Regression-Based Human Motion Capture From Voxel Data

Y. Sun[2] M. Bray[1] A. Thayananthan[3] B. Yuan[2] P.H.S. Torr[1]

[1]Oxford Brookes Univeristy, Department of Computing, Oxford, OX33 1HX, UK

[2]Beijing Jiaotong University, Institute of Information Science, Beijing, 100044, PRC

[3]University of Cambridge, Department of Engineering, Cambridge, CB2 1PZ, UK

Abstract

A regression based method is proposed to recover human body pose from 3D voxel data. In order to do this we need to convert the voxel data into a feature vector. This is done using a Bayesian approach based on Mixture of Probabilistic PCA that transforms a collection of 3D shape context descriptors, extracted from the voxels, to a compact feature vector. For the regression, the newly-proposed Multi-Variate Relevance Vector Machine is explored to learn a single mapping from this feature vector to a low-dimensional representation of full body pose. We demonstrate the effectiveness and robustness of our method with experiments on both synthetic data and real sequences.

1 Introduction

Human motion capture (MOCAP) is of interest to the academic and industrial communities due to its various applications ranging from film and game production to medical analysis. Marker based motion capture has been performed with good results using special equipment such as strobing cameras, reflective markers and user intervention. For the last 10 years research has focused on markerless MOCAP to obviate such constraints. The arsenal of methods that have been proposed for this purpose can be divided into two categories: model-based and non-model based approaches.

The model-based (or generative) approaches [10, 12, 19, 17] are usually expressed within the analysis-by-synthesis paradigm. An explicit model is usually designed which is similar to the target (observation) and an error measure between these two is defined and then minimized at each frame. Model-based methods are well-known to be accurate but computationally expensive. They require a good initialization (mostly manual) and few of them can recover from tracking failures (e.g. caused by local minima). The non-model based approaches can be further classified into parametric and non-parametric subcategories, or more intuitively, regression-based and examplar-based. While examplar-based methods [9, 15] store a set of training examples with corresponding known poses and search for the ones similar given a new input, regression-based methods [1, 2, 22, 13, 4, 8, 14, 18, 3, 7] learn a compact mapping from observable image quantities to human pose space.

Previous regression-based methods mainly work on monocular images. Agarwal et al. [1] recover the body pose by nonlinear regression using image feature extracted from monocular silhouette, which is encoded by a histogram of 2D shape context vectors. They

BMVC 2006 doi:10.5244/C.20.29

compare regularized least squares and RVM regressors over both linear and kernel bases. It turns out that the combination of RVM and kernel bases provides the best performance. A learned autoregressive dynamical model is further incorporated into kernel function to smooth temporal jitter [2]. Tian et al. [22] estimate upper body pose from a single image by optimizing an objective function derived from Gaussian Process Latent Variable Model. Rosales et al. [13] map image feature to 2D joint locations using multi-layer perceptron (MLP). Other methods [4, 8] are also designed for use with a single camera. Pose recovery in monocular images suffer greatly from substantial loss in depth information and the resulting ambiguities. As individual projected image may correspond to numerous body poses, the estimated result may average or zig-zag among many possible solutions [1]. Although some researchers learn mixture of regressors to alleviate this problem [14, 18, 3], there should be no doubt that the most effective way is to use more views. Indeed for a commercial system motion capture system it might be argued that the research should really be focused on optimal multi-camera methods.

How should we deal with multiple camera inputs optimally? One may simply concatenate the image features from all views into one big feature vector to perform regression, a possible extension to Agarwal's framework. Grauman et al. [7] use Mixture of Probabilistic PCA (MPPCA) to model the manifold of big feature vector, which is formed by concatenating silhouette points from multiple views and 3D structure parameters. New multi-view contours are projected into probabilistic linear subspaces to reconstruct unknown structure parameters. However, a common inconvenience of the above solutions is that the regressor or manifold must be re-learned each time the camera setup is changed, which is highly time consuming.

If geometrical calibration as well as multiple views are available, we can obtain voxel data using shape from silhouettes [5]. Such 3D representation of human body combines the information from each view in an unbiased way, and is basically independent of camera setup given enough viewpoints. Therefore, to avoid relearning the regressor often, we propose an approach in this paper to estimate the pose from voxel data. Our method belongs to regression-based sub-category and then differs from other model-based methods which fit a body model to the voxel data (e.g. [10, 12]).

Given voxel data reconstructed from multiple views, the pipeline of our method is shown in Figure 1, in which main contributions to the literature are as follows:

- It is the first method that learns human body pose from voxel data.
- 3D Shape Context (3DSC) is improved to better describe voxel data.
- 3DSC vectors extracted from the voxel data are transformed in a Bayesian way into a compact feature vector, which acts as input to our regressor.
- The newly-proposed Multi-Variate Relevance Vector Machine (MVRVM) is explored to learn a single mapping from feature space to a low-dimensional manifold of full body pose space.

This paper is laid out as follows: How to parameterize pose space is explained in Section 2. Then, Section 3 describes MVRVM, a Bayesian non-linear regression algorithm. Improved 3DSC is presented in Section 4. A Bayesian approach based on MPPCA is introduced in Section 5 to construct the histogram of 3DSC vectors. Section 6 reports

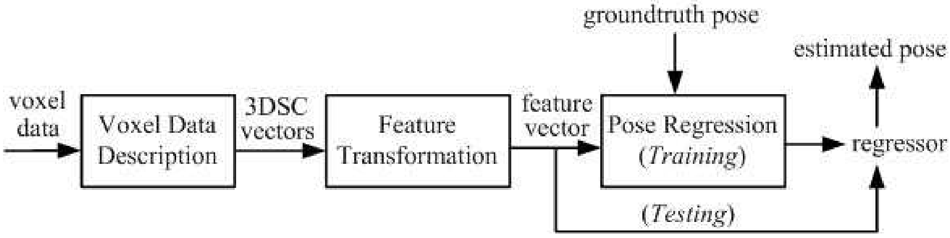

Figure 1: *The pipeline of our regression-based MOCAP method from voxel data. Voxel data are described as a collection of 3DSC vectors and then transformed to a feature vector. During the training stage, a regressor which maps the feature vector to groundtruth pose is learned. Given a testing input, its pose is estimated by feeding the regressor with the feature vector. Here pose is actually a low-dimensional representation of full body pose as explained in Section 2.*

quantitative comparison with a relevant reference [1] on synthetic data, along with qualitative results on real sequence to show the robustness of our method, followed by the conclusion in Section 7.

2 Pose Space Parameterization

We represent the human body by an implicit kinematic model where a possible combination of joint angles corresponds to a particular body pose. The number of degrees of freedom of our full body model is $m' = 42$ including 3 joint angles for each of the 14 major body joints (torso centre, neck, and two shoulders, elbows, wrists, hips, knees, ankles), which correspond to the marker-based MOCAP data that we use as ground truth. Thus, the pose is parameterized as a high dimensional state vector $y' \in \mathbb{R}_{m'}$, a redundant representation. We seek a mapping from y' to y in a low-dimensional manifold of full body pose space, and PCA is a convenient choice. It projects data onto an orthogonal M dimensional linear subspace, and reconstructs y' faster and more simply than non-linear methods. M is automatically set by constraining the reconstruction error caused by dimensionality reduction to an allowable level, less than 10% for instance.

$$M = \underset{1<M<m'}{argmin}\left(\frac{\sum_{i=1}^{M}\lambda_i}{\sum_{i=1}^{m'}\lambda_i} > 90\%\right) \tag{1}$$

where λ_i are the eigenvalues of the covariance matrix sorted in decreasing order.

3 Pose Regression Using Multi-Variate RVM

Given a set of training examples $\mathscr{V} = \{v^{(n)}\}_{n=1}^{N}$ consisting of pairs $v^{(n)} = \{(\mathbf{y}^{(n)}, \mathbf{x}^{(n)})\}$ of state vectors and feature vectors ($\vec{x}$ is described in Section 5), we want to learn a mapping from feature space to state space $\mathbb{R}_M$ using a Gaussian regression model:

$$\mathbf{y} = \mathbf{W}\phi(\mathbf{x}) + \xi, \tag{2}$$

where ξ is gaussian noise vector with $\mathbf{0}$ mean and diagonal covariance matrix. $\phi(\mathbf{x})$ is the vector of *data-centric* basis functions of the form

$$\phi(\mathbf{x}) = [1, G(\mathbf{x}, \mathbf{x}^{(1)}), G(\mathbf{x}, \mathbf{x}^{(2)}), ..., G(\mathbf{x}, \mathbf{x}^{(N)})]^T \tag{3}$$

where G can be any kernel function. In this work, we found that the use of Gaussian kernels provides robust results. During the learning stage, the weight matrix $\mathbf{W}$ is learned using an extension [21] of the RVM regression algorithm [23]. The attraction of the RVM comes from its good generalization performance, while achieving sparsity in the regressor. In our case this means that $\mathbf{W}$ have many zero columns, hence only a fraction of the total number of training examples with non-zero weights need to be stored.

Tipping's formulation in [23] only allows regression from multivariate input to a univariate output variable. One solution is to use a single RVM for each output dimension. It has the drawback that one needs to keep separate sets of selected examples for each RVM. The RVM framework has been recently extended to multivariate outputs [21], making it a general multivariate regression tool. In our case, this formulation allows us to choose the same subset of training examples for all output dimensions.

4 Voxel Data Description

Voxel data can be acquired using silhouettes from multiple cameras [5]. Although silhouette extraction is not within the scope of discussion in this work, satisfying results can be acquired by chroma-keying or simple background subtraction in controlled environments, or using more advanced techniques such as adaptive background model [20] or graph cut [16] in cluttered scenes.

Considering both accuracy and complexity of voxel data, the voxel size is set according to an empirical formula $l_{res} = \frac{h_{sub}}{60}$, where the height of the subject h_{sub} can be easily measured after processing the first frame using a moderate resolution (e.g. $l_{res} = 0.05m$). This parameter is made adaptive in order to acquire voxel data of basically the same scale for subjects of different sizes and avoid shape distortion.

For regression, we need a more compact representation of shape rather than the raw voxel data. In this section, we consider the conversion of 3D objects to canonical descriptors. We require the descriptor to be distinctive and noise-insensitive as well as translation and scaling invariant. As argued in [6], many global descriptors have difficulties identi-

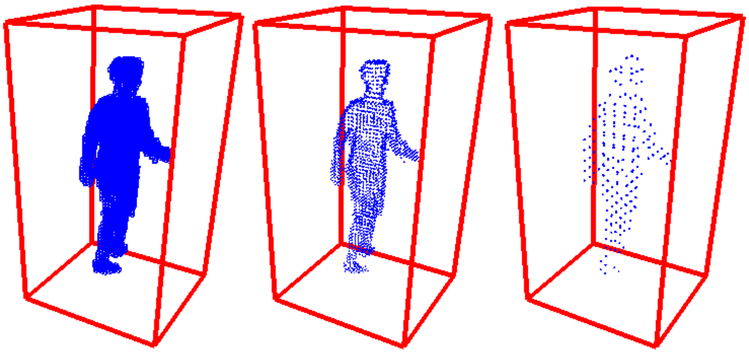

Figure 2: **Left**: *An example of voxel data (about 4000 voxels);* **Middle**: *Surface voxels (about 2000) capture accurately the 3D shape;* **Right**: *For efficiency we only calculate 3DSC vectors at basis voxels (about 200) obtained by down-sampling surface voxels. Basis voxel is the center of spherical support volume of 3DSC.*

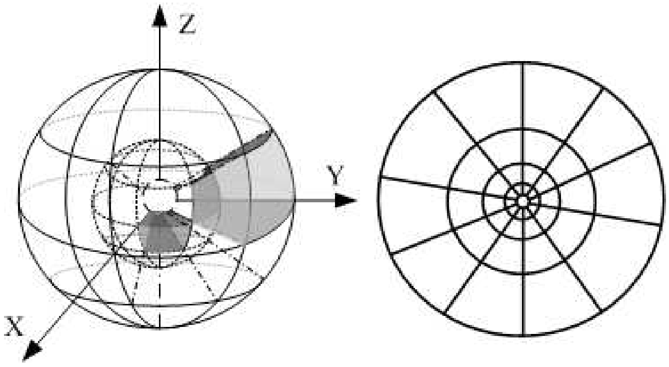

Figure 3: *Support volume of 3DSC. The local coordinate frame is aligned to world coordinate frame.* **Left:** *A toy example with 2 radial divisions, 4 elevation divisions and 8 azimuth divisions. 2 bins out of 64 are highlighted.* **Right:** *Cross section along X-Y, X-Z or Y-Z plane of the 3DSC in this work.*

fying subtle shape variations, and purely local descriptors such as surface normal are unstable when dealing with noisy data. We improve the 3D Shape Context (3DSC), a regional descriptor that lies midway between the two, to describe the property of the subject in local support volume.

3DSC was first introduced by Kortgen et al. [11]. At a point p_i, they calculate 3DSC vector as the distribution of relative positions of the remaining points in a spherical support volume, where p_i is the center of sphere. The sphere is split into combined shell-sector bins, and the distribution is essentially a histogram constructed by counting the number of points falling inside each bin. Frome et al. [6] add one more division in the azimuth dimension. In both work, 3DSC are rotation invariant which is not desirable in our case as we want to differentiate similar poses with different global orientation of the body, and noise-sensitive hard voting was adopted.

For efficiency, we only calculate 3DSC vectors at basis voxels, which are obtained by down-sampling surface voxels (i.e. voxels with at least one empty 6-neighborhood). See Figure 2. Similar to [6], our 3DSC has also radial, elevation and azimuth divisions (c.f. Figure 3). If R_{max} and R_{min} are the maximum and minimum radius of support volume, where $R_{max} = \frac{h_{sub}}{3}$ is chosen by cross-validation (see section 6) and R_{min} is set to $2 * l_{res}$, N_r is the number of radial divisions, then the logarithmically spaced radial boundaries are:

$$R_i = exp\{ln(R_{min}) + \frac{i}{N_r} ln(\frac{R_{max}}{R_{min}})\} \tag{4}$$

N_e elevation divisions and N_a azimuth divisions are evenly spaced along 180° and 360° ranges respectively. There are $N_r * N_e * N_a$ bins in total, and we experimentally choose $4 \times 5 \times 10 = 200$ bins in this work. As R_{max} and R_{min} are adaptive to subjects of different sizes, our 3DSC is scaling invariant and intrinsically translation invariant. Besides this, it has 3 new features:

- Instead of aligning the local coordinate frame of 3DSC to surface normal, we align it to the world coordinate frame. It can be noticed that this alignment makes our 3DSC rotation-dependent, because the identification of rotation around each axis is desirable.

- To deal with noise, soft voting substitutes hard voting when the histogram is constructed. If a voxel lies in the vicinity of any boundary, it gives divided vote to the bins on both sides. Further, all votes are weighted by $w_i = 1/\sqrt[3]{V_i}$ where V_i is the volume of a particular bin.
- A lookup table technique is applied for speedup. As all voxels are arranged on regular grids, they can be indexed by offset vectors $(\delta_x, \delta_y, \delta_z)$ with respect to a basis voxel. A lookup table relates an offset vector to the bin where its corresponding voxel should fall in. Then, all 3DSC vectors can be quickly collected by "voxel – offset vector – bin" indexing.

5 Feature Transformation

The collection of several hundred descriptors are not suitable for MVRVM regression. This is because the number of 3DSC vectors can slightly vary from frame to frame which prevents us from concatenating all 3DSC vectors. Therefore, a feature transformation step is required to convert all 3DSC vectors extracted from current voxel data to a feature vector in a high-dimensional space.

In [1], K-means clustering is applied to all 2DSC vectors from the training set, and each 2DSC vector votes for some near clusters. Hence, a collection of 2DSC vectors is transformed to a histogram, or feature vector. This method is simple but not elegant especially when dealing with high dimensional spaces, because it lacks of a clear definition on how much a descriptor contributes to each cluster. In contrast, we apply Mixture of Probabilistic PCA (MPPCA) introduced by Tipping [24] to model the density functions of descriptors and measure the contribution of 3DSC vectors to each component of the feature vector in a Bayesian manner.

Conventional PCA finds a low-dimensional linear projection that best represents the data in a least-squares sense. Without an associated probability model, it can not be used for Bayesian inference. In contrast, probabilistic PCA (PPCA) [24] determines the principal sub-space of the data via maximum-likelihood estimation of the parameters in a Gaussian latent variable model. Both PCA and PPCA only define a single global projection of the data. For complex data sets, different clusters may need different projection directions, so a mixture of local models is desirable. It is usually assumed that data t is generated from a mixture of component density functions, in which each component i corresponds to a cluster. As PPCA is defined as a probabilistic model, it can be easily extended to MPPCA, which is proved to outperform standard Gaussian Mixture Model [24]. The probability density of MPPCA with K components observing data t (3DSC vector in our case) is $p(t) = \sum_{i=1}^{K} \pi_i p(t \mid i)$, where $p(t \mid i)$ denotes PPCA density function for component i, which is a particular Gaussian distribution, and π_i is the mixing proportion. Accumulating all 3DSC vectors from the training set, the parameters of each PPCA and mixing proportions can be learned by maximizing the log-likelihood of the complete-data using EM algorithm.

After learning the MPPCA model, a compact feature vector $\vec{x}$ can be calculated conveniently to represent any 3D shape. We evaluate x_i to represent the averaged contribution of component i for generating a collection of 3DSC vectors extracted from current voxel

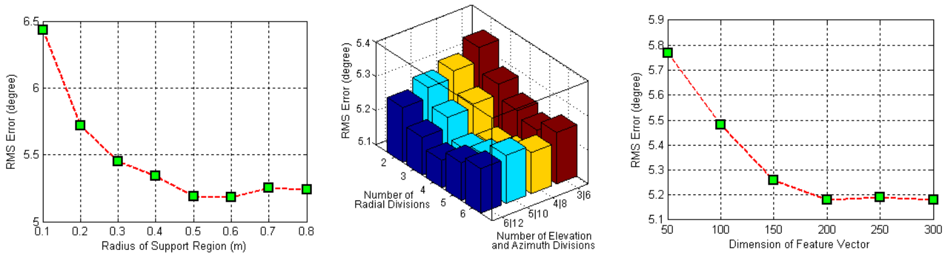

Figure 4: *Cross-validation experiments showing how the radius of support volume of 3DSC, number of 3DSC divisions and dimension of feature vector (number of components in MPPCA) influence the performance of our algorithm. 525 frames in the training set and 492 frames in the validation set are used. The optimal parameters chosen in this work are* $R_{max} = \frac{h_{sub}}{3}$, $N_r = 4$, $N_e = 5$, $N_a = 10$ *and* $K = 200$.

data via Bayesian inference.

$$x_i = \frac{1}{N_{sc}} \sum_{n=1}^{N_{sc}} p(i \mid t_n) = \frac{1}{N_{sc}} \sum_{n=1}^{N_{sc}} \frac{p(t_n \mid i)\pi_i}{\sum_{j=1}^{K} p(t_n \mid j)\pi_j}. \tag{5}$$

Where x_i is the $i-$th component of the feature vector which will be used as the input to regressor. N_{sc} is the number of 3DSC vectors collected from current voxel data. In this way, we are able to transform a collection of descriptors to compact feature vector effectively in a Bayesian framework.

6 Experimental Results and Analysis

To generate silhouette sequences (and voxel data consequently) for training and testing, we project an articulated body model represented by ellipsoids and spheres onto 6 circularly distributed viewpoints at each frame while performing motions. These motions are MOCAP data freely available from *www.ict.usc.edu/graphics/animWeb/humanoid* and *www.bvhfiles.com*. We report mean (over $m' = 42$ angles) RMS (over time) absolute difference errors between the true and estimated joint angle vectors[1], in degrees to evaluate the accuracy of our approach:

$$D(y', \hat{y}') = \frac{1}{m'} \sum_{i=1}^{m'} min(|y'_i - \hat{y}'_i|, |360 - |y'_i - \hat{y}'_i||) \tag{6}$$

Figure 4 shows how some free parameters influence the pose estimation, including radius of support volume of 3DSC, number of divisions in 3DSC and dimension of feature vector (number of clusters or components in MPPCA). If the 3DSC has a too small support volume, it is not able to encode local shape sufficiently for discriminating different body segments. Moreover, histograms with tiny bins are liable to shape distortion. As the radius of support region increases, the algorithm slows down as it needs to count more votes in each bin. It can be seen from Figure 4 (left) that the optimal choice for the radius lies between 0.5-0.6(m), which is about $1/3$ of the height of the articulated body model.

[1] The fact that Euler angles can wrap around 360° is considered. This equation is equivalent to that in [1].

Once $R_{max} = \frac{h_{sub}}{3}$ is fixed, the optimal number of radial divisions $N_r = 4$ can be easily determined from Figure 4 (middle). However, it is not clear how the numbers of elevation and azimuth divisions affect our regressor. As more divisions leads to more storage and computation, we choose $N_e = 5$ and $N_a = 10$ to balance accuracy and efficiency. In Figure 4 (right) while the dimension of feature vector keep increasing, the RMS error curve flattens out around optimal parameter $K = 200$. In other words, we can not expect substantial gains using more components in MPPCA only a much slower system.

Table 1 shows comparative results on the same MOCAP data between our method and the relevant reference [1]. Noticeably, they used a more detailed human body model including extra 12 subtle degrees of freedom, which vary minimally and tend to decrease their averaged error. Even so, we achieve 0.8° improvement on full body pose estimation. The improvement seems minor, but corresponds to substantial visual difference (See supplementary material), as the error is averaged over dozens of angles and hundreds of frames. We also implemented K-means in our system, listed in the third row of Table 1, to demonstrate the advantage of Bayesian feature transformation.

	full body	body heading angle	left shoulder	right hip
Our approach	5.2	8.8	6.3	3.2
[1]	6.0	17	7.5	4.2
Our approach (K-means)	5.4	12.0	6.5	3.8

Table 1: *RMS error over 418 frames of test MOCAP data (spiral walking). All results in second row can be found in [1]. The first and third row show comparative results between Bayesian and non-Bayesian feature transformation in our system.*

Our method can also be used to train a regressor for multiple motion types simultaneously. Table 2 summarizes the test results for 5 typical motions: regular walking, drunken walking, spiral walking, jogging and jumping, among which some complicated motions like drunken walking and jumping are rarely tested in the literature. Our 8-dimensional MVRVM regressor selects 418 (about 24%) relevance vectors out of 1744 training examples. Good performance on test set with 1411 frames can also be seen from Figure 5.

We carried out real-data experiment on 4-camera calibrated and segmented sequences from *www.cs.brown.edu/people/ls/Software*. As the subject in these sequences actually perform regular walking and spiral walking, the above regressor which is trained with MOCAP data from different people and different camera setup is applied here. Figure 6 illustrates the promising results of our method on real data.

7 Conclusions

This paper proposed a regression-based method for pose estimation from voxel data. On the one hand, as voxel data are basically viewpoint independent given enough cameras, we

motion	walking				
type	regular	drunken	spiral	jogging	jump
RMS error	2.5	5.0	5.3	5.6	4.9

Table 2: *The performance of regressor which is trained for 5 motion types simultaneously. 8-d* MVRVM *regressor selects 418 out of 1744 training examples.*

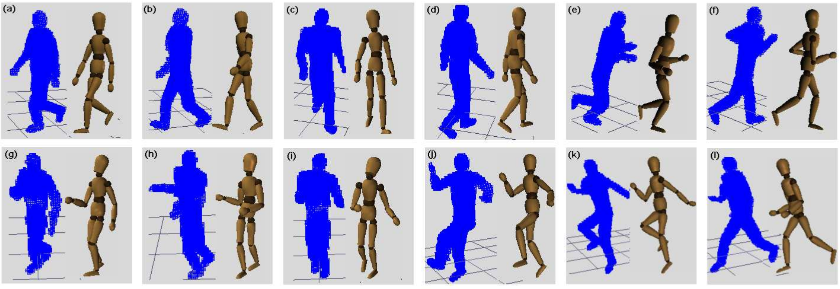

Figure 5: *A few results (voxel data and estimated pose) from the test set of 5 typical motion types. (a)-(b) regular walking. (c)-(d) spiral walking. (e)-(f) jogging. (g)-(i) drunken walking. (j)-(l) jumping.*

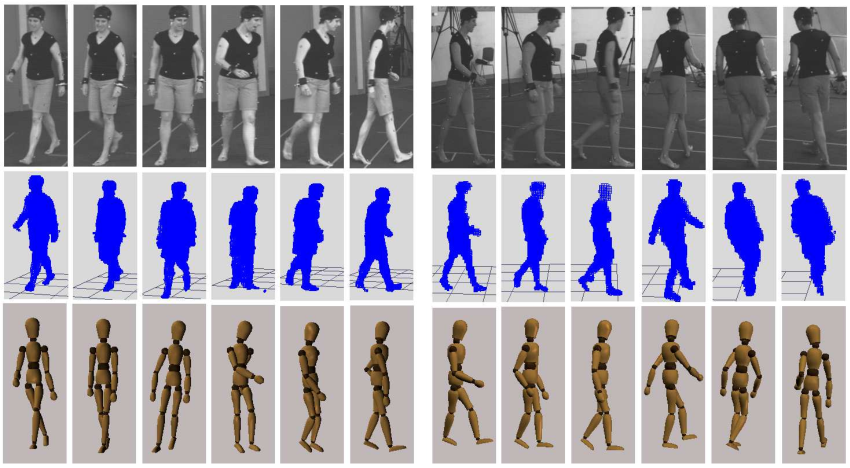

Figure 6: *Experiment on 4-camera calibrated real sequences. The left and right half show results from the 2nd and 4th viewpoints respectively. The regressor corresponding to Figure 5 is applied here to demonstrate the robustness of our method to different subjects and camera setup.*

do not need to re-learn the regressor each time the camera setup is changed. On the other hand, neither dynamic model nor explicit kinematic model are necessary in our method, so it can be used to initialize or reboot a model-based motion capture system automatically rather than manually as usual. Experiments on synthetic data and real sequences show the effectiveness and robustness of our method, even on complicated motions.

Acknowledgements : This work was funded by the National Natural Science Foundation of China (No. 60441002), the National Basic Research and Development Program (973) (No. 2004CB 318110) and University Key Research Project (No. 2003SE002).

References

[1] A. Agarwal and B. Triggs. 3d human pose from silhouettes by relevance vector regression. In *CVPR*, volume II, pages 882–888, 2004.

[2] A. Agarwal and B. Triggs. Learning to track 3d human motion from silhouettes. In *ICML*, volume II, pages 894–900, 2004.

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

[4] M. Brand. Shadow puppetry. In *ICCV*, pages 1237–1244, 1999.

[5] G.K.M. Cheung, T. Kanade, J. Bouguet, and M. Holler. A real time system for robust 3d voxel reconstruction of human motions. In *CVPR*, pages 714–720, 2000.

[6] A. Frome, D. Huber, R. Kolluri, T. Bulow, and J. Malik. Recognizing objects in range data using regional point descriptors. In *ECCV*, volume III, pages 224–237, 2004.

[7] K. Grauman, G. Shakhnarovich, and T. Darrell. Inferring 3d structure with a statistical image-based shape model. In *ICCV*, 2003.

[8] N. Howe, M. Leventon, and W. Freeman. Bayesian reconstruction of 3d human motion from single-camera video. In *NIPS*, 1999.

[9] N.R. Howe. Silhouette lookup for automatic pose tracking. In *Articulated and Nonrigid Motion Workshop*, 2004.

[10] R. Kehl, M. Bray, and L. Van Gool. Full body tracking from multiple views using stochastic sampling. In *CVPR*, volume II, pages 129–136, 2005.

[11] M. Kortgen, G.J. Park, M. Novotni, and R. Klein. 3d shape matching with 3d shape contexts. In *7th Central European Seminar on Computer Graphics*, 2003.

[12] I. Mikic, M. Trivedi, E. Hunter, and P. Cosman. Human body model acquisition and tracking using voxel data. *IJCV*, 53(3):199–223, 2003.

[13] R. Rosales and S. Sclaroff. Inferring body pose without tracking body parts. In *CVPR*, 2000.

[14] R. Rosales and S. Sclaroff. Learning body pose via specialized maps. In *NIPS*, 2001.

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

[16] Y. Sheikh and M. Shah. Bayesian modeling of dynamic scenes for object detection. *PAMI*, 27(11):1778–1792, 2005.

[17] H. Sidenbladh, M.J. Black, and D.J. Fleet. Stochastic tracking of 3d human figures using 2d image motion. In *ECCV*, volume II, pages 702–718, 2000.

[18] C. Sminchisescu, A. Kanaujia, Z. Li, and D. Metaxas. Discriminative density propagation for 3d human motion estimation. In *CVPR*, 2005.

[19] C. Sminchisescu and A. Telea. Human pose estimation from silhouettes: a consistent approach using distance level sets. In *WSCG International Conference on Computer Graphics, Visualization and Computer Vision*, 2002.

[20] C. Stauffer and W.E.L. Grimson. Learning patterns of activity using real-time tracking. *PAMI*, 22(8):747–757, 2000.

[21] A. Thayananthan, R. Navaratnam, B. Stenger, P.H.S. Torr, and R. Cipolla. Multivariate relevance vector machines for tracking. In *ECCV*, 2006.

[22] T.P. Tian, R. Li, and S. Sclaroff. Articulated pose estimation in a learned smooth space of feasible solutions. In *IEEE Workshop on Learning in CVPR*, 2005.

[23] M.E. Tipping. Sparse bayesian learning and the relevance vector machine. *Journal of Machine Learning Research*, 1:211–244, June 2001.

[24] M.E. Tipping and C.M. Bishop. Mixtures of probabilistic principal component analysers. *Neural Computation*, 11(2):443–482, 1999.