Invariant Human Pose Feature Extraction for Movement Recognition and Pose Estimation

by

Bo Peng

A Dissertation Presented in Partial Fulfillment
of the Requirements for the Degree
Doctor of Philosophy

Approved April 2011 by the
Graduate Supervisory Committee:

Gang Qian, Chair
Jieping Ye
Baoxin Li
Andreas Spanias

ARIZONA STATE UNIVERSITY

August 2011

ABSTRACT

Reliable extraction of human pose features that are invariant to view angle and body shape changes is critical for advancing human movement analysis. In this dissertation, the multifactor analysis techniques, including the multilinear analysis and the multifactor Gaussian process methods, have been exploited to extract such invariant pose features from video data by decomposing various key contributing factors, such as pose, view angle, and body shape, in the generation of the image observations. Experimental results have shown that the resulting pose features extracted using the proposed methods exhibit excellent invariance properties to changes in view angles and body shapes. Furthermore, using the proposed invariant multifactor pose features, a suite of simple while effective algorithms have been developed to solve the movement recognition and pose estimation problems. Using these proposed algorithms, excellent human movement analysis results have been obtained, and most of them are superior to those obtained from state-of-the-art algorithms on the same testing datasets. Moreover, a number of key movement analysis challenges, including robust online gesture spotting and multi-camera gesture recognition, have also been addressed in this research. To this end, an online gesture spotting framework has been developed to automatically detect and learn non-gesture movement patterns to improve gesture localization and recognition from continuous data streams using a hidden Markov network. In addition, the optimal data fusion scheme has been investigated for multi-camera gesture recognition, and the decision-level camera fusion scheme using the product rule has been found to be optimal for gesture recognition using multiple uncalibrated cameras. Furthermore, the challenge of optimal camera selection in multi-camera gesture recognition has also been tackled. A measure to quantify the complementary strength across cameras has been proposed. Experimental results obtained from a real-life gesture recognition dataset have shown that the optimal camera combinations identified according to the proposed complementary measure always lead to the best gesture recognition results.

To my parents and my wife.

ACKNOWLEDGEMENTS

First and foremost, I would like to thank my advisor, Prof. Gang Qian, for his support and guidance throughout my graduate study. He trained me how to conduct research rigorously, and his insightful suggestions often helped me a lot when I faced difficulties in my research projects. He also provided me with great help on academic writing and presentation.

Many thanks to Prof. Todd Ingalls, my application advisor, for giving me advice on applying my research findings to real world problems.

I would also like to thank Prof. Andreas Spanias, Prof. Jieping Ye, and Prof. Baoxin Li for serving on my committee and attending my qualifying exam, comprehensive exam and final defense.

Also, I would like to thank the School of Arts, Media & Engineering for providing me with the advanced experimental platforms and equipments for my research, which helped me greatly in collecting data for my projects. Special thanks to Mr. Assegid Kidane and Ms. Kelly Phillips for setting up experimental hardware for my projects and assisting me with related technical issues, and to Prof. Loren Olson for assisting me with software issues in my experiments.

Thanks to all the ASU faculty who instructed me in my graduate courses for equipping me with advanced knowledge in various fields.

Thanks to Dr. Stjepan Rajko, Dr. Huan Jin and Dr. Feng Guo, whose prior work in our lab served as great references for my research.

Finally, I would like to thank Mr. Jiqing Zhang, Mr. Yangzi Liu and all other collaborators in my research projects.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

Chapter 1

INTRODUCTION

1.1 Background and Motivations

Movement-driven human computer interaction (HCI) systems have received considerable attention in the past decade. Such systems allow users to communicate with computers through movements in a much more intuitive and natural manner than the traditional human-computer interfaces based on mouse clicks and keystrokes. Embodied HCI systems have many important applications, including immersive virtual reality systems [3] such as GrImage [4], augmented reality [5–7], industrial control [8], human-robot interaction [9, 10], health care and cognitive assistance [11, 12], sign language analysis [13], computer games [14], and interactive media systems [15–18].

There are mainly two types of communication cues applied in movement-driven HCI systems, namely, static body poses and continuous body gestures. To enable machines to read and understand these two types of communication cues is critical for developing such HCI systems. Machines mainly understand movement cues by recognizing the type of pose or gesture a subject is performing, or by estimating the joint angle or joint position of the subject. Therefore, pose recognition, gesture recognition and pose estimation are three important problems in movement sensing.

Movement sensing systems using markers or other wearable sensors such as the inertial sensors (accelerometers, gyros, and magnetometers) [19] can be used in pose recognition, gesture recognition and pose estimation. (e.g., [20–23]). However, such movement sensing techniques are intrusive and they require placing passive (e.g., retro-reflective markers for optical motion capture) or active (e.g., LED markers and inertial sensors) sensors on the subject's body to collect movement data. Placing additional sensors on the subject is cumbersome and time-consuming and it may also restrain the subject's movement. Therefore, it is preferred to develop nonintrusive gesture recognition systems without placing sensors on the subject. To this end, video-based sensing techniques have been widely applied to nonintrusive movement analysis.

Video-based movement sensing systems need to process streams of images obtained by video cameras. The dimensionality of an image is usually very high. In addition, in most cases, much of the data captured within a movement image is redundant. Therefore, to extract movement features from images is essential.

1.2 Existing Video-Based Movement Feature Extraction Methods

Existing video-based movement feature extraction methods can generally be categorized into the global methods and the local methods. The global methods treat an image, or a 3D shape obtained from images using the shape-from-silhouette techniques [24], as a whole when extracting movement features. One common global feature extraction approach is to embed high-dimensional image data into a low-dimensional feature space using linear or nonlinear methods. For example, in [25], the principal component analysis (PCA) has been applied for linear low-dimensional embedding for movement feature extraction. Furthermore, in [26], the locality preserving projections (LPP) method has also been used in movement feature extraction. In [27], the authors have applied the local linear embedding (LLE) method to extract movement features. In [28], high-dimensional visual inputs are also embedded in a nonlinear manifold. Additional examples of low-dimensional embedding include discriminant embedding [29] and kernel based embedding [30]. Another type of global features are statistical features. Statistical moments such as the Hu moments [31] and the Zernike moments [32] are typical examples of this type of feature. Also, the histogram-based 2D and 3D shape contexts are applied respectively in [33] and [34].

In contrast to the global methods, the local methods focus on finding local features in images or 3D shapes. One type of local features are body landmarks. For example, in [35–37], positions of selected joints are tracked as movement features. Another type of local features are salient points. One example of such features is the Harris corners [38], which are the points where a shift in any direction will result in large pixel value changes. In [39], the scale invariant feature transform (SIFT) features have been developed by finding the extrema from the differences of images convolved by Gaussian filters with different scales. Speeded up robust features (SURF) [40] is an improved version of the SIFT features. In [41], the radial gradient transform is

applied for fast feature extraction. In [42], the linear discriminant analysis (LDA) has been used to extract discriminative local features through postprocessing. Besides local features in 2D images, spatial-temporal features can also be extracted from video streams. For example, in [43] and [44], the Harris corners are detected in three orthogonal planes of a video volume. In [45], the formulation of the Harris corner detector is extended to 3D. In [46] and [1], features have been detected by convolving the video volume with a combined filter. The Gaussian filter is applied in the spatial domain and Gabor filter in the temporal domain. In [47] local steering kernels are applied for spatial-temporal feature extraction. Also, in [48] a combination of optical flow and spatial gradient are applied in feature extraction.

1.3 Invariant Pose Feature Extraction

It is important to extract pose features that are invariant to other factors, such as the view angle of the camera and the body shape of the subject. View-invariance is an important factor for video-based pose and gesture recognition in many HCI applications. It requires the recognition system to identify poses and gestures invariant to changes in the camera view angle. When cameras are fixed, this is equivalent to recognition invariant to the orientation of the subject with respect to the camera system. Many existing pose recognition [49, 50] and gesture recognition [51–57] methods are view-dependent, i.e., assuming that the relative torso orientation with respect to the cameras is known. While it is a valid assumption in some scenarios, such as automatic sign language interpretation, having to know body orientation with respect to the camera presents an undesirable constraint which hampers the flexibility, and sometimes, the usability of an HCI system in applications such as interactive dance [18] and embodied learning [58]. In these applications, it is preferable that a gesture can be recognized from any view point with respect to the cameras so that the subject can freely move and orient in the space. The extraction of pose features invariant to body shape is also important. Tests using two challenging gesture recognition datasets, the KTH dataset [59] and the IXMAS dataset [60], reveal that it is challenging to obtain a gesture recognition rate higher than 95%. One of the reasons is that these two datasets contain gestures performed by multiple subjects with quite different body shapes.

Using the existing global and local feature extraction methods, extracting pose features invariant to the changes in the view angle and body shape is challenging. Body kinematic features, such as joint angles and joint positions, are invariant to view angle and not sensitive to body shape variation. Several kinematic-based gesture recognition approaches are also view-invariant, based on tracking landmark points or joint angles. For example, Shen and Foroosh [36, 61, 62] have applied the invariance of homography and fundamental ratio of joint point triplets in view-invariant pose recognition and action recognition. In [63], curvature of the trajectory of a single point was applied. In [37], the "area-cross-ratio" computed from joint points was applied in action recognition. Unfortunately, such methods are subject to tracking failures due to self-occlusion especially for complicated movements.

Besides recovering body kinematics, another strategy for achieving view-invariance has been taken by extracting view-invariant pose features from volumetric reconstruction of the subject from multiple views. Once extracted, view-invariant features can be directly used to match the input 3D volumetric data and the training templates without additional view alignment. To extract view-invariant features, the volumetric data is first transformed into certain alternative representations, such as the 3D shape context (e.g., in [34, 64]) and the 3D motion history volume (MHV) (e.g., in [60]). Data points in these alternative 3D representations can be indexed in a body-centered cylindrical coordinate system using (h, r, θ) coordinates, which respectively correspond to the height, radius and angular location of the data point. The h-axis of the body-centered coordinate system coincides with the vertical central axis of the subject. To further obtain view-invariant features, the angular dimension θ is suppressed in the feature extraction process so that the final extracted feature is independent of the data point distribution in θ. To achieve this goal, in [34, 60, 64], data points are first grouped into rings centered at and orthogonal to the h-axis so that data points on the same ring correspond to different θ, while sharing the same h and r. Then a ring-based feature is extracted from each ring and these ring-based features of all the rings constitute the final $\theta-$independent feature vector of the input volumetric data. A number of methods have been used to obtain such ring-based features. In the case of 3D shape context [34, 64], the sum of the bin values on a ring is taken as the corresponding ring-based feature. Similarly, in the case of 3D MHV [60], the Fourier transform of the data points along a ring is first computed and then the sum of their Fourier magnitudes is

taken as the ring-based feature. Pose features extracted using these methods are view-invariant since the orientation of the subject no longer affects the extracted features. However, on the other hand, suppression of the angular dimension may cause information loss and introduce ambiguity in gesture recognition. Note that in [65,66], another invariant human pose descriptor based on 3D shape context has also been introduced and used for pose classification [65] and action clustering [66]. This pose descriptor is obtained as the weighted average of a number of local 3D shape context features centered at sample points from a reference visual hull. This pose descriptor has a good invariance property for translation and scaling. However, it is only possibly invariant to rotation [66] and it is unclear to what extent this descriptor is invariant to rotation. To summarize, reliable extraction of invariant pose features remains a challenge because kinematic values are hard to track and summing up data along rings causes loss of information.

1.4 The Proposed Approaches

In this dissertation, two approaches to invariant pose feature are proposed. An important principle in both approaches is to model body pose, view angle (or body orientation when cameras are static) and body shape as separate contributing factors to observations of poses. By multilinear analysis, pose observations obtained with video cameras are projected onto independent low-dimensional pose feature spaces. Using the multifactor Gaussian process (MGP) model, a mapping is also modeled as from separate latent spaces to pose observation space. Invariant pose representation is achieved by inferring the latent variable in pose feature space.

After invariant pose features are obtained, pose recognition can be achieved using the support vector machine (SVM). To address the problem of pose recognition, SVM classifiers are applied in a one-versus-the-rest manner. For each pose in the vocabulary, a binary SVM classifier is trained. Recognizing a new pose involves traversal of all classifiers to find the best pose class to which it belongs.

Also using extracted pose features, the hidden Markov models (HMMs) are applied for pre-segmented gesture recognition. To tackle this problem, an HMM is applied to model each of the gestures in the gesture vocabulary. An unknown gesture, represented as a sequence of pose

feature vectors, can be classified to the gesture class whose corresponding HMM yields the maximum likelihood of the feature sequence.

Based on the method of pre-segmented gesture recognition, online gesture spotting has also been achieved using a network of HMM models. In the HMM network, not only gesture models but also specific non-gesture models have been incorporated to represent both gesture and non-gesture movement patterns. An approach to detect and model non-gestural movement patterns automatically from continuous training data has been developed. As shown in our experimental results, using specific non-gesture models significantly improves gesture spotting by reducing false alarm rates and increasing the recognition reliability, without significantly sacrificing the recognition rates.

From the viewpoint of sensor fusion, different strategies are explored for multiple camera fusion for video-based gesture recognition. Using the presented gesture recognition framework based on invariant pose features and HMM as a benchmark gesture framework, the performances of data-level, feature-level and decision-level fusion in gesture recognition are systematically analyzed. For a fixed set of cameras, the performances of different fusion strategies have been analyzed. The optimal configuration for a fixed number of cameras has also been analyzed.

Finally, the invariant pose features are applied for pose estimation. In order to estimate joint positions of the subject when performing the pose, a mapping from pose features to joint positions has been established using the relevance vector machine (RVM).

1.5 Contributions

This dissertation presents the research on invariant pose feature extraction and its applications in movement recognition and pose estimation. Obtaining invariant pose features is fundamental to this research. In order to extract invariant pose features from pose observations, two multifactor analysis techniques, the multilinear analysis and the multifactor Gaussian process (MGP), are exploited. The resulting invariant pose features are then used for pose recognition using SVM and gesture classification using HMM. A framework of online gesture spotting is

further developed using HMM networks. In addition, different multi-camera fusion techniques for gesture recognition are explored from sensor fusion point of view. Finally, invariant pose features are applied in pose estimation using RVM.

The major contributions of this dissertation are as follows. Firstly, two methods of invariant pose feature extraction were developed based on multilinear analysis and MGP. Part of the results of invariant feature extraction have been archived in the following publication.

- Bo Peng, Gang Qian, Yunqian Ma and Baoxin Li, ***Multifactor feature extraction for human movement recognition*** , Computer Vision and Image Understanding. vol. 115, no. 3, pp. 375 - 389, 2011

Secondly, invariant pose features are successfully applied for pose recognition using SVM. Some pose recognition results have been presented in the following publications.

- Bo Peng and Gang Qian, ***Binocular Dance Pose Recognition and Body Orientation Estimation via Multilinear Analysis***, in proceedings of IEEE International Conference on Computer Vision and Pattern Recognition (CVPR) workshops, 2008
- Bo Peng , Gang Qian and Yunqian Ma, ***View-Invariant Pose Recognition Using Multilinear Analysis and the Universum***, in proceedings of International Symposium on Visual Computing (ISVC), 2008
- Bo Peng and Gang Qian, ***Binocular Full-Body Pose Recognition and Orientation Inference Using Multilinear Analysis***, in Tensors in Image Processing and Computer Vision, pp. 215 - 236, Springer London, May 2009
- Bo Peng and Gang Qian and Yunqian Ma, ***Recognizing Body Poses using Multilinear Analysis and Semi-Supervised Learning***. Pattern Recognition Letters. vol. 30, no. 14, pp. 1289 - 1294 , 2009

Thirdly, invariant pose features are applied in gesture recognition using HMM and in pose estimation using RVM. Results of gesture recognition using the proposed framework have been presented in the following publications.

- Bo Peng, Gang Qian and Stjepan Rajko, ***View-Invariant Full-Body Gesture Recognition from Video***, in proceedings of International Conference on Pattern Recognition, 2008

- Bo Peng, Gang Qian and Stjepan Rajko, ***View-Invariant Full-Body Gesture Recognition via Multilinear Analysis of Voxel Data***, in proceedings of International Conference on Distributed Smart Cameras, 2009

In the second publication listed above, this gesture recognition method was tested on the challenging IXMAS data set and our results are state of the art.

Fourthly, an online gesture spotting framework was developed based on invariant pose features and HMM network. A method was developed to automatically detect non-gesture patterns and train non-gesture models. Results of online gesture spotting have been presented in the following paper.

- Bo Peng and Gang Qian, ***Online Gesture Spotting From Visual Hull Data***, in IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI), vol. 33, no. 6, pp. 1175 - 1188, 2011.

Finally, different multi-camera fusion techniques for gesture recognition have been explored systematically. The optimal camera fusion scheme for uncalibrated cameras has been identified. A cross-camera complementary measure and an incremental camera selection scheme have also been developed and verified. The relevant research and experimental results have been reported in the following submitted paper.

- Gang Qian and Bo Peng, ***Multi-Camera Fusion for Gesture Recognition***, submitted to IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI), 2011.

Chapter 2

INVARIANT POSE FEATURE EXTRACTION

The observation of a human pose is mainly affected by three factors: body shape of the subject, joint angle configuration of the subjects (different poses) and view angle (or body orientation of the subject if cameras are fixed).

In this chapter, approaches to extraction of pose features invariant to the other two factors are discussed. This problem can be modeled in two ways. Firstly, we can try to minimize the effect body shape by normalization and focusing on extracting view-invariant pose features. We can also try to extract pose features invariant to both view (orientation) and body shape. In both models, a key point is to separate the contributions of body pose and other factors to an observation of a pose. This issue can be addressed by applying multilinear analysis and multi-factor Gaussian Process (MGP) models. Using these two methods, invariant pose features are successfully extracted.

2.1 Invariant Pose Feature Extraction Using Multilinear Analysis

In this section, an approach to extraction of invariant pose features using multilinear analysis is discussed. The method of extracting view-invariant pose features is first discussed. Then, it is extended to extracting pose features invariant to both view and body shape.

2.1.1 An Introduction to Multilinear Analysis

In multilinear analysis, tensors are applied as the representations of multi-mode data collections. As introduced in [67], a tensor, also known as n-way array or multidimensional matrix or n-mode matrix, is a higher order generalization of a vector (1-mode tensor) and a matrix (2-mode tensor). High-order tensors can represent a collection of data in a more complicated way. When a data vector is determined by a combination of m factors, the collection of the data vectors can be represented as an $(m+1)$-mode tensor $\mathcal{T} \in \mathbb{R}^{N_v \times N_{f1} \times N_{f2} \cdots \times N_{fm}}$, in which N_v is the dimensionality of the data vector and N_{fi}, $(i = 1, 2, \ldots, m)$ is the number of possible values

of the *i*th contributing factor.

A tensor can be unfolded into a matrix along each mode. The mode-j unfolding matrix of a tensor $\mathcal{A} \in \mathbb{R}^{N_1 \times N_2 \cdots \times N_n}$ is denoted as $\mathbf{A}_{(j)}$, and $\mathbf{A}_{(j)} \in \mathbb{R}^{N_j \times (N_1 \ldots N_{j-1} N_{j+1} \ldots N_n)}$. An illustration of the unfolding of a 3-mode tensor is shown in Figure 2.1.

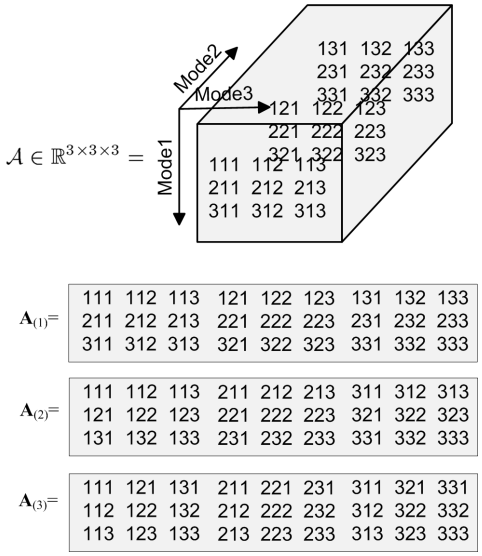


Fig. 2.1. Unfolding of a 3-mode tensor.

As an analogue to matrix-matrix multiplication, an n-mode tensor can be multiplied by compatible matrices in each mode. The mode-j multiplication of an n-mode tensor $\mathcal{A} \in \mathbb{R}^{N_1 \times N_2 \cdots \times N_n}$ with a matrix $\mathbf{M} \in \mathbb{R}^{N'_j \times N_j}$ can be denoted as $\mathcal{R} = \mathcal{A} \times_j \mathbf{M}$, $\mathcal{R} \in \mathbb{R}^{N_1 \times \cdots \times N_{j-1} \times N'_j \times N_{j+1} \times \cdots \times N_n}$. The entries of $\mathcal{R}$ are defined as

$$\mathcal{R}_{i_1,\ldots,i_{j-1},i,i_{j+1}\ldots,i_n} = \sum_{k=1}^{N_j} \mathcal{A}_{i_1,\ldots,i_{j-1},k,i_{j+1},\ldots,i_n} \mathbf{M}_{i,k}, \tag{2.1}$$

in which $i = 1, 2, \ldots, N'_j$. The unfolded matrix representation of mode-j multiplication of $\mathcal{A}$ by $\mathbf{M}$ is as follows.

$$\mathbf{R}_{(j)} = \mathbf{M}\mathbf{A}_{(j)}. \tag{2.2}$$

As an example, the multiplication of a 3-mode tensor and a matrix in each mode is illustrated in Figure 2.2 (a). When a 3-mode tensor is multiplied by a compatible row vector, it will degenerate into a matrix, as illustrated in Figure 2.2 (b).

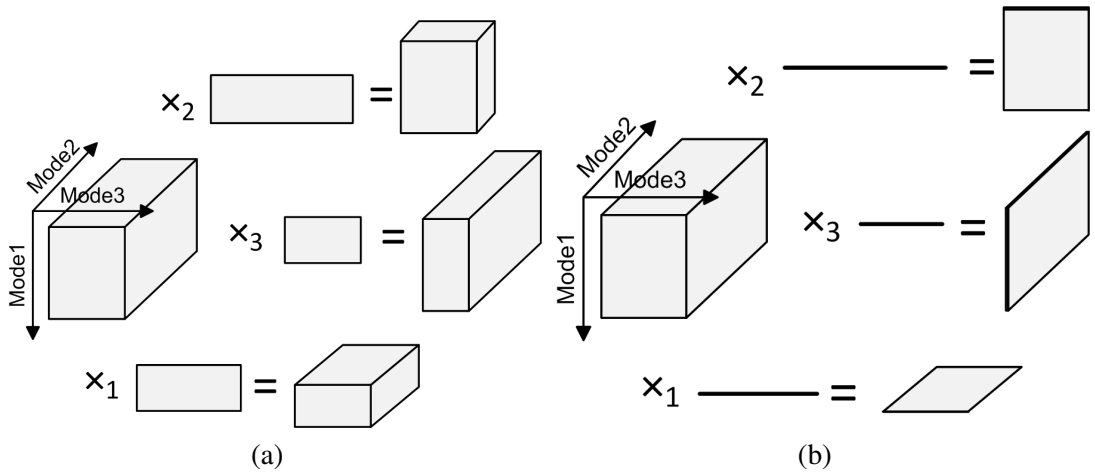


Fig. 2.2. Multiplication of a 3-mode tensor with a matrix (a) and a vector (b).

As a generalization of singular value decomposition (SVD) on matrices, we can also perform high order singular value decomposition (HOSVD) [68] on tensors. A tensor $\mathcal{A} \in \mathbb{R}^{N_1 \times N_2 \cdots \times N_n}$ can be decomposed into

$$\mathcal{A} = \mathcal{S} \times_1 \mathbf{U}_1 \times_2 \mathbf{U}_2 \cdots \times_n \mathbf{U}_n, \tag{2.3}$$

where $\mathbf{U}_j \in \mathbb{R}^{N_j \times N'_j}$ $(N'_j \leq N_j)$ are mode matrices containing orthonormal column vectors which are analogous to the left and right matrices in SVD. $\mathcal{S} \in \mathbb{R}^{N'_1 \times N'_2 \cdots \times N'_n}$ is called the core tensor which is analogous to the diagonal matrix in SVD.

In order to calculate mode matrices $\mathbf{U}_j\,(j = 1 \ldots n)$, we can first calculate the SVD of the unfolding matrix $\mathbf{A}_{(j)}$. Then $\mathbf{U}_j$ can be obtained by taking the columns of the left matrix of the SVD of $\mathbf{A}_{(j)}$ corresponding to the N'_j largest singular value. Then, the core tensor $\mathcal{S}$ can be calculated as follows.

$$\mathcal{S} = \mathcal{A} \times_1 \mathbf{U}_1^T \times_2 \mathbf{U}_2^T \cdots \times_n \mathbf{U}_n^T. \tag{2.4}$$

Denote $\mathbf{u}_{j,k}$ to be the k'th row vector of matrix $\mathbf{U}_j$, then the decomposed tensor possesses the property as follows [69].

$$\mathcal{A}_{i_1,i_2,\ldots,i_n} = \mathcal{S} \times_1 \mathbf{u}_{1,i_1} \times_2 \mathbf{u}_{2,i_2} \cdots \times_n \mathbf{u}_{n,i_n}. \tag{2.5}$$

Denote $\mathcal{A}(i_1,\ldots,i_{j-1},:,i_{j+1},\ldots i_n)$ to be the column vector containing the elements of $\mathcal{A}_{i_1,\ldots,i_j,\ldots,i_n}, i_j = 1\ldots N_j$. Then we can also get

$$\begin{aligned}&\mathcal{A}(i_1,\ldots,i_{j-1},:,i_{j+1},\ldots i_n)\\ &= \mathcal{S}\times_j \mathbf{U}_j \times_1 \mathbf{u}_{1,i_1} \times_2 \mathbf{u}_{2,i_2} \cdots \times_{j-1} \mathbf{u}_{j-1,i_{j-1}}\\ &\quad \times_{j+1} \mathbf{u}_{j+1,i_{j+1}} \cdots \times_n \mathbf{u}_{n,i_n}.\end{aligned} \tag{2.6}$$

In this way, $\mathcal{A}(i_1,\ldots,i_{j-1},:,i_{j+1},\ldots i_n)$ can be represented as a multilinear combination of the column vectors $(\mathcal{S}\times_j \mathbf{U}_j)(i_1,\ldots,i_{j-1},:,i_{j+1},\ldots i_n)$, $i_j = 1\ldots N_j$, $j = 1\ldots n$. The coefficients $\mathbf{u}_{k,i_k}$ in each mode can be considered as independent factors contributing to the data vector, and the interaction of these factors are governed by the tensor $\mathcal{S}\times_j \mathbf{U}_j$.

The mode matrices obtained from HOSVD can be considered as independent contributing factors, and the interaction of these factors are governed by the core tensor. Using this property, multilinear factorization has been used successfully in decomposing ensembles of static data such as image and 3D volumetric data, into perceptually independent sources of variations. Previous successful applications include multifactor face image representation in the form of TensorFace [67], modeling of 3D face geometry [70], texture and reflectance [71], and image synthesis for articulated movement tracking [72]. The TensorFace framework [67] is a well known framework, which incorporates many factors that affect the resulting face image, such as facial geometry (different persons), head pose, and illumination. With mutilinear analysis by tensor decomposition, each of these affecting factors can be analyzed separately.

2.1.2 Extraction of View-Invariant Pose Features

In order to extract view-invariant pose features using multilinear analysis, the first step is to form a pose tensor.

A 3-mode pose tensor can be formed using the observation vectors of a set of key poses in different orientations, as shown in Figure 2.3. Details on key pose selection and obtaining corresponding observation vectors can be found in the experimental section of this chapter. These observation vectors are centered by subtracting the mean value of all vectors. In this way, the binary (0 or 1) values of the vector elements become relatively continuous values

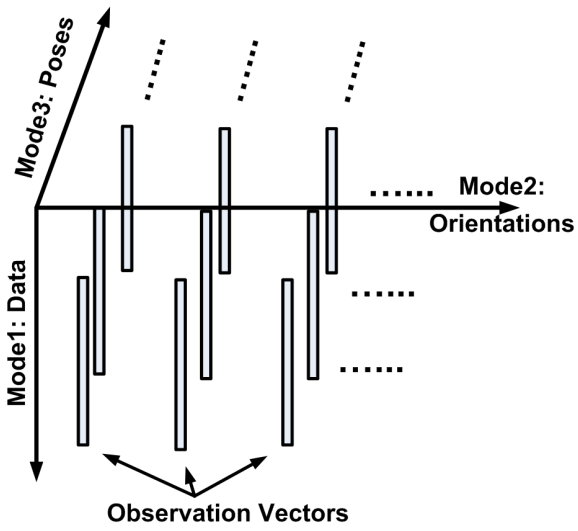


Fig. 2.3. The structure of the pose tensor

ranging from -1 to 1. Denote N_d to be the number of elements in an observation vector, N_o the number of body orientations and N_p the number of key poses, then the dimension of the pose tensor is $\mathcal{A}$ is $N_d \times N_o \times N_p$.

As described in the previous subsection, we can perform HOSVD on the 3-mode pose tensor to extract the core tensor and mode matrices. In the framework discussed in this chapter, dimension reduction is not conducted in any of the modes. According to (2.3), the training tensor $\mathcal{A} \in \mathbb{R}^{N_d \times N_o \times N_p}$ can be decomposed into:

$$\mathcal{A} = \mathcal{S} \times_1 \mathbf{U}_d \times_2 \mathbf{U}_o \times_3 \mathbf{U}_p, \tag{2.7}$$

in which $\mathcal{S}$ is the core tensor of the same size as $\mathcal{A}$. $\mathbf{U}_d \in \mathbb{R}^{N_d \times N_d}$, $\mathbf{U}_o \in \mathbb{R}^{N_o \times N_o}$, $\mathbf{U}_p \in \mathbb{R}^{N_p \times N_p}$ are orthogonal matrices representing respectively the data mode, the orientation mode and the pose mode. In order to extract orientation-invariant pose features, only the mode matrices $\mathbf{U}_o$ and $\mathbf{U}_p$ need to be calculated, and $\mathbf{U}_d$ can be combined with the core tensor $\mathcal{S}$. Denote

$$\mathcal{D} = \mathcal{S} \times_1 \mathbf{U}_d. \tag{2.8}$$

Then we can get

$$\mathcal{A} = \mathcal{D} \times_2 \mathbf{U}_o \times_3 \mathbf{U}_p. \tag{2.9}$$

The decomposed tensor possesses the following property.

$$\mathcal{A}(:,i,j) = \mathcal{D} \times_2 \mathbf{u}_{o,i} \times_3 \mathbf{u}_{p,j}, \tag{2.10}$$

where $\mathcal{A}(:,i,j)$ stands for an observation vector of pose j in orientation i. $\mathbf{u}_{o,i}$ and $\mathbf{u}_{p,j}$ are respectively the ith row of $\mathbf{U}_o$ and the jth row of $\mathbf{U}_p$. Alternatively speaking, $\mathbf{u}_{o,i}$ is the coefficient vector representing the ith orientation, and $\mathbf{u}_{p,j}$ is the coefficient vector representing the jth pose. Therefore, an observation of a pose is modeled as the bilinear combination of columns in $\mathcal{D}$ with a orientation vector and a pose vector independent on each other.

Based on (2.10), a new pose observation vector $\mathbf{z}$ can be projected onto the pose coefficient space and the orientation coefficient space by solving the bilinear problem defined as follows:

$$\mathbf{z} = \mathcal{D} \times_2 \mathbf{v}_o \times_3 \mathbf{v}_p, \tag{2.11}$$

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

The problem of finding the pose and orientation coefficient vectors of a new input vector can be solved by using the iterative alternating least square (ALS) algorithm [73] as follows. Let the previous estimate of the orientation coefficient vector be $\hat{\mathbf{v}}_o^{(n)}$. Then $\mathcal{D}$ can be flattened into a matrix $\mathbf{C}_o^{(n)}$ by multiplying $\hat{\mathbf{v}}_o^{(n)}$.

$$\mathbf{C}_o^{(n)} = \mathcal{D} \times_2 \hat{\mathbf{v}}_o^{(n)}, \tag{2.12}$$

Inserting (2.12) into (2.11) we can get

$$\mathbf{z} = \mathbf{C}_o^{(n)} \mathbf{v}_p. \tag{2.13}$$

Thus, the current estimate of the pose coefficient vector $\mathbf{v}_p^{(n+1)}$ can be easily obtained by solving the linear system (2.13). Similarly, using the current pose coefficient vector $\hat{\mathbf{v}}_p^{(n+1)}$, we can update the estimate of the orientation coefficient vector $\mathbf{v}_o$ by solving a similar linear equation.

$$\mathbf{z} = \mathbf{C}_p^{(n+1)} \mathbf{v}_o, \tag{2.14}$$

where $\mathbf{C}_p^{(n+1)} = \mathcal{D} \times_3 \hat{\mathbf{v}}_p^{(n+1)}$. Given the initial value of $\mathbf{v}_p^{(0)}$ or $\mathbf{v}_o^{(0)}$, we can obtain both vectors by solving (2.13) and (2.14) alternately until convergence.

Initialization is important for ALS. One possibility to initialize ALS is to use the row vectors in $\mathbf{U}_o$ (standard orientation vectors) as one initial value of $\mathbf{v}_o$ and obtain a set of candidate

solutions. Then the final solution can be chosen to be the one resulting in the minimum reconstruction error. Using multiple initial values requires the ALS procedure to be performed multiple times and hence it is computationally expensive. In order to improve the computational efficiency and at the same time maintain the stability of the solution, an improved initialization method has been developed as follows. First, all the row vectors $\{\mathbf{u}_{o,i}\}_{i=1}^{N_o}$ of $\mathbf{U}_o$ are used 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 (2.13) ***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, as defined below.

$$\mathbf{v}_p^{(0)} = \arg\max_{\mathbf{v}_{p,i}} \max_j \frac{\mathbf{v}_{p,i} \cdot \mathbf{u}_{p,j}}{\|\mathbf{v}_{p,i}\| \cdot \|\mathbf{u}_{p,j}\|}. \tag{2.15}$$

where $i = 1,\ldots,N_o$ is the orientation index, and $j = 1,\ldots,N_p$ is the pose index. By choosing only one initial value $\mathbf{v}_p^{(0)}$ to solve (2.11), the computational efficiency is improved while good experimental results were still achieved, which will be shown later. By solving the bilinear equation (2.11), each input observation vector can be projected into a pose coefficient vector $\mathbf{v}_p$ which is independent of body orientation.

2.1.3 Extraction of View and Shape-Invariant Pose Features

The feature extraction framework presented in the previous subsection can be extended to extract pose features invariant to both view and body shape. The same pose tensor described in Section 2.1.2 can be constructed using observations of each of N_s people. When these tensors are assembled together, the pose tensor is expanded to $\mathcal{A} \in \mathbb{R}^{N_d \times N_o \times N_p \times N_s}$. Similar to (2.9), $\mathcal{A}$ can be decomposed into

$$\mathcal{A} = \mathcal{D} \times_2 \mathbf{U}_o \times_3 \mathbf{U}_p \times_4 \mathbf{U}_s. \tag{2.16}$$

in which $\mathbf{U}_s \in \mathbb{R}^{N_s \times N_s}$ is the matrix representing the body shape mode. Similar to (2.10), the decomposed tensor possesses property that

$$\mathcal{A}(:,i,j,k) = \mathcal{D} \times_2 \mathbf{u}_{o,i} \times_3 \mathbf{u}_{p,j} \times_4 \mathbf{u}_{s,k}, \tag{2.17}$$

in which $\mathbf{u}_{s,k}$ is the kth row of $\mathbf{U}_s$. Therefore, the generation of a new observation z can be modeled as a trilinear equation

$$\mathbf{z} = \mathcal{D} \times_2 \mathbf{v}_o \times_3 \mathbf{v}_p \times_4 \mathbf{v}_s, \tag{2.18}$$

in which $\mathbf{v}_s$ is the body shape coefficient vector, and $\mathbf{v}_p$ is the pose coefficient vector invariant to orientation and body shape.

The ALS method can also be applied to solve the trilinear equation (2.18). If $\hat{\mathbf{v}}_o^{(n)}$ and $\hat{\mathbf{v}}_s^{(n)}$ are known, we can get

$$\mathbf{C}_{os}^{(n)} = \mathcal{D} \times_2 \hat{\mathbf{v}}_o^{(n)} \times_4 \hat{\mathbf{v}}_s^{(n)}, \tag{2.19}$$

and $\hat{\mathbf{v}}_p^{(n+1)}$ can be obtained by solving the linear equation.

$$\mathbf{z} = \mathbf{C}_{os}^{(n)} \mathbf{v}_p. \tag{2.20}$$

Then, $\hat{\mathbf{v}}_o^{(n+1)}$ and $\hat{\mathbf{v}}_s^{(n+1)}$ can be updated by solving

$$\mathbf{z} = \mathbf{C}_{po}^{(n+1)} \mathbf{v}_s, \tag{2.21}$$

and

$$\mathbf{z} = \mathbf{C}_{ps}^{(n+1)} \mathbf{v}_o, \tag{2.22}$$

where

$$\mathbf{C}_{po}^{(n)} = \mathcal{D} \times_2 \hat{\mathbf{v}}_o^{(n)} \times_3 \hat{\mathbf{v}}_p^{(n+1)} \tag{2.23}$$

and

$$\mathbf{C}_{ps}^{(n+1)} = \mathcal{D} \times_3 \hat{\mathbf{v}}_p^{(n+1)} \times_4 \hat{\mathbf{v}}_s^{(n+1)}, \tag{2.24}$$

Therefore, by initializing $\hat{\mathbf{v}}_o^{(0)}$ and $\hat{\mathbf{v}}_s^{(0)}$, we can solve (2.20), (2.21) and (2.22) alternatively until convergence.

An initialization method similar to the method described in Section 2.1.2 has been adopted. First, all combinations of row vectors $\{\mathbf{u}_{o,i}\}_{i=1}^{N_o}$ of $\mathbf{U}_o$ and row vectors $\{\mathbf{u}_{s,k}\}_{k=1}^{N_s}$ of $\mathbf{U}_s$ are used as the initial values for $\mathbf{v}_o$ and $\mathbf{v}_s$. For each combination of $\mathbf{u}_{o,i}$ and $\mathbf{u}_{s,k}$, we find the corresponding pose vector $\mathbf{v}_{p,ik}$ by solving (2.20) ***only once***. Then the pose vector most similar to one of the standard poses $\{\mathbf{u}_{p,j}\}_{j=1}^{N_p}$ is chosen as the initial pose coefficient vector $\mathbf{v}_p^{(1)}$, as defined below.

$$\mathbf{v}_p^{(1)} = \arg\max_{\mathbf{v}_{p,ik}} \max_j \frac{\mathbf{v}_{p,ik} \cdot \mathbf{u}_{p,j}}{\|\mathbf{v}_{p,ik}\| \cdot \|\mathbf{u}_{p,j}\|}. \tag{2.25}$$

Then, $\mathbf{v}_p^{(1)}$ and its corresponding $\mathbf{u}_{o,\hat{i}}$ can be applied to initiate the iteration by renewing $\mathbf{v}_s^{(1)}$ by solving (2.21).

2.2 Invariant Pose Feature Extraction Using MGP

Invariant pose features can also be extracted using multifactor Gaussian process (MGP). Recently, kernel-based multifactor analysis has been developed to separate content and style from motion capture data. Such kernel-based approach has been found to be effective to represent and model potential nonlinearity involved in the contributing factors. Although the kernel-based multifactor analysis [74] has been proposed, it has been mainly used in modeling and synthesizing movement data. It has not yet been exploited in view/body-invariant pose feature extraction. In this section, an MGP based approach is applied to tackle the challenge of achieving invariance in extracting pose descriptors.

2.2.1 An Introduction to Multifactor Gaussian Process

Recently, the multifactor Gaussian process (MGP) model has been proposed [74] as a probabilistic kernel-based multifactor analysis framework for separation of style and content of movement data and movement synthesis. MGP was developed based on the Gaussian process (GP) method [75, 76] and the Gaussian process latent variable model [77] by inducing kernel functions based on the multilinear model. Because of the use of kernel functions, MGP is able to represent more general multifactor models beyond multilinear relationship. In addition, since MGP is rooted on GP, it is inherently a probabilistic framework.

Let f be a zero-mean Gaussian process, defined as a function of $\mathbf{X} = \{\mathbf{x}_1, \mathbf{x}_2 \ldots \mathbf{x}_N\}$ and the function values $\mathbf{f} = \{f(\mathbf{x}_n) | n = 1, 2, \ldots, N\}$ satisfy a multivariate Gaussian distribution

$$p(\mathbf{f}; \mathbf{K}) = \mathcal{N}(\mathbf{f}; \mathbf{0}, \mathbf{K}). \tag{2.26}$$

In (2.26), the elements of the covariance matrix $\mathbf{K}$ are defined based on the covariance function or *kernel function* $\mathbf{K}_{i,j} = k(\mathbf{x}_i, \mathbf{x}_j)$. Two of the most commonly applied kernel functions are the

linear kernel

$$k(\mathbf{x},\mathbf{x}') = \mathbf{x}\cdot\mathbf{x}', \tag{2.27}$$

and the RBF kernel

$$k(\mathbf{x},\mathbf{x}') = \exp(-\frac{\gamma}{2}\|\mathbf{x}-\mathbf{x}'\|^2). \tag{2.28}$$

Gaussian processes can be applied to predict the function value of unknown point. If a set of input variables $\mathbf{X}$ and corresponding function values $\mathbf{f}$ are known, the conditional distribution of function value f_* at a new input point $\mathbf{x}_*$ is [76]

$$p(f_*|\mathbf{f}) = \mathcal{N}(f_*;\mathbf{k}_*\mathbf{K}^{-1}\mathbf{f}, k_{**}-\mathbf{k}_*\mathbf{K}^{-1}\mathbf{k}_*^T), \tag{2.29}$$

in which $k_{**} = k(\mathbf{x}_*,\mathbf{x}_*)$ is the unconditional variance of f_*; $\mathbf{k}_*$ is the cross covariance vector of f_* and $\mathbf{f}$, i.e. $\mathbf{k}_{*,i} = k(\mathbf{x}_*,\mathbf{x}_i)$; and $\mathbf{K}$ is the covariance matrix of $\mathbf{f}$ determined by $\mathbf{X}$ and the kernel function as defined above. This conditional distribution of f_* given $\mathbf{f}$ is still Gaussian since they are jointly Gaussian. The conditional mean of f_* is a linear combination of sample function values $\mathbf{f}$. The conditional variance of f_* is smaller than the unconditional variance due to the knowledge of the sample points. In addition, this conditional distribution of f_* is completely determined by the kernel function and the input point $\mathbf{x}_*$ when the sample input and output points are given.

MGP is a special case of Gaussian process with a special design of kernel functions. MGP models the effect of multiple independent factors on the output. Therefore, the input space is divided into multiple factor spaces. An input variable $\mathcal{X} = \{\mathbf{x}^{(1)},\ldots,\mathbf{x}^{(M)}\}$ is made up of a variable in each factor space. The kernel function of MGP is then given by

$$k(\mathcal{X},\mathcal{X}') = \prod_{i=1}^{M} k_i(\mathbf{x}^{(i)},\mathbf{x}^{(i)'}) + \beta^{-1}\delta(\mathcal{X},\mathcal{X}'), \tag{2.30}$$

in which $\delta(\mathcal{X},\mathcal{X}')$ is 1 when $\mathcal{X} = \mathcal{X}'$ and zero otherwise, and β is the output noise factor [77]. The kernel function of each factor $k_i(\mathbf{x}^{(i)},\mathbf{x}^{(i)'})$ can be independently defined. Therefore, the influence of each factor on the output can be defined separately.

MGP can be applied to model the mapping from a latent space consisting of multiple factor spaces to a high-dimensional observation space. In this case, each dimension of the observation vector is an MGP, and kernel functions for all dimensions are the same or at most differed by a

linear scaling factor. If given an observation point set, the latent points are unknown, the model can be viewed as a special case of Gaussian process latent variable model (GPLVM) [77].

2.2.2 *Extraction of View-Invariant Pose Features Using MGP*

In order to extract view-invariant pose features, observation of a pose is modeled to be affected by two factors: pose and orientation. Therefore, a latent point can be represented as $\mathcal{X} = \{\mathbf{x}^{(p)}, \mathbf{x}^{(o)}\}$, in which $\mathbf{x}^{(p)}$ is the pose vector and $\mathbf{x}^{(o)}$ is the orientation vector.

In order to learn MGP models, observation vectors of a set of key poses in a set of orientations need to be generated. Details on formation of the training set can be found in the experimental section of this chapter. Let the training samples be a matrix $\mathbf{Y}_t$ in which each row is is an observation vector. Since zero-mean MGPs are applied, a mean vector needs to be subtracted from each row of $\mathbf{Y}_t$. After centering the training data, the training data are also normalized by dividing elements in each dimension by the standard deviation of the dimension. The resulting training observations are denoted as $\mathbf{Y}$. Let N_p be the number of key poses and N_o the number of designated orientations. Denoting $N = N_p \times N_o$ to be the number of training samples and D to be the dimensionality of the observation vector, the dimensionality of $\mathbf{Y}$ is $N \times D$.

The body orientation variable has one degree of freedom. In order to model the periodicity of rotation, a 2D orientation vector $x^{(o)}$ is used to represent the body orientation, which can be intuitively considered to be the corresponding location of the body orientation angle on the unit circle. On the other hand, D_p the dimensionality of the pose feature vector $x^{(p)}$ can be determined according to the distribution of the latent points. Initially, D_p is set to be equal to N_p and the latent points corresponding to $\mathbf{Y}$ are obtained according to the training procedure presented in the following paragraph. These latent points contain redundant information, and dimension reduction can be performed to improve computational efficiency in pose feature extraction. Similar to dimension reduction in the principal component analysis (PCA), D_p is found based on the eigen-analysis of the covariance matrix of the laten points. To reduce D_p, in our research we first compute $\mathbf{M}$ the covariance matrix of the learned pose vectors and find the eigenvalues of $\mathbf{M}$, which correspond to the energy of the covariance matrix. We then find the smallest set of eigenvalues so that a certain percentage of the energy can still be preserved

when the remaining eigenvalues are dropped. D_p is then selected to be the cardinality of the smallest set of eigenvalues satisfying the energy preservation constraint.

Given the dimensions of the latent space, the following training procedure can be taken to learn the latent points. During the training process, it is constrained that the latent points of samples belonging to the same pose remain the same. Likewise, the orientation coefficients of samples belonging to the same orientation will maintain to be the same. In this training process, the orientation coefficients of the designated orientations are initialized to be 2D points evenly distributed on the unit circle. The latent points of the key poses are initialized to be nonnegative D_p dimensional vectors. The pair-wise distances of these initial pose latent points are also made similar to each other. With such constraints and initial latent points, given training observations $\mathbf{Y}$, the corresponding set of latent points $\mathbf{X}^* = \{\mathcal{X}_n^*\}_{n=1}^N$ and parameters of the kernel function $\Gamma^* = \{\gamma_p^*, \gamma_o^*, \beta^*\}$ are optimized by maximizing the following log likelihood function with respect to $\mathbf{X} = \{\mathcal{X}_n\}_{n=1}^N$ and $\Gamma = \{\gamma_p, \gamma_o, \beta\}$:

$$\begin{aligned} L(\mathbf{X},\Gamma) &= \log p(\mathbf{Y}|\mathbf{X},\Gamma) = \log\left(\prod_{i=1}^{D} p(\mathbf{y}_i|\mathbf{K}(\mathbf{X},\Gamma))\right) \\ &= \frac{ND}{2}\log(2\pi|\mathbf{K}(\mathbf{X},\Gamma)|) - \frac{1}{2}\sum_{i=1}^{D}\mathbf{y}_i^T\mathbf{K}(\mathbf{X},\Gamma)^{-1}\mathbf{y}_i \end{aligned} \tag{2.31}$$

where $\mathbf{y}_i$ is the ith row vector of $\mathbf{Y}$. The covariance matrix $\mathbf{K}(\mathbf{X},\Gamma)$ is specified as $\mathbf{K}_{i,j}(\mathbf{X},\Gamma) = k(\mathcal{X}_i,\mathcal{X}_j,\Gamma)$ and the kernel function is defined as

$$\begin{aligned} &k(\mathcal{X},\mathcal{X}',\Gamma) = \\ &\exp(-\frac{\gamma_p}{2}\|\mathbf{x}^{(p)} - \mathbf{x}^{(p)'}\|^2)\exp(-\frac{\gamma_o}{2}\|\mathbf{x}^{(o)} - \mathbf{x}^{(o)'}\|^2) + \beta^{-1}\delta(\mathcal{X},\mathcal{X}'). \end{aligned} \tag{2.32}$$

in which $\Gamma = \{\gamma_p, \gamma_o, \beta\}$ is the parameter set of the kernel function.

The quasi-Newton [78] method is applied for optimization. At each optimization step, the latent points and kernel parameters are optimized together. In this case, the RBF kernel is applied for pose and orientation factors. Other types of kernel functions and their combinations will be explored in future research.

After the latent points $\mathbf{X}^*$ and model parameters Γ^* are learned from the training data $\mathbf{Y}$, the conditional distribution of the observation $\mathbf{y}_*$ corresponding to a new latent point $\mathcal{X}_* = \{\mathbf{x}_*^{(p)}, \mathbf{x}_*^{(o)}\}$ can be found as a normal distribution according to the Gaussian process theory.

Define $\mathbf{k}_* = \mathbf{k}(\mathcal{X}_*, \mathbf{X}^*, \Gamma^*)$ to be the cross-covariance vector with $\mathcal{X}_*$ and $\mathbf{X}^*$. The ith element of $\mathbf{k}(\mathcal{X}, \mathbf{X}', \Gamma)$ is given by

$$\mathbf{k}_i(\mathcal{X}, \mathbf{X}', \Gamma) = k(\mathcal{X}, \mathcal{X}'_i, \Gamma), \tag{2.33}$$

in which the kernel function k is defined in (2.32). Define $\mathbf{K}^* = \mathbf{K}(\mathbf{X}^*, \Gamma^*)$ and $k_* = k(\mathcal{X}_*, \mathcal{X}_*, \Gamma^*)$. The conditional distribution of $\mathbf{y}_*$ is given by

$$p(\mathbf{y}_*|\mathcal{X}_*, \mathbf{Y}, \mathbf{X}^*, \Gamma^*) = \prod_{i=1}^{D} \mathcal{N}(y_*^{(i)}; \mathbf{k}_* \mathbf{K}^{*-1} \mathbf{y}_i, k_* - \mathbf{k}_* \mathbf{K}^{*-1} \mathbf{k}_*^T), \tag{2.34}$$

where $y_*^{(i)}$ is the ith element of $\mathbf{y}_*$ and $\mathbf{y}_i$ is the ith row vector of $\mathbf{Y}$.

To extract invariant pose descriptors, we need to infer the latent point $\mathcal{X}_* = \{\mathbf{x}_*^{(p)}, \mathbf{x}_*^{(o)}\}$ from a new observation $\mathbf{y}_*$. In the proposed framework, this is done by solving the following optimization problem

$$\mathcal{X}_* = \arg\max_{\mathcal{X}} \log p(\mathbf{y}_*|\mathcal{X}, \mathbf{Y}, \mathbf{X}^*, \Gamma^*), \tag{2.35}$$

in which the distribution $p(\mathbf{y}_*|\mathcal{X}, \mathbf{Y}, \mathbf{X}^*, \Gamma^*)$ is defined in (2.34).

To solve this optimization problem, the quasi-Newton method is applied. Since the quasi-Newton method can only find the local optimal points, it is important to find a good initial point for the optimization process. Furthermore, it is often plausible to use multiple initial points to improve the optimization result. In the proposed framework, m points in $\mathbf{X}^*$ that yield m largest likelihood of $\mathbf{y}_*$ evaluated with function (2.34) are selected as the initial points. After optimization, m local optimal latent points can be obtained, and the latent point yielding the largest likelihood of $\mathbf{y}_*$ is chosen to be the solution for $\mathcal{X}_*$. In practice, there is a tradeoff in selecting m between precision (requiring a larger m) and computational efficiency (requiring a smaller m). In experiments discussed in this dissertation that are related to pose feature extraction using MGP, m=10 initial points are used.

By inferring the latent point $\mathcal{X}_*$, both pose coefficient $\mathbf{x}_*^{(p)}$ and orientation coefficient $\mathbf{x}_*^{(o)}$ can be obtained. The pose coefficient $\mathbf{x}_*^{(p)}$ can be applied as a feature vector describing the corresponding pose which is invariant to orientation.

2.2.3 Extraction of View and Shape-Invariant Pose Feature Using MGP

The MGP model described in the previous subsection can be extended to a 3-factor model, in which a latent point is expressed as $\mathcal{X} = \{\mathbf{x}^{(p)}, \mathbf{x}^{(o)}, \mathbf{x}^{(s)}\}$, and $\mathbf{x}^{(s)}$ is the coefficient vector representing the body shape. Then, the kernel function can be defined as

$$k(\mathcal{X}, \mathcal{X}', \Gamma) = \exp(-\frac{\gamma_p}{2}\|\mathbf{x}^{(p)} - \mathbf{x}^{(p)'}\|^2)\exp(-\frac{\gamma_o}{2}\|\mathbf{x}^{(o)} - \mathbf{x}^{(o)'}\|^2)\exp(-\frac{\gamma_s}{2}\|\mathbf{x}^{(s)} - \mathbf{x}^{(s)'}\|^2) + \beta^{-1}\delta(\mathcal{X}, \mathcal{X}'), \tag{2.36}$$

and the parameter set of the kernel function becomes $\Gamma = \{\gamma_p, \gamma_o, \gamma_s, \beta\}$.

To form the training data, a set of observation vectors of a set of pose performed by a set of people in a set of orientations are generated. Denoting N' to be number of observations in the training set, the latent point set corresponding to the training observations can be expressed as $\mathbf{X} = \{\mathcal{X}_n\}_{n=1}^{N'}$. Using updated kernel function (2.36), kernel parameters Γ and latent points $\mathbf{X}$, the 3-factor MGP model can be learned by optimizing the negative log likelihood function (2.31). Once the model parameters $\mathbf{X}^*$ and Γ^* are learned, the inference of $\mathcal{X}_*$ corresponding to observation $\mathbf{y}_*$ can be obtained by solving the optimization problem (2.35) in the same way as described in the previous subsection. Then the pose coefficient $\mathbf{x}_*^{(p)}$ contained in $\mathcal{X}_*$ is taken as the pose feature vector invariant to orientation and body shape.

2.3 Evaluations of View-Invariant Features

In this section, the quality of view-invariant pose features extracted using 2-factor models is evaluated.

2.3.1 The IXMAS Dataset

In order to systematically evaluate the view-invariance of pose features obtained using multilinear analysis and MGP, a set of experiments on the IXMAS action recognition dataset [60] were conducted. The IXMAS dataset contains calibrated multi-view silhouette and visual-hull data of 12 subjects performing 14 daily actions. For each subject, three trials of gesture data

were included in the dataset, each containing various execution of all 14 actions. To be consistent with [60] and [79], only data from 10 subjects and 11 actions were used and these subject and action sets are identical to those in [60] and [79].

2.3.2 Data Normalization

In the IXMAS dataset, 3D volumetric reconstructions of poses are provided. In order to account for the variations in volumetric reconstructions introduced by location and body shape differences across different gesture trials and subjects, the volumetric data need to be normalized. In the proposed framework, visual hull normalization involves centering and rescaling.

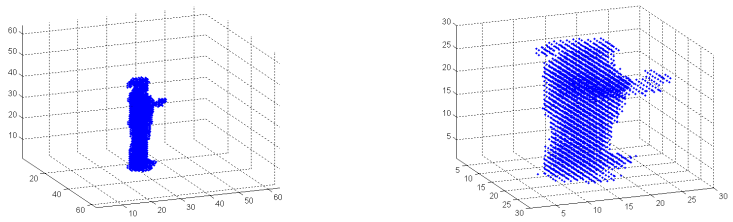
Let O_W be the original 3D world space used in the volumetric reconstruction with the third coordinate axis z upward. Assume that the resolution of O_W in voxel reconstruction is $D \times D \times D$. Let C be the coordinate set of all the valid voxels of the 3D body shape in O_W and $\mathbf{v} = (v_x, v_y, v_z)$ be the coordinates of a voxel in V. Let $\mathbf{c} = (c_x, c_y, c_z)$ be the centroid of the 3D body visual hull in O_W. To perform data centering, a body-centered local coordinate system O_B is first selected using $\mathbf{c}$ as the new origin. The coordinate axes of O_B are aligned with those of O_W. Let the resolution of O_B be $D' \times D' \times D'$. Voxel rescaling is required to reduce the effect of body shapes on the voxel reconstruction. It is carried out by mapping a valid voxel $\mathbf{v}$ in O_W onto a corresponding point $\mathbf{v}' = (v'_x, v'_y, v'_z)$ in O_B:

$$\begin{aligned} v'_x &= \frac{v_x - c_x}{M_R} D', \\ v'_y &= \frac{v_y - c_y}{M_R} D', \\ v'_z &= \frac{v_z - c_z}{M_Z} D', \end{aligned} \tag{2.37}$$

in which the mapping parameters M_R and M_Z are computed based on the horizonal and vertical spans of the original visual hull reconstruction.

$$\begin{aligned} M_R &= \max_{\mathbf{v} \in V} \sqrt{(v_x - c_x)^2 + (v_y - c_y)^2}, \\ M_Z &= \max_{\mathbf{v} \in V} |v_z - c_z| \end{aligned} \tag{2.38}$$

The scaling on z axis compensates the differences in height, and the scalings on x and y axes the differences in the other two dimensions. An example of visual hull normalization is shown in Figure 2.4.



(a) (b)

Fig. 2.4. An example of visual hull normalization. (a) Original visual hull (b) Normalized visual hull.

2.3.3 Key Pose Selection and Formation of Training Data

As discussed in previous sections, in order to enable extraction of view-invariant pose features, a training set needs to be first constructed.

The first step to construct this training set is to find a set of key poses. In order to do this, a number of key pose candidates are first detected from sample sequences of 3D volumetric reconstructions based on motion energy, and then these key pose candidates are clustered based on distances measured using their normalized volumetric reconstructions. These resulting cluster centers are then taken as the final set of key poses.

Let $\mathcal{V}_t$ be the 3D voxel data of the original reconstructed visual hull before normalization, at time t, and $\mathcal{V}_t(i,j,k)$ the value of the voxel at location (i,j,k). The reason for using the volumetric reconstructions in this step is to preserve the motion energy in the raw movement data. A difference visual hull $\mathcal{F}_t$ can be derived based on $\mathcal{V}_t$:

$$\mathcal{F}_t(i,j,k) = \begin{cases} 1 & \sum_{\tau=t-W/2}^{t+W/2} \mathcal{V}_\tau(i,j,k) > 0 \text{ and } \mathcal{V}_t(i,j,k) = 0 \\ 0 & \text{otherwise} \end{cases} \tag{2.39}$$

where W is the width of a time window. The number of valid voxel in $\mathcal{F}_t$ reflects the amount of motion during time window centered at t with length W. Given a continuous movement sequence of L frames, the "motion energy" at time t, for $W/2+1 \leq t \leq L-W/2$, is defined as the number of nonzero voxel in the difference visual hull $\mathcal{F}_t$, i.e.

$$E(t) = \sum_{i,j,k} \mathcal{F}_t(i,j,k) \tag{2.40}$$

In the proposed framework, given training samples of all the gestures in the gesture vocabulary, candidate key poses are first selected automatically when the motion energy $E(t)$ reaches local maximum or minimum.

Since some key pose candidates may repetitively appear in different gestures, the candidate key poses need to be clustered and only the poses at the cluster centers are selected as the final key poses. To cluster the candidate key poses, we need to compute the distance between two poses. In this step, the normalized volumetric reconstructions are used to suppress inter-pose distance caused by differences in body shapes and gesture execution locations. Further, the impact of view difference in distance computation between two candidate key poses are also removed.

We first define the distance between two volumetric reconstructions $\mathcal{V}$ and $\mathcal{V}$':

$$d_V(\mathcal{V},\mathcal{V}') = \frac{\| \mathcal{V} - \mathcal{V}' \|}{\| \mathcal{V} \cap \mathcal{V}' \|} \tag{2.41}$$

where $\| \cdot \|$ is the cardinality operator and it returns the number of valid (nonzero) voxels. The involved intersection ($\cap$) operation is carried out by treating binary visual hulls as logical data arrays.

Using $d_V(\cdot,\cdot)$, we can further derive $d_P(p_1,p_2)$, an orientation-independent distance measure between two poses p_1 and p_2. The difference between $d_P(\cdot,\cdot)$ and $d_V(\cdot,\cdot)$ is that $d_P(\cdot,\cdot)$ is computed in a way that it is an orientation-invariant distance measure between two poses while $d_V(\cdot,\cdot)$ is simply the distance between two visual hulls. Let $\mathcal{V}_1$ and $\mathcal{V}_2$ be the normalized visual hulls of two poses p_1 and p_2, respectively. The body orientations of the two poses in $\mathcal{V}_1$ and $\mathcal{V}_2$ are unknown and they can be arbitrary. To obtain an orientation-independent distance measure between p_1 and p_2, we define $d_P(\cdot,\cdot)$ as the following:

$$d_P(p_1,p_2) = \min_{\theta} d_V(\mathcal{V}_1, R(\mathcal{V}_2,\theta)) \tag{2.42}$$

where $R(\mathcal{V}_2,\theta)$ stands for the visual hull obtained by rotating $\mathcal{V}_2$ counterclockwise about the z axis of O_B in an angle of θ. In practice, given $\mathcal{V}_1$ and $\mathcal{V}_2$, $d_P(p_1,p_2)$ is found through exhaustive search over θ on a uniform grid in $(0,2\pi]$.

Using $d_P(\cdot,\cdot)$, the distance matrix of the candidate key poses can be reliably computed. Using this distance matrix, the ***normalized-cut*** [80] algorithm is applied to further cluster these poses.

There are two reasons for choosing the normalized-cut algorithm. Firstly, this algorithm works directly on the distance matrix. Secondly, using this algorithm we can obtain a global optimal solution instead of local optimal solutions obtained through iterative clustering algorithms such as *k*-means. After normalized-cut clustering, the center of each cluster is selected a member of the key pose set.

Using this key pose selection method, 25 key poses were selected from one of the movement piece performed by Florian (one of the ten subjects) in IXMAS dataset. This subject was selected because of the good quality of the volumetric reconstruction of his movement data. These selected 25 key poses are shown in Figure 2.5. Each pose is shown in the most distinguishable view.

After key poses were selected, observation vectors of each pose in 16 orientations evenly distributed in a circle are generated to form the training set for both multilinear and MGP feature extraction. Given a key pose, the volumetric reconstruction of the pose is rotated about the vertical axis (axis perpendicular to the ground plane) to form new reconstructions and vectorize these reconstructions to form desired observation vectors.

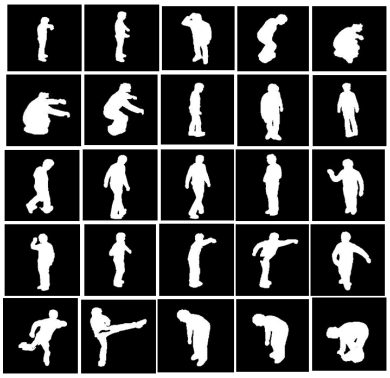


Fig. 2.5. Selected key poses from the IXMAS gesture dataset.

2.3.4 Data Reconstruction Test

Using trained multilinear and MGP models, the first test conducted is reconstruction of pose observation from pose feature. Given the volumetric reconstruction of a sample pose, pose features were extracted using both methods and then visual hulls were reconstructed using these

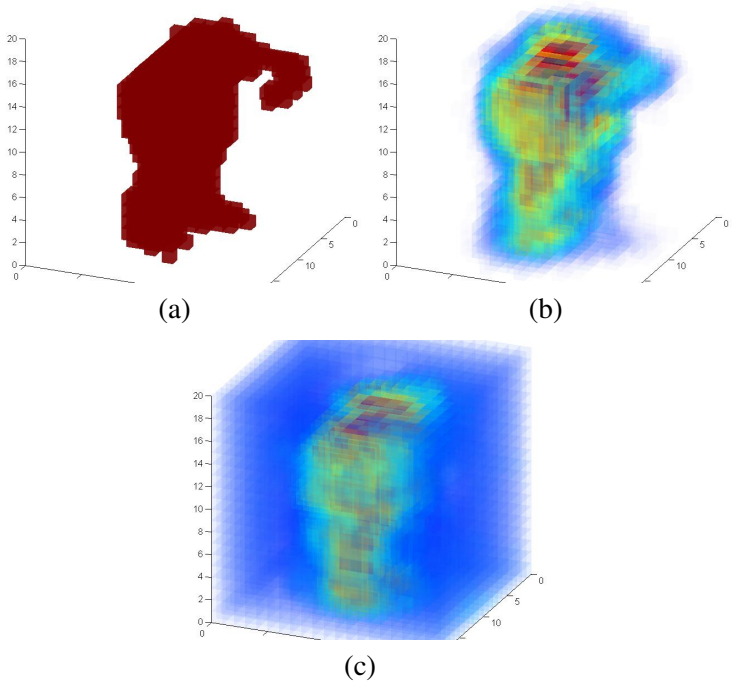


Fig. 2.6. Comparison of data reconstruction. (a) Original data (b) Reconstructed data using MGP. (c) Reconstructed data using multilinear projection.

pose feature vectors. The reconstructed 3D data are shown in Figure 2.6.

It can be seen that the data reconstructed using coefficients obtained by MGP mapping is much cleaner than that reconstructed using multilinear projection. Also, some details (hand) is missed in the latter reconstruction.

2.3.5 View-Invariance and Robustness Test

In order to test the view-invariance of pose features, two poses performed by a subject other than the subject whose data is used in training set are selected. One of them is close to a key pose in the training set and the other is not close to any key poses. The volumetric reconstructions of both poses are rotated to obtain their observations in 16 orientations. Then, both multilinear approach and MGP approach are taken to extract pose feature vectors from these observations. The volumetric reconstruction of the key pose in 16 orientations are shown in Figure 2.8 (a), and the 25 dimensional pose vectors extracted using the multilinear approach in Figure 2.8 (b) and the 12 dimensional pose vectors extracted using the MGP in Figure 2.8 (c). Similarly, Figure 2.9

(a) (b) and (c) presents volumetric reconstructions and the corresponding pose features of the non-key pose. From these figures it can be seen that the pose features extracted from volumetric reconstruction of the same pose viewed from different view angles are indeed close to each other.

In practice, visual-hull data are often noisy due to 3D reconstruction errors. Visual hull errors often exhibit as large blocks of uncarved background voxels and large blocks of miscarved foreground voxels. In this chapter, the first type of errors is referred to as protrusion errors and the second type partial occlusion errors. To verify and compare the robustness of the two proposed feature extraction approaches in the presence of such errors, pose features extracted from noisy visual-hull data have been examined. In this experiment, noisy visual-hull data were obtained by adding either the protrusion or partial occlusion errors to the original data used in the previous analysis (Figure 2.8 (a) and Figure 2.9 (a)). To add a protrusion error to a visual hull, a protrusion sphere with a random center in the background voxels and a diameter of a tenth of the side length of the volumetric reconstruction is first selected. To realistically synthesize a protrusion error, this random sphere has to overlap with the visual hull with overlapping volume less than half of the sphere. Otherwise, another random sphere will be selected and tested, until a valid protrusion error is synthesized. Once a protrusion sphere is found, all the voxels inside the sphere are considered protrusion voxels and their values set to 1. Likewise, to add a partial occlusion error, a partial occlusion sphere is first generated with a random center in the foreground voxels and a radius of three voxels. Then all the voxels within this sphere are considered occluded and their values set to 0. Examples of such noisy visual-hull data are shown in Figure 2.7.

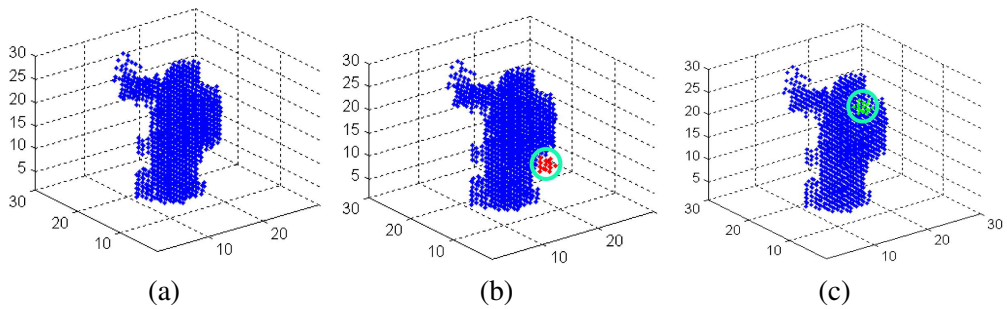


Fig. 2.7. Examples of two types of visual hull errors: (a) original visual hull (b) noisy data with a protrusion error (red) (c) noisy data with a partial occlusion error (green).

The noisy visual-hull data for the key pose and the corresponding pose vectors extracted using the multilinear and the MGP approaches are presented in the middle (protrusion error) and bottom (partial occlusion) rows of Figure 2.8. The noisy data and pose vectors of the non-key pose are given in Figure 2.9. It can be clearly seen from these figures that both the multilinear and the MGP approaches are robust to noises in volumetric reconstructions.

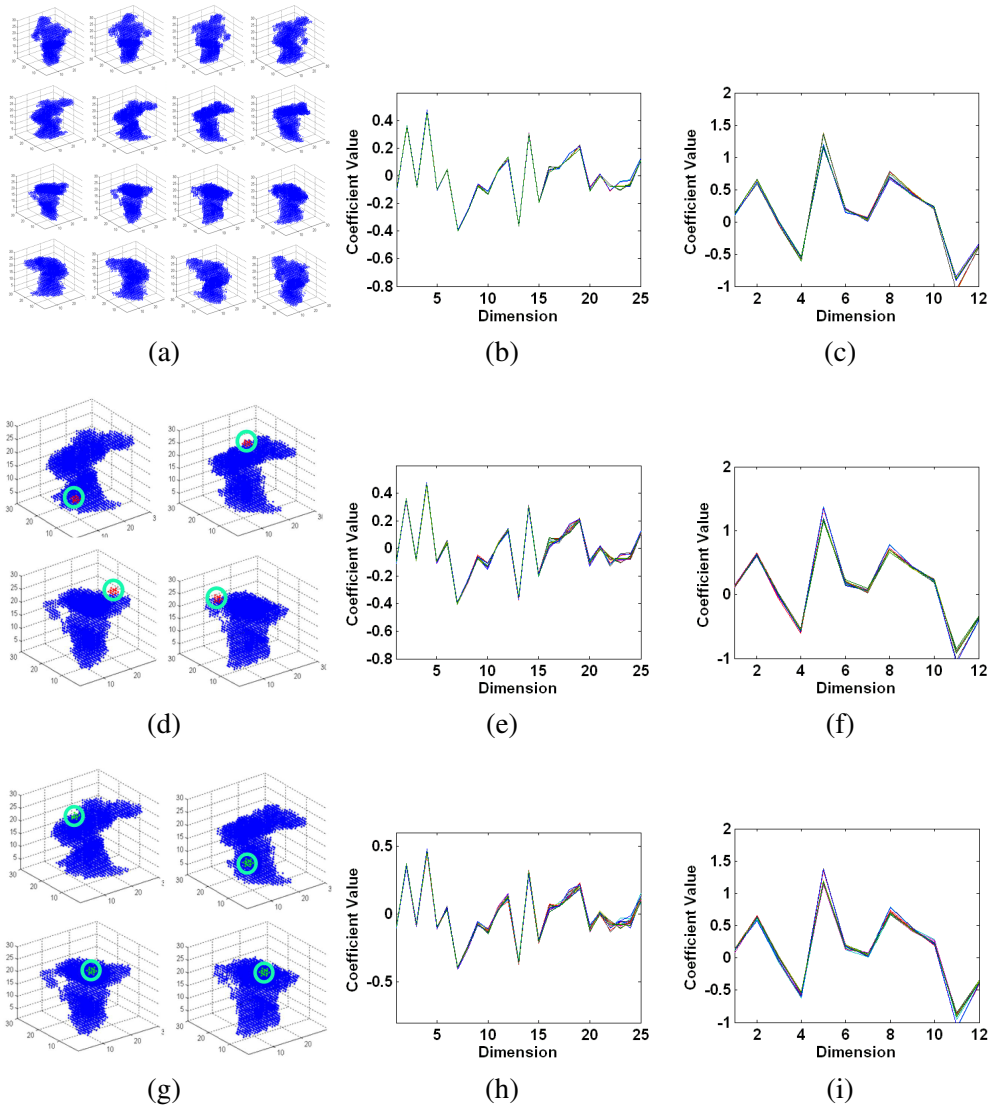


Fig. 2.8. Visual hull data of a key pose in 16 body orientations (upper-left) and its corresponding pose descriptors obtained using multilinear analysis (middle column) and MGP (right column) from original data (top row), sample data corrupted by protrusion errors (middle row), and sample noisy data with partial occlusion errors (bottom row).

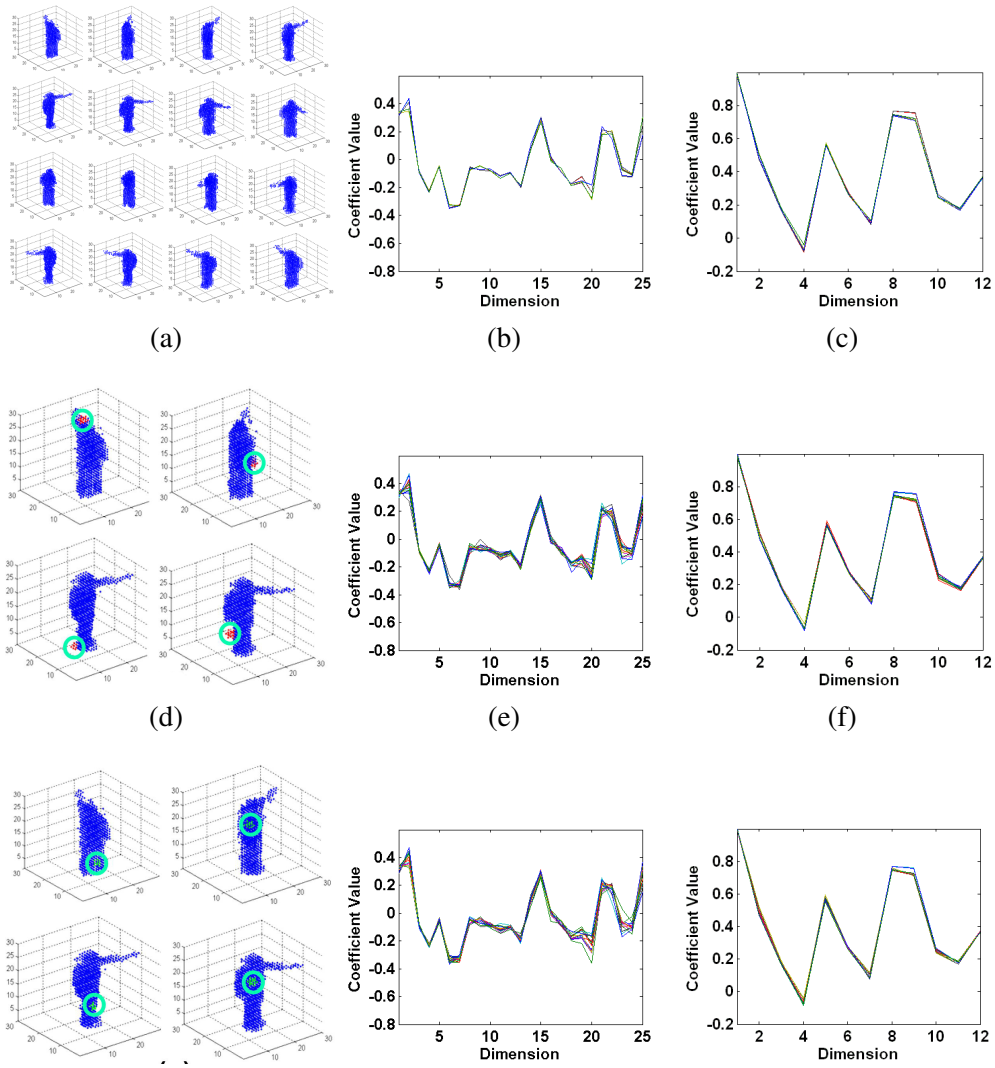


Fig. 2.9. Visual hull data of a non-key pose in 16 body orientations (upper-left) and its corresponding pose descriptors obtained using multilinear analysis (middle column) and MGP (right column) from original data (top row), sample data corrupted by protrusion errors (middle row), and sample noisy data with partial occlusion errors (bottom row).

To obtain a quantitative measure of the error-resilience for both approaches, the mean normalized inter-orientation distance (MNIOD) of the pose vectors has been computed for each scenario, indexed by the dataset (key pose vs. non-key pose), error type (protrusion-error vs. partial occlusion error), and the feature extraction method (multilinear vs. MGP). The normalization factor for each dataset and method is the mean of the norms of pose vectors obtained from the 16 visual hulls without noise. The resulting distances are shown in Table 2.1. According to Table 2.1, we can see that both the multilinear and the MGP approaches are robust

TABLE 2.1
MEAN NORMALIZED INTER-ORIENTATION DISTANCE (MNIOD) OF POSE VECTORS OBTAINED FROM VOLUMETRIC RECONSTRUCTIONS WITH DIFFERENT NOISE ADDED

Method & Noise Type	MNIOD for Key Pose	MNIOD for Non-Key Pose
Multilinear, original	0.055	0.112
Multilinear, protrusion	0.072	0.159
Multilinear, partial occlusion	0.085	0.177
MGP origional	0.074	0.025
MGP, protrusion	0.074	0.027
MGP, partial occlusion	0.083	0.028

to visual hull errors. Furthermore, the two types of errors appear to have similar impact to the MNIOD.

To evaluate the view-invariance in more general cases, tests were also performed on 100 poses randomly selected from the IXMAS dataset. The selected data include poses close to the key poses as well as those very much different from the key poses. Many of the visual-hull data of selected poses suffer from the protrusion errors and the partial occlusion errors introduced before. Examples of such volumetric construction errors are shown in Figure 2.10. Therefore, the results reported in this section also reflect the performance of the proposed features using noisy visual-hull data.

Using the selected visual-hull data, a view-invariance evaluation test dataset is then synthesized by rotating each of the testing poses to 16 orientations. Once the test dataset is constructed, for each pose, we can extract the corresponding pose features from these 16 visual hulls and obtain their pair-wise Euclidean distance. The maximum of these distances, defined as maximum inter-orientation distances (MIOD), are applied as a measure of view-invariance of pose vectors obtained at this frame. The histogram of the MIOD for all the testing poses will present a picture of the view-invariance of the corresponding pose feature.

The view-invariance property of pose descriptors extracted using both multilinear analysis and MGP were evaluated. In this study, 10 frames of visual-hull data were randomly selected from each of 10 subjects and totally 100 frames from the 10 subjects were used for view-invariance evaluation. Among these 100 frames, 18 frames are close to one of the key poses.

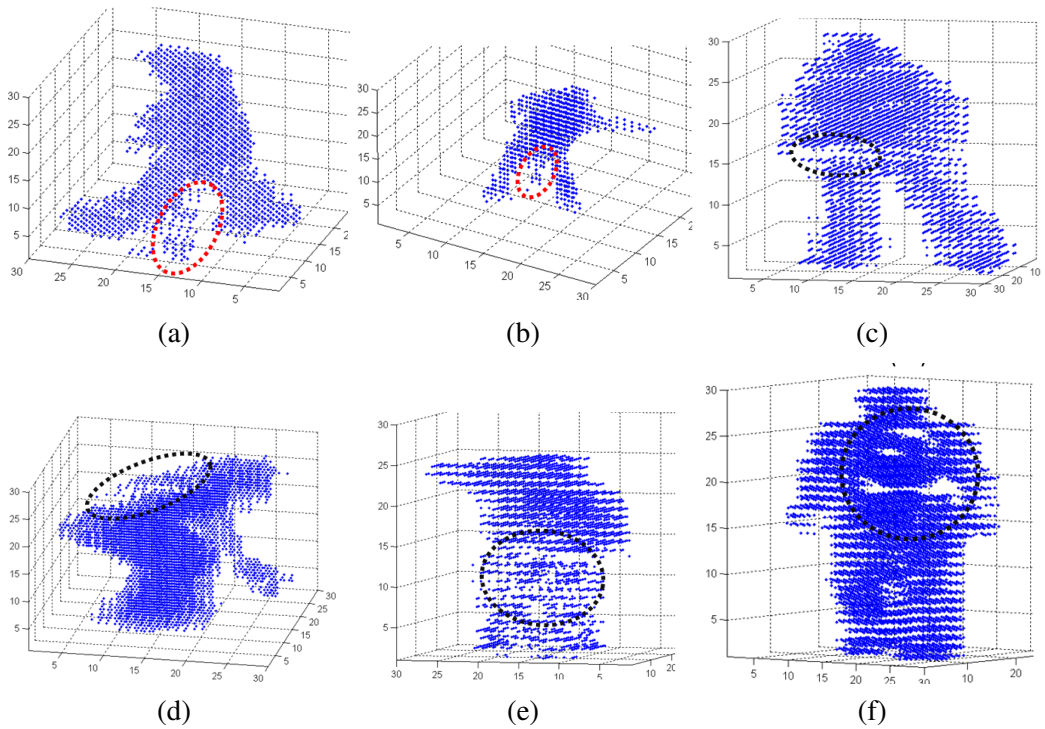
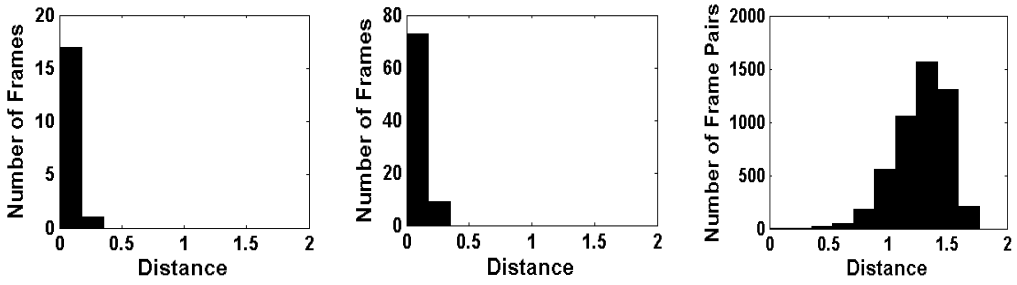


Fig. 2.10. Examples of noisy visual hull data in the IXMAS dataset. Typical errors (in the dotted circles) include remaining uncarved blocks in the background (a,b) and missing (wrongly carved) blocks in the foreground (c - f).

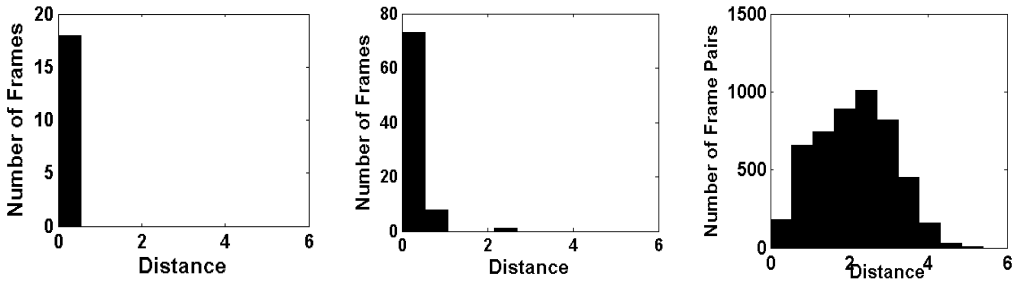
To put such a view-invariance difference measure into the proper context, the pair-wise inter-frame distances using the 100 original visual hull frames were also computed. The histogram of the MIOD of both the key pose frames and the non-key pose frames corresponding to the pose features obtained using multilinear analysis are shown in Figure 2.11 (a) and (b). The histogram of overall inter-frame distance is shown in Figure 2.11 (c). Using the same dataset, pose features were also extracted using the MGP method and the corresponding histograms are shown in Figure 2.12. Please note that the normal distance ranges of the multilinear and MGP features are different, 0 to 2 for the multilinear feature and 0 to 6 for the MGP feature (See Figures 2.11 and 2.12). To obtain a normalized view for the distance distributions, all the histograms are set to 10 bins, and the size of each bin is a tenth of the maximum overall inter-frame distance (MOIFD) of corresponding type of pose descriptor. From Figures 8 and 9 it can be seen that the MIOD values of nearly all the key-pose frames (17 out 18 for multilinear pose descriptors and all the 18 for MGP pose descriptors) and the majority (73 out 82 for multilinear pose descriptors and 78 out of 82 for MGP pose descriptors) of non-key pose frames

are less than a tenth of MOIFD. Therefore, only a small percentage (10% for multilinear pose descriptors and 4% for MGP pose descriptors) of the testing frames has high MIOD values. Hence, it was experimentally verified that the proposed pose feature extraction method using both multilinear analysis and MGP can effectively extract view-invariant features from visual-hull data. Moreover, between the two different view-invariant feature extraction approaches, the MGP-based approach exhibits slightly stronger view-invariance property than the multilinear analysis-based approach.



(a) (b) (c)

Fig. 2.11. Distance distributions of pose vectors obtained by multilinear analysis. (a) Inter-orientation distances of pose vectors obtained from key pose frames. (b) Inter-orientation distances of pose vectors obtained from non-key pose frames. (c) Inter-frame distances between pose vectors obtained from 100 frames.



(a) (b) (c)

Fig. 2.12. Distance distributions of pose vectors obtained using MGP. (a) Inter-orientation distances of pose vectors obtained from key pose frames. (b) Inter-orientation distances of pose vectors obtained from non-key pose frames. (c) Inter-frame distances between pose vectors obtained from 100 frames.

2.4 Evaluations of View and Shape-Invariant Pose Features

In this section, the quality of view and shape-invariant pose features extracted using 3-factor models is evaluated. The experiments in this section were still based on IXMAS dataset, as

Fig. 2.13. Selected 19 key poses for 3-factor model traning.

described in the previous section.

2.4.1 Formation of Training Data

In order to extract view and shape-invariant pose features using the multilinear and MGP methods, a set of training data needs to be constructed. In the previous section, 25 key poses were selected from poses performed by one of the subjects in the IXMAS dataset. However, not all key poses are commonly performed by all the subjects. In order to build a 4-mode (3-factor) training tensor for multilinear analysis, observation vectors of all the poses performed by all the subjects in all the orientations are necessary. Therefore, 19 key poses performed by all the 10 subjects except Chiara and Clare were selected to form the training poses. The selected key poses are show in Figure 2.13. The resulting training tensor was formed by the observations of 19 poses by 8 people in 16 orientations. The same training data was applied to train the 3-factor MGP model, except that due to limited computer memory, not all of the 16 observations of a pose performed by a person were included.

2.4.2 Invariance Test

The invariance property of the pose features was first tested using a sample pose. Instances of the sample pose performed by all ten subjects in IXMAS dataset were selected as the testing data. The testing instances did not include those applied in training. The 3D reconstructions of the testing instances of the sample pose are shown in Figure 2.14. It can be seen from Figure 2.14 that these pose instances were performed by subjects with different body shapes

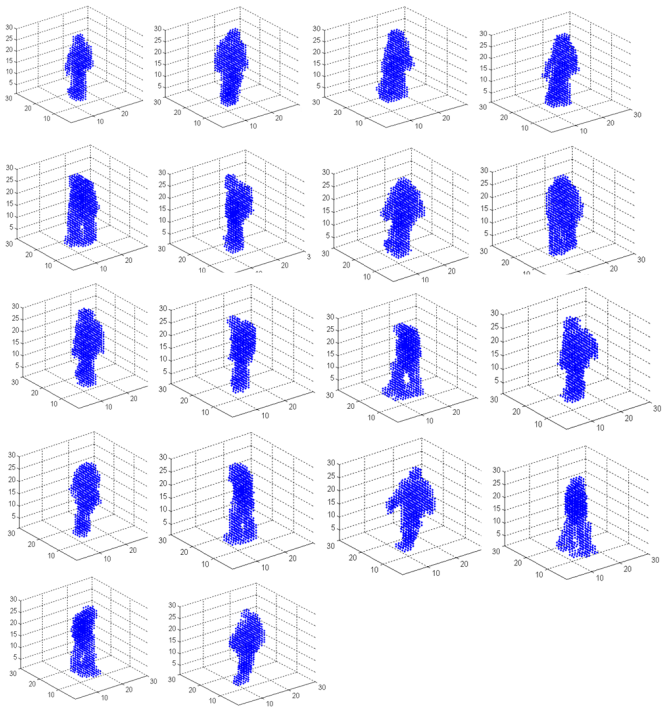


Fig. 2.14. Selected testing instances of a sample pose.

in different body orientations. From these 3D reconstructions, view-invariant pose features as well as view and shape-invariant pose features were extracted using both multilinear analysis and MGP. These features are shown in Figure 2.15.

In order to evaluate the invariance of the pose features quantitatively, the mean normalized inter-shape distance (MNISD) similar to MNIOD in the previous section was applied. For each feature extraction method, inter-body-shape distances are normalized by the mean of the norms of all the pose features extracted from the instances of a pose. The MNIODs of the pose features extracted from the sample pose are listed in Table 2.2. From both Figure 2.15 and Table 2.2 it can be seen that the invariance of the view and shape-invariant features extracted using the 3-factor MGP is the best. We can also see noticeable improvement in invariance from features extracted using the 2-factor MGP to those extracted using the 3-factor MGP. Application of the 3-factor model in multilinear analysis, however, does not improve pose feature invariance very much compared to the 2-factor model.

Feature invariance test described above has been repeated for all the 19 key poses applied in the training of 3-factor models. For each key pose, testing instances were selected from all

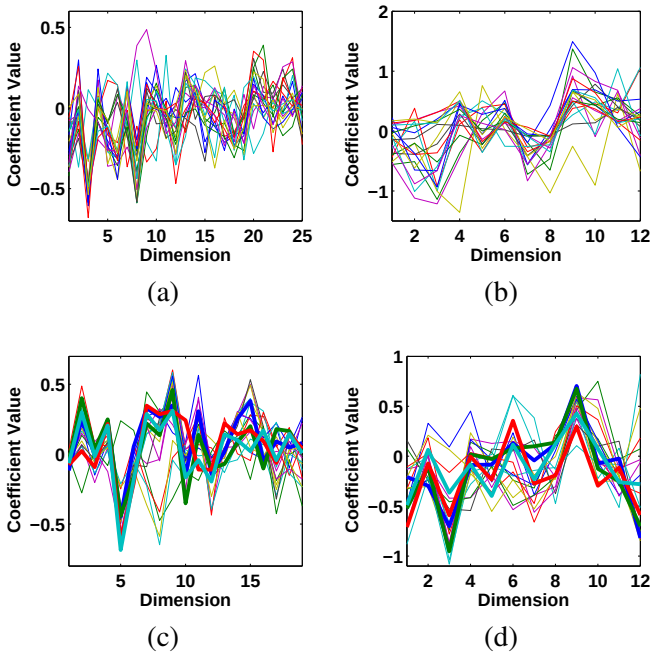


Fig. 2.15. Pose features extracted from 3D reconstructions shown in Figure 2.14. (a)View-invariant features obtained using multilinear analysis, (b)View-invariant features obtained using MGP, (c)View and shape-invariant features obtained using multilinear analysis, (d)View and shape-invariant features obtained using MGP.

TABLE 2.2
MEAN NORMALIZED INTER-SHAPE DISTANCE (MNISD) OF POSE FEATURES OBTAINED FROM THE INSTANCES OF THE SAMPLE POSE

Extraction Method	MNISD for view-invariant features	MNISD for view and shape-invariant features
Multilinear	0.943	0.912
MGP	1.01	0.886

the ten subjects in the IXMAS dataset in the same way as in the previous subsection. MNISD was calculated for each key pose, and the histograms of MNISDs are shown in Figure 2.16. For each feature extraction method, the mean of MNISDs of all the key poses are given in Table 2.3. From Figure 2.16 and Table 2.3 some general trends of invariance of different pose features can be obtained. First, in concurrence with the previous subsection, view and shape-invariant features extracted using 3-factor MGP show noticeable advantages in invariance over all the other feature extraction models. Second, it can be noticed that use of the 3-factor model in multilinear analysis does not improve and even deteriorates pose feature invariance compared to the 2-factor model. There are two reasons for such deterioration. First of all, observations

TABLE 2.3
MEAN OF MNISD OF POSE FEATURES OBTAINED FROM 19 KEY POSES

Extraction Method	Mean MNISD for view-invariant features	Mean MNISD for view and shape-invariant features
Multilinear	0.899	0.957
MGP	0.800	0.706

of poses are nonlinear in pose mode. Therefore the nonlinear MGP model performs better than multilinear analysis. In addition, it is harder to find a stable solution for a trilinear equation (2.18) than to solve a bilinear equation (2.11). Due to such deterioration, invariant pose features obtained using 3-factor multilinear analysis are not used in experiments in other chapters of this dissertation.

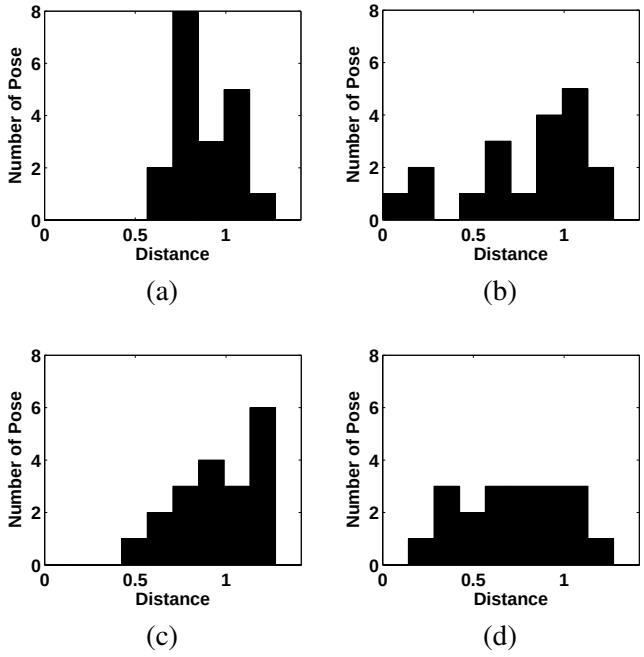


Fig. 2.16. Histogram of MNISD of features extracted from 19 key poses. (a) View-invariant features obtained using multilinear analysis, (b) view-invariant features obtained using MGP, (c) view and shape-invariant features obtained using multilinear analysis, (d) view and shape-invariant features obtained using MGP.

Chapter 3

POSE RECOGNITION USING INVARIANT POSE FEATURES AND SVM

The aim of pose recognition is to classify a pose observation into one of the poses in the vocabulary or identify it as an outlier. In this chapter, a method of video-based pose recognition using invariant pose features discussed in the previous chapter is presented.

3.1 Overview of Video-Based Pose Recognition

Video-based pose recognition has been extensively studied in the literature [81]. Existing methods can be mainly categorized into three groups according to types of features extracted, namely body kinematics [82, 83], 3D volumetric reconstruction [34] and 2-D silhouettes [84, 85].

Pose recognition would be straightforward when body kinematics such as joint angles can be reliably recovered from the input images. Recently video-based motion capture has seen tremendous progress using various generative-based (e.g. [86–88]) and discriminative-based (e.g. [89,90]) approaches. Various dynamical models have been used to represent the movement dynamics and at the same time reduce the dimensionality of the movement state space [91, 92]. Recent literature surveys can be found in [93–95]. Once body kinematics are recovered, poses can be recognized using body joint angles as the feature. However, video-based motion capture is mainly limited to pre-trained types of movements, such as walking and running. Reliable recovery and tracking of poses for general movement which has not been seen in the training remains a very challenging task. In some situations, such as dance performance, the subject can easily go through a wide variety of movements. It is unrealistic for a video-based motion capture system to keep tracking through such untrained movement.

As an alternative, many methods have been developed for silhouette based pose recognition. N. R. Howe [84] achieved pose tracking by looking up a collection of silhouettes of known poses. F. Huang, H. Di and G. Xu [85] proposed a viewpoint insensitive recognition system using “envelope shape” representation of poses, and performed experiments on several simple actions. F. Guo and G. Qian have also performed research on dance pose recognition in [96].

Gaussian Mixture Model (GMM) was used for feature extraction of the silhouette and relevance vector machine (RVM) for pose recognition.

Another alternative approach is to first recover the 3D volumetric reconstruction of the performer using e.g. visual-hull techniques [34,86] and then conduct pose recognition based on the 3D voxel data. Compared to silhouettes, 3D volumetric reconstruction is a much more informative representation of poses and great reduces the occlusion, but to recover a 3D body structure increases the demand for data-capturing hardware (certain amount of calibrated cameras (e.g. 6) are needed) and computational power.

In this chapter, a method of pose recognition using invariant features extracted from 2D silhouettes or 3D volumetric reconstructions is presented. In this method, support vector machine(SVM) is applied as the tool of classification. Using this method, promising experimental results have been obtained.

3.2 An Introduction to Support Vector Machines

This section briefly introduces binary SVM classifiers. A binary SVM classifier applies a linear model in a feature space in the form of

$$y(\mathbf{x}) = \mathbf{w}^T \boldsymbol{\phi}(\mathbf{x}) + b, \tag{3.1}$$

in which $\mathbf{x}$ is the input vector, $\mathbf{w}$ is the linear coefficient, b is the bias scalar and $\boldsymbol{\phi}(\mathbf{x})$ is a set of feature functions that transform input vectors into a feature space. The feature function set can be finite or infinite. Given a trained model in the form of (3.1), a dividing hyperplane would form in the feature space defined by $y(\mathbf{x}) = 0$, and a new input vector $\mathbf{x}^*$ can be classified according to the sign of $y(\mathbf{x})$. The distance from $\mathbf{x}^*$ to the dividing hyperplane in the feature space is given by

$$d_f(\mathbf{x}^*) = \frac{|y(\mathbf{x}^*)|}{\|\mathbf{w}\|}. \tag{3.2}$$

Given a set of N training input vectors $\mathbf{x}_1, \ldots, \mathbf{x}_N$ and their corresponding target values $t_1, \ldots, t_N$, where $t_n \in \{-1, 1\}$, and defining slack variables $\xi_1, \ldots, \xi_N$, the target of SVM is to maximize the classification margin while softly penalizing wrongly classified points by minimizing

$$C\sum_{n=1}^{N}\xi_n+\frac{1}{2}\|\mathbf{w}\|^2 \tag{3.3}$$

subject to

$$t_n y(\mathbf{x}_n)\geq 1-\xi_n \tag{3.4}$$

$$\xi_n\geq 0 \tag{3.5}$$

for $n=1,\ldots,N$. In (3.3), $C>0$ is a trade-off parameter.

The dual problem of this optimization problem is to minimize

$$\tilde{L}(\mathbf{a})=\sum_{n=1}^{N}a_n-\frac{1}{2}\sum_{n=1}^{N}\sum_{m=1}^{N}a_n a_m t_n t_m k(\mathbf{x}_n,\mathbf{x}_m) \tag{3.6}$$

subject to

$$0\leq a_n\leq C \tag{3.7}$$

$$\sum_{n=1}^{N}a_n t_n=0 \tag{3.8}$$

for $n=1,\ldots,N$. In (3.6), $k(\mathbf{x}_n,\mathbf{x}_m)=\phi(\mathbf{x}_n)^T\phi(\mathbf{x}_m)$ is the kernel function of $\mathbf{x}_n$ and $\mathbf{x}_m$.

Given this dual form, in an SVM framework, we do not need to explicitly define the feature functions. Instead, we only need to define the kernel function for any pair of input vectors. Solving the dual problem, we can obtain $a_n, n=1,\ldots,N$. Denoting $\mathcal{S}=\{n|a_n>0\}$, which is also known as the indices of support vectors, and denoting $N_{\mathcal{S}}=|\mathcal{S}|$, the discriminative function $y(\mathbf{x})$ can be calculated as

$$y(\mathbf{x})=\sum_{n=1}^{N}a_n t_n k(\mathbf{x},\mathbf{x}_n)+b, \tag{3.9}$$

in which

$$b=\frac{1}{N_{\mathcal{S}}}\sum_{n\in\mathcal{S}}(t_n-\sum_{m\in\mathcal{S}}a_m t_m k(\mathbf{x}_n,\mathbf{x}_m)). \tag{3.10}$$

Also, $\|\mathbf{w}\|$ can be computed as

$$\|\mathbf{w}\|=\sqrt{\sum_{n=1}^{N}\sum_{m=1}^{N}a_n a_m k(\mathbf{x}_n,\mathbf{x}_m)}. \tag{3.11}$$

Details of SVM classification can be found in [97].

3.3 Pose Recognition Using Multiple SVMs

Multi-class pose recognition can be achieved by using a set of support vector machine (SVM) classifiers in a one-versus-the-rest manner [98]. For each pose, we train a binary SVM classifier to identify whether the input pose feature "is" or "is not" the target pose. RBF kernels as defined in (2.28) are used in all binary SVM classifiers, and kernel parameters are set to be the same. Therefore, feature spaces of all the SVM classifiers are also the same.

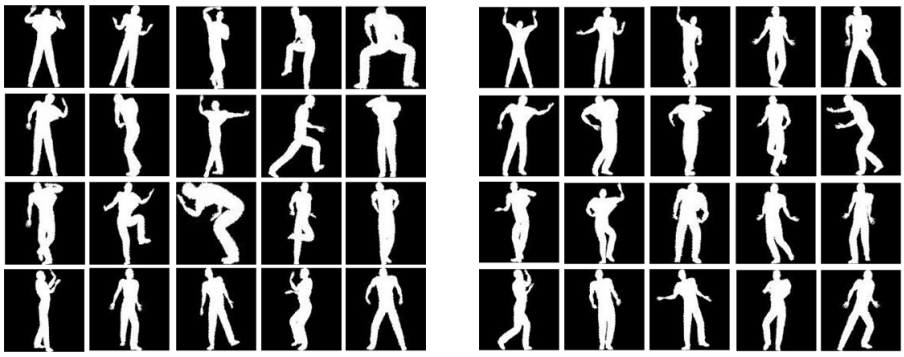
When all the classifiers are trained, recognition of a pose feature can be achieved by a traversal of all the classifiers. If all the classifiers returns negative results, then the pose feature is classified as an outlier. Otherwise, it is possible that one or more SVM classifiers return positive results. In this case, among SVM classifiers returning positive results, the pose feature is recognize as the pose corresponding to the classifier yielding the maximum distance from the testing point to the dividing hyperplane in feature space. This distance can be easily computed by combining (3.2),(3.9) and (3.11).

3.4 Experimental Results on Dance Pose Data

Pose recognition method described in this chapter was first tested on a two-view data set containing 20 dance poses choreographed by a professional dancer. These poses are shown in Figure 3.1 (a). This data set also contains 20 trick poses. These trick poses are outliers but are similar to one of the 20 standard poses. The trick poses are shown in Figure 3.1 (b). This experiment is designed to test the ability of the discussed method to perform multi-class pose recognition with outliers.

3.4.1 Data Acquisition and Preprocessing

In this test, pose images were obtained using two uncalibrated wide-baseline cameras. The configuration of the cameras is shown in Figure 3.2. These two cameras are mounted at approximately half body height with looking directions parallel to the ground plane and orthogonal to each other. The restrictions of the setting of cameras are not strict, so the subject has some flexibility to move around the intersection of the optical axes of the two cameras. Using such a



(a) (b)

Fig. 3.1. (a) 20 full-body dance poses used for pose recognition, (b) samples of trick poses

binocular approach can reduce ambiguities in body shape representation by reducing occlusion in some conditions (e.g. side view of poses that all the limbs are in the frontal plane). By setting up two cameras orthogonal to each other, the two captured images can be complementary most of the time.

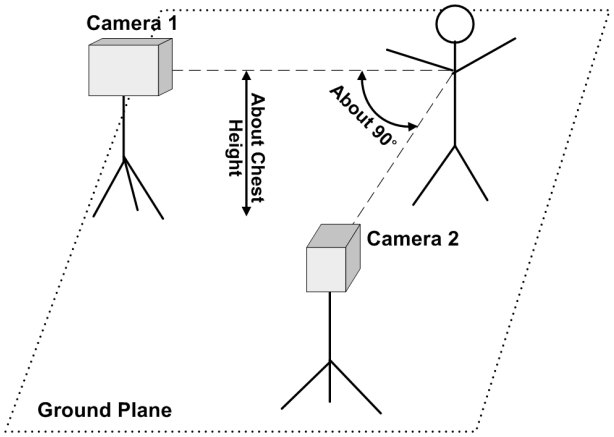


Fig. 3.2. Configurations of two uncalibrated cameras.

From each pair of images obtained from the two cameras, silhouettes are extracted. Each silhouette is then normalized and resized so that all the silhouettes are of the same height and horizontally centered. A pair of such normalized silhouettes are vectorized and concatenated to form a complete observation vector.

A pair of sample images taken from the two cameras and the silhouettes extracted from them are shown in Figure 3.3, and their normalized silhouettes are shown in Figure 3.4.

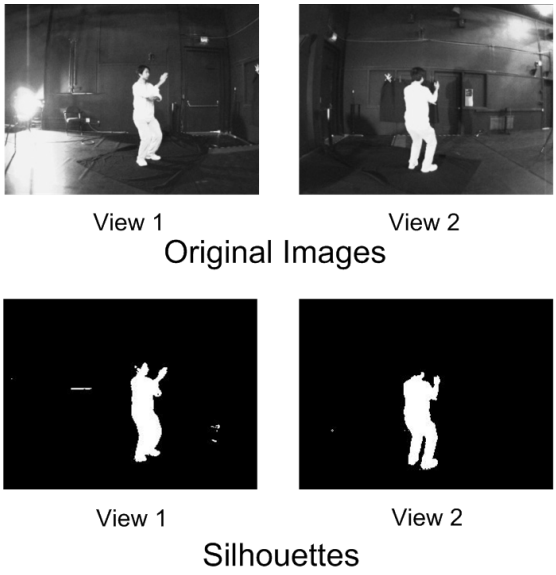


Fig. 3.3. Sample images obtained from two uncalibrated cameras and their silhouettes.

Fig. 3.4. Normalized silhouettes obtained from two uncalibrated cameras.

3.4.2 Synthesizing Data for Tensor Training

In tensor training, in order to obtain images of a pose in precise orientations, the 3D reconstruction of the pose is synthesized using motion capture data and animation software. Then, the 3D reconstruction is projected onto image planes in different angles to form 2D silhouettes, and these silhouettes are vectorized and concatenated to form desired observation vectors.

3.4.3 Pose Recognition Results

In pose recognition test, pose features were extracted using both multilinear and MGP method. In order to train the SVM classifiers, synthetic image as well as real images captured by two video cameras were applied. Images applied for testing were additional real images which were different from those in the training set. For each pose, there were 192 synthetic training sam-

TABLE 3.1
RECOGNITION RESULTS OF 20 DANCE POSES

Feature Extraction Method	SVM parameters	Recognition Rate	False Detection Rate
Multilinear Analysis	γ=0.7, C = 5	87.81%	5%
Multilinear Analysis	γ=1.2, C = 6	88.75%	8.75%
Multilinear Analysis	γ=1, C = 2	89.06%	10.63%
MGP	γ=0.2, C = 6	62.81%	5%
MGP	γ=0.3, C = 4	68.43%	8.75%
MGP	γ=0.4, C = 9	74.38%	10.63%

ples and 8 real training samples, and 16 real image pairs were used as testing samples. For trick poses, 320 real image pairs were used, 160 of them applied as training samples, and the remaining 160 applied as testing samples.

In the test, each testing sample is recognized as one of the 20 poses or as an outlier. Recognition results are evaluated using the recognition rate (RR) and the false detection rate (FDR). The recognition rate is the percentage of testing samples of standard poses that are correctly recognized as their corresponding poses. The false detection rate is the percentage of testing samples of trick poses that are wrongly recognized as one of the standard poses. The RR and FDR of pose recognition using both multilinear analysis based features and MGP based features are shown in Table 3.1. It can be seen that using features obtained from multilinear analysis has led to much better pose recognition results than those obtained using features obtained from the MGP method.

3.5 Experimental Results on the IXMAS Dataset

The pose recognition method is also tested on IXMAS dataset, which is introduced in the experimental section of previous chapter. The same 25 key poses as in the previous chapter were used for invariant feature extraction. To obtain training and testing data, poses similar to one of the key poses were selected from the movement pieces performed by the 10 subjects in the dataset. Among the 25 poses, five poses are discarded because do not commonly exist in movements of all subjects, The remaining 20 pose were chosen as the set of poses for recognition. The 20 pose are shown in Figure 3.5. On average, there are 23 samples of each pose

performed by all subjects for training and testing. From Figure 3.5 we can see that some of the pose pairs in these 20 poses are similarly to each other, making this dataset more challenging. This challenging dataset is applied as a benchmark dataset so that the discussed method can be compared with other classification methods.

Fig. 3.5. Twenty poses selected from the IXMAS data set.

For this data set, training and testing are performed in a cross-validation manner. At each cycle, poses performed by one subject is applied as testing data and the remaining are applied as training data. This process is repeated for all the 10 subjects.

In this experiment, all the testing samples are from the 20 target poses. Therefore, during pose recognition, a testing sample will be classified into one of the poses in the pose vocabulary. In this case, we simply assign the sample to the pose corresponding to the classifier yielding the maximum signed distance to the SVM dividing hyperplane in the feature space and every testing sample is assigned to one and only one pose class. To evaluate the performance of pose recognition, we have computed the recognition rate (RR) and the false alarm rate (FAR). For a particular pose p, the corresponding RR is computed as the percentage of correctly recognized in-class pose samples, and the corresponding FAR is the ratio of the number of testing samples misrecognized as pose p to the total number of out-class samples for pose p (i.e., testing samples of the other poses).

To compare different pose feature extraction and classification methods, pose recognition results were obtained from the view-invariant multilinear and the MGP features(extracted using 2-factor models), from view and shape-invariant MGP features (extracted using 3-factor models) as well as from raw visual hull data using the SVM and the K-nearest neighbors (K-NN) classifiers. The Euclidean distances have been used in all cases. In the case of SVM, the RBF

TABLE 3.2
RECOGNITION RESULTS OF 20 POSES IN IXMAS DATASET

Classifier	Feature Extraction Method	Recognition Rate	False Alarm Rate
SVM	Multilinear Analysis	74.40%	1.35%
	MGP	68.09 %	1.68%
	MGP-3 factor	74.58 %	1.34%
	Raw visual hull data	66.42 %	1.77%
K-NN	Multilinear Analysis	71.43 %	1.51%
	MGP	64.38%	1.88%
	MGP-3 factor	72.73%	1.43%
	Raw visual hull data	65.49%	1.81%

kernel has been used for the multilinear and MGP features. When the raw visual hull data is used in SVM, the linear kernel has been adopted, due to the high dimensionality of the visual hull data [99]. For each classifier-feature (or raw data) scenario, a grid search has been carried out in the corresponding parameter space to identify the best parameters. The pose recognition results obtained using the optimal classification parameters for each case are given in Table 3.2.

From Table 3.2, the following conclusions can be drawn. Firstly, 3-factor MGP features has led to the best pose recognition results using either SVM or K-NN. Compared to 2-factor MGP features, using 3-factor MGP features in pose recognition has led to great increase in accuracy. This further verified that 3-factor MGP features are more invariant with the change of body shapes. Secondly, among features not extracted with 3-factor model and raw data, the multilinear feature has led to the best results. This concurs with the results in the previous experiment. The reason for this is that in multilinear analysis, observations are projected onto the pose feature space discriminatively. Pose recognition accuracy using multilinear features and SVM is only about 0.2% lower than the accuracy achieved using 3-factor MGP features. Therefore, the two results are totally comparable. Furthermore, the average running time for extraction of multilinear features per frame is much smaller (about 4 seconds v.s. about 1 minute) than that for extraction of 3-factor MGP features.

Chapter 4

GESTURE CLASSIFICATION USING HMM

In this chapter, an approach to classification of gestures which are already segmented, i.e. the starts and ends are already known, is presented.

4.1 Overview of Video-Based Gesture Recognition

Many video-based methods have been developed for hand [100–102], arm [64, 103] and full-body [10,60] gesture recognition. Recent literature surveys on gesture recognition can be found in [104, 105].

According to the system methodology, video-based gesture recognition systems can be classified as either kinematic-based [10, 14, 35–37, 101, 106] or template-based approaches [33, 60, 79, 103, 107–109].

In kinematic-based gesture recognition [10, 14, 35–37, 101, 106], movement kinematic parameters related to the articulated body motion are first recovered as joint angle vectors [10], body-centered joint locations [35] or body part positions [14, 101, 106]. Gesture recognition is then conducted in such kinematic parameter spaces. The major weakness of kinematic-based gesture recognition is that the recovery of the movement kinematics is subject to tracking failures, especially in complicated full-body movements. As discussed in the previous section, reliably tracking joint angles from video, often referred to as *video-based motion capture*, is itself a very challenging task for computer vision. Although recently there has been noticeable progress in video-based motion capture, the state-of-the-art technology is only able to track the kinematic parameters for pre-trained movements.

On the other hand, template-based approaches such as [33,60,79,107–110] do not use such an intermediate kinematic representation, but instead directly represent actions using image information such as silhouettes or 3D volumetric reconstructions (e.g., visual hulls). Compared to the kinematic-based approaches, the template-based approaches are more practical and applicable in real life human-machine interactions. Based on how a gesture is represented, template-based

gesture recognition approaches fall into two categories: the *holistic* and *sequential* approaches. In the holistic approaches, e.g. [60, 107, 108, 110], to recognize a gesture, the entire gesture segment is first modeled as a spatio-temporal shape either in the 3D image-time space in a monocular setup, or in a 4D visual hull-time space in a multi-view scenario. Then features are extracted from such 3D or 4D gesture representations. Using features from training data, a gesture recognizer can be built using popular statistical pattern recognition techniques such as SVM, LDA. In contrast to the holistic approaches, sequential approaches, e.g. [33, 79, 109], represent a gesture as a temporal series of templates of a set of key poses selected from the gesture vocabulary in training. Pose features are then extracted from these key poses. A gesture is then modeled as a sequence of pose features of the associated key poses. Gesture recognition is achieved through sequential pattern recognition, e.g. using the HMM or the conditional random field (CRF) by treating the key pose as the states and pose features as the observations.

Compared to the holistic approaches, sequential approaches are more powerful in capturing and modeling varying movement dynamics in gestures, and in processing continuous incoming gesture data for gestural spotting. Varying execution speeds of the same gesture are common across different subjects, sometimes even within a single gesture execution. Such speed variation can be very well represented by state transition probabilities in an HMM. In contrast, holistic approaches are limited in representing such variations. Extracting gesture features invariant to temporal variations is challenging. Moreover, when processing continuous incoming gesture data, holistic approaches need to first identify the gesture boundaries prior to recognition, which will introduce extra delay to online gesture spotting. On the other hand, the sequential approaches are capable of simultaneously finding the gesture boundaries and computing the probabilities of the gestures for every incoming frame of data in a recursive manner. Therefore, template-based sequential gesture recognition is more suitable for practical HCI systems.

In this chapter, a sequential gesture classification method based on hidden Markov models (HMM) is used, and invariant pose features, as discussed in Chapter 2, are used as observations of HMMs. State-of-the-art results on a public dataset have been obtained using the proposed method. This method can be extended to online gesture spotting, which is discussed in the next chapter.

4.2 An Introduction to The Hidden Markov Model

Currently, hidden Markov models (HMMs) [111] and other related state-based probabilistic models comprise the state of the art in sequential data processing such as speech recognition, handwriting recognition and gesture recognition. Using state-based probabilistic modeling of the gestures, such models provide a robust and accurate framework for many kinds of pattern-based analysis.

4.2.1 Original Left-to-Right HMM

Figure 4.1 shows a traditional left-to-right HMM commonly applied in gesture recognition [2]. It is composed entirely of emitting states, except for the starting and end states s_b and s_e. In a model with n emitting states $E_1 \ldots E_n$, the states are enumerated as $S_0 = s_b$, $S_1 = E_1$, $S_2 = E_2$, …, $S_n = E_n$, $S_{n+1} = s_e$. To specify the transition probabilities between each pair of states, the traditional left-to-right HMM uses a transition probability matrix:

$$T(S_i, S_j) = \begin{cases} 0, & \text{if } j < i \text{ or } i = n+1 \\ t_{i,j}, & \text{otherwise} \end{cases} . \tag{4.1}$$

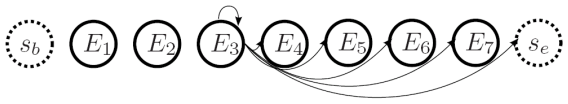


Fig. 4.1. (Extracted from [2]) Transitions for a traditional HMM model with 7 emitting states. Only transitions out of one state (E_3) are displayed for clarity. In general, any state can have a non-zero probability specified for transitions to itself or to any state depicted to its right.

4.2.2 Parameter-Reduced HMM

When applying the traditional HMM discussed above to gesture recognition, a large number of parameters are required to be trained in order to give satisfying recognition results. In particular, an n state HMM requires $O(n^2)$ parameters to be trained for the transition probability matrix, which limits its usability in environments where training data is limited.

In [2] a variation on HMMs has been proposed which reduces the number of parameters required to infer all transition probabilities to $O(n)$. In addition, our proposed model reduces

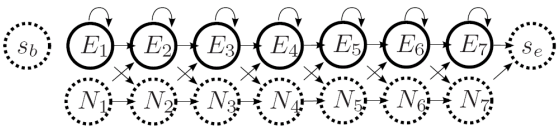


Fig. 4.2. (Extracted from [2]) Transitions for the reduced parameter models with 7 emitting states. For simplicity, transitions from the beginning state s_b are omitted. The displayed transitions are determined differently for each of the reduced parameter models.

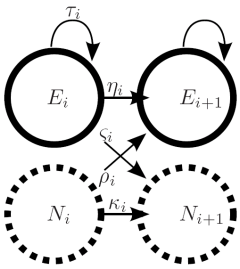


Fig. 4.3. (Extracted from [2]) Transition probability parameters for the reduced model.

the computational complexity of the inference algorithm, permitting the number of states used in the model to increase significantly while preserving real-time applicability.

The reduced model uses a constant number of parameters for each state to determine transition probabilities between all states. The parameters correspond to the probabilities of transitions depicted in Figure 4.2. In this case, the following parameters are used:

- τ_i is the probability of remaining in an emitting state ($T(E_i, E_i)$)
- η_i is the probability of going to the next emitting state ($T(E_i, E_i + 1)$)
- ς_i is the probability of skipping at least one emitting state ($T(E_i, N_{i+1})$)
- κ_i is the probability of skipping an additional emitting state ($T(N_i, N_{i+1})$)
- ρ_i is the probability of ending a skip sequence ($T(N_i, E_{i+1})$)

The parameters pertaining to an individual pair of emitting / non-emitting states is shown in Figure 4.3. Note that given these parameters for each state, an entire transition probability matrix that would correspond to the transition matrix used in the standard model can be constructed

(focusing on the transitions between emitting states):

$$T_{standard}(E_i, E_j) = \begin{cases} 0, & \text{if } j < i \\ \tau_i, & \text{if } j = i \\ \eta_i, & \text{if } j = i+1 \\ \varsigma_i \prod_{k=i+1}^{j-2} \kappa_k \rho_{j-1}, & \text{if } j > i+1 \end{cases} . \tag{4.2}$$

Since we constrain each state's outgoing probabilities to sum to 1, we note the following constraints:

$$\tau_i + \eta_i + \varsigma_i = 1 \tag{4.3}$$

$$\kappa_i + \rho_i = 1 \tag{4.4}$$

Hence, determining three of the parameters for each state is sufficient to construct all probabilities of transition coming out of a state. The reduced HMM can also be trained using the EM algorithm. Since the number of transitional parameters has been reduced to $O(n)$, the computational complexity for HMM training has been greatly reduced [2]. Due to the same reason, reduced HMM can be trained using less training samples than those needed for standard HMM.

4.3 Modeling and Recognizing Gestures Using HMM

In order to model gestures, the reduced-parameter HMMs introduced in the previous section are used. Each emitting state is modeled as a Gaussian mixture with diagonal covariance matrix. Assume that there are N gestures in the gesture vocabulary $\mathcal{G}$. For each gesture g to be recognized, a corresponding HMM with model parameter set Λ_g is learned using the EM algorithm from the associated training samples. These training samples were manually segmented from the training movement pieces. Once these gesture models are learned from training data, they can be directly used to classify pre-segmented movement data. For a pre-segmented movement piece of an unknown gesture, the corresponding pose feature sequence $\mathbf{O} = \{O_1, O_2, \ldots, O_t\}$ can be obtained. The pre-segmented data can then be classified to be gesture g^* based on the maximum likelihood principle, i.e.,

$$g^* = \arg\max_{g \in \mathcal{G}} p(\mathbf{O}|\Lambda_g). \tag{4.5}$$

The number of states in HMMs can be determined by cross validation and linear search.

4.4 Experimental Results

Tests of pre-segmented gesture recognition were conducted on IXMAS data set discussed in Chapter 2. This dataset has been used to evaluate a few state-of-the-art view-invariant gesture recognition and spotting algorithms [60], [79] and [33]. To be consistent with [60] and [79], in this experiment only data from 10 subjects and 11 actions were used (Table 4.3) and these subject and action sets are identical to those in [60] and [79].

In experiments described in this section, methods for key pose selection and training data formation for feature extraction are the same as described in Chapter 2.

4.4.1 Training and Testing Schemes

To obtain a good picture of the system performance, following [60] and [79] the gesture recognition method discussed in this chapter is evaluated through cross-validation. In each training and testing cycle, the movement data and associated pose features of nine of the ten subjects in the IXMAS dataset were used as the training data and those of the remaining subject were then used for testing. This procedure was repeated ten times so that each subject will be used once as the testing subject. The final results reported were based on the cumulative results obtained in all ten training-testing cycles.

Following [60] and [79], in this experiment a corresponding element movement was selected for each action as the corresponding representative action signature. For example, for the "check watch" action, the "raise hand" motion was selected as the representative action signature. The set of the 11 action signatures then became the gesture vocabulary for this experiment. All the movement segments corresponding to the action signatures were manually identified from the IXMAS dataset and used as training and ground-truth data. They are referred to as the *gesture segments* in this section.

In each training-testing cycle, on average about 283 gesture segments were used to train the HMMs for the 11 gestures in the gesture vocabulary. To be consistent with the experimental procedure reported in [60] for fair performance comparison, for gestures executed multiple times in a movement trial (three trials for each subject are included in the IXMAS dataset),

TABLE 4.1
NOTATIONS FOR PERFORMANCE EVALUATION USING PRE-SEGMENTED DATA

Notation	Definition
G	The number of gesture classes in a gesture set
N_i	The number of testing samples belonging to the ith gesture
C_i	The number of samples belonging to the ith gesture that are correctly recognized
F_i	The number of samples belonging to other gestures that are wrongly recognized as the ith gesture
$RR_i = \frac{C_i}{N_i}$	Gesture recognition rate of the ith gesture
$FAR_i = \frac{F_i}{\sum_{j \neq i} N_j}$	False alarm rate of the ith gesture
$RR = \frac{\sum_{i=1}^{G} C_i}{\sum_{i=1}^{G} N_i}$	The overall gesture recognition rate
$FAR = \frac{1}{G} \sum_{i=1}^{G} FAR_i$	The overall false alarm rate

only one of them was (randomly) selected to be included in the training set.

The testing data were from the gesture segments of the remaining testing subject. To be consistent with [60], for gestures executed multiple times in a single movement trial, only one of them was used in testing. On average, about 31 testing movement segments were used in each testing cycle.

4.4.2 Evaluation Criteria

To evaluate the gesture recognition results using the pre-segmented testing data, the corresponding recognition rates *RR* and the false alarm rates *FAR* have been calculated. The *RR* and *FAR* for both the individual gestures and entire gesture vocabulary are defined in Table 4.1.

4.4.3 Selection of Pose Feature Type

As shown in the previous chapter, view-invariant pose features extracted using multilinear analysis and view and shape-invariant features extracted using MGP perform almost equally

TABLE 4.2
COMPARISON OF GESTURE CLASSIFICATION RESULTS OF TWO TYPES OF POSE FEATURE IN THE TRIAL EXPERIMENT

Feature	*RR*
View and Shape-Invariant MGP Feature	78.9%
View-invariant Multilinear Feature	**91.6%**

well in pose recognition. As discussed in Section 2.4, extracting view and shape-invariant MGP features is very time consuming. Furthermore, in the cross-validation scheme in this gesture classification test, all view and shape-invariant pose features need to be recomputed in each training-testing cycle since key poses performed by all training subjects (different in each cycle) need to be applied for training for feature extraction. Therefore, to select a proper type of pose feature for gesture classification, a trial experiment was first carried out performing only the first three training-testing cycles (using first three subjects as testing subjects). The recognition rates using two types of features are compared in Table 4.2. It can be seen that although view and shape-invariant MGP features perform a little better in pose recognition, in gesture classification, view-invariant multilinear features perform much better. Based on this fact, in experiments of gesture classification and gesture spotting discussed in this dissertation, only view-invariant features obtained from multilinear analysis are applied.

4.4.4 Gesture Classification Results

Using view-invariant multilinear features, the complete gesture classification test results are presented in Table 4.3. The confusion matrix of gesture classification is presented in Figure 4.4.

To demonstrate the superiority of the proposed method, the method is first compared with a simple discrete method. In this method, for each volumetric frame, the volumetric distance measure of equation (2.42) is applied between that frame and each of the 25 key poses, and the ID of the key pose providing the best similarity is selected to represent each input frame. Then HMMs with the same structure as the proposed method but with discrete states are applied for gesture modeling and recognition. The results of the discrete method are shown in Table 4.4.

Results on gesture classification using pre-segmented data from the IXMAS dataset have

TABLE 4.3
GESTURE CLASSIFICATION RESULTS

Gesture	*RR*	*FAR*
Check watch	82.76%	0.70%
Cross arms	90%	1.05%
Scratch head	90.91%	0.34%
Sit down	100%	1.05%
Get up	100%	1.40%
Turn around	100%	0%
Walk	100%	0%
Wave hand	92.31%	1.38%
Punch	96.4%	0%
Kick	100%	0%
Pick up	86.67%	0%
Overall	**94.60%**	**0.54%**

GTG \ RG	1	2	3	4	5	6	7	8	9	10	11
1	82.8	3.4	0	0	0	0	0	13.8	0	0	0
2	6.7	90	0	0	3.3	0	0	0	0	0	0
3	0	4.5	90.9	0	4.5	0	0	0	0	0	0
4	0	0	0	100	0	0	0	0	0	0	0
5	0	0	0	0	100	0	0	0	0	0	0
6	0	0	0	0	0	100	0	0	0	0	0
7	0	0	0	0	0	0	100	0	0	0	0
8	0	0	3.8	0	3.8	0	0	92.3	0	0	0
9	0	3.6	0	0	0	0	0	0	96.4	0	0
10	0	0	0	0	0	0	0	0	0	100	0
11	0	0	0	10	3.3	0	0	0	0	0	86.7

GTG = Ground Truth Gesture RG = Recognized Gesture

Fig. 4.4. Confusion matrix of pre-segmented gesture recognition (in percentage).

been reported in [60] and [79]. The comparison of results obtained using method discussed in this chapter against those reported in [60] and [79] is shown in Table 4.4. It can be seen that our proposed framework using view-invariant features performed slightly better than the methods in [60] and [79].

TABLE 4.4
COMPARISON OF GESTURE CLASSIFICATION RESULTS ON IXMAS DATASET

Method	*RR*	*FAR*
Weinland 3D [60]	93.33%	$\sim$ 0.67%
Weinland 2D [79]	81.3%	$\sim$ 1.97%
Discrete method	35.35%	6.49%
The proposed method	**94.60%**	**0.54%**

Chapter 5

ONLINE GESTURE SPOTTING USING HMM NETWORK

In order for an HCI system to quickly respond to gestural commands in real time, the gesture recognition system should be able to accurately detect and recognize gestures from a continuous incoming movement data stream with minimum delay. This task is often referred to as **gesture spotting** in the literature. In this chapter, an approach to spotting of gestures online from a continuous movement data stream is discussed.

5.1 Overview of Online Gesture Spotting

In an HCI system, an effective **online** gesture spotting framework is often desired. Online gesture spotting requires that at the current time instant the gesture spotting decision is made based only on the current and past data without using any future movement data. Online gesture spotting presents a critical research issue due to the following reasons. In many gesture-driven HCI scenarios, the reaction time of the system needs to be short, i.e., immediate response of the system is desired once a gesture command is issued. Therefore, the gesture spotting system is required to perform real-time gesture recognition using observation data up to the current time instant. Moreover, due to the real-time interactive nature of HCI systems, once an HCI system has responded to the user's command based on the gesture spotting results in various forms of visual/aduio feedback, it is nearly impossible for the system to make any correction to such issued feedback if the gesture spotting system realized the previous result was wrong based on the afterwards newly available data. Therefore reliable online gesture spotting is a very practical pressing challenge for gesture-driven HCI systems.

Existing pattern spotting methods include DTW [112], HMM [111] and conditional models such as the maximum entropy Markov models (MEMM) [113] and CRF [57, 114]. DTW was originally designed to evaluate the similarity between two presegmented data sequences. Successful applications of DTW include recognition of speech [115, 116], music [117] and hand gesture [118, 119]. Variants of DTW have also been developed for gesture spotting. For example, in [52] DTW was successfully adapted and applied in America sign language spotting.

Compared to CRF and HMM, the biggest limitation of DTW is its lack of expressive modeling of system dynamics, which results in its limited ability to represent gesture variations.

Conditional models such as the MEMM [113] and the CRF [114] have recently been applied to pattern spotting with encouraging results, e.g. [57]. MEMM and CRF are discriminative state-based models describing the conditional probability of the state sequence given the observation sequence. MEMM and CRF have been claimed to be superior over generative models such as HMMs because of their improvement in observation dependencies [114, 120]. Flexible features using both the past and future observations are applied in these models to explicitly represent long-distance interactions and dependency. Such flexible and long-distance dependency may result in more natural models of sequential data than those based on HMM. However, MEMM and CRF have their own limitations. It is well known that MEMM suffers from the *label bias problem* [114]. For CRF, its training procedure is much more computationally expensive and converges much slower than those of HMM and MEMM [114, 120]. In addition, the scalability of CRF is also a problem. CRF builds a unified model consisting of all the patterns to be recognized. As a result, adding new patterns will require retraining the entire model and the previously trained model has to be discarded.

HMM [111] is a generative state-based framework widely applied in sequential pattern analysis. Using HMM, a sequence of observations (e.g., pose feature vectors) is modeled as being emitted from a sequence of hidden states. There are a number of HMM-based gesture spotting systems [20, 51, 55, 56, 121]. In these systems, each gesture is represented by an individual HMM. In addition, one or more HMMs are used to capture the non-gesture movement. These HMMs are parallell connected in an HMM network [51, 55] for gesture recognition. Although the long-distance interaction is not explicitly described in HMM as it is in MEMM and CRF, this actually does not present a problem for online gesture spotting, where a decision needs to be made immediately upon the arrival of the data. Without explicitly encoding long-distance interaction could potentially enforce the HMM to extract the most information from the past and current observation, which is actually in favor of online gesture spotting. To the contrary, because of its ability to encode long-distance interaction, decision making in CRF and MEMM might rely more on future data than on the past and current data. Consequently, the performance

of online gesture spotting using CRF and MEMM might deteriorate when the recognition decision has to be made on-the-fly without any future data. Therefore, compared to MEMM and CRF, HMM is more suitable for online gesture spotting.

Although promising results have been obtained using HMM-based approaches, how to effectively detect and model non-gesture movement remains a challenge. Many existing gesture spotting systems [20,51,55,56,121] make use of non-gesture models in HMM networks to provide thresholds for rejecting non-gestural movements. In [55, 56, 121], a threshold model was built as a weak universal movement model trying to represent all the gestures, which provides an adaptive thresholding mechanism for rejecting non-gesture movement. In [20], the threshold model was simplified into a single garbage gesture model which contains one emitting state with flat emitting probability over all the observations. In [51], two general garbage models were trained to represent non-gestural movement patterns. Although reasonable results have been reported, when the testing data includes complex non-gesture movement patterns resembling portions of the gestures, using one or two threshold or garbage models will not be able to effectively reject such non-gesture movement. Therefore, effective online gesture spotting in general movement sequences is still a challenging problem.

In this chapter, an approach to online gesture spotting based on HMM network is introduced. In order to tackle aforementioned challenges, a systematic approach is developed to detect and model non-gestural movement patterns automatically from continuous training data. By including specific non-gesture models in an HMM network, its ability of rejecting non-gesture movements is greatly improved without significantly sacrificing the ability of spotting true gestures.

5.2 HMM Network for Gesture Spotting

Using the pose descriptors obtained through multilinear analysis as observation vectors, gestures can be spotted from a continuous movement stream by using an HMM network [51, 55]. As illustrated in Figure 5.1, an HMM network is formed by connecting a set of movement HMMs together using a non-emitting starting state and a non-emitting end state. The HMMs involved in the HMM network for gesture spotting include gesture models, non-gesture models

and a general garbage gesture model. The gesture models are the same as the models trained for pre-segmented gesture recognition, as described in Chapter 4. The non-gesture models and the garbage gesture model are trained to represent the specific non-gesture movement patterns.

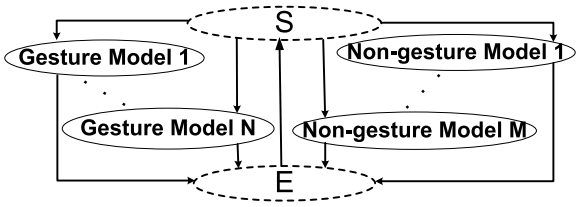


Fig. 5.1. The HMM network applied in gesture spotting.

5.3 Model Learning

As discussed above, the gesture models in the HMM network can be learned in the same way as in Chapter 4.

A garbage gesture model is also defined to model general non-gesture movement sequences. It has a single emitting state, with a flat probability distribution function over the entire observation space. The single state can loop back to itself (non-gesture continues), or exit (non-gesture ends). Applying this garbage model is equivalent to applying a normalized log-likelihood constraint to spotted gestures, since the log-likelihood of a non-gesture segment is just the log of the flat probability value times the length of the segment.

In complex testing scenarios, for example, when the testing data includes non-gesture movement patterns resembling portions of the gestures, using one or two threshold or garbage models will not be able to effectively reject such non-gesture movement. To tackle this challenge in the proposed gesture spotting framework, in addition to a general garbage gesture model, a number of specific non-gesture models are deployed to further improve the gesture spotting ability of the proposed system. These specific non-gesture models include automatically identified and manually specified models. These non-gesture models are aimed to represent specific non-gesture movement patterns observed from the training data and then reject these non-gesture movement patterns in gesture spotting. Some non-gesture movement patterns are manually picked, including

- repetitive inter-gesture patterns (such as stand still),
- false gestures that are similar to the true ones,
- movement patterns shared by two or more gestures.

These movement patterns can be commonly found in the training movement pieces. They can be manually segmented and used as the training samples of corresponding non-gesture models. HMMs for these manually selected non-gesture patterns can be trained in the same manner as the gesture models.

To improve the efficiency of non-gesture model detection and learning, a systematic approach to autonomous non-gesture model detection and learning from training data has also been developed, which is illustrated in Figure 5.2 (a). Key steps for the non-gesture movement detection and learning approach are as follows. First, the training movement sequences are automatically segmented into element pieces by finding the minima of the motion energy, which is defined in Section 2.3.3. Element pieces with large overlap with the training segments used for gesture models and manually specified non-gesture models (if there are any) are eliminated, leaving only unused element pieces corresponding to remaining non-gesture movement patterns. Then, a similarity matrix is calculated for the element pieces. The similarity between two element pieces is calculated using dynamic time warping (DTW) [112], and the distance between two observations is defined to be the Euclidean distance. Based on the similarity matrix, the non-gesture training data is grouped into a number of clusters using normalized-cut [80], one for each automatically detected non-gesture model. The number of clusters is essentially the number of detected non-gesture models. This number is preset manually. Finally, the element pieces similar enough to the cluster centers (similarity value exceeds a pre-chosen threshold) are applied as training samples for the corresponding non-gesture model. By applying this automatic model training scheme, the number of non-gesture models can be flexibly controlled according to the actual requirement of a specific application.

After model training, a collection of HMMs is available for gesture spotting, including HMMs for the gesture vocabulary $\mathcal{G}$, and the non-gesture set $\mathcal{F}$ corresponding to the gen-

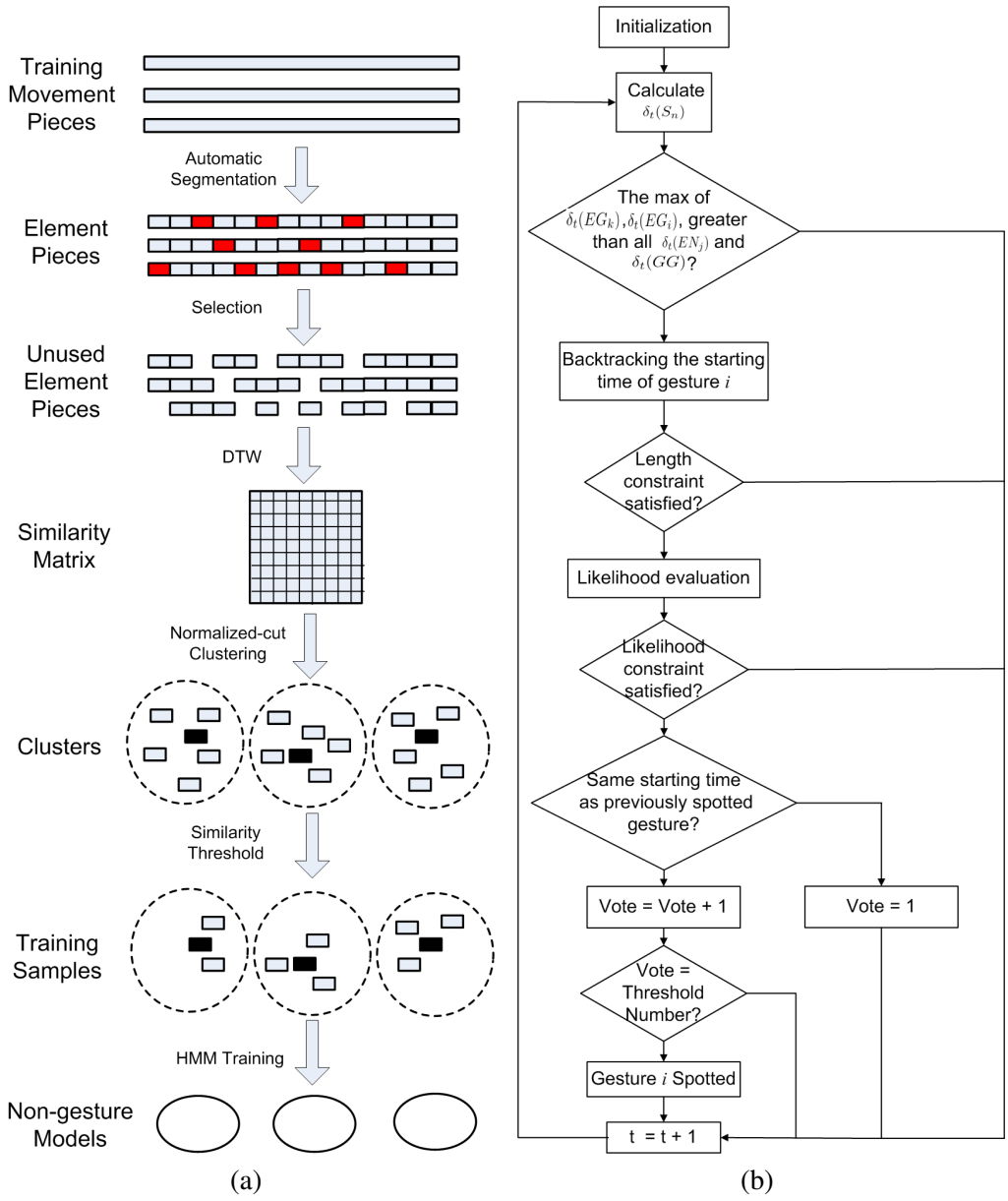


Fig. 5.2. (a) Automatic detection and training of non-gesture movement patterns. (b) The flowchart of the gesture spotting algorithm

eral garbage gesture model, the manually selected non-gesture models, and the automatically detected non-gesture models.

5.4 Gesture Spotting

Using both gesture and non-gesture movement models, gesture spotting can be achieved by evaluating the joint probability of the observation sequence and the path of hidden state

transition in an HMM network.

For online gesture spotting, an observation o_t is fed into the HMM network at each time instant t. Let q_t be the hidden state at time instant t, S_n the nth hidden state in the HMM network, $\mathcal{S}$ the collection of all the states of the HMM network, and $\mathbf{O}_t$ the observation sequence from the beginning of the movement piece up to time instant t. Define

$$\delta_t(S_n) = \max_{q_1,\ldots,q_{t-1}} p(\mathbf{O}_t, q_1, \ldots, q_{t-1}, q_t = S_n|\Lambda) \tag{5.1}$$

to be the probability of the optimal state path to the current state S_n given the current observation sequence. In (5.1), Λ denotes the parameter set of the entire HMM network, including the model parameter sets of both the gesture and non-gesture models. Using the Viterbi algorithm, $\delta_t(S_n)$ can be computed based only on the information of the previous time instant, $\delta_{t-1}(S), \forall S \in \mathcal{S}$, and the current observation. Therefore, the evaluation of $\delta_t(S_n)$ can be performed in an incremental manner.

As described in Section 4.2.2, each gesture model contains a non-emitting end state. Reaching this end state means that the corresponding gesture has been fully executed. Let E_h be the end state of HMM h. At time instant t, if the end probability of a gesture g^* is the largest among all the gestures and non-gestures, g^* is spotted, i.e.,

$$g^* = \arg\max_{h \in \mathcal{G} \cup \mathcal{F}} \delta_t(E_h) \text{ and } g^* \in G \tag{5.2}$$

Once a gesture is detected, the starting time instant of the gesture can also be easily found by backtracking the most probable path.

This preliminary spotting result is further refined according to the length of the spotted gesture segment and the corresponding likelihood. A length constraint and a likelihood constraint are set up to reject outliers. In this experiment, the length of a spotted gesture was constrained to be less than 50 frames (since the length of ground truth gesture segments are in the range of 10 to 35 frames) and the likelihood of the segment given recognized gesture was constrained to be larger than 10^{-70} (the majority of the likelihood are in the range of 10^{-15} to 10^{-40}). Movement segments satisfying the length and likelihood constraints are considered to be gesture segment candidates. Moreover, temporal consistency is also used to stabilize the gesture spotting results. To be specific, only when gesture segment candidates sharing the same starting frame are

continually detected T times without any other candidates detected in the middle, where T is a prechosen threshold, then a final online spotting decision will be made. The overall gesture spotting scheme is summarized using a flowchart in Figure 5.2 (b).

Given a continuous stream of movement data, the final gesture spotting result is a series of recognized gesture segments, including the gesture label, the beginning and end frame numbers of the segment, and the corresponding likelihood.

5.5 Experimental Results

Gesture spotting experiments were also conducted on IXMAS dataset. The subjects and gesture vocabulary applied were also the same.

5.5.1 Training and Testing Scheme

Training and testing for gesture spotting follows the same cross-validation manner as described in the experimental section of Chapter 4. In each training-testing cycle, training of gesture models is also the same as in Chapter 4. For gesture spotting, non-gesture models also need to be trained besides gesture models. When manually specified non-gesture models were used, their corresponding movement segments of the same nine training subjects were hand-picked from the dataset to train the associated HMMs. Likewise, the HMMs of the automatically detected non-gestures were trained by applying the training samples obtained according to the method presented in Section 5.3.

To test gesture spotting from a continuous data stream, in each training-testing cycle the three complete movement trials of the testing subject were used as testing data. All the gesture and non-gesture models were integrated in the HMM network presented in Section 5.4 for gesture spotting. The manually identified gesture segments from these movement trials were then used as ground-truth to evaluate the gesture spotting results from the proposed system.

TABLE 5.1
NOTATIONS USED IN PERFORMANCE EVALUATION OF GESTURE SPOTTING

Notation	Definition
N_T	The number of true gesture segments in the testing data
N_C	The number of gesture segments that are correctly detected and recognized
E_S	The number of gesture segments that are detected but misrecognized (substitution error)
E_M	The number of gesture segments that are missed (i.e. not detected)
E_T	The number of gesture segments that are not correctly spotted, $E_T = N_T - N_C$
E_I	The number of insertion errors
T_I	The cumulative number of frames of all insertion errors
T_{NG}	The cumulative frame numbers of non-gesture segments

5.5.2 Evaluation Criteria

In gesture spotting using continuous testing data, for each input testing stream the algorithm discussed in this chapter returns a series of spotted gesture segments, including the label of the recognized gesture, the beginning and end frame numbers of the segment, and the corresponding likelihood. Due to gesture spotting errors, a spotted gesture might not be a true gesture. To evaluate the performance of the discussed approach, such gesture spotting results were compared against the ground-truth gesture segment data and analyzed in a number of aspects, including temporal matching accuracy, recognition and false alarm rates, and reliability of recognition. In this section, the method to measure the temporal matching accuracy of a spotted gesture segment is first introduced. Then, other performance indicators derived from gesture spotting results and ground-truth data are discussed.

Let $F_b(i)$ and $F_e(i)$ be the beginning and end frame numbers of the ith true gesture segment in the testing data. The length of this segment is thus

$$L_{GT}(i) = F_e(i) - F_b(i) + 1. \tag{5.3}$$

Let $S_b(i)$ and $S_e(i)$ be the beginning and end frame numbers of a spotted gesture segment. Define the ***absolute temporal matching score*** $O_A(i)$ as the number of the overlapped frames

between the spotted and ground-truth gesture segments

$$O_A(i) = \min\{S_e(i), F_e(i)\} - \max\{S_b(i), F_b(i)\}, \tag{5.4}$$

and the ***absolute temporal matching score*** $O_R(i)$ be the ratio of $O_A(i)$ to the length of the true segment.

$$O_R(i) = \frac{O_A(i)}{L_{GT}(i)}, \tag{5.5}$$

When $O_R(i)$ is larger than a pre-chosen threshold η, $(0 < \eta \leq 1)$, the spotted gesture is considered to be temporally matched to the ground-truth segment. In this experiment, the default value of η was 0.5. Results were also obtained using different values of η and examined how different values of η can affect different performance measures.

Once a spotted gesture segment is temporally matched to a ground-truth gesture segment, their gesture labels are compared. If they share the same gesture label, the spotted segment is then considered to be correctly recognized and the number of correctly spotted segments N_C will increase by 1. Otherwise, the spotting gesture segment will be counted as a substitution error. If a gesture segment is not temporally matched to any ground-truth gesture segment with respect to η, it will be treated as an insertion error. On the other hand, if a ground-truth gesture segment was not matched to any spotted gesture segment, it is then counted as a missing error. Notations of these performance indicators are summarized in Table 5.1. It is easy to see that the substation and missing errors constitute the overall errors that can happen to the true gesture segments. When a true gesture segment is not correctly spotted, there are two possibilities: it is either not detected or detected but misrecognized. Therefore $E_T = N_T - N_C = E_S + E_M$.

By using these performance indicators, additional performance measures including the recognition (spotting) rate *RR*, the reliability measure *RL*, and the false alarm rate *FAR* can be further derived and evaluated. In addition to measuring the numbers of correctly and incorrectly spotted gesture segments, the overall temporal matching accuracy of correctly spotted gestures was also measured. The corresponding indicators are average overlapping $\bar{O}_A$, average relative overlapping $\bar{O}_R$, average beginning delay $\bar{B}_A$, average absolute beginning delay $\bar{B}_{AB}$, average relative beginning delay $\bar{B}_R$, average end delay $\bar{E}_A$, average absolute end delay $\bar{E}_{AB}$, and average relative end delay $\bar{E}_R$. All the performance measures used in experiments in this section are detailed def-

initions in Table 5.2. By using these performance measures, a clear picture of the performance of the proposed gesture spotting algorithm can be obtained as shown in the next section.

TABLE 5.2
PERFORMANCE INDICATORS FOR GESTURE SPOTTING (SUMMATIONS ARE OVER ALL THE CORRECTLY SPOTTED GESTURE SEGMENTS)

Indicator	Notation	Definition
Recognition Rate	RR	$\frac{N_C}{N_T}$
Reliability	RL	$\frac{N_C}{N_C+E_S+E_I}$
False Positive Rate	FAR	$\frac{T_I}{T_{NG}}$
Average Overlapping	$\bar{O}_A$	$\frac{1}{N_C}\sum O_A(i)$
Relative Overlapping	$\bar{O}_R$	$\frac{1}{N_C}\sum \frac{O_A(i)}{L_{GT}(i)}$
Average Beginning Delay	$\bar{B}_A$	$\frac{1}{N_C}\sum (S_b(i)-F_b(i))$
Average Absolute Beginning Delay	$\bar{B}_{AB}$	$\frac{1}{N_C}\sum \lvert S_b(i)-F_b(i)\rvert$
Average Relative Beginning Delay	$\bar{B}_R$	$\frac{1}{N_C}\sum \frac{(S_b(i)-F_b(i))}{L_{GT}(i)}$
Average End Delay	$\bar{E}_A$	$\frac{1}{N_C}\sum (S_e(i)-F_e(i))$
Average Absolute End Delay	$\bar{E}_{AB}$	$\frac{1}{N_C}\sum \lvert S_e(i)-F_e(i)\rvert$
Average Relative End Delay	$\bar{E}_R$	$\frac{1}{N_C}\sum \frac{(S_e(i)-F_e(i))}{L_{GT}(i)}$

5.5.3 Gesture Spotting Results

To examine the impact of using specific non-gesture models in gesture spotting, gesture spotting has been conducted with different model configurations. In the beginning, only the gesture models (GM) and the general garbage gesture models (GGM) were used. Then, automatically specified non-gesture models (ANGM) were gradually added to the HMM network for gesture spotting. Finally, manually specified non-gesture models (MNGM) were included.

The gesture spotting accuracy and temporal matching accuracy ($\eta = 0.5$) using various model configurations are given by Table 5.3 and Table 5.4, respectively. From Table 5.3 it can be seen that the inclusion of more specific non-gesture models greatly reduced the insertion errors without significantly diminishing correct recognitions. Consequently, the reliability of the spotted gestures also greatly increased. It can also be seen Table 5.3 that when the number of ANGMs increased, adding MNGMs only slightly improved the spotting accuracy. From Table 5.4 we can see that different gesture model configurations had only very slight impact on the temporal matching accuracy of the spotted gesture segments. The measures of the temporal matching accuracy obtained using various model configurations are all at reasonable levels.

The influence of the temporal matching threshold parameter η on the performance measures of gesture spotting was also examined. Different values of η were applied to derive the corresponding performance measures. The two HMM network configurations using 15 ANGMs and with and without MNGMs, i.e. MNGM+15ANGM+GM+GGM and GM+GGM+15ANGM, were tested and the corresponding results are shown in Table 5.5. It can be seen from Table 5.5 that when η decreased, both the recognition rate and recognition reliability increased and and at the same time the insertion error and the false alarm rate were reduced. This is because when η is low, more spotted gesture segments can be considered to be matched to the corresponding true gesture segments.

Another observation can be made from Table 5.5 is that the value of η affected the error distribution between the substitution error and the missing error. As mentioned early, these two types of errors constitute the overall errors that can happen to the true gesture segments, i.e., $E_T = E_S + E_M$. It can be seen from Table 5.5 that when η was decreasing, both E_T and E_M were

TABLE 5.3
GESTURE RECOGNITION ACCURACY ($\eta = 0.5$)

Models used in the HMM network	N_T	N_C	E_S	E_M	E_I	*RR*	*RL*	*FAR*
GM+GGM	539	451	68	20	511	83.67%	43.79%	30.44%
5ANGM+GM +GGM	539	446	57	36	344	82.75%	52.66%	20.93%
10ANGM+GM +GGM	539	441	53	45	292	81.82%	56.11%	18.20%
15ANGM +GM+GGM	**539**	**436**	**41**	**62**	**207**	**80.89%**	**63.74%**	**12.85%**
MNGM+GM +GGM	539	437	41	61	279	81.08%	57.73%	18.37%
MNGM+5ANGM +GM+GGM	539	441	41	57	226	81.82%	62.29%	15.42%
MNGM+10ANGM +GM+GGM	539	437	36	66	194	81.08%	65.52%	13.07%
MNGM+15ANGM +GM+GGM	**539**	**432**	**33**	**74**	**156**	**80.15%**	**69.57%**	**10.16%**

decreasing while E_S was increasing. This is because reducing η allowed more spotted gestures to be temporarily matched to true gestures (thus reducing E_M). On the other hand, some of the newly matched spotted gestures did not share the same gesture label with the true gesture segment, which led to increased E_S.

Gesture spotting results using continuous streams from the IXMAS dataset have been reported in [60] and [33]. In this section, gesture spotting results obtained using discussed method is also compared against those reported in [60] and [33].

In [60], Weinland et al. reported experiments and results comparable to gesture spotting. The result comparison is shown in Table 5.6. It can be seen that the proposed method evaluated using different η consistently achieved higher recognition rates and lower false positive rates than the results reported in [60].

An important point about this comparison that needs to be made is that the way to evaluate the gesture spotting accuracy in this dissertation is much stricter and more complete than that used in [60]. Different from this experiment where gestures were spotted directly from a con-

TABLE 5.4
TEMPORAL MATCHING ACCURACY OF CORRECTLY RECOGNIZED GESTURES ($\eta = 0.5$)

Models used in the HMM network	$\bar{O}_A$	$\bar{O}_R$	$\bar{B}_A$	$\bar{B}_{AB}$	$\bar{B}_R$	$\bar{E}_A$	$\bar{E}_{AB}$	$\bar{E}_R$
GM+GGM	14.35	0.84	1.20	2.42	0.060	0.63	3.49	0.10
5ANGM+GM +GGM	14.29	0.84	1.14	2.46	0.056	0.57	3.47	0.09
10ANGM+GM +GGM	14.28	0.84	1.08	2.44	0.054	0.51	3.44	0.09
15ANGM +GM+GGM	**14.31**	**0.84**	**1.11**	**2.42**	**0.057**	**0.40**	**3.41**	**0.08**
MNGM+GM +GGM	14.57	0.84	1.25	2.54	0.062	0.76	3.58	0.11
MNGM+5ANGM +GM+GGM	14.58	0.84	1.19	2.55	0.058	0.90	3.66	0.12
MNGM+10ANGM +GM+GGM	14.54	0.84	1.23	2.52	0.062	0.75	3.64	0.11
MNGM+15ANGM +GM+GGM	**14.56**	**0.85**	**1.19**	**2.52**	**0.059**	**0.82**	**3.49**	**0.11**

TABLE 5.5
GESTURE RECOGNITION ACCURACY WITH VARIOUS η VALUES AND TWO HMM NETWORK MODELS: 15ANGM+GM+GGM / MNGM+15ANGM+GM+GGM

η	N_T	N_C	E_S	E_M	E_I	*RR*	*RL*	*FAR*
0.05	538	456/448	74/56	9/35	153/115	84.60%/ 83.12%	66.76%/ 72.37%	9.02%/ 7.25%
0.1	538	456/448	71/54	12/37	156/117	84.60%/ 83.12%	66.76%/ 72.37%	9.21%/ 7.38%
0.2	538	455/448	65/50	19/41	163/121	84.42%/ 83.12%	66.61%/ 72.37%	9.72%/ 7.65%
0.5	**538**	**436/432**	**41/33**	**62/74**	**207/156**	**80.89%/ 80.14%**	**63.74%/ 69.57%**	**12.84%/ 10.16%**
0.7	538	366/372	23/20	150/147	296/230	67.90%/ 69.02%	53.43%/ 59.81%	18.87%/ 15.38%
0.8	538	295/291	17/14	227/234	373/317	54.73%/ 53.99%	40.07%/ 46.78%	23.69%/ 21.21%

tinuous movement stream, in [60] Weinland et al. first segmented the movement stream using a segmentation algorithm based on the motion energy and then classified these movement segments as either gestures in the vocabulary or non-gesture movement segments. To compute the recognition and false alarm rates, ground-truth was obtained manually on top of the segmented

data. Therefore, in the gesture spotting results reported in [60], segmentation errors were not taken into account. For example, when the segmentation algorithm wrongly grouped two gestures, the combined segment will be treated as a non-gesture movement segment in [60] and this segmentation error will not be reflected in the gestural spotting results since the ground-truth was taken on top of the segmented data. Obviously, obtaining ground-truth purely based on the segmented data and omitting segmentation errors in the calculation of gesture recognition rate is suboptimal in evaluating a gesture spotting method. The true gesture recognition rate should be the number of correctly recognized gesture (N_C) divided by the number of actual gestures (N_T) in the given testing continuous data, which is the exact recognition rate used in the proposed approach. However, when the errors introduced by wrong segmentation are not considered, the resulting gesture recognition rate is then N_C divided by N_S, the number of correctly segmented gestures. Since N_S is always less than or equal to N_T, the recognition rate without counting the segmentation errors will be always higher than or at most equal to the actual recognition rate. In practice, these segmentation errors will surely introduce errors in gesture spotting from live continuous movement data. Even using a stricter method for computing the gesture recognition rate, it can be seen in Table 5.6 that the proposed method consistently outperformed the method in [60].

Another point worthy to be mentioned is that in [60] the false positive rate is defined based on the segmented data, as the percentage of the non-gesture segments that are classified as a gesture. In this experiment, as shown in Table 5.2, frame-wise false positive rate, defined as the ratio of the cumulative number of frames of inserted gestures and the total time of non-gesture movements, was used to evaluate the performance.

To demonstrate the advantage of the parameter-reduced HMM applied in this gesture spotting method, the gesture spotting results is also compared with the results using conventional left-to-right chain HMM model without skip in Table 5.6. It can be seen that result obtained using proposed method significantly outperforms the chain models especially in the sense of false alarm rate.

The gesture spotting results were also compared with those presented in [33]. In [33], the percentage of correctly labeled frames is used as a measure of gesture spotting accuracy. In [33],

the global optimal path of HMM states was obtained at the end of the data stream and used for performance evaluation. For fair comparison, gesture labeling results using global optimal path, in addition to results computed online, were also computed in this experiment. The result comparison is given in Table 5.7. It can be seen that if global optimal path was applied, the proposed method outperformed the method in [33]. It is clear that using the global optimal path does increase gesture spotting accuracy measured based on the correctly labeled frames. One thing worthy of mentioning is that in [33], a total of 15 actions, including the 14 actions originally in the IXMAS dataset and a "stand still" action identified from the same dataset by the authors of [33], were used for recognition. To be consistent with [60] and [79], in this experiments only ten actions were used. The comparison with [33] was done based on the data of the ten common actions used in both this experiment and [33].

Another thing to be noted is that although in [33] only one view is applied, it requires a lot of synthesized images from motion capture data. This is somewhat equivalent to applying 3D information, since the training images are generated by projecting a 3D avatar performing the poses into arbitrary view plains. Also, the authors in [33] assume the tilt angle of the camera to be known. This is a strong constraint of the experimental condition. Without this constraint, the search space of the method proposed in [33] will greatly increase. With the conditions stated above, comparison between the proposed method and the method proposed in [33] is reasonable.

TABLE 5.6
COMPARISON OF GESTURE RECOGNITION ACCURACY

Framework	*RR*	*FAR*
Weinland 3D [60]	78.79%	14.08%
Chain HMM (MNGM+15ANGM+GM+GGM, $\eta = 0.5$)	77.92%	20.18%
Chain HMM (MNGM+15ANGM+GM+GGM, $\eta = 0.2$)	83.30%	15.62%
The proposed method (15ANGM+GM+GGM, $\eta = 0.5$)	**80.89%**	**12.84%**
The proposed method (15ANGM+GM+GGM, $\eta = 0.2$)	**84.42%**	**9.72%**
The proposed method (MNGM+15ANGM+GM+GGM, $\eta = 0.5$)	**80.14%**	**10.16%**
The proposed method (MNGM+15ANGM+GM+GGM, $\eta = 0.2$)	**83.12%**	**7.65%**

TABLE 5.7
COMPARISON OF PER-FRAME ACCURACY OF GESTURE SPOTTING

Gesture	Lv and Nevatia [33]	**The proposed method, global path**	**The proposed method, online**
Check watch	82.5%	81.0%	72.9%
Cross arms	82.1%	80.6%	71.2%
Scratch head	80.2%	83.2%	55.6%
Sit down	83.7%	80.8%	85.9%
Get up	84.3%	63.5%	37.6%
Turn around	78.8%	83.5%	88.4%
Walk	79.7%	92.5%	91.7%
Wave hand	79.9%	73.9%	81.1%
Punch	86.8%	82.8%	45.3%
Kick	87.7%	95.5%	96.6%
Pick up	83.2%	92.6%	88.5%
Overall	**$\sim$81.6%**	**85.0%**	**80.2%**

Chapter 6

MULTI-CAMERA FUSION FOR GESTURE RECOGNITION

For view-invariant pose and gesture recognition, application of multiple cameras is necessary to reduce ambiguity introduced by self-occlusion. In previous chapters, data obtained from multiple cameras are combined in only two ways. In Chapter 3, when testing on dance pose data, silhouettes extracted from two uncalibrated cameras are directly concatenated. When applying IXMAS dataset in Chapters 3, 4 and 5, visual hulls obtained from five calibrated cameras are applied. In this chapter, more techniques of multi-camera fusion for gesture recognition are discussed.

6.1 Overview

Recently, multiple cameras have been deployed for robust gesture recognition to reduce ambiguity introduced by self-occlusion and to improve view-invariance, which requires a system to recognize gestures equally well independent of the facing direction of the subject. Multi-camera gesture recognition essentially belongs to homogeneous multi-sensor fusion [122, 123], and the data and information from different cameras can be fused and integrated at the data level, the feature level, or the decision level [122, 123]. Data-level fusion directly combines image data from multiple cameras. When multiple calibrated cameras (i.e., both the internal and external camera parameters are known) are used, data-level fusion has been the dominant sensor fusion scheme for multi-camera gesture recognition. For example, a typical data-level multi-camera fusion scheme for gesture recognition using calibrated cameras in [34, 60, 79, 124–126] has been to first reconstruct the 3D visual hull data of the subject from multi-view images using the shape-from-silhouette method [24], and then extract pose or gesture features from the visual hull data for gesture recognition. Data-level fusion has also been used in gesture recognition using multiple uncalibrated cameras without reconstructing 3D visual hull data. Compared to calibrated cameras, uncalibrated cameras are easy to set up and can be quickly deployed for gesture recognition in real-life scenarios. For example, in [109], images obtained from two uncalibrated cameras have been concatenated to form the integrated image observation vector for

gesture recognition. In the feature-level gesture recognition fusion scheme, features extracted independently from individual cameras are combined into an integrated feature vectors for gesture recognition. For example, in [1] and [127], spatio-temporal features have been extracted from each camera view and then integrated for gesture recognition. In decision-level fusion, gesture recognition is first conducted independently for each camera view and then the individual recognition results are combined according to various classifier integration rules [128], including the sum rule (e.g., [126]), the product rule (e.g., [1]), the maximum likelihood rule (e.g., [129]), and the majority vote rule (e.g., [130, 131]).

Unlike the case of using multiple calibrated cameras for gesture recognition, where the data-level fusion scheme has been dominant through the extraction of 3D visual hull data, despite the large body of existing research, two fundamental research problems remain largely unaddressed for gesture recognition using multiple uncalibrated cameras: a) given the *fixed set* of cameras, which *multi-camera data fusion scheme* can produce the best gesture recognition results? and, b) given the *fixed number* of cameras, how to find the optimal *camera combination* that leads to the best gesture recognition results? In the multi-camera gesture recognition literature, there has been no systematic comparison between the data-level, the feature-level, and the decision-level fusion schemes using the same image feature and benchmark gesture recognition framework on a common testing dataset. Camera selection has been discussed in the camera network research [132–138]. However, existing research on camera selection is tuned to solve specific camera network problems using customized camera selection criteria and metric and might not transfer very well to solving the camera selection problem for multi-camera gesture recognition. For example, in [132, 133], the optimal camera combination is selected so that image data captured from the selected cameras can be used to best synthesize the image from a desired target camera view. In [134–136], the optimal camera combination is found from a wireless camera network to reach a balance between computational and communication efficiency and the quality of 3D reconstruction. In [137, 138], the size, view, and position in the field of view of a tracked person have been adopted as criteria for optimal camera selection for multi-camera multi-person tracking. These existing camera selection criteria cannot effectively capture the desired relationship among collaborating cameras in multi-camera gesture recognition. For example, in the decision-level fusion scheme for multi-camera gesture recog-

nition, a requirement for effectively combining results from different cameras is that results from individual cameras should not strongly correlate with each other in their misclassifications. Furthermore, it is desired that in the case of misclassification by one camera view for a testing data, another camera view produces the correct classification. Hence, optimal camera combination for multi-camera gesture recognition should be identified according to the complementary strength within the camera combination. The key challenge is to develop a measure of complementary strength across cameras. In this chapter, the research that attempts to address these fundamental research problems for gesture recognition using multiple uncalibrated cameras is presented. To obtain valid gesture recognition comparison across various data fusion schemes, in this chapter, the publicly available IXMAS gesture recognition dataset [60] has been used as the benchmark testing and training dataset for mult-camera gesture recognition. Moreover, the same gesture recognition framework using the multilinear human pose features and the hidden Markov models (HMM) presented in [109, 125, 139] has been adopted for all multi-camera data fusion schemes as the common benchmark gesture recognition framework to ensure fair comparison.

Further analyzing these gesture recognition results has led to the following striking observations and findings. The experimental results have shown that the decision-level fusion with the product rule is the optimal multi-camera data fusion scheme consistently for all the tested camera combinations using the benchmark gesture recognition framework. To my best knowledge, systematic comparison of different data fusion schemes for multi-camera gesture recognition using a common benchmark gesture recognition framework has not been done in existing research. The consistent superiority of the product rule-based decision-level fusion scheme over the data-level and the feature-level fusion schemes across a number of key camera combinations suggests that decision-level fusion with the product rule is the most effective way to integrate information from multiple uncalibrated cameras for gesture recognition. This is the first key contribution of this chapter. This finding is compelling also because it verifies existing principles for combining classifiers in decision-level fusion in the literature [140] in the particular case of multi-camera gesture recognition, which has not been done before either.

In addition, the gesture recognition results obtained using the benchmark gesture recognition

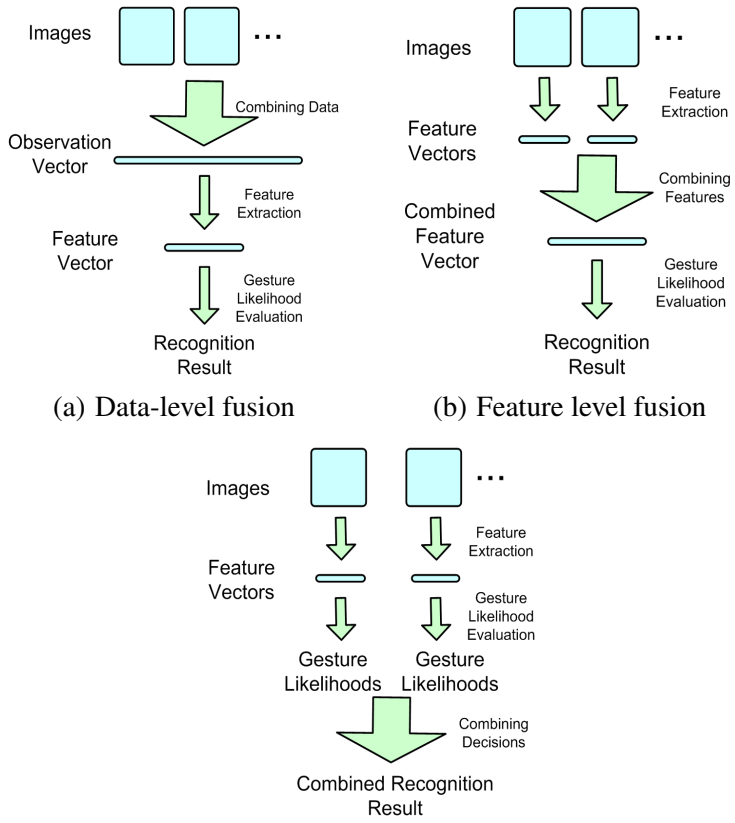
framework based on the multilinear human pose features and HMM are always better than the existing results reported on the same testing data using the same camera combination and sensor fusion scheme. This observation validates and justifies the use of the benchmark gesture recognition framework in the comparison study across data-fusion schemes reported in this chapter.

To address the challenge for selecting the optimal camera combination, as the second key contribution of this chapter, the *complementary coefficient* has been has been proposed as a simple while revealing measure of the complementary strength between cameras. Based on the complementary coefficient computed across the IXMAS cameras, it has been further identified that the optimal camera combinations for the number of camera $C = 2, 3, 4$, which indeed correspond to the best performing camera combinations according to the actual gesture recognition results on the IXMAS dataset. This observed consistency between the optimal camera combinations and their complimentary coefficients validates the the proposed approach to the complementary strength measure across cameras. Such an inter-camera complementary strength measure can find important applications in camera selection for multi-camera vision problems.

Another interesting observation from the results is that the optimal C-camera combinations for $C \geq 3$ always contain the overhead camera. This observation is striking because in traditional multi-camera gesture recognition, the overhead camera is often considered nonessential when other side-view cameras are available (e.g., [127]). The research in this chapter has revealed that using the multilinear human pose features, the overhead camera has exhibited strong complementary strength with the side-view cameras in the IXMAS dataset, which explains why the overhead camera plays a more important role using the benchmark gesture recognition framework.

6.2 The Benchmark Gesture Recognition Framework

In this chapter, the benchmark gesture recognition framework adopted is the HMM based framework described in Chapter 4. The observations of HMMs are multilinear human pose features described in Chapter 2. This benchmark framework is applied to compare the performance of various sensor fusion schemes and camera configurations.



(a) Data-level fusion (b) Feature level fusion

(c) Decision-level fusion

Fig. 6.1. General frameworks of three types of multi-camera fusion.

6.3 Multi-Camera Fusion for Gesture Recognition

Multi-camera fusion for gesture recognition using uncalibrated cameras can be achieved at the data level, the feature level, and the decision level. The block diagrams corresponding to each data fusion scheme using the proposed benchmark framework for gesture recognition are illustrated in Fig. 6.1.

6.3.1 Data-Level Fusion

The data-level fusion scheme for multi-camera gesture recognition using multiple uncalibrated cameras is illustrated by Fig. 6.1a. In this data fusion scheme, the complete observation vector for multilinear pose feature extraction is formed by concatenating vectorized foreground silhouette images from different cameras. The camera order for image concatenation is required to be fixed to secure the data consistency of the complete observation vector. Once

such complete observation vectors are obtained, pose features can be extracted from the observation vectors using the ALS method as described in Section 2.1.2, and these pose features are further applied in gesture modeling and recognition using HMM. In this data-level fusion scheme, although the camera calibration parameters are not required to be known, the camera configuration, such as the number of cameras and the relative locations and optical axis directions of these cameras, needs to be consistent between system training (including the core tensor extraction and HMM learning) and testing (including the extraction of the pose features from new observation vectors and gesture recognition using HMM). In other words, the learned core tensor and HMM models are tightly coupled with the corresponding camera configuration. Data-level fusion using a pair of uncalibrated cameras with orthogonal optical axes has been exploited in [109].

6.3.2 Feature-Level Fusion

The feature-level fusion scheme for multi-camera gesture recognition using multiple uncalibrated cameras is illustrated by Fig. 6.1b. In this data fusion scheme, the pose features are first independently extracted from individual cameras. Then, these pose features from different cameras are concatenated to form a combined complete feature vector. The camera order for feature concatenation is required to be fixed to secure the data consistency of the complete feature vector. The combined pose features are then applied in gesture modeling and recognition. Similar to the data-level fusion, in the feature-level fusion the camera configuration also needs to be consistent between system training and testing. In the feature-level fusion, since the pose features are first independently extracted from different cameras, in general each camera has its corresponding core tensor for multilinear pose feature extraction. On the other hand, when the orientations of a group of cameras differ only in the pan angle (the rotation angle about the axis perpendicular to ground plane) and share the same tilt angle, these cameras can use the same pose tensor and core tensor for pose feature extraction.

6.3.3 Decision-Level Fusion

The decision-level fusion scheme for multi-camera gesture recognition using multiple uncalibrated cameras is illustrated by Fig. 6.1c. In this data fusion scheme, system training and gesture recognition is first carried out independently for each cameras view. Then the gesture recognition results from individual cameras are combined to infer the final gesture recognition result using certain rules for combining classifiers. Let $\mathbf{O}^c$ be the pose feature sequence of a gesture segment extracted from camera c, and Λ_g^c the HMM parameters learned for gesture g in camera c, where $c = 1, \cdots, C$ and $g = 1, \cdots, G$. Given a testing gesture segment, the pose feature sequences $\{\mathbf{O}^c\}_{c=1}^C$ are first extracted from all the cameras, and then for each camera c the likelihood of all the gestures $\{p(\mathbf{O}^c|\Lambda_g^c)\}_{g=1}^G$ with respect to each pose feature sequence is evaluated according to the learned HMMs. In decision-level fusion, the following rules [140] can be applied to combine results from different cameras to determine the final recognition result g^*.

Sum Rule:

$$g^* = \arg\max_g \sum_c p(\mathbf{O}^c|\Lambda_g^c). \tag{6.1}$$

Product Rule:

$$g^* = \arg\max_g \prod_c p(\mathbf{O}^c|\Lambda_g^c). \tag{6.2}$$

Max Rule:

$$g^* = \arg\max_g [\max_c p(\mathbf{O}^c|\Lambda_g^c)]. \tag{6.3}$$

Min Rule:

$$g^* = \arg\max_g [\min_c p(\mathbf{O}^c|\Lambda_g^c)]. \tag{6.4}$$

The majority voting rule introduced in [140] is not applied in this chapter because ties of votes may easily appear especially when using even number of cameras. In decision-level fusion, the core tensor extraction and HMM learning are usually done separately for each camera. On the other hand, similar to the feature-level fusion, a group of cameras with the same tilt angle can share the same core tenor for multilinear pose feature extraction and the same set of HMM parameters Λ_g^c for gesture recognition. Unlike the data-level and feature-level fusion scheme,

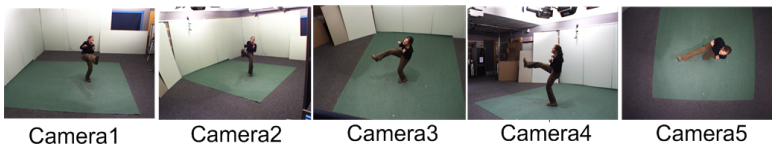


Fig. 6.2. Images of a sample pose obtained from the five cameras in IXMAS dataset. The brightness of the images are adjusted for better display.

where camera configurations between system training and testing are required to be consistent, decision-level fusion scheme allows for completely different camera configurations between training and testing. The only requirement for camera configuration in the decision-level fusion is that a testing camera must share a similar tilt angle with one of the training cameras so that pose features can be correctly extracted from the testing camera using the corresponding core tensor.

6.4 Experimental Results

6.4.1 The Benchmark IXMAS Gesture Recognition Dataset

In order to systematically compare various data fusion schemes and camera configurations, the same IXMAS dataset as applied in Chapter 4 is used as the benchmark dataset.

A total of five synchronized cameras have been used in the creation of the IXMAS dataset. The corresponding pan and tilt angles of these cameras can be easily extracted from their projection matrices as shown in Table 6.1. Sample images simultaneously obtained from these cameras are shown in Fig. 6.2. It can be seen that Camera 5 is an overhead camera and that the the optical axes of Cameras 1 to 4 are approximately parallel to the ground plane. In this experiment, Cameras 1 to 4 have been treated to be approximately sharing the same tilt angle so that they share the same core tensor for multilinear pose feature extraction. The camera calibration parameters are also given in the IXMAS dataset. In this experiment, such camera calibration information has been discarded during system testing to enforce the assumption of gesture recognition using multiple uncalibrated cameras.

TABLE 6.1
PAN AND TILT ANGLES OF THE IXMAS CAMERAS

Camera ID	1	2	3	4	5
Pan Angle	-84°	-49°	18°	41°	-16°
Tilt Angle	-21°	-11°	-39°	-3°	-80°

TABLE 6.2
GESTURE RECOGNITION RATES OBTAINED ON THE IXMAS DATASET USING VARIOUS DATA FUSION SCHEMES AND GESTURE RECOGNITION METHODS

Method	Fusion Level	Camera Combinations								
		1 3	1 4	2 4	3 5	1 3 5	1 2 3	1 2 3 5	1 2 3 4	1 2 3 4 5
Weinland 2006 [60]	Data (3D)	N.A.	N.A.	N.A.	N.A.	N.A.	N.A.	N.A.	N.A.	93.3%
Gu 2010 [124]	Data (3D)	N.A.	N.A.	N.A.	N.A.	N.A.	N.A.	N.A.	N.A.	94.4%
Multilinear+HMM	Data (3D)	N.A.	N.A.	N.A.	N.A.	N.A.	N.A.	N.A.	N.A.	***94.6%***
Weinland 2007 [79]	Data (2D)	N.A.	N.A.	81.3%	61.6%	70.2%	N.A.	75.9%	81.3%	N.A.
Multilinear+HMM	Data (2D)	82.8%	85.4%	*84.1%*	*82.8%*	*88.2%*	86.9%	*91.7%*	*89.5%*	92.0%
Srivastava 2009 [127]	Feature	75.6%				79.1%		81.4%		N.A
Multilinear+HMM	Feature	*86.0%*	*85.7%*	*84.1%*	*82.8%*	*89.8%*	*88.2%*	*91.7%*	*91.4%*	92.4%
Yan 2008 [126]	Decision (Sum Rule)	71%	N.A.	71%	N.A.	N.A.	60%	N.A.	78%	N.A
Multilinear+HMM	Decision (Sum Rule)	*80.3%*	80.9%	*79.0%*	76.8%	78.3%	*81.9%*	80.9%	*80.3%*	79.3%
Liu 2008 [130]	Decision (Vote Rule)	N.A.	N.A.	N.A.	N.A.	N.A.	N.A.	N.A.	82.8%	N.A
Reddy 2009 [131]	Decision (Vote Rule)	N.A.	N.A.	N.A.	N.A.	N.A.	N.A.	N.A.	72.6%	N.A
Multilinear+HMM	Decision (Max Rule)	80.3%	80.9%	79.0%	76.1%	78.0%	81.9%	80.6%	80.3%	79.0%
Multilinear+HMM	Decision (Min Rule)	79.6%	79.6%	81.2%	79.3%	82.5%	81.6%	84.4%	83.8%	84.4%
Yang 2008 [1]	Decision (Product Rule)	∼85%	∼85%	83.8%	78.8%	82.8%	∼83%	83.8%	85.9%	∼84%
Multilinear+HMM	Decision (Product Rule)	***87.6%***	***86.3%***	***85.7%***	***84.7%***	***92.4%***	***91.1%***	***93.6%***	***91.7%***	***93.0%***

6.4.2 Pose Tensor Formation Using 2D Observations

Unlike in Chapter 4, 2D silhouette images were applied in most experiments in this chapter. Therefore, there were some changes in formation of the training tensor for multilinear pose feature extraction. The 25 selected key poses, as described in the experimental section of Chapter 2, were still applied for pose tensor formation. To obtain 2D image observations, the 3D reconstruction of each key pose was rotated and then projected to the desired imaging planes. Then, these image observations were vectorized and assembled as the training tensor.

6.4.3 Comparison Across Data Fusion Schemes

Using the same cross-validation scheme as described in Chapter 4, gesture recognition rates for various camera combination scenarios and data fusion schemes have been obtained as shown in Table 6.2. Representative results from other methods on the same dataset have also been in-

cluded in the same table. In Table 6.2, each row corresponds to one gesture recognition method using one data fusion scheme and contains gesture recognition rates for various camera combination scenarios. A gesture recognition rate spanning over multiple columns is the best result among the corresponding camera combination scenarios. The second column of Table 6.2 indicates the corresponding data fusion schemes, including the data-level fusion schemes using 3D visual hull data and 2D silhouette image data, the feature-level fusion scheme, and the decision-level fusion schemes using various rules. For the sake of completeness, gesture recognition results using 3D visual hull data obtained from calibrated cameras are also included in the first three rows of Table 6.2. It can be seen from Table 6.2 that the gesture recognition results using the 3D visual hull data reconstructed from all five calibrated cameras obviously are better than the gesture recognition results using the uncalibrated cameras. This is not surprising because of the use of additional calibration information.

A close examination of Table 6.2 reveals two key observations. First of all, using the same camera combination and data fusion scheme, the benchmark gesture recognition framework using the view-invariant multilinear pose features and reduced-HMM has always produced the best results (*italic* in Table 6.2) among all the competing methods. This observation validates the selection of the benchmark gesture recognition framework in the reported comparison across data fusion schemes for multi-camera gesture recognition. Furthermore, when only the camera combination is fixed and the data fusion scheme can vary, the decision-level fusion scheme with the product rule always yields the best gesture recognition results (**bold** in Table 6.2) using the benchmark framework across all the tested camera combination scenarios. The importance of this finding is twofold. First, to my best knowledge, systematic comparison of different data fusion schemes for multi-camera gesture recognition using a common benchmark gesture recognition framework has not been done in existing research. The consistent superiority of the product rule-based decision-level fusion scheme over the data-level and the feature-level fusion schemes across a number of key camera combinations suggests that decision-level fusion with the product rule is the most effective way to integrate information from multiple uncalibrated cameras for gesture recognition.

Secondly, the fact that the product rule outperforms the other competing rules in classi-

fier combination for decision-level fusion in multi-camera gesture recognition essentially verifies existing data fusion principles in the sensor fusion and information integration literature. In [140], it has been shown that the product rule usually leads to superior classification performance when combining multiple classifiers for multi-class problems provided that the class posterior probability are well estimated. The results essentially verify this principle in the particular case of multi-camera gesture recognition using uncalibrated cameras, which has not been done before either. In general, the product rule takes into account the information obtained from all cameras in the recognition process, so clearly it is a better generalization of all views than the max rule and the min rule which only select the information from a representative camera. It is also interesting to compare the product rule and the sum rule and to investigate why the former outperforms the latter. It can be seen from Table 6.2 that the performances of the sum rule and the max rule are very similar, and the sum rule is only slightly better than the max rule in few camera combinations. This is because in this experiment the ratio between the largest and second largest likelihoods of the same gesture obtained from different views is generally very large (around 1000). In other words, the largest likelihood is generally dominant, making the sum of the likelihoods almost equivalent to the maximum of likelihoods. In contrast, the product rule applied to the likelihood is equivalent to adding up the logarithms of likelihoods from all the camera views, which are in general on the same scale. Consequently, the product rule is more effective in combining recognition results from different cameras than the other rules for the decision-level fusion.

To summarize, given the *fixed set* of uncalibrated cameras (i.e., camera combination), the decision-level fusion scheme using the product rule has been producing the best gesture recognition results across all the tested camera combinations using the benchmark gesture recognition framework.

6.4.4 Selection of Optimal Camera Combinations

When the number of cameras is fixed, how to identify the optimal camera combination is an important challenge for multi-camera gesture recognition. Intuitively, such optimal camera combination depends on the specific data fusion scheme and the image features used in multi-

camera gesture recognition. As shown in the previous section, for a particular camera combination, the benchmark gesture recognition framework using the decision-level fusion scheme and the product rule has been the optimal multi-camera gesture recognition method on the IXMAS dataset. Therefore, in this chapter focus has been given on identifying the optimal camera combination given the fixed number of cameras using the benchmark method with product rule-based decision-level fusion.

As pointed out in [128], a requirement for effectively combining results from different classifiers (i.e., the gesture recognition results from individual cameras in multi-camera gesture recognition) is that results from individual classifiers should not strongly correlate with each other in their misclassifications. Furthermore, it is desired that in the case of misclassification by one classifier (camera view) for a testing data, another classifier (cameras view) produces the correct classification. In this chapter, such these two cameras are referred to as being *complementary* for the given testing data. For multi-camera gesture recognition, it is important to measure the complementary strength across different cameras with respect to extracted image features. Such a complementary measure essentially reflect how much these two cameras disagree with other in the case of misclassification, and it is useful for camera selection and may provide valuable insight to the formation of optimal camera combinations. In this chapter, a simple while revealing inter-camera complementary strength measure have been proposed. Based on this measure the optimal camera combinations for the IXMAS cameras have been further identified in an incremental manner. As shown later in this section, these optimal camera combinations are actually the best performing ones among all the camera combinations with the same number of cameras on the IXMAS dataset.

One of the desired features for the inter-camera complementary measure is that the measure can be easily evaluated without explicitly conducting gesture/pose recognition, ideally directly from the image features extracted from different camera views. Such simple, image feature-based complementary measure can find important applications in camera selection. In this chapter, such a simple complementary measure have been proposed which can be directly evaluated using image pose features. In the approach to the inter-camera complementary measure, a number of key-pose image sets are first slected in which each set contains multi-view image

data synchronously collected using all the cameras corresponding to one of key poses identified for gesture recognition. For each key-pose image set, the multilinear pose features are then extracted from each camera view for all the cameras. Then, for each camera, pairwise Euclidean distances are obtained among its pose features, and these distances can be further split into two sets, a within-pose distance set containing pairwise pose feature distances between pose features from the same key pose, and a between-pose distance set containing distances between poses from different key poses. Let $D_W^c = \{d_W^{c,m}\}_{m=1}^M$ be the within-pose distance set and $D_B^c = \{d_B^{c,n}\}_{n=1}^N$ the between-pose distance set for Camera c, $c = 1,\ldots,C$. Assume that the distance indices m and n for the within-pose and between-pose distance sets from different cameras are aligned so that all distances sharing the same index in different views are between the pose features in the corresponding views from the same pair of key-pose image sets. Given two pose features extracted from the same camera, it is a binary classification problem to determine if they correspond to the same key pose or not. Equivalently, the same problem can be cast as a binary classification problem based on the pairwise pose feature distances so that a pairwise pose feature distance can be classified into the within-pose class or the between-pose class. In the approach to the inter-camera complementary measure, the complementary strength between two cameras when the above binary classification problem is solved has been measured. In other words, it is examined how much two cameras disagree with each other in the case of misclassification when solving the binary classification problem. In this experiment, such a simple complementary measure has been shown useful in selecting the optimal camera combinations in gesture recognition.

For each camera view, the above binary classification problem can be solved using a naive Bayesian classifier. In this experiment, the following "flipped" exponential distribution has been used to approximate the empirical distributions of the pose feature distances for both the within-pose and the between-pose classes.

$$p(d) = \lambda \exp^{-\lambda(a-d)}, d \leq a \tag{6.5}$$

where $a = \sqrt{2}$ is the maximum pose feature distance due to the fact that the multilinear pose features are scale-invariant unit vectors, i.e., for any multilinear pose feature $\mathbf{v}$, $||\mathbf{v}|| = 1$, and $\forall \alpha \neq 0$, $\alpha\mathbf{v}$ and $\mathbf{v}$ correspond to the same pose. In this experiment, the parameter λ is estimated

from training data by fitting the empirical probability density function (PDF) using the nonlinear least squares method. Although λ can also be found through maximum likelihood estimation, the independence of the training data is questionable. Therefore, the curve fitting approach has been taken to estimate λ. For a given camera view c, once the PDF parameter λ_W and λ_B have been estimated from training data both the within-pose and the between-pose classes, using uniform priors, the classification of a testing distance d_* using the naive Bayesian classifier becomes a likelihood-ratio test, which further boils down to the following threshold problem:

$$\begin{aligned} d_* \leq \Delta_c \quad &: \quad \text{within-pose distance class} \\ d_* > \Delta_c \quad &: \quad \text{between-pose distance class,} \end{aligned} \tag{6.6}$$

where $\Delta_c = \sqrt{2} - \frac{\ln \lambda_B - \ln \lambda_W}{\lambda_B - \lambda_W}$.

The complementary strength between two cameras can be measured based on the binary classification results using an aligned testing distance dataset. Consider two cameras α and β, $\alpha \neq \beta$ and $1 \leq \alpha \leq C$. Let Δ_α and Δ_β be the corresponding thresholds for the binary within/between-pose classification computed from training data. Let $D_T = \{D_T^\alpha, D_T^\beta\} = \{(d_k^\alpha, d_k^\beta)\}_{k=1}^K$ be the aligned testing distance set. Using Δ_α and Δ_β, the testing data in D_T^α and D_T^β can be independently classified into the within-pose distance class and the between-pose distance class. Let $\mathcal{E}$ be the set of indices at which a misclassification has occurred in at least one camera view, i.e.,

$$\mathcal{E} = \{k | d_k^\alpha \text{ is misclassified or } d_k^\beta \text{ is misclassified, } 1 \leq k \leq K\}. \tag{6.7}$$

The misclassification index set $\mathcal{E}$ can be easily obtained from the classification results in both views and the ground-truth data. Furthermore, let $\mathcal{C}$ be the set of indices at which there is a conflict between classification results from two cameras, i.e.,

$$\mathcal{C} = \left\{ k | (d_k^\alpha - \Delta_\alpha) \cdot (d_k^\beta - \Delta_\beta) < 0, \quad 1 \leq k \leq K \right\}. \tag{6.8}$$

It is easy to see that $\mathcal{C} \subseteq \mathcal{E}$, since when the classification results from cameras α and β are different, one of them must be a misclassification. To measure the complementary strength between cameras α and β, it is examined how much their classification results disagree with each other in the case of misclassification by defining the following *complementary coefficient*.

$$\rho_{\alpha,\beta} = \frac{||\sum_{k \in \mathcal{C}} (d_k^\alpha - \Delta_\alpha) \cdot (d_k^\beta - \Delta_\beta)||}{\sigma_\alpha \cdot \sigma_\beta} \tag{6.9}$$

where $\sigma_c^2 = \sum_{k \in \mathcal{E}} (d_k^c - \Delta_c)^2$, and it can be easily seen that $\rho_{\alpha,\beta} = \rho_{\beta,\alpha}$ and $0 \leq \rho_{\alpha,\beta} \leq 1$. A close connection can be drawn between the proposed complementary coefficient of two cameras and the correlation coefficient of two signals. In (6.9), the numerator can be treated as the "covariance" between the conflicting classification results from individual cameras centered at their corresponding classification thresholds. Likewise, σ_c^2 essentially captures the "classification" energy for camera c, $c = \alpha, \beta$, when there is a misclassification in either of these two cameras. Large $\rho_{\alpha,\beta}$ indicates strong discrepancy between the results from individual cameras when there is a misclassification, and combining classification results from cameras with large complementary coefficient may lead to greater improvement than combining results from cameras with small complementary coefficient. Furthermore, $\rho_{\alpha,\beta}$ can be extended to the case when α is a camera set containing more than one cameras. Such situation may arise when a camera needs to be selected from a number of candidates to add to an existing camera set. In this case, the complementary strength between a camera candidate and the existing camera set is required. To this end, the pairwise distance values between pose features from individual cameras in the camera set α are first combined by taking their corresponding root mean square across cameras. Thus, the camera set α can be treated as a virtual camera with its own pairwise distance measures, and the complementary coefficient between a camera set α and a single camera β can be computed as if α were also a single camera. In this experiment, when α is a camera set, the empirical distributions of the within-pose distance training data D_W^α and the between-pose distance training data D_B^α both approximately follow a normal distribution with similar variances. Hence, the corresponding classification threshold Δ_α in (6.6) is simply taken as $\Delta_\alpha = \frac{\mu_W^\alpha + \mu_B^\alpha}{2}$, and μ_W^α and μ_B^α are the sample means of the corresponding training sets.

In this research, the proposed inter-camera complementary coefficient has been utilized to identify the optimal camera combinations for the IXMAS dataset in an incremental manner. The optimal pairwise camera combination is first identified by finding the camera pair possessing the largest complementary coefficient. Based on the optimal pairwise camera combination, the problem of finding the optimal 3-camera combination is casted as selecting the best third camera from the remaining cameras to add to the optimal 2-camera combination. Likewise, the optimal 4-camera combination is found in a similar way based on the optimal 3-camera combination. To identify the best pairwise camera combination, it is needed to evaluate the

TABLE 6.3
COMPLEMENTARY COEFFICIENTS ACROSS THE IXMAS CAMERAS OBTAINED USING POSE FEATURES

α	β	$\rho_{\alpha,\beta}$
1	2	0.4804
1	**3**	**0.5083**
1	4	0.4824
1	5	0.4946
2	3	0.5078
2	4	0.4913
2	5	0.4949
3	4	0.4800
3	5	0.5069
4	5	0.4970
(1,3)	2	0.4724
(1,3)	4	0.4572
(1,3)	**5**	**0.4863**
(1,3,5)	**2**	**0.4776**
(1,3,5)	4	0.4685

complementary coefficients for all the camera pairs. To this end, a total of 520 key-pose image sets have been manually selected. For each key-pose image set, the multilinear pose features and pairwise distances among these pose features are obtained from each camera view for all the five IXMAS cameras. These pairwise distances are further split into the within-pose and between-pose distance sets. In this experiment, to compute $\rho_{\alpha,\beta}$, half of the distance data in each camera have been randomly selected for classifier training to learn the classification threshold Δ_α and Δ_β, and the other half is used for testing the classifier. The corresponding complementary coefficient $\rho_{\alpha,\beta}$ can be evaluated using (6.9). Note that the distance data in the selected training and testing datasets remain aligned across cameras α and β. In this experiment, for each camera pair, such procedure has been repeated 100 times and the average is taken as the final complementary coefficient between the two cameras. Table 6.3 shows such average complementary coefficients for all the pairwise camera combinations in the IXMAS dataset. It can be seen from Table 6.3 that camera combination (1,3) possesses the largest complementary coefficient among all the camera pairs. Hence, camera combination (1,3) is selected as the optimal camera pair for 2-camera gesture recognition. Furthermore, given the optimal pairwise camera combination $(1,3)$, the best third camera to form the optimal 3-camera combination can be found

according to the complementary coefficients $\rho_{\alpha,\beta}$ between $\alpha = (1,3)$ and $\beta \in \{2,4,5\}$. The average complementary coefficients over 100 trials are also shown in Table 6.3. It can be seen that Camera 5, the overhead camera, should be selected to add to (1,3) since this combination has the largest complementary coefficient. Therefore, camera combination (1,3,5) is identified to be the optimal combination for 3-camera gesture recognition. The same experiment has also been done to select the best fourth camera to add to this optimal 3-camera combination. In this case, $\rho_{\alpha,\beta}$ have been obtained where $\alpha = (1,3,5)$ and $\beta \in \{2,4\}$ as shown in Table 6.3. It can be seen that $\rho_{\alpha,\beta}$ is larger when $\beta = 2$, indicating Camera 2 should be added to (1,3,5) to form the best 4-camera combination (1,2,3,5). The optimal camera combinations and their complementary coefficients are highlighted in **bold** in Table 6.3.

To evaluate the accuracy of the identified optimal camera combinations with the fixed number of cameras using the proposed complementary coefficient, the gesture recognition results for all the multi-camera combination scenarios in the IXMAS dataset have been obtained using the benchmark framework with product rule-based decision-level fusion through the same cross-validated training and testing scheme described in Section 4.4.1. The corresponding gesture recognition results are presented in Table 6.4 and Fig. 6.3. In Table 6.4, the optimal camera combinations and their corresponding gesture recognition results using the benchmark method are in **bold**. By comparing results in Tables 6.3 and 6.4, it is clear that all the optimal camera combinations identified according to the complementary coefficients in Table 6.3 indeed correspond to the best performing camera combinations in Table 6.4. This observed consistency between the optimal C-camera combinations ($C = 2,3,4$) in Table 6.4 and their complimentary coefficients in Table 6.3 validates the proposed approach to the complementary strength measure across cameras. Such an inter-camera complementary strength measure can find important applications in camera selection for multi-camera vision problems.

For comparison purposes, gesture recognition results for these multi-camera combination scenarios reported in [1] have also been included in Table 6.4 and illustrated in Fig. 6.3. Three observations can be made by comparing results obtained using proposed method and those in [1]. First of all, it is clear that the benchmark method with product rule-based decision-level fusion is consistently superior to the method in [1] for all the multi-camera combination scenar-

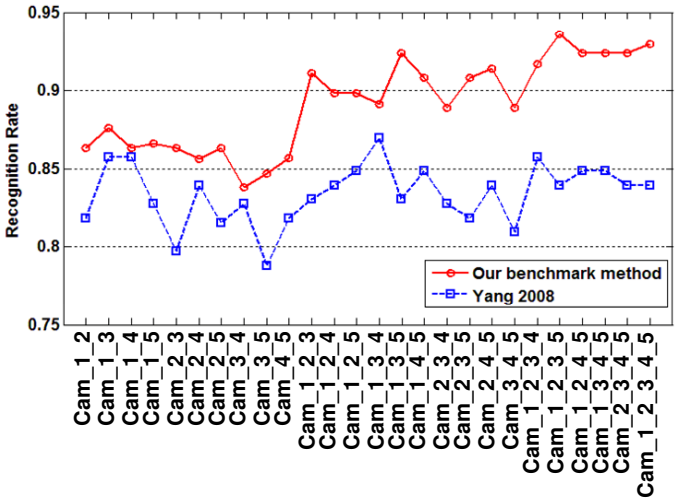


Fig. 6.3. Performance comparison between the benchmark method and Yang 2008 [1].

ios. Furthermore, it can be seen that using the benchmark method, adding additional cameras to an existing camera combination always increases the recognition rate, except for only two cases when Camera 4 is added to camera combinations (1,3,5) and (1,2,3,5). In contrast, as summarized in Table 6.5, there are far more cases (about 20) in the results from [1] in which additional cameras lead to decreased gesture recognition rates. This observation indicates that the new information provided by additional cameras can be well adopted by the benchmark method to improve the accuracy of multi-camera gesture recognition. Finally, when the number of cameras C is fixed and $C \geq 3$, the corresponding optimal camera combinations have always included the overhead Camera 5. This is striking because in traditional multi-camera gesture recognition, the overhead camera is often considered nonessential when other side-view cameras are available. For example, in [127], the overhead Camera 5 has been excluded from being used in multi-camera gesture recognition due to lack of discriminative features from this camera. The reason behind this striking fact is that Camera 5 is the most complementary with the optimal pairwise camera combination $(1,3)$ as shown by Table 6.3.

TABLE 6.4
GESTURE RECOGNITION RATES FOR DIFFERENT CAMERA COMBINATIONS USING DECISION-LEVEL FUSION WITH THE PRODUCT RULE

Camera Combination	**Recognition Rate**	
	Benchmark	**Yang 2008** [1]
(1,2)	86.3%	81.8%
(1, 3)	**87.6%**	85.8%
(1,4)	86.3%	85.8%
(2,3)	86.3%	82.7%
(2,4)	85.6%	79.7%
(3,4)	83.8%	83.9%
(1,5)	86.6%	81.5%
(2,5)	86.3%	82.7%
(3,5)	84.7%	78.8%
(4,5)	85.7%	81.8%
(1,2,3)	91.1%	83.0%
(1,2,4)	89.8%	83.9%
(1,2,5)	89.8%	84.9%
(1,3,4)	89.1%	87.0%
(1, 3, 5)	**92.4%**	83.0%
(1,4,5)	90.5%	84.9%
(2,3,4)	88.9%	82.7%
(2,3,5)	90.8%	81.8%
(2,4,5)	91.4%	83.9%
(3,4,5)	88.9%	80.9%
(1,2,3,4)	91.7%	85.8%
(1, 2, 3, 5)	**93.6%**	83.9%
(1,2,4,5)	92.4%	84.9%
(1,3,4,5)	92.4%	84.9%
(2,3,4,5)	92.4%	83.9%
(1, 2, 3, 4, 5)	**93.0%**	83.9%

TABLE 6.5
CRITICAL CAMERA COMBINATION SCENARIOS IN [1]

Original Cameras	(1,3)	(1,4)	(2,3)	(3,4)	(2,5)	(4,5)	(1,2,5)	(1,3,4)	(1,4,5)	(1,2,3,4)	(1,2,4,5)	(1,3,4,5)
Added Camera(s)	2, 5, (2,5)	2, 5, (2,5)	5	2, 5	3	3	3, (3,4)	2, 5, (2,5)	(2,3)	5	3	2

Chapter 7

POSE ESTIMATION USING INVARIANT FEATURES AND RELEVANCE VECTOR MACHINE

Human pose estimation, i.e. estimating the joint configurations of a person, is an important problem in human movement sensing. In some HCI systems, such as the interactive rehabilitation system in [141], human joint angles or positions have been used to evaluate the training performance of subjects. In this chapter, the application of invariant pose features in pose estimation is presented.

7.1 Overview

Estimating poses of a highly articulated and self-occluding non-rigid human body from images is a challenging problem. Existing pose estimation methods can be generally divided into generative methods and discriminative methods.

Generative pose estimation methods [142–147] first define a likelihood function of an observation given a pose. The likelihood function is usually based on matching the query observation with exemplars in the training set [147] or the projected observation from human models [144]. The Bayesian methods are often used to obtain the posterior distribution of a pose based on the likelihood function. Using generative methods, the dimensionality of the search space is usually very high, making these methods computationally expensive.

Discriminative pose estimation methods [148–151] establish a mapping from observation space to pose space. One of the discriminative approaches is to detect body parts from observations [148, 149]. The mapping can also be established using regression methods, as introduced in [150, 151]. One challenge of discriminative methods is that the mapping from image observations to poses is usually one-to-many. This challenge can be tackled by either obtaining 3D volumetric reconstruction from images [148], or dividing observation space into subspaces where one-to-one mappings exist [150].

In this chapter, a discriminative pose estimation method is introduced. This method applies

the relevance vector machine to establish a mapping from the invariant pose features to the joint locations. The application of the invariant pose features simplifies the process of regression. Using our simple method, the pose estimation results on a public dataset is comparable to the existing results.

7.2 An Introduction to Relevance Vector Machine

The Relevance Vector Machine (RVM) is a sparse Bayesian regression method proposed by Tipping [152]. The idea of RVM starts from a simple linear regression problem, in which the relationship of the input $\mathbf{x}$ and output y is modeled as

$$y(\mathbf{x}) = \mathbf{w}^T \phi(\mathbf{x}), \tag{7.1}$$

where ϕ is a set of feature functions that map input vector $\mathbf{x}$ to a feature vector in order to model nonlinearity, and $\mathbf{w}$ is the weight vector. To estimate $\mathbf{w}$, a set of training inputs $\mathbf{x}_i$ and target values t_i, $i = 1,\ldots,N$, are used. The target t_i is modeled as

$$t_i = y(\mathbf{x}_i) + \varepsilon_i, \tag{7.2}$$

in which $\varepsilon_i \sim \mathcal{N}(0,\sigma^2)$ is a zero-mean Gaussian noise with variance σ^2. The likelihood of the target vector $\mathbf{t} = [t_1,\ldots,t_N]^T$ can be expressed as

$$p(\mathbf{t}|\mathbf{w},\sigma^2) = (2\pi\sigma^2)^{-N/2} \exp\left\{-\frac{1}{2\sigma^2}\|\mathbf{t}-\Phi\mathbf{w}\|^2\right\}, \tag{7.3}$$

in which Φ is the design matrix and $\Phi_{ij} = \phi_j(\mathbf{x}_i)$.

The maximum likelihood estimation of $\mathbf{w}$ and σ^2 based on (7.3) often leads to overfitting. To regularize $\mathbf{w}$, a prior distribution can be added as follows.

$$p(\mathbf{w}|\alpha) = \prod_{i=1}^{M} \mathcal{N}(w_i|0,\alpha_i^{-1}), \tag{7.4}$$

in which M is the number of feature functions and α is a set of hyperparameters describing the variances of the weights.

Ideally, for Bayesian inference, parameters $\mathbf{w}$, α and σ^2 should be estimated by maximizing the posterior probability $p(\mathbf{w},\alpha,\sigma^2|\mathbf{t})$. However, this cannot be done analytically. Alternatively, this problem can be solved in an iterative manner. The posterior distribution can be

decomposed as

$$p(\mathbf{w}, \alpha, \sigma^2|\mathbf{t}) = p(\mathbf{w}|\alpha, \sigma^2, \mathbf{t})p(\alpha, \sigma^2|\mathbf{t}). \tag{7.5}$$

Using Bayesian rules, the first element on the right side of (7.5) can be obtained and expressed as

$$p(\mathbf{w}|\alpha, \sigma^2, \mathbf{t}) = \mathcal{N}(\mathbf{m}, \Sigma), \tag{7.6}$$

in which the mean and the covariance matrix are

$$\mathbf{m} = \sigma^{-2}\Sigma\Phi^T\mathbf{t}, \tag{7.7}$$

$$\Sigma = (\mathbf{A} + \sigma^{-2}\Phi^T\Phi)^{-1}, \tag{7.8}$$

where $\mathbf{A} = \text{diag}(\alpha_1, \ldots, \alpha_M)$. To evaluate $\mathbf{m}$ and Σ we need to find α and σ^2 that maximize the the second element on the right side of (7.5), which can be decomposed as

$$p(\alpha, \sigma^2|\mathbf{t}) \propto p(\mathbf{t}|\alpha, \sigma^2)p(\alpha)p(\sigma^2). \tag{7.9}$$

By assuming uniformly distributed α and σ^2, we can ignore $p(\alpha)$ and $p(\sigma^2)$. Then maximizing $p(\alpha, \sigma^2|\mathbf{t})$ becomes maximizing

$$p(\mathbf{t}|\alpha, \sigma^2) = \int p(\mathbf{t}|\mathbf{w}, \sigma^2)p(\mathbf{w}|\alpha)d\mathbf{w}. \tag{7.10}$$

By defining $\gamma_i = 1 - \alpha_i\Sigma_{ii}$, the maximizers of probability $p(\mathbf{t}|\alpha, \sigma^2)$ can be expressed as

$$\alpha_i = \frac{\gamma_i}{m_i^2}, \tag{7.11}$$

$$\sigma^2 = \frac{\|\mathbf{t} - \Phi\mathbf{m}\|^2}{N - \sum_i \gamma_i}. \tag{7.12}$$

Therefore, given initial values of α and σ^2, we can estimate the values of these two parameters by using (7.7), (7.8) and (7.11), (7.12) iteratively until convergence. After this, we can compute $\mathbf{m}$ using (7.7), which is just the optimal $\mathbf{w}$ in model (7.1).

In practice, during the estimation process, many of the α_i approach infinity. An infinity value of α_i causes the corresponding w_i to be zero, which means corresponding feature function ϕ_i can be discarded. Therefore the resulting model is usually very sparse.

7.3 Pose Estimation Using RVM

In this chapter, a human pose is represented as the 3D positions of M joints, denoted to be $\mathbf{m}_i \in \mathbb{R}^3$, $i = 1,\ldots,M$. When calculating the coordinates of these joint positions, the global translation of the pelvis and its rotation about the axis perpendicular to ground plane (body orientation) is removed. Therefore, this pose representation is orientation-invariant. The coordinates are also normalized by the height of the subject in order to minimize the differences in joint positions caused by different body shapes.

To estimate poses using invariant pose features, a mapping need to be established from the invariant pose feature vector to every dimension of each $\mathbf{m}_i$. The invariant pose features are extracted from 3D volumetric reconstructions in order to avoid ambiguity. In this presented framework, each mapping is modeled using a linear model (7.1), and the weights of the model are learned using the method presented in the previous section. For each mapping, we apply the same set of feature functions. If given N training inputs $\mathbf{x}_1,\ldots,\mathbf{x}_N$, $N+1$ feature functions are applied. The first N functions are defined using RBF kernel as follows.

$$\phi_i(\mathbf{x}) = k(\mathbf{x},\mathbf{x}_i) = \exp(-\frac{1}{\gamma}\|\mathbf{x}-\mathbf{x}_i\|^2), \quad i = 1,\ldots,N, \tag{7.13}$$

in which γ the kernel parameter. The last feature function is defined to be $\phi_{N+1}(\mathbf{x}) = 1$ in order to model the bias term.

7.4 Experimental Results and Analysis

7.4.1 The HumanEva-I Dataset

The pose estimation framework presented in this chapter was tested on the HumanEva-I dataset [153] created by Brown University. It contains synchronized image and motion capture data obtained from 4 subjects. Images of the movements were captured by 3 color cameras and 4 grayscale cameras. Each subjects has performed 6 types of movements, namely walking, jogging, gesturing, throwing and catching a ball, boxing and combo. Following the most common usage of this dataset, the first 3 subjects and the first 5 movements were used for training and testing in the experiments in this section.

In this dataset, two trials have been recorded for each subject performing each movement, and the motion capture data was withheld for the second trial for testing purposes. Therefore, for convenience of the experiment, only the first trial of each movement performed by each subject was applied. Part of the trial ("training" frames defined in [153]) was used in training, and the rest ("validation" frames defined in [153]) was applied in testing.

7.4.2 Pose Feature Extraction

The pose features applied in pose estimation are extracted from 3D volumetric reconstructions. In order to reconstruct the 3D shapes of the poses, the silhouette images were first extracted using background subtraction. In the HumanEva-I dataset, the quality of the silhouettes extracted from grayscale cameras are relatively poor and false foreground regions often exist. Therefore, the 3D reconstruction has been conducted in the following method. First, the capture volume is discretized into voxels and these voxels are projected onto the image planes of all the cameras. Then the voxels that are projected inside the foreground areas of all color cameras are taken as valid points constructing the 3D pose (marked as 1). The rest of the voxels are carved out (marked as 0).

As described in Section 2.3.3, a set of key poses are needed to construct the training set for pose feature extraction. In this experiment, the similar key pose selection method used in Section 2.3.3 was also applied. The only difference is that the motion energy and poses distances were defined base on the differences in the joint positions. After key pose were selected, the corresponding 3D reconstructions were applied to construct the training set for pose feature extraction. In this experiment, 50 key poses were selected from pose performed by the second subject in the HumanEva-I dataset.

For pose estimation, it is desired that the pose features are as continuous as possible, i.e. pose features extracted from similar poses should be similar. For this purpose, the normalization procedure of visual hulls as described in Section 2.3.2 is modified. In the normalization procedure for pose estimation, the bottom of the visual hulls are always on the floor, and a constant scaling factor according to the height of the subject is applied to all dimensions. Furthermore, the proper type of pose features also needs to be selected. If multilinear analysis is applied

for feature extraction, the features of key poses are forced to be orthonormal regardless of actual difference between poses. MGP features, on the other hand, do not have this constraint. Therefore, in the pose estimation experiment, the MGP pose features were selected to improve continuity of the pose features. Both the view-invariant MGP features (extracted using 2-factor model) and the view and shape-invariant MGP features (extracted using 3-factor model) were tested for pose estimation.

7.4.3 Error Measure

In the HumanEva-I dataset, a pose is represented by the positions of 15 joints. Since the global movement of the pelvis is removed, $M = 14$ joint remain. In this experiment, the average Euclidean distance (AED) defined in [153] is applied for measuring the distance between the estimated and ground truth poses. Denoting the estimated pose to be $\hat{\mathbf{x}}$ and the ground truth pose $\mathbf{x}$, the AED is defined as

$$D(\hat{\mathbf{x}}, \mathbf{x}) = \frac{1}{14} \sum_{i=1}^{14} \|\mathbf{m}_i(\hat{\mathbf{x}}) - \mathbf{m}_i(\mathbf{x})\| \cdot H_s, \tag{7.14}$$

where $\mathbf{m}_i$ is the normalized coordinate of joint positions, and H_s is the height of the subject (the normalization term). When evaluating pose estimation for a stream of poses, the average Euclidean distance is taken across all frames.

7.4.4 Pose Estimation Results

Using the training frames of all the five movements performed by all the three subjects, the RVM regressors were trained. Then, these regressors were applied to estimate poses of the testing frames. The AED errors of the estimations using view-invariant features and view and shape-invariant features are listed in Table 7.1 and Table 7.2, respectively. In these tables, N.A. means no valid motion capture data exist for the corresponding movement and subject. The results obtained using the proposed method are also compared with results reported in [154]. The comparison is shown in Table 7.3. From Table 7.3 it can first be observed that the application of view and shape-invariant features has improved the pose estimation accuracy. It can also be observed that results obtained using the proposed method is comparable to the

TABLE 7.1
POSE ESTIMATION ERRORS (IN MINIMETER) OF HUMANEVA-I DATASET USING VIEW-INVARIANT FEATURES

Subject	Walking	Jogging	Gesturing	Boxing	Throw and Catch
S1	61	72	27	74	N.A.
S2	45	51	81	95	82
S3	81	50	87	102	N.A.
Mean	**62**	**56**	**60**	**94**	**82**

TABLE 7.2
POSE ESTIMATION ERRORS (IN MINIMETER) OF HUMANEVA-I DATASET USING VIEW AND SHAPE-INVARIANT FEATURES

Subject	Walking	Jogging	Gesturing	Boxing	Throw and Catch
S1	53	60	25	59	N.A.
S2	45	52	76	102	84
S3	83	48	77	98	N.A.
Mean	**59**	**52**	**55**	**89**	**84**

TABLE 7.3
COMPARISON OF POSE ESTIMATION RESULTS

Method	Walking	Jogging	Gesturing	Boxing	Throw and Catch	Mean
The proposed method, 2-factor feature	62	56	60	94	82	71
The proposed method, 3-factor feature	59	52	55	89	84	68
Bo's Method [154]	53	49	43	64	76	57

existing results reported in [154]. The minor inferiority of the proposed method could be mainly cased by the quality of the 3D volumetric reconstructions. In future work, efforts will be made to improve the quality of the 3D reconstructions.

Chapter 8

CONCLUSIONS AND FUTURE WORK

In this dissertation, I present my research on invariant video-based pose feature extraction and its application in movement recognition and pose estimation. Two approaches based on the multilinear analysis and MGP have been developed to extract invariant pose features. Features with decent view-invariance property have been successfully extracted using both approaches. By including the body shape as additional factor in the MGP model, the body shape invariance of pose features improves noticeably.

Using the invariant pose features and SVM, promising pose recognition results have been obtained. Also, using such pose features and HMM, state-of-the-art gesture classification results have been obtained on the IXMAS dataset. Based on the approach to pre-segmented gesture classification, the challenging problem of online gesture spotting has been solved using an HMM-network with specific non-gesture models. Including specific non-gesture models in the HMM-network improves gesture spotting by reducing false alarm rates without significantly sacrificing the recognition rates. The invariant pose features also simplifies the regression process in pose estimation. Using proposed pose features and RVM, promising pose estimation results have been obtained on the HumanEva-I dataset.

Furthermore, different strategies of multi-camera fusion for gesture recognition have been explored and decision-level camera fusion using the product rule has been found to be the optimal fusion scheme for a fixed set of uncalibrated cameras. Also, a cross-camera complementary measure has been developed. Using such complementary measure, the optimal camera combinations for gesture recognition on the IXMAS dataset have been successfully identified.

In the future, the following improvements can be made to this research. First of all, to improve computational efficiency, parallel computing can be exploited. Parallel computing can be applied in silhouette extraction, visual-hull reconstruction as well as the feature-level and decision-level fusion schemes. Furthermore, the RBF kernels are currently applied to all the modes in MGP-based pose feature extraction methods. In the future, other types of kernels can be explored. Different types of kernels can be applied to different modes to better represent the

corresponding modes. Finally, in the current pose estimation framework, temporal information is not applied. In the future, approaches making use of the temporal information can be explored to improve the pose estimation.

[1] Y. Yang, A. Hao, and Q. Zhao, "View-invariant action recognition using interest points," in *Proceeding of the 1st ACM international conference on Multimedia information retrieval*, 2008, pp. 305–312.

[2] S. Rajko and G. Qian, "HMM parameter reduction for practical gesture recognition," in *Proceedings of IEEE International Conference on Face and Gesture Recognition*, 2008, pp. 1–6.

[3] J. Ren, T. Vlachos, and V. Argyriou, "Immersive and perceptual human-computer interaction using computer vision techniques," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2010, pp. 66 –72.

[4] J. Allard, J. Franco, C. Menier, E. Boyer, and B. Raffin, "The GrImage platform: A mixed reality environment for interactions," in *Proceedings of IEEE International Conference on Computer Vision Systems*, 2006, pp. 46–52.

[5] I. Barandiaran, C. Paloc, and M. Graa, "Real-time optical markerless tracking for augmented reality applications," *Journal of Real-Time Image Processing*, vol. 5, pp. 129–138, 2010.

[6] D. Gelb, A. Subramanian, and K.-H. Tan, "Augmented reality for immersive remote collaboration," in *Proceedings of 2011 IEEE Workshop on Person-Oriented Vision*, 2011, pp. 1 –6.

[7] H.-Y. Lin and T.-W. Chen, "Augmented reality with human body interaction based on monocular 3D pose estimation," in *Advanced Concepts for Intelligent Vision Systems*, 2010, vol. 6474, pp. 321–331.

[8] J. Billingsley and R. Bradbeer, *Mechatronics and Machine Vision in Practice*. Springer, 2008.

[9] O. C. Jenkins, G. González, and M. M. Loper, "Tracking human motion and actions for interactive robots," in *Proceedings of the ACM/IEEE International Conference on Human-Robot Interaction*, 2007, pp. 365–372.

[10] S.-W. Lee, "Automatic gesture recognition for intelligent human-robot interaction," in *Proceedings of the IEEE International Conference on Automatic Face and Gesture Recognition*, 2006, pp. 645–650.

[11] S. Wachsmuth, S. Wrede, and M. Hanheide, "Coordinating interactive vision behaviors for cognitive assistance," *Computer Vision and Image Understanding*, vol. 108, no. 1-2, pp. 135 – 149, 2007.

[12] B. Fang, F. Oliveira, and F. Quek, "Using vision based tracking to support real-time graphical instruction for students who have visual impairments," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2010, pp. 9 –14.

[13] S. C. Ong and S. Ranganath, "Automatic sign language analysis: a survey and the future beyond lexical meaning," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 27, no. 6, pp. 873–891, 2005.

[14] H. S. Park, D. J. Jung, and H. J. Kim, "Vision-based game interface using human gesture," in *Advances in Image and Video Technology*. Berlin/Heidelberg: Springer, 2006, pp. 662–671.

[15] A. Camurri, S. Hashimoto, M. Ricchetti, A. Ricci, K. Suzuki, R. Trocca, and G. Volpe, "Eyesweb: Toward gesture and affect recognition in interactive dance and music systems," *Computer Music Journal*, vol. 24, no. 1, pp. 57–69, 2000.

[16] J. James, T. Ingalls, G. Qian, L. Olsen, D. Whiteley, S. Wong, and T. Rikakis., "Movement-based interactive dance performance," in *Proceedings of the ACM International Conference on Multimedia*, 2006, pp. 470–480.

[17] J. Kjolberg, "Designing full body movement interaction using modern dance as a starting point," in *Proceedings of the Conference on Designing Interactive Systems*, 2004, pp. 353 – 356.

[18] G. Qian, F. Guo, T. Ingalls, L. Olson, J. James, and T. Rikakis, "A gesture-driven multimodal interactive dance system," in *Proceedings of the IEEE International Conference on Multimedia and Expo*, 2004, pp. 1579–1582.

[19] H. Zhou, H. Hu, and N. Harris, "Wearable inertial sensors for arm motion tracking in home-based rehabilitation," in *Proceedings of the 9th International Conference on Intelligent Autonomous Systems*, 2006, pp. 930–937.

[20] S. Rajko, G. Qian, T. Ingalls, and J. James, "Real-time gesture recognition with minimal training requirements and on-line learning," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2007, pp. 1–8.

[21] M. A. O. Vasilescu, "Human motion signatures: Analysis, synthesis, recognition," in *Proceedings of International Conference on Pattern Recognition*, 2002, pp. 456–460.

[22] J. Davis and H. Gao, "An expressive three-mode principal components model of human action style," *Image and Vision Computing*, vol. 21, no. 11, pp. 1001–1016, 2003.

[23] ——, "Recognizing human action efforts: an adaptive three-mode pca framework," in *Proceedings of International Conference on Computer Vision*, 2003, pp. 1–7.

[24] A. Laurentini, "The visual hull concept for silhouette-based image understanding," *PAMI*, vol. 16, no. 2, pp. 150–162, 1994.

[25] M. Hasanuzzaman, T. Zhang, V. Ampornaramveth, P. Kiatisevi, Y. Shirai, and H. Ueno, "Gesture based human-robot interaction using a frame based software platform," in *Proceedings of IEEE International Conference on Systems, Man and Cybernetics*, vol. 3, 2004, pp. 2883 – 2888.

[26] L. Wang and D. Suter, "Learning and matching of dynamic shape manifolds for human action recognition," *IEEE Transactions on Image Processing*, vol. 16, no. 6, pp. 1646 –1661, 2007.

[27] T. Ding, "A robust identification approach to gait recognition," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, 2008, pp. 1 –8.

[28] C.-S. Lee and A. Elgammal, "Coupled visual and kinematic manifold models for tracking," *International Journal of Computer Vision*, vol. 87, pp. 118–139, 2010.

[29] H. Cai, K. Mikolajczyk, and J. Matas, "Learning linear discriminant projections for dimensionality reduction of image descriptors," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 33, no. 2, 2011.

[30] M. Kim and V. Pavlovic, "Central subspace dimensionality reduction using covariance operators," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 33, no. 4, pp. 657 –670, 2011.

[31] M.-K. Hu, "Visual pattern recognition by moment invariants," *IRE Transactions on Information Theory*, vol. 8, no. 2, pp. 179 –187, 1962.

[32] X. Sun, M. Chen, and A. Hauptmann, "Action recognition via local descriptors and holistic features," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2009, pp. 58 –65.

[33] F. Lv and R. Nevatia, "Single view human action recognition using key pose matching and viterbi path searching," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2007, pp. 1–8.

[34] C. Chu and I. Cohen, "Pose and gesture recognition using 3D body shapes decomposition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2005, pp. 69–78.

[35] A. Yilmaz, "Recognizing human actions in videos acquired by uncalibrated moving cameras," in *Proceedings of IEEE International Conference on Computer Vision*, 2005, pp. 150–157.

[36] Y. Shen and H. Foroosh, "View-invariant action recognition from point triplets," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 31, no. 10, pp. 1898–1905, 2009.

[37] V. Parameswaran and R. Chellappa, "View invariance for human action recognition," *International Journal of Computer Vision*, vol. 66, no. 1, pp. 83–101, 2006.

[38] C. Harris and M. Stephens, "A combined corner and edge detector," in *Proceedings of the 4th Alvey Vision Conference*, 1988, pp. 147–151.

[39] D. Lowe, "Object recognition from local scale-invariant features," in *Proceedings of IEEE International Conference on Computer Vision*, vol. 2, 1999, pp. 1150 –1157.

[40] H. Bay, T. Tuytelaars, and L. V. Gool, "Surf: Speeded up robust features," in *Proceedings of European Conference on Computer Vision*, 2006, pp. 404–417.

[41] G. Takacs, V. Chandrasekhar, S. Tsai, D. Chen, R. Grzeszczuk, and B. Girod, "Unified real-time tracking and recognition with rotation-invariant fast features," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, 2010, pp. 934 –941.

[42] M. Brown, G. Hua, and S. Winder, "Discriminative learning of local image descriptors," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 33, no. 1, pp. 43 –57, 2011.

[43] A. Gilbert, J. Illingworth, and R. Bowden, "Action recognition using mined hierarchical compound features," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 33, no. 5, pp. 883 –897, 2011.

[44] I. Laptev, M. Marszalek, C. Schmid, and B. Rozenfeld, "Learning realistic human actions from movies," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, 2008, pp. 1–8.

[45] G. Willems, T. Tuytelaars, and L. Gool, "An efficient dense and scale-invariant spatio-temporal interest point detector," in *Proceedings of European Conference on Computer Vision: Part II*. Berlin, Heidelberg: Springer-Verlag, 2008, pp. 650–663.

[46] P. Dollár, V. Rabaud, G. Cottrell, and S. Belongie, "Behavior recognition via sparse spatio-temporal features," in *Proceedings of IEEE International Workshop on Visual Surveillance and Performance Evaluation of Tracking and Surveillance*, 2005, pp. 65–72.

[47] H. J. Seo and P. Milanfar, "Action recognition from one example," *IEEE Transactions onPattern Analysis and Machine Intelligence*, vol. 33, no. 5, pp. 867 –882, 2011.

[48] A. Oikonomopoulos, I. Patras, and M. Pantic, "Spatiotemporal localization and categorization of human actions in unsegmented image sequences," *IEEE Transactions on Image Processing*, vol. 20, no. 4, pp. 1126–1140, 2011.

[49] C.-C. Li and Y.-Y. Chen, "Human posture recognition by simple rules," in *Proceedings of the IEEE International Conference on Systems, Man and Cybernetics*, 2006, pp. 3237 – 3240.

[50] P. Guo, Z. Miao, and Y. Yuan, "Posture and activity recognition using projection histogram and pca methods," in *Proceedings of International Congress on Image and Signal Processing*, 2008, pp. 397 – 401.

[51] S. Eickeler, A. Kosmala, and G. Rigoll, "Hidden markov model based continuous online gesture recognition," in *Proceedings of the International Conference on Pattern Recognition*, 1998, pp. 1206–1208.

[52] J. Alon, V. Athitsos, Q. Yuan, and S. Sclaroff, "A unified framework for gesture recognition and spatiotemporal gesture segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 31, no. 9, pp. 1685–1699, 2009.

[53] T. Starner, J. Weaver, and A. Pentland, "Real-time american sign language recognition using desk and wearable computer based video," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 20, no. 12, pp. 1371–1375, 1998.

[54] Y. Zhu and G. Xu, "A real-time approach to the spotting, representation, and recognition of hand gestures for humancomputer interaction," *Computer Vision and Image Understanding*, vol. 85, pp. 189–208, 2002.

[55] H.-K. Lee and J. Kim, "An HMM-based threshold model approach for gesture recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 21, no. 10, pp. 961–973, 1999.

[56] H.-D. Yang, A.-Y. Park, and S.-W. Lee, "Gesture spotting and recognition for human-robot interaction," *IEEE Transactions on Robotics*, vol. 23, no. 2, pp. 256–270, 2007.

[57] H.-D. Yang, S. Sclaroff, and S.-W. Lee, "Sign language spotting with a threshold model based on conditional random fields," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 31, no. 7, pp. 1264–1277, 2009.

[58] D. Birchfield, T. Ciufo, and G. Minyard, "Smallab: a mediated platform for education," in *Proceedings of the 33rd International Conference and Exhibition on Computer Graphics and Interactive Techniques in conjunction with SIGGRAPH*, 2006.

[59] C. Schldt, I. Laptev, and B. Caputo, "Recognizing human actions: A local svm approach," in *Proceedings of International Conference on Pattern Recognition*, 2004, pp. 32–36.

[60] D. Weinland, R. Ronfard, and E. Boyer, "Free viewpoint action recognition using motion history volumes," *Computer Vision and Image Understanding*, vol. 104, no. 2-3, pp. 249–257, 2006.

[61] Y. Shen and H. Foroosh, "View-invariant recognition of body pose from space-time templates," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2008, pp. 1–6.

[62] ——, "View-invariant action recognition using fundamental ratios," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2008, pp. 1–6.

[63] C. Rao, A. Yilmaz, and M. Shah, "View-invariant representation and recognition of actions," *International Journal of Computer Vision*, vol. 50, no. 2, pp. 203–226, 2002.

[64] M. Holte and T. Moeslund, "View invariant gesture recognition using 3D motion primitives," in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*, 2008, pp. 797 – 800.

[65] I. Cohen and H. Li, "Inference of human postures by classification of 3D human body shape," in *International Workshop on Analysis and Modeling of Faces and Gestures*, 2003, pp. 74–81.

[66] M. Pierobon, M. Marcon, A. Sarti, and S. Tubaro, "Clustering of human actions using invariant body shape descriptor and dynamic time warping," in *Proceedings of IEEE Conference on Advanced Video and Signal Based Surveillance*, 2005, pp. 22–27.

[67] M. A. O. Vasilescu and D. Terzopoulos, "Multilinear analysis of image ensembles: Tensorfaces," in *Proceedings of European Conference on Computer Vision*, 2002, pp. 447–460.

[68] L. D. Lathauwer, B. D. Moor, and J. Vandewalle, "A multilinear singular value decomposition," *SIAM Journal on Matrix Analysis and Applications*, vol. 21, no. 4, pp. 1253–1278, 2000.

[69] L. Elden, *Matrix Methods in Data Mining and Pattern Recognition*. Philadelphia: SIAM, 2007.

[70] D. Vlasic, M. Brand, H. Pfister, and J. Popovi, "Face transfer with multilinear models," in *Proceedings of ACM SIGGRAPH*, 2005, pp. 426 – 433.

[71] M. A. O. Vasilescu and D. Terzopoulos, "Tensortextures: Multilinear image-based rendering," *ACM Transactions on Graphics*, vol. 23, no. 3, pp. 334–340, 2004.

[72] C.-S. Lee and A. Elgammal, "Modeling view and posture manifolds for tracking," in *Proceedings of the IEEE International Conference on Computer Vision*, 2007, pp. 1–8.

[73] H. A. L. Kiers, "An alternating least squares algorithms for parafac2 and three-way dedicom," *Computational Statistics & Data Analysis*, vol. 16, no. 1, pp. 103 – 118, 1993.

[74] J. M. Wang, D. J. Fleet, and A. Hertzmann, "Multifactor Gaussian process models for style-content separation," in *Proceedings of the 24th international conference on Machine learning*, 2007, pp. 975–982.

[75] D. J. C. MacKay, "Introduction to Gaussian processes," in *Neural Networks and Machine Learning*, 1998, pp. 133–166.

[76] C. E. Rasmussen, *Gaussian processes for machine learning*. Cambridge: MIT Press, 2006.

[77] N. Lawrence, "Probabilistic non-linear principal component analysis with Gaussian process latent variable models," *Journal of Machine Learning Research*, vol. 6, pp. 1783–1816, 2005.

[78] J. Nocedal and S. J. Wright, *Numerical Optimization*. Verlag: Springer, 1999.

[79] D. Weinland, E. Boyer, and R. Ronfard, "Action recognition from arbitrary views using 3D exemplars," in *Proceedings of the IEEE International Conference on Computer Vision*, 2007, pp. 1–7.

[80] J. Shi, S. Belongie, T. Leung, and J. Malik, "Image and video segmentation: The normalized cut framework," in *Proceedings of the IEEE International Conference on Image Processing*, 1998, pp. 943–947.

[81] Y. Wu and T. S. Huang, "Vision-based gesture recognition: A review," in *GW '99: Proceedings of the International Gesture Workshop on Gesture-Based Communication in Human-Computer Interaction*. London, UK: Springer-Verlag, 1999, pp. 103–115.

[82] M. W. Lee and R. Nevatia, "Integrating component cues for human pose tracking," in *Proceedings of Joint IEEE International Workshop on VS-PETS*, 2005, pp. 41–48.

[83] R. Navaratnam, A. Thayananthan, P. Torr, and R. Cipolla, "Hierarchical part-based human body pose estimation," in *Proceedings of British Machine Vision Conference*, 2005, pp. 1–10.

[84] N. R. Howe, "Silhouette lookup for monocular 3D pose tracking," *Image and Vision Computing*, vol. 25, no. 3, pp. 331–341, 2007.

[85] F. Huang, H. Di, and G. Xu, "Viewpoint insensitive pose representation for action recognition," in *Proceedings of Conference on Articulated Motion and Deformable Objects*, 2006, pp. 143–152.

[86] K. M. Cheung, S. Baker, and T. Kanade, "Shape-from-silhouette of articulated objects and its use for human body kinematics estimation and motion capture," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2003, pp. 77–84.

[87] I.Mikic, M.M.Trivedi, E.Hunter, and P.C.Cosman, "Human body model acquisition and tracking using voxel data," *International Journal of Computer Vision*, vol. 53, no. 3, pp. 199–223, 2003.

[88] I. A. Kakadiaris and D. Metaxas, "Model-based estimation of 3D human motion with occlusion based on active multi-viewpoint selection," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 1996, pp. 81–87.

[89] A. Elgammal and C. Lee, "Inferring 3D body pose from silhouettes using activity manifold learning," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, vol. 2, 2004, pp. 681–688.

[90] R. Rosales and S. Sclaro, "Learning body pose via specialized maps," in *Proceedings of Conference on Neural Information Processing Systems*, 2002, pp. 1263–1270.

[91] R. Li, M. H. Yang, S. Sclaro, and T. P. Tian, "Monocular tracking of 3D human motion with a coordinated mixture of factor analyzers," in *Proceedings of European Conference on Computer Vision*, 2006, pp. 137–150.

[92] D. F. R. Urtasun and P. Fua, "3D people tracking with gaussian process dynamical models," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2006, pp. 238 – 245.

[93] T. B. Moeslund and E. Granum, "A survey of computer vision-based human motion capture," *Computer Vision and Image Understanding*, vol. 81, no. 3, pp. 231–268, 2001.

[94] T. B. Moeslund, A. Hilton, and V. Kruger, "A survey of advances in vision-based human motion capture and analysis," *Computer Vision and Image Understanding*, vol. 104, no. 2, pp. 90–126, 2006.

[95] W. L. Wang and T. Tan, "Recent development in human motion analysis," *Pattern Recognition*, vol. 36, pp. 585–601, 2003.

[96] F. Guo and G. Qian, "Dance pose recognition using wide-baseline orthogonal stereo cameras," in *Proceedings of the IEEE International Conference on Automatic Face and Gesture Recognition*, 2006, pp. 481 – 486.

[97] C. M. Bishop, *Pattern Recognition and Machine Learning*. New York: Springer, August 2006.

[98] V. N. Vapnik, *Statistical Learning Theory*. New York: Wiley, 1998.

[99] C.-W. Hsu, C.-C. Chang, and C.-J. Lin, "A practical guide to support vector classification," Department of Computer Science, National Taiwan University, Tech. Rep., 2003. [Online]. Available: http://www.csie.ntu.edu.tw/ cjlin/papers/guide/guide.pdf

[100] H. Francke, J. R. del Solar, , and R. Verschae, "Real-time hand gesture detection and recognition using boosted classifiers and active learning," in *Advances in Image and Video Technology*. Berlin/Heidelberg: Springer, 2007, pp. 533–547.

[101] G. Ye, J. J. Corso, D. Burschka, and G. D. Hager, "Vics: A modular hci framework using spatiotemporal dynamics," *Machine Vision and Applications*, vol. 16, no. 1, pp. 13–20, 2004.

[102] G. Ye, J. J. Corso, and G. D. Hager, "Gesture recognition using 3D appearance and motion features," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2004, pp. 160–166.

[103] 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*, vol. 27, no. 3, pp. 351–364, 2005.

[104] S. Mitra and T. Acharya, "Gesture recognition: A survey," *IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews*, vol. 37, no. 3, pp. 311–324, 2007.

[105] R. Poppe, "A survey on vision-based human action recognition," *Image and Vision Computing*, vol. 28, no. 6, pp. 976–990, 2010.

[106] A. Bobick and Y. Ivanov, "Action recognition using probabilistic parsing," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 1998, pp. 196–202.

[107] L. Gorelick, M. Blank, E. Shechtman, M. Irani, and R. Basri, "Actions as space-time shapes," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 29, no. 12, pp. 2247–2253, 2007.

[108] A. F. Bobick and J. W. Davis, "The recognition of human movement using temporal templates," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 23, pp. 257–267, 2001.

[109] B. Peng, G. Qian, and S. Rajko, "View-invariant full-body gesture recognition from video," in *Proceedings of the International Conference on Pattern Recognition*, 2008, pp. 1–5.

[110] S. Ali and M. Shah, "Human action recognition in videos using kinematic features and multiple instance learning," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 32, no. 2, pp. 288–303, 2010.

[111] L. R. Rabiner, "A tutorial on hidden Markov models and selected applications in speech recognition," *Proceedings of the IEEE*, vol. 77, no. 2, pp. 257–286, 1989.

[112] H. Sakoe, "Dynamic programming algorithm optimization for spoken word recognition," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 26, pp. 43–49, 1978.

[113] A. Mccallum, D. Freitag, and F. Pereira, "Maximum entropy markov models for information extraction and segmentation," in *Proceedings of International Conference on Machine Learning*. Morgan Kaufmann, 2000, pp. 591–598.

[114] J. D. Lafferty, A. McCallum, and F. C. N. Pereira, "Conditional random fields: Probabilistic models for segmenting and labeling sequence data," in *Proceedings of International Conference on Machine Learning*, 2001, pp. 282–289.

[115] N. Nair and T. Sreenivas, "Multi-pattern dynamic time warping for automatic speech recognition," in *TENCON 2008, IEEE Region 10 Conference*, 2008, pp. 1–6.

[116] C. Myers, L. Rabiner, and A. Rosenberg, "Performance tradeoffs in dynamic time warping algorithms for isolated word recognition," *IEEE Transactions on Acoustics, Speech and Signal Processing*, vol. 28, no. 6, pp. 623–635, 1980.

[117] A. Pikrakis, S. Theodoridis, and D. Kamarotos, "Recognition of isolated musical patterns using context dependent dynamic time warping," *IEEE Transactions on Speech and Audio Processing*, vol. 11, no. 3, pp. 175–183, 2003.

[118] J. Alon, V. Athitsos, Q. Yuan, and S. Sclaroff, "Simultaneous localization and recognition of dynamic hand gestures," in *Proceedings of the IEEE Workshop on Motion and Video Computing*, 2005, pp. 254–260.

[119] J. Lichtenauer, E. Hendriks, and M. Reinders, "Sign language recognition by combining statistical dtw and independent classification," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 30, no. 11, pp. 2040–2046, 2008.

[120] T. G. Dietterich, "Machine learning for sequential data: A review," in *Proceedings of the Joint IAPR International Workshop on Structural, Syntactic, and Statistical Pattern Recognition*, 2002, pp. 15–30.

[121] H.-D. Yang, A.-Y. Park, and S.-W. Lee, "Robust spotting of key gestures from whole body motion sequence," in *Proceedings of the IEEE International Conference on Automatic Face and Gesture Recognition*, 2006, pp. 231–236.

[122] J. Llinas and E. Waltz, *Multisensor Data Fusion*. Boston: Artech House, 1990.

[123] M. E. Liggins, D. L. Hall, and J. Llinas, *Handbook of Multisensor Data Fusion: Theory and Practice*. Boca Raton, FL: CRC Press, 2008.

[124] J. Gu, X. Ding, S. Wang, and Y. Wu, "Action and gait recognition from recovered 3-d human joints," *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, vol. 40, no. 4, pp. 1021 –1033, 2010.

[125] B. Peng, G. Qian, and S. Rajko, "View-invariant full-body gesture recognition via multilinear analysis of voxel data," in *Proceedings of IEEE/ACM International Conference on Distributed Smart Cameras*, 2009, pp. 1–8.

[126] P. Yan, S. M. Khan, and M. Shah, "Learning 4d action feature models for arbitrary view action recognition," in *Proceedings of IEEE International Conference on Computer Vision and Pattern Recognition*, 2008, pp. 1 –8.

[127] G. Srivastava, H. Iwaki, J. Park, and A. Kak, "Distributed and lightweight multi-camera human activity classification," in *Proceedings of IEEE/ACM International Conference on Distributed Smart Cameras*, 2009, pp. 1 –8.

[128] J. Kittler, M. Hatef, R. Duin, and J. Matas, "On combining classifiers," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 20, no. 3, pp. 226 –239, 1998.

[129] S. Cherla, K. Kulkarni, A. Kale, and V. Ramasubramanian, "Towards fast, view-invariant human action recognition," 2008, pp. 1 –8.

[130] J. Liu and M. Shah, "Learning human actions via information maximization," in *Proceedings of IEEE International Conference on Computer Vision and Pattern Recognition*, 2008, pp. 1 –8.

[131] K. Reddy, J. Liu, and M. Shah, "Incremental action recognition using feature-tree," in *Proceedings of IEEE International Conference on Computer Vision*, 2009, pp. 1010 – 1017.

[132] J. Park, P. C. Bhat, and A. C. Kak, "A look-up table based approach for solving the camera selection problem in large camera networks," in *Proceedings of the International Workshop on Distributed Smart Cameras*, 2006.

[133] S. Soro and W. Heinzelman, "Camera selection in visual sensor networks," in *Proceedings of the 2007 IEEE Conference on Advanced Video and Signal Based Surveillance*, 2007, pp. 81–86.

[134] L. Tessens, M. Morbee, H. Lee, W. Philips, and H. Aghajan, "Principal view determination for camera selection in distributed smart camera networks," in *Proceedings of the Second ACM/IEEE International Conference on Distributed Smart Cameras*, 2008, pp. 1–10.

[135] M. Morbee, L. Tessens, H. Lee, W. Philips, and H. Aghajan, "Optimal camera selection in vision networks for shape approximation," in *Proceedings of the IEEE Workshop on Multimedia Signal Processing*, 2008, pp. 46–51.

[136] H. Lee, L. Tessens, M. Morbee, H. Aghajan, and W. Philips, "Sub-optimal camera selection in practical vision networks through shape approximation," in *Proceedings of the 10th International Conference on Advanced Concepts for Intelligent Vision Systems*. Berlin, Heidelberg: Springer-Verlag, 2008, pp. 266–277.

[137] Y. Li and B. Bhanu, "Utility-based dynamic camera assignment and hand-off in a video network," in *Proceedings of the Second ACM/IEEE International Conference on Distributed Smart Cameras*, 2008, pp. 1–9.

[138] ——, "Task-oriented camera assignment in a video network," in *Proceedings of the 16th IEEE international conference on Image processing*, 2009, pp. 3437–3440.

[139] B. Peng and G. Qian, "Online gesture spotting from visual hull data," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 33, no. 6, pp. 1175 – 1188, 2011.

[140] D. M. J. Tax, M. van Breukelen, R. P. W. Duin, and J. Kittler, "Combining multiple classifiers by averaging or by multiplying?" *Pattern Recognition*, vol. 33, no. 9, pp. 1475 – 1485, 2000.

[141] Y. Chen, H. Huang, W. Xu, R. I. Wallis, H. Sundaram, T. Rikakis, T. Ingalls, L. Olson, and J. He, "The design of a real-time, multimodal biofeedback system for stroke patient rehabilitation," in *Proceedings of the 14th annual ACM international conference on Multimedia*, 2006, pp. 763 – 772.

[142] X. Zhao and Y. Liu, "Generative tracking of 3D human motion by hierarchical annealed genetic algorithm," *Pattern Recognition*, vol. 41, no. 8, pp. 2470 – 2483, 2008.

[143] I. Rius, J. Gonzlez, J. Varona, and F. X. Roca, "Action-specific motion prior for efficient bayesian 3D human body tracking," *Pattern Recognition*, vol. 42, no. 11, pp. 2907 – 2921, 2009.

[144] R. Li, T.-P. Tian, S. Sclaroff, and M.-H. Yang, "3D human motion tracking with a coordinated mixture offactor analyzers," *International Journal of Computer Vision*, vol. 87, pp. 170–190, 2010.

[145] T.-P. Tian, R. Li, and S. Sclaroff, "Tracking human body pose on a learned smooth space," Technical Report 2005-029, Boston University, 2005.

[146] A. Agarwal and B. Triggs, "Tracking articulated motion with piecewise learned dynamical models," in *Proceedings of European Conference on Computer Vision*, 2004.

[147] R. Poppe, "Evaluating Example-based Pose Estimation: Experiments on the HumanEva Sets," in *CVPR 2nd Workshop on Evaluation of Articulated Human Motion and Pose Estimation (EHuM2)*, 2007, pp. 1–8.

[148] A. Sundaresan and R. Chellappa, "Model driven segmentation and registration of articulating humans in laplacian eigenspace," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 30, no. 10, pp. 1771–1784, 2008.

[149] J.-W. Hsieh, C.-H. Chuang, S.-Y. Chen, C.-C. Chen, and K.-C. Fan, "Segmentation of human body parts using deformable triangulation," *IEEE Transactions on Systems, Man and Cybernetics, Part A: Systems and Humans*, vol. 40, no. 3, pp. 596 –610, 2010.

[150] R. Urtasun and T. Darrell, "Sparse probabilistic regression for activity-independent human pose inference," in *Proceedings of IEEE Conference onComputer Vision and Pattern Recognition*, 2008, pp. 1–8.

[151] L. Bo and C. Sminchisescu, "Twin gaussian processes for structured prediction," *International Journal of Computer Vision*, vol. 87, pp. 28–52, 2010.

[152] M. E. Tipping and A. Smola, "Sparse bayesian learning and the relevance vector machine," *Journal of Machine Learning Research*, vol. 1, pp. 211–244, 2001.

[153] L. Sigal, A. Balan, and M. Black, "HumanEva: Synchronized Video and Motion Capture Dataset and Baseline Algorithm for Evaluation of Articulated Human Motion," *International Journal of Computer Vision*, vol. 87, no. 1, pp. 4–27, 2010.

[154] L. Bo and C. Sminchisescu, "Structured output-associative regression," in *Proceddings of IEEE Conference on Computer Vision and Pattern Recognition*, 2009.