Available online at www.sciencedirect.com
ScienceDirect
Pattern Recognition Letters 29 (2008) 1302–1310
Pattern Recognition Letters
www.elsevier.com/locate/patrec

Pose estimation and tracking using multivariate regression

Arasanathan Thayananthan [a], Ramanan Navaratnam [a], Björn Stenger [b,*], Philip H.S. Torr [c], Roberto Cipolla [a]

[a] University of Cambridge, Department of Engineering, Cambridge CB2 1PZ, UK
[b] Toshiba Cambridge Research Laboratory, Cambridge, CB4 0GZ, UK
[c] Oxford Brookes University, Department of Computing, Wheatley, Oxford OX33 1HX, UK

Received 1 September 2006; received in revised form 8 October 2007
Available online 20 February 2008

Communicated by R.P.W. Duin

Abstract

This paper presents an extension of the relevance vector machine (RVM) algorithm to multivariate regression. This allows the application to the task of estimating the pose of an articulated object from a single camera. RVMs are used to learn a one-to-many mapping from image features to state space, thereby being able to handle pose ambiguity.

Keywords: Regression; Relevance vector machines; Tracking; Articulated motion

1. Introduction

This paper considers the problem of estimating the 3D pose of an articulated object such as the human body from a single view. This problem is difficult due to the large number of degrees of freedom and the inherent ambiguities that arise when projecting a 3D structure into the 2D image (Brand, 1999; Howe et al., 1999). Once the pose estimation task is solved, temporal information can be used to smooth motion and resolve potential pose ambiguities. This divides continuous pose estimation into two distinct tasks: (1) estimate a distribution of possible configurations from a single frame, (2) combine frame-by-frame estimates to obtain smooth trajectories.

Generative methods estimate the pose by projecting a geometric model into the scene and evaluating a likelihood function that measures agreement with the image. Single frame pose estimation then becomes a complex optimization problem that can be approached with methods such as dynamic programming (Felzenszwalb and Huttenlocher, 2005); MCMC (Lee and Cohen, 2004) or hierarchical search (Stenger et al., 2006).

Discriminative methods follow a different approach by trying to learn a mapping from image features directly to the 3D pose. A straightforward way to do this is to generate a database of images from a 3D model and use efficient search to find the best match (Shakhnarovich et al., 2003; Stenger et al., 2006). The number of templates required to represent the pose space depends on the range of possible motion and required accuracy, and can be in the order of hundreds of thousands of templates. Only a fraction of the them is searched for each query image, however all templates need to be stored.

The method for hand pose estimation from a single image by Rosales et al. addressed some of these issues (Rosales et al., 2001). Image features were directly mapped to likely hand poses using a set of *specialized mappings*. A 3D model was projected into the image in these hypothesized poses and evaluated using an image based cost function. The features used were low-dimensional vectors of

[*] Corresponding author. Fax: +44 1223436909.
E-mail addresses: at315@eng.cam.ac.uk (A. Thayananthan), rn246@eng.cam.ac.uk (R. Navaratnam), bjorn@cantab.net (B. Stenger), philiptorr@brookes.ac.uk (P.H.S. Torr), cipolla@eng.cam.ac.uk (R. Cipolla).

0167-8655/$ - see front matter © 2008 Elsevier B.V. All rights reserved.
doi:10.1016/j.patrec.2008.02.004

silhouette shape moments, which are often not discriminative enough for precise pose estimation.

Agarwal and Triggs proposed a method for selecting relevant features using RVM regression (Agarwal and Triggs, 2004; Agarwal and Triggs, 2006). The image features were shape-contexts descriptors of silhouette points and pose estimation was formulated as a one-to-one mapping from the feature space to pose space. This mapping required about 10% of the training examples. The method was further extended to include dynamic information by joint regression with respect to two variables, the feature vector and a predicted state obtained with a dynamic model (Agarwal and Triggs, 2004).

The mapping from silhouette features to state space is inherently one-to-many, as similar features can be generated by regions in the parameter space that are far apart, see Fig. 1a. Hence, it is important to maintain multiple hypotheses over time. In this paper the pose estimation problem from template matching is formulated as learning one-to-many mapping functions that map from the feature space to the state space. The features are Hausdorff matching scores, which are obtained by matching a set of shape templates to the edge map of the input image, see Fig. 1b. A set of RVM mapping functions is then learned to map these scores to different state-space regions to handle pose ambiguity, see Fig. 1a. Each mapping function achieves sparsity by selecting a small fraction of the total number of templates. However, each RVM function will select a different set of templates. This work is closely related to the work of Sminchisescu et al. (2005), Agarwal and Triggs (2005) and Agarwal and Triggs (2006). Both follow a mixture of experts (Jordan and Jacobs, 1994) approach to learn a number of mapping functions (or experts). A gating function is learned for each mapping function during training, and these gating functions are then used to assign the input to one or many mapping functions during the inference stage. In contrast, we use likelihood estimation from projecting the 3D-model to verify the output of each mapping function.

The rest of the paper is organized as follows: the algorithm for learning the one-to-many mapping using multiple RVMs is introduced in Section 3. Section 4 describes a scheme for training a single RVM mapping function with multivariate outputs. The pose estimation and tracking framework is presented in Section 5, and results on hand and full body tracking are shown in Section 6.

2. Learning multiple RVMs

The pose of an articulated object, in our case a hand or a full human body, is represented by a parameter vector $\mathbf{x} \in \mathbb{R}^M$. The features $\mathbf{z}$ are Canny edges extracted from the image. Given a set of training examples or templates $\mathscr{V} = \{v^{(n)}\}_{n=1}^{N}$ consisting of pairs $v^{(n)} = \{(\mathbf{x}^{(n)}, \mathbf{z}^{(n)})\}$ of state vector and feature vector, we want to learn a one-to-many mapping from feature space to state space. We do this by learning K different regression functions, which map the input $\mathbf{z}$ to different regions in state space. We choose the following model for the regression functions

$$\mathbf{x} = \mathbf{W}^k \boldsymbol{\phi}(\mathbf{z}) + \boldsymbol{\xi}^k, \tag{4}$$

where $\boldsymbol{\xi}^k$ is a Gaussian noise vector with $\mathbf{0}$ mean and diagonal covariance matrix $\mathbf{S}^k = \text{diag}\{(\sigma_1^k)^2, \ldots, (\sigma_M^k)^2\}$. Here, $\boldsymbol{\phi}(\mathbf{z})$ is a vector of basis functions of the form $\boldsymbol{\phi}(\mathbf{z}) = [1, G(\mathbf{z}, \mathbf{z}^{(1)}), G(\mathbf{z}, \mathbf{z}^{(2)}), \ldots, G(\mathbf{z}, \mathbf{z}^{(N)})]^{\mathrm{T}}$, where G can be any function that compares two sets of image features. The weights of the basis functions are written in matrix form $\mathbf{W}^k \in \mathbb{R}^{M \times P}$ and $P = N + 1$. We use an EM type algorithm, outlined in Fig. 2, to learn the parameters $\{\mathbf{W}^k, \mathbf{S}^k\}_{k=1}^{K}$ of the mapping functions. The regression results on a toy dataset are shown in Fig. 3.

The case of ambiguous poses means that the training set contains examples that are close or the same in feature space but are far apart in state space, see Fig. 1a. When a single RVM is trained with this data, the output states tend to average different plausible poses (Agarwal and Triggs, 2004). We therefore experimentally evaluated the effect of learning mapping functions with different numbers

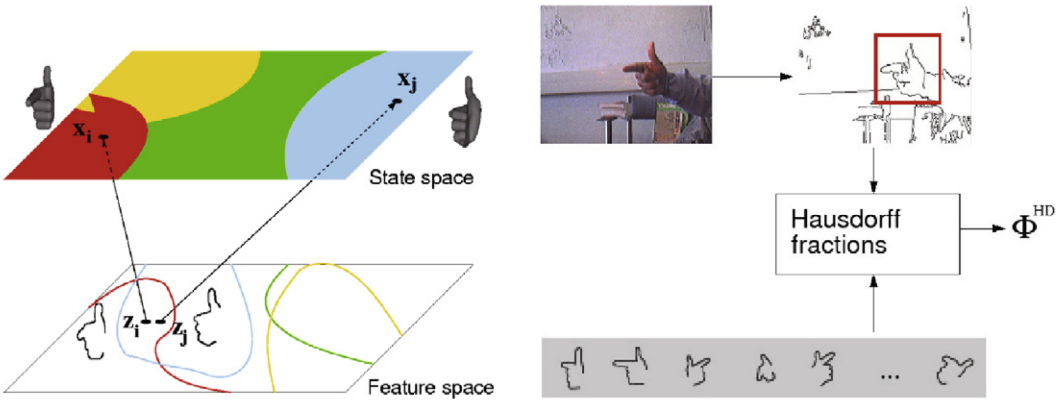

Fig. 1. (a) Multiple mapping functions. Given a single view, the mapping from image features to pose is inherently one-to-many. Mutually exclusive regions in state space can correspond to overlapping regions in feature space. This ambiguity can be resolved by learning several mapping functions from the feature space to different regions of the state space. (b) Feature extraction. The features are obtained from matching costs (Hausdorff fractions) of shape templates to the edge map. These costs are used for creating the basis function vector $\boldsymbol{\phi}^{\text{HD}}$.

1. Initialize

Partition the training set V into K subsets by applying the K-means algorithm on the state variable $\mathbf{x}_n$ of each data point v_n. Initialize probability matrix $\mathbf{C}$.

2. Iterate

(i) Estimate regression parameters

Given the matrix $\mathbf{C} \in \mathbb{R}^{N \times K}$, where element $c_{nk} = c_k^{(n)}$ is the probability that sample point n belongs to mapping function k, learn the parameters $\left\{\mathbf{W}^k, \mathbf{S}^k\right\}$ of each mapping function, by multivariate RVM regression minimizing the following cost function

$$L^k = \sum_{n=1}^{N} c_k^{(n)} \left(\mathbf{y}_k^{(n)}\right)^T \mathbf{S}^k \left(\mathbf{y}_k^{(n)}\right), \text{ where } \mathbf{y}_k^{(n)} = x^{(n)} - \mathbf{W}^k \boldsymbol{\phi}(\mathbf{z}^{(n)}). \quad (1)$$

Note: for speed up, samples with low probabilities may be ignored.

(ii) Estimate probability matrix C

Estimate the probability of each example belonging to each of the mapping function:

$$p(\mathbf{x}^{(n)}|\mathbf{z}^{(n)}, \mathbf{W}^k, \mathbf{S}^k) = \frac{1}{2\pi|\mathbf{S}|^{1/2}} \exp\left\{-0.5 \left(\mathbf{y}_k^{(n)}\right)^T \mathbf{S}^k \left(\mathbf{y}_k^{(n)}\right)\right\}, \quad (2)$$

$$c_k^{(n)} = \frac{p(\mathbf{x}^{(n)}|\mathbf{z}^{(n)}, \mathbf{W}^k, \mathbf{S}^k)}{\sum_{j=1}^{K} p(\mathbf{x}^{(n)}|\mathbf{z}^{(n)}, \mathbf{W}^j, \mathbf{S}^j)}. \quad (3)$$

Fig. 2. EM for learning multiple mapping functions $\mathbf{W}_k$.

of RVMs (with Hausdorff fractions as the input to the mapping functions). The data was generated by random sampling from a region in the four-dimensional state space of global rotation and scale, and projecting a 3D hand model into the image. The size of the training set was 7000 and the size of the test set was 5000. Different numbers of mapping functions were trained to obtain a one-to-many mapping from the features to the state space. The results are shown in Fig. 4a. Training multiple mapping functions reduces the estimation error and creates sparser template sets. Additionally, the total training time is reduced because the RVM training time increases quadratically with the number of data points and the samples are divided among the different RVMs.

3. Training an RVM with multivariate outputs

The RVM is a Bayesian regression framework, in which the weights of each input example are governed by a set of hyperparameters. These hyperparameters describe the posterior distribution of the weights and are estimated iteratively during training. Most hyperparameters approach infinity, causing the posterior distributions of the effectively setting the corresponding weights to zero. The remaining examples with non-zero weights are called *relevance vectors*. The attraction of the RVM is that it has good generalization performance, while achieving sparsity in the representation. The formulation in (Tipping, 2001) only allows regression from multivariate input to a univariate output variable. One solution is to use a single RVM for each output dimension. For example, Williams et al. used separate RVMs to track the four parameters of a similarity transform of an image region (Williams et al., 2003). This solution has the drawback that one needs to keep separate sets of selected examples for each RVM. We introduce the multivariate RVM (MVRVM), which extends the RVM framework to multivariate outputs, making it a general regression tool.[1]

The data likelihood is obtained as a function of weight variables and hyperparameters. The weight variables are then analytically integrated out to a obtain marginal likelihood as function of the hyperparameters. An optimal set of hyperparameters is obtained by maximizing the marginal likelihood over the hyperparameters using a version of the fast marginal likelihood maximization algorithm (Tipping and Faul, 2003). The optimal weight matrix is obtained using the optimal set of hyperparameters. The rest of this section details our proposed extension of the RVM framework to handle multivariate outputs and how this is used to minimize the cost function described in equation (1) and learn the parameters of a mapping function, $\mathbf{W}^k$ and $\mathbf{S}^k$. We can rewrite equation (1) in the following form

$$L^k = \sum_{n=1}^{N} \log \mathcal{N}(\hat{\mathbf{x}}_k^{(n)} | \mathbf{W}^k \widehat{\boldsymbol{\phi}}_k(\mathbf{z}^{(n)}), \mathbf{S}^k), \quad (5)$$

where

$$\hat{\mathbf{x}}_k^{(n)} = \sqrt{c_k^{(n)}}\mathbf{x}^{(n)} \quad \text{and} \quad \widehat{\boldsymbol{\phi}}_k(\mathbf{z}^{(n)}) = \sqrt{c_k^{(n)}}\boldsymbol{\phi}(\mathbf{z}^{(n)}). \quad (6)$$

[1] Code is available from http://mi.eng.cam.ac.uk/~at315/MVRVM.htm.

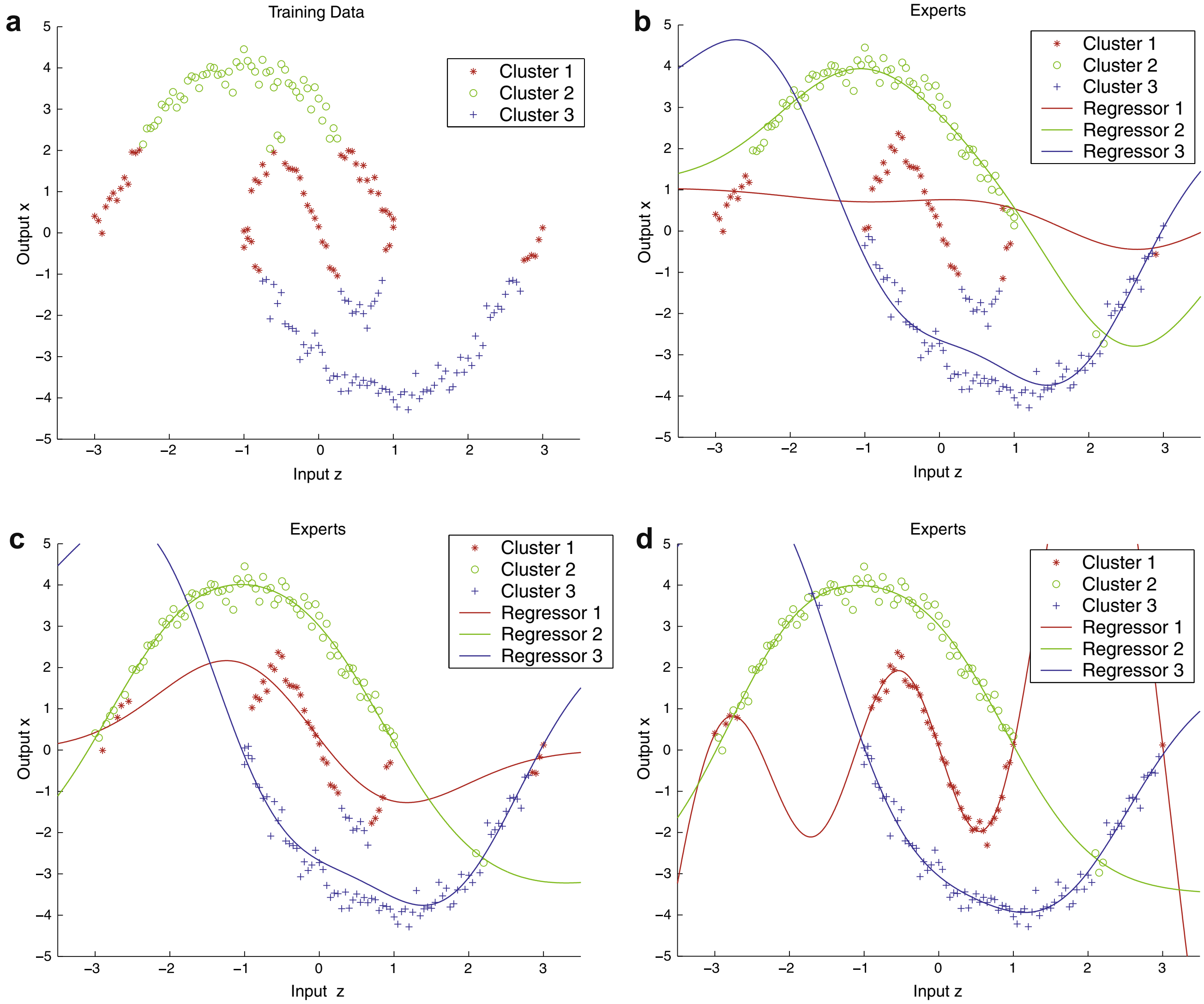

Fig. 3. RVM regression on a toy dataset. The data set consists of 200 samples from three polynomial functions with added Gaussian noise. (a) Initial clustering using *K*-means. (b–d) Learned RVM regressors after the 1st, 4th and 10th iteration, respectively. Each sample data is shown with the colour of the regressor with the highest probability. A Gaussian kernel with a kernel width of 1.0 was used to create the basis functions. Only 14 samples were retained after convergence.

We need to specify a prior on the weight matrix to avoid overfitting. We follow Tipping's relevance vector approach (Tipping, 2001) and assume a Gaussian prior for the weights of each basis function. Let $\mathbf{A} = \mathrm{diag}(\alpha_1^{-2}, \ldots, \alpha_P^{-2})$, where each element α_j is a hyperparameter that determines the *relevance* of the associated basis function. The prior distribution over the weights is then

$$p(\mathbf{W}^k|\mathbf{A}^k) = \prod_{r=1}^{M}\prod_{j=1}^{P}\mathcal{N}(w_{rj}^k|0,\alpha_j^{-2}), \tag{7}$$

where w_{rj}^k is the element at (r,j) of the weight matrix $\mathbf{W}^k$. We can now completely specify the parameters of the *kth* mapping function as $\{\mathbf{W}^k, \mathbf{S}^k, \mathbf{A}^k\}$. As the form and the learning routines of parameters of each expert are the same, we drop the index k for clarity in the rest of the section. A likelihood distribution of the weight matrix $\mathbf{W}$ can be written as

$$p(\{\hat{\mathbf{x}}^{(n)}\}_{n=1}^{N}|\mathbf{W},\mathbf{S}) = \prod_{n=1}^{N}\mathcal{N}(\hat{\mathbf{x}}^{(n)}|\mathbf{W}\widehat{\boldsymbol{\phi}}(\mathbf{z}^{(n)}),\mathbf{S}). \tag{8}$$

Let $\mathbf{w}_r$ be the weight vector for the rth component of the output vector $\mathbf{x}$, such that $\mathbf{W} = [\mathbf{w}_1, \ldots, \mathbf{w}_r, \ldots, \mathbf{w}_M]^{\mathrm{T}}$ and let $\boldsymbol{\tau}_r$ be the vector with the rth, component of all the example output vectors. Exploiting the diagonal form of $\mathbf{S}$, the likelihood can be written as a product of separate Gaussians of the weight vectors of each output dimension:

$$p(\{\hat{\mathbf{x}}^{(n)}\}_{n=1}^{N}|\mathbf{W},\mathbf{S}) = \prod_{r=1}^{M}\mathcal{N}(\boldsymbol{\tau}_r|\mathbf{w}_r\widehat{\boldsymbol{\Phi}},\sigma_r^2), \tag{9}$$

where $\widehat{\boldsymbol{\Phi}} = [\mathbf{1}, \widehat{\boldsymbol{\phi}}(\mathbf{z}_1), \widehat{\boldsymbol{\phi}}(\mathbf{z}_2), \ldots, \widehat{\boldsymbol{\phi}}(\mathbf{z}_N)]$ is the *design matrix*. The prior distribution over the weights is rewritten in the following form:

a

#RVMs	relevant templates	approx. total training time	mean RMS error
1	13.48 %	360 min	15.82
5	13.04 %	150 min	7.68
10	10.76 %	90 min	5.23
15	9.52 %	40 min	4.69
20	7.78 %	25 min	3.89

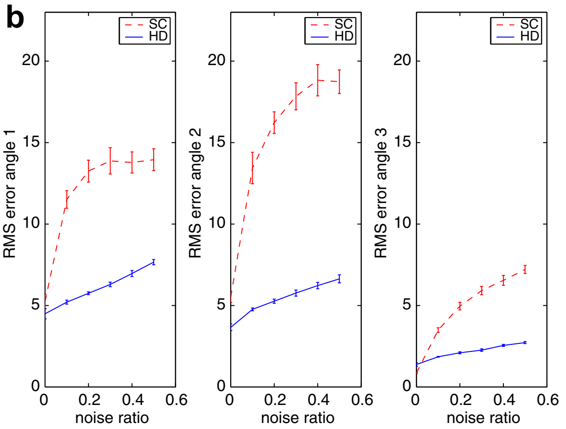

Fig. 4. (a) Single vs. multiple RVMs. Results of training different numbers of RVMs on the same dataset. Multiple RVMs learn sparser models, require less training time and yield a smaller estimation error. (b) Robustness analysis. Pose estimation error when using two different types of features: histograms of shape-contexts (SC) and Hausdorff matching costs (HD). Plotted is the mean and standard deviation of the RMS error of three estimated pose parameters as a function of image noise level. Hausdorff features are more robust to edge noise.

$$p(\mathbf{W}|\mathbf{A}) = \prod_{r=1}^{M}\prod_{j=1}^{P}\mathcal{N}(w_{rj}|0,\alpha_j^{-2}) = \prod_{r=1}^{M}\mathcal{N}(\mathbf{w}_r|\mathbf{0},\mathbf{A}). \tag{10}$$

Now the posterior on $\mathbf{W}$ can be written as the product of separate Gaussians for the weight vectors of each output dimension:

$$p(\mathbf{W}|\{\hat{\mathbf{x}}\}_{n=1}^{N},\mathbf{S},\mathbf{A}) \propto p(\{\hat{\mathbf{x}}\}_{n=1}^{N}|\mathbf{W},\mathbf{S})p(\mathbf{W}|\mathbf{A}). \tag{11}$$

$$\propto \prod_{r=1}^{M}\mathcal{N}(\mathbf{w}_r|\boldsymbol{\mu}_r,\boldsymbol{\Sigma}_r), \tag{12}$$

where $\boldsymbol{\mu}_r = \sigma_r^{-2}\boldsymbol{\Sigma}_r\boldsymbol{\Phi}^{\mathrm{T}}\boldsymbol{\tau}_r$ and $\boldsymbol{\Sigma}_r = (\sigma_r^{-2}\boldsymbol{\Phi}^{\mathrm{T}}\boldsymbol{\Phi}+\mathbf{A})^{-1}$ are the mean and the covariance of the distribution of $\mathbf{w}_r$. Given the posterior for the weights, we can choose an optimal weight matrix if we obtain a set of hyperparameters that maximise the data likelihood in Eq. (12). The Gaussian form of the distribution allows us to the remove the weight variables by analytically integrating them out. Exploiting the diagonal form of $\mathbf{S}$ and $\mathbf{A}$ once more, we marginalize the data likelihood over the weights:

$$p(\{\hat{\mathbf{x}}\}_{n=1}^{N}|\mathbf{A},\mathbf{S}) = \int p(\{\hat{\mathbf{x}}\}_{n=1}^{N}|\mathbf{W},\mathbf{S})p(\mathbf{W}|\mathbf{A})\mathrm{d}\mathbf{W}. \tag{13}$$

$$= \prod_{r=1}^{M}\int \mathcal{N}(\boldsymbol{\tau}_r|\mathbf{w}_r\widehat{\boldsymbol{\Phi}},\sigma_r^2)\mathcal{N}(\mathbf{w}_r|\mathbf{0},\mathbf{A}). \tag{14}$$

$$= \prod_{r=1}^{M}|\mathbf{H}_r|^{-\frac{1}{2}}\exp\left(-\frac{1}{2}\boldsymbol{\tau}_r^{\mathrm{T}}\mathbf{H}_r^{-1}\boldsymbol{\tau}_r\right), \tag{15}$$

where $\mathbf{H}_r = \sigma_r^2\mathbf{I}+\widehat{\boldsymbol{\Phi}}\mathbf{A}^{-1}\widehat{\boldsymbol{\Phi}}^{\mathrm{T}}$. An optimal set of hyperparameters $\{\alpha_j^{\mathrm{opt}}\}_{j=1}^{P}$ and noise parameters $\{\sigma_r^{\mathrm{opt}}\}_{r=1}^{M}$ is obtained by maximising the marginal likelihood using bottom-up basis function selection as described by Tipping et al. in Tipping and Faul (2003). Again, the method was extended to handle the multivariate outputs. Details of this extension can be found in (Thayananthan, 2005). The optimal hyperparameters are then used to obtain the optimal weight matrix:

$$\begin{aligned}\mathbf{A}^{\mathrm{opt}} &= \mathrm{diag}(\alpha_1^{\mathrm{opt}},\ldots,\alpha_P^{\mathrm{opt}})\boldsymbol{\Sigma}_r^{\mathrm{opt}}\\ &= ((\sigma_r^{\mathrm{opt}})^{-2}\widehat{\boldsymbol{\Phi}}^{\mathrm{T}}\widehat{\boldsymbol{\Phi}}+\mathbf{A}^{\mathrm{opt}})^{-1}.\end{aligned} \tag{16}$$

$$\boldsymbol{\mu}_r^{\mathrm{opt}} = (\sigma_r^{\mathrm{opt}})^{-2}\boldsymbol{\Sigma}_r^{\mathrm{opt}}\boldsymbol{\Phi}^{\mathrm{T}}\boldsymbol{\tau}_r,\mathbf{W}^{\mathrm{opt}} = [\boldsymbol{\mu}_1^{\mathrm{opt}},\ldots,\boldsymbol{\mu}_M^{\mathrm{opt}}]^{\mathrm{T}}. \tag{17}$$

We performed experiments comparing the feature robustness in the presence of noise. This was done for features G^{HD} based on the Hausdorff distance and features based on 100-dimensional shape-context histograms G^{SC} (Agarwal and Triggs, 2006). A set of images is created by sampling a region in state space, in this case three rotation angles over a limited range, and using the sampled pose vectors to project a 3D hand model into the image. Because the Hausdorff features are neither translation nor scale invariant, additional training images of scaled and locally shifted examples are generated. After RVM training, a set of around 30 templates out of 200 are chosen for both, shape-context and Hausdorff features. However, note that the templates chosen by the RVM for each methods may differ. For testing, 200 poses are generated by randomly sampling the same region in parameter space and introducing different amounts of noise by introducing edges of varying length and curvature. Fig. 4b shows the dependency of the RMS estimation error (mean and standard deviation) on the noise level. Hausdorff features are more robust to edge noise than shape-context features.

4. Pose estimation and tracking

Given a candidate object location in the image we obtain K possible poses from the mapping functions, see Fig. 6a. For each mapping function $\mathbf{W}_k$ the templates selected by the RVM are matched to the input and the resulting Hausdorff fractions (Huttenlocher et al., 1993) form the basis function vector $\boldsymbol{\phi}^{\mathrm{HD}}$. We then use regression to obtain K pose estimates via $\mathbf{x}_k = \mathbf{W}^k\boldsymbol{\phi}^{\mathrm{HD}}$. A set of candidate object locations is obtained by skin colour detection for hands and background estimation for full human body motion. Given M candidate positions we thus obtain $K \times M$ pose hypotheses, which are used to project the 3D object model into the image and obtain image likelihoods.

The observation model for the likelihood computation is based on edge and silhouette cues. As a likelihood model for hand tracking we use the function proposed in (Stenger et al., 2006), which combines chamfer matching with foreground silhouette matching, where the foreground is found by skin colour segmentation. The same likelihood function is used in the full body tracking experiments, with the difference that in this case the foreground silhouette is estimated by background subtraction.

The question arises as to whether it is necessary to use the more computationally expensive model projection process for the likelihood evaluation. Being based on Gaussian regression models, RVMs do provide likelihood estimates. However, we observed that in some cases the RVM variances are too low. In order to demonstrate this we generated silhouette data from two different motions in the CMU mocap database (Carnegie-Mellon MoCap Database, xxxx), one running (173 frames) and one exercise motion sequence (4591 frames). The 62-dimensional state vector was first reduced to six-dimensions via PCA. The RVM selected 16 training samples. When estimating the motion of the unseen exercise sequence, one expects a high uncertainty, thus large σ values for the RVM predictions, as shown in Fig. 5. However, in some cases the σ values are low even though the pose is far from any pose in the training set. In other words, the RVM can be overconfident, particularly in areas of the state space that had little or no representation in the training data. Therefore, in contrast to (Agarwal and Triggs, 2006) our method uses the RVM predictions only as hypotheses for a likelihood evaluation based on projecting a 3D body model into the image. It can therefore be seen as a hybrid of learning based and model based approaches.

Temporal information is needed to resolve the ambiguous poses and to obtain a smooth trajectory through the state-space after the pose estimation is done at every frame. We embed pose estimation with multiple RVMs within a probabilistic tracking framework, which involves representing and maintaining distributions of the state $\mathbf{x}$ over time.

The distributions are represented using a piecewise Gaussian model (Cham and Rehg, 1999) with L components. The evaluation of the distribution at one time instant t involves the following steps (see Fig. 6b):

(1) Predict each of the L components.
(2) Perform RVM regression to obtain K hypotheses.
(3) Evaluate likelihood computation for each hypothesis.
(4) Compute the posterior distribution for each of $L \times K$ components.
(5) Select L components to propagate to next time step.

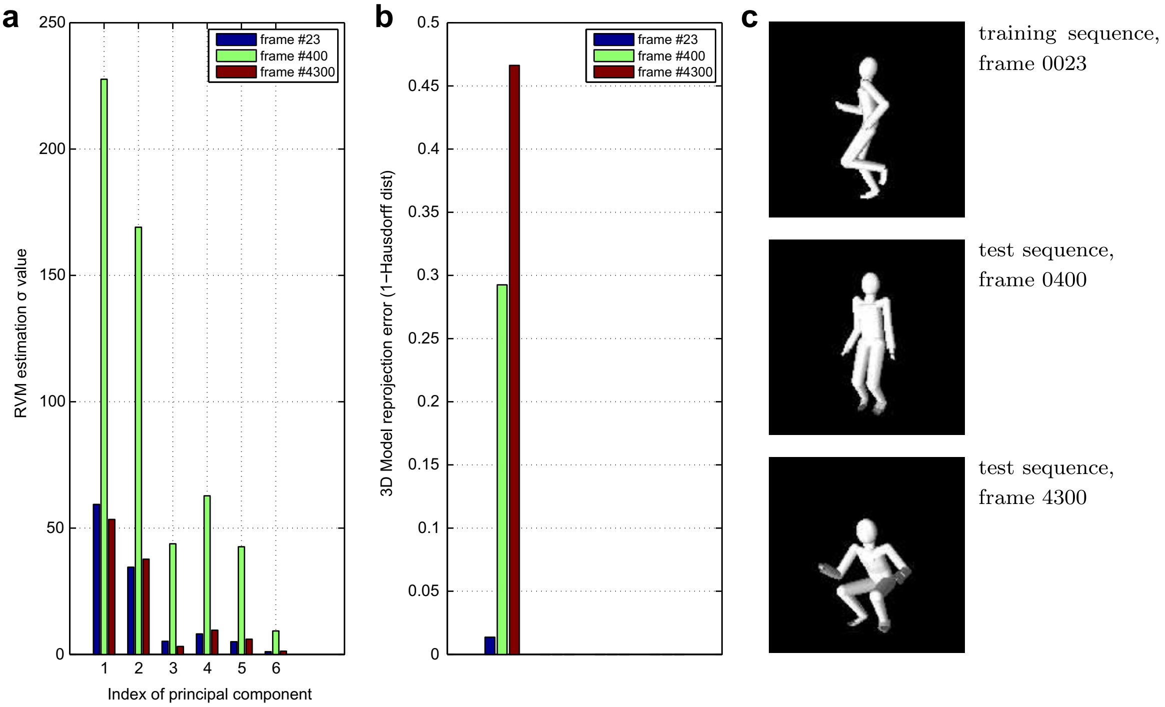

Fig. 5. Overconfidence of RVM estimates. This figure shows the variance values (a) for RVM predictions for three inputs (c): one of the training sequence and two of a test sequence with previously unseen motion. We expect low uncertainty, i.e., small σ values for the training input, but large uncertainty for the unseen inputs (as for frame 400). However, in some cases the RVM is overconfident, see values for frame 4300, which are nearly identical to the values for the training input. On the other hand, the likelihood computed from the reprojection error (b) results in values that are very different from the training input.

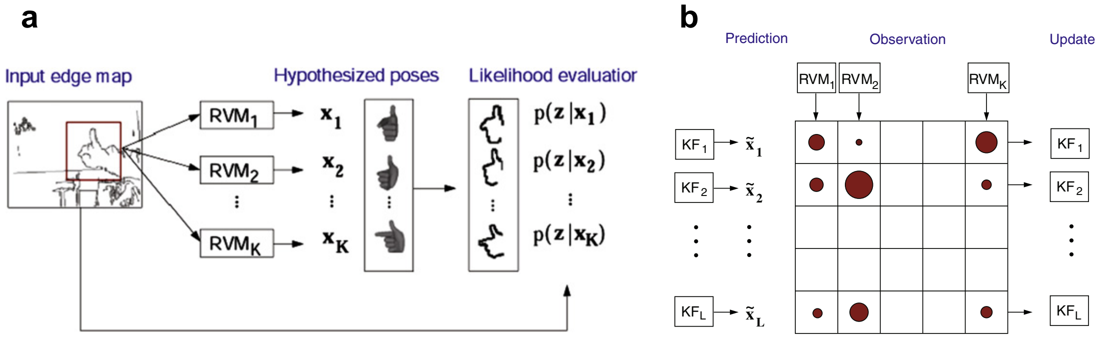

Fig. 6. (a) Pose estimation. At each candidate location the features are obtained by Hausdorff matching and the RVMs yield pose estimates. These are used to project the 3D model and evaluate likelihoods. (b) Probabilistic tracking. The modes of likelihood distribution, obtained through the RVM mapping functions, are propagated through a bank of Kalman filters (Cham and Rehg, 1999). The posterior distributions are represented with an L-mode piecewise Gaussian model. At each frame, the L Kalman filter predictions and K RVM observations are combined to generate possible $L \times K$ Gaussian distributions. Out of these, L Gaussians are chosen to represent the posterior probability and propagated to the next level. The circles in the figure represent the covariance of Gaussians.

The dynamics are modelled using a constant velocity model with large process noise (Cham and Rehg, 1999), where the noise variance is set to the variance of the mapping error estimated at the RVM learning stage. At step (5) k-means clustering is used to identify the main components of the posterior distribution in the state space, similar to (Vermaak et al., 2003). Components with the largest posterior probability are chosen from each cluster in turn, ensuring that not all components represent only one region of the state-space.

For a given frame the correct pose does not always have the largest posterior probability. Additionally, the uncertainty of pose estimation is larger in some regions in state space than in others, and a certain number of frames may be needed before the pose ambiguity can be resolved. The largest peak of the posterior fluctuates among different trajectories as the distribution is propagated. Hence a history of the peaks of the posterior probability needs to be considered before a consistent trajectory is found that links the peaks over time. In our experiments a batch Viterbi algorithm is used to find such a path (Navaratnam et al., 2005).

5. Experimental results

The RVM tracking framework is applied to the problem of tracking 3D global and articulated motion of the hand and the whole human body. Throughout the experiments a Gaussian kernel with a standard deviation of 0.5 is used, a value that was found in initial experiments. The number of principal components was determined for each data set separately and was set to the minimum number of components that result in a reprojection error of less than 10%.

5.1. Full body articulation

In order to track full body motion, we use a data set from the CMU motion capture database of walking persons (∼9000 data points). In order to reduce the RVM training time, the data is projected onto the first six principal components.

The first input sequence is a person walking fronto parallel to the camera. The global motion is mainly limited to translation. The eight-dimensional state-space is defined by two global and six articulation parameters. A set of 13,000 training samples were created by sampling the region. We use 4 RVM mapping functions to approximate the one-to-many mapping. A set of 118 relevant templates is retained after training. Background subtraction is used to remove some of the background edges. The tracking results are shown in Fig. 7. The second input sequence is a video of a person walking in a circle from (Sidenbladh et al., 2000). The range of global motion is set to 360° around axis normal to the ground plane and 20° in the tilt angle. The range of scales is 0.3–0.7. The nine-dimensional state-space region is defined by these three global and six articulation parameters. A set of 50,000 templates is generated by sampling this region. We use 50 RVM mapping functions to approximate the one-to-many mapping. A set of 984 relevant templates is retained after training. Background subtraction is used to remove some of the background edges. The tracking results are shown in Fig. 8.

5.2. Hand articulation

In this experiment we estimate the rigid body parameters as well as a lower-dimensional representation of the articulation parameters of an opening and closing hand. The method is applied to the hand sequence containing 88 frames from (Stenger et al., 2006), where approximately 30,000 templates were required for tracking. To capture typical hand motion data, we use a large set of 10-dimensional joint angle data obtained from a data glove. The pose data was approximated by the first four principal components. We then projected original hand glove data into those four-dimensions. The global motion of the hand

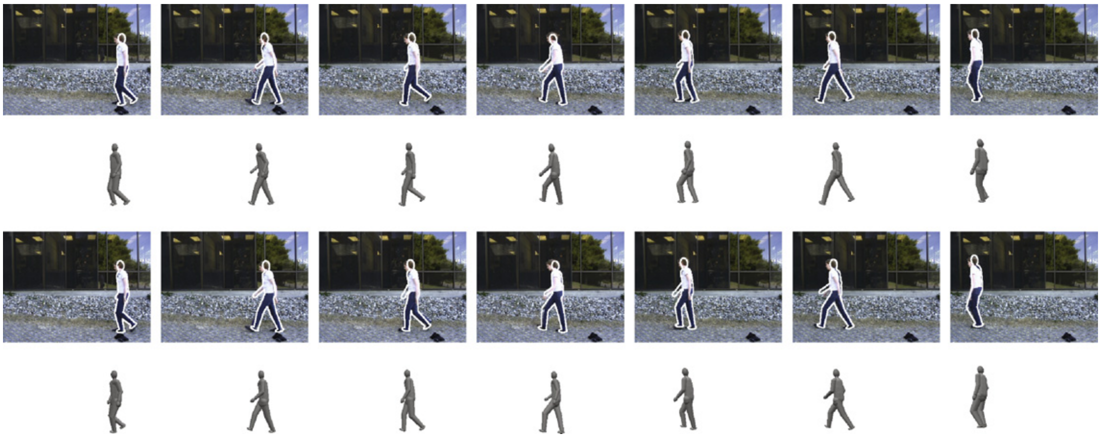

Fig. 7. Tracking a person walking fronto parallel to the camera. The first and second rows shows the frames from (Sidenbladh et al., 2000), overlaid with the body pose corresponding to the optimal path through the posterior distribution and the corresponding the 3D model, respectively. Similarly, second and third rows show the second best path. Notice that the second path describes the walk equally well except for the right–left leg flip which is one of the common ambiguity that arises in human pose estimation from monocular view. A total of 118 templates with 4 RVM mapping functions were used.

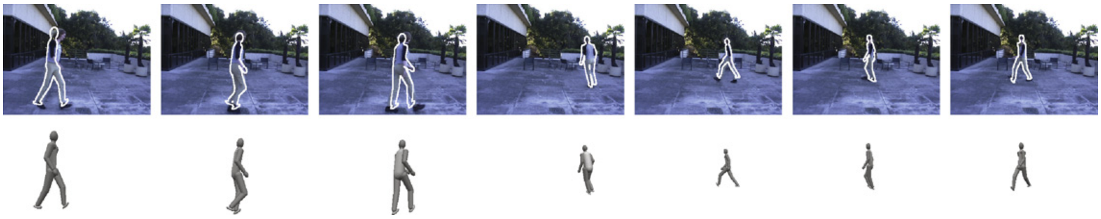

Fig. 8. Tracking a person walking in a circle. This figure shows the results of the tracking algorithm on a sequence from (Sidenbladh et al., 2000). Overlaid is the body pose corresponding to the optimal path through the posterior distribution, the 3D model is shown below. A total of 1429 templates with 50 RVM mapping functions were used.

in that sequence was limited to a certain region of the global space (80°, 60° and 40° in rotation angles and 0.6 to 0.8 in scale). The eight-dimensional state space is defined by the four global and four articulation parameters. A set of 10,000 templates is generated by random sampling in this state space. After training 10 RVMs, 455 templates out of 10,000 are retained. Due to the large amount of background clutter in the sequence, skin colour detection is used in this sequence to remove some of the background edges for this sequence. Tracking results are shown in Fig. 9.

5.3. Computation time

The execution time in the experiments varies from 5 to 20 s per frame (on a Pentium IV, 2.1 GHz PC), depending on the number of candidate locations in each frame. The computational bottleneck is the model projection in order to compute the likelihoods (approximately 100 per s). For example, for 30 search locations and 50 RVM mapping functions result in 1500 model projections, requiring 15 seconds. It can be observed that most mapping functions do not yield high likelihoods, thus identifying them early will help to reduce the computation time.

6. Summary

This paper has introduced a framework for single camera pose estimation and tracking that is a hybrid of learning based and generative model based approaches. To this end we have developed a multivariate generalization of

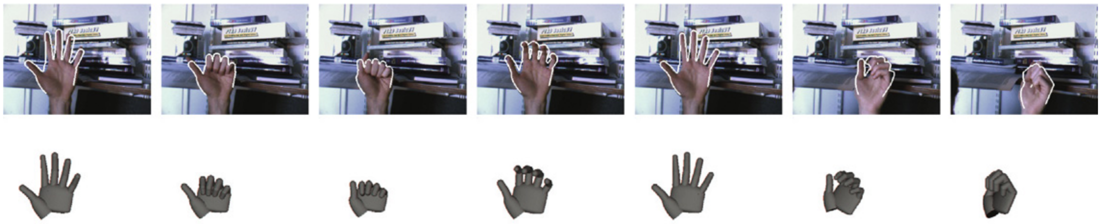

Fig. 9. Tracking an opening and closing hand. This sequence shows tracking of opening and closing hand motion together with global motion on a sequence from (Stenger et al., 2006). A total of 537 relevant templates were used with 20 RVM mapping functions for pose estimation. As a comparison (Stenger et al., 2006) used about 30,000 templates to track the same sequence.

Tipping and Faul's Tipping and Faul (2003) bottom-up method for learning a sparse RVM regressor. The method has been used as a component of a system for tracking and estimating 3D human pose from monocular image sequences. We have combined several techniques to solve this problem: (1) multivalued regression based on Gaussian mixtures to allow for multiple solutions, (2) multivariate RVM for the individual regressors, (3) reprojection of a 3D model to evaluate a posteriori probabilities for the resulting 3D hypothesis, and (4) global optimization using dynamic programming to find 3D trajectories through the resulting sets of static 3D pose hypotheses.

Appendix A. Supplementary material

Supplementary data associated with this article can be found, in the online version, at doi:10.1016/j.patrec.2008.02.004.

References

Agarwal, A., Triggs, B., 2004. 3D human pose from silhouettes by relevance vector regression. In: Proc. Conf. on Computer Vision and Pattern Recognition, vol. II. Washington, DC, July, pp. 882–888.

Agarwal, A., Triggs, B., 2004. Learning to track 3D human motion from silhouettes. In: Proc. 21st Internat. Conf. on Machine Learning, Banff, Canada, pp. 9–16.

Agarwal, A., Triggs, B., 2005. Monocular human motion capture with a mixture of regressors. In: IEEE Workshop on Vision for Human Computer Interaction.

Agarwal, A., Triggs, B., 2006. Recovering 3D human pose from monocular images. Trans. PAMI 28 (1), 44–58.

Brand, M., 1999. Shadow puppetry. In: Proc. 7th Internat. Conf. on Computer Vision, vol. II. Corfu, Greece, September, pp. 1237–1244.

Carnegie-Mellon MoCap Database. <http://mocap.cs.cmu.edu>.

Cham, T.J., Rehg, J.M., 1999. A multiple hypothesis approach to figure tracking. In: Proc. Conf. Computer Vision and Pattern Recognition, vol. II. Fort Collins, CO, June, pp. 239–245.

Felzenszwalb, P., Huttenlocher, D., 2005. Pictorial structures for object recognition. Int. J. Comput. Vision 61 (1).

Howe, N.R., Leventon, M.E., Freeman, W.T., 1999. Bayesian reconstruction of 3D human motion from single-camera video. In: Advanced Neural Information Processing Systems, Denver, CO, November, pp. 820–826.

Huttenlocher, D.P., Noh, J.J., Rucklidge, W.J., 1993. Tracking non-rigid objects in complex scenes. In: Proc. 4th Internat. Conf. on Computer Vision, Berlin, May, pp. 93–101.

Jordan, M., Jacobs, R., 1994. Hierarchical mixtures of experts and the EM algorithm. Neural Comput. 6, 181–214.

Lee, M.W., Cohen, I., 2004. Human upper body pose estimation in static images. In: Proc. 8th European Conf. on Computer Vision, vol. II. pp. 126–138.

Navaratnam, R., Thayananthan, A., Torr, P.H.S., Cipolla, R., 2005. Hierarchical part-based human body pose estimation. In: Proc. British Machine Vision Conference, London, UK.

Rosales, R., Athitsos, V., Sigal, L., Scarloff. S., 2001. 3D hand pose reconstruction using specialized mappings. In: Proc. 8th Internat. Conf. on Computer Vision, vol. I. Vancouver, Canada, July, pp. 378–385.

Shakhnarovich, G., Viola, P., Darrell, T., 2003. Fast pose estimation with parameter-sensitive hashing. In: Proc. 9th Internat. Conf. on Computer Vision, vol. II. pp. 750–757.

Sidenbladh, H., De la Torre, F., Black, M.J., 2000. A framework for modeling the appearance of 3D articulated figures. In: IEEE Internat. Conf. on Automatic Face and Gesture Recognition, Grenoble, France, pp. 368–375.

Sminchisescu, C., Kanaujia, A., Li, Z., Metaxas, D., 2005. Discriminative density propagation for 3D human motion estimation. In: Proc. Conf. Computer Vision and Pattern Recognition, pp. 217–323.

Stenger, B., Thayananthan, A., Torr, P.H.S., Cipolla, R., 2006. Model-based hand tracking using a hierarchical bayesian filter. Trans. PAMI 28 (9), 1372–1384.

Thayananthan, A., 2005. Template-based pose estimation and tracking of 3D hand motion. Ph.D. Thesis, University of Cambridge, UK.

Tipping, M.E., 2001. Sparse Bayesian learning and the relevance vector machine. J. Mach. Learn. Res., 211–244.

Tipping, M.E., Faul., A., 2003. Fast marginal likelihood maximisation for sparse bayesian models. In: Proc. 9th Internat. Workshop on Artificial Intelligence and Statistics, Key West, FL, January.

Vermaak, J., Doucet, A., Pérez, P., 2003. Maintaining multi-modality through mixture tracking. In: Proc. 9th Internat. Conf. on Computer Vision.

Williams, O. Blake, A., Cipolla, R., 2003. A sparse probabilistic learning algorithm for real-time tracking. In: Proc. 9th Internat. Conf. on Computer Vision, vol. I. Nice, France, October. pp. 353–360.