Latent Gaussian Mixture Regression for Human Pose Estimation

Yan Tian[1,3], Leonid Sigal[2], Hernán Badino[3], Fernando De la Torre[3], Yong Liu[1]

[1] Beijing University of Posts and Telecommunications, Beijing, P.R.China
[2] Disney Research, Pittsburgh, US
[3] Carnegie Mellon University, Pittsburgh, US

Abstract. Discriminative approaches for human pose estimation model the functional mapping, or conditional distribution, between image features and 3D pose. Learning such multi-modal models in high dimensional spaces, however, is challenging with limited training data; often resulting in over-fitting and poor generalization. To address these issues latent variable models (LVMs) have been introduced. Shared LVMs attempt to learn a coherent, typically non-linear, latent space shared by image features and 3D poses, distribution of data in that latent space, and conditional distributions to and from this latent space to carry out inference. Discovering the shared manifold structure can, in itself, however, be challenging. In addition, shared LVMs models are most often non-parametric, requiring the model representation to be a function of the training set size. We present a parametric framework that addresses these shortcoming. In particular, we learn latent spaces, and distributions within them, for image features and 3D poses separately first, and then learn a multi-modal conditional density between these two low-dimensional spaces in the form of Gaussian Mixture Regression. Using our model we can address the issue of over-fitting and generalization, since the data is denser in the learned latent space, as well as avoid the necessity of learning a shared manifold for the data. We quantitatively evaluate and compare the performance of the proposed method to several state-of-the-art alternatives, and show that our method gives a competitive performance.

1 Introduction

Monocular pose estimation has been a focus of much research in vision due to abundance of applications for marker-less motion capture in activity recognition and human computer interaction. Despite much research, however, monocular pose estimation remains a difficult task; challenges include high-dimensionality of the state space, image clutter, occlusions, lighting and appearance variations, to name a few.

Most prior works can be classified into two classes of approaches: *generative* and *discriminative*. *Generative* approaches [1, 2] define an image formation model by predicting appearance of the body $\mathbf{x}$ given a hypothesized state of the body (pose) $\mathbf{y}$; an inference framework is then used to infer the posterior,

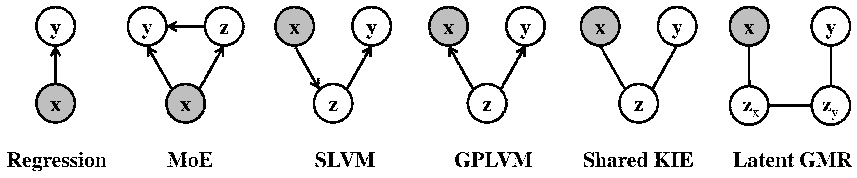

Fig. 1. Graphical model representations of models used for discriminative human pose estimation, including Regression Models [3, 13], Mixture Models (*e.g.*, Mixture of Experts (MoE) [4, 14]), Spectral Latent Variable Models (SLVM) [11], Gaussian Process Latent Variable Models [17, 12] and Shared Kernel Information Embeddings (sKIE) [18]. In all illustrations $\mathbf{x}$ denotes observed input variable corresponding to image features, $\mathbf{y}$ denotes the inferred 3D pose, and $\mathbf{z}$ corresponds to auxiliary latent variables (in case of Mixture of Gaussians (MoE) corresponding to the latent mixture component identity).

$p(\mathbf{y}|\mathbf{x}) \propto p(\mathbf{x}|\mathbf{y})p(\mathbf{y})$ over time. Since the inference often takes the form of non-convex search in a high-dimensional space of body articulations, these methods are computationally expensive and can suffer from local convergence (typically requiring a good initial guess for pose to seed the search).

Discriminative approaches [3, 16] avoid building an explicit imaging model, and instead opt to learn regression function, $\mathbf{y} = f(\mathbf{x})$, that maps from image features, $\mathbf{x}$, to 3D pose, $\mathbf{y}$; or probabilistically, a conditional distribution $p(\mathbf{y}|\mathbf{x})$ directly. The main goal is to learn a model from labeled training data, $\{\mathbf{x}^{(i)}, \mathbf{y}^{(i)}\}_{i=1}^{N}$, that provides efficient and effective generalization for new examples at test time. The difficulty with this class of methods is twofold: (1) the conditional probability of pose given image features, $p(\mathbf{y}|\mathbf{x})$, is typically multi-modal: different image features can be explained by several poses; and (2) learning high dimensional regression functions, or conditional distributions, using limited training data is challenging and often results in over-fitting. Here we focus on discriminative pose estimation.

Discriminative methods can further be categorized into: *parametric* and *non-parametric*. *Parametric* methods are appealing because the model representation is fixed[1]. Simple parametric models, *e.g.*, Linear Regression (LR) [3] or Relevance Vector Machine (RVM) [3, 10], however, are (i) unable to deal with a multi-modal nature of the problem and (ii) unable to model the fine non-linear relationship between image features and pose. *Non-parametric* methods, *e.g.*, Nearest Neighbor Regression [13] or Kernel Regression [13], are able to model arbitrary complex relationships between input features and output poses, subject to the availability of the training data.

To deal with multi-modality, on the parametric side, mixture models were introduced, *e.g.*, Mixture of Regressors [4] or Mixture of Experts [14]. On the non-parametric side, local models that cluster data into convex sets and use

[1] Complexity of the model is not a function of the number of training examples.

uni-modal predictions within each cluster became popular (*e.g.*, Local Gaussian Process Latent Variable Models (Local GPLVM) [16]). In both cases over-fitting and generalization remained an issue, due to the need for large training datasets, as noted in [12].

To alleviate this problem Latent Variable Models (LVMs) were introduced as an intermediate representation. Kanaujia *et al.*, [11], proposed Spectral LVMs to learn a non-linear latent embedding of the 3D pose data and a separately trained mixture model to map from the image features to the plausible latent positions in the sub-space. The relationship between the image features and latent space, however, was assumed to be linear within each mixture component. As an alternative, Shared Gaussian Processes Latent Variable Model (Shared GPLVM) was introduced in [12] and [17], where the latent embedding was learned to preserve the joint structure of image features and 3D poses simultaneously; the forward non-linear mappings from the latent space to the input and output spaces were also learned at the same time. Due to the lack of backward mapping from the image features to the latent space, inference remained expensive, requiring multiple optimizations at the cost of $O(N^2)$, where N is the number of training examples. Shared Kernel Information Embeddings (sKIE) [18] provided closed form mappings to and from the latent space reducing the training and inference complexity by an order of magnitude. Both Shared GPLVM and sKIE are non-parametric, with the model complexity being a function of the training set size; this makes them less appealing for use with larger datasets.

We present a parametric counterpart framework to the non-parametric latent models discussed above.

1. We learn a multi-modal joint density model between the image features and the 3D pose, in the form of a Gaussian Mixture Model (GMM). GMM allows us to deal with multi-modality in the data and derive explicit conditional distributions for inference, in the form of Gaussian Mixture Regression (GMR).
2. To alleviate the need for large training sets while at the same time limiting over-fitting, we formulate the GMM learning in the latent spaces for both image features and 3D pose.
3. Since the manifold structure of both image features and 3D poses is complex and cannot be well approximated by simple linear latent spaces, we propose to use Locality Preserving Projections (LPP) [19] that while learning linear mapping can discover non-linear manifold structure [19]. LPP also provides us with closed form forward and backward mappings between the latent space(s) and input/output space(s).

As a result our model is able to: (1) deal with multi-modalities in the data, (2) model complex structure of the image feature and pose manifolds, (3) provides both forward and backwards mapping between the respective manifolds and original image feature or pose spaces, and (4) alleviates the need for learning, a sometimes hard to obtain[2], shared manifold structure.

[2] Shared manifold structure can be hard to obtain, for example, if the input and output features have vastly different dimensionality.

2 Gaussian Mixture Regression

Non-parametric regression methods rely on manifold local smoothness in a typically high-dimensional input/output spaces to model the regression function; however, they can suffer from local sparsity problems. When the data is sparse (which is typically the case for high-dimensional spaces) and a test point is far from the training data, the kernels tend to produce poor estimates. In addition, the complexity of non-parametric methods is typically a function of the training set size (*e.g.*, $O(N)$ for KIE and $O(N^2)$ for GPLVM), making them hard to scale to large datasets. In this paper, we employ a parametric Gaussian Mixture Regression to address these problems.

Given observations (*e.g.*, image features), $\mathbf{x} \in \mathbb{R}^{d_\mathbf{x}}$, and targets (*e.g.*, 3D poses), $\mathbf{y} \in \mathbb{R}^{d_\mathbf{y}}$, where $d_\mathbf{x}$ is dimensionality of the observation, and $d_\mathbf{y}$ is dimensionality of the target space, we assume the joint data samples, $(\mathbf{x}, \mathbf{y})$, follow the Gaussian mixture distribution with K mixture components,

$$P(\mathbf{x}, \mathbf{y}) = \sum_{k=1}^{K} \pi_k P(\mathbf{x}, \mathbf{y}; \mu_k, \mathbf{\Lambda}_k) \tag{1}$$

where $P(\mathbf{x}, \mathbf{y}; \mu_k, \mathbf{\Lambda}_k)$ is the multivariate Gaussian density function. The parameters of model include prior weights, π_k, means, $\mu_k = [\mu_{k,\mathbf{x}}\ \mu_{k,\mathbf{y}}]^T$, and variances, $\mathbf{\Lambda}_k = [\mathbf{\Lambda}_{k,\mathbf{x}}\ \mathbf{\Lambda}_{k,\mathbf{xy}}; \mathbf{\Lambda}_{k,\mathbf{yx}}\ \mathbf{\Lambda}_{k,\mathbf{y}}]$, of each Gaussian component.

The joint density can be expressed as the sum of the products of the marginal density of $\mathbf{x}$, and the probability density function of $\mathbf{y}$ conditioned on $\mathbf{x}$:

$$P(\mathbf{x}, \mathbf{y}) = \sum_{k=1}^{K} \pi_k P(\mathbf{y}|\mathbf{x}; m_k, \sigma_k^2) P(\mathbf{x}; \mu_{k,\mathbf{x}}, \mathbf{\Lambda}_{k,\mathbf{x}}). \tag{2}$$

Similarly, the marginal distribution,

$$P(\mathbf{x}) = \sum_{\mathbf{y}} P(\mathbf{x}, \mathbf{y}) = \sum_{k=1}^{K} \pi_k P(\mathbf{x}; \mu_{k,\mathbf{x}}, \mathbf{\Lambda}_{k,\mathbf{x}}), \tag{3}$$

is also a mixture.

The global regression function can be obtained by combing (2) and (3):

$$P(\mathbf{y}|\mathbf{x}) = \frac{P(\mathbf{x}, \mathbf{y})}{P(\mathbf{x})} = \frac{\sum_{k=1}^{K} \pi_k P(\mathbf{x}; \mu_{k,\mathbf{x}}, \mathbf{\Lambda}_{k,\mathbf{x}}) P(\mathbf{y}|\mathbf{x}; m_k, \sigma_k^2)}{\sum_{k=1}^{K} \pi_k P(\mathbf{x}; \mu_{k,\mathbf{x}}, \mathbf{\Lambda}_{k,\mathbf{x}})} \tag{4}$$

This can be expressed as a mixture of conditional distributions, $P(\mathbf{y}|\mathbf{x}) = \sum_{k=1}^{K} \omega_k P(\mathbf{y}|\mathbf{x}; m_k, \sigma_k^2)$, where the mixing weights ω_k are defined as:

$$\omega_k = \frac{\pi_k P(\mathbf{x}; \mu_{k,\mathbf{x}}, \mathbf{\Lambda}_{k,\mathbf{x}})}{\sum_{j=1}^{K} \pi_j P(\mathbf{x}; \mu_{j,\mathbf{x}}, \mathbf{\Lambda}_{j,\mathbf{x}})}. \tag{5}$$

The mean and the variance of the conditional distribution $P(\mathbf{y}|\mathbf{x})$ can be acquired in closed form by:

$$m_k = \mu_{k,\mathbf{x}} + \mathbf{\Lambda}_{k,\mathbf{yx}}\mathbf{\Lambda}_{k,\mathbf{x}}^{-1}(\mathbf{x} - \mu_{k,\mathbf{x}}) \tag{6}$$

$$\sigma_k^2 = \mathbf{\Lambda}_{k,\mathbf{y}} - \mathbf{\Lambda}_{k,\mathbf{yx}}\mathbf{\Lambda}_{k,\mathbf{x}}^{-1}\mathbf{\Lambda}_{k,\mathbf{xy}} \tag{7}$$

The learning can be achieved with a simple Gaussian Mixture Model, using Expectation Maximization (EM) procedure with K-means initialization. The prediction given a new input can be obtained by computing expectation over $P(\mathbf{y}|\mathbf{x})$:

$$E[P(\mathbf{y}|\mathbf{x})] = \sum_{k=1}^{K} \omega_k m_k. \tag{8}$$

Alternatively, if the conditional relationship is truly multi-modal, it is better to look at the modes given by m_j directly. In general, we can have up to K distinct modes in the conditional distribution for a given input, $\mathbf{x}$.

Relationship to Other Methods. Notice that the regression function (8) derived from the joint mixture Gaussian density is of the form of a kernel estimator. However, there is a key difference with non-parametric regression: the mixture weights, ω_k, are not determined by the local structure of the data, but rather by the components of a global Gaussian mixture model.

The Nadaraya-Watson kernel smoother [20] is a Gaussian Mixture Regression model with $K = N$ components, where N is the total number of training points. At the other end of the spectrum, $K = 1$ is approximately the classical linear regression model. Hence, the Gaussian Mixture Regression model can, in principal, represent a spectrum of regression models, ranging from the non-parametric kernel regression, where $K = N$, to the classical linear regression, $K = 1$.

Mixture Gaussian Regression is also closely related to the Mixture of Regression model and Mixture of Experts model (with a particular form of experts and gaits). For more discussion of this, see [21], Section 2.2.3.

3 Latent GMR Body Pose Estimation

As described in the previous section, we could use image features for inputs and 3D poses for targets and learn a GMR model in the original high-dimensional space. This has two shortcomings, however: (1) this would involve estimation of large number of parameters and hence require lots of training data, and (2) this assumes essentially a piece-wise multi-linear relationship between image features and 3D pose. For these reasons, we postulate that learning GMR in the latent space of both, image features and pose, actually results in better generalization and overall quality of the model.

To test this assumption we run a simple illustrative experiment with canonical correlation analysis. Canonical correlation analysis (CCA) [22] is a technique to extract common features from a pair of multivariate data. CCA, first proposed by Hotelling in 1936 [23], finds linear basis vectors for two sets of variables, such

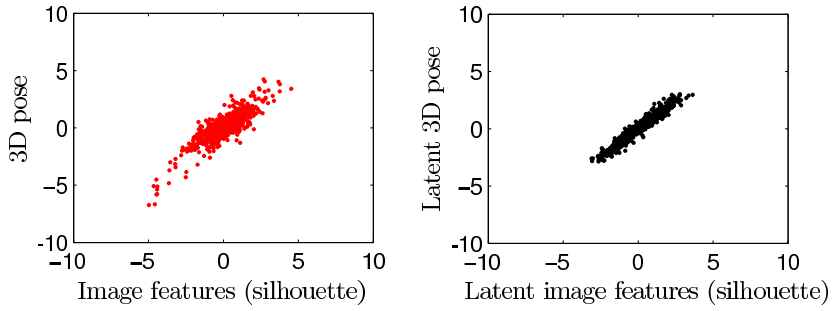

Fig. 2. Canonical component analysis of the silhouette and the human pose in the original and latent spaces (see text for more details).

that the correlation between the projections of variables onto these basis vectors are mutually maximized. We learn two CCA models based on 200 image-pose pairs: one for raw silhouette binary features $\in \mathbb{R}^{2450}$ and pose features encoded using 3D joint positions $\in \mathbb{R}^{69}$ (Fig. 2 (left)); and one for latent projections of the image and poses into 100 and 7 dimensional linear sub-spaces, obtained using PCA (Fig. 2 (right)); we illustrate only the first dimension along each of the axis in Fig. 2. It is clear from Fig. 2 that pose and image features are more closely correlated when projected into latent spaces (which reduces noise and optimally weights features). However, CCA is likely suffer from overfitting when having small training sets, and regularizing the solution introduces additional parameters to tune. Moreover, to model non-linear relations between image features and pose parameters kernel methods need to be applied, and it is unclear how to learn the functional form of the kernel and the kernel parameters specially in presence of limited training samples. In next section we propose to use Locality Preserving Projection (LPP) as an efficient and effective dimensionality reduction algorithm to capture the subtle manifold structure of the data. Additionally, LPP is not as prone to over-fitting and does not make assumptions about the global distribution of the data.

3.1 Locality Preserving Projections

Nonlinear dimensionality reduction techniques like Isomap [24], Locally Linear Embedding [10, 7], or Gaussian Process Latent Variable Models [12] identify a low dimensional embedding of the data, but are defined only for the training data points (*i.e.*, only give a mapping from the manifold to the original data space); it is unclear how to obtain a latent position for a new test points. This makes inference challenging, often involving optimization [12] of the latent position based on the initial guess given by a set of nearest neighbors in the original space.

In contrast, the Locality Preserving Projections (LPP) [19], like PCA, can be simply applied to any new data point to locate it in the reduced representation

space by finding the optimal linear approximations to the eigenfunctions of the Laplace Beltrami operator on the manifold. Therefore, we use LPP to find low-dimensional embeddings of both image features and 3D poses.

For example, given a training dataset of N poses, $\mathbf{Y} = \{\mathbf{y}^{(1)}, \mathbf{y}^{(2)}, ..., \mathbf{y}^{(N)}\} \in \mathbb{R}^{d_\mathbf{y} \times N}$, we want to find a transformation matrix $\mathbf{A} = [\mathbf{a}_1, ..., \mathbf{a}_{d_\mathbf{z}}]^T$ of basis vectors, $\mathbf{a}_i$, that maps these points to a set of latent points $\mathbf{Z} = \{\mathbf{z}^{(1)}, \mathbf{z}^{(2)}, ..., \mathbf{z}^{(N)}\} \in \mathbb{R}^{d_\mathbf{z} \times N}$ ($d_\mathbf{z} \ll d_\mathbf{y}$), such that $\mathbf{z}^{(i)}$ is a low dimensional manifold embedding representation of a high dimensional space pose $\mathbf{y}^{(i)}$. Following [19], this can be expressed as:

$$\begin{aligned} \min_{\mathbf{A}} \quad & \mathbf{A^T Y L Y^T A} \\ \text{subject to} \quad & \mathbf{A^T Y D Y^T A = I} \end{aligned} \tag{9}$$

Where $\mathbf{D}$ is a diagonal matrix whose entries are column sums of weight matrix $\mathbf{W}$, and $\mathbf{W}$ incurs a heavy penalty if neighboring training points are mapped far apart; $\mathbf{L = D - W}$ is Laplacian matrix.

3.2 Learning

Learning of the proposed model, is formulated as a three step procedure. Given a dataset of labeled feature-pose pairs, $\{\mathbf{x}^{(i)}, \mathbf{y}^{(i)}\}_{i=1}^N$, we: (1) learn a low-dimensional embedding of the 3D pose data, $\{\mathbf{y}^{(1)}, \mathbf{y}^{(2)}, ..., \mathbf{y}^{(N)}\}$, by solving optimization in Eq. 9; (2) learn a low-dimensional embedding of the image features by solving similar optimization for $\{\mathbf{x}^{(1)}, \mathbf{x}^{(2)}, ..., \mathbf{x}^{(N)}\}$; (3) learning a Gaussian Mixture Model (GMM) for the latent features and pose representations, $\{\mathbf{z}_\mathbf{x}^{(i)}, \mathbf{z}_\mathbf{y}^{(i)}\}_{i=1}^N$, obtained in (1) and (2).

3.3 Inference

Given a learned model, the inference for a new test image, represented in terms of image features $\hat{\mathbf{x}}$, involves: (1) getting a latent representation of $\hat{\mathbf{x}}$, $\hat{\mathbf{z}}_\mathbf{x}$, by applying a learned LPP mapping, $\mathbf{A}_\mathbf{x}$; (2) closed form conditioning of Gaussian Mixture Model (GMM), using $\hat{\mathbf{z}}_\mathbf{x}$, to obtain a Gaussian Mixture Regression (GMR) function; (3) inferring the latent 3D pose, $\hat{\mathbf{z}}_\mathbf{y}$, by either computing expectation over GMR (for uni-modal predictions) or using modes (for multi-modal predictions); (4) reconstructing the high-dimensional 3D pose from the latent estimate(s), by applying an inverse LPP mapping, $\mathbf{A}_\mathbf{y}$.

4 Experiments

4.1 Data sets

We test the performance of our method on three datasets: (1) Poser dataset – synthetic sequences produced by Poser software [25], (2) CMU dataset – real

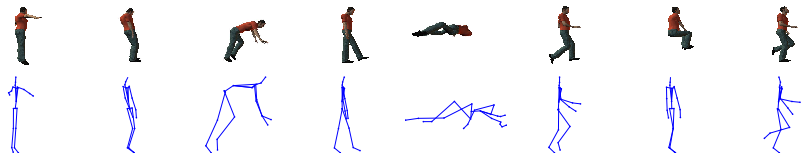

Fig. 3. Synthesized data generated by Poser 7 software.

image/mocap dataset publicly available from [26], and (3) standard dataset with provided error metrics made available by Agarwal and Triggs [3].

Poser dataset. We synthesize image data from motion capture sequences using Poser 7 software. The motion sequences come from 8 categories: walk, run, dance, fall, prone, sit, transitions and misc (see Fig. 3). A total of 5 sequences within each category are broken into: 3 training and 2 testing sequences, with each sequence containing approximately 500 frames. The size of each synthetic image is 500×490. We represent body pose in terms of 3D positions of 23 joints, resulting in $d_{\mathbf{y}} = 69$. All poses are represented in relative terms by subtracting the skeleton root (pelvis) from all other joint centers in every frame.

CMU dataset. From CMU Graphics Lab Motion Capture Database (see Fig. 4), we choose sequences of Subject 2 as training data, and use sequences in Subject 1, 8, 15 and 17 as test data. The size of each image is 240×352. We represent body pose in terms of 3D positions of 31 joints, resulting in $d_{\mathbf{y}} = 93$. Again, all poses are represented relative to the skeleton root (pelvis).

Image features. A number of representations for image features have been introduced over the years, *e.g.*, Scale invariant feature transform (SIFT) [15, 6] or histogram of shape context [27, 3, 4, 14], to name a few. Similar to prior work, we rely on silhouette features and encode them using a simpler 60D global shape context representation.

Error measure. We use a standard average joint position error inline as done by [18]. We report RMSE of average joint error in centimeters (cm).

Agarwal and Triggs dataset. To compare to other published techniques, we also utilize a publicly available benchmark dataset, that contains 1927 training and 418 test images, synthetically generated from mocap data. The pose is encoded using 54 joint angles in this case. The image features and error metric are provided with the dataset [3]. Silhouette features are represented using 100-dimensional feature vectors encoding the image silhouette using vector-quantized shape contexts. The mean RMSE error is computed over joint angles and is measured in degrees (for details see [3]).

4.2 Comparison

We compare our **Latent GMR** model with a number of alternatives, including: non-parametric regression model (kernel regression (**KR**)) and parametric regression models (linear regression (**LR**), mixture of linear regressors (**MLR**),

Error (cm)		KR	LR	MLR [4]	MoE [14]	GMR	Latent GMR
dance	S1	10.85	5.83	5.76	5.72	7.79	**5.60**
	S2	10.37	5.23	5.10	5.04	6.23	**4.91**
falls	S1	15.32	10.40	10.27	10.25	10.82	**10.05**
	S2	16.31	11.50	11.32	11.26	12.99	**10.92**
miscs	S1	8.32	3.59	3.53	3.42	3.86	**3.28**
	S2	19.27	12.19	12.11	12.10	14.44	**11.80**
prone	S1	11.36	6.55	6.46	6.40	7.06	**5.88**
	S2	12.46	6.36	6.32	6.28	7.00	**6.19**
run	S1	8.94	4.70	4.65	4.64	6.23	**4.31**
	S2	11.65	5.96	5.85	5.79	6.89	**5.62**
sit	S1	18.56	13.29	13.24	13.20	14.14	**13.01**
	S2	9.65	4.92	4.87	4.81	6.03	**4.24**
transition	S1	11.23	5.78	5.50	5.44	6.17	**5.32**
	S2	10.65	6.07	5.92	5.91	6.50	**5.81**
walk	S1	11.65	6.36	6.18	6.06	6.71	**5.93**
	S2	9.15	3.55	3.38	3.34	3.73	**3.15**
Average		12.23	7.01	6.90	6.85	7.91	**6.62**

Table 1. Evaluation of different algorithms on the Poser dataset (for details see text).

Error (cm)		KR	LR	MLR [4]	MoE [14]	GMR	Latent GMR
Subject 1	01-01	18.27	16.18	2.80	15.79	17.76	**14.49**
	01-05	22.27	23.01	22.05	21.77	20.36	**18.49**
	01-08	32.88	34.74	34.46	34.12	35.02	**32.76**
Subject 8	08-02	19.00	13.93	13.43	13.14	15.00	**11.78**
	08-03	17.59	19.95	19.34	19.17	19.90	**18.66**
	08-04	16.69	22.22	18.55	18.33	18.10	**15.45**
Subject 15	15-06	20.24	13.64	13.31	13.25	14.26	**13.14**
	15-11	15.90	14.26	13.81	13.59	14.88	**13.42**
	15-13	28.82	23.18	23.12	23.01	27.46	**22.95**
Subject 17	17-03	29.49	25.49	24.02	23.68	26.73	**23.23**
	17-05	21.43	13.93	13.70	13.42	15.96	**12.01**
	17-07	21.43	15.84	15.05	14.77	16.69	**14.55**
Average		21.99	19.68	18.83	18.61	20.13	**17.54**
Train time		0	0.06	75	79.18	17.82	19.38
Test time		10.28	0.02	0.16	0.17	14.89	1.14

Table 2. Evaluation of different algorithms in CMU motion capture database; the learning and inference time is also given in (*seconds*).

mixture of experts (**MoE**), and Mixture Gaussian Regression (**MGR**)) in the originals high-dimensional space. The results are shown in Table 1 and Table 2. We use the same training and test datasets for all methods, and we also use a fixed set of parameters, for all sequences. For example, we train all the mixture models with $K = 8$ components. Other parameters are chosen by cross-validation: *e.g.*, the width of the RBF kernel in KR. In our method, the Locality Preserving Projections (LPP) is trained to keep 95% of the original energy. The results for [4, 14] in Table 1 and Table 2 are based on re-implementations of the original work[3]. In all cases we compare the *expectations* computed under the models with ground truth.

[3] For the purpose of comparison, we do not explore the temporal prior which is employed in [10].

Error (cm)	KR	LR	MLR [4]		MoE [14]		GMR	
			Exp	B8	Exp	B8	Exp	B8
Orig. Space	37.98	25.69	26.73	23.40	25.51	23.26	29.75	20.74
PCA	43.21	35.18	25.04	22.33	24.81	22.22	25.14	16.37
LPP	43.57	22.75	22.70	21.45	22.61	21.29	23.15	**12.79**

Table 3. Detailed experiments on CMU motion capture database. We train on Subject 17, Sequences 01–05 and test on Subject 17, Sequences 07–10.

We can see that since our features and data are sparse, kernel regression (KR) tends to work poorly in these cases. The performance of mixture models degrades as the data points start to fall close to the boundary between the two experts (since we are using expectation for inference). For this reason, sometimes the performance of mixture models is lower than that of uni-modal linear regression. Our Latent GRM model tends to produce better performance than competing methods.

Since the proposed Latent GMR contains two parts, *i.e.*, latent representation for the data and GMR model for inference, we attempt to study the interplay of both by running additional experiments on sub-set of data. In addition to prior experiments, we test PCA, as an alternative to LPP, for latent representation and a variety of regression models for inference within the latent spaces. We also show the performance of the expectation (Exp) as well as of multi-hypothesis mode prediction (B8) (assuming existence of oracle that chooses among the 8 mixture components). The results are illustrated in Table 3.

Based on Table 3 we make the following 4 observations: (1) inference in the latent space is nearly always better than in the original space; (2) LPP outperforms PCA in terms ability to preserve the manifold structure; (3) LR, MLR, MoE and GMR perform similarly on uni-modal prediction task; and (4) GMR outperforms all other methods with multiple predictions. We belive that (4) is due to the ability of GMR to generatively model the full density over the latent features and poses (as opposed to other more direct regression methods).

Finally, to compare to published methods we run on Agarwal and Triggs dataset [3], where we achieve error of 6.71 degrees, which is better the Nearest Neighbor regression and Linear Regression as reported in [17] and [18] respectively. However, we cannot match the performance of non-parametric shared LVMs, like Shared GPLVM and Shared KIE, that achieve errors of 6.50 and 5.95 degrees respectively. This is not surprising given that non-parametric models can represent more complex manifold structure; however, they do come at a cost of inference and learning which, unlike in our method, are a function of the training set size.

5 Conclusions and Future Work

In this paper, we present a parametric discriminative framework for 3D pose inference. Our model has a number of appealing properties, mainly: (1) it can

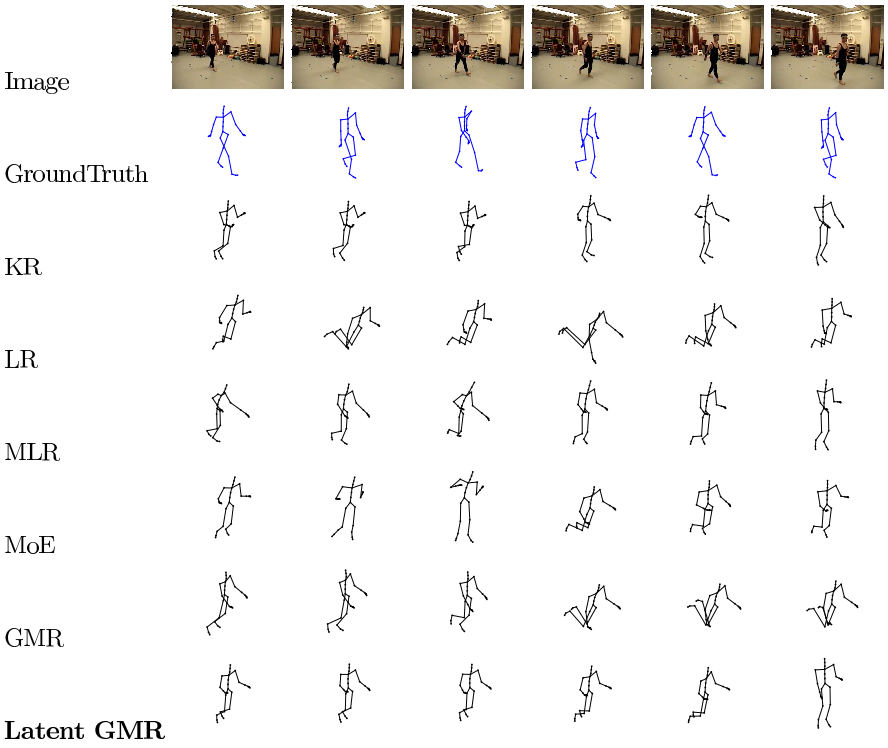

Fig. 4. Evaluation on frames: 38, 48, 58, 68, 78, and 88 of the 08-04 sequence from the CMU motion capture database.

deal with multi-modalities in the data, (2) model complex structure of the image feature and pose manifolds, (3) provides both forward and backwards mapping between the respective manifolds and original image feature or pose spaces, simplifying the inference and (4) alleviates the need for learning, a sometimes hard to obtain shared non-linear manifold structure. We show that our performance is comparative or superior to parametric and non-parametric models in the original high-dimensional space. In the future, we intend to look at learning the model in a unified manner through a single (as opposed to stage-wise) learning procedure.

References

1. Sidenbladh, H., Black, M., Fleet, D.: Stochastic tracking of 3d human figures using 2d image motion. In: ECCV. (2000)
2. Sminchisescu, C., Triggs, B.: Covariance scaled sampling for monocular 3d body tracking. In: CVPR. (2001)
3. Agarwal, A., Triggs, B.: 3d human pose from silhouettes by relevance vector regression. In: CVPR. (2004)
4. Agarwal, A., Triggs, B.: Monocular human motion capture with a mixture of regressors. In: CVPR. (2005)

5. Bissacco, A., Yang, M., Soatto, S.: Fast human pose estimation using appearance and motion via multi-dimensional boosting regression. In: CVPR. (2007)
6. Bo, L., Sminchisescu, C.: Structured output-associative regression. In: CVPR. (2009)
7. Elgammal, A.M., Lee, C.S.: Inferring 3d body pose from silhouettes using activity manifold learning. In: CVPR. (2004)
8. Fathi, A., Mori, G.: Human pose estimation using motion exemplars. In: ICCV. (2007)
9. Guo, F., Qian, G.: Learning and inference of 3d human poses from gaussian mixture modeled silhouettes. In: ICPR. (2006)
10. Jaeggli, T., Koller-Meier, E., Van Gool, L.: Learning Generative Models for Multi-Activity Body Pose Estimation. International Journal of Computer Vision **83** (2009) 121–134
11. Kanaujia, A., Sminchisescu, C., Metaxas, D.: Spectral latent variable models for perceptual inference. In: ICCV. (2007)
12. Navaratnam, R., Fitzgibbon, A., Cipolla, R.: The Joint Manifold Model for Semi-supervised Multi-valued Regression. In: ICCV. (2007)
13. Shakhnarovich, G., Viola, P., Darrell, T.: Fast pose estimation with parameter-sensitive hashing. In: ICCV. (2003)
14. Sminchisescu, C., Kanaujia, A., Li, Z., Metaxas, D.: Discriminative density propagation for 3d human motion estimation. In: CVPR. (2005)
15. Sminchisescu, C., Kanaujia, A., Metaxas, D.: Learning joint top-down and bottom-up processes for 3d visual inference. In: CVPR. (2006)
16. Urtasun, R., Darrell, T.: Sparse probabilistic regression for activity-independent human pose inference. In: CVPR. (2008)
17. Ek, C., Torr, P., Lawrence, N.: Gaussian process latent variable models for human pose estimation. In: Workshop on Machine Learning and Multimodal Interactions. (2007)
18. Sigal, L., Memisevic, R., Fleet, D.J.: Shared Kernel Information Embedding for Discriminative Inference. In: CVPR. (2009)
19. He, X., Niyogi, P.: Locality preserving projections. In: NIPS. (2003)
20. Nadaraya, E.: On estimation regression. Theory of Probability and its Applications **9** (1964) 141–142
21. Sminchisescu, C., Kanaujia, A., Li, Z., Metaxas, D.: Learning to reconstruct 3d human motion from bayesian mixtures of experts: A probabilistic discriminative approach. Technical Report CSRG-502, University of Toronto (2004)
22. Thorndike, R.: Canonical correlation analysis. Applied multivariate statistics and mathematical modeling (2000) 237–263
23. Hotelling, H.: Relations between two sets of variates. Biometrika **28** (1936) 321–377
24. Tian, T., Li, R., Sclaroff, S.: Articulated pose estimation in a learned smooth space of feasible solutions. In: Worshop on Learning in Computer Vision and Pattern Recognition, San Diego. (2005)
25. E–frontier. Curious Labs Poser. Computer Software.
26. CMU Motion Capture Database. http://mocap.cs.cmu.edu/.
27. Sigal, L., Balan, A., Black, M.: Combined discriminative and generative articulated pose and non-rigid shape estimation. In: NIPS. (2007)