Temporal-Spatial Local Gaussian Process Experts for Human Pose Estimation

Xu Zhao[1], Yun Fu[2], and Yuncai Liu[1]

[1] Institute of Image Processing & Pattern Recognition,
Shanghai Jiao Tong University, Shanghai 200240, China
[2] BBN Technologies, 10 Moulton Street,
Cambridge, MA 02138, USA

Abstract. Within a discriminative framework for human pose estimation, modeling the mapping from feature space to pose space is challenging as we are required to handle the multimodal conditional distribution in a high-dimensional space. However, to build the mapping, current techniques usually involve a large set of training samples in the learning process but are limited in their capability to deal with multimodality. In this work, we propose a novel online sparse Gaussian Process (GP) regression model combining both temporal and spatial information. We exploit the fact that for a given test input, its output is mainly determined by the training samples potentially residing in its neighbor domain in the input-output unified space. This leads to a local mixture GP experts system, where the GP experts are defined in the local neighborhoods with the variational covariance function adapting to the specific regions. For the nonlinear human motion series, we integrate the temporal and spatial experts into a seamless system to handle multimodality. All the local experts are defined online within very small neighborhoods, so learning and inference are extremely efficient. We conduct extensive experiments on the real HumanEva database to verify the efficacy of the proposed model, obtaining significant improvement against the previous models.

1 Introduction

Recovering human pose from visual signals is a fundamental yet extremely challenging problem in computer vision research. A wide spectrum of real-world applications [1] in control, human computer interaction, multimedia communication, and surveillance scenarios motivate the endeavors to find robust and effective solutions to this problem.

Among the large amount of studies on pose estimation, discriminative approaches [2] have recently seen a revival due to their flexible frameworks adapting to different learning methods and the ability of fast inference in real-world databases. Discriminative approaches for human pose estimation [3,4,5,6,7] aim to model the direct mapping from visual observations to pose configurations. The methods range from nearest-neighbor retrieval [8,9] and manifold learning [4] to regression [10,7] and probabilistic mixture of predictors [2,5]. However, all of the discriminative approaches have to face the difficult problem of how to effectively model a multimodal conditional distribution in a high-dimensional space with small size training data.

Current techniques to deal with multimodality are mainly in the category of mixture of models. In [2,5], the conditional Bayesian Mixture of Experts (BME) was used to

H. Zha, R.-i. Taniguchi, and S. Maybank (Eds.): ACCV 2009, Part I, LNCS 5994, pp. 364–373, 2010.

represent the multimodal image-to-pose mapping. This model is flexible in modeling multimodality by introducing the input sensitive gate function. However, this parametric model is prone to fail if the input dimension is too high. Moreover, the estimation accuracy of the BME model heavily depends on the distribution of training samples in ambiguous regions, so it is hard to obtain satisfactory results on small size datasets.

Recently, a few attempts have been made to estimate human pose by using Gaussian Process (GP) [11] algorithms, within both discriminative [12,7] and generative [13] frameworks. GP regression has proven to be a powerful tool in many applications. In the discriminative models, pose estimation is mainly built on the basis of GP regression. The model defines a prior probability distribution over infinite function space. This leads to a non-linear probabilistic regression framework working along with the kernelized covariance function. The flexibilities in kernel selection and non-parametric nature of GP model are advantageous to find efficient solutions of pose estimation on small size databases [13,7]. However, the full GP regression suffers from two inevitable limitations: relative expensive computing cost and incapability to handle multimodality.

To tackle the computing limitations, a lot of efforts have been made on the sparse approximations of full GP [14,11]. These methods use only a subset of training inputs [15] or a set of inducing variables [16] to approximate the covariance matrix. Although the computational expenses are reduced by such approximations, the models still work within the global voting framework and might lack effective mechanisms to avoid the averaging effect. Another kind of method proposed to handle above two limitations is mixture of Gaussian process experts [17,18,19]. Similar to mixture of experts architecture [20], in these models the input space is divided into different regions by a gating network, each of which is dominated by a specific GP expert. In the model, the cubic computing cost on the entire dataset is reduced to that on part of the data. In the meantime, the covariance functions are localized to adapt to different regions. However, learning the mixture GP experts is intimately coupled with the gating network. The determination of gating network is another complex problem.

In this paper, we propose a novel mixtures of local GP experts model, utilizing both temporal and spatial information. Our method is inspired by the recent work on human pose inference using sparse GP regression [12]. In their model, the local experts are trained offline and the local regressors are defined online for each test point. Derived from the neighborhood of the test point in the appearance space, each local GP is defined to be consistent in the pose space. Unlike the mixture of GP experts, this model avoids the tremendous efforts in computing the gating network. We generalize the localization strategy in [12] and design the local GP experts model with three contributions:

(1) We propose to define the local GP experts in the unified input-output space, therefore each GP expert is composed of samples that are localized in both input and output space. This strategy is different from that proposed in [12], where the neighborhood is defined separately in input and output space. Such scheme prone to fail in dealing with more-to-one mapping because the neighborhood relationship in output space would be changed in the input space. In comparison, our model can flexibly handle the two-way multimodality.

(2) We introduce the temporal local GP experts. In the unified space, we integrate the temporal and spatial experts into a whole to make prediction and handle multimodality.

(3) We evaluate the proposed Temporal-Spatial Local (TSL) GP model on the public real HumanEva database [21] and achieve significant improvements against both full GP model and the local sparse GP model.

2 Local Gaussian Process Experts Model

2.1 Gaussian Process Regression

Gaussian process is the generalization of Gaussian distributions defined over infinite index sets [11]. Suppose we have a training dataset $\mathcal{D} = \{(\mathbf{x}_i, \mathbf{y}_i), i = 1, \cdots, N\}$, composed of inputs $\mathbf{x}_i$ and noisy outputs $\mathbf{y}_i$. We consider a regression model defined in terms of the function $f(\mathbf{x})$ so that $\mathbf{y}_i = f(\mathbf{x}_i) + \epsilon_i$, where $\epsilon_i \sim \mathcal{N}(0, \beta^{-1})$ is a random noise variable and the hyperparameter β represents the precision of the noise. From the Gaussian assumption of prior distribution over functions $f(\mathbf{x})$, the joint distribution of outputs $\mathbf{Y} = [\mathbf{y}_1, \cdots, \mathbf{y}_N]^T$ conditioned on input values $\mathbf{X} = [\mathbf{x}_1, \cdots, \mathbf{x}_N]^T$, is given by

$$p(\mathbf{Y}|\mathbf{X}) = \int p(\mathbf{Y}|\mathbf{f}, \mathbf{X}) p(\mathbf{f}|\mathbf{X}) \mathrm{d}\mathbf{f} = \mathcal{N}(\mathbf{Y}|0, \mathbf{K}), \tag{1}$$

where $\mathbf{f} = [f_1, \cdots, f_N]^T$, $f_i = f(\mathbf{x}_i)$ and the covariance matrix $\mathbf{K}$ has elements

$$K_{i,j} = k(\mathbf{x}_i, \mathbf{x}_j) + \beta^{-1}\delta_{ij}, \tag{2}$$

where δ_{ij} is the Kronecker delta function. In this paper, we use a kernel function k which is the sum of an isotropic exponential covariance function, a noise term and a bias term, all with hyperparameters, $\bar{\theta}$. For a new test input $\mathbf{x}_*$, the conditional distribution, $p(\mathbf{y}_*|\mathbf{X}, \mathbf{Y}, \mathbf{x}_*) = \mathcal{N}(\mu, \sigma)$, is a Gaussian distribution with mean and covariance given by

$$\mu(\mathbf{x}_*) = \mathbf{k}_{*,\zeta}\mathbf{K}_{\zeta,\zeta}^{-1}\mathbf{Y}_\zeta, \quad \sigma(\mathbf{x}_*) = k_{*,*} - \mathbf{k}_{*,\zeta}\mathbf{K}_{\zeta,\zeta}^{-1}\mathbf{k}_{\zeta,*}, \tag{3}$$

where ζ's are the indices of the N training inputs, $\mathbf{K}_{\zeta,\zeta}$ is the covariance matrix with elements given by (2) for $i, j = 1, \ldots, N$, the vector $\mathbf{k}_{\zeta,*} = \mathbf{k}_{*,\zeta}^T$ is the cross-covariance of the test input and the N training inputs, and scalar $k_{*,*} = k(\mathbf{x}_*, \mathbf{x}_*) + \beta^{-1}$ is the covariance of the test input.

Note that the mean (3) can be viewed as a weighted voting from N training outputs

$$\mu(\mathbf{x}_*) = \sum_{n=1}^{N} w_n \mathbf{y}_n, \tag{4}$$

where w_n is the n^{th} component of $\mathbf{k}_{*,\zeta}\mathbf{K}_{\zeta,\zeta}^{-1}$. With this insight, we can view the GP regression as a voting process, where each training output has a weighted vote to determine what the test output should be.

2.2 Local Mixture of GP Experts

To reduce the computing cost and handle multimodality, we need to sparsify the full GP regression model. Current GP sparse techniques [11,14] mainly focus on globally sparsifying the full training dataset based on some selection criteria such as online learning

[22], greedy posterior maximization[23], maximum information gain [24], and matching pursuit [25]. By using this kind of methods, the computational complexity of full GP , $O(N^3)$, are reduced to $O(m^3)$ or $O(Nm^2)$, where N, m are the sizes of full training dataset and the selected subset respectively. However, for very large database, the reduction is not enough. Moreover, these ideas still work within the global voting framework. That means for every test input, no matter which local distribution mode they belong to, the selection of the training samples and covariance function are global.

Actually, for a special test input, the training samples in its neighborhood usually have more impacts on the prediction than those far from it. In voting view, the weights of the local voters are bigger than others (see (4)). In the GP model, kernel function provides a metric to measure the similarity between the inputs. Ideally, this metric should be adjusted dynamically to adapt different local regions.

Motivated by above considerations, we develop the local mixture of GP experts. Like the model proposed in [12], for a given test input, we select different local GP experts in its neighborhood. The training samples of each expert are also selected locally. These local experts build up a local mixture GP experts system to make the prediction. To this end, in our model, the mean prediction for a given test input $\mathbf{x}_*$ is given by

$$\mu(\mathbf{x}_*) = \sum_{i=1}^{T} \pi_i \mathbf{k}_{*,\zeta_i} \mathbf{K}_{\zeta_i,\zeta_i}^{-1} \mathbf{Y}_{\zeta_i} = \sum_{i=1}^{T} \sum_{j=1}^{S} \pi_i w_{ij} \mathbf{y}_{ij}, \tag{5}$$

where T is the number of local experts, S is the size of each expert, ζ_i is the index set of samples for the i-th expert, π_i is the prediction weight of the i-th expert and $\mathbf{y}_{ij}$ is the j-th training output belonging to the i-th expert, w_{ij} is its weight. Both T and S are parameters of our model. In practice, small values are sufficient for accurate predictions. Each π_i is set to be a function of the inverse variance of the expert's prediction.

Different from the localization strategy in [12], our model define the neighborhood in the input-output unified space $\mathbb{U}$, where the data points are the concatenation of the input and output vector. The advantages of our strategy are two folds:

(1) The neighborhood relationship is closer to the real distribution in $\mathbb{U}$ than in the single input and output space. For example in pose estimation, two image feature points which are very close in feature space might be quite different in pose space, and vice versa. In $\mathbb{U}$, this kind of ambiguity can be avoided to a large extent.

(2) Our strategy can deal with two-way multimodal distributions. For the more-to-one input to output mapping, the data points would be scattered in the input space just using the neighborhood definition in the output space. But in $\mathbb{U}$, this situation can be avoided.

In implementation, the unified data space $\mathbb{U}$ is divided into R different local regions with a clustering algorithm. Each region is dominated by a local GP expert trained offline. Given a test input, starting from its neighborhood in the input space, we find its local neighbors in $\mathbb{U}$ to build the local mixture of GP experts model. The algorithm is summarized in Algorithm 1, where, the data set in $\mathbb{U}$ is represented as $\mathbf{D} = [\mathbf{d}_1, \cdots, \mathbf{d}_N]$ with $\mathbf{d}_i = (\mathbf{x}_i, \mathbf{y}_i)$. The function findNN($\mathbf{X}, \mathbf{x}, S$) finds S nearest neighbors of $\mathbf{x}$ in $\mathbf{X}$. The function kmeans($\mathbf{D}, R$) performs k-means clustering on data set $\mathbf{D}$ and returns the R centers $\mathbf{C}_{\mathcal{R}}$ and clusters $\mathbf{D}_{\mathcal{R}}$.

Algorithm 1. Local mixture of GP experts: learning and inference

OFFLINE: Training of the Local Experts
R: number of local GP experts
$(\mathbf{C}_{\mathcal{R}}, \mathbf{D}_{\mathcal{R}}) = \text{kmeans}(\mathbf{D}, R)$
for $i = 1 \dots R$ **do**
  $\{\bar{\theta}^i\} \Leftarrow \min(-\ln p(\mathbf{Y}_{\mathcal{R}_i}|\mathbf{X}_{\mathcal{R}_i}, \bar{\theta}^i))$
end for
ONLINE: Inference of test point $\mathbf{x}_*$
T: number of experts, S: size of each expert
$\eta = \text{findNN}(\mathbf{X}, \mathbf{x}_*, T)$
for $j = 1 \dots T$ **do**
  $\zeta = \text{findNN}(\mathbf{D}, \mathbf{d}_{\eta_j}, S)$
  $t = \text{findNN}(\mathbf{C}_{\mathcal{R}}, \mathbf{d}_{\eta_j}, 1)$
  $\bar{\theta} = \bar{\theta}^t$
  $\mu_j = \mathbf{k}_{*,\zeta}\mathbf{K}_{\zeta,\zeta}^{-1}\mathbf{Y}_{\zeta}$
  $\sigma_j = k_{*,*} - \mathbf{k}_{*,\zeta}\mathbf{K}_{\zeta,\zeta}^{-1}\mathbf{k}_{\zeta,*}$
end for
$p(\mathbf{y}_*|\mathbf{X}, \mathbf{Y}) \approx \sum_{i=1}^{T} \pi_i \mathcal{N}(\mu_i, \sigma_i^2)$

3 Temporal-Spatial Local GP Experts

In order to handle multimodality more effectively, on the basis of the spatial experts, we introduce the temporal experts to construct the temporal-spatial combined mixture of GP experts model. In this model, the spatial local experts learn the relationship between the input space and output space, the temporal local experts explore the underlying context of the output space. Suppose we work with sequential data. By adding the temporal constrain, the regression models can be formulated as

$$\mathbf{y}_t = f(\mathbf{x}_t) + \epsilon_{x,t} \quad \text{and} \quad \mathbf{y}_t = g(\mathbf{y}_{t-1}) + \epsilon_{y,t}, \tag{6}$$

where t is the temporal tag, $\epsilon_{x,t} \sim \mathcal{N}(0, \beta_x^{-1})$ and $\epsilon_{y,t} \sim \mathcal{N}(0, \beta_y^{-1})$ are noise processes. We use the first-order Markov dynamical model to account for the dependence in the output space. For (6), considering dynamic mapping on the data set $\mathbf{Y} = [\mathbf{y}_1, \cdots, \mathbf{y}_N]^T$ in the output space, the joint distribution of $\mathbf{Y}$ is given by

$$p(\mathbf{Y}) = p(\mathbf{y}_1) \int \prod_{t=2}^{N} p(\mathbf{y}_t|\mathbf{y}_{t-1}, \mathbf{g}) p(\mathbf{g}) \mathrm{d}\mathbf{g}, \tag{7}$$

where $\mathbf{g} = [g_1, \cdots, g_{N-1}]^T$, $g_i = g(\mathbf{y}_i)$. In view of the nonlinear dynamical nature of human motion, we use an RBF plus linear kernel

$$k(\mathbf{y}_i, \mathbf{y}_j) = \theta_0 \exp\left\{-\frac{\theta_1}{2} \parallel \mathbf{y}_i - \mathbf{y}_j \parallel^2\right\} + \theta_2 + \theta_3 \mathbf{y}_i^T \mathbf{y}_j. \tag{8}$$

To build the local temporal experts model, we use similar localization strategy described in Algorithm 1. Once the local temporal experts give the prediction $\hat{\mathbf{y}}$, we proceed to

Algorithm 2. Online inference with temporal-spatial local GP experts

Require: $\mathbf{x}_t^*$, $\hat{\mathbf{y}}_t$: the output at last time instant
1. **COMBINATION of two class of local experts**
2. T_1: number of spatial experts
   T_2: number of temporal experts
   S: size of each expert
3. $\eta^{(s)} = \text{findNN}(\mathbf{X}, \mathbf{x}_t^*, T_1)$;
4. $\hat{\mathbf{d}}_t = (\mathbf{x}_t^*, \hat{\mathbf{y}}_t)$;
5. $\eta^{(t)} = \text{findNN}(\mathbf{D}, \hat{\mathbf{d}}_t, T_2)$;
6. $\eta = \eta^{(s)} \cup \eta^{(t)}$;
7. **ONLINE inference**
8. $T = T_1 + T_2$: number of all experts
9. **for** $j = 1 \ldots T$ **do**
10. $\zeta = \text{findNN}(\mathbf{D}, \mathbf{d}_{\eta_j}, S)$
11. $t = \text{findNN}(\mathbf{C}_{\mathcal{R}}, \mathbf{d}_{\eta_j}, 1)$
12. $\bar{\theta} = \bar{\theta}^t$
13. $\mu_j = \mathbf{k}_{*,\zeta}\mathbf{K}_{\zeta,\zeta}^{-1}\mathbf{Y}_\zeta$
    $\sigma_j = k_{*,*} - \mathbf{k}_{*,\zeta}\mathbf{K}_{\zeta,\zeta}^{-1}\mathbf{k}_{\zeta,*}$
14. **end for**
15. $p(\mathbf{y}_t^*|\mathbf{X}, \mathbf{Y}) \approx \sum_{i=1}^{T} \pi_i \mathcal{N}(\mu_i, \sigma_i^2)$

make the prediction supported by the the local spatial experts in the unified space $\mathbb{U}$. Formally this process is described by $p(y_t|y_{t-1}, x_t) = \int p(y_t|\hat{y}_t, x_t)p(\hat{y}_t|y_{t-1})d\hat{y}_t$.

In summary, we can build up the temporal-spatial combined local GP model as follows. Given the training data set $\mathbf{X} = [\mathbf{x}_1, \cdots, \mathbf{x}_N]^T$ and $\mathbf{Y} = [\mathbf{y}_1, \cdots, \mathbf{y}_N]^T$, we firstly learn a set of hyperparameters $\{\bar{\theta}^i\}$ for the local spatial GP experts following the process described in the offline part of Algorithm 1. Then, the local temporal models is also built up by the same way using the training data $\mathbf{Y}_1 = [\mathbf{y}_1, \cdots, \mathbf{y}_{N-1}]^T$ and $\mathbf{Y}_2 = [\mathbf{y}_2, \cdots, \mathbf{y}_N]^T$. At the time instant $t-1$, one can get the prediction $\hat{\mathbf{y}}_t$ under the process of local temporal experts model. Then, at the time instant t, we import $\mathbf{x}_t^*$ and $\hat{\mathbf{y}}_t$ into our temporal-spatial combined local experts model to get the final prediction $\mathbf{y}_t^*$. The algorithm is described in Algorithm 2.

Computational complexity. We compare the computational complexity of our models with that of full GP in Table 1. Note that for both learning and inference, our models are linear in N stemming from the operators of finding nearest neighbors ($O(RN)$) and k-means clustering ($O(RdN)$). The complexity of inverting the local GP is not a function of the number of examples, since the local GP experts are of fixed size. When $N \gg S$,

Table 1. Computational complexity: both local models are linear in N for both learning and inference, where d is the dimension of the data points. In experiments, $T, S, R \ll N$

	Full GP	Local Sparse GP Experts	Temporal-Spatial Local GP Experts
Learning	$O(N^3)$	$O(RS^3 + R(d+1)N)$	$O(2RS^3 + 2R(d+1)N)$
Inference	$O(N^3)$	$O(TS^3 + TN)$	$O(TS^3 + TN)$

the computational cost is significantly reduced. Moreover, in general, R is also a small value comparing to N, therefore the complexity of our model is much smaller than that of full GP. It's computational beneficial in dealing with very large size databases.

4 Experiments

4.1 Regression on the Multimodal Functions

In this experiments, full GP, local sparse (LS) GP (Algorithm 1) and Temporal-Spatial Local (TSL) GP (Algorithm 2) are tested on two sets of toy data (see the caption of Fig. 1 for the detailed description of the data set). The regression results are shown in Fig. 1. We can find that for the multimodal function (first row of Fig. 1), the full GP just globally averages the outputs of different modes. The local sparse GP can partly handle the multimodality and avoid the averaging effect but the outputs frequently skip between different modes in the multimodal regions (see Fig. 1 (b)). Therefore it's hard to get a smooth prediction. This problem can be fixed in the TSL GP model due to the utilization of temporal information. Notice that in Fig. 1(c), the skips are eliminated and the prediction is smooth. Another data set provides a unimodal input-to-output mapping. The regression results are illustrated in Fig. 1(d-f). In this situation, the full GP

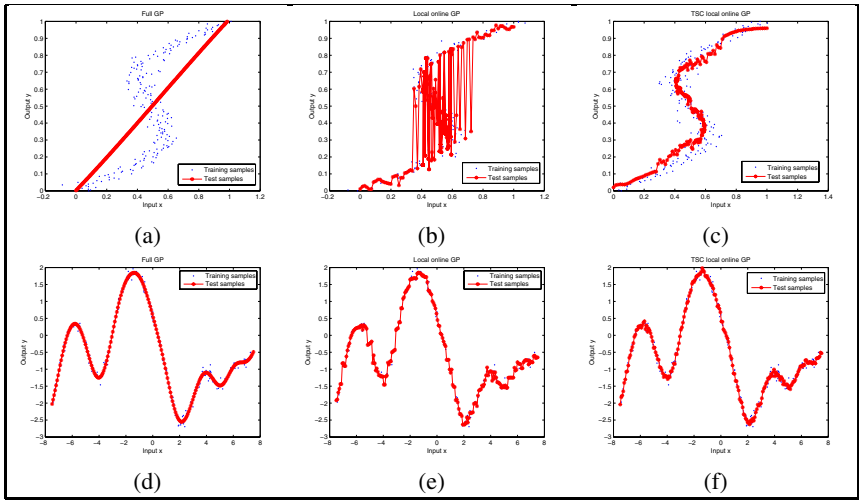


Fig. 1. Model comparisons between full GP, Local Sparse GP, and TSL GP on two sets of illustrative data.The first data set is consists of about 200 training pairs of (x, y), where y generated uniformly in $(0, 1)$ and evaluated as $x = y + 0.3\sin(2\pi y) + \epsilon$, with ϵ drawn from a zero mean Gaussian with standard deviation 0.05. Notice here $p(y|x)$ is multimodal. Test points ($N_t = 200$) are sampled uniformly from $(0, 1)$. The second data set is obtained by sampling ($N = 100$) a GP with covariance matrix obtained from an RBF. About 200 test inputs are sampled uniformly in $(-7.5, 7.5)$. The regression results are shown in: (a,d) Full GP. (b,e) Local Sparse GP. (c,f) TSL GP. For better viewing, please see enlarged color pdf file.

Table 2. Average RMS error (in degree) over all joint angles for walking, box, and jog actions of the three subjects. The performances of four regression models are evaluated.

	S1			**S2**			**S3**		
	Walking	Box	Jog	Walking	Box	Jog	Walking	Box	Jog
Full GP	6.4155	5.9226	6.3579	5.6821	5.0510	3.5510	7.0424	7.2785	2.5060
LS-GP(S)	6.6130	5.6815	6.2040	5.4859	4.9035	3.4183	6.9563	7.0344	2.5219
LS-GP(U)	6.3567	5.5951	6.1352	5.4498	4.6334	3.2458	6.7356	6.9226	2.3725
TSL-GP	5.5846	5.2913	5.0348	4.7816	4.4119	2.5085	6.0349	6.2152	2.0682

gives perfect results because the global voting mechanism can deal with the unimodal mapping very well. Here, the local sparse GP also gives good results although there still exist some jitters. The prediction of TSL GP is smoother than that of the LS GP model.

4.2 Results on the HumanEva Database

We also validate our models on the HumanEva database [21]. The database provides synchronized video and motion capture streams. It contains multiple subjects performing a set of predefined actions with repetitions. The database was originally partitioned into training, validation, and testing sub-sets. We use sequences in the original training sub-set for training and those in the original validation sub-set for testing. A total of 2,932 frames for walking motion, 2,050 frames for jog motion, and 1,889 frames for box motion are used. The pose is represented by Euler angles and the dimension of the pose space (output space) is 26. We use patch-based image feature described by the SIFT descriptor on the dense interest points with position information. The dimension of the feature space is 100. All the images we used are captured by the camera C1.

We report the mean RMS absolute difference errors between the true and estimated joint angles, in degrees. The performance of four models: full GP, LS-GP(U) defined in the unified space, LS-GP(S) defined in separate space (proposed in [12]), and TSL-GP are evaluated. In the experiments, we take the values of R, T, S as $100, 10, 50$, respectively. The results are reported in Table 2. It is obvious that the TSL-GP model outperforms other models with significant improvements. Other two local GP models are slightly better than full GP. We also find that in the unified space, the local GP gets some performance improvement although it is not very distinct. Fig. 2-3 show the performance comparisons between three models: full GP, LS-GP(U), and TSL-GP with relative errors (normalized by the range of variations of the joint angles), where the errors are averaged over all the subjects but specified for the three actions. For most of joint angles, the TSL-GP model get the best performance. The performance of LS-GP(U) is better than that of full GP model. In Fig. 4, the estimation results and ground truth of two joint angles over the whole sequence in walking and jog action are plotted. We compare the results of full GP and TSL-GP. It can be observed that the curves of the TSL-GP model are more smooth and close to the ground truth than the full GP model by using the temporal information.

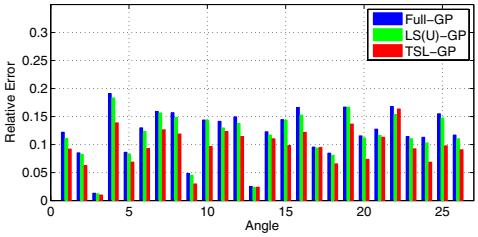


Fig. 2. Performance comparison of three models for the walking action

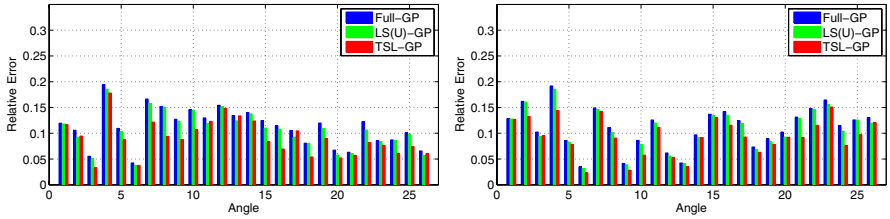
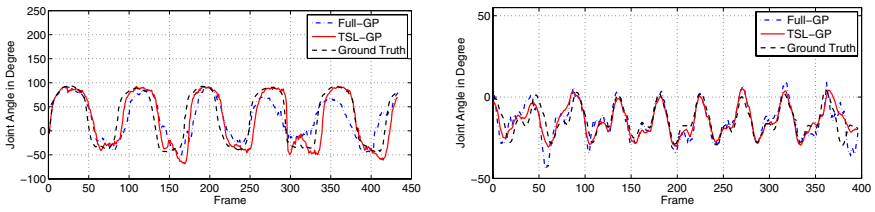


Fig. 3. Performance comparisons of three models for the jog (left) and box (right) actions



(a) **(b)**

Fig. 4. Curve comparisons of joint angles: ground truth and estimations with TSL-GP and Full GP regression. (a) Left shoulder (x-axis) of subject S2 in walking action. (b) right hip (x-axis) of subject S3 in jog action.

5 Conclusions

In this paper, we presented a novel temporal-spatial combined local GP experts model for efficient estimation of 3D human pose from monocular images. The proposed model is essentially a kind of mixture of GP experts in which we integrate both spatial and temporal information into a seamless system to handle multimodality. The local experts are trained in the local neighborhood. Different from previous work, the neighbor relationship is defined in the unified input-output space in this model. Therefore we can flexibly handle two-way multimodality. Both spatial and temporal local experts are defined online within very small neighborhoods, so learning and inference are extremely efficient. We conducted the experiments on the real HumanEva database to validate the efficacy

of the proposed model and achieved accurate results. This model is general purposed therefore its adaption to other problems is straightforward.

References

1. Moeslund, T., Hilton, A., Krüger, V.: A survey of advances in vision-based human motion capture and analysis. CVIU 104(2-3), 90–126 (2006)
2. Sminchisescu, C., Kanaujia, A., Li, Z., Metaxas, D.: Discriminative density propagation for 3d human motion estimation. In: CVPR (2005)
3. Agarwal, A., Triggs, B.: Recovering 3D human pose from monocular images. PAMI 28(1), 44–58 (2006)
4. Elgammal, A., Lee, C.: Inferring 3D body pose from silhouettes using activity manifold learning. In: CVPR (2004)
5. Ning, H., Wei, X., Gong, Y., Huang, T.: Discriminative learning of visual words for 3D human pose estimation. In: CVPR (2008)
6. Bissacco, A., Yang, M.H., Soatto, S.: Fast human pose estimation using appearance and motion via multi-dimensional boosting regression. In: CVPR (2007)
7. Zhao, X., Ning, H., Liu, Y., Huang, T.: Discriminative estimation of 3D human pose using Gaussian processes. In: ICPR (2008)
8. Shakhnarovich, G., Viola, P., Darrell, T.: Fast pose estimation with parameter-sensitive hashing. In: ICCV (2003)
9. Tomasi, C., Petrov, S., Sastry, A.: 3d tracking= classification+ interpolation. In: ICCV (2003)
10. Agarwal, A., Triggs, B.: 3D human pose from silhouettes by relevance vector regression. In: CVPR (2004)
11. Rasmussen, C., Williams, C.: Gaussian processes for machine learning (2006)
12. Urtasun, R., Darrell, T.: Local probabilistic regression for activity-independent human pose inference. In: CVPR (2008)
13. Urtasun, R., Fleet, D., Hertzmann, A., Fua, P.: Priors for people tracking from small training sets. In: ICCV (2005)
14. Quiñonero-Candela, J., Rasmussen, C.: A unifying view of sparse approximate Gaussian process regression. JMLR 6, 1939–1959 (2005)
15. Lawrence, N., Seeger, M., Herbrich, R.: Fast sparse Gaussian process methods: The informative vector machine. In: NIPS (2003)
16. Snelson, E., Ghahramani, Z.: Sparse Gaussian processes using pseudo-inputs. In: NIPS (2006)
17. Rasmussen, C., Ghahramani, Z.: Infinite mixtures of Gaussian process experts. In: NIPS (2002)
18. Tresp, V.: Mixtures of Gaussian processes. In: NIPS (2000)
19. Meeds, E., Osindero, S.: An alternative infinite mixture of Gaussian process experts. In: NIPS (2001)
20. Jacobs, R., Jordan, M., Nowlan, S., Hinton, G.: Adaptive mixtures of local experts. Neural Computation 3(1), 79–87 (1991)
21. Sigal, L., Black, M.: Humaneva: Synchronized video and motion capture dataset for evaluation of articulated human motion. Tech.Report CS-06-08, Brown University (2006)
22. Csato, L., Opper, M.: Sparse on-line Gaussian processes. Neural Computation 14(3), 641–668 (2002)
23. Smola, A., Bartlett, P.: Sparse greedy Gaussian process regression. In: NIPS (2001)
24. Seeger, M., Williams, C., Lawrence, N.: Fast forward selection to speed up sparse Gaussian process regression. In: Proc. of the Ninth Int'l Workshop on AI and Statistics (2003)
25. Keerthi, S., Chu, W.: A matching pursuit approach to sparse Gaussian process regression. In: NIPS (2006)