J. Software Engineering & Applications, 2010, 3, 1060-1066
doi:10.4236/jsea.2010.311125 Published Online November 2010 (http://www.SciRP.org/journal/jsea)

3D Human Pose Estimation from a Monocular Image Using Model Fitting in Eigenspaces

Geli Bo, Katsunori Onishi, Tetsuya Takiguchi, Yasuo Ariki

Graduate School of Engineering, Kobe University, Kobe, Japan.
Email: {takigu, ariki}@kobe-u.ac.jp

Received October 4th, 2010; revised October 26th, 2010; accepted November 3rd, 2010.

ABSTRACT

Generally, there are two approaches for solving the problem of human pose estimation from monocular images. One is the learning-based approach, and the other is the model-based approach. The former method can estimate the poses rapidly but has the disadvantage of low estimation accuracy. While the latter method is able to accurately estimate the poses, its computational cost is high. In this paper, we propose a method to integrate the learning-based and model-based approaches to improve the estimation precision. In the learning-based approach, we use regression analysis to model the mapping from visual observations to human poses. In the model-based approach, a particle filter is employed on the results of regression analysis. To solve the curse of the dimensionality problem, the eigenspace of each motion is learned using Principal Component Analysis (PCA). Finally, the proposed method was estimated using the CMU Graphics Lab Motion Capture Database. The RMS error of human joint angles was 6.2 *degrees using our method, an improvement of up to* 0.9 *degrees compared to the method without eigenspaces.*

Keywords*: *HOG, Regression Analysis, Eigenspaces, Particle Filter, Pose Estimation

1. Introduction

The 3D configuration estimation of complex articulated objects from monocular images has been widely studied. Once the technology is perfected, there will be potential applications in many fields related to human pose and kinematic information, such as computer interfaces that utilize gesture input, interaction with the robots, video surveillance, and entertainment. However, monocular human pose estimation is extremely challenging due to the complicated nature of human motion and the limited amount of information in 2D images.

The methods of human pose estimation can be summarized into two approaches: learning-based and model-based. In the learning-based method [1-4] features are directly extracted from the image, and the mapping function for the human poses is trained using the image features. Through this mapping, the human pose of an image can be estimated. Once the training process is completed, the pose estimation is performed rapidly. However, the estimation precision decreases when the input image is not included in the training data. In the model-based method [5-8], the pose estimation method follows Bayes' theorem and models the posterior probability density using observation likelihood or cost function. This method is computationally expensive, in general, and dependent on an initial pose.

To solve these problems, we propose a method to integrate the learning-based and model-based methods to improve the estimation accuracy. An initial pose is determined using regression analysis in the learning-based approach, and the estimation method is switched to a particle filter in the model-based approach to improve the precision. Unfortunately, given the large dimensionality of a 3D human model space, it is almost impractical to apply particle filtering directly as a large number of particles is required to adequately approximate the underlying probability distribution in the human pose space. Therefore, we first use PCA to learn the eigenspace of each motion. Then, the optimal human pose is efficiently searched in the eigenspaces selected according to the estimated type of human motion in the input images.

2. Features

2.1. Image Features

We use the HOG feature [9], which can describe the shape of an object in an appearance-based approach [10]. HOG was proposed as a gradient-based feature for general object recognition, where HOG describes the feature

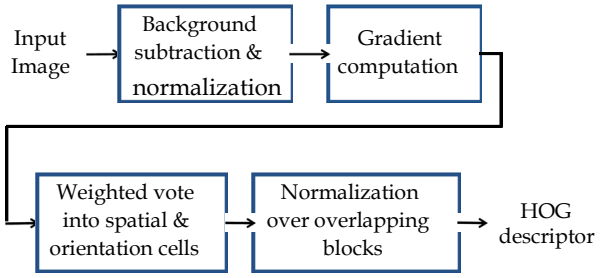


Figure 1. The flow of feature extraction.

over the given region. This means that HOG can represent the rough shape of an object. Moreover, since HOG can tolerate a range of varying illumination, it is suitable for pose estimation [11]. **Figure 1** presents the complete processing chain of the HOG feature extraction briefly. We will now discuss the HOG encoding algorithm in this section in detail.

2.1.1. Gradient Computation

Before extraction of the HOG feature, we first separate the human region from the input image using background subtraction, where the size of the human region is normalized, and the human region is located in the center position of the image. Then the image gradient is computed as follows.

$$\begin{cases} f_x(x,y) = I(x+1,y) - I(x-1,y) & \forall x,y \\ f_y(x,y) = I(x,y+1) - I(x,y-1) & \forall x,y \end{cases} \quad (1)$$

where f_x and f_y denote x and y components of the image gradient, respectively. $I(x,y)$ denotes the pixel intensity at the position (x,y). The magnitude $m(x,y)$ and orientation $\theta(x,y)$ are computed by

$$m(x,y) = \sqrt{f_x(x,y)^2 + f_y(x,y)^2} \quad (2)$$

$$\theta(x,y) = \tan^{-1}(f_y(x,y)/f_x(x,y)) \quad (3)$$

In order to make the HOG feature insensitive to clothing and the facial expression, we use the unsigned orientation of the image gradient, which is computed as follows.

$$\tilde{\theta}(x,y) = \begin{cases} \theta(x,y)+\pi, & if \quad \theta(x,y) < 0 \\ \theta(x,y), & otherwise \end{cases} \quad (4)$$

2.1.2. Orientation Histograms

The gradient image is divided into cells $c_w \times c_h$ pixels as shown in **Figure 2**. At each cell, the orientation $\tilde{\theta}(x,y)$ is quantized into c_b orientation bins, weighted by its magnitude $m(x,y)$ to make a histogram. That is, a histogram with the c_b orientations is computed for each cell.

2.1.3. Block Normalization

Figure 2 shows the orientation histogram extracted at

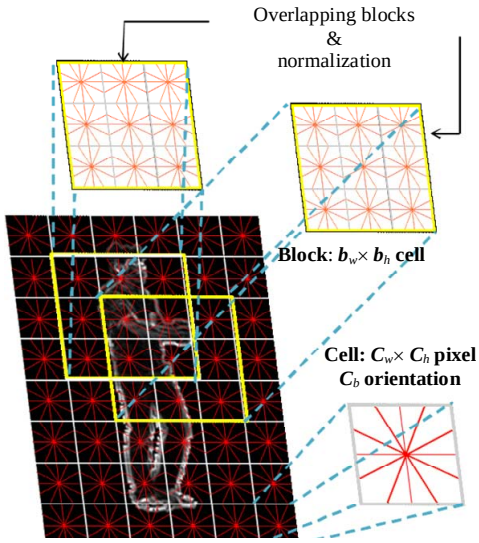


Figure 2. Block normalization.

every cell and the larger spatial blocks with $b_w \times b_h$ cells. Since a cell has c_b orientations, the feature dimension of each block is $d_b = b_w \times b_h \times c_b$ for each block. Let **v** denote a feature vector in a block, h_{ij} denote the unnormalized histogram of the cell in the position (i,j), $\{1 \le i \le b_w, 1 \le j \le b_h\}$ in a block. The feature vector of a certain block is normalized as follows.

$$h'_{ij} = \frac{h_{ij}}{\sqrt{\|v\|^2 + \varepsilon}} (\varepsilon = 1) \quad (5)$$

Since the normalization is done by overlapping the block, the histograms h_{ij} are repeatedly normalized by a different block.

2.2. 3D Human Model

The human body can be regarded as a multi-joint object that transforms into various shapes. In addition, the segmented part that connects two joints can be regarded as rigid object. Therefore, it is possible to express a 3D human model with joint angles. In this research, we used the motion capture data in the CMU Graphics Lab Motion Capture Database [12]. The 3D human model is represented by 56 joint angles, so the dimension of the pose state vector is 56. **Figure 3** shows an example of the 3D human model.

3. The Pose Estimation Method

Human pose estimation is carried out using two approaches. In the model-fitting method, human pose is estimated by an iteration procedure [5]. However, it is problematic in that the initial value has to be given manually. Therefore, we adopt the learning-based method to automatically obtain the initial value, which is

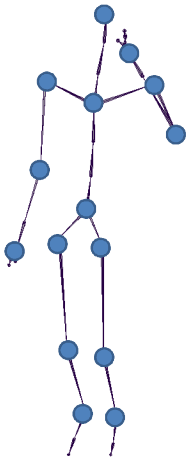


Figure 3. An example of 3D human model.

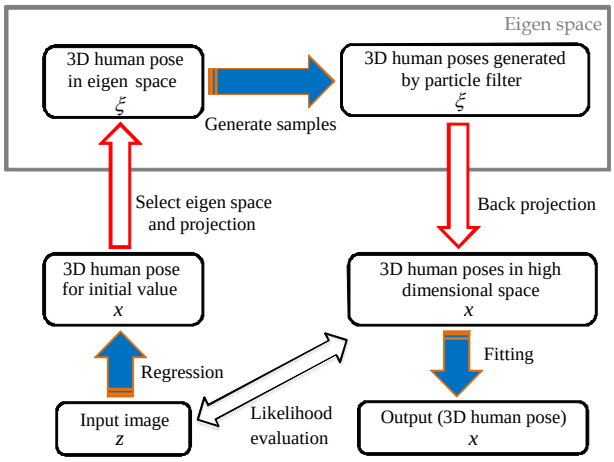


Figure 4. Pose estimation system.

integrated into the iteration procedure of the model-based method. One drawback in the model-based method is the high dimensionality of the state space, which makes the algorithm computationally ineffective. Thus we use PCA to reduce the dimension of the pose state and establish the eigenspace of each motion. **Figure 4** shows the proposed pose estimation system.

3.1. Learning-Based Method Using Regression Analysis

In the learning-based method, we adopt regression analysis [1,13] to estimate the pose of input image. Let x denote the vector composed of the angles at joints in the 3D human model. The relation between the HOG feature vector z and 3D pose vector x is linearly approximated using the following formula:

$$x = Rz + \varepsilon \tag{6}$$

where ε is residual error vector. The 3D human pose is

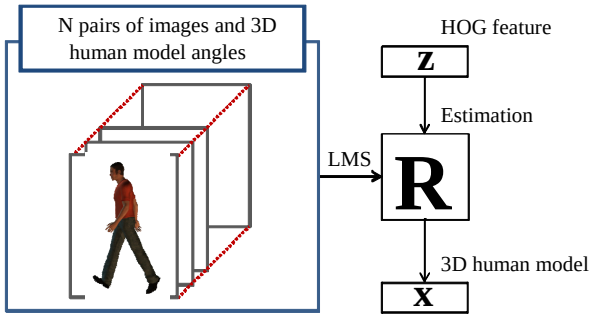


Figure 5. Regression-based estimation method.

estimated by converting the input image feature z to the 3D human model vector x. In model training (estimate R), a set of n training pairs $\{(x_i, z_i) | i = 1 \cdots n\}$ is given (in our case, 3D poses and the corresponding image HOG features). The conversion matrix R is estimated by minimizing the mean square error. Packing the training data into 3D pose matrix $X \equiv (x_1 x_2 \cdots x_n)$ and image feature matrix $Z \equiv (z_1 z_2 \cdots z_n)$, the training is performed as follows:

$$R := \arg\min_R \left\| RZ - X \right\|^2 \tag{7}$$

In the testing phase, the 3D human posture vector x is estimated by converting HOG features vector z using the computed conversion matrix R. **Figure 5** demonstrates the regression-based estimation method.

3.2. Model-Based Method Using a Particle Filter

In the model-based method, a particle filter [14] is employed. Following a notation similar to [14], we define x_t as the state vector at time t, with z_t denoting the measurements at time t. Furthermore, let all the measurements until time t be given by $Z_t = (z_1, \cdots, z_t)$. Particle filtering based on Bayes' theorem is used to obtain a posterior probability $p(x_t | Z_t)$ at each time-step using all the available information as shown bellow.

$$p(x_t | Z_t) \propto p(z_t | x_t) p(x_t | Z_{t-1}) \tag{8}$$

This equation is evaluated recursively as described below. The fundamental idea of particle filtering is to approximate the posterior probability density function (pdf) over x_t by a weighted sample set S_t. Suppose that N samples from the posterior pdf $p(x_t | Z_t)$ are available and denote them as $x_t^{(i)}$. Then the i' th weighted sample at time t is represented by $s_t^{(i)} = (x_t^{(i)}, \pi_t^{(i)}) \in S_t$.

First, a cumulative histogram of all the samples' weights is computed at time $t-1$. Then, according to each particle's weight $\pi_{t-1}^{(i)}$, its number of successors is determined according to its relative probability in this cumulative histogram. At the prediction step, the new state x_t is computed using the following Chapman-

Kolmogorov equation.

$$p(x_t \mid Z_{t-1}) = \int_{x_{t-1}} p(x_t \mid x_{t-1}) p(x_{t-1} \mid Z_{t-1}) dx_{t-1} \quad (9)$$

At the measurement step, the new state x_t is weighted according to its likelihood to the new measurement z_t. The posterior density $p(x_t \mid Z_t)$ is represented by a set of weighted particles $\{(s_t^{(0)}, \pi_t^{(0)}) \cdots (s_t^{(N)}, \pi_t^{(N)})\}$, where the weights $\pi_t^{(n)} \propto p(z_t \mid x_t = s_t^{(i)})$ are normalised so that $\sum_N \pi_t^{(n)} = 1$. The new state x_t can be estimated by

$$p(x_t \mid Z_t) \approx \sum_{i=1}^{N} \pi_t^{(n)} (s_t^{(i)}) \quad (10)$$

The measurement step of Equation (10) and the prediction step of Equation (9) together form the Bayes' formulation Equation (8).

In the ordinary model-based method, a particle filter is utilized to match the 3D model with the input image, and the initial value needs to be set manually [15,16]. In our method, the initial value can be obtained from the learning-based method. Therefore, the former manual configuration will be replaced by an automatic estimation process.

The pose estimated by regression analysis is used as an initial value, and the particles are sampled around it. The likelihood of each particle is evaluated as its weight, and the particles are generated by a resampling process based on the weight. After repeating resampling several times, the particle with the highest likelihood is considered as the final state.

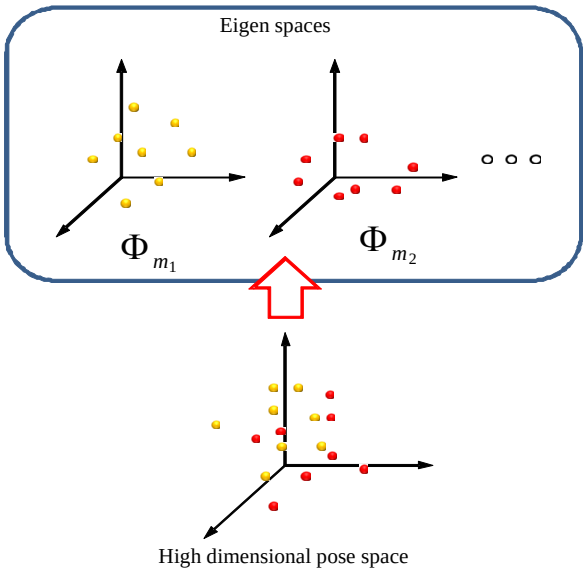


Figure 6. Original space and eigenspaces.

3.3. State Space

The 3D human model was introduced in Subsection 2.2. In this section, it is treated as the state vector of particle filter. However, in a real environment, the dimension of state space is normally very high. That would cause both low computational efficiency and poor convergence performance.

We propose the method of utilizing an eigenspace of each motion constructed by PCA as a motion prior, which constraints corresponding motion [17]. Simultaneously, it is possible to search efficiently in the low-dimensional space using dimension reduction. Suppose that the number of motion types is M. When PCA is carried out using the training data of a certain motion $m \in M$, the 3D human pose x_m is projected into the eigenspace as follows:

$$\xi_m = P_m (x_m - \bar{x}_m) \quad (11)$$

where $\bar{x}_m$ denotes the mean pose vector of a certain motion m, P_m is the base vector matrix, and ξ_m denotes the pose vector in the eigenspace. Because PCA is carried out for M types of motions in the training data, M kinds of eigenspaces are constructed. The pose vector x_m is projected to the eigenspace of each motion and is used as the state of particle as shown in **Figure 6**. The dimension reduction using PCA is decided according to the 95% cumulative proportion rate.

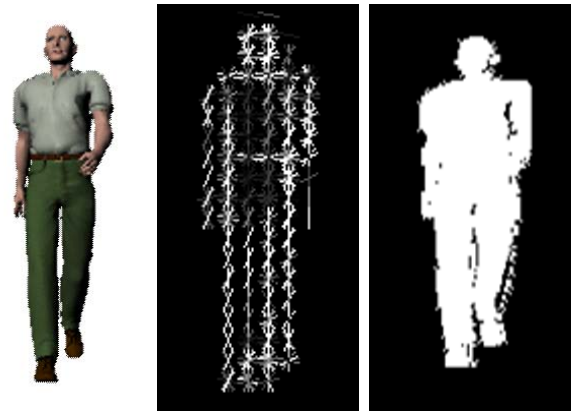
3.4. Likelihood

In the likelihood calculation stage, the state vector of every particle is converted to a 3D human pose in a high dimension space using PCA Inverse Transformation. Then, we use MAYA to produce the CG image that represents the pose of every particle. This CG image compared with the input image. The performance of the particle filter depends largely on the image features that are used to calculate the likelihood. The ideal image features should remain stable in various scenarios and be easy to extract. In our case, we adopt two features to construct weighting function: HOG for representing the human conformation and silhouetting for evaluating the human region.

Configuration:

HOG represents human conformation and is robust regarding changes in color, clothes and illumination. The CG image is generated from the particle state, and HOG is extracted (**Figure 7(b)**). The distance between the input image x and the state ξ is calculated as follows:

$$E_{hog}(x, \xi) = \frac{1}{Dim} \chi^2(z, z') \quad (12)$$



(a) (b) (c)

Figure 7. Feature extraction (a) CG image (b) HOG descriptor (c) silhouette.

$$\chi^2(z,z') = \sum_{c=1}^{Dim} \frac{(z^c - z^{c'})^2}{z^c + z^{c'}} \tag{13}$$

where Dim denotes the dimension of the HOG, $\chi^2(z,z')$ is χ^2-distance between z and z'. z^c indicates the c-th element of the HOG feature vector.

Region:

The silhouette image is extracted using the background subtraction method (**Figure 7(c)**). The silhouette can be used for evaluation with stability because it is also robust to the changes of color, clothes and illumination. After the silhouette is extracted, once again a pixel map is constructed, this time with foreground pixels set to 1 and back ground to 0, and the distance is computed as follows:

$$E_{region}(x,\xi) = \frac{1}{K}\sum_{i=1}^{K}(p_i(x,\xi)) \tag{14}$$

where K is the number of pixels, and $p_i(x,\xi)$ the values of the binary $EXOR$ operation between input image and state.

Next, fitness $C(\xi)$ on the image is computed as follows.

$$C(\xi) = \exp\left\{-(E_{hog}(x,\xi) + E_{region}(x,\xi))\right\} \tag{15}$$

In our method, the solution search is restricted to the eigenspace of the corresponding motion. Nevertheless, the pose of an image can be further constrained in a certain range of the eigenspace, where we use the trajectory of motion as a prior constraint. We regard the sequence of vectors in the eigenspace as the trajectory of this motion. Then the distance is calculated between state vector ξ and vectors of training data in eigenspaces, and it can be used for the likelihood evaluation as a penalty P. Finally, the best solution is obtained by the rule shown in Equation (16).

$$L = \lambda C(\xi)\mathrm{P}^{-1} \tag{16}$$

3.5. Selecting the Eigenspace

In our method, it is necessary to select the proper eigenspace according to the estimated motion of the input image because the eigenspace is constructed for each motion. First, many samples $\left\{\xi_m^i \middle| m \in M, i = 1,\cdots,S_m\right\}$ are embedded into each eigenspace using the training data. The mean distance between the samples and initial pose x obtained by the regression analysis is computed, and the input motion is decided as the motion with the nearest mean distance.

The mean distances are computed in the eigenspace and the high-dimensional space respectively as follows.

$$\omega_m = \frac{1}{S_m}\sum_{i=1}^{S_m}\frac{1}{D_m}\left\|\xi - \xi_m^i\right\|_2 \tag{17}$$

$$\Omega_m = \frac{1}{S_m}\sum_{i=1}^{S_m}\frac{1}{D}\left\|x - x_m^i\right\|_2 \tag{18}$$

Here D_m and D denote the dimension of eigenspace and an original space respectively, and ξ denotes the low-dimensional pose vector to which x is mapped in an eigenspace. x_m denotes the high-dimensional pose vector in which ξ_m is reverted to an original space using PCA Inverse Transformation. S_m is the number of the samples. The motion of the input image is decided by minimizing the sum of two distances defined by Equation (17) and Equation (18) as shown in Equation (19). $f(m)$ is defined as a function to determine which an eigenspace Φ_m the motion m belongs to:

$$\Phi_m = f(\arg\min_{m\in[1,\cdots,M]}\{\omega_m + \Omega_m\}) \tag{19}$$

3.6. Estimation of the Human Orientation

Even if the orientation of the human ϕ is unknown, the human pose can be estimated with regression analysis because the image feature changes according the orientation. However, if the orientation is not estimated in the model-based method, the image cannot be matched.

Consequently, the images in all orientations of 360 degrees are generated by CG using the results of the regression analysis, and human orientation is estimated using Equation (15) as follows:

$$\hat{\phi} = \arg\max_{1\le\phi\le 360^\circ} C(x^\phi) \tag{20}$$

where $C(x^\phi)$ denotes the fitness between input image and the generated image from pose x with orientation ϕ. The model-fitting is carried out using the estimated orientation $\hat{\phi}$ obtained by Equation (20).

4. Estimation of the Human Orientation

4.1. Experiment Setup

We conducted the experiment using the CMU Graphics Lab Motion Capture Database. First, we use motion capture data to produce a CG animation whose resolution is 640×480 pixels. Then we rotate the figure on the horizontal plane in eight directions and take the CG image in each direction as experiment data. We carried out experiments for three kinds of motion (walking, running, and jumping). The images used for training are summarized in **Table 1**.

If the test motion is a cyclic movement, only four images of the typical pose are needed to represent a certain cyclic motion. In order to represent the motion sufficiently, we used eight images in each direction for a continuous action. **Table 2** lists the details of the test data.

4.2. Experiment Results

RMS of absolute difference errors was computed between the true joint angles x and estimated joint angles x', by Equation (21). m indicates the number of joints.

$$D(x, x') = \frac{1}{m}\sum_{i=1}^{m}\left|(x_i - x_i') \bmod \pm 180^\circ\right| \qquad (21)$$

Figure 8 shows the RMS error over all joints angles for all the motions. The estimation precision was improved by iteration procedure.

The horizontal axis indicates the number of iterations. The regression analysis results are shown in the primary iteration. After the next iteration, pose estimation is achieved by repeatedly applying the particle filter. The results show that the accuracy of estimation can be improved significantly because of the use of the trajectory as a constraint.

Table 3 shows the result of eigenspace selection described in Subsection 3.5. One reason for the increased accuracy is that, in our method, eigenspace selection is not applied to sequence data but to just one frame image. Therefore, the motion recognition accuracy is not so

Table 1. The number of training data.

pose	The number of frames	
	1 orientation	Total (8 orientations)
Walking	90	720
Running	189	1,512
Jumping	192	1,536
Total	471	3,768

Table 2. The number of test data.

pose	The number of frames	
	1 orientation	Total (8 orientations)
Walking	8	64
Running	8	64
Jumping	8	64
Total	24	192

Table 3. Confusion matrix [%].

	Walking	Running	Jumping
Walking	**84.38**	3.12	12.5
Running	9.37	**78.13**	12.5
Jumping	15.63	4.68	**79.69**

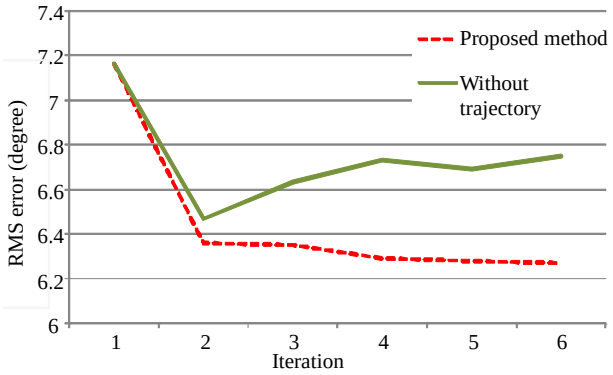


Figure 8. Pose estimation results.

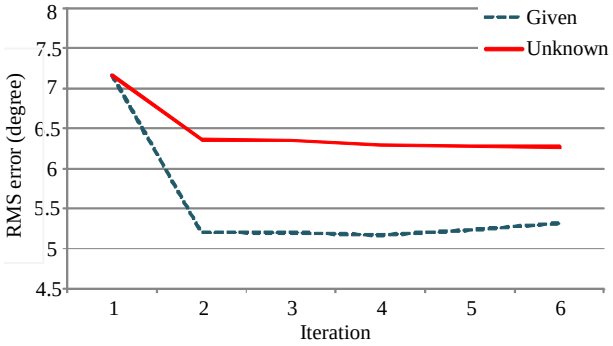


Figure 9. Estimation precision in eigenspaces.

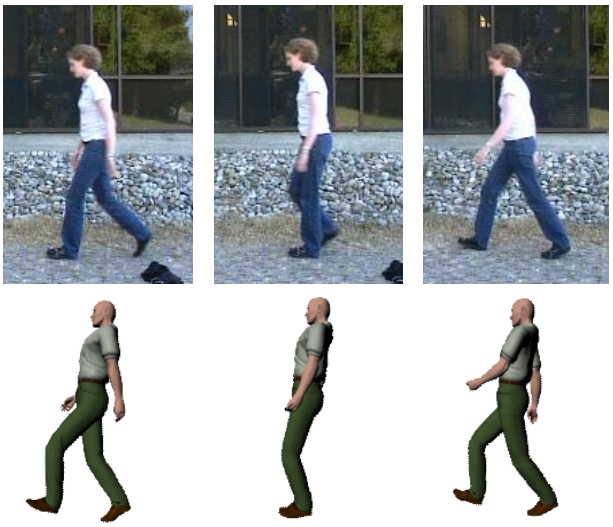


Figure 10. Experimental result of the real image.

good.

The selection of a proper eigenspace will greatly affect the final estimation results. **Figure 9** compares the accuracy of two different estimation methods. The blue broken line represents the scenario in which a motion type of input image has been given, and its estimation is carried out in the corresponding eigenspace. Comparing these results with the red line for an unknown motion type, it can be concluded clearly that our method shows a better performance in terms of estimation precision.

The results of human pose estimation from real images [18,19] are shown in **Figure 10**. The first row is the input real images and the second row represents the synthetic images generated from the estimated poses. It is confirmed that our method works effectively for the real images.

5. Conclusions

In this paper, we presented an approach to estimate 3D human pose from a monocular image, which integrates the learning-based and model-based estimation methods into one framework. Furthermore, through the construction of an eigenspace, more efficient particle filter performance is obtained.

Consequently, the precision of estimation is obviously improved, and experimental results demonstrated that our approach is effective. The particle filter did not need to have the initial value provided manually, and it could get the convergence solution in the situation with less iterations. In future work, in order to further improve the estimation accuracy, we are planning to use video data for the input data because the temporal coherence between frames may provide useful information regarding the selection of the eigenspace of each motion.

REFERENCES

[1] A. Agarwal and B. Triggs, "3D Human Pose from Silhouettes by Relevance Vector Regression," *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, Vol. 2, 2004, pp. 882-888.

[2] X. Zhao, H. Ning, Y. Liu and T. Huang, "Discriminative Estimation of 3D Human Pose Using Gaussian Processes," *Proceedings of* 19*th International Conference on Pattern Recognition* (*ICPR*'08), December 2008, pp. 1-4.

[3] C. Sminchisescu, A. Kanaujia and D. N. Metaxas, "BM^3E : Discriminative Density Propagation for Visual Tracking," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 29, No. 11, November 2007, pp. 2030-2044.

[4] H. Ning, Y. Hu and T. Huang, "Efficient Initialization of Mixtures of Experts for Human Pose Estimation," 15*th IEEE International Conference on Image Processing* (ICIP2008), October 2008, pp. 2164-2167.

[5] M. Lee, I. Cohen, "A Model-Based Approach for Estimating Human 3D Poses in Static Images," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 28, No. 6, June 2006, pp. 905-916.

[6] T. Jaeggli, E. Koller-Meier and L. V. Gool, "Learning Generative Models for Monocular Body Pose Estimation," *Proceedings of the* 8*th Asian Conference on Computer Vision*, Vol. 1, 2007.

[7] S. Hou, A. Galata, F. Caillette, N. Thacker and P. Bromiley, "Real-time Body Tracking Using a Gaussian Process Latent Variable Model," *IEEE* 11*th International Conference on Computer Vision*, October 2007, pp. 1-8.

[8] G. Peng, W. Alexander, A. O. Balan and M. J. Black, "Estimating Human Shape and Pose from a Single Image," *IEEE* 12*th International Conference on Computer Vision* (*ICCV*2009), September 2009, pp. 1381-1388.

[9] N. Dalal and B. Triggs, "Histograms of Oriented Gradients for Human Detection," *IEEE Computer Society Conference on Computer Version and Pattern Recognition* (*CVPR*2005), Vol. 1, June 2005, pp. 886-893.

[10] G. Mori and J. Malik, "Recovering 3D Human Body Configurations using Shape Contexts," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 28, No. 7, 2006, pp. 1052-1062.

[11] K. Onishi, T. Takiguchi and Y. Ariki, "3D Human Posture Estimation Using the HOG Features from Monocular Image," 19*th International Conference on Pattern Recognition* (*ICPR*2008), December 2008, pp. 1-4.

[12] CMU Human Motion Capture Database. Available online: http://mocap.cs.cmu.edu/

[13] A. Fossati, M. Salzmann and P. Fua, "Observable subspaces for 3D human motion recovery," *IEEE Conference on Computer Version and Pattern Recognition* (*CVPR*2009), June 2009, pp. 1137-1144.

[14] M. Isard and A. Blake, "Condensation-Conditional Density Propagation for Visual Tracking," *International Journal of Computer Vision*, Vol. 29, No. 1, 1998, pp. 5-28.

[15] J. Deutscher, A. Blake and I. Reid, "Articulated Body Motion Capture by Annealed Particle Filtering," *IEEE Computer Society Conference on Computer Version and Pattern Recognition*, Vol. 2, June 2000, pp. 126-133.

[16] L. Ye, Q. Zhang and L. Guan, "Use Hierarchical Genetic Particle Filter to Figure Articulated Human Tracking," *International Conference on Multimedia and Expo* (*ICME*2008), 2008, pp. 1561-1564.

[17] X. Zhao and Y. Liu, "Tracking 3D Human Motion in Compact Base Space," *IEEE Workshop on Applications of Computer Vision* (*WACV*'07), February 2007, p. 39.

[18] H. Sidenbladh, M. Black and D. Fleet, "Stochastic Tracking of 3D Human Figures Using 2D Image Motion," *Computer Vision — ECCV* 2000, Vol. 1843, 2000, pp. 702-718.

[19] R. Urtasun, D. Fleet and P. Fua, "Monocular 3D Tracking of the Golf Swing," *IEEE Computer Society Conference on Computer Version and Pattern Recognition* (*CVPR*2005), Vol. 2, June 2005, pp. 932-938.

