Combining generative and discriminative models in a framework for articulated pose estimation

Rómer Rosales*
Computer Science and
Artificial Intelligence Laboratory
Massachusetts Institute of Technology
Cambridge, MA 02139 USA
romer@csail.mit.edu

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

Abstract

We develop a method for the estimation of articulated pose, such as that of the human body or the human hand, from a single (monocular) image. Pose estimation is formulated as a statistical inference problem, where the goal is to find a posterior probability distribution over poses as well as a maximum a posteriori (MAP) estimate. The method combines two modeling approaches, one discriminative and the other generative. The discriminative model consists of a set of mapping functions that are constructed automatically from a labeled training set of body poses and their respective image features. The discriminative formulation allows for modeling ambiguous, one-to-many mappings (through the use of multi-modal distributions) that may yield multiple valid articulated pose hypotheses from a single image. The generative model is defined in terms of a computer graphics rendering of poses. While the generative model offers an accurate way to relate observed (image features) and hidden (body pose) random variables, it is difficult to use it directly in pose estimation, since inference is computationally intractable. In contrast, inference with the discriminative model is tractable, but considerably less accurate for the problem of interest. A combined discriminative/generative formulation is derived that leverages the complimentary strengths of both models in a principled framework for articulated pose inference. Two efficient MAP pose estimation algorithms are derived from this formulation; the first is deterministic and the second non-deterministic. Performance of the framework is quantitatively evaluated in estimating articulated pose of both the human hand and human body.

Keywords: Human body pose, hand pose, nonrigid and articulated pose estimation, statistical inference, generative and discriminative models, mixture models, Expectation Maximization algorithm.

*Most of this work was done while the first author was with Boston University. Current e-mail: romer.rosales@siemens.com

1 Introduction

An essential task for vision systems is to infer or estimate the state of the world given some form of visual observations. From a computational/mathematical perspective, this typically involves facing an ill-posed problem; relevant information is lost via projection of the three-dimensional world into a two-dimensional image. In this paper, the focus is on inferring the pose of an articulated object in an image, in particular the pose of a human body or human hand. Humans can often solve such pose inference problems, even when given only a relatively poor-quality, low-resolution, monocular image. It is believed that humans employ extensive prior knowledge about human body structure and motion in solving this ill-posed task [23]. In this paper, we consider how a computer vision system might learn such knowledge in the form of probabilistic models, and how to employ such models in an algorithm for reliable pose inference.

For purposes of computation, the estimation/inference task can be defined as follows: given an observation vector $\mathbf{x} \in \Re^c$ that was extracted from an image of a person, estimate the parameterized articulated pose as a vector $\mathbf{h} \in \Re^t$. The cue and target vector spaces $\Re^c$ and $\Re^t$ are continuous. Generally speaking, using a machine learning approach, this task may be regarded as a function $\varphi : \Re^c \rightarrow \Re^t$ that maps an input vector of visual observations to an output vector describing the *best* articulated pose; we refer to this task as (MAP) estimation. More generally, in probabilistic inference, the mapping function could produce a posterior probability distribution, $\varphi : \Re^c \rightarrow \mathcal{P}$, where $\mathcal{P}$ is a family of probability density functions on $\Re^t$. Note that in general, solving the inference problem does not imply solving the MAP estimation problem. A number of general questions arise. What form should the mapping function φ take? How can the mapping function be estimated from training data? How can the approach incorporate prior knowledge about the problem structure? How can approximate inference be performed efficiently and accurately if exact inference is intractable? These questions are fundamental and common in statistical learning, and only in limited instances are the answers immediately clear (*e.g.*, see [32]).

Several perspectives or principles could be employed to approach learning tasks. Often it is not clear which is more suitable for the problem at hand. It will be useful for the purpose of this paper to distinguish two perspectives: the generative and the discriminative learning perspectives (*e.g.*, see [29]). In the case of learning generative models, a joint distribution $p(\mathbf{x}, \mathbf{h})$ over the random variables of the model (for simplicity consider only $\mathbf{h}$ and $\mathbf{x}$) is estimated from data. Then, given an observation, *e.g.*, $\mathbf{x}$, a posterior probability $p(\mathbf{h}|\mathbf{x})$ over the unobserved random variables could, in theory[1], be calculated by invoking Bayes rule. In contrast, using discriminative models the posterior distribution $q(\mathbf{h}|\mathbf{x})$ is directly built or learned

[1]However, in practice this task can be intractable or lack analytic solutions. This is an important problem in statistics.

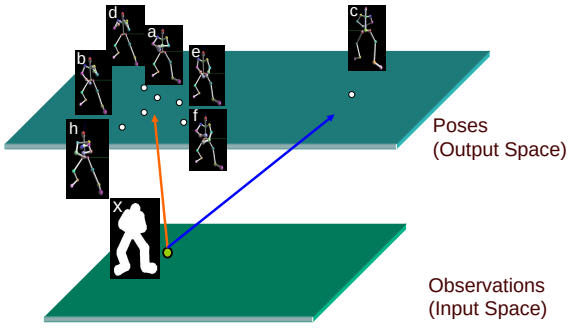


Figure 1: Example ambiguity in mapping body silhouette cues in $\Re^c$ to articulated body poses in $\Re^t$. Given silhouette $\mathbf{x}$, poses $\mathbf{a}$–$\mathbf{h}$ are all valid hypotheses. In general, entire regions in $\Re^t$ may contain valid poses.

(see *e.g.*, [27, 38, 29] for further comparisons between these viewpoints). In this paper, we favor the idea that for pose estimation, the advantages of each of these viewpoints could be exploited in a single framework.

If we try to learn a mapping directly, let us say by estimating the parameters of a parameterized function $\phi : \Re^c \rightarrow \Re^t$ as in a discriminative approach, we encounter several problems. The form required for ϕ may not be simple, because the mapping from observations to articulated poses is generally ambiguous (one-to-many), and therefore no single function may perform this mapping. An example is illustrated in Fig. 1; the arm locations cannot be uniquely inferred given the silhouette $\mathbf{x}$ and therefore, $\mathbf{a}$–$\mathbf{h}$ are all plausible pose configurations. The hands and arms can move in such a way that the silhouette does not change. Note also that pose $\mathbf{c}$ is the reflection of $\mathbf{a}$: the camera looks at the back rather than at the front of the body. There may be different regions in $\Re^t$ that correspond to ranges of valid poses, and these regions may not be connected; *e.g.*, , some viewed from the front and others from behind. An alternative, to be used in this paper, is to model this image-pose relationship using multimodal distributions. Our approach will allow us to keep this discriminative model simple.

Let us now consider the *inverse* problem: given an articulated pose vector $\mathbf{h}$, generate its silhouette $\mathbf{x}$. With a computer graphics model of the human body, one can easily render the silhouette $\mathbf{x}$. Thus, using computer graphics we can build a function $\zeta : \Re^t \rightarrow \Re^c$ (mapping pose parameters to image features) that can be employed to define a generative model. This will play an important role in developing the framework presented in this paper. Note that this generative process is not necessarily a one-to-one mapping, even for given camera parameters, because of noise, clothing, anthropometric variations, etc.; nonetheless, it provides an acceptable approximation in practice. While the inverse mapping ζ provides very useful information about the structure of the problem, unfortunately it cannot be incorporated easily in a discriminative

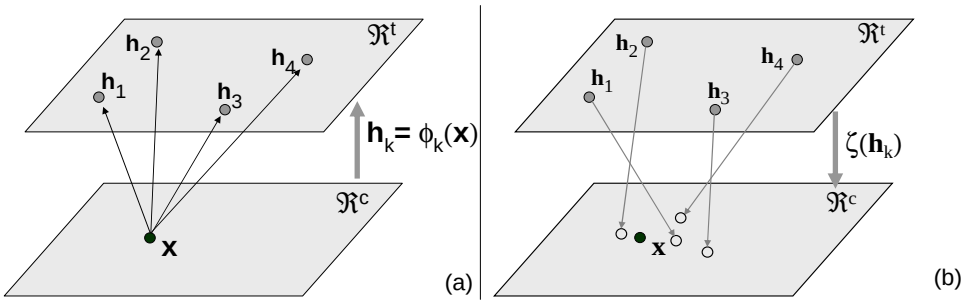


Figure 2: Simplified graphical illustration of our method for estimating body pose (deterministic algorithm): (a) given an input vector $\mathbf{x}$, we generate a set of hypotheses, (b) the inverse mapping function ζ is employed in evaluating each hypothesis.

approach. Despite how simple it is to evaluate ζ, its inverse may still be complex or may not exist. In other words, inferring $\mathbf{h}$ from $\mathbf{x}$ may be difficult.

In summary, the one-to-many nature of the problem of mapping image features to body poses precludes the use of discriminative supervised learning methods that fit a single or a finite number of functions to the data; *e.g.*, , neural networks regression, support vector machine regression, boosting, etc. On the other hand, we have access to $\zeta : \Re^t \rightarrow \Re^c$ that given a body pose can produce the corresponding image features, which can be used to define a very accurate generative model. However, as will be shown later, this accurate generative model is challenging to use directly in body pose inference. The view taken in this paper is that it can be effective to use the individual advantages of these two complimentary approaches (discriminative and generative) to formulate an efficient solution to the pose inference/learning problem.

The paper is structured as follows. Sec. 2 presents the related work and how our work fits in with, and differs from existing approaches for pose estimation. Sec. 3 starts by proposing independent discriminative and generative models for the problem at hand without explicitly creating a connection between them. Sec. 4 assumes that these two models are given and introduces inference. First inference is presented for each model separately and its shortcomings discussed, then a method that combines both models is introduced. Sec. 4.3 presents the general formulation, while Sec. 4.4 and 4.5 concentrate on algorithms. Sec. 5 shows how to learn the proposed models. Sec. 6 describes the applications considered and Sec. 7 presents the results of the experimental evaluation. Sec. 8 provides a discussion of strengths and limitations of the proposed approach, conclusions, and directions for future work.

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 [6, 11, 14, 30, 33, 39, 12, 41]. Mesh models have also been used as an alternative to link-joint models [16]. At each frame, these geometric models are fitted to the image to minimize some cost function that favors the overlap of the model and associated image regions (or motion). Although usually not stated, the fitting or cost function in many cases implicitly defines (or can be used to define) a generative model of the observed image. 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; this can sometimes be equivalent to MAP estimation with a complex generative model. 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 [22, 30, 31]. 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], where a subspace of allowable motions is learned from a set of examples. These examples and the model (usually linear) are expected 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 also limits the generalization power to motions very similar from those seen in the training set. The underlying process to be modeled is generally non-linear given the representations that are commonly used. We believe this process cannot be effectively 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 [19] a statistical approach is employed to reconstruct 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 [42] 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. This formulation implicitly restricted the probability model to the class of distributions defined by graphical models with *tree-width* equal to three; thus, inference was computationally feasible [24, 32]. Still, in this approach, the image points and model initialization must be provided by hand or through some other method.

In [5], the manifold of human body dynamics is modeled 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 just referred, 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 computational resources required can be impractical. As a consequence, models with unacceptably large priors towards specific motions are generated. Although by not modeling 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. In addition, it is not as sensitive to temporary frame errors (*e.g.*, dropped frames). This happens simply because this model treats configurations as temporally independent of each other. Other approaches that use a single image include [3, 15, 43]; however, most of these methods also require that projected joint locations be given as input. In the approach presented in this paper, this is not necessary.

The approach introduced is simple and also practical. It can be described as that of mapping visual features to likely body configurations and can be, roughly speaking, summarized as follows: At learning time, several functions that map visual features to pose parameters are approximated from training data employing a machine learning paradigm. A unique aspect of our approach is the combined use of (1) these mapping functions (defining a discriminative model) with (2) the inverse mapping function ζ (defining a generative model). At inference time, after multiple poses have been found using each of the above functions from just the input visual features, then ζ transforms these pose configurations back to the visual feature (observation) space. In this space, we can then automatically *choose* among a set of *reconstruction* hypotheses according to a criterion of interest (see Fig. 1). Our approach avoids the need for manual initialization or tracking; it thereby avoids the consequent disadvantages of tracking. Remarkably, relatively few computations are

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	$\psi_i \in \Re^t$
generative and discriminative models probability distributions	p,q (respectively)
observation random variable (*e.g.,* image moments)	$\mathbf{x} \in \Re^c$
hidden random variable of pose parameters	$\mathbf{h} \in \Re^t$
inverse (rendering) function (for generative model)	$\zeta : \Re^t \rightarrow \Re^c$
number of samples during inference	S
a particular observation or input image feature	$\mathbf{x}^*$
output (pose) hypothesis (a sample from $q(\mathbf{h}\|\mathbf{x}^*)$)	$\mathbf{h}_k$
estimate of most likely output hypothesis	$\hat{\mathbf{h}}$
discrete set of labels for mixture components	$\mathcal{C} = \{1, \ldots, M\}$
hidden random variables assigning mixture component to training samples	$\mathbf{y} = (y_1, \ldots, y_N), y_i \in \mathcal{C}$
prior probability that mixture component k will be used	$\lambda_k = Q(y = k)$
mapping function parameter vector	θ_k
discriminative model parameters (to be learned)	$\theta = (\theta_1, \ldots, \theta_M, \lambda_1, \ldots, \lambda_M)$
posterior probability of k-th mixture component for $\mathbf{z}_i$ during EM	$\tilde{Q}(y_i = k) = Q(y_i = k\|\psi_i, \upsilon_i, \theta)$

Table 1: Some mathematical symbols used in this paper.

required for inference. We will now formalize and explain our approach in detail.

3 Probabilistic Models

We propose a probabilistic, nonlinear method for combining generative and discriminative models for articulated pose estimation. The framework employs a set of M functions $\phi_k : \Re^c \rightarrow \Re^t$, each associated to a mixture component in a mixture distribution; together, these functions are able to approximate one-to-many mappings. In our approach, the functions are jointly estimated automatically from training data via a variant of the Expectation-Maximization algorithm. The learned conditional distribution over the output space is then used as an approximation to that implied by a more accurate model defined with the help of the inverse function ζ (the generative model), for which inference is intractable. This basic idea is shown in a schematic way in Fig. 1. The approximation is employed in a similar way as a proposal distribution is used to approximate sampling from a more complex distribution.

We begin by formally defining both the discriminative and generative models to be employed. The discriminative model will be estimated from training data and the generative model will be defined by a rendering function ζ. These models represent two views of the same problem and will be used together in our framework.

3.1 The Discriminative Model

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 this formulation is provided in Table 1.

We will approach our discriminative 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 domain the discrete set $\mathcal{C} = \{1, \ldots, M\}$ of labels for the specialized functions, and can be thought of as the function index used to map the i-th training pair, $\mathbf{z}_i$. Thus M is the number of specialized functions. Our model uses parameters $\theta = (\theta_1, \ldots, \theta_M, \lambda_1, \ldots, \lambda_M)$, where θ_k represents the parameters of the k-th mapping function, and is the prior probability that the k-th mapping function will be used to map an input-output pair.

Assuming independence of observations given θ, we seek to maximize the sum of conditional log-probabilities:

$$\theta^* = \arg\max_\theta \sum_i \log q(\psi_i|\nu_i, \theta) \quad (1)$$

$$= \arg\max_\theta \sum_i \log \sum_k q(\psi_i|\upsilon_i, y_i = k, \theta), \quad (2)$$

Due to the sum of terms inside the logarithm of Eq. 2, this optimization is computationally costly for large M. However, a variety of practical approximate optimization methods exist, for example, methods that are based on alternating optimizations [9]. Expectation Maximization (EM) [10, 28] updates are described in Sec. 5.

3.1.1 Choice of a Likelihood Function

Note that the above formulation is general. In particular, the form of the probability $q(\psi|\upsilon, y, \theta)$ was not specifi ed. A key question in instantiating our approach is: what form should $q(\psi|\upsilon, y, \theta)$ take? This is, the probability that output ψ was generated by the 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: $q(\upsilon, \psi|y, \theta) = \mathcal{N}((\upsilon, \psi); \mu_y, \Sigma_y)$.

2. A Gaussian distribution, whose mean is the output of the y-th mapping function: $q(\psi|\upsilon, y, \theta) = \mathcal{N}(\psi; \phi_y(\upsilon, \theta), \Sigma_y)$.

This formulation can accept other forms for the likelihood function.

3.2 The Generative Model

Our approach also involves the use of a generative model of images (or image features). In the problem of human body pose estimation from a single image this generative model can be defined in a simple way. We will assume that an image or image features are generated by sampling a pose from a prior distribution $p(\mathbf{h})$ and then generating an image using the rendering function ζ such that:

$$p(\mathbf{x}|\mathbf{h}) = \mathcal{N}(\mathbf{x}; \zeta(\mathbf{h}), \Sigma_\zeta). \tag{3}$$

It is important to notice that despite the fact that the generative model can be defined in a simple manner, the function ζ is of a complex form. In our case, this makes probabilistic inference intractable as will be explained later.

4 Inference

In this section, we refer to probabilistic inference as finding a full probability distribution for $\mathbf{h}$ once an observation $\mathbf{x} = \mathbf{x}^*$ has been made (some image features were observed).

4.1 Inference using the Discriminative Model Alone

A valid approach to estimating/inferring $\mathbf{h}$ is to use the discriminative model alone. In our context, inference involves finding a full probability distribution for $\mathbf{h}$ given $\mathbf{x}^*$; the discriminative model directly provides this expression.

In MAP estimation we just have to maximize this expression. That is, we want to find the most probable output $\mathbf{h} \in \Re^t$ for a given observation $\mathbf{x}^* \in \Re^c$: $\hat{\mathbf{h}} = \arg\max_{\mathbf{h}} q(\mathbf{h}|\mathbf{x}^*) = \arg\max_{\mathbf{h}} \sum_y q(\mathbf{h}|\mathbf{x}^*, y)Q(y)$, where $q(\mathbf{h}|\mathbf{x}^*)$ is a shorthand for $q(\mathbf{h}|\mathbf{x} = \mathbf{x}^*)$. 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 $q(\mathbf{h}|\mathbf{x}, y) = \mathcal{N}(\mathbf{h}; \phi_y(\mathbf{x}), \Sigma_y)$. Thus, in either case we have that $q(\mathbf{h}|\mathbf{x}^*)$ is a mixture of Gaussians and if we want to find the MAP estimate we need to solve: $\hat{\mathbf{h}} = \arg\max_{\mathbf{h}} \sum_y \mathcal{N}(\mathbf{h}; \phi_y(\mathbf{x}^*), \Sigma_y)Q(y)$.

This result was obtained by employing the MAP principle using our discriminative model alone. Here we have assumed that we know the model. In practice we need to estimate or *learn* it (learning will be covered in the next section), but in general, $q(\mathbf{h}|\mathbf{x})$ will usually be an approximation to the true conditional distribution, obtained using the training data. Even though we could simply adopt the above MAP estimate

as a solution, it should not be surprising that we could improve upon this by using our knowledge of p, the generative model.

4.2 Inference Using the Generative Model Alone

Using the generative model, inference involves finding the posterior $p(\mathbf{h}|\mathbf{x}=\mathbf{x}^*)$ ($p(\mathbf{h}|\mathbf{x}^*)$ as a shorthand):

$$p(\mathbf{h}|\mathbf{x}^*) = \frac{1}{p(\mathbf{x}^*)}p(\mathbf{x}^*|\mathbf{h})p(\mathbf{h}) = \frac{1}{Z_p}\mathcal{N}(\mathbf{x}^*;\zeta(\mathbf{h}),\Sigma_\zeta)p(\mathbf{h}) \quad (4)$$

$$Z_p = \int \mathcal{N}(\mathbf{x}^*;\zeta(\mathbf{h}),\Sigma_\zeta)p(\mathbf{h})d\mathbf{h}. \quad (5)$$

There are however at least two difficult obstacles for achieving this:(1) The integral in Eq. 5 cannot be solved easily and (2) we do not have an expression for $p(\mathbf{h})$.

In MAP estimation we do not need to be concerned about obstacle (1) since in MAP the goal is to find $\hat{\mathbf{h}} = \arg\max_{\mathbf{h}} p(\mathbf{h}|\mathbf{x}) = \arg\max_{\mathbf{h}} \mathcal{N}(\mathbf{x};\zeta(\mathbf{h}),\Sigma_\zeta)p(\mathbf{h})$ because Z_p is a constant with respect to this optimization problem. However, solving for $\hat{\mathbf{h}}$ given the observed $\mathbf{x}^*$ is a daunting task; the space of $\mathbf{h}$ is too large to explore exhaustively and $\zeta(\mathbf{h})$ too complex to apply standard directed search techniques adequately. If we could start the search using a point $\mathbf{h}_0$ that we knew was close enough to the best $\mathbf{h}$ then this problem could be mitigated. This idea is often employed in solving tracking problems, *i.e.*, when we have close enough frames (in time and space) and the previous frame estimate(s) can be trusted. However, the goal in this paper is to solve for pose from a single image, and so tracking is not possible.

Both obstacles could be overcome if somehow we could accurately obtain samples distributed according to $p(\mathbf{h}|\mathbf{x})$. Those samples could be used to (1) approximate this posterior and (2) find the sample with highest probability and use it as a MAP estimate or as an initial point to search for a better estimate. However, we cannot even evaluate $p(\mathbf{h}|\mathbf{x})$ (otherwise the inference problem would have been solved) and, in addition, accurately sampling from a given distribution is an open problem in general [26].

4.3 Combining Generative and Discriminative Models. Importance Sampling

In general, sampling can be used to estimate expectations of a given function $I(\mathbf{h})$ with respect to some probability density $\pi(\mathbf{h})$ that we can evaluate at any point, but from which we cannot sample. Let us say we need to calculate the integral $\mathcal{I} = \int \pi(\mathbf{h})I(\mathbf{h})d\mathbf{h}$, by approximating $\mathcal{I}$ employing S samples: $\hat{\mathcal{I}} = \frac{1}{S}\sum_{r=1}^{S} I(\mathbf{h}^{(s)})$. Let $p(\mathbf{h}|\mathbf{x}^*)$ correspond to $\pi(\mathbf{h})$ ($I(\mathbf{h})$ can be any function of the pose), but note we cannot evaluate $p(\mathbf{h}|\mathbf{x}^*)$. However, in the importance sampling method, it is only necessary to evaluate the

distribution up to a multiplicative factor. It turns out that in our problem we can evaluate the joint $p(\mathbf{h}, \mathbf{x}^*)$ which is enough since it is proportional to $p(\mathbf{h}|\mathbf{x}^*)$.

The question is how to appropriately generate the samples to obtain the best estimate. In the importance sampling method we first come up with a proposal distribution $\pi'(\mathbf{h})$, which we can also evaluate but from which it is possible to sample accurately; then we sample from $\pi'(\mathbf{h})$, but also correct for the bias introduced when sampling, obtaining:

$$\hat{\mathcal{I}} = \frac{1}{S} \sum_{r=1}^{S} \frac{p(\mathbf{h}^{(s)}, \mathbf{x}^*)}{\pi'(\mathbf{h}^{(s)})} I(\mathbf{h}^{(s)}). \tag{6}$$

It can be shown that when $S \to \infty$, $\sqrt{S}(\hat{\mathcal{I}} - \mathcal{I}) \sim \mathcal{N}(0, \sigma^2_{\pi'})$, with: $\sigma^2_{\pi'} = \int (\frac{p(\mathbf{h}^{(s)}, \mathbf{x}^*)}{\pi'(\mathbf{h})} I(\mathbf{h}) - \mathcal{I})^2 \pi'(\mathbf{h}) d\mathbf{h}$. Thus, the expected variance of our estimate is proportional to $\sigma^2_{\pi'}$ and inversely proportional to S [26]. Since minimizing variance is a reasonable criterion to consider, we would like to know what the optimal proposal distribution π' is in terms of minimizing the estimate variance $\sigma^2_{\pi'}$ for a fixed S. The optimal proposal distribution is given by a result in [37, 7]:

$$\pi'(\mathbf{h}) = p(\mathbf{h}, \mathbf{x}^*) \int p(\mathbf{h}, \mathbf{x}^*) d\mathbf{h}, \tag{7}$$

which is equal to $p(\mathbf{h}|\mathbf{x}^*)$.

This makes sense in our simple case (for a general proof, see [37]), since this is the distribution in the initial integral we wanted to solve. One would expect that in the limit of infinite samples, the best estimate for $\mathcal{I}$ whatever the function I is, should be obtained when sampling from the exact distribution involved in the integral. Of course, we know that in our case we cannot sample from it. However, now we know (1) that from an importance sampling perspective, we should sample from $p(\mathbf{h}|\mathbf{x}^*)$ to minimize variance, which is a reasonable criterion, and also (2) that in this result there is no reference to the explicit $p(\mathbf{h})$.

In this paper, the main reason behind using generative and discriminative models together is to tackle this particular problem of sampling from a good distribution. We will use the learned distribution $q(\mathbf{h}|\mathbf{x})$ (the discriminative model) to approximate $p(\mathbf{h}|\mathbf{x})$, but just at $\mathbf{x} = \mathbf{x}^*$. As we will see in the next section, we will build this approximation employing the maximum likelihood principle. Given the well known relationship between ML and KL divergence, this can also be seen as finding a discriminative (conditional) distribution q that is close to the sampled conditional p distribution (empirical distribution) in terms of the KL divergence [2] (see appendix for details).

4.4 Non-deterministic MAP Estimation: Multiple Samples (MS)

We are usually interested in providing likely samples from the posterior distribution, in particular we might be interested in the most likely $\mathbf{h}$. This is the idea behind MAP estimation, where we are interested in finding $\hat{\mathbf{h}} = \arg\max_{\mathbf{h}} p(\mathbf{h}|\mathbf{x}^*) = \arg\max_{\mathbf{h}} p(\mathbf{x}|\mathbf{h}^*)p(\mathbf{h})$.

We know that the discriminative model distribution $q(\mathbf{h}|\mathbf{x})$ tries to approximate $p(\mathbf{h}|\mathbf{x})$, and therefore it is good at minimizing the variance of the estimator. Due to this, we will use the discriminative model distribution to provide samples for MAP estimation. In MAP estimation, we sample $\mathcal{H}_{Spl} = \{\mathbf{h}_s\}_{s=1...S}$ using the proposal distribution $q(\mathbf{h}|\mathbf{x}^*)$. Given the samples, the problem the becomes a discrete optimization problem that can be solved easily (see Fig. 3 for pseudo-code):

$$\hat{s} = \arg\max_s p(\mathbf{x}^*|\mathbf{h}_s) = \arg\min_s (\mathbf{x}^* - \zeta(\mathbf{h}_s))^\top \Sigma_\zeta (\mathbf{x}^* - \zeta(\mathbf{h}_s)), \tag{8}$$

by using the Gaussian form of $p(\mathbf{x}|\mathbf{h})$ as given in Eq. 3. We remark that one can use the samples $\mathcal{H}_{Spl}$ as starting points to other more sophisticated methods. For example we could use Markov chain Monte Carlo (MCMC) sampling [26, 46] to search for regions of higher probability. Also, instead of stochastic methods, we could employ standard gradient descent methods to locally search for more likely poses $\mathbf{h}$ (as in tracking). These methods may be helpful for some distributions but in general have several drawbacks: (1) They are usually very slow in high dimensions and (2) given finite time, they are not very useful/accurate if the posterior probability is very complex. Some methods have been proposed to alleviate these problems, but this goes beyond our current contribution. Keeping this extension in mind, in this paper we simply use the original samples $\mathcal{H}_{Spl}$ to search for a MAP estimate. These estimates proved to be sufficiently accurate during our experiments.

4.5 Deterministic MAP Estimation: Mean Output (MO)

In certain applications, it might be advantageous to count with a very fast method for computing MAP estimates. Two examples are: when working with multiple articulated bodies and in dynamic or on-line settings where it is necessary to provide estimates at high rates. Even though the time complexity of MS scales linearly with the number of samples, this might not be fast enough. Motivated by speed constraints, here we propose a very fast and simple MAP estimation algorithm that still performs well in experiments. Unlike MS, this algorithm is deterministic.

The structure of the problem, as well as the form of the discriminative distribution components (*i.e.*, conditioned on the mixture label) $q(\mathbf{h}|\mathbf{x}, y)$ employed (Gaussian), make it possible to construct this deterministic approximation. The basic intuition is straightforward. For a given $\mathbf{x} = \mathbf{x}^*$, we *ask* each mapping

function ϕ_k to give its most likely estimate for $\mathbf{h}$. We then evaluate the probability of each function's estimate via the generative model distribution $p(\mathbf{x}^*|\mathbf{h})$. From our experiments, we believe this approximation is good in practice.

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 of the random variable. Formally, in both Case (1) and Case (2) described earlier, $q(E[\mathbf{h}|\mathbf{x}^*, y, \theta]|\mathbf{x}^*, y, \theta]) \geq q(\mathbf{h}'|\mathbf{x}^*, y, \theta)$, 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 $\forall y, \max_k q(\mathbf{h}_k^\phi|\mathbf{x}^*, y) \geq \max_s q(\mathbf{h}_s|\mathbf{x}^*, y)$, simply by setting $\mathbf{h}_k^\phi = \phi_k(\mathbf{x}^*, \theta)$.

This basic 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 mapping function (*i.e.*, each mixture component in the discriminative model), given the input. We expect the discriminative model to be a good approximation of our generative model posterior distribution as discussed above. However, in general the MO approximation need not be very accurate. The smaller the overlap among the distributions associated with each function, the better the accuracy of this approximation (this in turn depends on the means and covariances of the mixture components).

In MO approximate inference, the expression to be minimized is the same as that used in Eq. 8, except for the use of the M means instead of the S samples (see Fig. 3 for pseudo-code):

$$\hat{k} = \arg\max_{k \in \mathcal{C}} p(\mathbf{x}^*|\mathbf{h}_k^\phi) = \arg\min_{k \in \mathcal{C}} (\mathbf{x}^* - \zeta(\mathbf{h}_k^\phi))^\top \Sigma_\zeta (\mathbf{x}^* - \zeta(\mathbf{h}_k^\phi)). \tag{9}$$

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

5 Learning

In order to learn the discriminative model 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. 2. Since the EM algorithm is well known [10, 2, 28], we will only provide derivations specifi c to our formulation.

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

Summary of MAP Estimation Algorithms

Inputs: visual features $\mathbf{x}^*$ computed from single image, generative (p), and discriminative (q) models.

- MO Algorithm

  1. For each $k = 1, ..., |\mathcal{C}|$ (each function ϕ_k)
     (a) Compute $\mathbf{h}_k^\phi = \phi_k(\mathbf{x}^*)$ using the trained discriminative model
     (b) Compute $p(\mathbf{x}^*|\mathbf{h}_k^\phi)$ using the generative model by rendering from $\mathbf{h}_k^\phi$ (*i.e.,* apply ζ or $\hat{\zeta}$ to $\mathbf{h}_k^\phi$)

  Output: MAP estimate $\hat{\mathbf{h}} \leftarrow$ pick the $\mathbf{h}_k^\phi$ that maximizes $p(\mathbf{x}^*|\mathbf{h}_k^\phi)$ over k (use Eq. 10)

- MS Algorithm (extra input required: number S of samples)

  1. Generate S samples $\mathbf{h}_s$ from $q(\mathbf{h}|\mathbf{x}^*)$
  2. For each $s = 1, ..., S$
     (a) Compute $p(\mathbf{x}^*|\mathbf{h}_s)$ using the generative model by rendering from $\mathbf{h}_s$ (*i.e.,* apply ζ or $\hat{\zeta}$ to $\mathbf{h}_s$)

  Output: MAP estimate $\hat{\mathbf{h}} \leftarrow$ pick the $\mathbf{h}_k$ that maximizes $p(\mathbf{x}^*|\mathbf{h}_k)$ over s (use Eq. 9)

Figure 3: Summary of MO and MS algorithms for MAP estimation.

denote this posterior $Q(y_i = k|\psi_i, \upsilon_i, \theta)$ using the shortcut notation $\tilde{Q}^{(t)}(y_i = k)$. We then have:

$$\tilde{Q}^{(t)}(y_i = k) = \lambda_k q(\psi_i|\upsilon_i, y_i = k, \theta^{(t-1)}) / \sum_{j \in \mathcal{C}} \lambda_j q(\psi_i|\upsilon_i, y_i = j, \theta^{(t-1)}). \tag{10}$$

Stated differently, this step estimates the responsibility of each mapping function, ϕ_k for each data point, $\mathbf{z}_i$. $\tilde{Q}^{(t)}(y_i = k)$ represents the so called responsibility of function k for data pair i. Also recall that $\lambda_k = Q(y_i = k)$ is the prior probability that function k be used.

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

$$\theta^{(t)} = \arg\max_\theta \sum_i \sum_{k \in \mathcal{C}} \tilde{Q}^{(t)}(y_i = k)[\log q(\psi_i|\upsilon_i, y_i = k, \theta) + \log Q(y_i = k|\theta)]. \tag{11}$$

We now present solutions for the cases described above.

5.1 Case (1)

In this case we have:

$$q(\upsilon, \psi|y, \theta) = \mathcal{N}(\upsilon, \psi; \mu_y, \Sigma_y) = \mathcal{N}(\begin{bmatrix} \upsilon \\ \psi \end{bmatrix}; \begin{bmatrix} \mu_\upsilon \\ \mu_\psi \end{bmatrix}, \begin{bmatrix} \Sigma_{\upsilon\upsilon} \Sigma_{\upsilon\psi} \\ \Sigma_{\upsilon\psi}^\top \Sigma_{\psi\psi} \end{bmatrix})_y, \tag{12}$$

where the subscript y is simply the mapping function number. We can show that the parameter learning problem is reduced to a mixture of Gaussian estimation, for which it is straightforward to estimate θ using EM. Moreover, the probability of ψ given an observed υ is also Gaussian: $q(\psi|\upsilon, y, \theta) = \mathcal{N}(\psi; \mu_\psi + \Sigma_{\upsilon\psi}^\top \Sigma_{\upsilon\upsilon}^{-1}(\upsilon - \mu_\upsilon), \Sigma_{\psi\psi} - \Sigma_{\upsilon\psi}^\top \Sigma_{\upsilon\upsilon}^{-1} \Sigma_{\upsilon\psi})_y$. Therefore in case (1), each function ϕ_k is just the mean of the conditional distribution

$$\phi_k(\upsilon, \theta) = (\mu_\psi + \Sigma_{\upsilon\psi}^\top \Sigma_{\upsilon\upsilon}^{-1}(\upsilon - \mu_\upsilon))_{y=k}. \tag{13}$$

The confi dence 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, note that this expression does not depend on the input, a sometimes undesirable consequence of the model employed. Note also that each function ϕ_k is linear in the input vector from $\Re^c$.

5.2 Case (2)

In this case we have:

$$\frac{\partial E}{\partial \lambda_k} = \sum_i \tilde{Q}^{(t)}(y_i = k) \frac{\partial}{\partial \lambda_k} \log Q(y_i = k|\theta) \tag{14}$$

$$\frac{\partial E}{\partial \Sigma_k} = \sum_i \tilde{Q}^{(t)}(y_i = k) \frac{\partial}{\partial \Sigma_k} \log q(\psi_i|y_i = k, \upsilon_i, \theta_k) \tag{15}$$

$$\frac{\partial E}{\partial \theta_k} = \sum_i \tilde{Q}^{(t)}(y_i = k)[(\frac{\partial}{\partial \theta_k} \phi_k(\upsilon_i, \theta_k))^\top \Sigma_k^{-1}(\psi_i - \phi_k(\upsilon_i, \theta_k))], \tag{16}$$

where E is the cost function that we would like to maximize in Eq. 11.

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:

$$\lambda_k^{(t)} = \frac{1}{N} \sum_i \tilde{Q}^{(t)}(y_i = k) \tag{17}$$

$$\Sigma_k^{(t)} = \sum_i \tilde{Q}^{(t)}(y_i = k)(\psi_i - \phi_k(\upsilon_i, \theta_k))(\psi_i - \phi_k(\upsilon_i, \theta_k))^\top / \sum_i \tilde{Q}^{(t)}(y_i = k) \tag{18}$$

To keep the formulation general, we have not yet defi ned the form of the mapping functions ϕ_k. Whether or not we can fi nd 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 fi nd $\theta_k^{(t)}$. If ϕ_k yields a quadratic form, then a closed form update exists.

Now, regarding our generative model, there is is very little learning involved. If ζ is very accurate, then we could also tell very accurately the image that will be generated given a body pose $\mathbf{h}$. In practice ζ can be defined only approximately. We account for this by appropriately setting Σ_ζ depending of how much noise is expected to be present in the observations. This can also account for inaccuracies in the geometric model.

6 Example Application: Articulated Pose from Visual Features

The formulation presented in this paper is general enough to be applied in a number of supervised learning problems for which the output-to-input (inverse) map is relatively easy to compute; thus allowing us to specify an accurate generative model (but for which inference is difficult). To demonstrate and test our framework, we have developed a system that uses our approach to infer articulated pose from low-level visual features. In particular, we focused on pose estimation of the human hand and body from a single image containing a silhouette of the object. In this class of applications, datasets of poses can be obtained via motion capture gloves or body suits. Computer graphics rendering can then be used to generate the input-output pairs needed for our supervised learning. We will now give details of this demonstration system.

6.1 3D Hand Pose Estimation

The 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 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 [44]. 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.

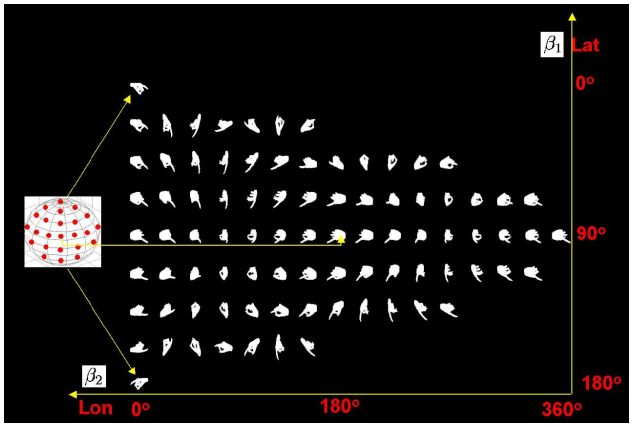


Figure 4: Example of the 86 silhouettes obtained via computer graphics rendering for a given a 3D hand pose. Views are distributed approximately uniformly over the view sphere.

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 subjects. Using computer graphics and an artificial hand model, we then rendered each captured hand pose from multiple viewpoints on the view sphere. 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. 4 shows the 86 viewpoints used in the dataset for a particular configuration.

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 [20] 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, our general 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.

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 [40, 36].

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 represented in terms of marker positions at a predetermined set of joints. We will estimate the 2D positions of these body markers in the image plane.

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, the dataset used by [5], 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. We have chosen Alt moments [1] as our visual features, mainly due to their ease of computation and invariance to translation and scaling. We call this 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 [18, 45]. 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 using maximum-likelihood, where each pixel is classified as belonging to the background or the foreground (human body).

6.3 Common Implementation Details

We now briefly discuss implementation details common to both applications.

6.3.1 Mapping Functions

In Sec. 3, it was not specified what class of (deterministic) mapping functions ϕ_k were to be used. Our framework is practically independent of this choice. However, from Eq. 16 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 power of these functions is limited. In our implementation each function takes the form of a multi-layer perceptron with one hidden layer (MLP); a widely used feedforward neural network architecture. For this non-linear

function there does not exist a closed-form solution for Eq. 16, but it can be seen that the M-step is like a weighted version of backpropagation repeated for each MLP in the mixture. We used four to five iterations of the conjugate gradient descent method per M-step.

6.3.2 Generative Model Details: Inverse Functions

There are at least two ways to define this function. On 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 ideas. For ζ, we used computer graphics renderings of our hand and body models obtained via OpenGL. For $\hat{\zeta}$, we used a one-layer MLP, with twenty hidden nodes (however, the method is overall independent of the functional form chosen). In our experience, this provided an adequate and efficient approximation for our dataset.

The approximate inverse 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 inverse mapping is assumed to be simple (one-to-one or even many-to-one) or that it has a known form, otherwise if we assume too simple functional forms, we would only introduce more estimation errors. In our case, this is just a practical issue. If the inverse mapping is too complex to approximate easily, we could always rely on the available inverse 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 MLPs with five hidden nodes each. The system can infer body poses at approximately 11 frames per second, using the Mean Output (MO) algorithm. This approach's 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 inverse 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 our approach in estimating the pose of the human hand and body. For many additional performance experiments not included due to space limitations, the reader is referred to [35] and for several MO estimation videos to http://www.csail.mit.edu/∼romer/DGHandVideos.htm. The application independent Matlab code can be found at http://www.csail.mit.edu/∼romer/DGCode.htm.

7.1 Hand Pose Estimation Given a Fixed Camera Viewpoint

In our first experiments, our approach 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).

The number of mixture components for the discriminative model (mapping functions) was set to 20. This number was found to be optimal in the sense of the Minimum Description Length (MDL) principle [34]; we found this number via a rough model search (testing MDL and getting the score for the optimized model with 10,12,...,24 functions). Each mapping function (for each of the Gaussians in the mixture) was a MLP with seven hidden nodes.

7.1.1 Quantitative Results

We randomly selected approximately 4,000 frames not included in the training set. Since ground-truth is available, we used the average absolute difference per joint angle (between ground-truth and estimate) as error measure. Table 2 summarizes our results (see caption).

	MO-MAP ($\hat{\zeta}$)	MS-MAP ($\hat{\zeta}$)	MS-20 ($\hat{\zeta}$)	MO-MAP (ζ)	MS-MAP (ζ)	MS-20 (ζ)	Rand/train	Range
$\hat{\mathcal{E}}$	0.1322	0.1667	0.1465	0.1651	0.1769	0.1785	0.4294	1.55
$\sigma^2_{\hat{\mathcal{E}}}$	0.0317	0.0415	0.0371	0.0425	0.0452	0.0547	0.1630	-

Table 2: Mean absolute error $\hat{\mathcal{E}}$ and variance $\sigma^2_{\hat{\mathcal{E}}}$. Inference performance using different rendering functions (ζ and $\hat{\zeta}$) and inference algorithms (MO-MAP and MS-MAP). Also shown, the average error of the 20 most probable reconstructions given by MS (MS-20). Note that the error for MS-20 does not have to be higher than that for MS-MAP. As a point of comparison, results are presented for an algorithm that randomly chooses one of the training examples as result (Rand/train). The average range of the data is also shown as a reference point. All units are in radians.

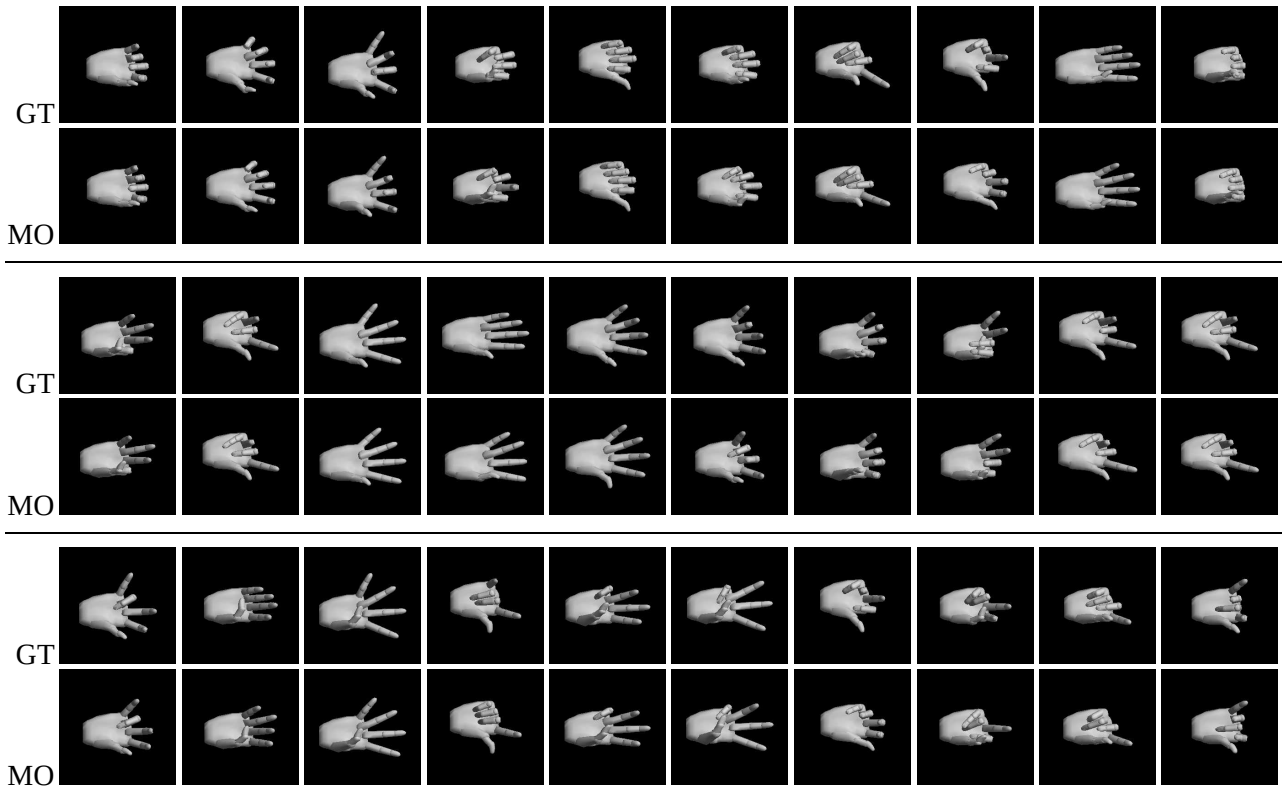


Figure 5: 40 examples of estimated hand poses chosen uniformly at random. Reconstruction found using the Mean Output (MO) approach. The inverse function used was estimated from data. Each example consists of a pair of images: ground-truth (top), and estimate obtained using the mean output algorithm (bottom).

These experiments quantitatively confirmed that MO inference provides a reasonable approximation, at least for this dataset. Recall from Sec. 4.5 that MO inference was based on the premise that the most-likely reconstruction given by each discriminative mixture component provides a good approximation to the best solution given by the full probability distribution.

Fig. 5 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, 16, 27, etc.

Ambiguous configurations are indeed very common with a binary image representation. Note that in other ambiguous cases shown in Fig. 5 reconstruction is closer to ground truth, *e.g.,* pairs 19, 20, 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 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 point in the manifold.

2. The learned discriminative model was accurate at modeling the given input using a single mixture component; *i.e.,* 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 discriminative model needed to be accurate enough at approximating the true posterior so that samples were relevant at all.

7.1.2 Performance Comparison with Respect to Discriminative Model Alone and a Competing Approach

In this section we experimentally compare our method with the purely discriminative part of the formulation, where no generative model is employed. One can see this test as a way to measure how effectively the generative model disambiguates among poses; thus, illustrating its level of contribution in the overall approach. In addition, for further validation, we also compare our method against the standard MLP, trained using backpropagation to *globally* map image features to 3D poses.

As before we follow the MAP principle to determine the best pose $\hat{\mathbf{h}}$ given input features and a model. Recall from Sec. 4.1 that the MAP estimate is given by $\hat{\mathbf{h}} = \arg\max_{\mathbf{h}} \sum_y \mathcal{N}(\mathbf{h}; \phi_y(\mathbf{x}^*), \Sigma_y) Q(y)$. Since this function is not concave, we used a simple heuristic to choose a maximum. We performed gradient ascent starting at each of the M points $\{\phi_y(\mathbf{x}^*)\}_{y \in \mathcal{C}}$, and set $\hat{\mathbf{h}}$ to the highest point ever reached.

As expected this method performed poorly. The mean absolute error and variance for this dataset were 0.3702 and 0.2117 respectively, just better than randomly choosing a pose from the training set (which pro-

Number of Hidden Nodes	16	**22**	28	34	40	46	52	58	64
$\hat{\mathcal{E}}_{MLP}$	0.2039	**0.1953**	0.1851	0.1784	0.1733	0.1735	0.1792	0.1891	0.2003
$\sigma^2_{\hat{\mathcal{E}}_{MLP}}$	0.0354	**0.0324**	0.0294	0.0280	0.0266	0.0259	0.0341	0.0419	0.0512

Table 3: Performance of the MLP approach as described in the text. The table shows the mean absolute error and variance using the same training/test sets as our method (see Table 2). Each entry in the table shows the average of ten trials. Overall the performance of our method is from 1.09 to 1.48 times better than MLP (corresponding to the worst and best performance from Table 2) using the same number of free parameters, corresponding to 22 hidden nodes in the MLP. Even when the MLP has 46 hidden nodes – requiring over twice as many free parameters – performance of our method is still superior.

vided mean and variance of 0.4294 and 0.1630 respectively). This is not surprising since the discriminative model alone is not designed to "know" what the right mixture component is, given any input presented. More formally, the mixture parameters $Q(\mathbf{y})$ do not depend on the input. The high variance can be attributed to the inconsistent usage of good and bad functions to map the input. The role of the generative model in our approach is essentially that of providing information about what function (mixture component) is appropriate for the given input.

We also compared our full approach against the widely used MLP. Note that unlike above, here we used one MLP in the standard way, that is as a function approximation approach to map input to outputs using the whole training set in the standard backpropagation learning scheme. MLP is an *off-the-shelf* yet, commonly effective method.

For this comparison, we varied the number of parameters (number of weights and biases) in a considerably broad range. Results are shown in Table 3 as a function of the number of hidden nodes. In order to establish a fair comparison with our model, we need to use the same number of parameters. It turns out that the number of hidden nodes of the MLP must be equal to $K\sqrt{M}$, where K is the number of hidden nodes for each function in our approach and M is the number of functions. For this experiment this number is 22.

By comparing the results from Tables 2 and 3 we can observe that when the MLP employs the same number of free parameters, our proposed discriminative-generative method gives 1.09 to 1.48 times (relative to worst and best performance from Table 2) better accuracy. When MLP is allowed to have more parameters, our method still outperforms the MLP on average (0.97 to 1.31 times better); however, for a few inputs the performance is similar or better for MLP. Note that when the number of parameters for the MLP is larger, the variance also diminishes considerably. However, we should remark that to achieve such performance, the MLP needed to employ roughly 1.8 times the number of parameters employed by our model.

7.1.3 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. 6. 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 model cannot perform well in all configurations (this is true for almost any machine learning model) due to the following reasons:

1. The proposal distribution $q(\mathbf{h}|\mathbf{x})$ does not resemble the true posterior distribution $p(\mathbf{h}|\mathbf{x})$ at the particular $\mathbf{x} = \mathbf{x}^*$: 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 (about the pose) 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. 7 shows the estimates found using MS. These estimates can be interpreted as possible hypotheses of hand configurations given the silhouettes. Note that MS tends to bias the hypotheses towards samples from the distribution $q(\mathbf{h}|\mathbf{x}^*)$, but we can account for this when building a full probability distribution, as explained in Sec. 4.3.

7.2 3D Hand Pose Reconstruction Given an Unrestricted Camera Viewpoint

Our approach 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.

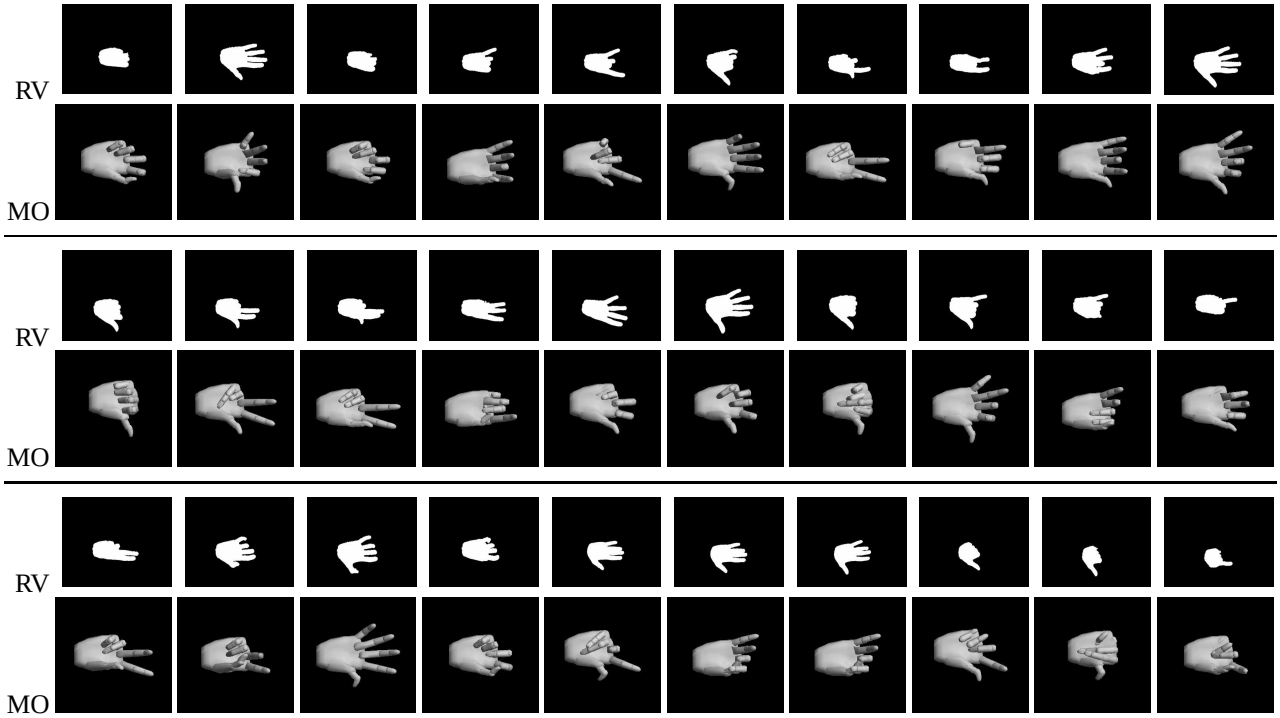


Figure 6: 40 examples of estimated hand poses captured every 0.9 secs from real video (RV). Reconstruction found using the Mean Output (MO) approach. The inverse function was computed using computer graphics rendering.

The number of mixture components (or mapping functions) was set to 45. This number was determined via the MDL criterion, as before (testing for the best MDL score using a model with 35,37,...,51 functions). Each function was a MLP with seven hidden nodes.

7.2.1 Quantitative Results

As before, we computed the absolute error in estimating hand pose, and quantitatively compared this measure across views. Fig. 8 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 variance seems similar across views. As expected, the average error is higher than that obtained for the fixed view hand pose reconstruction experiments. It seems that for unrestricted hand views it is a bit advantageous to use the computer graphics inverse 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. 9 shows the results using the MS approach. Fig. 9(a) shows the error associated with the best sample. This error behaves very similarly to the MO error. Fig. 9(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

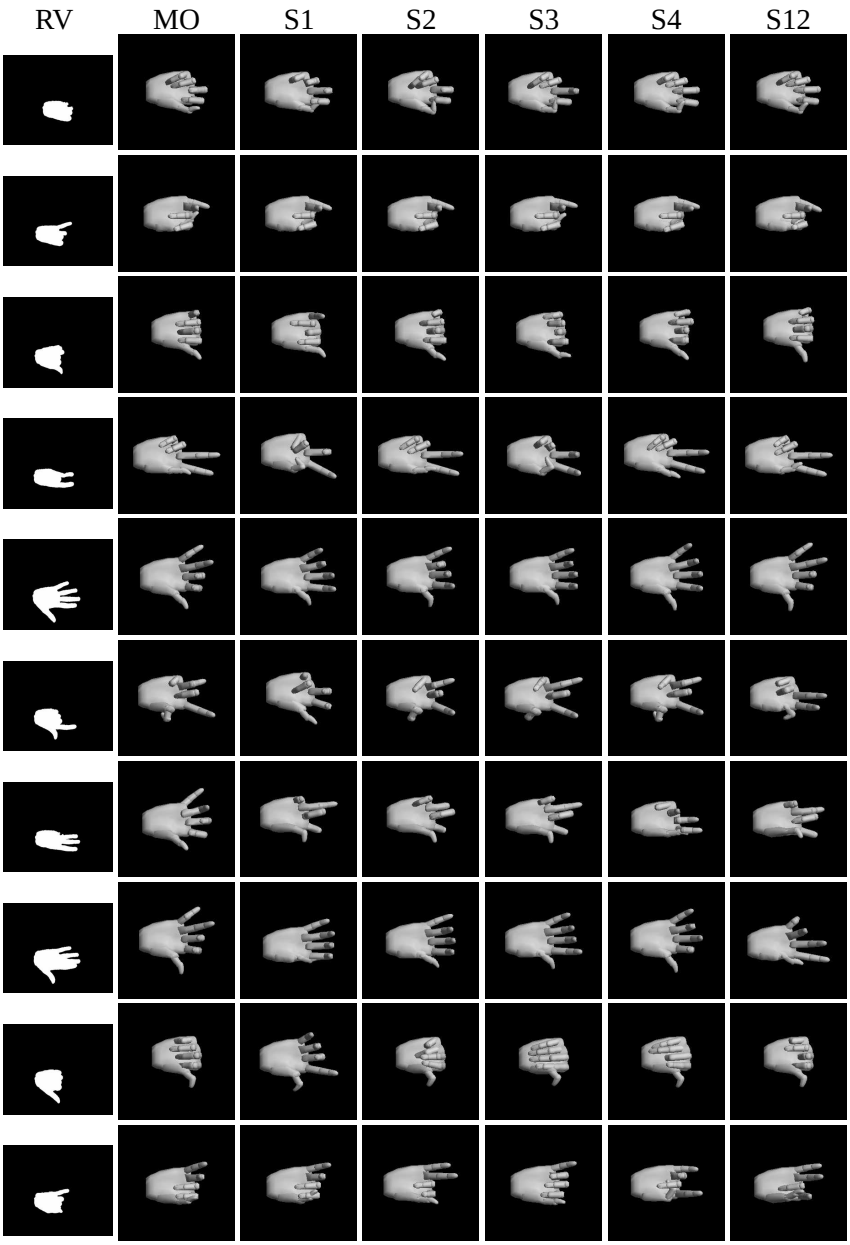
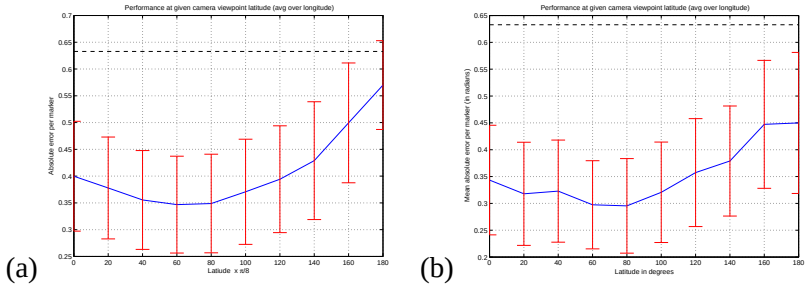


Figure 7: Example estimated hand poses obtained using the Multiple Sample (MS) approach and real video (RV). The inverse function was estimated from data. Columns 1-2 show the input video frame and the MO solution, columns 3-7 show sorted samples (1-4 and 12) obtained via the MS approach where S1 is the most probable sample..

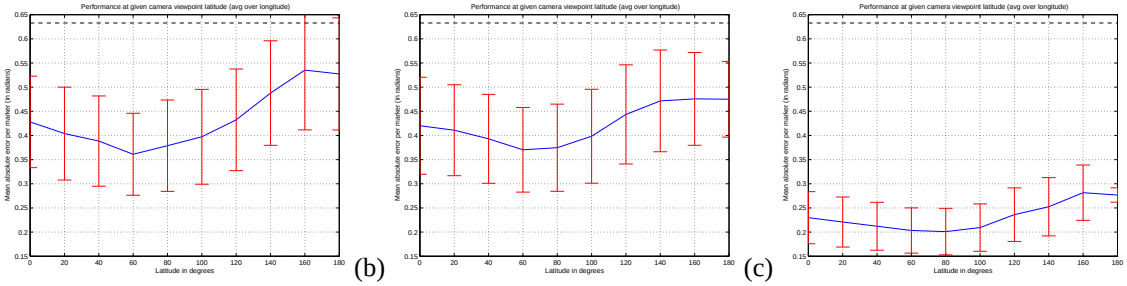
result given that the best sample is determined without having knowledge of ground-truth. In fact, if the average error of the best 20 samples were lower than that of the best sample, then we could infer that our algorithm is very inaccurate at determining what samples are better. Thus this result positively endorses the MS algorithm.

For comparison, we used the ground-truth to select the best sample, based on minimum error. In other words, we have an *oracle* that picks the sample closest to the ground-truth. The resulting performance graph



(a) (b)

Figure 8: Mean Output (MO) inference performance for unrestricted view tests at given viewpoint latitudes (averaging over longitude). The inverse 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 $\beta_2 = 0$. For reference, the performance of an algorithm that chooses the estimate at random from the training data is shown. The angle range is in average 1.87 radians.



(a) (b) (c)

Figure 9: Multiple Samples (MS) inference for unrestricted view tests at given viewpoint latitudes (averaging over longitude). The inverse function 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). For reference, the performance of an algorithm that chooses the estimate at random from the training data is shown. The angle range is in average 1.87 radians.

is shown in Fig. 9(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 inverse models.

7.2.2 Performance Comparison with Respect to Discriminative Model Alone and a Competing Approach

In parallel with Sec.7.1.2, we now compare our full method against the purely discriminative portion. This is done to illustrate the value of the generative model in the overall approach. As before, we also compare our method against the standard MLP for this dataset.

Using the discriminative model alone, the mean absolute error and variance for this dataset were 0.6102 and 0.5117 respectively. Since the importance of the generative method in the overall approach should, by now, be more clear, we will not discuss this point further. Results are analogous to those from Sec. 7.1.2.

As before, we compare our full approach against the standard MLP, which is trained on the same training set as our approach. Results are shown in Table 4. Note that when the MLP contains 47 hidden nodes, the

Number of Hidden Nodes	35	**47**	59	71	83	95
$\hat{\mathcal{E}}_{MLP}$	0.5775	**0.5714**	0.5585	0.5534	0.5511	0.5514
$\sigma^2_{\hat{\mathcal{E}}_{MLP}}$	0.3572	**0.3680**	0.3512	0.3637	0.3794	0.4111

Table 4: Performance of the standard MLP approach. Each entry in the table shows the mean absolute error and variance averaged over ten trials. Overall the performance of our method is 1.31 times better than this approach using the same number of free parameters, corresponding to 47 hidden nodes in the MLP. The performance of our approach is 1.28 times better when the MLP has 83 hidden nodes.

number of parameters is comparable with that of our discriminative-generative model. The performance of our approach for this dataset is shown in Figs. 8 and 9.

At first sight the performance comparison seems similar to that of our previous task with fixed viewpoint. However, a more careful look at Table 4 reveals that (1) our method clearly outperforms the MLP even when the MLP uses more than double the number of parameters with respect to our model, a significant difference from Sec. 7.2.2 (fixed viewpoint) where performance was more even when letting the MLP have more parameters; (2) also unlike Sec. 7.2.2 the variance is much larger than that of the estimates computed by our approach. The key difference between the fixed viewpoint dataset and this dataset (unrestricted view) was that mapping visual-features to hand-pose is much more ambiguous when any view is allowed. This illustrates that function approximation methods are generally not well-suited for one-to-many inference problems, and our method can indeed provide a more clear advantage in these cases.

7.2.3 Experiments with Real Images

We test our approach using video of hands (in any orientation) collected from a single uncalibrated camera. Pose estimates from 40 frames (taken every 0.9 secs apart) obtained via the MO approach are shown in Fig. 10. 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. 11 shows the estimates obtained via the MS approach. The frames shown were taken approximately every 0.9 seconds. 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 are close to each other in terms of their probability, given the features. 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.

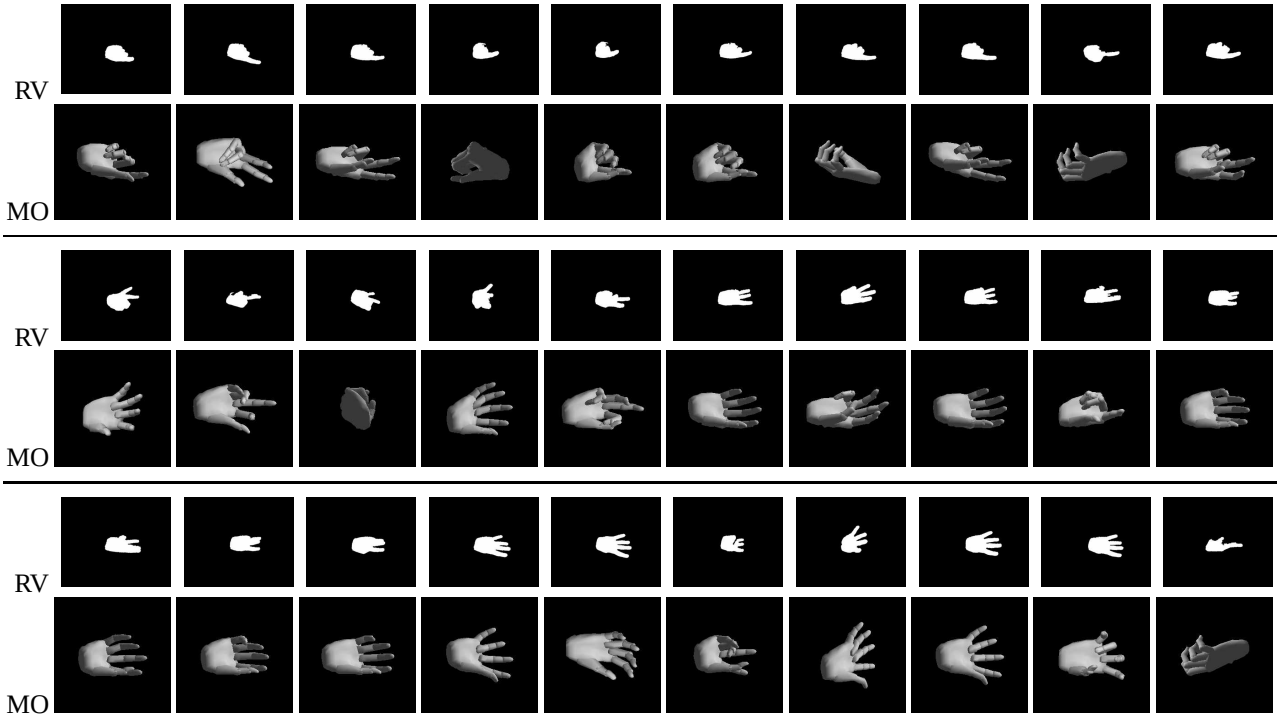


Figure 10: 40 examples of estimated hand poses captured every 0.9 secs from real video (RV). Reconstruction found using the Mean Output (MO) approach. The inverse function was computed using computer graphics rendering.

7.3 2D Human Body Pose Reconstruction

In order to show that our approach can be employed, with no change, to perform other similar tasks (possibly with a different representation), here we now conduct performance tests in the task of estimating human body pose from a single image. 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 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 mixture components in the discriminative model was set to 15. This number was determined via the MDL criterion, exactly as before. Each function is a MLP with seven hidden nodes.

7.3.1 Quantitative Results

Fig. 12 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.

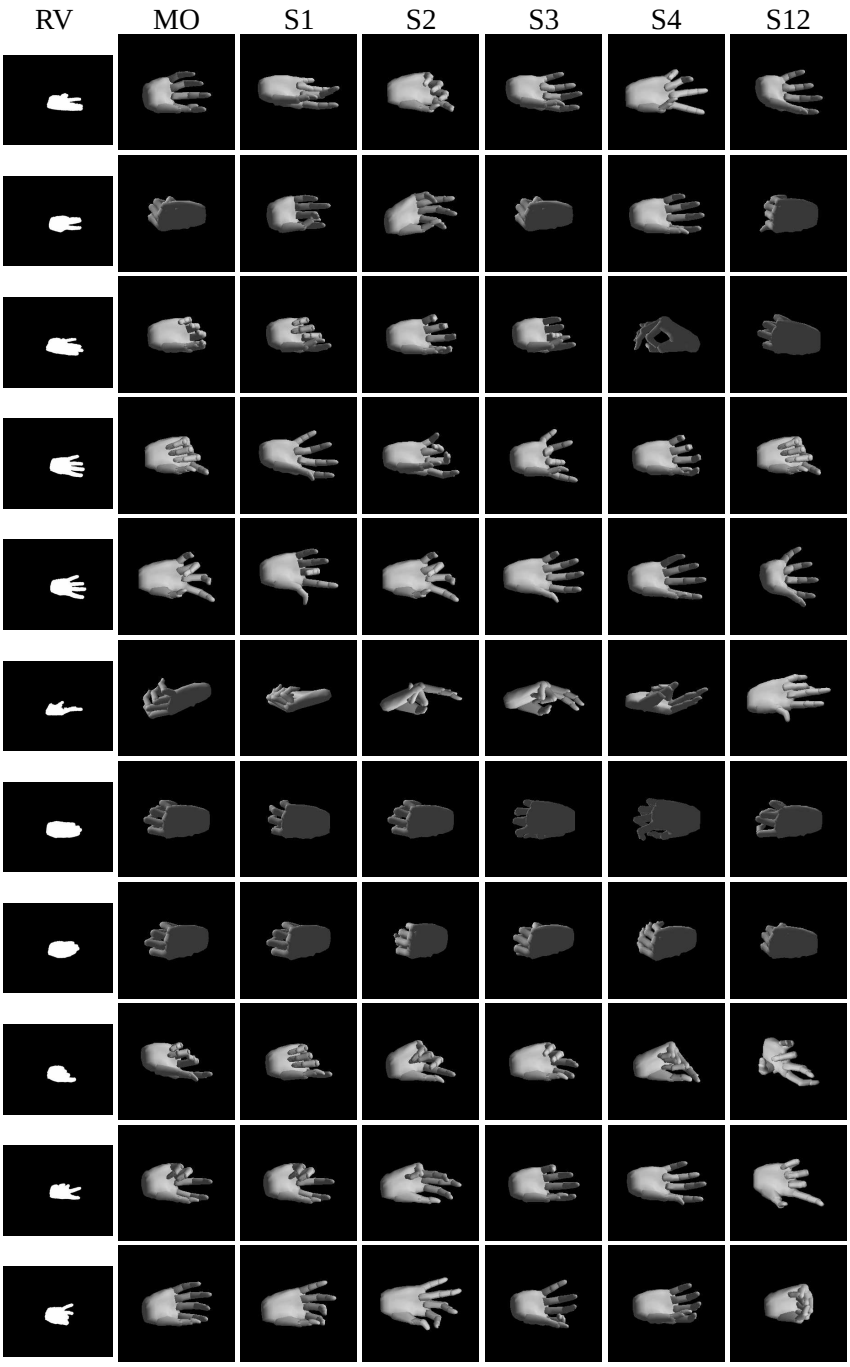


Figure 11: Example estimated hand poses obtained using the Multiple Sample (MS) approach and real video (RV). The inverse function was computed using computer graphics rendering. Columns 1-2 show the input video frame and the MO solution, columns 3-7 show sorted samples (1-4 and 12) obtained via the MS approach where S1 is the most probable sample..

Fig. 13 shows the average marker error and variance per body orientation in percentage of body height. 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

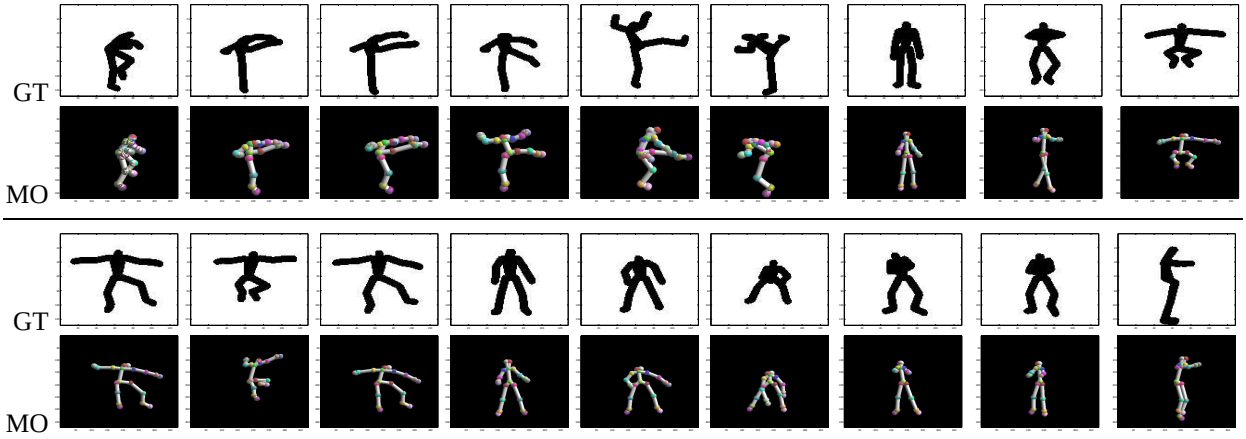


Figure 12: Example reconstruction of frames from test sequences with computer graphics-generated silhouettes.

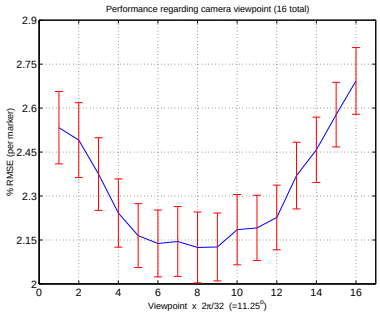


Figure 13: Root mean-square-error (divided by number of markers) and variance per camera viewpoint (every $2\pi/32$ rads.). Units are percentage of body height. Approx. 110,000 test poses were used.

promising, given the complexity of the task and the simplicity of the approach. Just as a reference point, by choosing poses at random from those in the training set, the RMSE was 10.35% of body height (with a standard deviation of 4.4%). 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. 14 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 accurate. In the Figure, we can see two particularly difficult configurations at the second row of real video (RV) images, fourth-sixth columns; the arm configuration is difficult to estimate. This could be

due to the lack of relevant training data, as a consequence the discriminative model q may not approximate the generative model p very well around the input vector. In general, an important issue to keep in mind is that the visual differences between the rendered model and the real body observed could become critical and thus accurate rendering may be desirable. This varies from application to application; however in any case the general inference approach presented here remains the same.

In this work, we did not pursue use of a more realistic human body renderer. Due to differences in shape and width of body components observed in training versus testing, the visual features may differ. This is a relevant point since in almost all learning models, it is expected that the training data be a good approximation to the real test data. Improving the match between visual features used in training and testing, and thus potentially the overall performance, is an area that we plan to investigate in future research. Despite the fact that we have ignored differences in anthropometric characteristics between CG and real silhouettes, the performance observed for both articulated objects (hands - human bodies) is very promising given that only a single image is assumed available.

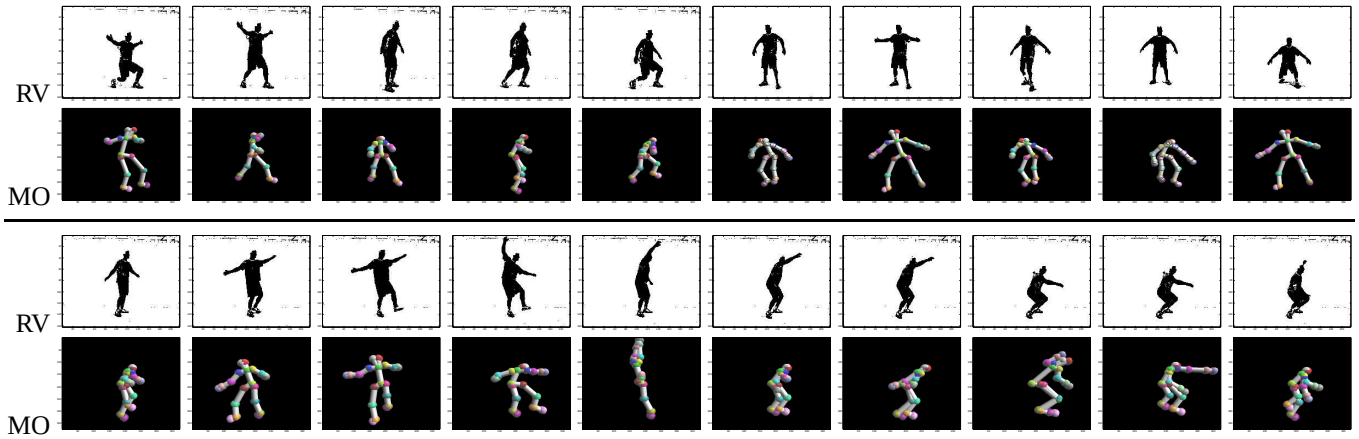


Figure 14: Reconstruction obtained from observing a human subject (every 10th frame).

8 Conclusions

In this paper, we have described a novel method that allows us to infer 3D and 2D articulated body pose from observed visual features in a single image, a problem usually regarded as ill-posed. This was done by combining generative and discriminative models to solve the complex probabilistic inference problem. This approach is most useful when the generative model is accurate (*e.g.*, we have an inverse mapping function) but it is diffi cult to perform inference using this model alone.

In order to solve the inference problem (and also perform MAP estimation), we have shown that a

mathematically sound approach is to use a discriminative model and learn its parameters using relevant training data. The probability distribution implied by the discriminative model can be used as a proposal distribution to generate samples and find a posterior probability distribution (perform approximate inference) under the (accurate but complex) generative model.

When comparing it to other relevant methods, we can find alternative interpretations of this framework. The use of a generative model (through ζ) affords an alternative to complex discriminative models; for example, it is an alternative to the gating networks of the Mixture of Experts model [25]. In general, instead of learning increasingly complex discriminative models such as [17, 13], we can exploit an accurate generative model and learn a simpler discriminative model. A clear advantage of using a generative model in this way is that it can provide useful information on the structure of the problem, a structure that discriminative models try to *blindly* uncover from the available data.

Our approach 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 a single image. This is a particularly difficult problem because this mapping is highly ambiguous and it is infeasible to perform inference using the generative model. We have obtained promising results even using a very simple set of image features, such as moment invariants of the body silhouette. Further experimental evaluation can be found in [35].

This approach offers several advantages over many previous methods for articulated pose estimation. These 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). We have shown how in some cases these alternative approaches could be seen as inferring a posterior distribution using the generative model only. While we have used a camera model in defining our generative model, we have not attempted to solve the resulting optimization problem directly; instead we have had the help of the proposed discriminative model. Thus, in contrast with many past approaches, no iterative minimization methods are used in pose inference. Moreover, inference is fully automatic – no manual initialization of the articulated model is required.

Our method does not use iterative optimization for inference, but a valid question is why not iterate the input-output mappings several times? In fact, there exist approaches that use a series of top-down along with bottom-up iterations for estimation, with the further claim that these iterations may perform error correction (*e.g.*, when the observation is noisy). However, the process of re-iterating *up and down* and obtaining estimates alternating between a pair of spaces (or sets) does not, in general, guarantee (1) convergence towards the desired value (pose in our case) (2) monotonic improvement of the solution, or (3) convergence

at all. Only under specific conditions can this desirable behavior be attained (see *e.g.*, [9]). In our case, these conditions cannot be proven, and therefore, we prefer not to use this re-iterating methodology (one condition requires that the spaces be convex).

A set of previous approaches attempt to learn articulated model dynamics [5, 19, 42]; however, learning dynamics requires substantially more training data, and tends to produce systems that are biased towards specific motions (this can be good news if the range of motions, rather than just that of configurations, is known beforehand or comprises the training set). Our framework avoids this and infers pose from a single image only. It is clear that in highly-constrained domains and where motion is available, models of dynamics can provide an enormous advantage. In this paper we have approached a different problem, estimating body pose from a single image, where multiple frames are not provided.

Several interesting problems remain for future work. Within the context of articulated pose estimation, performance can be improved in several ways. We have observed that in practice segmentation of real video is noisy, especially when compared with the clean segmentation obtained in the experiments with synthetic data. As we found in our experiments, our method produces qualitatively good pose estimates for real video. However, significant differences in the overall limb (arms/legs) width in the silhouette can sometimes lead to errors in pose estimation. Such differences arise due to morphological image operations and segmentation parameters, significant clothing, or anthropometric differences between the subjects and the computer graphics model used in generating training data. We believe that it should be possible to address this by including training data that is representative of the variations we expect across humans (through a more complex graphics rendering process or noise model). An alternative solution is to *adapt* the system to work with the body morphology properties *specific* to the input images observed; for example by defining an additional *morphological* parameter to relate *user specific* body properties to the *standardized* computer generated morphology used for training. A tighter integration of pose estimation with image segmentation is a more difficult problem worth exploring, that could provide greater robustness to noise or even occlusion. Some of these topics are the subject of current research.

Another general problem is to learn what the best features are for specific problems or datasets. This classic problem has spawned numerous approaches. From general information theoretic [8] techniques based on maximizing mutual information to approaches specific to image processing [21]. Roughly speaking, one wants to obtain features that can distinguish among the patterns we care about, for a specific task or data set. In general this problem is difficult because the structure of the space of all possible features cannot be represented in simple way, and is not amenable to efficient optimization. This concept can be seen as that of *learning the features*, and it is closely connected to that of *learning the mapping functions*, in fact they

can be seen as two views of this problem.

Methods for incorporating knowledge of dynamics in the same framework should be investigated. In this work we have concentrated on estimating pose from single images. This of practical importance in many tasks, such as model initialization (*e.g.*, for tracking), recovery (*e.g.*, when tracking is lost or not reliable), pose from single image (*e.g.*, when photographs need to be used as sources), etc. The "pose from a single image" problem is different from, and in some ways more diffi cult than, pose tracking with dynamics information.

While promising advances have been made in estimating pose from a single view, extending our framework to incorporate the above concepts remains a topic for future investigation.

Acknowledgments

The hand sequences used in our experiments were collected in collaboration with Vassilis Athitsos at Boston University. We thank Tommi Jaakkola at MIT, Quaid Morris at University of Toronto, and Matt Brand at MERL for their valuable suggestions and for interesting discussions. This research was supported in part by the U.S. Offi ce of Naval Research under grants N000140310108 and N000140110444, and the U.S. National Science Foundation under grants IIS-0208876 and IIS-9809340.

Appendix

The KL divergence between the empirical distribution p_e (represented by the training data) and the model q is:

$$\mathrm{KL}(p_e(\mathbf{x},\mathbf{h})||q(\mathbf{x},\mathbf{h})) = \int p_e(\mathbf{x},\mathbf{h}) \log[p_e(\mathbf{x},\mathbf{h})/q(\mathbf{x},\mathbf{h})] d\mathbf{h} d\mathbf{x}. \tag{19}$$

If θ parameterizes the conditional $q(\mathbf{h}|\mathbf{x})$, then the minimum of the above expression can be proven to be equivalent to:

$$\arg\min_{\theta} E_{p_e(\mathbf{x})}[\mathrm{KL}(p_e(\mathbf{h}|\mathbf{x})||q(\mathbf{h}|\mathbf{x}))], \tag{20}$$

In practice, the expectation becomes a sum over the training data pairs, and we obtain Eq. 1. Thus, the optimal distribution in this sense is the one that results from solving Eq. 1, to obtain $q(\mathbf{h}|\mathbf{x})$. Of course, we assume that the data is composed by representative examples from p, so that the empirical distribution p_e is at all useful.

Eq. 7 justifies this choice since it tells us that in order to find a good approximation for the posterior $p(\mathbf{h}|\mathbf{x})$ we should find a proposal distribution that is similar to it, as intuitively expected. We may then ask if we could use this proposal distribution alone. The reason why this is not a good idea is that, since we cannot usually find a proposal distribution that matches the true posterior perfectly, using this proposal distribution alone is expected to perform worse than when combined with our accurate generative model. In fact, this was experimentally verified in Secs. 7.1.2 and 7.2.2. This is mainly because in regions where the proposal distribution q is bad at approximating p, we can always evaluate p and note the error or discrepancy.

The distribution $q(\mathbf{h}|\mathbf{x})$ is an approximation to $p(\mathbf{h}|\mathbf{x})$ in the space of all distributions with the structure specified by the discriminative model (a mixture model in our case). For Gaussian mixture models, it is know that this approximation can be made as accurate as we wish in the limit of infinite data and mixture components. Interestingly, obtaining a good approximation to the posterior does not explicitly require knowledge of the prior $p(\mathbf{h})$ in our generative model. Note that the training data indirectly provides some of this information through the learned discriminative model (in fact the data could further be used to directly estimate $p(\mathbf{h})$ if necessary).

Throughout the paper we showed MAP estimation. For the sake of completeness, if we are interested in computing an approximation to the probability of a body pose $\mathbf{h}$, given an observation of features $\mathbf{x}^*$, we use the expression:

$$\hat{p}(\mathbf{h}|\mathbf{x}^*) = \frac{1}{\hat{Z}_p}\mathcal{N}(\mathbf{x}^*; \zeta(\mathbf{h}), \Sigma_\zeta)p(\mathbf{h}), \tag{21}$$

with $\hat{Z}_p$ given by $\frac{1}{S}\sum_{s=1}^{S} p(\mathbf{x}^*, \mathbf{h}^{(s)})/q(\mathbf{h}^{(s)}|\mathbf{x}^*)$, using importance sampling with proposal distribution $q(\mathbf{h}|\mathbf{x}^*)$ to obtain the samples $\mathbf{h}^{(s)}$.

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 *Proc. Computer Vision and Pattern Recognition*, pages 669–676, 2000.

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

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

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

[7] J. Cheng and M. Druzdzel. AIS-BN: An adaptive importance sampling algorithm for evidential reasoning in large bayesian networks. *Journal of Artificial Intelligence Research*, 13:155–188, 2000.

[8] T. Cover and J. Thomas. *Elements of Information Theory*. Wiley Series in Telecommunications. John Wiley & Sons, New York, NY, USA, 1991.

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

[10] 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.

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

[12] P. Felzenszwalb and D. Huttenlocher. Efficient matching of pictorial structures. In *Proc. Computer Vision and Pattern Recognition*, 2000.

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

[14] 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.

[15] 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.

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

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

[18] 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.

[19] 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.

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

[21] T. Iijima, H. Genchi, and K. Mori. A theory of character recognition by pattern matching method. In *Proc. First Int'l Joint Conf. Pattern Recognition*, pages 50–56, 1973.

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

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

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

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

[26] D. Mackay. Introduction to Monte Carlo methods. *Learning in Graphical Models*, 1998.

[27] G. J. McLachlan. *Discriminant Analysis and Statistical Pattern Recognition*. Wiley, New York, 1992.

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

[29] A. Ng and M. Jordan. On discriminative vs. generative classifi ers: a comparison of logistic regression and naive Bayes. In *Advances in Neural Information Processing Systems*, pages 841–848, 2001.

[30] 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.

[31] 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.

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

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

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

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

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

[37] R. Rubinstein. *Simulation and the Monte Carlo method*. John Wiley & Sons, 1981.

[38] Y. Rubinstein and T. Hastie. Discriminative vs. informative learning. In *3rd International Conference on Knowledge Discovery and Data Mining*, pages 49–56, 1997.

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

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

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

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

[43] C. 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.

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

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

[46] S.C. Zhu, C. Guo, and Y. Wu. Modeling visual patterns by integrating descriptive and generative models. *International Journal of Computer Vision*, 53(1):5–29, 2003.