Sparse Probabilistic Regression for Activity-independent Human Pose Inference

Raquel Urtasun and Trevor Darrell
UC Berkeley EECS & ICSI
MIT CSAIL

Abstract

Discriminative approaches to human pose inference involve mapping visual observations to articulated body configurations. Current probabilistic approaches to learn this mapping have been limited in their ability to handle domains with a large number of activities that require very large training sets. We propose an online probabilistic regression scheme for efficient inference of complex, high-dimensional, and multimodal mappings. Our technique is based on a local mixture of Gaussian Processes, where locality is defined based on both appearance and pose, and where the mapping hyperparameters can vary across local neighborhoods to better adapt to specific regions in the pose space. The mixture components are defined online in very small neighborhoods, so learning and inference is extremely efficient. When the mapping is one-to-one, we derive a bound on the approximation error of local regression (vs. global regression) for monotonically decreasing covariance functions. Our method can determine when training examples are redundant given the rest of the database, and use this criteria for pruning. We report results on synthetic (Poser) and real (Humaneva) pose databases, obtaining fast and accurate pose estimates using training set sizes up to 10^5.

1. Introduction

Learning a mapping from visual observation to articulated body configuration is the foundation of discriminative approaches to pose estimation; such methods have recently become popular due to their ability to estimate pose from a single image without initialization. We are interested in the discriminative inference of arbitrary poses without restriction to a relatively limited set of predefined activities, e.g., running or walking, and wish to have a method which can perform inference efficiently enough to provide pose estimates at interactive rates (i.e. near-real time). Learning such a transformation is extremely challenging, due to the multimodality of the mapping, the high dimensionality of the input and output spaces, and the fact that activity-independent pose mappings have considerable variability and therefore require very large training sets to be accurately defined.

In this paper we develop a method to learn a complex appearance-to-pose mapping for arbitrary motions using probabilistic regression. We take advantage of Gaussian Process (GP) models, which offer a general framework for probabilistic regression and have been shown to generalize well when the training data are few in number [16, 23]. However, current GP models are limited in their ability to handle large training sets, allowing at most a few thousand training examples [8, 14, 20]. Also, in their standard form, GP models do not directly handle multimodality, and assume a single set of hyperparameters is sufficient to model the distribution of the data. Adapting the models locally is critical for human pose estimation since the training data density, noise levels and/or smoothness may vary considerably across the pose space.

We propose a new sparsification technique for Gaussian Processes, where local regressors are defined online for each test point. Local neighborhoods are very small, so training and inference are efficient. The use of a GP framework offers accurate probabilistic pose estimates from small neighborhoods, and naturally defines a redundancy criteria for pruning. Our method's computational complexity and memory requirements are dramatically reduced when compared to classic GP inference: inference is very fast with large databases of hundreds of thousands of examples. By using an online strategy our technique adapts to local regions of the space and does not suffer from the boundary problems that can affect static sparsification techniques or offline mixture models. Our method handles multimodality by forming online mixture components which are local both in terms of appearance and pose.

We next review related work, and then present our online local probabilistic regression framework. We demonstrate how redundancy detection and pruning is possible within our approach, and derive a bound on the error induced by our approximation when the mapping is a function. Finally, we show accurate pose estimation results on both synthetic and real images of hands and whole body poses, using a variety of input feature types, with databases ranging from 10^2 to 10^5 examples.

2. Related Work

Discriminative learning-based approaches to pose estimation avoid the use of expensive likelihood functions and the need for initialization by directly learning a mapping from image observations to pose. Discriminative Conditional Models [19] represent multimodal mappings with

a mixture of experts (e.g., Gaussian kernel regressors). Sparse regression approaches (e.g., RVM-based [1]) have been shown to infer pose using a restricted subset of informative examples. However, the latter can not handle multimodality when inferring pose from a single image. In addition, both methods and related techniques have been limited to mappings that could be accurately learned from a few thousand examples, and therefore domains with a relatively small set of activities. Ramanan et al. [15] propose methods for "parsing" and dynamic bootstrapping of human body models based in part on the efficient inference technique in [4]; their method is one of the few that are not restricted to specific classes of activity, but has been evaluated generally only on 2-D tasks.

GP-based Latent Variable Models (GP-LVM) have been shown to allow compact and effective description of activity-specific human pose [23, 21, 12] and dynamics priors [22, 13]. However, modeling an activity-independent appearance-to-pose mapping with a GP-LVM is computationally intractable due to the size of the training set required. In contrast with GP-LVM based methods, the GP approach we propose does not enforce a low dimensional representation and can handle general pose.

Gaussian Process training is well known to require $\mathcal{O}(N^3)$ time complexity, where N is the size of the training set. Existing GP sparsification techniques approximate the covariance matrix with an active set [8] or a set of inducing variables [14, 20], but they still have been limited to a few thousand training examples. In contrast, our online sparsification can handle efficiently very large training sets.

Our method draws inspiration from locally weighted regression Nearest-Neighbor techniques [3]. Local approaches to learning the appearance-to-pose mapping are generally appealing as the individual mappings are less complex, can be individually learned from fewer examples, and can be adapted to local regions that might have very different behaviors. When accurate estimates can be obtained from compact support neighborhoods, learning and inference can be both fast and accurate. In our experience online local models can accommodate orders of magnitude more examples than would be possible with a global model. Previous Nearest-Neighbor approaches [18, 2] had no provision to determine whether a certain number of examples was sufficient, no direct probabilistic interpretation, and did not provide a pruning algorithm to determine whether examples are redundant.

Existing local approaches employ mixture of experts [19, 17] that are learned offline. Offline partitioning can be computationally expensive with very large training sets, and can suffer from accuracy problems at the boundary of the experts unless there are many overlapping local models. At the limit one might need a local model for each training point, which is what our online approach computes.

In the remainder of the paper we review Gaussian Processes, present our local online approach, and show the effectiveness of the method in synthetic and real-world scenarios.

3. Gaussian Processes Review

Gaussian Processes have become popular because they are simple to implement, flexible (i.e. they can capture complex behaviors through a simple parameterization), and fully probabilistic. The latter enables them to be easily incorporated in more complex systems, and provides an easy way of expressing and evaluating prediction uncertainty.

A Gaussian Process is a collection of random variables, any finite number of which have consistent joint Gaussian distributions [14]. Given a training set $\mathcal{D} = \{(\mathbf{x}_i, \mathbf{y}_i), i = 1, \cdots, N\}$, composed of inputs $\mathbf{x}_i$ and noisy outputs $\mathbf{y}_i$, we assume that the noise is additive, independent and Gaussian, such that the relationship between the function, $f(\mathbf{x})$, and the observed noisy targets, $\mathbf{y}$, are given by

$$\mathbf{y}_i = f(\mathbf{x}_i) + \epsilon_i \,, \tag{1}$$

where $\epsilon_i \sim \mathcal{N}(0, \sigma^2_{noise})$ and σ^2_{noise} is the noise variance.

GP regression is a Bayesian approach that assumes a GP prior over functions,

$$p(\mathbf{f}|\mathbf{X}) = \mathcal{N}(0, \mathbf{K}) \,, \tag{2}$$

where $\mathbf{f} = [f_1, \cdots, f_n]^T$ is the vector of function values, $f_i = f(\mathbf{x}_i)$, $\mathbf{X} = [\mathbf{x}_1, \cdots, \mathbf{x}_N]^T$, and $\mathbf{K}$ is a covariance matrix whose entries are given by a covariance function, $K_{i,j} = k(\mathbf{x}_i, \mathbf{x}_j)$. GPs are non-parametric models and are entirely defined by their covariance function (and training data); the set of possible covariance functions is defined by the set of Mercer kernels. In practice, we use a covariance function which is the sum of an RBF, a bias term, and a noise term, all with hyperparameters $\bar{\beta}$. During training, the model parameters, $\bar{\beta}$, are learned by minimizing

$$-\ln p(\mathbf{X}, \bar{\beta} \,|\, \mathbf{Y}) = \frac{D}{2} \ln |\mathbf{K}| + \frac{1}{2} \mathrm{tr}\left(\mathbf{K}^{-1}\mathbf{Y}\mathbf{Y}^T\right) + C \,, \tag{3}$$

where $\mathbf{Y} = [\mathbf{y}_1, \cdots, \mathbf{y}_N]^T$, C is a constant, and D is the dimension of the output.

GP inference and each iteration of training classically requires inverting a $(N \times N)$ matrix. With $\mathcal{O}(N^3)$ matrix inversion cost, this becomes computationally prohibitive for large training sets.

Different techniques have been proposed to statically sparsify Gaussian Processes and reduce their computational complexity. Several approaches simply select a subset of the data; this is similar in spirit to the selection of support vectors in SVM. Selection criteria are typically based on mutual information and sparsification proceeds by selecting points that are not well predicted by the reduced Gaussian Process [8]. The computational complexity is then reduced

to $\mathcal{O}(m^3)$, where m is the cardinality of the subset. But when dealing with large training sets, to reduce the computational complexity to an affordable value, these techniques are usually not accurate since the subset is sparse. More sophisticated techniques introduce a set of inducing variables and under some independence assumptions, one can reduce the computational complexity to $\mathcal{O}(Nm^2)$, where m is the number of inducing variables and $m << N$. Although these techniques are more accurate than using only a subset of the data, when the amount of training data is very large and the input dimensionality is very high, the computational cost still remains prohibitive, since the number of inducing variables required to represent the GP might still be very high (m large). Moreover, this might introduce overfitting, since now one has to estimate not only the kernel hyperparameters but also the inducing variables. We now introduce a new sparsification technique that reduces considerably the computational complexity of GPs, making possible to do learning and inference with very large datasets.

4. Online Local Learning of Appearance-to-Pose Mappings

In this paper we present a local probabilistic regression approach to learn multimodal appearance-to-pose mappings. Our model is online, and forms local models at runtime for each new test point (§4.1). We combine appearance and pose information to deal with multimodal mappings: we define multiple GP-based experts, each expert focusing on a mode of the pose distribution (§4.2). Inference in our framework is computationally inexpensive since the local experts are defined in very small neighborhoods, typically at most 50 neighbors (§4.3).

4.1. Online Local Gaussian Processes

In this section we propose a new sparsification technique that reduces the computational complexity and memory requirements of Gaussian Processes.

When using monotonically decreasing covariance functions (e.g. RBF), the covariance matrix is sparse; $k_{i,j}$ is very small for all the entries where $||\mathbf{x}_i - \mathbf{x}_j||$ is large. A full GP can then be approximated locally online by a much smaller GP centered at the given test point, reducing considerably the computational cost and allowing learning and inference with extremely large datasets. Note the difference in philosophy of the static and online sparsification techniques: static sparsification reduces the amount of training data use to do inference globally, while online sparsification approximates the covariance matrix locally but retains all the data for future inference. When the mapping is multimodal, the local mapping is more accurate than the global one (see Fig. 1). By centering the neighborhoods that define each local expert at the test point, we can avoid the boundary problems that static sparsification based on clustering can suffer from. In Appendix A we provide a bound of the approximation of a local mapping vs a global mapping in the case where the mapping is unimodal, a single set of hyperparameters is enough to accurately represent the mapping and the covariance function is monotonically decreasing with respect to the input distance.

To speed up inference we do not learn kernel hyperparameters at test time, and instead precompute a set of local models from which to determine run-time hyperparameters. Given the observation that local neighborhoods behave similarly, one can estimate the hyperparameters for only a subset R of all the possible sets of local GPs; the hyperparameters for each local expert are simply set to the hyperparameters of the learned local GP closest in pose space. The R local experts were selected at random in our experiments. Similar results were obtained by selecting them using a clustering algorithm.

Inference in the new sparse GP model is straightforward. Assuming a joint GP prior over training, $\mathbf{f}$, and testing, $\mathbf{f}_*$, variables, marginalizing the training variables can be done in closed form and yields a Gaussian predictive distribution, $p(\mathbf{f}_*|\mathbf{Y}) = \mathcal{N}(\mu, \sigma)$, with

$$\mu(\mathbf{x}_*) = K_{*,\zeta}(\mathbf{K}_{\zeta,\zeta} + \sigma^2_{noise}\mathbf{I})^{-1}\mathbf{Y}_\zeta \quad (4)$$

$$\sigma(\mathbf{x}_*) = k_{*,*} - K_{*,\zeta}(\mathbf{K}_{\zeta,\zeta} + \sigma^2_{noise}\mathbf{I})^{-1}K_{\zeta,*} \quad (5)$$

where $\mathbf{x}_*$ is the input test data, ζ are the indices of the local neighbors of $\mathbf{f}_*$, $\mathbf{K}_{\zeta,\zeta}$ is the covariance of the local neighborhood, $k_{*,*}$, the covariance of the test data, and $\mathbf{K}_{\zeta,*} = \mathbf{K}^T_{*,\zeta}$ is the cross-covariance of the local neighborhood and test data. To avoid clutter in the notation we have dropped the dependency on the training data.

4.2. Handling Multimodality

A Gaussian Process can only model functions. As a consequence, when dealing with multimodal outputs classical GP prediction would average the modes, resulting in an extremely inaccurate mapping. This problem is illustrated in Fig. 1 where we generate data from a multimodal distribution with two modes. The global GP Fig. 1(f) predicts none of the modes, but provides a mean estimate that averages the true modes. To solve this problem we ensure that the local neighbors of each expert only contain examples of one mode by defining each GP to be consistent in pose space: each local GP is composed of examples that are local in pose as well as appearance yielding accurate multimodal regression (Fig. 1(h)) .

We combine different local experts to produce estimates of different modes, where each expert is centered on a neighbor in appearance of the test point. (Note that one cannot choose the neighbors based on pose for the test data since that is unknown at test time!) If the mapping is multimodal in this region, each expert will provide an estimate of the mode to which the neighbor in appearance belongs.

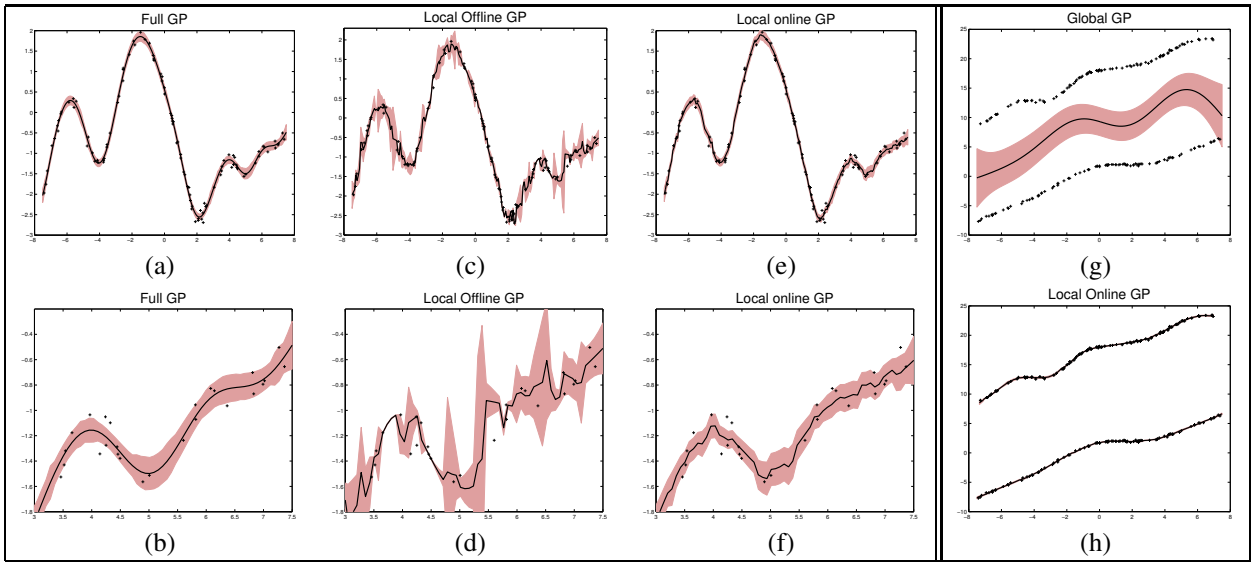

Figure 1. **Advantages of online local regression**: (a–f) An underlying function is obtained by sampling ($N = 100$) a Gaussian Process with covariance matrix obtained from an RBF with hyperparameters $[0.1, 0.1]$, and additive noise with variance $\sigma^2_{noise} = 0.1$. Test data ($N_t = 200$) are uniformly sampled on the interval $[-7.5, 7.5]$. The prediction variance is depicted in pink. (a) Results with a full GP with the hyperparameters used to generate the data. As expected, this GP perfectly fits the test data. (c) Results using a local *offline* GP, where the data is clustered in input space to produce 10 local GPs each with 10 training examples. Results with a local *online* GP (e), where we learn hyperparameters from 5 randomly selected points and use local GPs of size 10. Note how accurately the local *online* GPs represent the mean and variance of the original GP. The *offline* local GP overestimates the variance in points close to boundaries, and the prediction is noisier. (b,d,f) Shows the mappings of (a,c,e) in more detail. (g-h) For data from a multimodal distribution a global GP (g) averages the two modes with high variance, while the local GP (h) gives a good estimate of both modes (variance is accurately estimated but so low as to not be visible in (h)).

The number of experts, T, and the size, S, of each local GP, are parameters of our model. In practice, as shown in Section 6, small values of both parameters are sufficient to produce accurate estimates. The predictive distribution is

$$p(\mathbf{f}_*|\mathbf{Y}) \approx \sum_{i=1}^{T} \pi_i p(\mathbf{f}_*|\mathbf{Y}_{\zeta^i}) = \sum_{i=1}^{T} \pi_i \mathcal{N}(\mu_i, \sigma_i^2)\,, \qquad (6)$$

where $\mathbf{Y}_{\zeta^i}$ are the neighbors in pose of $\mathbf{y}_{\eta_i}$, $\mathbf{X}_\eta$ are the neighbors in appearance of $\mathbf{x}_*$, $\eta = \{\eta_j\}$, π_i is the probability of a given expert, and μ_i, σ_i^2, are the predictive mean and variance of the i-th expert, defined in Eqs. 4 and 5. Each π_i is set to be a function of the inverse variance of the prediction of that expert. The final algorithm is summarized in Algorithm 1, where the function $findNN(\mathbf{X}, \mathbf{x}', S)$ selects the S nearest neighbors of $\mathbf{x}'$ in $\mathbf{X}$.

4.3. Computational Complexity

Table 1 compares the complexity of our technique with respect to the complexity of estimating a full GP. In our approach the only factor that grows with the size of the database is finding the nearest neighbors. The complexity of inverting the local GPs is not a function of the number of examples, since the local GPs are of fixed size.[1] When dealing with very large databases, the computational time of computing the exact nearest neighbors might be prohibitive (i.e. linear time). Using tree-based [10] or random hash function-based [18] approximate nearest neighbor methods can reduce this cost to be sublinear time.

In particular, the computational complexity of learning

[1] One can precompute and/or cache these inverses to gain efficiency. But since the local experts are computed with very small neighborhoods, they can be directly computed without much decrease in performance.

Algorithm 1 Learning and inference with a mixture of Online local GPs

OFFLINE: Learning hyperparameters
R: number of local GP to learn
for $n = 1 \ldots R$ **do**
  $i = rand(N)$
  $\kappa = findNN(\mathbf{X}, \mathbf{x}_i, S)$
  $\{\bar{\beta}^i\} \leftarrow \max\ p(\mathbf{X}_\kappa, \bar{\beta}^i|\mathbf{Y}_\kappa)$
  $\mathbf{Y}_R = [\mathbf{Y}_R, \mathbf{y}_i]$
end for
ONLINE: Inference of test point $\mathbf{x}_*$
T: number of experts, S: size of each expert
$\eta = findNN(\mathbf{X}, \mathbf{x}_*, T)$
for $j = 1 \ldots T$ **do**
  $\zeta = findNN(\mathbf{Y}, \mathbf{y}_{\eta_j}, S)$
  $t = findNN(\mathbf{Y}_R, \mathbf{y}_{\eta_j}, 1)$
  $\bar{\beta} = \bar{\beta}^t$
  $\mu_j = K_{*,\zeta}(\mathbf{K}_{\zeta,\zeta} + \sigma^2_{noise}\mathbf{I})^{-1}\mathbf{Y}_\zeta$
  $\sigma_j = k_{*,*} - K_{*,\zeta}(\mathbf{K}_{\zeta,\zeta} + \sigma^2_{noise}\mathbf{I})^{-1}K_{\zeta,*}$
end for
$p(\mathbf{f}_*|\mathbf{y}) \approx \sum_{i=1}^{T} \pi_i \mathcal{N}(\mu_i, \sigma_i^2)$

	Learning	Inference
Our approach	$O(RS^3 + RN)$	$O(TS^3 + TN)$
Global GP	$O(N^3)$	O(N^2)

Table 1. **Computational complexity:** our method is linear in N for both learning and inference. Note that in all the examples shown in the paper $S \leq 50$, $T \leq 10$ and $R \leq 500$, while $N \geq 1000$.

in our approach is the time of learning the R local experts of size S, $O(RS^3)$, plus the time of estimating its neighbors, $O(RN)$, where N is the total number of examples (see Table 1). In contrast, the cost of learning a full GP is $O(N^3)$. In our experiments $R \leq 500$, and as a result even with a few thousand examples the complexity of our model is much smaller than that of a full GP.

For inference the computational savings are also important, resulting in an algorithm that can perform inference in near real time. For all experiments reported in this paper, the framerate with un-optimized Matlab code was at least two frames per second, not including the time to compute features.

5. Detecting Outliers and Pruning

When learning from examples, certain questions need to be addressed. Should I trust the predictor? Is my training data sufficient? Do I need more examples? If yes, where do I need to populate my training data? Are all the examples necessary? In this section we show how the uncertainty in the prediction can be used to answer these questions.

The probabilistic nature of GPs allows us to determine whether to trust the prediction.We detect outliers of the prediction by detecting high variances of the local experts. Fig. 2 depicts the test mean error as a function of the rank variance (i.e., percentage of the test data with variance smaller than a threshold). The variance is a good indicator of the uncertainty since the curves are monotonically increasing; the detected outliers are those test points where the local GPs have the biggest error. Moreover, this uncertainty can also be used to identify where the database is undersampled, and therefore where we want to acquire more data, or where it is oversampled, and where we want to prune; these are respectively the regions of the space with high and low variance.

The probabilistic nature of GPs can be utilized to determine how many examples are necessary. If a training point can be accurately predicted in terms of mean and variance by its neighbors, it is no longer necessary and can be removed. Our method looks iteratively at each training point $\{\mathbf{x}_i, \mathbf{y}_i\}$, and sees whether it is redundant by computing the KL divergence of the predictive distributions with or without that point, $KL(p(\mathbf{f}_*|Y_\eta)||p(\mathbf{f}_*|Y_{\hat{\eta}}))$. Since $p(\mathbf{f}_*|Y_\eta)$ and $p(\mathbf{f}_*|Y_{\hat{\eta}})$ are both Gaussian, the KL divergence can be computed in closed form. Our pruning scheme is summarized in Algorithm 2.

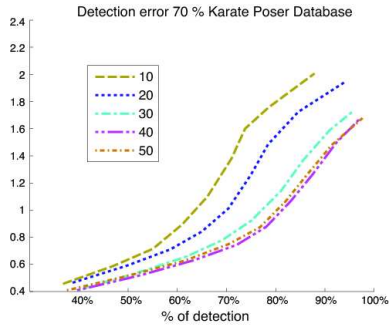

Figure 2. **Detection of outliers**: for a database composed of $15,000$ Poser examples of 30 different activities, where 70% of the dataset is used for training. Mean errors (degrees) as a function of the number of inliers (detection percentage) for different sizes of the local experts are depicted. As the curves monotonically increase, the variance becomes a good measure of uncertainty.

Algorithm 2 Pruning scheme

for $i = 1 \ldots N$ **do**
  for $j = 1 \ldots T$ **do**
    $\eta = findNN(\mathbf{Y}, \mathbf{y}_j, S)$
    $\hat{\eta} = \eta_{1:S-1} \bigcup i$
    $t = findNN(\mathbf{Y}_R, \mathbf{y}_j, 1)$
    $\bar{\beta} = \bar{\beta}^t$
    $KL_{i,j} = KL(p(f_*|\mathbf{Y}_\eta) \,||\, p(f_*|\mathbf{Y}_{\hat{\eta}}))$
  end for
  if $\min(KL_i) < Thr$ **then**
    prune $\{\mathbf{x}_i, \mathbf{y}_i\}$
  end if
end for

6. Experimental Evaluation

We validate our approach using synthetic hand and whole body figures rendered from motion capture data [11] using Poser and with real-images from the benchmark Humaneva [6] database composed of sequences of different subjects performing various activities.

In the first experiment we demonstrate our method across a wide range of training set sizes, from a very restrictive set of activities (small training set) to a very broad general motion database (large training set), as summarized in Table 2. 3D Pose was represented as a vector of 47 joint angles and was estimated from Chamfer distances to silhouettes. The $15,000$ example database is composed of 30 different activities of combat sports. The activities are comprised of fast motions, and vary from kicking, punching, to receiving kicks. The $50,000$ example dataset from [11] consisted of 60 different activities of combat sports, soccer and dancing (see Fig. 3), including ballet, techno and twist dancing. In the soccer examples, we have motions of the referee, the goal-keeper, regular player and the coach. We compare the results of our method for different sizes of the local experts in two different scenarios as depicted by Table 2 and Fig. 4. In the first scenario we take the expert that produces the minimum variance (solid blue) and compare it to the first

DB size	1-NN	Best of-10-NN	GP ($S = 10$)	GP ($S = 20$)	GP ($S = 30$)	GP ($S = 40$)
1,500	0.88 ± 1.77	0.71 ± 1.38	0.83 ± 1.53	0.98 ± 1.70	$\mathbf{0.56 \pm 1.40}$	0.70 ± 1.45
15,000	1.92 ± 2.76	1.49 ± 1.81	1.32 ± 2.07	1.10 ± 1.88	1.03 ± 1.81	$\mathbf{0.99 \pm 1.77}$
50,000	1.83 ± 2.62	1.34 ± 1.47	1.10 ± 1.85	0.91 ± 1.64	0.90 ± 1.66	$\mathbf{0.87 \pm 1.58}$

Table 2. **Inference in a wide range of training set sizes:** our method produces accurate results (less than 1 degree) in a wide range of training data sizes.

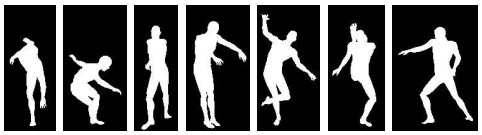

Figure 3. **Samples from large database** with $50,000$ examples of Poser generated silhouettes. The difficulty of the data is due to ambiguities inherent to silhouettes, occlusions, and the large variation in poses and viewpoints.

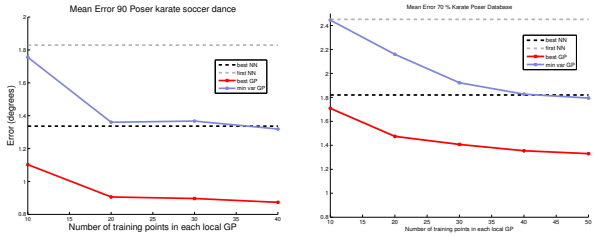

Figure 4. **Inference in very large datasets** (left) $15,000$ and (right) $50,000$ example Poser database composed of multiple activities. The mean errors were smaller than 1 degree.

Nearest-Neighbor (NN) (dashed gray). In the second scenario, we assume that we have a process that can choose the expert with the minimum error (e.g. from dynamics in a tracking framework). We then compare the mean error made by the best local GP among 10 experts vs. the best NN of 10 neighbors. Note that in both scenarios the GP significantly outperforms the NN approach. The large difference between 1-NN and the best among 10-NN is an indicator of the ambiguities inherent in estimating 3D pose from single images for the set of activities present in the database. The local experts require very few training points ($\leq$ 50) and are very accurate, with approximately 1 degree of mean test error.

Our method is general and can work with varying image features and pose representations. In addition to the previous experiments using Chamfer distance, we have obtained accurate results with similarity measures based on Hierarchical features [7] and with the Pyramid Match Kernel (PMK) [5] defined with SIFT, Steerable Filters, and Shape Context base features. Fig. 5 depicts the results for these measures: with Hierarchical features, results with position error as low as 0.35 mm were obtained. We also trained a global GP and evaluated the best-of-ten-nearest-neighbors method on this dataset to compare local vs. global performance; this was feasible in this case, since there were only 1,000 training points, but in general was impossible for our larger datasets. A global GP trained with all examples us-

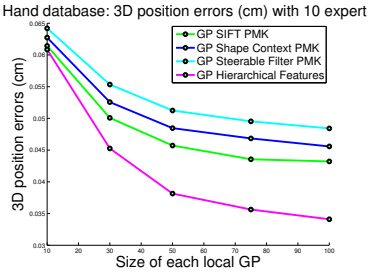

Figure 5. **Generalization to Different kernels**: our mixture of local online GPs provides accurate results with kernels based on different features. Mean prediction errors for a hand database when using Pyramid Match kernels [5] based on SIFT, Shape Context, Steerable filters, or Hierarchical features [7] are shown.

Figure 6. **Hand Database**: The 3D pose is estimated and reprojected in the original image for comparison. Note that we estimate the joint locations, and thus there is no joint at the tip of the fingers.

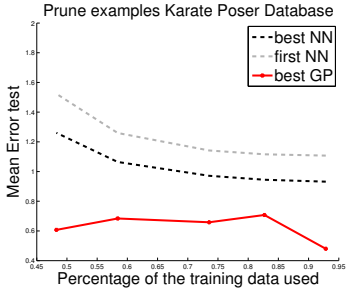

Figure 7. **Pruning**: a Poser database composed of $1,500$ training points. 50% of the database can be pruned with almost no test error increase. 1-NN and best among 10 NN baselines are shown.

ing Hierarchical features on this task yielded errors of 0.40 mm, and the best-of-ten-NN method yielded error of 0.60 mm indicating that local models both offer increased efficiency and the ability to handle multimodality, potentially improving performance over a global mapping.

We demonstrate our pruning algorithm in the database of karate motions. Fig. 7 depicts the mean error as a function of the amount of training data pruned, showing that with constant test error, one can prune up to 50% of the database. The fluctuations in the curve might be due to the iterative scheme, where the pruning order might be relevant. To avoid this problem, one can randomly choose the train-

Figure 8. **Humaneva Database**: Our pose estimation is depicted in red.

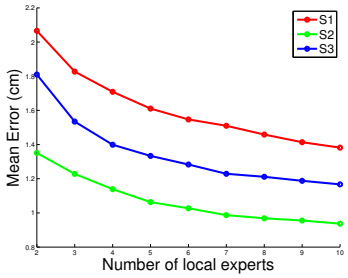

Figure 9. **Influence of the number of local experts:** Only a small number of online experts is necessary per test point for accurate prediction. The 3D mean error in (cm) is depicted for the three subjects we use in our experiments.

ing points to evaluate whether they can be pruned or not.

Experiments were also run on the Humaneva benchmark database, using the hierarchical features of [7]; these features do not require the computation of a precise bounding box around the object of interest. In the first experiment, different percentages of the database where used for training on a per-frame basis, and both 2D and 3D pose were inferred from the remaining examples. Table 3 depicts the errors for the different subjects and percentages of the database. In these experiments the errors are smaller than 1.5 cm in 3D and 2 pixels in 2D. In this database the size of human bodies ranges from 200 to 300 pixels. Fig. 9 shows the error as a function of the number of local experts, T. Note that only a small number of experts per test point is required to accurately infer the pose.

In the second Humaneva experiment, we divided the training data per sequence such that one sequence was use for training and a different sequence was used for testing. Note that even though the poses and appearances in the training and testing data might have been relatively different, our method accurately infers the 2D and 3D poses (see Table 4 and Fig. 8). For comparison, Table 5 shows errors reported in Humaneva-I [6]; all in the range of 3.1 to 12 cm, the best result being reported by [9] for walking sequences. Note that even though those errors are not directly comparable since they have been reported in different sequences and with different error metrics, our technique performs comparatively well.

7. Conclusions

In this paper we presented an online sparse probabilistic regression scheme for efficient inference of complex, high-dimensional, and multimodal mappings defined from very large training sets such as the activity-independent appearance of human pose. Previous approaches to learning such mappings were limited in their ability to provide a probabilistic estimate of pose given appearance and to work in domains that require very large training sets. We develop an online approach to GP sparsification that centers local regressors at each test point, avoiding the boundary problems inherent in offline (clustering) approaches. Our method works efficiently across a wide range of training set sizes, can handle multimodal mappings, can learn hyperparameters specific to local regions of the space, and can prune database examples based on probabilistic criteria. We presented results showing accurate pose estimation from single frames on synthetic (Poser) and real-world (Humaneva) pose inference tasks with thousands to tens of thousands of examples. In future work, we plan to build an ever larger database and learn a distance metric providing invariance to clothing.

	walk	jog	box	mono.	discrim.	dyn.
Lee et al. I	3.4	–	–	yes	no	no
Lee et al. II	3.1	–	–	yes	no	yes
Pope	4.53	4.38	9.43	yes	yes	no
Muendermann et al.	5.31	–	4.54	no	no	yes
Li et al.	–	–	20.0	yes	no	yes
Brubaker et al.	10.4	–	–	yes	no	yes
Our approach	3.27	3.12	3.85	yes	yes	no

Table 5. **Humaneva I comparative results:** reported 3D errors in cm in the EHUM-I and EHUM-II workshops [6]. Note that this results are not directly comparable since different methods have different error measures and they are test in different sequences. The best error performance was reported by Lee et al. [9] using a generative model with dynamics, however the error measure was normalized, the global orientation error was removed.

Acknowledgments

We thank Robert Wang for generating the Poser Hand Database and the authors of [7] for sharing their executable to compute the Hierarchical features.

Appendix A: Approximation Bound

Assuming that the mapping can be well represented with a single set of hyperparameters, and that they are known, the change in the uncertainty of the prediction at any given point $\mathbf{x}_*$ by removing a training point $\mathbf{x}_{S+1}$ is

$$\Delta\sigma^2(\mathbf{x}_*) = \sigma_S^2 - \sigma_{S+1}^2 = \frac{(\mathbf{k}_S \hat{\mathbf{K}}_S^{-1}\mathbf{m} - k(\mathbf{x}_{S+1}, \mathbf{x}_*))^2}{k(\mathbf{x}_{S+1}, \mathbf{x}_{S+1}) - \mathbf{m}^T \mathbf{K}_S^{-1}\mathbf{m}} \quad (7)$$

where $\hat{\mathbf{K}}_S = [\mathbf{K}_S + \sigma_{noise}^2 \mathbf{I}]$, $\mathbf{K}_S$ is the covariance function formed from $\mathbf{X}_S = [\mathbf{x}_1, \cdots, \mathbf{x}_S]^T$, $\mathbf{k}_S = [k(\mathbf{x}_1, \mathbf{x}_*), \cdots, k(\mathbf{x}_S, \mathbf{x}_*)]^T$, σ_i^2 is the uncertainty when using a GP with i training points, and $\mathbf{m} = [k(\mathbf{x}_1, \mathbf{x}_{S+1}), \cdots, k(\mathbf{x}_S, \mathbf{x}_{S+1})]^T$.

Let $\mathbf{x}_{S+1}, \cdots, \mathbf{x}_N$ be a set of points ordered by their minimum distance to $\{x_i\}, i = 1, \cdots, S$. Assuming that

	2D Error (pixels)			3D Error (cm)		
% database	50 %	80 %	90 %	50 %	80 %	90 %
S1	2.01 ± 1.80	1.44 ± 1.46	1.28 ± 1.35	1.37 ± 1.06	0.94 ± 0.90	0.91 ± 0.97
S2	1.22 ± 1.05	0.87 ± 0.85	0.80 ± 0.86	0.93 ± 0.65	0.68 ± 0.55	0.60 ± 0.51
S3	1.67 ± 1.69	1.33 ± 1.57	1.24 ± 1.76	1.16 ± 0.91	0.94 ± 0.81	0.88 ± 0.89

Table 3. **Humaneva dataset:** Mean errors in (cm) and (pixels) when using different percentages of the database for training and testing. Our approach accurately estimates the pose, with maximum errors of 1 cm and 2 pixels. In this database the size of human figures ranged from 200 to 300 pixels.

	2D Error (pixels)			3D Error (cm)		
	Walking	Jog	Box	Walking	Jog	Box
S1	4.89 ± 2.47	6.05 ± 3.00	6.43 ± 3.87	3.14 ± 1.36	3.71 ± 1.41	3.75 ± 1.93
S2	2.76 ± 1.31	5.88 ± 2.54	5.53 ± 2.66	1.93 ± 0.71	3.76 ± 1.14	3.31 ± 1.44
S3	7.90 ± 3.21	2.63 ± 0.97	8.09 ± 7.44	4.74 ± 1.69	1.89 ± 0.75	4.51 ± 3.55

Table 4. **Generalizing to different sequences:** with the Humaneva benchmark, our method is trained with one sequence and tested in another sequence of the same subject. Note that those sequences can be quite different, and thus the training data can thus be relatively sparse. Errors in pixels for 3 different subjects and 3 activities are depicted.

$k(\mathbf{x}, \mathbf{x}')$ is a monotonically decreasing funtion of $||\mathbf{x} - \mathbf{x}'||$ (e.g., RBF), the uncertainty can be bounded by

$$\Delta\sigma^2(\mathbf{x}_*) \leq \sum_{i=S}^{N-1} \frac{(\mathbf{k}_S \hat{\mathbf{K}}_S^{-1}\mathbf{m} - k(\mathbf{x}_{i+1}, \mathbf{x}_*))^2}{k(\mathbf{x}_i, \mathbf{x}_i) - \mathbf{m}^T \hat{\mathbf{K}}_S^{-1}\mathbf{m}} \qquad (8)$$

where we have used the fact that the uncertainty decrement, $-\Delta\sigma^2(\mathbf{x}_*)$, has a minimum when considering the closest point (i.e., $\mathbf{x}_{S+1}$) to the query $\mathbf{x}_*$.

The contribution of each point is a function of the RBF width and its distance to $\{\mathbf{x}_i\}, i = 1, \cdots, S$. In practice, the contribution of most of the points is negligible. In the worst case scenario, all the excluded points are at a fixed radius $Q = ||\mathbf{x}_{S+1} - \mathbf{x}_*||$, i.e./ at the boundary. Then

$$\Delta\sigma^2(\mathbf{x}_*) \leq (N - S) \frac{(\mathbf{k}_S \hat{\mathbf{K}}_S^{-1}\mathbf{m} - k(\mathbf{x}_{S+1}, \mathbf{x}_{S+1}))^2}{k(\mathbf{x}_{S+1}, \mathbf{x}_{S+1}) - \mathbf{m}^T \hat{\mathbf{K}}_S^{-1}\mathbf{m}} . \quad (9)$$

This bound can be used to set the size of the local experts.

References

[1] A. Agarwal and B. Triggs. Recovering 3D human pose from monocular images. *PAMI* 28(1):44–58, January 2006.

[2] V. Athitsos and S. Sclaroff. Database indexing methods for 3d hand pose estimation. In *Gesture Workshop*, 2003.

[3] C. G. Atkeson, A. W. Moore and S. Schaal. Locally weighted learning. *Artif. Intell. Rev.*, 11:11–73, 1997.

[4] P. F. Felzenszwalb and D. P.Huttenlocher Efficient matching of pictorial structures . *CVPR* 2000.

[5] K. Grauman and T. Darrell. The Pyramid Match Kernel: Discriminative Classification with Sets of Image Features. *ICCV* Beijing, China, October 2005.

[6] Humaneva. http://vision.cs.brown.edu/humaneva/.

[7] A. Kanaujia, C. Sminchisescu, and D. Metaxas. Semi-supervised Hierarchical Models for 3D Human Pose Reconstruction. *CVPR* 2007.

[8] N. D. Lawrence, M. Seeger, and R. Herbrich. Fast sparse Gaussian process methods: The informative vector machine. *NIPS* pages 609–616. 2003.

[9] C.S. Lee and A. Elgammal. Body Pose Tracking From Uncalibrated Camera Using Supervised Manifold Learning. In *NIPS EHUM Workshop*, 2006.

[10] T. Liu, A. Moore, A. Gray. Efficient exact k-nn and nonparametric classification in high dimensions. *NIPS* 2003.

[11] Mocap data. http://www.mocapdata.com.

[12] R. Navaratnam, A. Fitzgibbon and R. Cipolla. The Joint Manifold Model for Semi-supervised Multi-valued Regression. *ICCV* Rio de Janeiro, Brazil, October 2007.

[13] K. Moon and V. Pavlovic. Impact of Dynamics on Subspace Embedding and Tracking of Sequences. *CVPR* 2006.

[14] J. Quiñonero-Candela and C. E. Rasmussen. A unifying view of sparse approximate gaussian process regression. *Journal of Machine Learning Research*, 2005.

[15] D. Ramanan, D.A.Forsyth and A. Zisserman. Tracking people by learning their appearance. *PAMI* 2007.

[16] C. E. Rasmussen and C. K. Williams. *Gaussian Process for Machine Learning*. MIT Press, 2006.

[17] R. Rosales and S. Sclaroff. Infering Body Pose without Tracking Body Parts. *CVPR* pages 506–511, 2000.

[18] G. Shakhnarovich, P. Viola, T. Darrell. Fast pose estimation with parameter-sensitive hashing. *ICCV* 2003.

[19] C. Sminchisescu, A. Kanaujia, Z. Li, and D. Metaxas. Discriminative Density Propagation for 3D Human Motion Estimation. *CVPR* 2005.

[20] E. Snelson and Z. Ghahramani. Sparse gaussian processes using pseudo-inputs. *NIPS*, 2006.

[21] T. Tian, R. Li, and S. Sclaroff. Articulated Pose Estimation in a Learned Smooth Space of Feasible Solutions. In *CVPR Learning Workshop*, volume 3, San Diego, CA, 2005.

[22] R. Urtasun, D. J. Fleet, and P. Fua. 3D people tracking with gaussian process dynamical models. *CVPR* 2006.

[23] R. Urtasun, D. J. Fleet, A. Hertzman, and P. Fua. Priors for people tracking from small training sets. *ICCV* pages 403–410, Beijing, China, October 2005.