rendering can be used to generate the input-output pairs used in supervised learning. We will now give details of this demonstration system.

6.1 3D Hand Pose Estimation

In this application, our goal is to recover detailed 3D hand pose from silhouette features computed from a single color image. Hand pose is defined in terms of the hand joint angles. In general, we are also interested

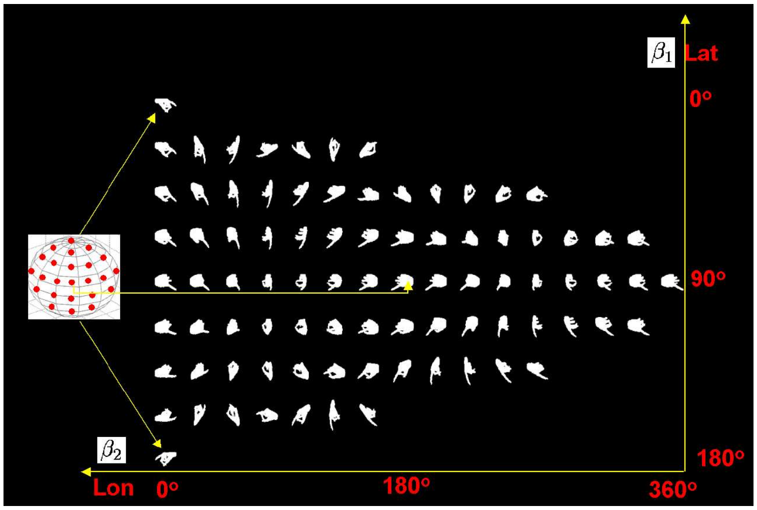

Figure 3: Example of the 86 silhouettes obtained via computer graphics rendering for a given a 3D hand pose. Views are distributed approximately uniformly over the view sphere.

in global orientation of the hand. We explore two applications: estimation of the internal joint angles only, and later, estimation of both internal joint angles and global orientation of the hand.

6.1.1 Hand Model

We utilize the hand model provided in the VirtualHand programming library [41]. The model parameters are 22 joint angles. For the index, middle, ring and pinky finger, there is an angle for each of the distal, proximal and metacarpophalangeal joints. For the thumb, there is an inner joint angle, an outer joint angle and two angles for the trapeziometacarpal joint. There are also abduction angles between the following pairs of successive fingers: index/middle, middle/ring and ring/pinky. Finally, there is an angle for the palm arch, an angle measuring wrist flexion and an angle measuring the wrist bending towards the pinky finger. However, because the former two wrist angles also encode global orientation, we decided not to model them in our application. Hence, ignoring these two angles, our model has 20 DOF for the internal hand configuration.

All of these 20 angles are relative to two global orientation angles. These two angles will encode the camera viewpoint (or alternatively hand 3D rotation). Imagine a sphere surrounding the hand model, *i.e.*, a fixed hand center point is at the center of the sphere. For ease of reference, we will employ the widely used latitude and longitude notions. The first angle β_1 represents the latitude from which we are looking at the hand, the second angle β_2 represents the longitude. We have defined $\beta_1 \in [0, \pi]$, with zero and π being the *poles* of the sphere and $\beta_2 \in [0, 2\pi)$. Thus, in summary our full hand model has 22 DOF.

6.1.2 3D Hand Motion Datasets

Using a CyberGlove, we collected approximately 9,000 examples of 3D hand poses. This data included hand configurations from American Sign Language (ASL) and other configurations informally performed by several members of our research group. Using computer graphics and an artificial hand model, we then rendered each captured hand pose from multiple viewpoints on the view sphere. In our implementation, we defined a set of 86 viewpoint angle pairs (β_1, β_2) so that the sphere surface is sampled approximately uniformly. Thus we obtained a full dataset of $9,000 \times 86$ views. Each view has an associated binary image mask (silhouette), and a 22 DOF pose vector. Fig. 3 shows the 86 viewpoints used in the dataset.

¿From these silhouettes, we extract the visual features that will be used for further processing. In our implementation, we used two classes of features (these features are not used together): Hu moments and Alt moments. Alt moments [1] are translation and scale invariant, but not rotation invariant. Hu moments [19] are invariant to translation and scaling, but also invariant to rotation in the image plane. These moment features were used in our implementation because they are relatively easy to compute, and they provide invariants that are appropriate for our demonstration application. However, the general SMA formulation can be used with other visual feature representations if desired. Detailed examination of the feature selection problem is outside the scope of this paper, and remains a topic for future research.

The above process yields a set of input-output (cue-pose) pairs to be used in our experiments. We define two experimental datasets:

1. *Hand-Single-View:* In this dataset, the hand is viewed from only one viewpoint ($\beta_1 = \pi/2$, $\beta_2 = 0$), generally making the palm of the hand visible. Silhouette features are computed using Alt moments. This yields approximately 9,000 input-output pairs.

2. *Hand-All-Views:* In this dataset, the hand is viewed from all 86 viewpoints. Silhouette features are computed using Hu moments. This yields approximately 750,000 input-output pairs.

6.1.3 Hand Detection and Segmentation

For live video input, we will use video sequences collected with a color digital camera. It will be assumed that these sequences have a static background and only one person is present. In this implementation, we are not considering hand occlusion analysis, which by itself is a difficult task. Our system tracks both hands of the user automatically using a skin color tracker [37, 34].

6.2 2D Human Body Pose Estimation

In this application, our goal is to recover the articulated pose of a human body observed in a single image. The methodology followed is very similar to that used in the estimation of hand pose. However, instead of joint angles, body pose will be specified in terms of marker positions at a predetermined set of joints. The SMA will estimate the 2D positions of these body markers in the image plane, given visual features as input.

6.2.1 Human Body Model

The human body model is defined in terms of 20 3D marker positions (60 DOF). The 20 markers are distributed as follows: three markers for the head, three markers for the hip/back bone articulation, plus one marker for each shoulder, elbow, wrist, hand, knee, ankle, and foot. For computer graphics rendering, the body model is composed of cylinders of equal width. The cylinders connect the markers to form the standard human body structure. The thorax is modeled using a wider cylinder. Because we are only interested in the shape of the projected model, we do not include texture or illumination in our rendering.

6.2.2 Human Body Pose Dataset

Human body motion capture data was obtained from several sources: http://www.biovision.com, Matt Brand's dataset [6], and several demo sequences in the software package *Character Studio*. In total there are 32 captured sequences that depict variations of different activities: dancing, walking, kicking, waving, throwing, jumping, signaling, crouching down. The total number of frames collected is approximately 7,000, mostly at 30 frames/second. Using computer graphics and our artificial body model, we then rendered each frame from 16 equally-spaced viewpoints on the equator of the view sphere centered at the hip of the body model. For each view, we also used the camera model to obtain the 2D marker positions in the image plane. Thus we obtained a full dataset of approximately $7,000 \times 16$ views. Each view has an associated binary image mask (silhouette), and a 40 DOF projected marker vector.

¿From the silhouettes, we extract the visual features that will be used as input to the SMA. For this application, we have chosen Alt moments [1] as our visual features, mainly due to their ease of computation and invariance to translation and scaling.

The above process yields a set of input-output (cue-pose) pairs to be used in our experiments. In this case, the cues are the Alt moments for a particular view, and the pose is encoded in terms of the projected locations of the body markers in the image plane (40 DOF). We call this dataset the *Body-All-Views* dataset.

6.2.3 Detection and Segmentation

For live video input, we use sequences collected with a color digital camera. It is assumed that these sequences have a static background, only one person is present, and the person is fully-visible. We use a simple and widely-used human body segmentation scheme [17, 42]. The technique employs statistical learning to acquire a model of the background appearance, where each pixel's color (luminance) is represented by a Gaussian distribution. Segmentation is then approached in a maximum-likelihood fashion, where each pixel is classified as belonging to one of two classes: the background or the foreground (human body).

6.3 Common Implementation Details

We know briefly discuss implementation details common to both applications.

6.3.1 Mapping Functions

In Sec. 3, it was not specified what class of mapping functions ϕ_k were to be used. The SMA framework is practically independent of this choice. However, from Eq. 14 we can notice that there are clear advantages in the M-step if these functions are differentiable with respect to their parameters. In the case of quadratic or linear functions, the M-step can be performed exactly in one step. However, the flexibility of these functions is limited. In our implementation each mapping function is a multi-layer perceptron with one hidden layer (MLP). For the non-linear one hidden layer perceptrons, there does not exist a closed-form solution for Eq. 14. In our implementation, we used four to five iterations of the conjugate gradient method per M-step.

6.3.2 Feedback Functions

In the previous sections we made reference to the inverse or feedback function denoted ζ. There are at least two ways to define this function. On the one hand, ζ could be a computer graphics rendering function. On the other hand, we could estimate an approximate $\hat{\zeta}$ given a set of output-input training examples. In our implementation, we experimented with both approaches. For ζ, we used computer graphics renderings of our hand and body models obtained via OpenGL. For $\hat{\zeta}$, we used a one hidden-layer perceptron, with twenty hidden nodes. In our experience, this provides an adequate and efficient approximation.

The approximate feedback function is useful primarily because it is faster to compute than a graphical rendering followed by visual feature computation. The key issue to keep in mind is that the feedback mapping is assumed to be simpler (one-to-one or even many-to-one), otherwise simple functional forms would only introduce more estimation errors. Of course, this is just a practical issue. If the feedback mapping is too complex to approximate easily, we could always rely on the available feedback function ζ.

6.3.3 Computational Performance

For an Athlon 1400 PC with 2GB memory, running unoptimized Matlab 6.0 code, it takes approximately five hours to train a model with 10 dimensions (input) and 10 dimensions (output), using 4500 patterns, and 40 single hidden layer perceptrons with five hidden nodes each.

Using the same setting, the system can infer body poses at approximately 11 frames per second, using the Mean Output (MO) algorithm. SMA related computations take approximately 70% of this time. This time includes OpenGL-based rendering of body poses in ζ. The rest is spent in segmentation and feature calculations. The Multiple Sample (MS) algorithm takes time proportional to the number of samples used. Of course, segmentation and feature computation for the segmented image is done only once. We noticed that for our implementation, if we use the approximate feedback function, $\hat{\zeta}$, the rendering time is reduced to approximately one-fourth.

6.3.4 Early Stopping During Training

During model training, we used cross-validation for early stopping and to avoid over-fitting as follows:

- *Training data:* Stop if the log-likelihood changes less than 0.5% averaged over the last ten iterations.
- *Held out data:* Stop if the held out data log-likelihood average change is negative over the last ten iterations. Held out data was chosen in the same way as the training and test data.
- *Number of iterations:* Stop if a maximum of 200 iterations is reached.

7 Experimental Results

We now present experimental results obtained using the SMA in estimating the pose of the human hand and body. For many additional experiments not included due to space limitations, the reader is referred to [33].

7.1 Hand Pose Estimation Given a Fixed Camera Viewpoint

In our first experiments, the SMA is tested in the task of recovering 3D human hand pose given a fixed camera viewpoint: a view towards the palm of the hand. For training, we used the *Hand-Single-View* dataset, which contains a total of approximately 9,000 examples. Of these, 3,000 were used for training and the rest for testing. All experiments were performed on a test set that shared no common poses with the training set. The input-output pairs were then defined as follows. The input consisted of 10 Alt moments computed from the silhouette of the hand, as described in Sec. 6.1. The output consisted of 20 joint angles of a human hand linearly encoded by nine values using Principal Component Analysis (PCA).

In this experiment, the number of specialized functions was set to 20. This number was found to be optimal in the sense of the Minimum Description Length (MDL) principle [32]; an exhaustive search is impractical, so we find this number via approximate search. Each mapping function was a one hidden layer, feed-forward network (multi-layer perceptron) with seven hidden neurons.

7.1.1 Quantitative Results

To measure the accuracy of the hand pose reconstruction, we randomly selected approximately 4,000 frames not included in the training set. This test set has the advantage that ground truth is available. Using the estimated feedback function $\hat{\zeta}$ in the Mean Output approach (MO), the average L_2 error between reconstruction and ground-truth was 0.1863 radians (approximately 10^o), with variance 0.0185. These error estimates are averaged over joint angles. We ran this experiment with the same test set, but instead used the computer graphics rendering feedback function ζ. When using ζ, similar accuracy was obtained. The average L_2 error between reconstruction and ground-truth in this case was 0.241 radians, with variance 0.0312. In [33], we explain in detail the reasons for this relatively small difference in performance.

Fig. 4 shows example reconstructions obtained via the MO approach. In many cases, the reconstruction is close to the ground truth. In other cases, the silhouette is highly-ambiguous, and the reconstruction does not match ground truth. A good example is shown in image pair number 34 (the last row-pair, fourth column), where the camera's image plane is perpendicular with the axis of the pinky finger. Note that the estimated hand pose disagrees with the ground-truth in the several joint angles associated with this finger. Similar effects with other joint angles can be seen in example pairs 8, 26, 37, etc.

Ambiguous configurations are indeed very common with a binary image representation. Note that in other ambiguous cases shown in Fig. 4 reconstruction is closer to ground truth, *e.g.,* pairs 29, 30, etc. Possible reasons for this agreement are diverse:

1. The input is not really ambiguous (probabilistically speaking) in the observation space. The other possible outputs (geometrically speaking) associated with this input may be very unlikely given the training set. This depends on the underlying structure of the configuration manifold. One of the main

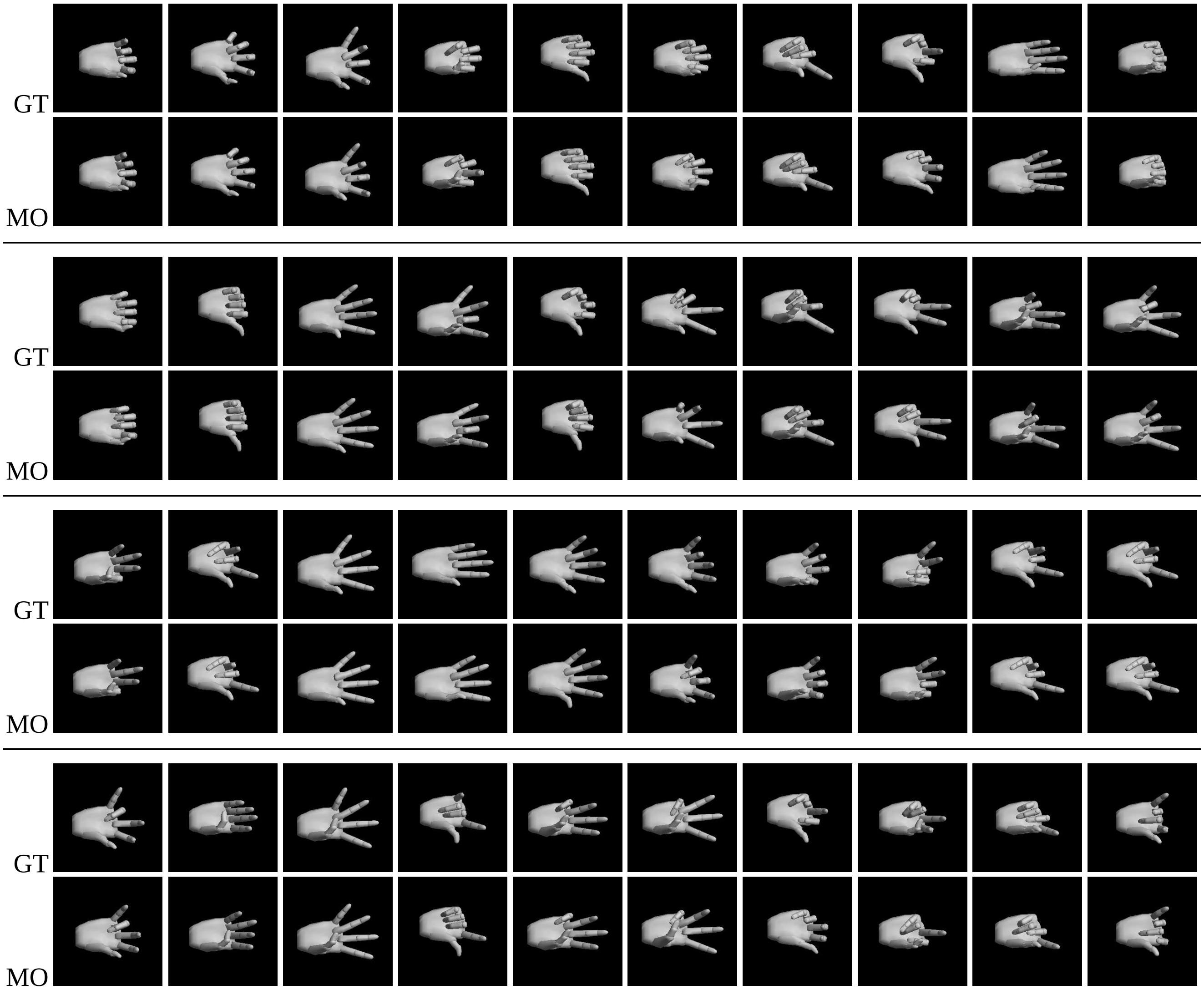

Figure 4: 40 examples of estimated hand poses chosen uniformly at random. Reconstruction found using the Mean Output (MO) approach. The feedback function used was estimated from data. Each example consists of a pair of images: ground-truth (top), and estimate obtained using the mean output algorithm (bottom).

goals of a learning algorithm is to find this structure. Indeed these results show that our algorithm is finding this structure, since in most cases, MO finds a valid sample from the manifold.

2. Few mapping functions were trained to map this input, therefore the rest of the functions produced irrelevant (bad) outputs.

3. By chance, among many very similarly probable solutions, the *right* one was chosen. Of course, even with the help of chance in this case, the mapping functions needed to provide the right mapping for the given input $\mathbf{x}$.

The accuracy of the Multiple Samples (MS) inference approach was tested in similar experiments with approximately $4,000$ randomly chosen test examples not included in the training set. When the estimated feedback function $\hat{\zeta}$ was used, the mean L_2 error of the most likely sample to the ground-truth was 0.2202 radians with variance 0.0228. The mean error and variance from the best 20 samples was 0.308 and 0.3023 respectively. When we performed the same experiment, but instead used the computer graphics feedback function ζ, we observed very small quantitative differences. We obtained a mean error of 0.2628 radians with variance 0.0242 for the most likely sample. The mean error of the best 20 samples was 0.3128 radians with variance 0.3000.

These experiments confirmed that MO inference seems to provide a reasonable approximation, at least for this dataset. Recall from Sec. 5.3 that MO inference was based on the premise that the most-likely reconstruction given by each specialized function provides a good approximation to the best solution given by the full probability distribution.

7.1.2 Experiments with Real Images

We now test our approach using uncalibrated video sequences, where the camera is pointing towards the palm of a person's hand. On average, the hand occupied an area of approximately 200×200 pixels. Segmentation was obtained as described in Sec. 6.1.3.

In the first experiment, we use the MO approach to obtain a single *best* estimate for each segmented hand. Estimates for 40 frames, taken 0.9 seconds apart, are shown in Fig. 5. Visually we can notice that in most cases the estimate is a plausible explanation of the segmented silhouette. However, there are also a few inaccurate reconstructions.

In general, it is expected that the SMA model cannot perform well in all configurations (this is true for almost any machine learning model) due to the following reasons:

1. Learning is the result of optimizing an *expected* or average error.

2. The real hand and synthetic hand model features are similar but not the same. Anthropometric differences can influence inference accuracy.

3. Even the best model could fail in some configurations. Information theory tells us that this is always the case except when the *information* in the features is equal to the entropy of the body pose configurations; in other words, when features tell us everything needed about the configuration. Otherwise, there might be multiple explanations for a given visual feature vector.

In order to test the ability of the system to provide these multiple explanations, we tested the Multiple

Samples (MS) approach. Fig. 6 shows the estimates found using MS. These estimates can be interpreted as possible hypotheses of hand configurations given the silhouettes.

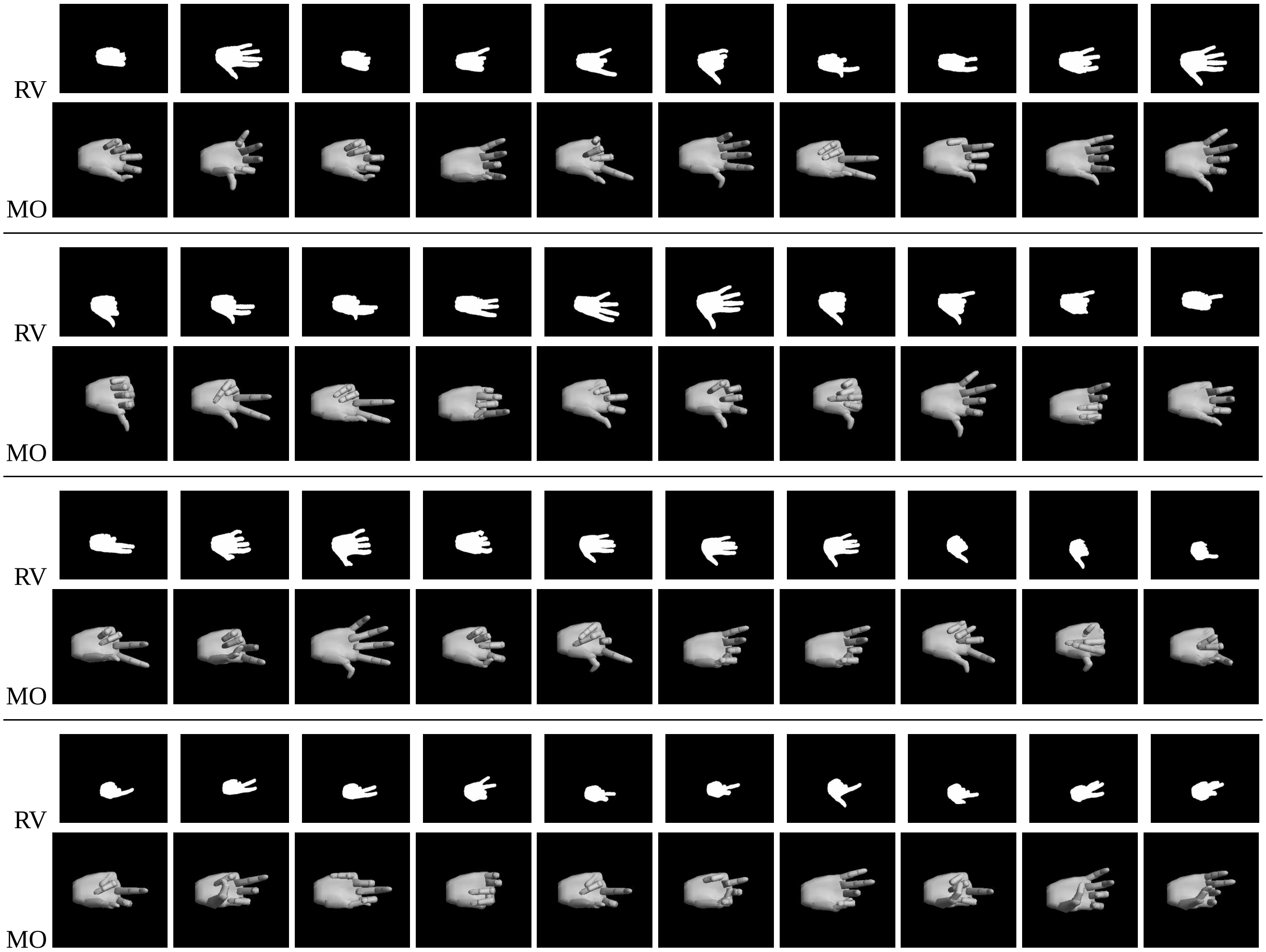

Figure 5: 40 examples of estimated hand poses captured every 0.9 secs from real video (RV). Reconstruction found using the Mean Output (MO) approach. The feedback function was computed using computer graphics rendering.

7.2 3D Hand Pose Reconstruction Given an Unrestricted Camera Viewpoint

The SMA is now tested in the task of recovering 3D human hand pose from an unknown camera viewpoint. For training, we used the *Hand-All-Views* dataset, which contains a total of approximately 750,000 examples. Of these, 18,000 were used for training and the rest for testing. The input-output pairs were then defined as follows. The input consisted of seven Hu moments computed from the silhouette of the hand, as described in Sec. 6.1. The output consisted of 20 internal joint angles of the hand and two orientation angles. This 22 DOF representation was linearly encoded by nine values using PCA.

The number of specialized functions was set to 45. This number was determined via the MDL criterion,

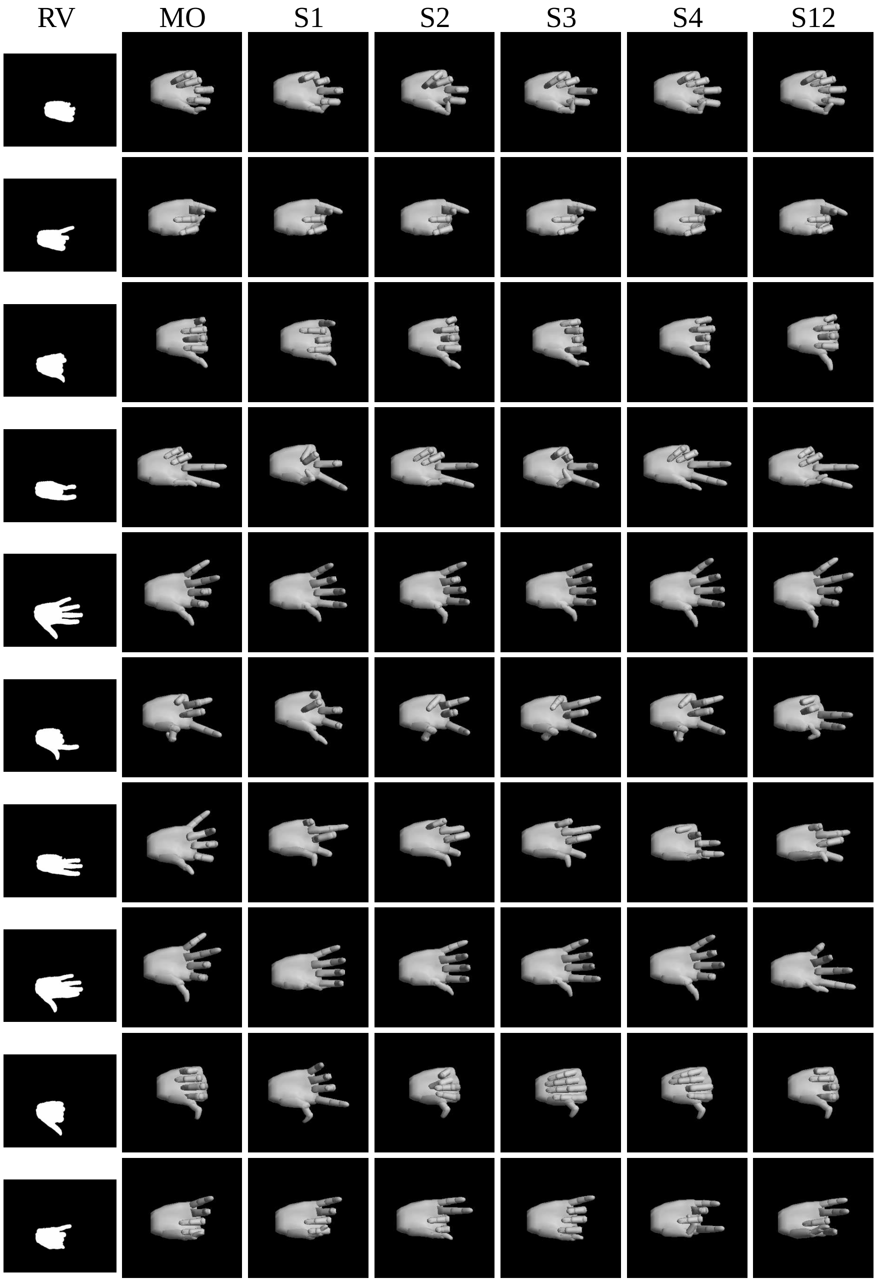

Figure 6: Example estimated hand poses obtained using the Multiple Sample (MS) approach using real video (RV). The feedback function was estimated from data.

as before. Each specialized function was a one hidden layer, feed-forward network with seven hidden nodes.

7.2.1 Quantitative Results

We computed the L_2 error in estimating hand pose, and quantitatively compared this measure across views. Fig. 7 shows the error of the most likely estimate found using the MO approach. From the graphs we see that views towards the palm of the hand (90°) are slightly easier to reconstruct on average, while the

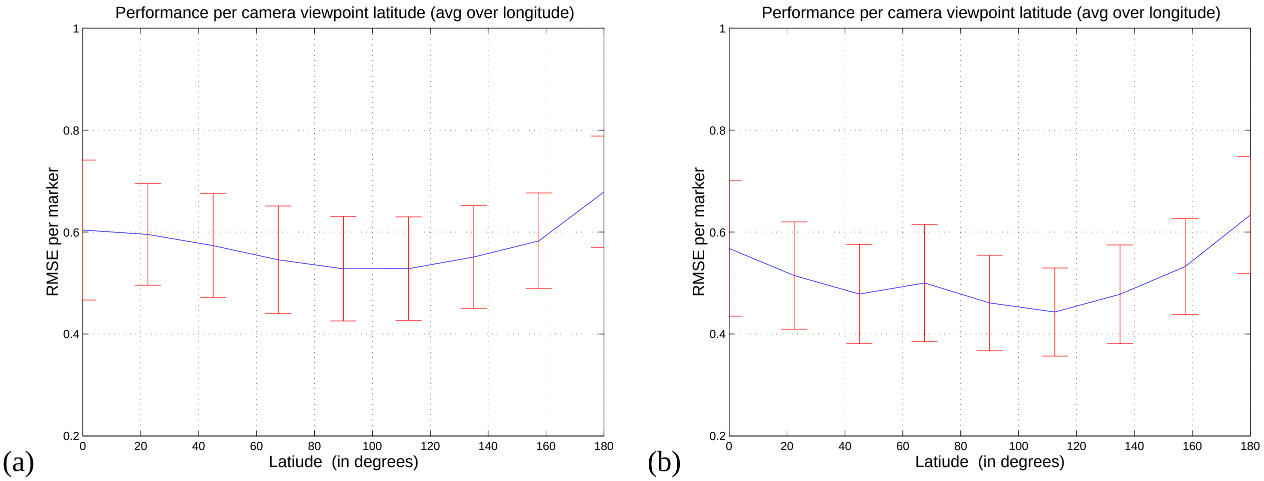

Figure 7: Mean Output (MO) inference performance for unrestricted view tests at given viewpoint latitudes (averaging over longitude). The feedback function is (a) the estimated $\hat{\zeta}$ (b) the computer graphics rendering ζ. A frontal view of the hand palm is at latitude $\beta_1 = \pi/2$, longitude $\beta_2 = 0$.

variance seems similar across views. As expected, the average error is higher than that obtained for the fixed view hand pose reconstruction experiments. The differences in performance obtained from using ζ or $\hat{\zeta}$ are relatively small. However, it seems that for unrestricted hand views it is advantageous to use the computer graphics feedback function ζ. This is probably because estimating this inverse mapping $\hat{\zeta}$ over unrestricted viewpoint is more complicated than for only frontal hand views (and the mapping is likely to be more complex also).

Fig. 8 shows the results using the MS approach. Fig. 8(a) shows the error associated with the best sample. This error behaves very similarly to the MO error. Fig. 8(b) shows the average error computed using the best 20 samples. This error is higher than that of the best sample. Note that this is not an obvious result given that the best sample is determined without having knowledge of ground-truth. In fact, if the average error of the best 20 samples were lower than that of the best sample, then we could infer that our algorithm is very inaccurate at determining what samples are better. Thus this result positively endorses our MS algorithm.

For comparison, we used the ground-truth to select the best sample, based on minimum RMSE. In other words, we have an oracle that picks the sample closest to the ground-truth. The resulting performance graph is shown in Fig. 8(c). This represents the lower-bound on the reconstruction error using the learned forward model. The graph is interesting in the sense that it separates the errors from the forward and feedback models. The feedback model produces a RMSE < 0.35 across views. This is roughly half the total RMSE error produced by the SMA overall.

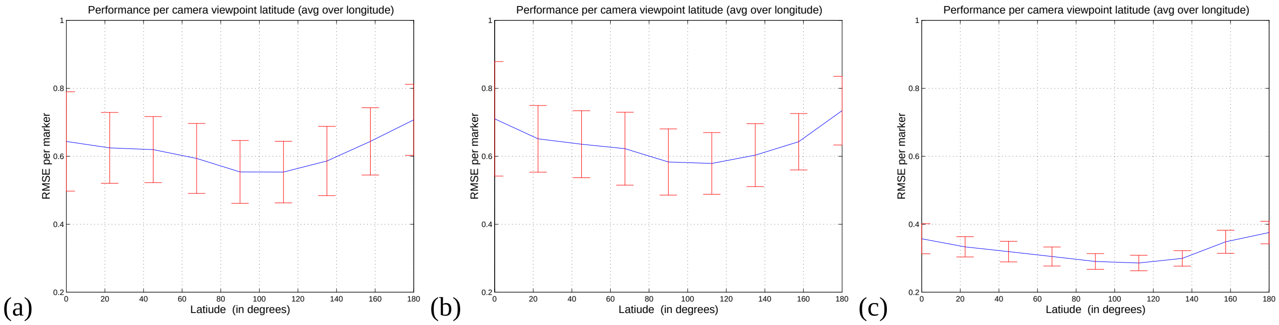

Figure 8: Multiple Samples (MS) inference for unrestricted view tests at given viewpoint latitudes (averaging over longitude). Feedback functions is the estimated $\hat{\zeta}$. A frontal view to the hand palm is at latitude $\beta_1 = \pi/2$, longitude $\beta_2 = 0$. (a) Most probable sample. (b) Average over all samples (20 most probable samples taken). (c) Best sample (determined using ground-truth information for comparison).

7.2.2 Experiments with Real Images

As before, we test our approach using video collected from a single uncalibrated camera. However, in this case, the person's hand can appear at any orientation.

Pose estimates from 40 frames (taken every 0.9 secs apart) obtained via the MO approach are shown in Fig. 9. Note that there are incorrectly-segmented hands in this sequence. We decided to leave these in to avoid frame rearrangements (losing the uniform frame sampling), to show that segmentation does not always work correctly, and to show that this approach is inherently robust to extreme segmentation errors. In this experiment, there was usually visual agreement between reconstruction and estimate as seen in the figure. Note that even for a human observer, looking at the segmented silhouettes in the figure, reconstruction is sometimes ambiguous. There are also some configurations for which the system did not perform correctly.

Fig. 10 shows the estimates obtained via the MS approach. The frames shown were taken approximately every 0.9 seconds. In the second row, we can see some limitations of the Hu moment feature space: sometimes, different hand orientations are very similar in the feature space. These apparently different hypotheses may actually be close to each other in terms of their probability, given the features. The same effect repeats clearly in the third and sixth row. This problem might be alleviated by using a different input feature space. At an extreme one might consider the full silhouette as a feature. Of course there are important trade-offs to take into account when considering different features; e.g., invariants, and dimensionality.

7.3 2D Human Body Pose Reconstruction

The SMA is next tested in the task of estimating human body pose. The goal is to estimate the 2D locations of body markers in the image, given visual features computed from the person's silhouette. In this experiment, we use the *Body-All-Views* dataset, which contains a total of of over 100,000 samples. Of these, 8,000 were

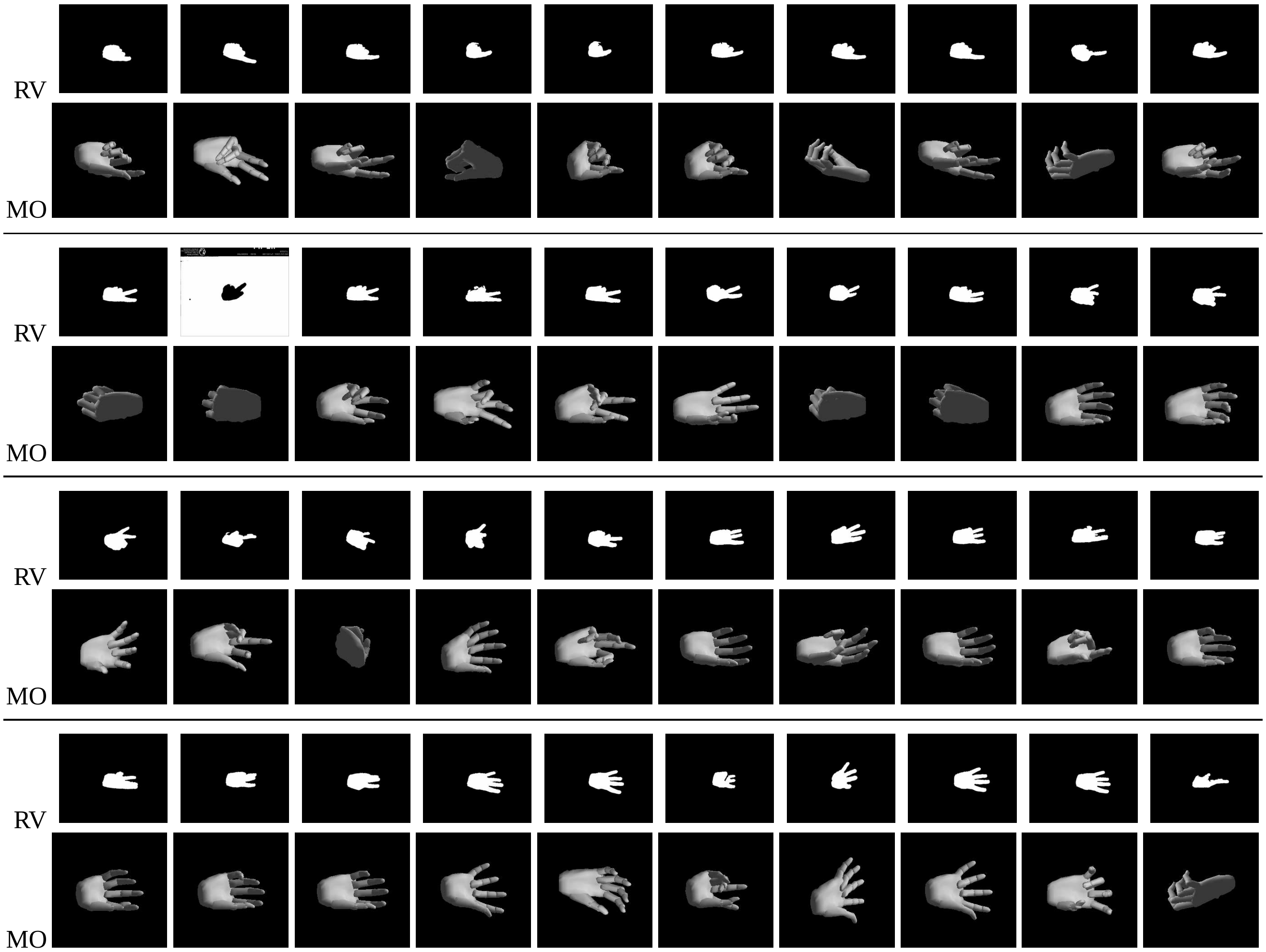

Figure 9: 40 examples of estimated hand poses captured every 0.9 secs from real video (RV). Reconstruction found using the Mean Output (MO) approach. The feedback function was computed using computer graphics rendering.

used for training and the rest for testing. The input-output pairs were defined as follows. The input consisted of the 10 Alt moments computed from the silhouette. The output consisted of 20 2D marker positions (40 DOF), which were then linearly encoded by nine values using PCA.

The number of specialized functions was set to 15. This number was determined via the MDL criterion, as before. Each specialized function is a one hidden layer, feed-forward network with seven hidden nodes.

7.3.1 Quantitative Results

Fig. 11 shows the reconstruction obtained with the MO approach for frames taken from three synthetic sequences excluded from the training set. The agreement between reconstruction and observation is easy to perceive for all frames. Also, for self-occluding configurations, the estimate is still similar to ground-truth.

Fig. 12 shows the average marker error and variance per body orientation in percentage of body height.

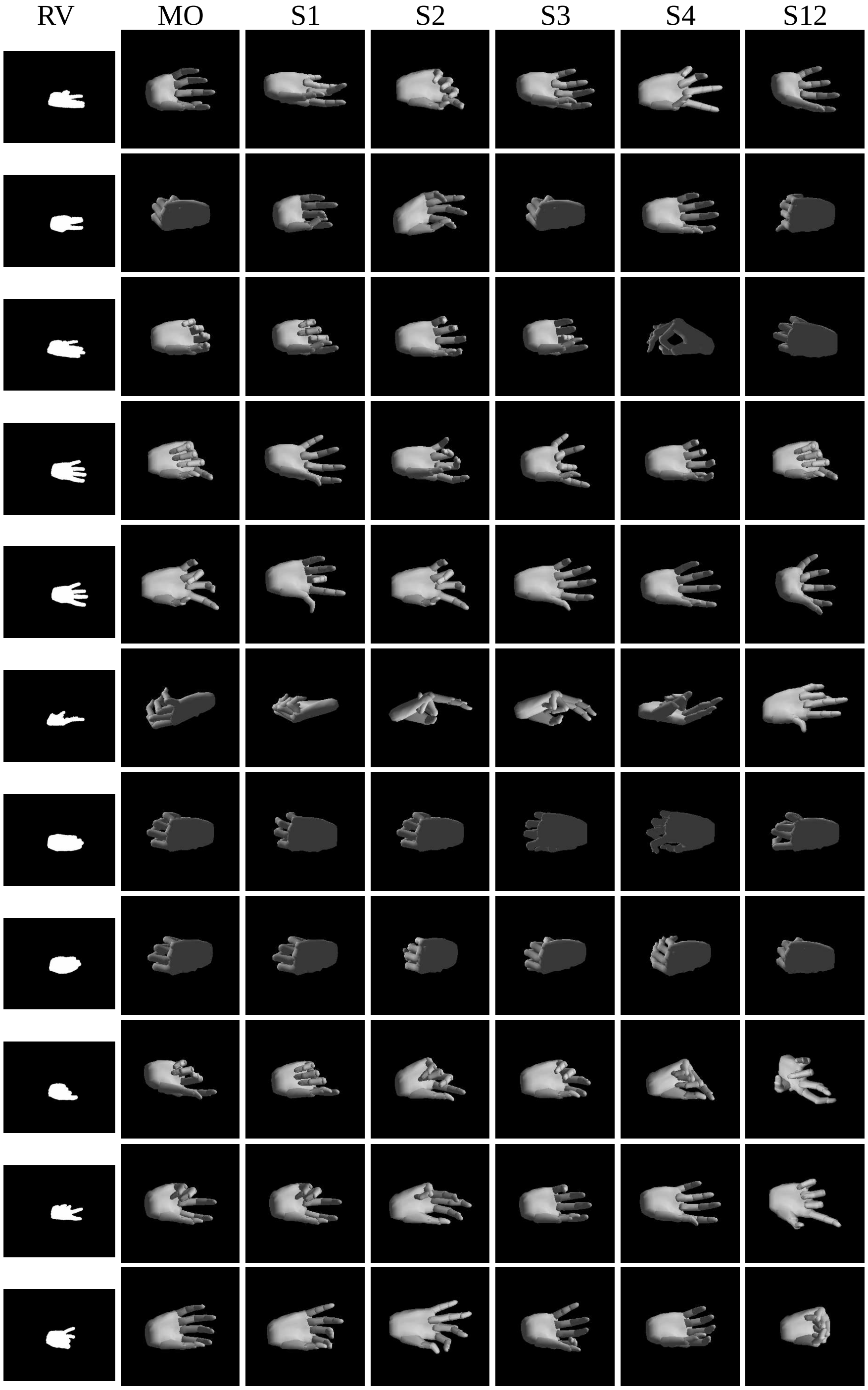

Figure 10: Example estimated hand poses obtained using the Multiple Sample (MS) approach and real video (RV). The feedback function was computed using computer graphics rendering.

Note that the error is bigger for orientations closer to 0 and π radians. This intuitively agrees with the notion that at those angles (side-views), there is less visibility of the body parts. We consider this performance promising, given the complexity of the task and the simplicity of the approach. By choosing poses at random

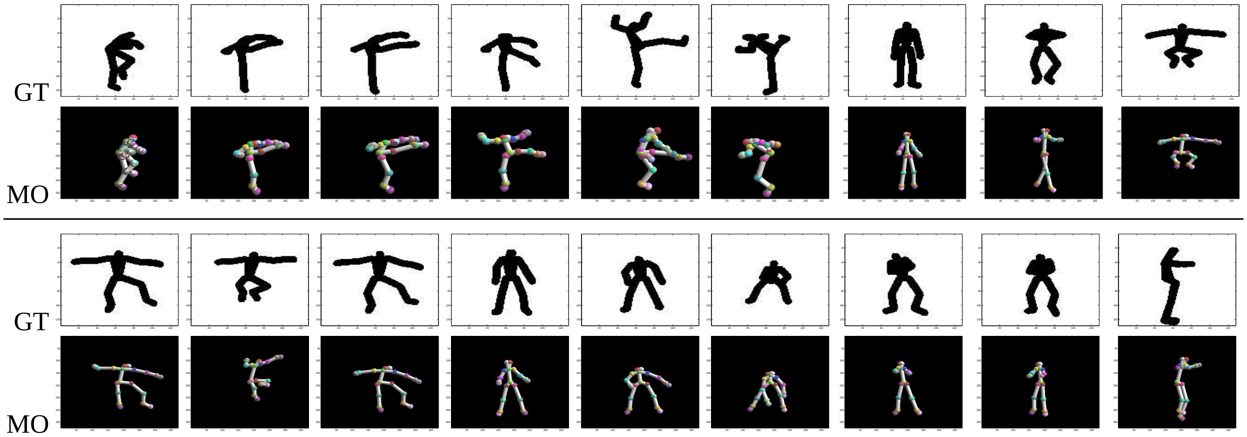

Figure 11: Example reconstruction of frames from test sequences with computer graphics-generated silhouettes.

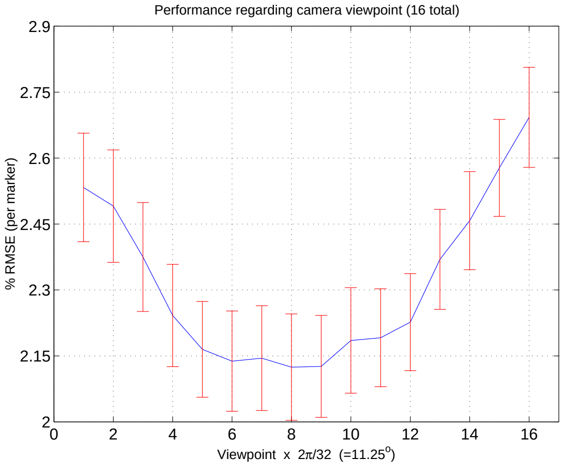

Figure 12: Marker root-mean-square-error and variance per camera viewpoint (every $2\pi/32$ rads.). Units are percentage of body height. Approx. 110,000 test poses were used.

from those excluded from the training set, the RMSE was 10.35% of body height (with 20% variance). In related work, quantitative performance has usually been ignored, in part due to the lack of ground-truth and standard evaluation datasets.

7.3.2 Experiments with Real Images

We now test the approach using real video sequences of human body motion. We use the basic segmentation approach described in Sec. 6.2.3 to obtain silhouettes.

Fig. 13 shows examples of system performance obtained via the MO approach for several relatively complex motion sequences. Even though the characteristics of the segmented body differ from the ones used for training, good performance is still achieved. Most reconstructions are visually close to what can be thought of as the right pose reconstruction. Body orientation is also generally accurate.

Fig. 14 shows the top-ranked pose samples obtained via the MS approach. Note that despite low-quality segmentation, the system outputs reasonably accurate pose hypotheses. Orientation is accurate and the relative limb relationships are maintained. However, we can observe that some poses are inherently difficult and the estimate lacks enough pose detail to be perceived as a good estimate. For example, the eighth row shows a side view of a person raising one arm while keeping the other arm at rest. The resulting MS estimates all show a side-view, however none has the correct arm configuration. This could be due to the lack of relevant training data, or due to differences between the rendered model and the real body observed.

In this work, we did not pursue use of a more realistic human body renderer. This could affect the performance with real data since, as in most learning methods, it is critical that the training data be a good approximation to the data the algorithm will be tested with. Due to differences in shape and width of body components observed in training versus testing, the visual features may differ. Improving the match between visual features used in training and testing is an area that we plan to investigate in future research. In theory this could allow us to adapt our algorithm to different body or hand anthropometric characteristics.

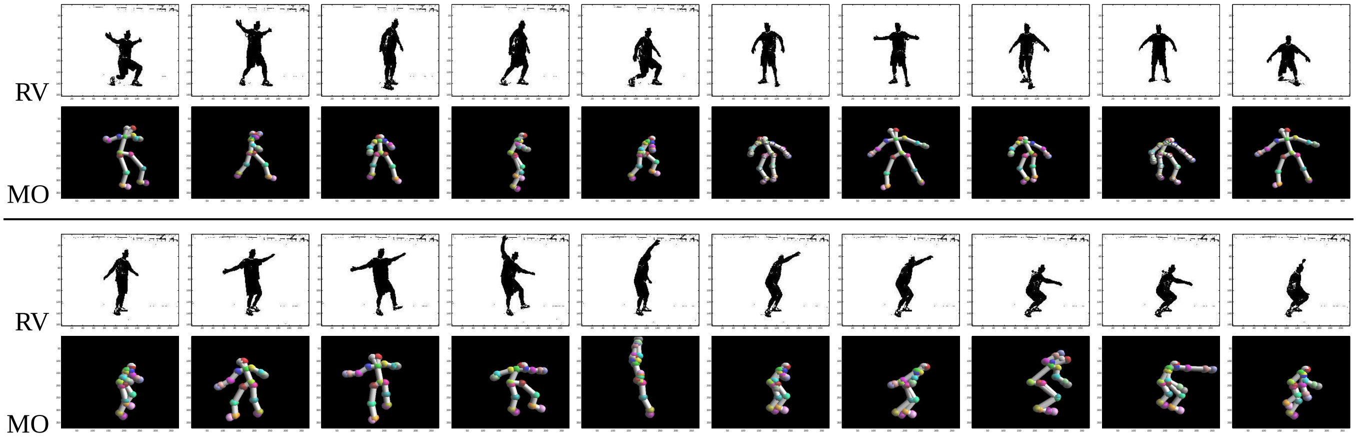

Figure 13: Reconstruction obtained from observing a human subject (every 10th frame).

8 Conclusions

In this paper, we have described a novel supervised learning framework: the Specialized Mappings Architecture (SMA). The SMA employs a set of several mapping functions that are learned from training data. Each specialized function maps certain domains of the input space onto the output space. The SMA learning formulation uses ideas from Maximum Likelihood estimation and latent variable models. A variant of the Expectation-Maximization algorithm is used for simultaneous learning of the specialized domains along

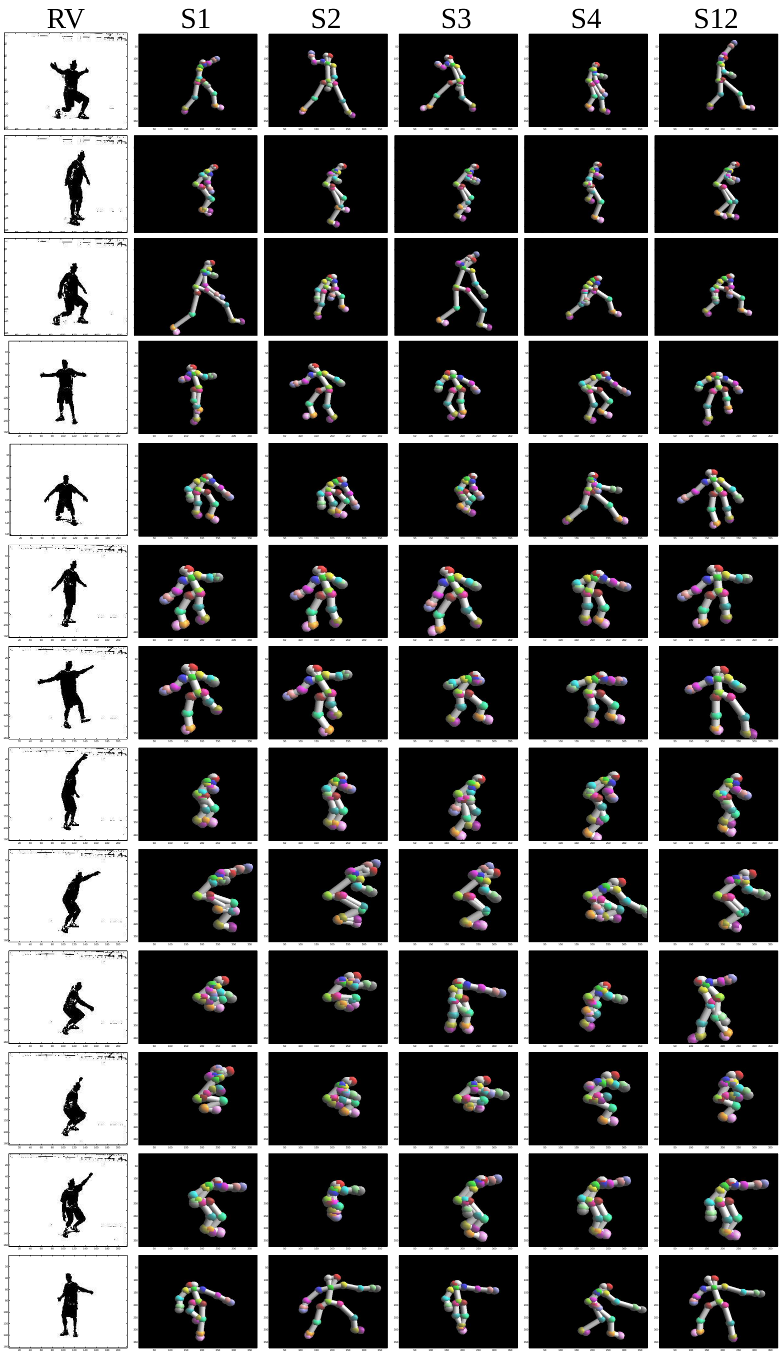

Figure 14: Estimated body poses from real sequences obtained via MS inference.

with the mapping functions. One key advantage of the SMA is that it can model ambiguous, one-to-many mappings that may yield multiple valid output hypotheses.

Another key advantage of the SMA formulation is its incorporation of a feedback or inverse function, ζ in statistical inference. Use of ζ affords an alternative to the gating networks of the Mixture of Experts paradigm [24] in that it allows for simpler forward models (also see [16, 12] for other models). The forward model in the SMA assumes that the mixing factors are independent of the input, as seen in Sec. 3. At first sight, this seems to limit the architecture's expressiveness. However, the SMA's combination of forward and inverse models eliminates this independence assumption, as seen in Sec. 5.1. In other words, ζ provides an alternative that avoids increasing the forward model complexity without restricting model expressiveness. Note that in the SMA formulation, different sets of appropriate conditional independence assumptions are specified by the forward and inverse models. In applications such as those presented in this paper, ζ can be a computer graphics rendering function or an approximation $\hat{\zeta}$ can itself be learned from training data. Thus, the SMA exploits available prior information about the structure of the problem.

The SMA framework was demonstrated in a computer vision system that can estimate the articulated pose parameters of a human body or human hands, given features computed from an image silhouette. Articulated pose reconstruction from a single image is a particularly difficult problem because this mapping is highly-ambiguous and complex. We have obtained promising results even using a very simple set of image features, such as moment invariants of the hand or body's image silhouette. Choosing the best subset of image features for this application is by itself a complex problem, and a topic of ongoing research.

The SMA offers several advantages over many previous methods for articulated pose estimation. Many previous approaches have tried in numerous ways to use camera geometry and/or model registration to perform pose estimation, resulting in iterative procedures that require careful choice of initial conditions (model placement). In the SMA approach no iterative minimization methods are used in pose inference. Moreover, SMA inference is fully automatic – no manual initialization of the articulated model is required. Another set of previous approaches attempt to learn articulated model dynamics [6, 18, 39]; however, learning dynamics requires substantially more training data, and tends to produce systems that are biased towards specific motions. The SMA framework avoids this and learns/estimates pose from a single image only.

It is also important to note that the SMA is a general nonlinear supervised learning algorithm. Thus, applications of the SMA need not be limited to the vision domain. As a simple example, one could apply the SMA approach in speech recognition problems, where the input space is given by features computed on acoustic signals (*e.g.,* cepstral coefficients), and the output space could be the space of phonemes. In this case, the feedback function would involve an acoustical rendering of phonemes.

Several interesting problems remain for future work. Within the context of articulated pose estimation, one topic for future investigation is how to adapt the system to a specific body morphology. Integration of SMA pose estimation with image segmentation for a fully-integrated detection and pose reconstruction formulation is also needed, and may enable greater robustness to occlusion and noise. More generally, methods for incorporating knowledge of dynamics in the SMA framework should be investigated, as discussed in

[33]. Another general problem is how to learn what the best (*e.g.*, visual) features are for specific problems or datasets. While promising advances have been made in boosting of features [11], extension of the SMA framework to incorporate such concepts remains a topic for future investigation.

Acknowledgments

The hand sequences used in our experiments were collected in collaboration with Vassilis Athitsos. This research was supported in part by the U.S. Office of Naval Research under grants N000140310108 and N000140110444, and the U.S. National Science Foundation under grants IIS-0208876 and IIS-9809340.

References

[1] F.L. Alt. Digital pattern recognition by moments. *Journal of the Association for Computing Machinery*, 9(2):240-258, April 1962.

[2] S. I. Amari. Information geometry of the EM and *em* algorithms for neural networks. *Neural Networks*, 8(9):1379–1408, 1995.

[3] C. Barron and I. Kakadiaris. Estimating anthropometry and pose from a single image. In *Proceedings Computer Vision and Pattern Recognition*, pages 669–676, 2000.

[4] M. Black and A. Jepson. A probabilitic framework for matching temporal trajectories: Condensation-based recognition of gestures and expressions. In *Proceedings European Conference on Computer Vision*, volume 1406, pages 909–917, 1998.

[5] M. Black and Y Yacoob. Tracking and recognizing rigid and non-rigid facial motion using local parametric models of image motion. In *Proceedings International Conference on Computer Vision*, 1995.

[6] M. Brand. Shadow puppetry. In *Proceedings International Conference on Computer Vision*, pages 1237–1244, 1999.

[7] C. Bregler. Tracking people with twists and exponential maps. In *Proceedings Computer Vision and Pattern Recognition*, pages 8–15, 1998.

[8] I. Csiszar and G. Tusnady. Information geometry and alternating minimization procedures. *Statistics and Decisions*, 1:205–237, 1984.

[9] A. Dempster, N. Laird, and D. Rubin. Maximum likelihood estimation from incomplete data. *Journal of the Royal Statistical Society (B)*, 39(1):1–38, 1977.

[10] J. Deutscher, A. Blake, and I. Reid. Articulated body motion capture by annealed particle filtering. In *Proceedings Computer Vision and Pattern Recognition*, 2000.

[11] Y. Freund and R. Schapire. A decision-theoretic generalization of on-line learning and an application to boosting. In *European Conference on Computational Learning Theory*, pages 23–37, 1995.

[12] J.H. Friedman. Multivatiate adaptive regression splines. *The Annals of Statistics*, 19,1-141, 1991.

[13] D. Gavrila and L. Davis. Towards 3-d model-based tracking and recognition of human movement: a multi-view approach. In *International Workshop on Automatic Face and Gesture Recognition*, pages 272–277, 1995.

[14] I. Haritaouglu, D. Harwood, and L. Davis. Ghost: A human body part labeling system using silhouettes. In *International Conference on Pattern Recognition*, pages 77–82, 1998.

[15] T. Heap and D. Hogg. Towards 3d hand tracking using a deformable model. In *Proceedings International Conference on Automatic Face and Gesture Recognition*, pages 140–145, 1996.

[16] G. Hinton, B. Sallans, and Z. Ghahramani. A hierarchical community of experts. *Learning in Graphical Models, M. Jordan (editor)*, pages 479–494, 1998.

[17] D. Hogg, S. Dudani, K. Breeding, and R. McGhee. Model-based vision: A program to see a walking person. *Image and Vision Computing*, 1(1):5–20, 1983.

[18] N.R. Howe, M.E. Leventon, and W.T. Freeman. Bayesian reconstruction of 3d human motion from single-camera video. In *Advances in Neural Information Processing Systems*, volume 12, pages 820–826, 2000.

[19] M. K. Hu. Visual pattern recognition by moment invariants. *IRE Transactions Information Theory*, IT(8):179–187, 1962.

[20] M. Isard and A. Blake. Condensation – conditional density propagation for visual tracking. *International Journal of Computer Vision*, 29(1): 5-28, 1998.

[21] R. Jacobs, M. Jordan, S. Nowlan, and G. E. Hinton. Adaptive mixtures of local experts. *Neural Computation*, 3(1):79–87, 1991.

[22] G. Johansson. Visual perception of biological motion and a model for its analysis. *Perception and Psychophysics*, 14(2): 210-211, 1973.

[23] M. Jordan. *Learning in graphical models*. Kluwer Academic, The Netherlands, 1999.

[24] M. Jordan and R. Jacobs. Hierarchical mixtures of experts and the EM algorithm. *Neural Computation*, 6, 181-214, 1994.

[25] H. Lee and Z. Chen. Determination of 3d human body posture from a singleview. *Image Understanding*, 30:148–168, 1985.

[26] R. Neal and G. Hinton. A view of the em algorithm that justifies incremental, sparse, and other variants. *Learning in Graphical Models, M. Jordan (editor)*, pages 355–368, 1998.

[27] D. Ormoneit, H. Sidenbladh, M. Black, and T. Hastie. Learning and tracking cyclic human motion. In *Advances in Neural Information Processing Systems 13*, pages 894–900, 2001.

[28] J. O'Rourke and N. Badler. Model-based image analysis of human motion using constraint propagation. *IEEE Transactions on PAMI, Nov. 1980*, 2:522-536(6), 1980.

[29] V. Pavlović, J. Rehg, and J. MacCormick. Learning switching linear models of human motion. In *Advances in Neural Information Processing Systems 13*, pages 981–987, 2001.

[30] J. Pearl. *Probabilistic Reasoning in Intelligent Systems*. Morgan-Kaufman, 1988.

[31] J. M. Rehg and T. Kanade. Model-based tracking of self-occluding articulated objects. In *Proceedings International Conference on Computer Vision*, pages 612–617, 1995.

[32] J. Rissanen. Stochastic complexity and modeling. *Annals of Statistics*, 14,1080-1100, 1986.

[33] R. Rosales. *The Specialized Mappings Architecture, with Applications to Vision-Based Estimation of Articulated Body Pose*. PhD thesis, Boston University, 2002.

[34] R. Rosales, V. Athitsos, L. Sigal, and S. Sclaroff. 3d hand pose estimation using specialized mappings. In *Proceedings International Conference on Computer Vision*, pages 378–387, 2001.

[35] R. Rosales and S. Sclaroff. Learning body pose using specialized maps. In *Advances in Neural Information Processing Systems 14*, 2001.

[36] N. Shimada, Y. Shirai, Y. Kuno, and J. Miura. Hand gesture estimation and model refinement using monocular camera - ambiguity limitation by inequality constraints. In *Proceedings International Conference on Automatic Face and Gesture Recognition*, pages 268–273, 1998.

[37] L. Sigal, S. Sclaroff, and V. Athitsos. Estimation and prediction of evolving color distributions for skin segmentation under varying illumination. In *Proceedings Computer Vision and Pattern Recognition*, pages 152–159, 2000.

[38] C. Sminchisescu and B. Triggs. Covariance scaled sampling for monocular 3d body tracking. In *Proceedings Computer Vision and Pattern Recognition*, pages 447–454, 2001.

[39] Y. Song, X. Feng, and P. Perona. Towards detection of human motion. In *Proceedings Computer Vision and Pattern Recognition*, pages 810–817, 2000.

[40] Camillo J. Taylor. Reconstruction of articulated objects from point correspondences in a single uncalibrated image. *Computer Vision and Image Understanding: CVIU*, 80(3):349–363, December 2000.

[41] Virtual Technologies, Inc., Palo Alto, CA. *VirtualHand Software Library Reference Manual*, August 1998.

[42] C. Wren, A. Azarbayejani, T. Darrell, and A. Pentland. Pfinder: Real time tracking of the human body. *PAMI*, 19(7):780-785, 1997.