

View-Invariant Full-Body Gesture Recognition from Video

Bo Peng^{1,2}, Gang Qian^{1,2} and Stjepan Rajko^{1,3}

¹*Arts Media & Engineering Program*

²*Department of Electrical Engineering*

³*Department of Computer Science and Engineering*

Arizona State University, Tempe, AZ 85287, U.S.A

{bo.peng.1, gang.qian, stjegan.rajko}@asu.edu

Abstract

In this paper, we propose a video-based full-body gesture recognition system independent of the view angle of the cameras. We performed multilinear analysis on the silhouette images of the static poses making up the gestures by tensor decomposition and projection. Each pair of silhouette images is projected to a view-invariant low dimensional pose coefficient vector space. These pose vectors are then used as input vectors in hidden Markov model (HMM) for gesture recognition. This system worked effectively in our experiments using real videos.

1. Introduction

Gesture-driven human computer interaction (HCI) has been an active topic in recent years. The key component of gesture-driven HCI is a gesture recognition system, which identifies the gestures to provide input to the decision system of HCI. Many methods have been developed for both hand gesture and full body gesture recognition [7]. These recognition tasks are challenging because both human hand and body are highly articulated.

In order to develop an efficient gesture recognition system, an important issue is to extract features to describe the highly articulated body. As one solution, marker positions extracted using infrared-reflective marker systems are utilized in many interactive environments, such as in [10]. Marker-based systems can reliably capture the 3D coordinates of the markers placed on bony landmarks of the body. However, wearing markers is cumbersome to the subject using the system. Additionally, commercial marker-based motion capture systems are very expensive. Therefore,

a video-based gesture recognition system is preferred for non-intrusive and low cost sensing.

In existing video-based gesture recognition systems, tracking certain “landmarks” in the motion images is a widely applied strategy to extract motion features. In [8], human hands and head are tracked using skin detector and face detector. In [16] and [4], some “visual interesting points” or “visual cues” are used to describe the motion in each image frame. Landmark-based feature extraction is prone to tracking failure in many cases, especially when the view angle of the camera changes, or equivalently the body orientation (facing direction) of the subject changes. In such cases, many landmarks could be occluded. The system described in [6] is landmark-free, but it is a view dependent system. There are some systems that perform view-invariant gesture recognition, such as in [2] and [15], but these systems mainly rely on obtaining 3D information of the subject using more complicated camera systems.

In this paper, we present a view-invariant video based full-body gesture recognition system that applies a simple camera system. Based on our previous work [9], we use multilinear analysis on the silhouette image of each captured video frame in order to obtain view-independent feature vectors for the static poses. Then these feature vectors are fed into a hidden Markov model (HMM) to perform gesture recognition. We have performed experiments on real video sequences to verify the efficacy of the system.

2. Overview of the Proposed System

In this paper, we propose a view-invariant video based full-body gesture recognition system which can be applied in gesture-driven HCI. To the best of our knowledge, our system is the first one addressing full-body human gesture recognition from video without the

recovery of body kinematics or the 3D volumetric reconstruction. In our system, the subject can preform the gestures in any body orientation (rotation angle about the axis perpendicular to the ground plane).

The system applies two uncalibrated wide baseline video cameras for image acquisition. These cameras are mounted at approximately half body height. Their looking directions are roughly parallel to the ground plane and orthogonal to each other.

The two cameras capture videos continuously. For each frame, a pair of images are acquired and their silhouettes are extracted. The pair of silhouettes are then analyzed in the multilinear framework, and a view-invariant pose coefficient vector can be calculated for each image frame. Using these pose coefficients as feature vectors, we apply HMM to perform gesture recognition. A block diagram of the proposed system is shown in Figure 1.

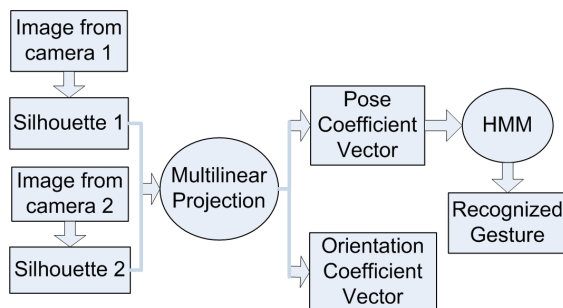


Figure 1. An overview of proposed gesture recognition system.

3. Acquiring View-invariant Pose Vectors via Multilinear Analysis

A gesture is made up of a series of static poses (articulated body shapes). In order to perform effective recognition of gestures, it is important to find a good descriptor of each static pose. The descriptor should be independent of the view angle of the camera. In our case, when cameras are fixed, the pose descriptor should be independent of the body orientation, as defined in Section 2. This descriptor can be obtained by performing multilinear analysis on a pair of silhouette images of the pose.

3.1. A Brief Introduction to Multilinear Analysis

Multilinear factorization has been used successfully in decomposing ensembles of static data such as im-

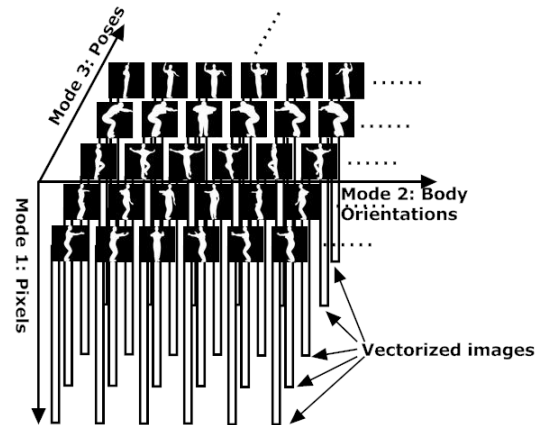


Figure 2. The Structure of TensorPose.

age and 3D volumetric data, into perceptually independent sources of variations. Previous successful applications include multifactor face image representation in the form of TensorFace [12], modeling of 3D face geometry [14], texture and reflectance [13], and image synthesis for articulated movement tracking[5]. The TensorFace framework [12] is a well known framework, which incorporates many factors that affect the resulting face image, such as facial geometry (different person), head pose, and illumination. With multilinear analysis by tensor decomposition, each of these affecting factors can be analyzed separately.

3.2. The TensorPose Framework

A set of silhouette images of a pose is mainly affected by three factors: body model of the subject (different subject), joint angle configuration of the subject (different poses) and the subject's body orientation about the camera system. The influence of body model to the images are comparatively small, and we can incorporate these differences by using different body models in training. In the proposed system, we mainly concentrate on the factors of pose and orientation. We developed a TensorPose framework which is similar to TensorFace.

In the TensorPose framework, the input data are pairs of silhouette images of the poses. Each image is normalized, and a pair of normalized images are then vectorized and concatenated to form a complete input vector. As shown in Figure 2, these input vectors span along the orientation mode and pose mode to form the training tensor.

3.3. Tensor Decomposition Using HOSVD

As in [1], we perform high order singular value decomposition (HOSVD) on the 3-mode training tensor.

In our TensorPose framework, we do not conduct dimension reduction in any of the modes. The tensor $\mathcal{A} \in \mathbb{R}^{N_i \times N_o \times N_p}$ can be decomposed into:

$$\mathcal{A} = \mathcal{D} \times_1 \mathbf{U}_i \times_2 \mathbf{U}_o \times_3 \mathbf{U}_p, \quad (1)$$

in which \mathcal{S} is the core tensor of the same size as \mathcal{A} . $\mathbf{U}_i \in \mathbb{R}^{N_i \times N_i}$, $\mathbf{U}_o \in \mathbb{R}^{N_o \times N_o}$, $\mathbf{U}_p \in \mathbb{R}^{N_p \times N_p}$ are orthogonal matrices representing respectively the image pixel mode, orientation mode and pose mode. N_i , N_o and N_p are respectively the dimensionalities in the three modes. In our approach, tensor decomposition is performed only in the second and third modes, and \mathbf{U}_i is an $N_i \times N_i$ identity matrix. Hence

$$\mathcal{A} = \mathcal{D} \times_2 \mathbf{U}_o \times_3 \mathbf{U}_p. \quad (2)$$

The decomposed tensor possesses the property as follows:

$$\mathcal{A}(:, i, j) = \mathcal{D} \times_2 \mathbf{u}_{o,i} \times_3 \mathbf{u}_{p,j}, \quad (3)$$

where $\mathcal{A}(:, i, j)$ stands for a vector in a tensor that contains the pixels of a pair of image of pose j in orientation i . $\mathbf{u}_{o,i}$ and $\mathbf{u}_{p,j}$ are respectively the i 'th row of \mathbf{U}_o and the j 'th row of \mathbf{U}_p . Alternatively speaking, $\mathbf{u}_{o,i}$ is the coefficient vector representing the i 'th orientation, and $\mathbf{u}_{p,j}$ is the coefficient vector representing the j 'th pose.

3.4. Projection of an Input Vector Using Core Tensor

Based on the property defined in (3), a new input vector \mathbf{z} can be projected to the pose coefficient space and orientation coefficient space by solving the bilinear problem defined as follows:

$$\mathbf{z} = \mathcal{D} \times_2 \mathbf{v}_o \times_3 \mathbf{v}_p, \quad (4)$$

where \mathbf{v}_o is the orientation coefficient vector and \mathbf{v}_p is the pose coefficient vector.

This problem can be solved using the iterative alternating least square (ALS) algorithm [3] as follows. Let the previous estimate of the orientation coefficient vector be $\hat{\mathbf{v}}_o^{(n)}$, and

$$\mathbf{C}_o^{(n)} = \mathcal{D} \times_2 \hat{\mathbf{v}}_o^{(n)}, \quad (5)$$

where \mathcal{D} is degenerated into a matrix $\mathbf{C}_o^{(n)}$ by multiplying a row vector. Inserting (5) into (4) we can get

$$\mathbf{z} = \mathbf{C}_o^{(n)} \mathbf{v}_p. \quad (6)$$

Thus, the current estimate of the pose coefficient vector $\mathbf{v}_p^{(n+1)}$ can be easily obtained by solving a linear equation (6).

Similarly, if the pose coefficient is known to be $\hat{\mathbf{v}}_p^{(n)}$, we can update the estimate of the orientation coefficient vector \mathbf{v}_o by solving the following equation.

$$\mathbf{z} = \mathbf{C}_p^{(n)} \mathbf{v}_o, \quad (\mathbf{C}_p^{(n)} = \mathcal{D} \times_3 \hat{\mathbf{v}}_p^{(n)}). \quad (7)$$

Given initial values of $\mathbf{v}_p^{(0)}$ or $\mathbf{v}_o^{(0)}$, we can solve both vectors by solving (6) and (7) alternately until the solution converges.

When applying ALS, different initial values of $\mathbf{v}_p^{(0)}$ or $\mathbf{v}_o^{(0)}$ may converge to different solutions. Since the ground truth of the body orientation angle should be close to one of the standard angles (e.g. when the number of orientations $N_o = 16$, the maximum deviation is $360/(16 \times 2) = 11.25^\circ$), one possible way to find a stable solution is to use each row vector in \mathbf{U}_o (standard orientation vectors) as the initial value of \mathbf{v}_o and obtain a set of candidate solutions. We can then choose our final solution to be the one with minimum reconstruction error.

However, applying multiple initial values is computationally expensive because the ALS procedure needs to be performed multiple times. In order to improve computational efficiency while maintaining the stability of the solution, we have developed an improved initialization method as follows. Firstly, we use all the row vectors $\{\mathbf{u}_{o,i}\}_{i=1}^{N_o}$ of \mathbf{U}_o as the initial values for \mathbf{v}_o . For each $\mathbf{u}_{o,i}$, we find the corresponding pose vector $\mathbf{v}_{p,i}$ by solving (6) *only once*. Then among $\{\mathbf{v}_{p,i}\}_{i=1}^{N_o}$, the pose vector that is the most similar to one of the standard poses $\{\mathbf{u}_{p,i}\}_{i=1}^{N_p}$ is chosen as the initial pose coefficient vector $\mathbf{v}_p^{(0)}$ to initialize ALS.

$$\mathbf{v}_p^{(0)} = \arg \max_{\mathbf{v}_{p,i}, j} \frac{\mathbf{v}_{p,i} \cdot \mathbf{u}_{p,j}}{\|\mathbf{v}_{p,i}\| \cdot \|\mathbf{u}_{p,j}\|}. \quad (8)$$

where $i = 1, \dots, N_o$ is the orientation index, and $j = 1, \dots, N_p$ the pose index. Since we choose only one initial value $\mathbf{v}_p^{(0)}$ to solve (4), the computational efficiency is high. The solutions obtained using this initialization method performed stably in the experiments.

By solving the bilinear equation (4), each input pair of images can be projected to a pose coefficient vector \mathbf{v}_p which is independent of body orientation. We use these view-invariant pose vectors as the feature vectors for gesture recognition.

4. Gesture Recognition using HMM

In our system, a 12-state left-to-right HMM is used for gesture recognition. As is typically done with HMMs in the context of gesture recognition, we train the model using the expectation maximization (EM) algorithm, but with one slight modification. A traditional

training approach would construct a probability distribution for each state that represents the set of observations (feature vectors) associated with it. The probability distribution would then be used in the inference phase to determine the probability that an observation of an unclassified gesture has been generated by the state. In our case, however, instead of using the associated feature vectors to train a probability distribution, we simply record the feature vectors with the state. Then, in the inference phase, instead of using the probability of a particular observation \mathbf{v}_o being generated by a state s_j , we utilize a similarity metric between the state and the observation. This similarity metric is based on a similarity metric between two feature vectors, whose definition follows.

We define the similarity metric between two feature vectors as

$$s = \frac{\mathbf{a}^T \mathbf{b}}{\|\mathbf{a}\| \|\mathbf{b}\|}, \quad (9)$$

and the similarity metric between an observation and a state was calculated as the maximum similarity between the observation and the feature vectors stored in a state. Therefore we can denote

$$p(\mathbf{v}_o | s_j) = \max_{\mathbf{x}_i \in s_j} \frac{\mathbf{v}_o^T \mathbf{x}_i}{\|\mathbf{v}_o\| \|\mathbf{x}_i\|}, \quad (10)$$

The full implementation of the described approach is released under an open source license in the AME Patterns library [11].

5. Experiments

5.1. The Gesture Set and Key Poses

Table 1. The list of 6 gestures used in the experiment.

Gesture No.	Description
1	Pushing
2	Spreading right arm
3	Enclosing arms and crouch
4	Doing “cactus”
5	Retreating right arm from side
6	Lifting left arm

In this paper, the gesture set we applied consists of six gestures choreographed by a professional dancer for an interactive environment. These gestures are listed in Table 1. We manually selected 28 key poses from the starting stages, ending stages and important intermediate stages of these gestures. The key poses are shown in Figure 3.

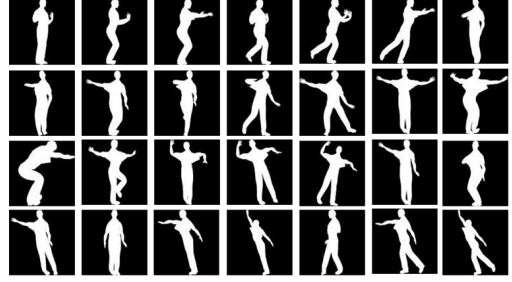


Figure 3. The key poses selected from the gestures.

Table 2. Gesture recognition results.

Gesture No.	N_e	R_r	R_{fa}
1	0	100%	0.31%
2	0	100%	0%
3	0	100%	0%
4	2	96.87%	0%
5	0	100%	0%
6	0	100%	0.31%
Overall	2	99.48%	0.12%

5.2. Formation of the Training Tensor

In order to form the training tensor of our framework, the silhouette images of each key pose performed in 16 orientations (evenly divided in a circle) are synthesized using Maya. Each image is normalized, resized to 50 by 50 and then vectorized to form the input vectors in the training tensor.

5.3. Gesture Recognition Results

In our experiment, each gesture in the set defined above was performed in two slightly different versions to train and test the proposed system. Each version was performed 40 times and the subject is free to rotate about the axis perpendicular to the ground before performing each trial to verify the view-invariance of the system. We took the video data of the first 16 trails of each gesture, 8 in each version, as training set. The data of remaining 64 trails of each gesture were used as testing data. The testing results are shown in Table 2, in which N_e is the number of missed gesture samples, R_r is the recognition rate and R_{fa} the false alarm rate.

5.4. Comparison with Existing Gesture Recognition Systems

To the best of our knowledge, there are very few existing video based systems that perform view-invariant full body gesture recognition. We have compared our system with some existing systems performing view-dependent gesture recognition [4, 6, 16], partial-body gesture recognition [2] or performing full body gesture using 5 calibrated cameras [15]. The comparison of recognition rates R_r and false alarm rates R_{fa} is listed in Table 3. It can be shown that our system outperforms the existing systems.

Table 3. Comparison of performance with existing gesture recognition systems.

System	R_r			R_{fa}
	Highest	Lowest	Overall	
Kirishima [4]	100%	60%	83%	N.A.
Ye [16]	96.8%	86.8%	92.6%	N.A.
Lee [6]	100%	90%	97.4%	N.A.
Holte [2]	88.3%	78.3%	82.9%	2.08%
Weinland [15]	100%	80%	93.3%	N.A.
The Proposed System	100%	96.87%	99.48%	0.12%

6. Conclusion and Future Work

In this paper, a video based full body gesture recognition system is presented. By applying view-invariant pose coefficient vectors as feature vectors, this system is independent of the body orientation of the subject performing the gesture with respect to the camera system. Experiments has shown promising performance of our system.

In the future, we can apply this system in many interactive environments which uses full body gestures as control signals. Since our system requires only a simple set of video cameras for motion capture, it can greatly reduce the cost of such systems.

7. Acknowledgement

This paper is based upon work partly supported by U.S. National Science Foundation on CISE-RI no. 0403428 and IGERT no. 0504647. Any opinions, findings and conclusions or recommendations expressed in

this material are those of the authors and do not necessarily reflect the views of the U.S. National Science Foundation (NSF).

References

- [1] L. Elden. *Matrix Methods in Data Mining and Pattern Recognition*. SIAM, Philadelphia, 2007.
- [2] M. Holte and T. Moeslund. View invariant gesture recognition using 3d motion primitives. In *Proc. ICASSP*, pages 797 – 800, 2008.
- [3] H. A. L. Kiers. An alternating least squares algorithms for parafac2 and three-way dedicom. *Computational Statistics & Data Analysis*, 16(1):103 – 118, 1993.
- [4] T. Kirishima, K. Sato, and K. Chihara. Real-time gesture recognition by learning and selective control of visual interest points. *Pattern Analysis and Machine Intelligence*, 27(3):351–364, 2005.
- [5] C.-S. Lee and A. Elgammal. Modeling view and posture manifolds for tracking. In *Proc. ICCV*, pages 1–8, 2007.
- [6] S.-W. Lee. Automatic gesture recognition for intelligent human-robot interaction. In *Proc. FGR*, pages 645–650, 2006.
- [7] S. Mitra and T. Acharya. Gesture recognition: A survey. *Systems, Man, and Cybernetics, Part C: Applications and Reviews*, 37(3):311–324, 2007.
- [8] H. S. Park, D. J. Jung, and H. J. Kim. Vision-based game interface using human gesture. In *Advances in Image and Video Technology*, pages 662–671. Springer, Berlin/Heidelberg, 2006.
- [9] B. Peng and G. Qian. Binocular dance pose recognition and body orientation estimation via multilinear analysis. In *Proc. CVPR Workshops*, pages 1–8, 2008.
- [10] G. Qian, F. Guo, T. Ingalls, L. Olson, J. James, and T. Rikakis. A gesture-driven multimodal interactive dance system. In *Proc. ICME*, pages 1579–1582, 2004.
- [11] S. Rajko. Ame patterns library [computer software]. <http://ame4.hc.asu.edu/amelia/patterns/>, 2008.
- [12] M. A. O. Vasilescu and D. Terzopoulos. Multilinear analysis of image ensembles: Tensorfaces. In *Proc. ECCV*, pages 447–460, 2002.
- [13] M. A. O. Vasilescu and D. Terzopoulos. Tensortextures: Multilinear image-based rendering. *ACM Transactions on Graphics*, 23(3):334–340, 2004.
- [14] D. Vlasic, M. Brand, H. Pfister, and J. Popovi. Face transfer with multilinear models. In *Proc. ACM SIGGRAPH*, pages 426 – 433, 2005.
- [15] D. Weinland, R. Ronfard, and E. Boyer. Free viewpoint action recognition using motion history volumes. *Computer Vision and Image Understanding*, 104(2-3):249–257, 2006.
- [16] G. Ye, J. J. Corso, D. Burschka, and G. D. Hager. Vics: A modular hci framework using spatiotemporal dynamics. *Machine Vision and Applications*, 16(1):13–20, 2004.