Gaussian Process Latent Variable Models for Human Pose Estimation

Carl Henrik Ek[1], Philip H.S. Torr[1], and Neil D. Lawrence[2]

[1] Oxford Brookes University, Department of Computing, United Kingdom
cek@brookes.ac.uk, philiptorr@brookes.ac.uk
http://cms.brookes.ac.uk/research/visiongroup/
[2] University of Manchester, School of Computer Science, United Kingdom
neill@cs.man.ac.uk
http://www.cs.man.ac.uk/~neill/

Abstract. We describe a method for recovering 3D human body pose from silhouettes. Our model is based on learning a latent space using the Gaussian Process Latent Variable Model (GP-LVM) [1] encapsulating both pose and silhouette features Our method is generative, this allows us to model the ambiguities of a silhouette representation in a principled way. We learn a dynamical model over the latent space which allows us to disambiguate between ambiguous silhouettes by temporal consistency. The model has only two free parameters and has several advantages over both regression approaches and other generative methods. In addition to the application shown in this paper the suggested model is easily extended to multiple observation spaces without constraints on type.

1 Introduction

We consider the problem of estimating 3D articulated human pose from monocular silhouettes. Silhouettes are commonly used for pose estimation [2,3,4,5,6] as they contain strong cues for pose while at the same time being invariant to texture and lighting. Pose estimation from silhouettes is difficult because of inherent ambiguities leading to a one to many mapping from silhouette to pose. These ambiguities can be split into two types, (i) mis-labeling or limb flips and (ii) out of plane rotations. The first type appears for in plane rotations when lack of occlusion cues makes it hard to differentiate between limbs. The out-of-plane ambiguities appear when the subject is facing the view plane: the perspective distortions do not give strong enough cues to disambiguate limbs position out-of-plane. Algorithms with silhouette inputs need to handle these ambiguities.

There are two lines of work on pose estimation from silhouettes, (i) methods modeling the silhouette as a generative process from pose [4,6,7], (ii) methods based on regression from image observations to pose [2,5,8,9]. Generative methods model the space of silhouettes as a function of pose. This will correctly reflect the structure of the problem as each silhouette could have been generated by several different poses but each pose can only generate one single silhouette. The

A. Popescu-Belis, S. Renals, and H. Bourlard (Eds.): MLMI 2007, LNCS 4892, pp. 132–143, 2007.

problem arises when trying to infer pose from silhouette as, due to the multi-modality no inverse functional mapping from silhouette to pose exists. Finding each mode in pose space for a given silhouette is often very complicated due to the high-dimensionality of the pose space, even approximative methods like particle filters will be very expensive as a very large number of particles are needed to explore the high-dimensional pose space.

Regression based techniques try to model the pose space as a function of silhouettes. However due to the multi-modality no such functional exists. To overcome this problem it has been suggested to divide the silhouette space into subspaces for which functionals exist [2,5]. The pose space can then be described as a mixture of these single regressors. The structure of such a mixture has to reflect the multi-modality such that a *one-to-one* mapping exist between silhouette and pose for each subspace. In [2] the mixture centers region of support are decided by clustering in pose space. This is based on the assumption that ambiguities will occur between poses that have a significantly different joint angle configuration.

In [5] clustering is initially done in silhouette feature space then each cluster is split into several sub-clusters based on their corresponding poses. A clustering approach will not resolve all the ambiguities as it is based on the assumption that ambiguous silhouettes are "clearly" separated in pose space. This is only true for a small subset of ambiguities as especially the out-of-plane type occurs for continuous ranges in pose space. The final number of regressors will need to be decided based on some heuristic assumption about the occurrence of ambiguities or an error measure. There is a trade-off between generalization and training error as the minimal error would be given if each pose were to be represented by a separate regressor, but this would remove all generalizing capabilities of the model.

In this paper we take a learning based approach where we model both silhouette observations, joint angles and their dynamics as generative models from a shared low dimensional latent representations using the GP-LVM [1]. In line with other work on pose estimation [2,3,4,5,6] we have chosen to represent each image by its silhouette. As in [2,5] each silhouette is represented using shape context histograms [10]. We subsample each contour with one pixel spacing, acquiring about $100 - 150$ histograms for each image. To reduced the dimensionality of the descriptor and remove the the the effects of ordering we vector quantize the histograms using K-means clustering as described in [11], resulting in a 100D silhouette descriptor.

As described above a generative process will correctly handle the multi-modality between silhouette and pose. As we are learning a low-dimensional representation of pose we are not forced to fall-back on approximative methods for solving the inverse of this generative mapping. Our latent representation reflects the dynamics of of the data and can therefore predict poses over time in simple manner. The model requires no manual initialization when predicting sequential data but automatically initializes from training data.

2 Gaussian Processes

Gaussian Processes (GP) [12] are generalizations of Gaussian distributions defined over infinite index sets. Thereby a GP can be used to specify distribution over functions. It is completely defined by its mean function $\mu(\mathbf{x}_i)$, which is often taken to be zero, and its covariance function $k(\mathbf{x}_i, \mathbf{x}_j)$. The covariance function k characterizes the nature of the functions that can be sampled from the process. One widely used covariance function is

$$k(\mathbf{x}_i, \mathbf{x}_j) = \theta_1 e^{-\frac{\theta_2}{2}||\mathbf{x}_i - \mathbf{x}_j||^2} + \theta_3 + \beta^{-1}\delta_{ij}, \tag{1}$$

where the parameters are given by $\Phi = \{\theta_1, \theta_2, \theta_3, \beta\}$ and δ_{ij} is Kronecker's delta function. This covariance function combines an RBF function, a bias and a white-noise term. The parameters Φ of the covariance function k will be referred to as the hyper-parameters of the GP.

2.1 Prediction

By definition of a GP any finite number of variables specified by the process will have a joint Gaussian distribution [12]. For regression $y_i = f(\mathbf{x}_i) + \epsilon$, with noise $\epsilon \sim N(0, \beta^{-1})$, where $y_i \in \Re$ and $\mathbf{x}_i \in \Re^q$ placing a GP prior with zero mean and covariance function $k(x_i, x_j)$[1] over f, leads to the joint distribution,

$$\begin{bmatrix} \mathbf{y} \\ y_* \end{bmatrix} \sim N\left(\mathbf{0}, \begin{bmatrix} \mathbf{K} & \mathbf{K}_* \\ \mathbf{K}_*^\mathbf{T} & k(\mathbf{x}_*, \mathbf{x}_*) \end{bmatrix}\right) \tag{2}$$

of a set of observed data $\{\mathbf{x}_i, y_i\}_{i=1}^N$ and an unseen point $\mathbf{x}_*$, where $K_{ij} = k(\mathbf{x}_i, \mathbf{x}_j)$. Conditioning on the observed data leads to a posterior distribution over functions. From this posterior we obtain the predictive equations of a GP for an unseen point $\mathbf{x}_*$,

$$\overline{y}_* = k(\mathbf{x}_*, \mathbf{X})\mathbf{K}^{-1}\mathbf{Y} \tag{3}$$
$$\sigma_*^2 = k(\mathbf{x}_*, \mathbf{x}_*) - k(\mathbf{x}_*, \mathbf{X})^T\mathbf{K}^{-1}k(\mathbf{x}_*, \mathbf{X}), \tag{4}$$

where $\mathbf{X} = [\mathbf{x}_1, \ldots, \mathbf{x}_N]^T$, $\mathbf{Y} = [y_1, \ldots, y_N]^T$, $\overline{y}_*$ is the mean prediction and σ_*^2 is the variance.

2.2 GP Training

By maximizing the marginal likelihood over functions f,

$$p(\mathbf{Y}|\mathbf{X}, \Phi) = \int p(\mathbf{Y}|f, \mathbf{X}, \Phi)p(f|\mathbf{X}, \Phi)df \tag{5}$$
$$p(f|\mathbf{X}, \Phi) = N(\mathbf{0}, \mathbf{K}),$$

[1] Including a white-noise term with variance β^{-1}.

the hyper-parameters Φ of the GP can be learned from the observed data. This is referred to as training in the GP framework. It might seem undesirable to optimize over the hyper-parameters as the model might over-fit the data[2] Inspection of the logarithm of equation (5),

$$\log p(\mathbf{Y}|\mathbf{X}) = \underbrace{-\frac{1}{2}\mathrm{tr}\left(\mathbf{Y}^T\mathbf{K}^{-1}\mathbf{Y}\right)}_{data-fit} - \underbrace{\frac{1}{2}\log|\mathbf{K}|}_{complexity} - \frac{N}{2}\log 2\pi, \tag{6}$$

shows two "competing terms", the data-fit and the complexity term. The complexity term measures and penalizes the complexity of the model, while the data-fit term measures how well the model fits the data. This "competition" encourages the GP model not to over-fit the data.

3 GP-LVM

Lawrence [13] proposed an algorithm for dimensionality reduction using Gaussian Processes called the Gaussian Process Latent Variable Model (GP-LVM). The GP-LVM is a generative model where each observed data point, $\mathbf{y}_i \in \Re^D$, is generated through a noisy process from a latent variable $\mathbf{x}_i \in \Re^q$,

$$\mathbf{y}_i = f(\mathbf{x}_i) + \epsilon, \tag{7}$$

where $\epsilon \sim N(\mathbf{0}, \beta^{-1}\mathbf{I})$. Placing a zero mean GP-prior on the generative function f the marginal likelihood $P(\mathbf{Y}|\mathbf{X}, \Phi)$ can be formulated by integration over f,

$$P(\mathbf{Y}|\mathbf{X}, \Phi) = \prod_{j=1}^{D} \frac{1}{(2\pi)^{\frac{N}{2}}|\mathbf{K}|^{\frac{1}{2}}} e^{-\frac{1}{2}\mathbf{y}_{:,j}^T\mathbf{K}^{-1}\mathbf{y}_{:,j}}, \tag{8}$$

where $\mathbf{y}_{:,j}$ is the jth column from the data matrix, $\mathbf{Y}$. The GP-LVM maximizes the marginal likelihood (8) with respect to both the latent points $\mathbf{X}$ and the hyper-parameters Φ of the covariance function,

$$\{\hat{\mathbf{X}}, \hat{\Phi}\} = \mathrm{argmax}_{\mathbf{X},\Phi} P(\mathbf{Y}|\mathbf{X}, \Phi). \tag{9}$$

In general[3] there is no closed form solution for (9) and we must turn to gradient based optimization to make progress. The only parameter of the GP-LVM that can not be found through maximum likelihood is the dimensionality of the latent space, q, which must be set by hand.

[2] By setting the noise variance β^{-1} to zero the function f will pass exactly through the observed data $\mathbf{Y}$.

[3] An exception is when the linear kernel is used, in which case the optimization becomes an eigenvalue problem [1].

3.1 Back Constrained GP-LVM

Using a smooth covariance function the GP-LVM will specify a smooth mapping from the latent space $\mathbf{X}$ to the observation space $\mathbf{Y}$, this means that points close in the latent space will be close in the observed space. Having a smooth generative mapping does not imply that an inverse functional mapping exists.

Recently Lawrence and Quiñonero Candela [14] proposed an extension to the GP-LVM where the model is constrained by representing each latent point as a smooth parametric mapping from its corresponding observed data point, $\mathbf{x}_i = g(\mathbf{y}_i, \mathbf{W})$, where $\mathbf{W}$ is the mapping parameter set. This constrains points that are close in the observed space to also be close in the latent space. The mapping from observed data $\mathbf{Y}$ to $\mathbf{X}$ will be referred to as back-constraint. Including a back-constraint in the GP-LVM model changes the maximization in equation (9) from optimization with respect to the latent variables $\mathbf{X}$ to optimizing the parameters of the back-constraining mapping,

$$\{\hat{\mathbf{W}}, \hat{\Phi}\} = \text{argmax}_{\mathbf{W},\Phi} P(\mathbf{Y}|\mathbf{W}, \Phi). \tag{10}$$

3.2 GP Dynamics

For embedding sequential data Wang *et. al.* [15] proposed an extension to the GP-LVM to find a latent space that would reflect the ordering of the observed data. This is done by specifying a predictive function over the sequence in latent space,

$$\mathbf{x}_t = h(\mathbf{x}_{t-1}) + \epsilon_{dyn}, \tag{11}$$

where $\epsilon_{dyn} \sim N(\mathbf{0}, \beta_{dyn}^{-1}\mathbf{I})$. A GP prior can then be placed over the function $h(\mathbf{x})$. Marginalizing this mapping results in a distribution over the latent points which, through combination with the marginalized likelihood for the GP-LVM, specifies a new objective function,

$$\begin{aligned}\{\hat{\mathbf{X}}, \hat{\Phi}_Y, \hat{\Phi}_{dyn}\} = \text{argmax}_{X,\Phi_Y,\Phi_{dyn}} \\ P(\mathbf{Y}|\mathbf{X}, \Phi_Y) P(\mathbf{X}|\Phi_{dyn}).\end{aligned} \tag{12}$$

4 GP-LVM for Pose Estimation

The aim of our model is to learn a shared latent representation $\mathbf{X} = [\mathbf{x}_1, \ldots, \mathbf{x}_N]^T$ that relates corresponding pairs of feature $\mathbf{Y} = [\mathbf{y}_1, \ldots, \mathbf{y}_N]^T$ and pose $\mathbf{Z} = [\mathbf{z}_1, \ldots, \mathbf{z}_N]^T$. In [16] a shared latent structure between two joint angle spaces, one corresponding to a humanoid robot and the other corresponding to a human is learned. This is done by modifying the GP-LVM to learn separate sets of Gaussian Processes to each of the different observation spaces from a shared latent space. The latent representation is found by maximizing the joint marginal likelihood of the two observation spaces,

$$P(\mathbf{Y}, \mathbf{Z}|\mathbf{X}, \Phi_s) = P(\mathbf{Y}|\mathbf{X}, \Phi_Y) P(\mathbf{Z}|\mathbf{X}, \Phi_Z), \tag{13}$$

where $\Phi_s = \{\Phi_Y, \Phi_Z\}$. We want to learn a latent structure that preserves local distances from the pose space. This can be achieved by incorporating a back-constraint from the pose space onto the latent space by representing the latent points as a function of the pose. Incorporating a back-constraint from pose implies we are trying to enforce a *one-to-one* mapping between the pose space and the latent space. This is desirable as it will force the mapping from latent to feature to be *many-to-one* which means that we have contained the multi-modality of the system to this mapping.

To back-constrain the latent space, we represent the latent points by a regression over a kernel induced feature space that allows for non-linearities,

$$\mathbf{x}_i = \sum_{j=1}^{N} w_j \phi(\mathbf{z}_i, \mathbf{z}_j) \tag{14}$$

$$\phi(\mathbf{z}_i, \mathbf{z}_j) = e^{-\frac{\gamma}{2}(\mathbf{z}_i - \mathbf{z}_j)^T(\mathbf{z}_i - \mathbf{z}_j)} \tag{15}$$

This leads to a modified objective where the positions of the latent variables are optimized indirectly by maximizing $P(\mathbf{Y}, \mathbf{Z}|\mathbf{W}, \Phi_Y, \Phi_Z) = P(\mathbf{Y}|\mathbf{W}, \Phi_Y)P(\mathbf{Z}|\mathbf{W}, \Phi_Z)$ with respect to the parameters of the back-constraint. The latent representation is shared by the feature space and the pose space. By back-constraining the pose space we are encouraging the mapping between the latent space and pose space to be *one-to-one*, thereby forcing the GP-LVM from latent space to silhouette features to be *many-to-one* to reflect the ambiguities in the silhouette features.

4.1 Dynamical Model

Many of the pose ambiguities from our silhouette representation can be resolved by considering sequential data. Including a model that can predict poses over time allows us to resolve ambiguous silhouettes by temporal consistency. However, by learning a latent representation we can do even better. We can incorporate the dynamic model when learning the latent representation, forcing the latent representation to respect the data's dynamics. This can be done by specifying a GP over the latent space as in [15], incorporating this within our back-constrained, shared latent space representation leads to the following objective,

$$P(\mathbf{Y}, \mathbf{Z}, \mathbf{W}|\Phi) = P(\mathbf{Y}, \mathbf{Z}|\mathbf{W}, \Phi_s)P(\mathbf{W}|\Phi_{dyn}),$$

where $\Phi = \{\Phi_s, \Phi_{dyn}\}$. Sequences of pose usually form locally smooth but globally complex trajectories through joint angle space. This makes it difficult to fit a dynamic model when pose is represented as joint angles. Learning a dynamical model jointly with the latent representation is beneficial as the non-linear mapping from latent space to pose space allows for a significantly different structure for the latent representation and the joint angle representation. We use this property to smoothly[4] arrange the latent representation according to the dynamics

[4] Due to the smooth covariance function in the dynamic GP.

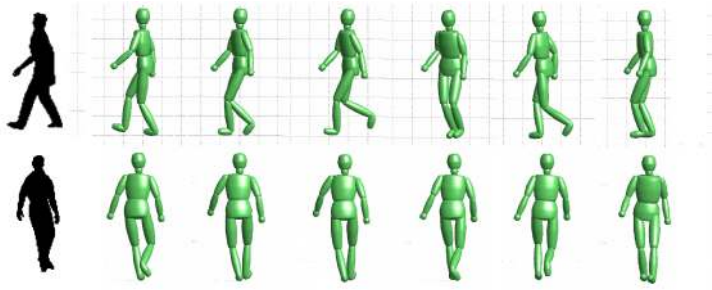

Fig. 1. Single Image Pose Estimation: Input silhouette followed by output poses associated with modes on the latent space ordered according to decreasing likelihood

of the data. Along with the dynamics our model also contains a back-constraint. The back-constraint will encourage that the local smoothness of the joint angle trajectories is preserved. Our experiments show that the structure of the latent space changes significantly when incorporating the dynamics, this difference is evidence of the complex nature of trajectories through joint angle space.

4.2 Model Summary

Training of the model implies that we are finding a shared latent representation $\mathbf{X}$ of both the observation spaces $\mathbf{Y}$ (the silhouette features) and $\mathbf{Z}$ (the pose angles), learning two sets of GP regressors from the latent space, $\mathbf{X}$, to reconstruct each of the observation spaces. Additionally the latent space incorporates a set of GPs which give temporal predictions. We are also learning a parametric mapping from the pose space, $\mathbf{Z}$, to the latent space, $\mathbf{X}$, thereby enforcing the latent space to preserve the local similarities of the pose space. All the remaining parameters of the model, except two, are found through maximum likelihood. The two remaining parameters are: (i) the width γ of the kernel that specifies the back-constraining mapping (15). This parameter was estimated by viewing the scatter matrix of the kernel response to the pose training data (in all our experiments it is set $\gamma = 10^{-3}$). (ii) The dimensionality of the latent space q which we set to 4 for all our experiments. All other parameters of the model (*i.e.* the parameters of each of the three covariance functions, the parameters of the back-constraint and the coordinates of latent representation) are learned from training data.

5 Pose Inference

Given a trained model which jointly represents the pose angles, $\mathbf{Z}$, and the silhouette features, $\mathbf{Y}$, in terms of a shared sub-space, $\mathbf{X}$, we wish to infer the most likely sequence of pose angles given a set of silhouette features. We will first describe inference for a single frame and then show how inference is done for a sequence of frames.

5.1 Single Image Pose Estimation

Inference in this model is a two stage process. In the first stage the position on the latent space that is most likely to have generated the observed features is found. This is done by maximizing the predictive likelihood for the GP that maps the latent space to the given silhouette features,

$$\hat{\mathbf{x}} = \text{argmax}_{\mathbf{x}_*} p(\mathbf{y}_*|\mathbf{x}_*, \mathbf{Y}, \mathbf{X}, \Phi_Y). \tag{16}$$

Having forced the multi modality to be handled by the GP from latent to feature we expect (16) to have several maxima for an ambiguous silhouette. Equation (16) needs to be maximized using gradient based methods which require initialization. For each initialization for $\mathbf{x}_*$ we will find one (not necessarily unique) maximum. To recover multiple solutions we need multiple initializations. We chose the initializations of $\mathbf{x}_*$ from the latent points, $\mathbf{X}$, that correspond to the training data, choosing the 20 most likely points to have generated $\mathbf{y}_*$.

As explained in the previous section we back-constrain the latent space with a smooth mapping from pose space with the aim of enforcing a *one-to-one* correspondence between the latent space and the pose space. This means that given a latent representation the pose can simply be found by mapping each of the optimized latent points to pose space using the mean prediction from each latent point (3) of the GP as the most likely pose for each of the modes,

$$\hat{\mathbf{z}} = k(\hat{\mathbf{x}}, \mathbf{X})^T \mathbf{K}^{-1} \mathbf{Z}, \tag{17}$$

and accounting for the width of the distribution around each mode using the variance (4).

5.2 Sequence Estimation

A single feature descriptor is likely to correspond to several different poses, however a sequence of feature descriptors are less likely to be ambiguous. Learning a GP to predict latent points over time we can formulate the joint likelihood for a sequence of features and their latent coordinates using the dynamical model. We can maximize this joint likelihood to find the most likely latent coordinates for the observed sequence of silhouette features.

$$\hat{\mathbf{X}} = \text{argmax}_{\mathbf{X}_*} p(\mathbf{Y}_*, \mathbf{X}_*|\mathbf{Y}, \mathbf{X}, \Phi_Y, \Phi_{dyn}) \tag{18}$$

Having found the corresponding latent points the most likely poses can, as in the case of the single frame estimation, be found through the mean prediction of the GP from the latent space to the pose space,

$$\hat{\mathbf{Z}} = k(\hat{\mathbf{X}}, \mathbf{X})^T \mathbf{K}^{-1} \mathbf{Z} \tag{19}$$

5.3 Sequence Initialization

As with the initialization for a single image we want to initialize each frames latent point with a point from the latent space from the training data $\mathbf{X}$. The most

likely sequence through the training data for an unseen sequence $\mathbf{Y}_*$ can be found by interpreting the sequence as a hidden Markov model (HMM) where the latent states of the HMM correspond to the training points. The likelihood for each observation is specified by the GP point likelihood (16) associated with each each latent point, and the transitions are given by the dynamical GP that predicts over time in the latent space. The most probable path $\mathbf{X}_{init} = \text{argmax}_{\mathbf{x}^{(1)},\dots\mathbf{x}^{(n)}} p(\mathbf{x}^{(1)},\dots\mathbf{x}^{(n)}|\mathbf{y}_*^{(1)},\dots\mathbf{y}_*^{(n)})$ through this lattice can be found using the Viterbi algorithm [17]. The optimization of the sequence objective in (18) can then be initialized with $\mathbf{X}_{init}$.

6 Results

We will consider the data presented in [2]. This dataset contains 1927 training poses and 418 poses for testing from human motion capture data. Each pose is parametrized by a 54 dimensional joint vector. From each pose vector an image has been generated using the computer graphic package Poser from Curious Labs.

6.1 Single Image

In Figure 1 results for a single pose estimate are shown. Each estimate is initialized using the 20 most likely points from the training data. The top row shows an ambiguous silhouette of the mis-labeling or limp flip type, followed by the model estimates sorted according to likelihood. We expect a mis-labeling ambiguity to correspond to a discrete set of poses with significantly different joint angle configurations. Only the three first estimates have a good correspondence to the silhouette, corresponding to significantly different joint configurations of both legs and arms as expected.

The bottom row show a silhouette corresponding to an out-of-plane ambiguity. This silhouette contains very little information about the position of the limbs moving out-of-plane, we can see that this is well reflected by the model, suggesting several different configuration of the legs and the arms. The least likely of the shown estimates is a heading ambiguity which is also a plausible estimate to the silhouette.

6.2 Sequence

In Figure 2 every $20th$ frame for a circular walk sequence is shown. Our model does well for most of the frames but misestimates one stride in a turn. This bad estimate is due to lack of training data, each turn in the training data from this position is taken with the opposite leg compared to the test data. Therefore the dynamical model does not agree with the observations and the estimated pose is a suboptimal minimum. The estimate waits until the stride with the "unexpected" leg is finished and then latches on in the correct stride. Outside this turn our model correctly estimates the true pose.

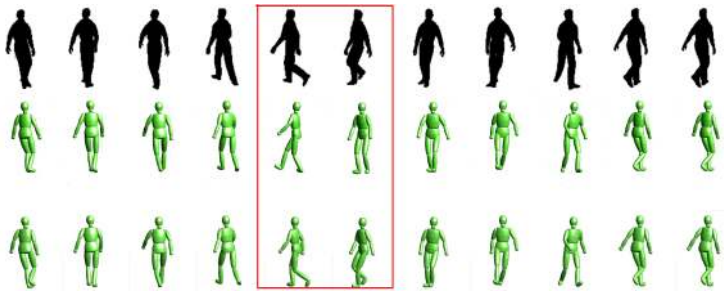

Fig. 2. Every $20th$ frame from a circular walk sequence, Top Row: Input Silhouette, Middle Row: Model Pose Estimate, Bottom Row: Ground Truth. The box indicates bad estimates by our model.

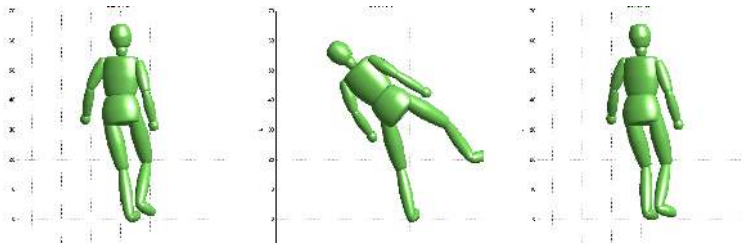

Fig. 3. Angle error: The image on the left is the true pose, the middle image has an angle error of 1.7°, the image on the right has an angle error of 4.1°. An angle error higher up in the joint-hierarchy will effect the positions for all joints further down. As the errors for the middle image are higher up in the hierarchy this will effect each limb connected further down the chain from this joint thereby resulting in a significantly different limp positions.

6.3 Quantitative Results

Results on Human Pose estimation are normally reported by the means square error between the estimated pose and ground truth [2,18]. A mean square error treats all dimension of the joint angle space with equal importance and do not reflect the hierarchical structure of the human physiology, Figure 3. Table 1 shows mean RMS angle and joint position error for our model, a set of regression algorithms and the mean pose in the training data over the test sequence. We can see that our single estimate is worse than RVM regression. This is expected as the multi-modal prediction in our model will either predict the correct pose or we find an ambiguous pose in these cases a regression based methods would predict the mean pose which will result in a smaller error. Neither angle or joint position error can correctly reflect visual similarity for sequences as humans have strong and complex priors with regards to motion.

Table 1. Mean RMS Angle and Joint Position Error normalized by the height of the model. Note that the only the GP-LVM Sequence method is using temporal information.

	Angle Error	Joint Position Error
Mean Training Pose	8.3°	$37.3 \cdot 10^{-2}$
Linear Regression	7.7°	$33.5 \cdot 10^{-2}$
RVM	5.9°	$15.8 \cdot 10^{-2}$
GP-LVM Single	6.5°	$17.2 \cdot 10^{-2}$
GP-LVM Sequence	5.3°	$15.0 \cdot 10^{-2}$

7 Discussion

The model presented in this paper learns a shared low-dimensional representation of a single observation space and a target domain. For the task of estimating pose from silhouette the information in the observation space in not sufficient to determine pose why we in this paper have used temporal consistency to disambiguate between multiple solutions. Another strategy is to incorporate additional observation spaces that together will better represent the target domain. A common example for the application presented is to use information from additional views, another example is combining information from both visual and audio cues. This leads to the problem of how to "merge" the different observation spaces into a single representation. In [8] features are extracted from multiple views and concatenated into a larger feature vector from which pose is inferred. In the presented model additional observations can simply be added and a shared low-dimensional representation can be learned of all the observation spaces and the target domain. An additional advantage with the model is that inference can be done if any non-zero subset of the observations are given, presenting additional observations will constrain the problem further.

8 Conclusion and Future Work

We have presented a method for human pose estimation from silhouettes using Gaussian Process Latent Variable Models. Our model represent both image observations and pose parameters in a shared latent space. The structure of the latent space is constrained to produce smooth trajectories over time by incorporating a GP to predict over the latent space. The model only has two free parameters and requires no manual initialization.

In future work we would like to extend the model to learn two separate sets of dynamics, one dynamic for the human relative motion (e.g. stride) and one model for the motion of the root of the body. This should hopefully solve the problems in our estimation and also reduced the amount of training data needed.

Acknowledgment

This work was supported by EPSRC, the IST Programme of the European Community, under the PASCAL Network of Excellence, IST-2002-506778 and

Sharp Laboratories Europe. We would like to thank Ankur Agarwal, Guido Sanguinetti and Nathaniel J. King.

References

1. Lawrence, N.D.: Probabilistic non-linear principal component analysis with gaussian process latent variable models. Journal of Machine Learning Research 6, 1783–1816 (2005)
2. Agarwal, A., Triggs, B.: Recovering 3d human pose from monocular images. IEEE Trans. Pattern Anal. Mach. Intell. 28(1), 44–58 (2006)
3. Grauman, K., Shakhnarovich, G., Darrell, T.: Inferring 3d structure with a statistical image-based shape model. In: ICCV 2003, pp. 641–648 (2003)
4. Kehl, R., Bray, M., Gool, L.J.V.: Full body tracking from multiple views using stochastic sampling. In: CVPR(2), pp. 129–136 (2005)
5. Sminchisescu, C., Kanaujia, A., Li, Z., Metaxas, D.N.: Discriminative density propagation for 3d human motion estimation. In: CVPR (1), pp. 390–397 (2005)
6. Sminchisescu, C., Telea, A.: Human pose estimation from silhouettes - a consistent approach using distance level sets. In: WSCG, pp. 413–420 (2002)
7. Sidenbladh, H., Black, M.J., Fleet, D.J.: Stochastic tracking of 3d human figures using 2d image motion. In: Vernon, D. (ed.) ECCV 2000. LNCS, vol. 1843, pp. 702–718. Springer, Heidelberg (2000)
8. de Campos, T.E., Murray, D.W.: Regression-based hand pose estimation from multiple cameras. In: CVPR(1), pp. 782–789 (2006)
9. Sun, Y., Bray, M., Thayananthan, A., Yuan, B., Torr, P.: Regression-based human motion capture from voxel data. In: BMVC (2006)
10. Belongie, S., Malik, J., Puzicha, J.: Shape context: A new descriptor for shape matching and object recognition. In: NIPS, pp. 831–837 (2000)
11. Mori, G., Belongie, S.J., Malik, J.: Efficient shape matching using shape contexts. IEEE Trans. Pattern Anal. Mach. Intell. 27(11), 1832–1837 (2005)
12. Rasmussen, C.E., Williams, C.K.: Gaussian Processes for Machine Learning. MIT Press, Cambridge (2006)
13. Lawrence, N.D.: Gaussian process latent variable models for visualisation of high dimensional data. In: NIPS (2003)
14. Lawrence, N.D., Candela, J.Q.: Local distance preservation in the gp-lvm through back constraints. In: ICML, pp. 513–520 (2006)
15. Wang, J., Fleet, D.J., Hertzmann, A.: Gaussian process dynamical models. In: NIPS (2005)
16. Shon, A.P., Grochow, K., Hertzmann, A., Rao, R.P.N.: Learning shared latent structure for image synthesis and robotic imitation. In: NIPS (2005)
17. Viterbi, A.J.: Error bounds for convolutional codes and an asymptotical optimum decoding algorithm. IEEE Transactions on Information Theory (1967)
18. Shakhnarovich, G., Viola, P.A., Darrell, T.: Fast pose estimation with parameter-sensitive hashing. In: ICCV, pp. 750–759 (2003)