

# MARKERLESS HUMAN MOTION CAPTURE AND POSE RECOGNITION

Feifei Huo<sup>1</sup>, Emile Hendriks<sup>1</sup>, Pavel Paclik<sup>2</sup>, Stijn Oomes<sup>1</sup>

<sup>1</sup>Delft University of Technology, The Netherlands

<sup>2</sup>PR Sys Design, Delft, The Netherlands

## ABSTRACT

In this paper, we present an approach to capture markerless human motion and recognize human poses. Different body parts such as the torso and the hands are segmented from the whole body and tracked over time. A 2D model is used for the torso detection and tracking, while a skin color model is utilized for the hands tracking. Moreover, 3D location of these body parts are calculated and further used for pose recognition. By transferring the 2D and 3D coordinates of the torso and both hands into normalized feature space, simple classifiers, such as the nearest mean classifier, are sufficient for recognizing predefined key poses. The experimental results show that the proposed approach can effectively detect and track the torso and both hands in video sequences. Meanwhile, the extracted feature points are used for pose recognition and give good classification results of the multi-class problem. The implementation of the proposed approach is simple, easy to realize, and suitable for real gaming applications.

## 1. INTRODUCTION

Nowadays, with the availability of faster and cheaper computer hardware, combined with cheaper and better digital cameras, video-based applications have become more and more widespread. A well-known video-based application is man-machine interaction, in which people can use their facial expressions, gestures and poses to control e.g. virtual actors or (serious) games. Human motion capture has received much attention due to such applications [1]. However, many of them are marker-based [2], [3]. People need to wear specific suits with markers on it to track the movement of different body parts, which is not convenient for real applications. To solve this problem, a markerless human motion capture system is desired. In this paper an approach to capture human motions without markers is presented and the extracted feature points are used for pose recognition.

## 2. PREVIOUS RESEARCH

Although required for many natural applications such as pose recognition, there is still no generic solution to markerless motion capture. In [4] the skeleton points of a human are computed by using a silhouette model. Instead of calculating the 3D position of skeleton points, a topology of the human body structure is employed for limb labeling. The proposed method can deal with various viewpoints of a person, such as front, rear and profile. It also gives a proper limb labeling for unspecified human postures. However, they only use the graph topology matching to label different body parts, which has difficulty in dealing with the situation that the arms are merged with the torso. A real-time

human motion analysis system is presented in [5], which combines a silhouette-based approach and a color-blob-based approach to get feature points. Although they realize real-time tracking by using a PC cluster to process images from six views, this implementation is still quite expensive for practical applications. In [6] the proposed algorithm uses 2D images for gesture recognition. The thresholded Radon transform coefficients are used to extract the most important local regions. One of the limitations of this algorithm is that it can not deal with self-occlusion of the human body.

In contrast to previous work, in this paper we introduce an effective method to track the movement of different body parts, such as torso and hands. The 3D location of these body parts are calculated and used for human pose recognition. The proposed human motion capture and pose recognition system is illustrated in Fig.1. The first step is human body detection and body parts segmentation, using multiple features, such as shape, contour and color. The second step is feature points representation and tracking in subsequent video frames. The 3D positions of selected feature points are calculated by using multiple calibrated cameras. The last step is pose recognition by using relative positions of selected feature points.

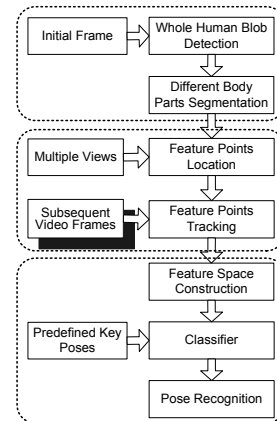


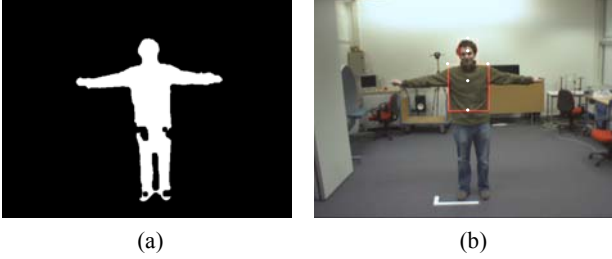
Fig.1. Human motion capture and pose recognition system.

## 3. METHODOLOGY

### 3.1. Background subtraction

Motion is one of the important visual cues to find out the “interesting object” in the scene. Therefore, in our approach, we use background subtraction to segment moving objects. The background image is built by using a mixture of  $k$  Gaussian models, which is presented in [7]. In order to deal with changing

lighting conditions, the background image is updated over time by current frames. This method can also handle tracking of moving objects through cluttered scenes. An example of the obtained foreground binary image is shown in Fig.2. (a).

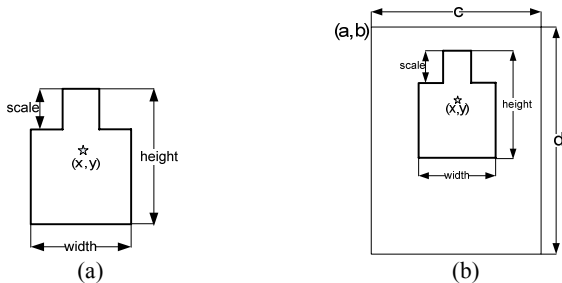


**Fig.2.** (a) Foreground binary image of the initial frame, (b) 2D upper-body model for human torso detection and tracking.

### 3.2. 2D model for human torso detection and tracking

For the detection and tracking of humans, we applied a basic 2D model of a human's head-shoulder-upperbody. This model is simple, but generic and successfully applied in [8]. The model is composed of two rectangles (Fig.2. (b)) and parameterized as  $P = (x, y, scale)$ .  $x, y$  represent the position of the model in a 2D image and scale indicates the size of the model. Since a full searching is not feasible due to time constraints, and particle filters can deal with non-Gaussian motion models and multiple instances, we applied a particle filter both for people detection and tracking.

For people detection, an initial frame in a video sequence is used. The initial frame is chosen as the frame that shows a person with a specific pose (Fig.2. (b)). It indicates the start of the system. In order to reduce the search region and realize multiple people detection, the binary image of the initial frame is first segmented into connected blobs. Blobs which are impossible to include persons are discarded by judging their size. Then a particle filter is used on each candidate blob to determine if it includes a person or not. In particle filtering, samples  $s^{(n)} = p^{(n)} = (x^{(n)}, y^{(n)}, scale^{(n)})$  are known as particles (Fig.3.(a)). The position parameters  $(x^{(n)}, y^{(n)})$  and scale parameters  $(scale^{(n)})$  are initialized with a Gaussian distribution. If the coordinate of the upper left corner of the blob bounding box is denoted as  $(a, b)$ , the center of position parameters  $(x^{(n)}, y^{(n)})$  is at  $(a + c/2, b + d/6)$ , with  $c$  and  $d$  the width and length of the blob bounding box along the  $x$  and  $y$  directions. It makes sure that the distribution of samples is centered at the upper middle of the blob bounding box (Fig.3. (b)).



**Fig.3.** (a) One sample in a particle filter, (b) the position distribution of particles is centered at the upper middle of the blob bounding box.

Particle filtering is an iterative process, which can also be extended to successive images in a video sequence for object tracking [9]. Since there exists a large correlation between consecutive video frames, detection results from the previous frame, such as position and scale of the person, are very relevant to that of the current frame. At the same time the current frame may differ from the previous frame, so a drift term is introduced to account for the new information in the current frame.

Although several feature points such as head top, head center, torso center, torso bottom and both shoulders can be estimated from the 2D model, our pose recognition system only use the torso center to present the person's location. The extraction of additional features is outlined in the next section.

### 3.3. Hand detection and tracking

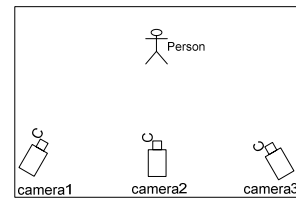
In addition to the 2D model mentioned in 3.2 for human's torso detection and tracking, foreground pixels are further segmented into skin-color and non-skin-color regions. A skin color model in the RGB color-space is used to select skin color pixels on the foreground image. This human skin color model is similar to the model in [10]. If foreground pixels mapping into the RGB color-space satisfy the following conditions, they will be considered as skin-color pixels.

$$\left| \arctan\left(\frac{B}{R}\right) - \frac{\pi}{4} \right| < \frac{\pi}{8}, \quad \left| \arctan\left(\frac{G}{R}\right) - \frac{\pi}{6} \right| < \frac{\pi}{18}, \quad \left| \arctan\left(\frac{B}{G}\right) - \frac{\pi}{5} \right| < \frac{\pi}{15}$$

After the skin color pixels are selected, two post-processing steps are used to get rid of false positive detections. The first step is to delete regions with very small size, which are impossible to be face and hand regions. In the second step, a motion mask is introduced to exclude regions which are far way from previous hand locations. It limits the movement of hands within a certain bounding box. Additionally, the face region can be separated from the hands regions, either by using the size of the connected skin color area, or the head location estimated from 3.2. In our approach, the size information is used, which is enough to distinguish the face region from the hand regions. From the remaining blobs, we calculate the centers of gravity and use them to represent the position of the hands.

### 3.4. 3D reconstruction

Until now the obtained torso center and both hand positions are from a single view, but the method can be easily applied to other views as well. The multiple camera set-up is shown in Fig.4; there are three cameras in total. One of the cameras (camera2 in Fig.4) is located at the front of the recording room, which captures the front view of the user. The other two (camera1 and camera3 in Fig.4) are in the corners of the room. They give two side views of the user.



**Fig.4.** Multiple camera settings.

Since these three cameras are synchronized, the 3D positions of a torso and hands of a human body can be obtained by using calibrated cameras. As for the hand position, we make the assumption that the left hand is always on the left side of the torso for all three views, and the right hand is on the right side of the torso. Therefore, left hand and right hand positions can be identified for all views and used to calculate 3D positions.

### 3.5. Feature space construction

The input of the proposed pose recognition system are 2D (front view camera) and 3D positions of the torso center and the hands. However, we transfer them into normalized feature space and train the classifier in this new feature space. The reason is that the pose recognition system should be scene invariant. That is, no matter where the person is in the scene, or how far the person is from the cameras, the predefined key poses should be recognized. Therefore the feature space is built by using relative positions between hands and torso center, such as distances and angles. Based on this, we construct 20 feature components as follows, which are denoted as

$$F_{set} = \{c_1, c_2, c_3, \dots, c_{20}\}.$$

$$c_1 = \frac{(x_2^l - x_2^r)}{s}, \quad c_2 = \frac{(y_2^l - y_2^r)}{s}, \quad c_3 = \frac{(x_2^r - x_2^l)}{s},$$

$$c_4 = \frac{(y_2^r - y_2^l)}{s}, \quad c_5 = \arctan \frac{y_2^l - y_2^r}{x_2^l - x_2^r}, \quad c_6 = \arctan \frac{y_2^r - y_2^l}{x_2^r - x_2^l},$$

$$c_7 = x_3^l - x_3^r, \quad c_8 = y_3^l - y_3^r, \quad c_9 = z_3^l - z_3^r,$$

$$c_{10} = x_3^r - x_3^l, \quad c_{11} = y_3^r - y_3^l, \quad c_{12} = z_3^r - z_3^l,$$

$$c_{13} = \frac{x_3^l - x_3^r}{s}, \quad c_{14} = \frac{y_3^l - y_3^r}{s}, \quad c_{15} = \frac{z_3^l - z_3^r}{s},$$

$$c_{16} = \frac{x_3^r - x_3^l}{s}, \quad c_{17} = \frac{y_3^r - y_3^l}{s}, \quad c_{18} = \frac{z_3^r - z_3^l}{s},$$

$$c_{19} = \sqrt{(x_3^l - x_3^r)^2 + (y_3^l - y_3^r)^2 + (z_3^l - z_3^r)^2} / s,$$

$$c_{20} = \sqrt{(x_3^r - x_3^l)^2 + (y_3^r - y_3^l)^2 + (z_3^r - z_3^l)^2} / s.$$

Here  $(x_2^l, y_2^l)$ ,  $(x_2^r, y_2^r)$  and  $(x_2^r, y_2^l)$  are the 2D positions of the torso center, left hand and right hand.  $(x_3^l, y_3^l, z_3^l)$ ,  $(x_3^r, y_3^r, z_3^r)$  and  $(x_3^r, y_3^l, z_3^l)$  are the 3D positions of the torso center, left hand and right hand. The obtained scale parameter in the 2D model  $P = (x, y, scale)$  is indicated by  $s$ . The classifier will be trained and tested on this defined feature space  $F_{set}$ .

## 4. EXPERIMENTS

### 4.1. Video recording

Before the start of the video recording, we take some snapshots for the purpose of calibration. We recorded videos of 15 volunteers from 6 races (Netherlands, China, France, Italy, Turkey and Syria). Five of them are female and the others are male. The predefined key poses are shown in Fig.5. The images are from the front view camera (camera2 in Fig.4).



Fig.5. Predefined key poses.

### 4.2. Implementation

In the background subtraction implementation, the number of the Gaussian models is chosen to be 3. In order to exclude shadows from the foreground image, a shadow removing approach is also employed. As for the particle filter, we choose the number of particles to be 500, which is a trade-off between precision and computing time.

### 4.3. Pose classification

The key poses are designed for gaming control, so they should be easy for users to remember and perform. The number of the poses should not be too high; otherwise it also increases the difficulty for users. In our system, we defined nine poses in total, as shown in Fig.5. From top row to bottom row and left to right, these nine poses are labeled as pose1 to pose9. In order to build a classifier, we manually labeled the frames containing the nine poses into nine classes. For each pose, the samples are selected from each of the 15 persons. Our experimental data set contains 1515 samples of 9 pose types (classes) and 20 features. On average, each pose class is represented by 170 samples.

## 5. RESULTS AND DISCUSSION

We evaluated pose classifiers using two cross-validation approaches. The first one is the leave-one-person-out (LOPO) where in each step (fold) we leave out all the samples corresponding to one person as the test set and use the samples of the remaining 14 persons for training the pose classifier. The LOPO procedure is repeated 15 times (folds). The other approach is randomly splitting each of the nine classes into 15 parts, using 14 as training set and one as testing set. We call the second approach 15-fold rotation (FORO). We compared performances of several statistical classifiers with different complexities. Specifically, we evaluated the nearest mean classifier (NMC), the linear classifier (LDC) and the quadratic classifier (QDC) assuming normal densities and the non-parametric Parzen classifier<sup>1</sup>. The QDC and Parzen were built either directly in the

<sup>1</sup>We used PRTools toolbox <http://prtools.org> and PRSD Studio <http://prsdstudio.com> software packages.

20D feature space or in the 8D subspace derived by a supervised feature extractor: linear discriminant analysis (LDA). The results are shown in Table1.

method	LOPO		FORO	
	mean pose err.	max pose err.	mean pose err.	max pose err.
NMC	0.06(0.09)	0.18(0.35)	0.04(0.02)	0.09(0.10)
LDC	0.06(0.07)	0.14(0.35)	0.01(0.01)	0.04(0.05)
QDC	0.10(0.11)	0.23(0.34)	0.01(0.01)	0.04(0.06)
LDA+QDC	0.07(0.09)	0.16(0.35)	0.02(0.01)	0.04(0.06)
Parzen	0.07(0.09)	0.16(0.35)	0.01(0.01)	0.02(0.04)
LDA+Parzen	0.06(0.07)	0.14(0.35)	0.00(0.00)	0.01(0.03)

**Table1.** Cross-validation results of pose classifiers (mean errors with standard deviation).

As shown in Table.1, there is a clear separation between pre-defined poses. The differences between LOPO and FORO illustrate that the latter is highly optimistically biased. The reason is that similar examples extracted from neighbouring frames of one person may end up in both training and test set. It is also interesting to notice that this difference grows with classifier complexity, which is a clear sign of over-fitting. We observe that the simplest method (NMC) provides comparable performance to more complex classifiers which need an extra dimensionality reduction step to avoid the curse of dimensionality. We conclude that the extracted features are informative and do not require use of more complex classifiers.

We also calculate the confusion matrices of the 9-class pose classifier (NMC). The results are shown in Table2, which are the sum of 15 per-fold (person) LOPO confusion matrices.

		Estimated Labels								
		P1	P2	P3	P4	P5	P6	P7	P8	P9
True Labels	P1	198	0	0	0	0	0	0	0	0
	P2	0	193	0	0	0	0	0	0	0
	P3	2	0	157	0	0	0	0	0	0
	P4	0	0	0	159	0	20	0	0	0
	P5	1	0	1	0	164	0	2	0	0
	P6	2	3	6	0	0	129	0	0	0
	P7	0	0	1	0	3	0	164	0	0
	P8	0	0	9	0	6	0	1	162	0
	P9	0	0	5	3	0	0	0	0	133

**Table2.** Confusion matrices of nine poses.

As can be seen from Table.2, the results are promising. Most of the poses can be recognized very well. However there is quite a large error between pose4 and pose6 and all the 20 misclassified samples are from the same person. We searched back in the dataset and found that the 3D positions of the person's right hand in the 20 samples are not correct due to wrong detections. We conclude that the wrong representation/detection of the feature points is the reason for the misclassification.

## 6. CONCLUSION

In this paper, we present an approach to capture markerless human motions and recognize human poses. By transferring the 2D and 3D positions of the selected feature points into a normalized feature space, a simple classifier is shown to be sufficient for multi-pose recognition. This is also quite attractive from a computational point of view. The processing time of each frame is

0.047 seconds, including background subtraction, torso and hand detection, and pose recognition. However, due to the small number of the selected feature points, some errors are introduced in the pose classification. Therefore, in our future work, we will focus on extracting more relevant features to improve the performance of the classifier. Moreover, we will investigate detectors to reject non-pose examples based on the proposed features.

## 7. ACKNOWLEDGMENTS

This research has been supported by the GATE (Game Research for Training and Entertainment) project, funded by the Netherlands Organization for Scientific Research (NWO) and the Netherlands ICT Research and Innovation Authority (ICT Regie).

## 8. REFERENCES

- [1] T. B. Moeslund, A. Hilton and V. Kruger, "A survey of advances in vision-based human motion capture and analysis," in *Computer Vision and Image Understanding*, vol. 104, pp. 90-126, November 2006.
- [2] G. Rogez, J. Rihan, S. Ramalingam and etc, "Randomized trees for human pose detection," in *Proceedings IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, IEEE, June 2008.
- [3] R. Okad and S. Soatto, "Relevant feature selection for human pose estimation and localization in cluttered images," in *European Conference on Computer Vision*, 2008, vol. II, pp. 434-445.
- [4] N. Thome, D. Merad and S. Miguet, "Human body part labeling and tracking using graph matching theory," in *International Conference on Video and Signal Based Surveillance*. IEEE, 2006, pp. 38-43.
- [5] R. Hoshino and D. Arita, "Real-time human motion analysis based on analysis of silhouette contour and color blob," in *Lecture Notes in Computer Science*, 2002, pp. 92-103.
- [6] M. Singh, M. Mandal and A. Basu, "Visual gesture recognition for ground air traffic control using the radon transform," in *International Conference on Intelligent Robots and Systems*, 2005, pp. 2586-2591.
- [7] C. Stauffer and W. Grimson, "Adaptive background mixture models for real-time tracking," in *Proceedings IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, IEEE, 1999, vol. II, pp. 246-252.
- [8] A. Micilotta, "Detection and tracking of humans for visual interaction," in *PhD. Dissertation, School of Electronics and Physical Sciences, University of Surrey*, September 2005.
- [9] M. Isard and A. Blake, "CONDENSATION-Conditional density propagation for visual tracking," in *International Journal of Computer Vision*, vol. 29, pp. 5-28, January 1998.
- [10] F.M., Porikli and O. Tuzel, "Human body tracking by adaptive background models and mean-shift analysis," in *IEEE International Workshop on Performance Evaluation of Tracking and Surveillance*, March 2003.