

Articulated Pose Estimation in a Learned Smooth Space of Feasible Solutions

Tai-Peng Tian, Rui Li and Stan Sclaroff*

Computer Science Department, Boston University

Boston, MA 02215

{tian, lir, sclaroff}@cs.bu.edu

Abstract

A learning based framework is proposed for estimating human body pose from a single image. Given a differentiable function that maps from pose space to image feature space, the goal is to invert the process: estimate the pose given only image features. The inversion is an ill-posed problem as the inverse mapping is a one to many process, hence multiple solutions exist. It is desirable to restrict the solution space to a smaller subset of feasible solutions. The space of feasible solutions may not admit a closed form description. The proposed framework seeks to learn an approximation over such a space. Using Gaussian Process Latent Variable Modelling. The scaled conjugate gradient method is used to find the best matching pose in the learned space. The formulation allows easy incorporation of various constraints for more accurate pose estimation. The performance of the proposed approach is evaluated in the task of upper-body pose estimation from silhouettes and compared with the Specialized Mapping Architecture. The proposed approach performs better than the latter approach in terms of estimation accuracy with synthetic data and qualitatively better results with real video of humans performing gestures.

1 Introduction

Many problems in computer vision involve estimating parameters of a particular model from input images. Examples include line fitting, camera calibration, image matching, surface reconstruction, motion analysis and pose estimation. Parameter estimation problems are generally formulated as optimization problems. For a given parameter estimation problem, different approaches exist due to various optimization techniques and different forms of parametrization.

In problems such as human pose estimation from images [2, 12, 15, 17] or hand pose estimation [3], the goal is to estimate parameters of a known model given images as observations. We propose a new framework in this paper for solving this class of parameter estimation problems

with the motivating application of upper body pose estimation. Previous approaches [2, 3, 4, 5, 8, 12, 14, 15, 17, 18] to the 2D/3D pose estimation have the following problems:

- do not scale well spatially and only provide a coarse representation of the solution space [2, 3, 5, 17, 15],
- computationally expensive [12],
- need a human in the loop [4, 8, 14, 18].

Our proposed framework exploits machine learning techniques to avoid the above listed limitations and it is fully automatic. Efficient and better estimation is achieved given the smooth parametrization provided by Gaussian Process Latent Variable Model (GPLVM) of an approximate feasible solution space. The advantages have been demonstrated in experiments designed for the 2D upper body pose estimation problem. We compared our approach with the approach of Specialized Mapping Architecture (SMA) [15]. The estimation accuracy of the SMA is at least one standard deviation worse than the proposed approach in experiments with synthetic data. In experiments with real video of humans performing gestures, the proposed approach produces qualitatively better estimation results. The proposed framework is general and could be applied for parameter estimation problems of a similar nature.

1.1 Problem Formulation

Pose estimation from a single image is formulated as a generic parameter estimation problem. The differentiable *forward function* $\Phi : \mathbb{R}^m \rightarrow \mathbb{R}^n$, describes the forward mapping from parameter space to feature space. For example, in Rosales and Sclaroff's work [15], they consider the forward function as a rendering function where the parameter space is a vector space of 2D human pose joint positions and the feature space is a vector space of Alt moments.

Given a feature vector $\mathbf{s} \in \mathbb{R}^n$, we seek the parameter $\mathbf{y} \in \mathbb{R}^m$ that best explains the feature vector. The quality of solution can be assessed by evaluating the difference between $\Phi(\mathbf{y})$ and \mathbf{s} through a cost function $C(\Phi(\mathbf{y}), \mathbf{s})$.

¹This research was funded in part through ONR grant N00014-03-1-0108 and NSF grants IIS-0208876, IIS-0308213, and IIS-0329009.

1.2 Overview of Proposed Framework

For most of parameter estimation problems, the *feasible parameter space* \mathbb{Y} is typically a smaller subset of \mathbb{R}^m . For example, in estimating joint angles of a hand pose, due to articulation constraints, the fingers cannot bend beyond a certain degree, thus not all points in \mathbb{R}^m correspond to valid hand poses.

We would like to construct an approximation $\tilde{\mathbb{Y}}$ using points sampled in \mathbb{Y} . At the same time, we would like to recover a smooth parametrization $L_{\tilde{\mathbb{Y}}}(\mathbf{x})$. The parameter \mathbf{x} is typically chosen to be low dimensional. For this work we use the Gaussian Process Latent Variable Model (GPLVM) [9] to learn $L_{\tilde{\mathbb{Y}}}(\mathbf{x})$. Section 3.2 describes the GPLVM learning process. In the context of GPLVM, the low dimensional space of \mathbf{x} 's is called *latent space* and \mathbf{x} 's are called *latent variables*.

In the framework of parameter estimation, given an input feature \mathbf{s} , we search over the latent space while minimizing the cost $C(\Phi(L_{\tilde{\mathbb{Y}}}(\mathbf{x})), \mathbf{s})$. Section 3.3 describes the optimization process in more detail.

2 Related Work

There is a broad range of related work that solves similar parameter estimation problems. In *example based estimation*, a large database of parameter and feature pairs is collected and indexed. Given a query feature, the database returns a parameter value with the closest matching feature. The main issue addressed in this line of work is how to perform a computationally expensive query quickly and accurately. For example, Shakhnarovich, *et al.* [17] use hashing functions to quickly construct approximate nearest neighbors of the solution in parameter space. The solution is further refined using Locally Weighted Regression (LWR). To speed up search, Athitsos, *et al.* [3] use Lipschitz embeddings to approximate a computationally intensive feature space matching algorithm. Casting an estimation problem as a database query problem has the advantage of leveraging on research done in the database community. Typically such an approach does not scale well spatially as a large number of samples are required to cover the parameter space adequately. In contrast, our approach has a more compact representation. By learning a smooth parametrization of the feasible parameter space, we effectively summarize the database using a few parameters. After the learning phase, we only keep a small fraction of the training set for use in the query stage.

Another line of prior work is based on learning the reverse process of Φ . Agarwal, *et al.* [2] directly learn a mapping from feature space to parameter space using Relevance Vector Machine. Rosales and Sclaroff [15] further recognize that such a mapping may be many to one and learn multiple inverse functions to explain such phenomena. The

fundamental idea is to generate a finite number of hypotheses through the inverse functions and find the best hypothesis by verifying it with the forward function. Extrapolating this idea, we can generate more and more hypotheses. Taking this idea to the extreme, a continuum of such hypotheses can be described using a function. We can search for an optimal solution in this continuum using optimization techniques. This is exactly what our framework advocates. For an input feature, we construct a continuum of plausible hypotheses by restricting the search in $\tilde{\mathbb{Y}}$. More specifically, we add constraints specifying that the parameters should generate features similar to the query. Therefore, instead of considering a finite number of solutions, we generalize this line of thought by considering a broader range of solutions described in terms of a function $L_{\tilde{\mathbb{Y}}}$.

Brand [5] uses a Hidden Markov Model (HMM) to represent a dynamic manifold. This is similar to representing the underlying density with a mixture of Gaussians. HMM learning also requires prior specification of the topology of the Markov Model. Our work uses the GPLVM which is based on Gaussian Processes (GP). Our representation has the advantage of being smoother as it is statistically non-parametric and GP representation can be easily captured using a few hyper-parameters [7, 13]

In the work of Lee, *et al.* [12], the parameter estimation problem is treated in a probabilistic framework. Estimating the parameter amounts to maximizing the posterior probability distribution. Such a distribution is usually complicated. Solutions are typically approximated using computationally expensive techniques, like the class of Markov Chain Monte Carlo algorithms. Our approach does not require a computationally intensive searching process. We reduce the complexity of search by learning a smooth parametrization of the feasible solution space. Using fast optimization techniques, our algorithm can converge to a solution quickly.

Some early work and extensions [4, 8, 14, 18] consider the case where corresponding points between the model and image are known. Grochow, *et al.*'s approach [8] also falls into this category though GPLVM is used in their model. Manually specified constraints have to be provided for missing motion capture information. Geometric constraints are used to estimate the parameters of the model. In contrast, our work is fully automatic and no correspondence is required for the parameter estimation and GPLVM is just one way to realize our generic framework.

There is also a large amount of work done on non-linear manifold embedding in low dimensional space as Principal Component Analysis (PCA) is inadequate to handle such non-linear behavior. Methods like Local Linear Embedding (LLE) and Isomap [16, 19] are representative. Both techniques provide a discretized embedding to the original manifold. For our purpose, a smooth representation in mapping

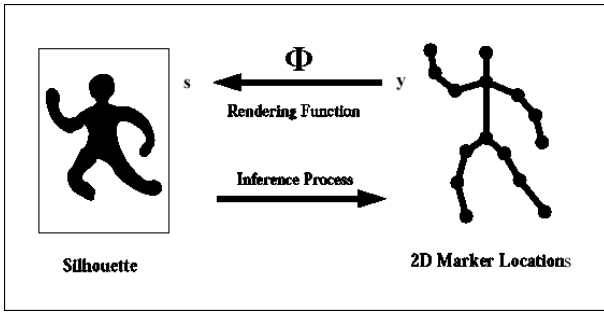


Figure 1: *Problem Overview.*

the original data to the lower dimensional space is desired for optimization and possible interpolation. A recent work using GPLVM [9] adopts a probabilistic approach to embed data into a lower dimensional space. It was originally meant for visualizing high dimensional data. We use it as a tool to learn the parametrization $L_{\tilde{\mathbb{Y}}}$ in our framework for its smooth latent space representation.

3 Pose From a Single Image

The problem we aim to solve can be loosely formulated as: given only a person’s silhouette, find the corresponding 2D body pose. More rigorously, let $\mathbf{y} \in \mathbb{Y}$ be the 2D pose of a human model and let $\mathbf{s} \in \mathbb{R}^n$ be the silhouette associated with it. There exists a function $\Phi : \mathbb{Y} \rightarrow \mathbb{R}^n$ that uniquely maps each \mathbf{y} to an \mathbf{s} . For example, Φ may be a rendering function that renders a 2D model into a silhouette. The function Φ is known to us and we are more interested in solving for the pose from given a single silhouette. Figure 1 shows this relation between the silhouette image and 2D pose.

We use the 3D ground truth data of a subject’s joint positions to generate training data. For a number of camera viewpoints the 3D data are mapped into the corresponding 2D image positions. The projected 2D joint positions form our training data set, $\{\mathbf{y}_i\}$. We learn $L_{\tilde{\mathbb{Y}}}$ by probabilistically projecting $\{\mathbf{y}_i\}$ into a smooth continuous low dimensional space representation $\{\mathbf{x}_i\}$ using the GPLVM. In probabilistic terms, the \mathbf{y}_i ’s are the observations and \mathbf{x}_i ’s are the latent variables.

Once the probabilistic relationship between \mathbf{y} and \mathbf{x} is learned through the GPLVM, the inverse problem is cast as an optimization problem. Given a new image with silhouette feature \mathbf{s} , we seek to find the corresponding optimal values of \mathbf{y} and \mathbf{x} such that the likelihood of observing \mathbf{y} given \mathbf{x} is optimized under the constraint that \mathbf{s} is fixed. Our approach consists of the following steps.

3.1 Learning Φ from Training Poses

The function Φ is learned through training a simple feed-forward neural network (similar to [15]) that takes the form

$$\Phi(\mathbf{y}) = \mathbf{w}_{out}\Omega(\mathbf{w}_{in}\mathbf{y} + \mathbf{b}_{in}) + \mathbf{b}_{out},$$

where $\Omega(\mathbf{x}) = 2/(1 + \exp(-2\mathbf{x})) - 1$, \mathbf{w}_{in} and \mathbf{w}_{out} are the weights associated with corresponding input and output nodes, \mathbf{b}_{in} and \mathbf{b}_{out} are the corresponding bias.

3.2 Learning $L_{\tilde{\mathbb{Y}}}$

Given training poses $\{\mathbf{y}_i\}$ as inputs, we use a GPLVM to define a smooth continuous low-dimensional representation of the original data, which is called *latent space*. It is spanned by the values of latent space variables \mathbf{x}_i , which comprise the lower dimensional representation of corresponding \mathbf{y}_i . During learning, we estimate \mathbf{x}_i for each input training example \mathbf{y}_i , along with the parameters of the GPLVM model (denoted by α and γ). This learning process is formulated as an optimization problem.

3.2.1 GPLVM Basics

The GPLVM is based on the Gaussian Process (GP) model, which describes the mapping between \mathbf{x} values and \mathbf{y} values. For a detailed tutorial on GP’s and the GPLVM, see [9, 13]. We only describe the basic mechanism and the implementation of GPLVM here.

The kernel matrix, \mathbf{K} , is the core of the GPLVM model. We use the Radial Basis Function (RBF) kernel function because it smoothly interpolates the latent space. The RBF kernel we use takes the form:

$$k_{RBF}(\mathbf{x}_i, \mathbf{x}_j) = \alpha \exp(-\frac{\gamma}{2}(\mathbf{x}_i - \mathbf{x}_j)^T(\mathbf{x}_i - \mathbf{x}_j)),$$

where $k_{RBF}(\mathbf{x}_i, \mathbf{x}_j)$ is the element in i -th row and j -th column of the kernel matrix \mathbf{K} , α controls the scale of the output functions, γ is the inverse width parameter. $k_{RBF}(\mathbf{x}_i, \mathbf{x}_j)$ measures the proximity between two points \mathbf{x}_i and \mathbf{x}_j in the input space.

3.2.2 GPLVM Learning

GPLVM learning is the process of learning the kernel parameters (α and γ) and latent variables \mathbf{x}_i ’s. Given a set of training data $\{\mathbf{y}_i\}$, each \mathbf{y}_i is a M dimension vector. We collect the m -th dimension of input \mathbf{y}_i ’s into \mathbf{Y}_m . Then we maximize the posterior $p(\{\mathbf{x}_i\}, \alpha, \gamma | \{\mathbf{y}_i\})$, which corresponds to minimizing the following objective function:

$$L = \frac{M}{2} \ln |\mathbf{K}| + \frac{1}{2} \sum_m \mathbf{Y}_m^T \mathbf{K}^{-1} \mathbf{Y}_m + \frac{1}{2} \sum_i \|\mathbf{x}_i\|^2, \quad (1)$$

with respect to the α , γ and \mathbf{x}_i ’s.

The intuition and derivation of L can be found in [10]. This optimization process is realized through the scaled conjugate gradient (SCG) method. The gradients needed for optimization are listed in the Appendix A.

To speed up the training, \mathbf{K} is only learned on a subset of the training data. This selected subset is then called the *active set* and denoted by \mathbf{I} . The active set can be considered as a sparse representation of the training data. The process of selecting the active set is described in [11]. The remaining points are denoted by \mathbf{J} . *Active set* selection allows us to optimize each point in \mathbf{J} independently [20]. We can solve for each \mathbf{x}_j in \mathbf{J} by minimizing the following objective function:

$$L_{\tilde{\mathbf{Y}}}(\mathbf{x}_j, \mathbf{y}_j) = \frac{\|\mathbf{y}_j - \mu(\mathbf{x}_j)\|^2}{2\sigma^2(\mathbf{x}_j)} + \frac{M}{2} \ln \sigma^2(\mathbf{x}_j) + \frac{1}{2} \|\mathbf{x}_j\|^2, \quad (2)$$

where

$$\mu(\mathbf{x}_j) = \mathbf{Y}^T \mathbf{K}_{I,I}^{-1} \mathbf{k}_{I,j}, \quad (3)$$

$\mathbf{K}_{I,I}$ denotes the kernel matrix learned from the *active set*. The vector $\mathbf{k}_{I,j}$ is made up of the rows in I from the j -th column of \mathbf{K} , and the variance is

$$\sigma^2(\mathbf{x}_j) = k(\mathbf{x}_j, \mathbf{x}_j) - \mathbf{k}_{I,j}^T \mathbf{K}_{I,I}^{-1} \mathbf{k}_{I,j}. \quad (4)$$

Taking gradients of $L_{\tilde{\mathbf{Y}}}$ with respect to \mathbf{x}_j do not depend on other data in J . The gradients of $L_{\tilde{\mathbf{Y}}}$ with respect to \mathbf{x} and \mathbf{y} are listed in Appendix A. The learning process is summarized in Algorithm 1.

Algorithm 1 GPLVM Learning Algorithm

Initialize size of active set, D , number of iterations T .
Initialize the \mathbf{X} from \mathbf{Y} through ISOMAP [19].
for T iterations **do**
 Select a new active set based on [11].
 Optimize L (Equation 1) using scaled conjugate gradient(SCG).
 Select a new active set.
 for each component j not in the active set, **do**
 Optimize $L_{\tilde{\mathbf{Y}}}$ (Equation 2) with respect to \mathbf{x}_j using SCG.
 end for
end for

3.3 Pose Estimation

To estimate body pose from a single silhouette image, which is represented by its Alt moments, \mathbf{s} , we first make use of the *active set* to initialize the pose. We then optimize an objective function $L_{\tilde{\mathbf{Y}}}(\mathbf{x}, \mathbf{y})$, which is derived from the GPLVM model, with respect to \mathbf{x} and \mathbf{y} , subject to the constraint that the estimated pose must have the same Alt moments. The function, $L_{\tilde{\mathbf{Y}}}$, describes the likelihood of the

estimated pose, given the initial pose, the learned model parameters and the constraints from silhouette feature \mathbf{s} . Optimizing \mathbf{x} and \mathbf{y} together ensures the most reliable estimation with respect to the training data.

For an input silhouette \mathbf{s}_{in} , the pose estimation is treated as the following optimization problem:

$$\arg \min_{\mathbf{x}, \mathbf{y}} (L_{\tilde{\mathbf{Y}}}(\mathbf{x}, \mathbf{y}) + w_1 C_{Alt}), \quad (5)$$

such that $C_{Alt} = \|\Phi(\mathbf{y}) - \mathbf{s}_{in}\|^2$. The optimization is realized by SCG. Equation 5 is highly non-linear and gradient-based techniques may be trapped in local minimum, hence proper initialization is important for the success of the estimation. For the initialization of (\mathbf{x}, \mathbf{y}) , we make use of the *active set*. We search through the *active set* to find the pair $(\mathbf{x}_i, \mathbf{y}_i)$ such that $\|\Phi(\mathbf{y}_i) - \mathbf{s}_{in}\|^2$ is the smallest. This is enforced by assigning a large value to w_1 so that C_{Alt} carries a large weight during the optimization.

3.4 Pose Estimation from Video Sequences

Given a gesture video sequence, we can make use of temporal consistency to improve pose estimation. The temporal consistency can be enforced by adding another constraint as follows:

$$\arg \min_{\mathbf{x}, \mathbf{y}} (L_{\tilde{\mathbf{Y}}}(\mathbf{x}, \mathbf{y}) + w_1 C_{Alt} + w_2 C_{temporal}), \quad (6)$$

where $C_{temporal} = \|\mathbf{y}(t) - \mathbf{y}(t-1)\|^2$, $\mathbf{y}(t)$ is the pose estimated in the current frame and $\mathbf{y}(t-1)$ is the pose estimated in the previous frame. We can use $\mathbf{y}(t-1)$ as the initial value of $\mathbf{y}(t)$ during optimization.

4 Implementation

We demonstrate the proposed approach on upper body pose estimation. The 2D articulated pose is defined in terms of the 2D locations of the person's joints in the image. Figure 2 shows the joint locations used for the 2D upper body. These joint locations are the parameters of a person's pose, defined as \mathbf{y} , where $|\mathbf{y}| = 24$ for upper body pose as shown in Figure 2. The silhouette features are represented using Alt moments, \mathbf{s}

$$\mathbf{s} = [\eta_{11}, \eta_{03}, \eta_{12}, \eta_{21}, \eta_{30}, \eta_{04}, \eta_{13}, \eta_{22}, \eta_{31}, \eta_{40}]^T,$$

where

$$\eta_{pq} = \frac{1}{n} \sum_{i=1}^n \left(\frac{u_i I_i - \bar{u}}{\sigma_u} \right)^p \left(\frac{v_i I_i - \bar{v}}{\sigma_v} \right)^q,$$

n is the number of pixels in the image, u_i and v_i are the row and column of pixel i . I_i is the intensity value of pixel i and \bar{u} and σ_u are the mean and variance. Alt moments have the

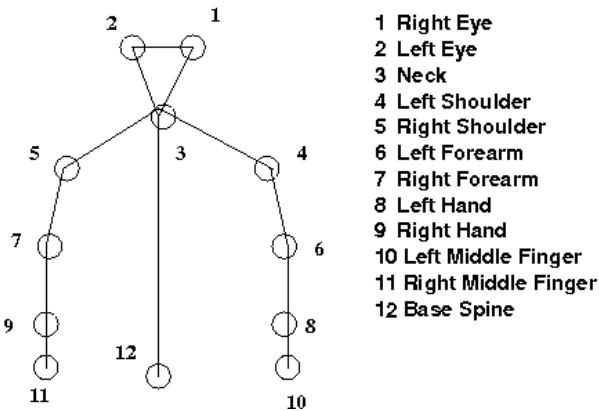


Figure 2: *Upper Body Joints*

advantage of being scale and translation invariant, but not rotation invariant. Rotation invariance is undesirable as all the input silhouettes are of people in the upright position. Other features might be possible; we tested our algorithm using Alt moments because we want to make a fair comparison with [15] during the experiments.

For the GPLVM model learning, we synthesize training data of upper body poses of a male character similar to [15]. The main poses present in the training data are a subset of the gestures used in aircraft signals [1]. The silhouette images are generated using a more accurate rendering function from Poser 5 [6]. Training with 3092 training poses takes around two hours to complete on a quad-processor 2.2GHz AMD Opteron(tm). A portion of the learned latent space is presented in Figure 3 together with corresponding silhouette images for easy visualization. In Figure 3, it can be seen that silhouette images of similar poses are placed near to each other and there are smooth transitions between different body poses.

Once the model is learned, we can use a captured silhouette image as input, first compute its Alt moments, and then use the estimation algorithm described in Section 3.3 to estimate the pose. Pose estimation takes less than 0.3 seconds on a dual-processor Intel P4 CPU 2.80GHz, using Matlab. With temporal consistency, we can further limit the search space, hence faster performance (0.1 seconds) and higher accuracy are achieved and reported in Section 5. Further speedup can be easily achieved for tracking applications by optimizing the Matlab implementation (we modified the GPLVM software downloaded from <http://www.dcs.shef.ac.uk/~neil/gplvm/>).

5 Experiments

We tested the proposed estimation algorithm on both synthetic and real data. The silhouette images for the test data

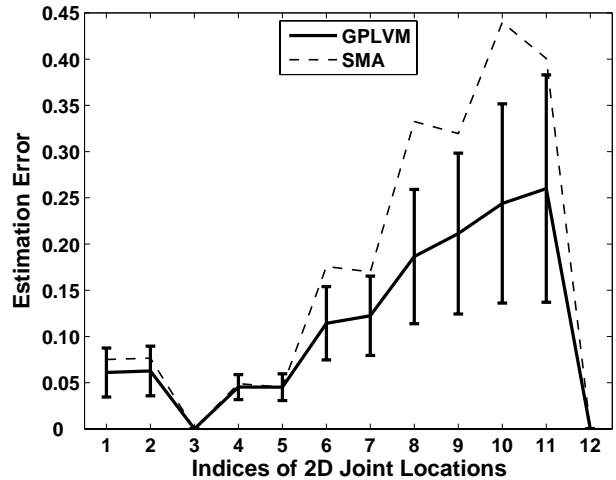


Figure 4: *Error Comparison of SMA vs. GPLVM (The errors are normalized by the length of neck (joint 3) to base spine (joint 12)).*

are synthesized by rendering poses (same set of poses used for training with a male character) of a female character from multiple viewpoints using Poser 5. The reason for using a character of different gender is to make the test data less like the training data, as a female character tends to have a different silhouette outline compared to a male character. Real life data is also used in the experiments. Due to the noise present in the video sequences, the real silhouette images are not as “clean” as the synthesized silhouette images. Figure 7 shows that our algorithm works well for real life data.

5.1 Synthetic Data

In the experiments with synthetic data, we compared our approach with that of SMA when trained with the same training data. 3000 synthesized silhouette images were used. Alt moments and 2D body poses have different scales in numeric value and to enforce C_{Alt} , the associated weight w_1 is set to 3000. The scalar w_2 is set to 30 for the constraint $C_{temporal}$. The weights are determined empirically.

To compare the performance of SMA and GPLVM, we first aligned the estimated poses with corresponding ground-truth poses by aligning the neck and base of the spine (joint 3 and joint 12 in Figure 2). This is to avoid any error introduced by scaling. Then we computed the mean squared error of the joint locations of the ground-truth poses and estimated poses. The quantitative comparison in terms of the joint location error is shown in Figure 4. It is clear our approach outperforms SMA, especially at arm joints and hand joints (joints 6-11). These joint locations convey the most information in 2D upper body pose. The error bars in the plot are the standard deviations of GPLVM

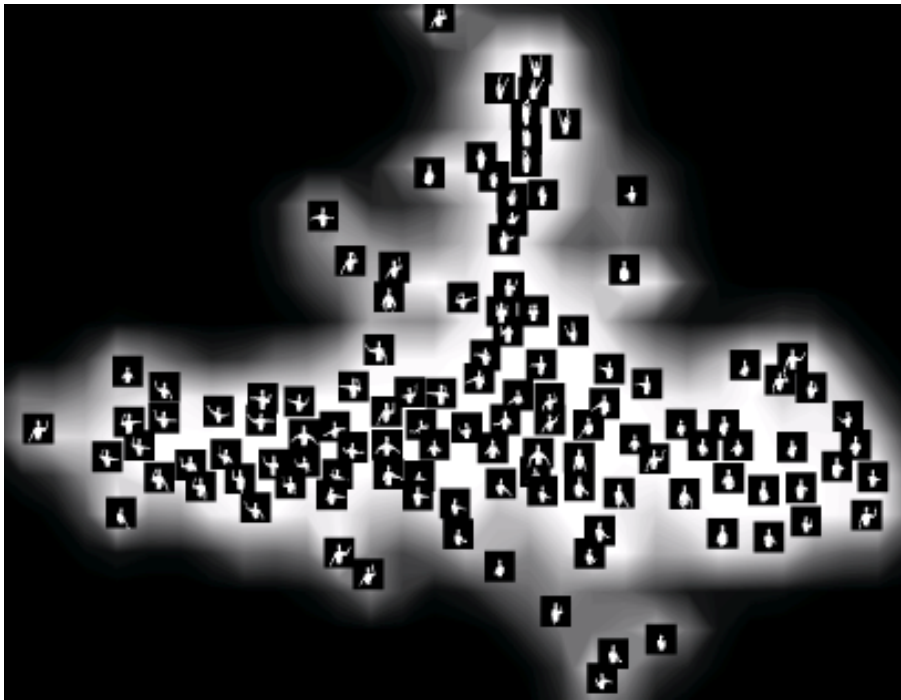


Figure 3: A portion of the GPLVM latent space (2D) of different upper body poses.

estimation errors at different joint locations. In terms of estimation accuracy at arm and hand joints (joints 6-11), SMA is at least one standard deviation worse.

In Figure 5, we show some examples of the estimation results. Qualitatively, GPLVM gives better or the same quality estimation of poses except in last column of Figure 5. It is difficult to judge the quality since both results are not as accurate.

Given a gesture sequence, we can make use of temporal consistency to improve the estimation results by assuming that the current pose should not differ from previous pose too much. The incorporation of this constraint is specified in Section 3.4. We tested the effectiveness of making use of temporal consistency on a few synthetic sequences. Figure 6 shows some frames from an “open wing” gesture sequence used in aircraft hand signals [1]. We can see that the results shown in row(d) of Figure 6, which is GPLVM with temporal consistency, captured the smooth transitions between different poses. Row (c) of Figure 6 shows the estimation results of GPLVM without using any temporal information; the transitions between different frames are not as smooth. The good results demonstrated here show the potential tracking applications of GPLVM.

5.2 Real Data

To demonstrate the robustness of our proposed estimation algorithm, we conducted experiments on 1000 silhouette

images from a captured video sequence of a human performing flight director gestures. The silhouette images in real videos are in general not as clean as the synthetic data. There are also incomplete silhouettes in the real data. Figure 7 shows some estimation results for real data. From the results, we can see that our algorithm produces qualitatively better results when compared with those obtained from SMA (row (b)). In row (c), even with incomplete silhouettes, the algorithm still produces reasonable results in the last two columns. Row (d) shows that by applying temporal consistency, we can improve the estimation result as shown in the second to last column.

6 Conclusions and Future Work

We propose a new learning framework to tackle the problem of estimating human body pose from a single image. Given the smooth parametrization obtained via GPLVM, our approach avoids the artificial “discretization” of SMA-like algorithms, where a few discrete functions have to be specified and the estimation is done by choosing the best among the multiple discrete outputs. Pose estimation can be made more accurate and efficient by incorporating proper constraints when appropriate. We solved the problem of 2D upper body pose estimation to show the strength of this framework. We expect the proposed framework can be applied to the 3D pose estimation problem by just adding the camera parameters (e.g. focal length, rotation and translation, etc.)

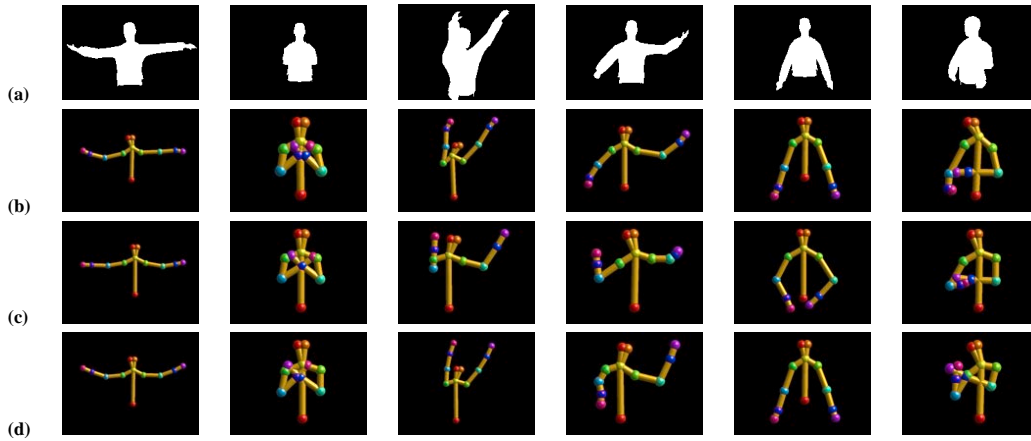


Figure 5: Experiment with synthetic data. (a)Input; (b)Ground-truth; (c)SMA; (d) GPLVM.

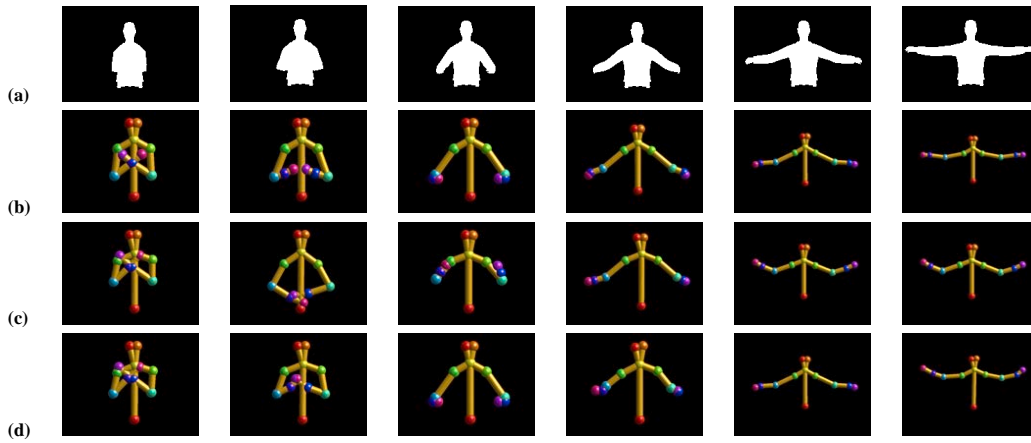


Figure 6: Experiment with synthetic data with temporal consistency. (a)Input; (b)Ground-truth; (c)GPLVM(without temporal consistency); (d) GPLVM(with temporal consistency).

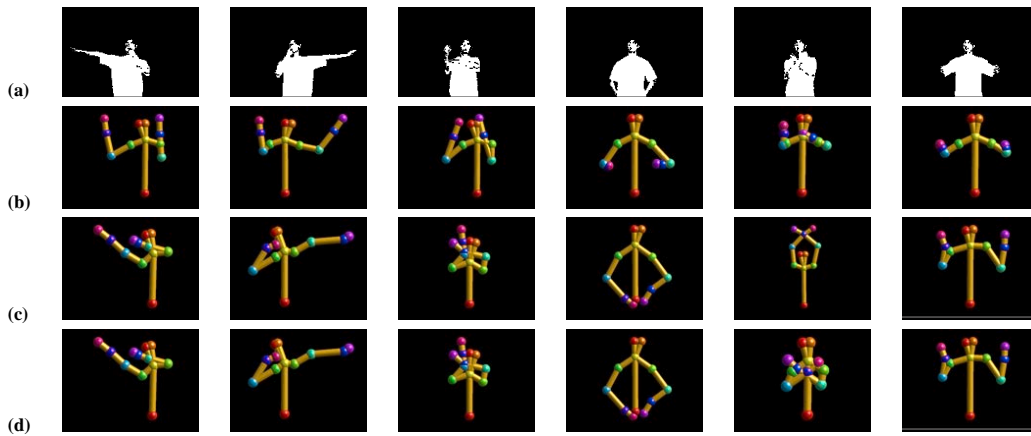


Figure 7: Experiment with real data. (a)Input; (b)SMA; (c) GPLVM(without temporal consistency); (d) GPLVM(with temporal consistency).

during the learning of the forward function Φ (Section 3.1).

Encouraging results have been obtained by incorporating temporal consistency. This naturally leads us extend our current work to the tracking problem in the near future.

A Appendix

The following gradients are used in the optimizing L :

$$\begin{aligned}\frac{\partial L}{\partial \mathbf{K}} &= -\mathbf{K}^{-1}\mathbf{Y}\mathbf{Y}^T\mathbf{K}^{-1} + \frac{1}{2}M\mathbf{K}^{-1}, \\ \frac{\partial k(\mathbf{x}, \mathbf{x}')}{\partial \mathbf{x}} &= -\gamma(\mathbf{x} - \mathbf{x}')k(\mathbf{x}, \mathbf{x}'), \\ \frac{\partial k(\mathbf{x}, \mathbf{x}')}{\partial \alpha} &= \exp\left(-\frac{\gamma}{2}(\mathbf{x} - \mathbf{x}')^T(\mathbf{x} - \mathbf{x}')\right), \\ \frac{\partial k(\mathbf{x}, \mathbf{x}')}{\partial \gamma} &= -\frac{1}{2}(\mathbf{x} - \mathbf{x}')^T(\mathbf{x} - \mathbf{x}')k(\mathbf{x}, \mathbf{x}').\end{aligned}$$

The following gradients are used in optimizing $L_{\tilde{\mathbf{y}}}$:

$$\begin{aligned}\frac{\partial L_{\tilde{\mathbf{y}}}}{\partial \mathbf{y}} &= -\mathbf{K}^{-1}\mathbf{Y}\mathbf{Y}^T\mathbf{K}^{-1} + \frac{1}{2}M\mathbf{K}^{-1}, \\ \frac{\partial \mu(\mathbf{x})}{\partial \mathbf{x}} &= \mathbf{Y}_{I,I}\mathbf{K}_{I,I} \frac{\partial \mathbf{k}_I(\mathbf{x})}{\partial \mathbf{x}}, \\ \frac{\partial \sigma^2(\mathbf{x})}{\partial \alpha} &= -2\mathbf{k}_I(\mathbf{x})^T\mathbf{K}_{I,I}^{-1} \frac{\partial \mathbf{k}_I(\mathbf{x})}{\partial \mathbf{x}}.\end{aligned}$$

References

- [1] *NAVAIR 00-80T-113 Aircraft Signals NATOPS Manual*.
- [2] A. Agarwal and B. Triggs. 3D human pose from silhouettes by relevance vector regression. In *CVPR*, 2004.
- [3] V. Athitsos and S. Sclaroff. Database indexing methods for 3D hand pose estimation. In *Proc. of the Gesture Workshop*, 2003.
- [4] C. Barron and I. Kakadiaris. Estimating anthropometry and pose from a single uncalibrated image. *Computer Vision and Image Understanding*, 81, 2001.
- [5] M. Brand. Shadow puppetry. In *ICCV*, 1999.
- [6] Curious Labs, Inc., Santa Cruz, CA. *Poser5 Reference Manual*, 2004.
- [7] M. N. Gibbs. *Bayesian Gaussian Processes for Regression and Classification*. PhD thesis, Univ. of Cambridge, 1997.
- [8] K. Grochow, S. L. Martin, A. Hertzmann, and Z. Popovic. Style-Based Inverse Kinematics. In *SIGGRAPH*, 2004.
- [9] N. Lawrence. Gaussian Process Latent Variable Models for Visualisation of High dimensional Data. In *NIPS*, 2004.
- [10] N. Lawrence. Probabilistic non-linear principal component analysis with Gaussian Process Latent Variable Models. Technical Report CS-04-8, Dept. of Computer Science, Univ. of Sheffield, 2004.
- [11] N. Lawrence, M. Seeger, and R. Herbrich. Fast sparse Gaussian process methods: The informative vector machine. In *NIPS*, 2003.
- [12] M. Lee and I. Cohen. Proposal maps driven MCMC for estimating human body pose in static images. In *CVPR*, 2004.
- [13] D. Mackay. Introduction to Gaussian processes. In C. Bishop, editor, *Neural Networks and Machine Learning*, NATO ASI Series, pages 133–166. Kluwer Academic Press, 1998.
- [14] V. Parameswaran and R. Chellapa. View independent human body pose estimation from a single perspective image. In *CVPR*, 2004.
- [15] R. Rosales and S. Sclaroff. Specialized mappings and the estimation of human body pose from a single image. In *IEEE Workshop on Human Motion*, 2000.
- [16] S. Roweis and L. Saul. Nonlinear dimensionality reduction by locally linear embedding. *Science*, 290, 2000.
- [17] G. Shakhnarovich, P. Viola, and T. Darrell. Fast pose estimation with parameter sensitive hashing. In *ICCV*, 2003.
- [18] C. Taylor. Reconstruction of articulated objects from point correspondences in a single uncalibrated image. *Computer Vision and Image Understanding*, 80(3):349–363, Dec 2000.
- [19] J. Tenenbaum, V. Silva, and J. Langford. A global geometric framework for nonlinear dimensionality reduction. *Science*, 290, 2000.
- [20] C. Williams. Prediction with Gaussian processes: From linear regression to linear prediction and beyond. In M. Jordan, editor, *Learning Graphical Models*, volume 89 of *Series D: Behavioural and Social Sciences*. Kluwer, 1998.