Recognition of Human Actions based on 3D Pose Estimation via Monocular Video Sequences

Shian-Ru Ke

A dissertation
submitted in partial fulfillment of the
requirements for the degree of

Doctor of Philosophy

University of Washington

2014

Reading Committee:

Jenq-Neng Hwang, Chair

Linda Shapiro

Shwetak N. Patel

Program Authorized to Offer Degree:

Department of Electrical Engineering

University of Washington

Abstract

Recognition of Human Actions based on 3D Pose Estimation via Monocular Video Sequences

Shian-Ru Ke

Chairperson of the Supervisory Committee:
Professor Jenq-Neng Hwang
Department of Electrical Engineering

We propose a system to recognize both isolated and continuous human actions, from monocular video sequences, based on 3D human pose estimation and cyclic hidden Markov models (CHMMs). First, for each frame in a monocular video sequence, a 3D human pose estimation scheme is applied to extract the 3D coordinates of joints of the human object with actions of multiple repeated cycles. The 3D coordinates are then converted into a set of geometrical relational features (GRFs) for dimensionality reduction and improved discrimination. For further dimensionality reduction, the k-means clustering is applied to those GRFs to generate clustered feature vectors. These vectors are used to train CHMMs separately for different types of actions based on the Baum-Welch reestimation algorithm. For recognition of continuous actions, which are concatenated from several distinct types of actions, a designed graphical model is used to systematically concatenate different separately trained CHMMs. The accurate estimation of the 3D human poses, the effective use of GRFs and CHMMs significantly improve the performance of both isolated and continuous human action recognition problems.

Table of Contents

List of Figures

List of Tables

Acknowledgements

I would like to express my deepest appreciation to my advisor, Professor Jenq-Neng Hwang, for his guidance and support for my PhD study. He continually and convincingly conveyed a spirit of adventure in regard to research and scholarship. His advice on both research as well as on my career have been priceless. Without his support and advice, this dissertation would have not been possible.

Besides my advisor, I would like to thank my PhD committee members, Professor Mark Ganter, Professor Linda Shapiro, Professor Shwetak Patel and Professor Howard Chizeck for their valuable suggestions and career advices.

Many thanks to Professor Maya Gupta and Professor Maryam Fazel for the insightful discussions on my research.

I am grateful to my colleagues and lab mates, Chih-Wei Huang, Victor Gau, Po-Han Wu and Chun-Te Chu for friendly support and huge help during my first year. Without their great help, I could not settle down my life in US and focus on my research.

I thank all the present members in Information Processing Lab: Kevin Lau, Xiang Chen, Meng-Che Chuang, Kuan-Hui Lee, Pei-An Lee, YoungGun Lee and Po-Han Wu, and the recently graduated members: Ruizhi Sun, YoungDae Lee and Chun-Te Chu. They have made the lab a great working place and provided precious suggestions to my research. Best wishes to all of them.

I also thank to all my friends whom I met in Seattle. With their friendship, I feel energetic in my life.

I will forever thank to my parents and my brothers. They have been supportive during my PhD study. Thanks to their understanding and constant encouragement, I am able to pursue the degree without worries.

Dedication

To my family

Chapter 1 – Introduction

Section 1: Motivation and Challenges

Human action recognition is a growing topic in video analysis and understanding - one of the most popular areas in the community of computer vision, thanks to its applications to surveillance, entertainment and healthcare. In surveillance, human actions can be recognized and analyzed from video to secure the society. In entertainment, human-computer interaction (HCI) can be more natural via human action recognition to increase the entertainment experience. In healthcare, human action recognition can detect abnormal gaits or assist patients' rehabilitation through analysis of patients' actions.

Moreover, the monocular video is more general and flexible for the applications. For multiple cameras, there are two main drawbacks. First, the multiple views are not always available. For example, when a pedestrian walks by the hallway in public space, the multiple views of the person may not be easily obtained. Second, multiple cameras require the camera calibration. Different cameras may have different illuminations, lighting conditions, and different angles of views. It took extra effort to have a good camera calibration for multiple cameras. Besides, for a depth camera, it usually requires higher cost. Therefore, we focus on a monocular camera to provide a more general and flexible solution to human action recognition.

However, it is challenging to recognize various human actions due to high degrees of freedom (DOFs) of the human body, large variations of human poses, change

of clothes colors, change of lighting and illumination, various viewpoints, and frequent self-occlusion. Moreover, the use of monocular video sequences further increases the difficulty for human action recognition.

Therefore, we propose a system to recognize single actions and continuous human actions concatenated from different types of actions by overcoming the challenges. In order to deal with changes of illumination, changes of clothes, changes of viewpoints and occlusions, the 3D human pose estimation [26], [27] is considered. The estimated 3D coordinates of human joints are further converted into Geometrical Relational Features (GRFs) [62], [63] as feature representation, and Cyclic Hidden Markov Models (CHMM) [65] are applied for the task of single action recognition based on the GRFs. Moreover, the graphical models designed for switching CHMMs are used to recognize continuous actions [46].

Section 2: Introduction to Pose Estimation

Pose estimation is the process of identifying the configuration of different body parts, including head, torso, limbs, hands and feet. Usually pose estimation follows the segmentation and tracking processes as shown in Figure 1.1.

Figure 1.1: The general process for pose estimation.

Based on how a human model is applied, pose estimation can be divided into three classes [1], [2]:

(i) Model-Free: no a priori model is used. The poses are represented as points such as head and hand blobs in [4], [14], simple shapes such as ellipses in [5] and stick-figures [6], [11], [18]. In this class, pose estimation relies on extensive training using ground truth data obtained by commercial motion capture systems.

(ii) Indirect Model: a priori model is used as a reference or look-up table from which relevant information may be extracted to guide the interpretation of measured data. In [7], the 2D ribbons, which are U-shaped edge segmentations, are used to describe the outline of the subject, and guide the labeling of the image data by searching for structure similar to the 2D ribbon model. A 2D model of a human's head-shoulder-upperbody [9] is also proposed to detect and track humans with particle filtering [19]. In this class, the estimated poses are generally not very detailed. It is not easy to handle occlusion.

(iii) Direct Model: a priori human model (explicit 3D geometric representation of human shape and kinematic structure) is directly used as the model representing the observed subject and is continuously updated by the observations. Many methods [10], [12], [13], [15], [16], [17], [23] have been proposed to reconstruct human poses with a human model. The majority of approaches in this class employ an analysis-by-synthesis methodology to optimize the similarity between the model projection and observed images. By introducing a human model, it is

able to handle occlusion and easy to incorporate various kinematic constraints into a system.

In order to incorporate kinematic constraints, handle occlusion, and reconstruct 3D poses from monocular video sequence, a predefined 3D human model is used in our proposed system.

Section 3: Introduction to Human Action Recognition

In recent decades, many papers have addressed human action recognition. Generally, two main stages are considered in human action recognition; one is the feature extraction and representation stage and the other is the classification stage [3].

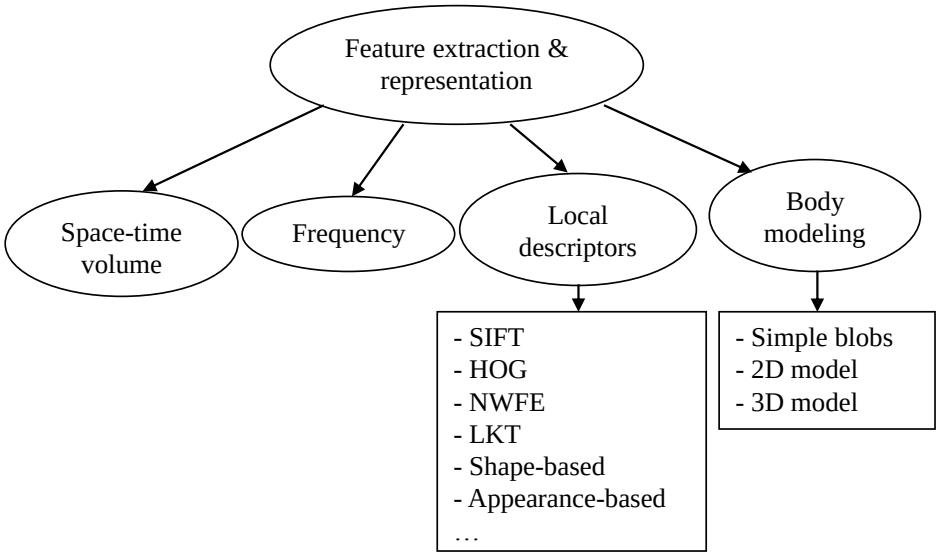


Figure 1.2: The categories for feature extraction and representation.

In the stage of feature extraction and representation, the features or characteristics of video frames such as silhouette, shape, color and motion are extracted and represented in a systematic and efficient way. Generally speaking, the features can be categorized into four groups: space-time information, frequency transform, local descriptors and body modeling as shown in Figure 1.2.

The space-time information is considered first. The space-time volume (STV) [53], [54], [55], [74] is built from the image features by concatenating the consecutive silhouettes of objects along the time axis. The extracted 3D XYT volume (along x-y spatial coordinates and time) can capture the continuity of human actions, but the STV is limited on non-periodic activities. Compared to spatial-temporal domain image, the frequency domain information can also be exploited. More specifically, the discrete Fourier transform (DFT) [75] has been widely used to represent information about the geometric structure of the object. The STV and DFT features belong to global features that consider the whole image, so they are limited on viewpoint changes and occlusion. Hence, some local descriptors are considered. The local descriptors [12], [13], [18], [56], [58], [76], [77], [78], [79], [80], such as SIFT [81], [82] and histogram of oriented gradient (HOG) [83], capture the characteristics of an image patch. They are ideally invariant to background clutters, appearance and occlusions, and also invariant to rotation and scale in some cases. However, the above-mentioned feature representations do not fully capture the whole body actions. Therefore, some human modeling methods [4], [5], [6], [7], [9], [10], [13], [15], [16], [17], [23], [26], [27], [84] are also proposed to model

the human body including simple blobs, 2D body modeling and 3D body modeling. Generally, the body modeling requires the 2D/3D pose estimation problem. Usually, after the pose estimation, the 2D/3D coordinates of the human body are further converted into other dimension-reduced or more discriminative feature representations, such as polar coordinate representation [85], Boolean features [48] and geometric relational features (GRF) [62], [63].

In the stage of the classification algorithm, the selected or converted features are sent to proper classification algorithms for detection and/or recognition. The classification algorithms can generally be categorized as dynamic time warping (DTW), generative models, discriminative models and others as shown in Figure 1.3.

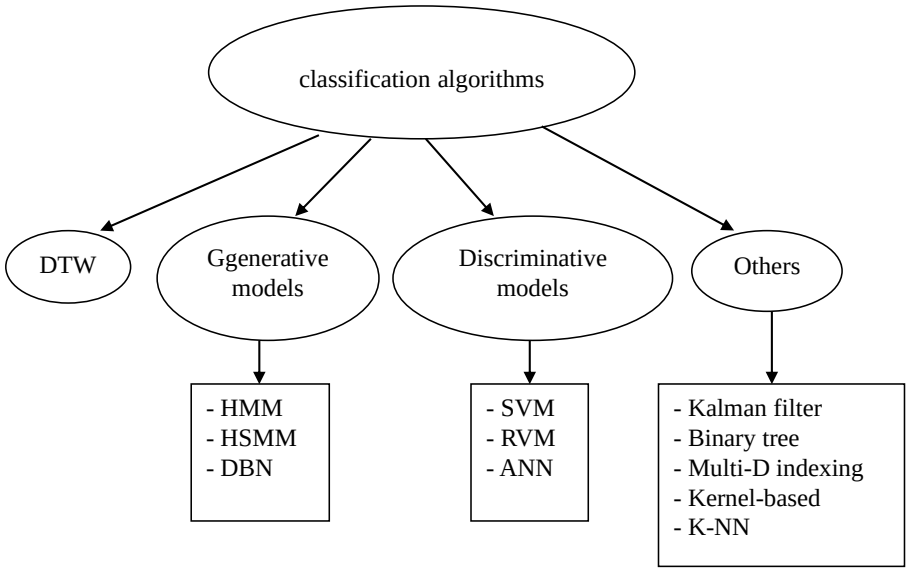


Figure 1.3: The categories for classification algorithms.

The dynamic time warping (DTW) [56], [86], a method for measuring similarity between two temporal sequences, which may vary in time or speed, is one of the most

common temporal classification algorithms due to its simplicity; however, DTW is not appropriate for a large number of classes with many variations. Some probability-based methods by generative models (dynamic classifiers) are proposed such as Hidden Markov Models (HMM) [50], [65], [66], [67], [87], [88], [89] and Dynamic Bayesian Network (DBN) [49], [90], [91]. On the other hand, discriminative models (static classifiers) such as Support Vector Machine (SVM) [52], [72], [92], [93], Relevant Vector Machine (RVM) [12], [21], [22] and Artificial Neural Network (ANN) [94], [95], [96], can also be used in this stage. In addition to the dynamic and static classifier difference nature, another main difference between generative models and discriminative models [97] is that the generative classifiers commonly learn a model of the joint probability, $p(x,y)$, of the input x and the label y, or equivalently the likelihood $p(x|y)$ according to the Bayes rule; while the discriminative classifiers model the posterior $p(y|x)$ directly. Therefore, the generative models can be used to simulate values of any variables in the models, while the discriminative models allow only sampling of the target variables conditional on the observed variables. For both of the probability model-based algorithms, including generative models and discriminative models, their performance relies on extensive training dataset. Therefore, other methods are proposed, such as Kalman filter [36], [98], binary tree [99], [100], multidimensional indexing [101], and K nearest neighbor (K-NN) [75]. Different classification algorithms usually require different sets of suitable feature representations.

Section 4: Contributions

The contributions of this dissertation for 3D human pose estimation and human action recognition are listed as follows.

In the part of 3D Human Pose Estimation:

- We proposed a front-view system for 3D pose estimation, which can satisfy the real-time requirement.
- We proposed a 2D body part tracking method by integrating shape, appearance, and time information.
- We proposed a mechanism to perform occlusion detection and handling for body parts.
- We designed a cost function to describe the difference between 2D features and 3D models, and applied the downhill simplex algorithm to search for the best 3D pose.
- We proposed a method of constrained multiple kernel tracking for human limbs.
- The proposed method is tested on several public datasets, and shows better performance by compared with other state-of-the-art methods.

In the part of Human Action Recognition:

- We designed GRFs (Geometrical Relational Features) for feature representation to increase discrimination and reduce dimensionality.
- We proposed a Cyclic HMM for the classification algorithm on single human action recognition.

- We designed a graphical model for continuous human action recognition.
- The proposed method is tested on several public datasets, and shows higher recognition rate than other state-of-the-art methods.
- The proposed method shows satisfactory recognition rate in single action recognition and fixed-length/variable-length continuous action recognition.

Section 5: Dissertation Roadmap

The dissertation is organized as followed:

Chapter 2: We review the recent works on human pose estimation and human action recognition. In human pose estimation, we review the related methods on the single camera and multiple cameras. In human action recognition, we review the related methods for feature representation and classification algorithms. Finally, we also briefly describe our considerations and proposed methods.

Chapter 3: The proposed human pose estimation methods are described here. First, a real-time front-view human pose estimation is described. Second, the view-invariant human pose estimation method is discussed. Third, the proposed method of constrained multiple kernel tracking on human limbs is described. Finally a discussion session is given.

Chapter 4: The proposed method for human action recognition is described here. First, a feature conversion method is proposed. Second, the classification algorithm is described. Third, the experiments on single human action recognition are described. Fourth, the

experiments on fixed-length and variable-length continuous human action recognition are mentioned. Finally, a discussion session is given.

Chapter 5: The conclusion of this dissertation is given by summarizing our main contributions. Moreover, the discussion of the extension work and potential research topics in the future is given.

Chapter 2 – Related Work

Section 1: Pose Estimation

To overcome the difficulties including various poses and self-occlusion, and reconstruct human poses, many methods have been proposed with single-view (monocular) [4], [6], [7], [11], [12], [13], [14], [15], [16], [17], [18], [23] or multiple-view [5], [8], [9], [10] images or video sequences. For multiple-view pose estimations [5], [8], [9], [10], multiple cameras are calibrated and synchronized to reconstruct 3D human poses by using a State Transition Map (STM) in [5], a pose search cycle in [8], 20 features components in [9], or feature identification in [10]. However, in many situations of the real world, multiple views for one subject may not be available, and the calibration and synchronization parameters of multiple cameras are not available either. Most video sequences for multiple-view images are not captured from daily life. Therefore, more and more pose estimation methods use monocular video sequences.

Pose estimation for single-view images or video sequences is considerably more difficult than multiple-view images or video sequences. In [4], [14], the skin blobs are detected as head and two hands, and the best poses are estimated by a maximum a posteriori (MAP) approach in [4] or silhouette matching in [14]. Moreover, in [6], [11], [18], the appearance models, such as stick images and pictorial structure, are learned by a genetic-based algorithm in [6], a linear Support Vector Machine (SVM) in [11], or a bottom-up approach for candidate body parts and a top-down approach for an entire

person in [18]. Sedai et. al. [13] even proposed a local appearance context (LAC) descriptor, which is trained by Relevance Vector Machine (RVM) [21], [22] regression.

Nevertheless, appearance information is easily influenced by illumination change and clothing color change due to the rotation of the subject. Therefore, edge-like features are used in [7]. Agarwal and Triggs [12] proposed a shape descriptor based on image silhouettes, and train shape descriptors by RVM regression. Moreover, Rogez et. al. [23], [24] use Gaussian Mixture Models (GMM) to fit a feature space made of shape-skeleton figures.

However, shape information is sensitive to noise. Lee, Cohen and Nevatia [15], [16], [17] combined edge (shape) and foreground (appearance) as features, and human poses are estimated by data-driven Markov Chain Monte Carlo (DD-MCMC) [20]. But the computation is significantly high with an average of 5 minutes for each frame.

In order to reach real-time requirement, the downhill simplex algorithm [25] is applied to efficiently reconstruct 3D human poses in our proposed real-time front-view 3D human pose estimation system [26]. Moreover, to reconstruct 3D human poses in arbitrary angles, the body orientation is estimated first, and then the shape, color and time continuity information are integrated for body parts tracking in our proposed view-invariant 3D human pose estimation system [27].

Section 2: Human Action Recognition

In the survey of the task of human action recognition [3], two main stages are considered: 1) the feature extraction and representation stage and 2) the classification stage.

In the stage of feature extraction and representation, the features or characteristics of video frames such as silhouette, shape, color and motion are extracted and represented in a systematic and efficient way. In a video sequence, the features that capture the space and time relationship are known as a space-time volume (STV). The space-time correlation is one of the most popular features in the video analysis community. Blank et al. [53] propose a method by stacking segmented silhouettes frame-by-frame to form a 3D spatial-temporal shape, from which the space-time features such as local space-time saliency, action dynamics, shape structure and orientation can be extracted. In a similar way, Ke et al. [54] build the STV as image features based on the consecutive silhouettes of objects along the time axis for shape-based matching, including spatial-temporal region extraction and region matching. Kim et al. [102] propose a spatio-temporal approach to detect the salient region in images and videos. Laptev et al. [71], [72] propose a method to extract space-time interest points (STIP) by maximizing a normalized spatiotemporal Laplacian operator over spatial and temporal scales. In addition to space and time information, the frequency domain information of an image is also considered. Kumari and Mitra [75] use discrete Fourier transforms (DFTs) of small uniformly partitioned image blocks as the selected features for activity recognition. The

STV, STIP and DFT are global features which are extracted by globally considering the whole image.

However, the global features are sensitive to noise, occlusion and variations of viewpoints. Therefore, some local descriptors are used to capture the characteristics of an image patch, such as the scale-invariant feature transformation (SIFT) [81], [82], histogram of oriented gradient (HOG) [83], nonparametric weighted feature extraction (NWFE) features [58] and Lucas-Kanade-Tomasi (LKT) features [103], [104]. For example, Scovanner et al. [78] introduce a 3D SIFT descriptor, which can reliably capture the spatio-temporal nature of video sequences as well as 3D imagery such as MRI data. The SIFT descriptor is popular due to its invariance to image rotation and scale and robustness to affine distortion, noise corruption and illumination changes. But SIFT has issues of high dimensionality and insufficient discrimination. Lu and Little [76] propose a template-based algorithm to track and recognize an athlete's actions based on a PCA-HOG descriptor. Moreover, Kataoka and Aoki [105] propose a method of extension of CoHOG (Co-occurrence Histograms and Oriented Gradients) for pedestrian detection. However, HOG features are extracted at a fixed scale; therefore, the size of the human body in the image has great influence on the performance. Moreover, by considering the distance information and the width feature of a silhouette, Lin et al. [58] design a new feature, called nonparametric weighted feature extraction (NWFE) for human activity recognition. Unfortunately, NWFE does not take advantage of the color appearance information. Furthermore, Lucas-Kanade [81] and Tomasi [82] propose a LKT feature

tracker based on the sum of squared intensity differences. Lu et al. [77] use an LKT feature tracker to track human joints and recognize non-rigid human actions. Since the LKT feature tracker assumes that neighboring pixels in a small window have the same flow vector, it poses a limitation to deal with large motion between frames.

Due to limitations on 2D global and local features, Weinland et al. [59] use 3D exemplars, 3D occupancy grids, as features. Without 3D reconstruction, the learned 3D exemplars are used to produce 2D silhouette frames for matching. However, their method [59] is limited to view variations. Junejo et al. [70] resolve the changes of view angles by Self-Similarity Matrix (SSM), obtained by computing self-similarities (distance between low level features) of action sequence over time. But SSM is only useful for body occlusion.

In order to resolve body occlusion and take body configuration into account, 3D human modeling is thus considered. Subsequently, the estimated 3D coordinates are converted into a feature space. For 3D human modeling, Rogez et al. [23] use a series of view-based shape-skeleton models for video surveillance systems by projecting the input image frames onto the training plane. But it needs extensive training dataset due to various viewpoints. Lee and Nevatia [16], [17] use a multi-level structure model for 3D pose estimation from monocular video sequences by addressing automatic initialization, data association, self- and inter- occlusions. The 3D human poses are inferred based on a method of data-driven Markov Chain Monte Carlo (DD-MCMC) [106], [107]. But the computation cost is extremely high. Considering the accuracy of pose estimation and the

time complexity at the same time, Ke et al. [26], [27] propose a method to track 2D body parts by integrating shape, color and temporal information to effectively estimate 3D human poses.

Generally, the 3D coordinates of human joints can be further converted into low-dimensional or more discriminative features, such as polar coordinate representation [85], Boolean features [48] and geometrical relational features (GRFs) [62], [63], for effective recognition purpose.

In the stage of the classification algorithm, the selected or converted features are sent to proper classification algorithms for detection and/or recognition. One of the well-known classification algorithms is dynamic time warping (DTW) [108], which is a similarity measurement between two sequences by a dynamic programming approach. Sempena et al. [86] use DTW to recognize various human activities such as waving, punching and clapping. DTW is simple and fast, but it might need extensive templates for various situations, resulting in high computation cost to match with these extensive templates.

Besides DTW, probability-based methods by discriminative models and generative models for classification have also been proposed. Discriminative models learn posterior probability distribution P(Y|X), of a specific class label Y given the observed variable X. The support vector machine (SVM) [92], [93] is one of the most popular discriminative models, which are to find the optimal dichotomic hyperplane that can maximize the margin of two classes. Schuldt et al. [52] apply SVMs to recognize

human activities by extracting local space-time features in a video. The main drawback of an SVM is the high computation burden for the constrained optimization programming used in the learning phase. On the other hand, generative models learn the joint probability distribution P(X,Y), which can be used to generate samples from the distribution. The hidden Markov model (HMM) [66] is one of the most popular generative models, which follow a doubly stochastic process with an underlying hidden first-order Markov stochastic process and an observed stochastic process that can produce the sequence of observed symbols. Yamato et al. [50] train HMMs based on the low-level image mesh features [109] to recognize actions of different tennis strokes by the Baum-Welch re-estimation algorithm. Considering the multiple cycles of human actions, Thuc et al. [65] proposed a cyclic HMM (CHMM) to effectively adapt to most quasi-periodic human action recognition tasks. But each separately trained CHMM can only recognize a single action, rather than continuous actions with different types of concatenated actions during the testing phase.

In this work, we propose a system to recognize single actions and continuous human actions concatenated from different types of actions [46] through monocular video sequences. Our considerations are shown as follows. First, in order to take the body configuration into account and handle occlusion, a 3D pose estimation method is proposed. Second, in order to increase discrimination of features and reduce dimension, a feature representation, named GRF (Geometrical Relational Feature) is proposed. Third, in order to deal with time sequential data, a classification algorithm, called CHMM

(Cyclic Hidden Markov Model), is proposed. Last, in order to recognize continuous human actions, a graphical model is designed.

Chapter 3 – 3D Human Pose Estimation

Section 1: Real-Time Front-View 3D Human Pose Estimation

We propose an approach to automatically estimate front-view 3D human poses from monocular videos with real-time and robust performance. An analysis-by-synthesis strategy is used to decompose the human body into different parts [28]. Only body parts that have been detected as moving are tracked by using multiple cues such as silhouette, edge and color. In the tracking stage, results of 2D feature (the location of head, hands, and feet) tracking are integrated with 3D tracking to improve the robustness and efficiency of our approach. In addition, tracking failures are well-handled so that the approach is less sensitive to accumulated tracking errors over long video sequences.

3.1.a System Overview

The overview of the proposed system is shown in Figure 3.1, including human body detection, 2D feature extraction, 2D feature tracking and 3D pose reconstruction modules. Initially, a Gaussian model is built for each location in the image plane from a few background frames. Then, the human body detection module is triggered by skin color model to detect whether a person enters into a scene, so that the person is segmented as the foreground. If a person is detected, the 2D feature extraction module will extract 2D features of the foreground including silhouette, edge and skin, and a mean-shift tracking model [29]-[31] will be appropriately initialized. Moreover, face,

hands and feet are tracked by a 2D feature tracking module. According to those 2D features, a 3D human body model is initialized and 3D poses of each decomposed body part in each frame are reconstructed by using a downhill simplex search algorithm [25] to minimize the cost between the projected 2D features from 3D poses and the extracted 2D features from video frames. The tracking if lost and occlusion events are also handled by the 3D pose reconstruction module. Finally, the results are sent to two proposed applications: one is to detect and recognize various surveillance events such as lifting a bag. The other is for 3D video game driven by the 3D tracking results.

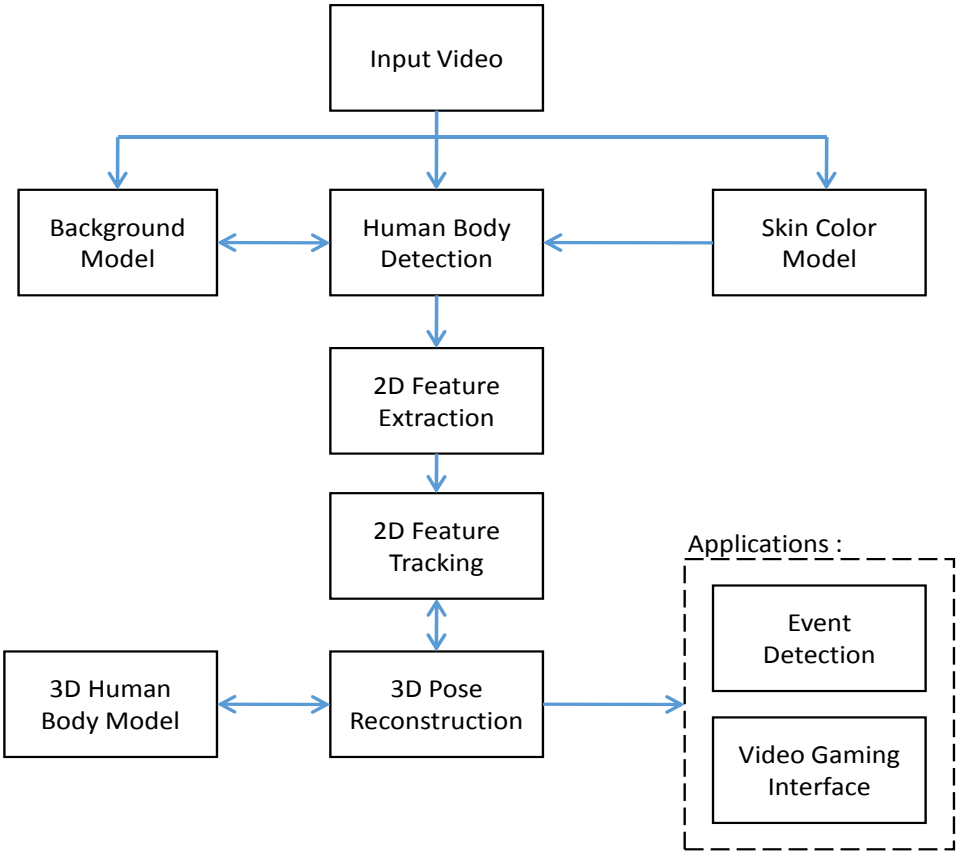


Figure 3.1: System overview of the front-view 3D human pose estimation.

3.1.b 2D Tracking

(i) Human Body Detection

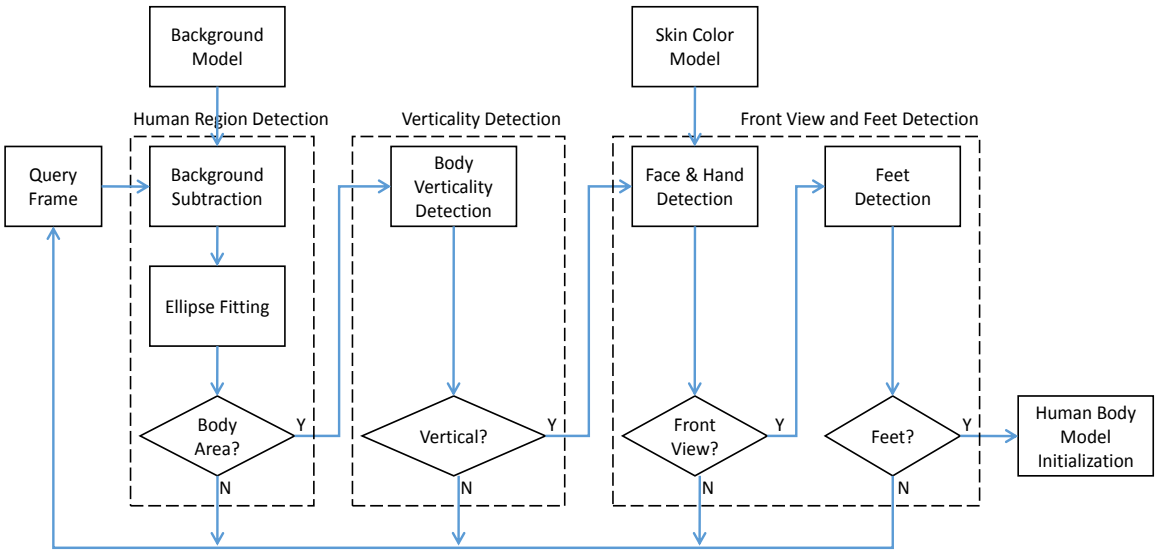


Figure 3.2: The flow chart of human body detection.

For better initialization of a 3D body model, the human being is initially assumed to enter the camera view in a predefined region of interest (ROI) with the full body, stand vertically with hands naturally hanging down, and face the camera. The flow chart of human body detection is shown in Figure 3.2. Firstly, the human region is extracted in each frame by using the background subtraction method. Once the human region is extracted and the verticality is checked successfully, the skin blobs of the face and hands are classified through a skin color model. Finally, the feet are detected by the high gradient at the bottom of the input frame. Once head, hands and feet blobs are detected successfully, the 3D human pose reconstruction process will be triggered.

(ii) Feature Extraction

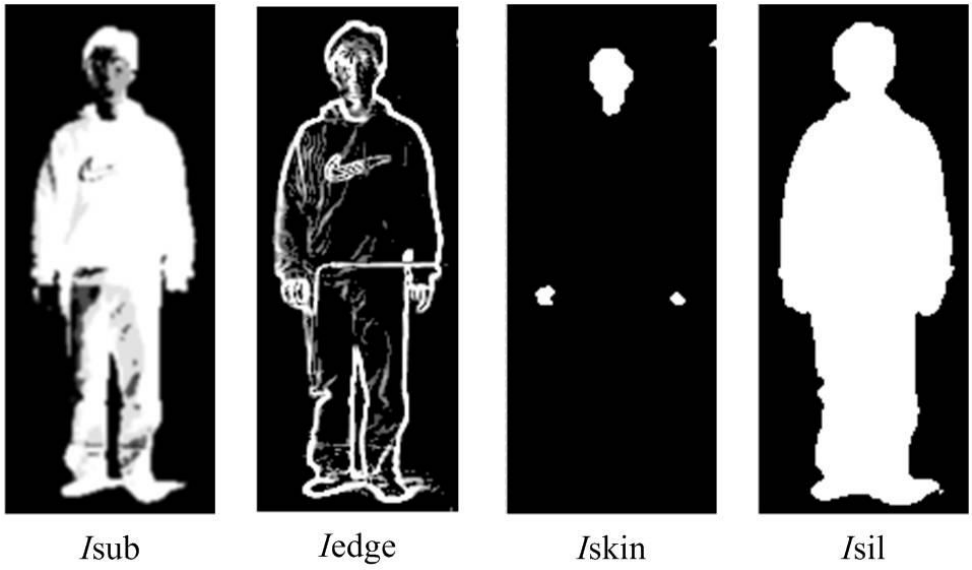


Figure 3.3: The silhouette image I_{sil} is composed as $I_{sub} \cup I_{edge} \cup I_{skin}$.

The subtracted image, I_{sub}, is obtained by background subtraction. The edge image, I_{edge}, is obtained by the Sobel operator. The first order derivative images of the current frame in X and Y directions are calculated by using the Sobel operator. Then, the two resulting derivative images are subtracted by two background derivative images, separately. Moreover, the gradient magnitude image is then computed from the subtracted derivative images. Finally, the resulting image is thresholded to obtain the edge image I_{edge} of the human body.

The skin image, I_{skin}, is obtained by detecting the skin pixels. The skin blobs are detected by a simple but efficient method [32], where the skin pixels are classified based

on Equation 3.1. The (R_i, G_i, B_i) denotes three color component values of the pixel x_i in the original frame, and τ_{low} and τ_{high} are predefined thresholds used to classify skin pixels. That is, the skin color is classified in a specified region in R-G color space. Subsequently, the morphological operations including closing and opening are performed on the classified skin frame. After that, the connected components, which are the candidates for effective skin blobs, are segmented. The connected components with too small or too large areas are disregarded. In our experiments, τ_{low} and τ_{high} are empirically set as 20 and 80. Finally, the silhouette image, I_{sil}, is obtained by $I_{sub} \cup I_{edge} \cup I_{skin}$, as shown in Figure 3.3.

Equation 3.1: $$\tau_{low} < R_i - G_i < \tau_{high}$$

(iii) Blob Detection

The head and hand blobs are detected by skin pixels. Based on the geometrical configuration and kinematic constraints, some skin blobs can be eliminated from the head/hand blob candidates. The nearest neighbor search (NNS) is applied on the head/hands blob locations in the previous frame to assign the appropriate skin blobs to the corresponding head/hand blobs in the current frame.

For feet detection, firstly, a reference point (cx, cy) is defined, where *cx* is the x component of the face blob and *cy* is the mean value of the y component of two hand blobs. Two tip points (x_0, y_0) and (x_1, y_1) are searched by the high gradient parts at the

bottom of the current frame with the largest Euclidean distance from (cx, cy) and its x component being greater or less than cx. If $abs(y_0 - y_1)$ is smaller than a predefined constant, then the detected two points, (x_0, y_0) and (x_1, y_1), are selected as feet positions as shown in Figure 3.4.

Figure 3.4: Feet identification. The blue circle is the reference point, and red circles are detected feet blobs.

3.1.c 3D Pose Estimation

(i) Tracking Lost Recovery

Once a human body is detected, the mean shift model will start four trackers with individual targets, i.e., RightArm, LeftArm, RightLeg and LeftLeg, for mean shift tracking [29]-[31]. One example of four mean shift targets is shown in Figure 3.5.

Figure 3.5: An example of four targets for mean shift tracking, including right/left arm, and right/left leg.

For each feature point, the tracking is detected to be lost if the weighted distance error between image feature points and projected model positions is greater than a threshold. More specifically, the weighted distance error is computed as $d\left(p_{img}^{i}, p_{\mathrm{mod}\,el}^{i}\right)* pr_i$ at the current frame or an accumulated value in several consecutive frames, where pr_i serves as a reliability factor, which is updated based on the code chip in Figure 3.6.

```
if   (blob_i.flag == active)
     blob_i.pr = 1.0;
else if   (blob_i.flag == lost)
     blob_i.pr = 0.5/(1 + blob_i.lostNum);
else if   (blob_i.flag == occlusion)
     blob_i.pr = 0.5;
```

Figure 3.6: The code chip to update the probability of blobs.

Once one of the four parts, RightArm, LeftArm, RightLeg and LeftLeg is detected as lost tracking, the corresponding mean shift tracker is triggered. Based on the color histogram in the target, the tracker will perform the mean shift tracking algorithm [29]-[31] to identify the best matched area as ROI (region of interest). Then 2D feature images will be constrained in the ROI and used in the cost function (defined in the next section). A lost tracking recovery example is shown in Figure 3.7.

Figure 3.7: A tracking lost recovery example.

(ii) Cost Function Minimization

The cost function $F\left(O_{image}^{2D}, O_{model}^{3D}\right)$, which measured the difference between the 2D features $\left(O_{image}^{2D}\right)$ and the 3D model $\left(O_{model}^{3D}\right)$, is composed of 4 scores, silhouette score C_{sil}, edge score C_{edge}, motion score C_{motion} and feature points score C_{fp}, as defined in Equation 3.2, where N_{XOR}/N_{AND} is the number of pixels after the logical operation XOR/AND; and w_{sil}, w_{edge}, w_{motion} and w_{fp} are weights for each score. The terms in Equation 3.2 are described in Table 3.1.

Equation 3.2: $$\mathrm{F}\left(\mathrm{O}_{image}^{2D}, O_{\mathrm{mod}\,el}^{3D}\right) = w_{sil}C_{sil} + w_{edge}C_{edge} + w_{motion}C_{motion} + w_{fp}C_{fp}$$

Silhouette Score: $$C_{sil} = N_{XOR}\left(S_{image}, S_{\mathrm{mod}\,el}\right)$$

Edge Score: $$C_{edge} = -N_{AND}\left(E_{image}, E_{\mathrm{mod}\,el}\right)$$

Motion Score: $$C_{motion} = -N_{AND}\left(M_{image}, E_{\mathrm{mod}\,el}\right)$$

Feature Point Score: $$C_{fp} = \sum_{i=1}^{5} dist\left(B_{image}^{i}, B_{\mathrm{mod}\,el}^{i}\right) * pr_i$$

Table 3.1: Terms in the Cost Function

Symbol	Description
S_{image}	silhouette image
$S_{\mathrm{mod}\,el}$	3D model projection image
E_{image}	edge image
$E_{\mathrm{mod}\,el}$	3D model outline image
M_{image}	edge motion image
B_{image}^{i}	the location of the *i*th 2D blob, $i \in \{head, rhand, lhand, rfoot, lfoot\}$
$B_{\mathrm{mod}\,el}^{i}$	the location of the *i*th corresponding 3D model projection blob
pr_i	the probability to estimate the degree of accuracy for the *i*th blob. Ex, the occluded blob has a lower pr_i.

The cost function is minimized with the downhill simplex algorithm [25] by fitting the 2D features for the 3D pose projection in an analysis-by-synthesis strategy. In the N-dimension problem, N+1 vertices, whose cost are computed based on the cost function, in N dimensions are generated to form a simplex (polygon in N dimensions). And the four operations, reflection, expansion, one-dimension contraction and multiple contractions are iteratively performed on the simplex to locate the best vertex $v^* \in R^N$ with the minimal cost.

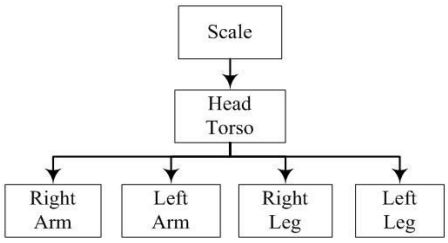
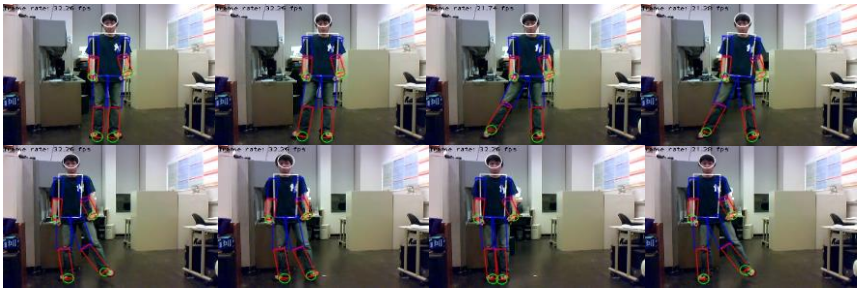


Figure 3.8: The 3D poses are searched hierarchically.

In order to further increase the computation efficiency, the optimized 3D poses are searched in a hierarchical way. As shown in Figure 3.8, the head and torso are taken as a base followed by the four limbs, sequentially right arm, left arm, right leg and left leg.

3.1.d Experimental Results

(a) Long-sleeved case



(b) Short-sleeved case

Figure 3.9: 3D tracking for long-sleeved and short-sleeved cases.

The system is implemented in C++ and runs on a laptop (CPU Intel Core 2 Duo T8100 2.1 GHz, RAM 3 GB, Windows 7). The image resolution is 320x240 pixels and the average frame processing rate is 26~32 frame per second (fps). The system reaches the real-time requirement. Figure 3.9 shows the long-sleeved and short-sleeved cases.

(i) Qualitative Results

The 948 frames in the monocular video sequence are processed by our proposed system to reconstruct 3D human poses for each frame. Some snapshots of the test video are shown in Figure 3.10. Once a person is detected, the system will start to track the person and plot the corresponding 3D model. Even for the occluded case of hands crossing in front of the chest, the person is still well-tracked.

Figure 3.10: Snapshots of the test video.

(ii) Quantitative Results

Videos with ground truth values are obtained from [33]. The disparity is calculated by a two-camera system, and it can be used to calculate the depth information.

The quantitative evaluation is shown in Figure 3.11. AVE_DIST denotes the average distance between the estimated projected 3D blobs and the ground-truth locations over 948 frames. Figure 3.11 shows the average error distances of the right hand, left hand, and even the right elbow joint are around 5 pixels. Moreover, the blue curve is the estimated depth (in pixel) of one blob over 948 frames, while the red curve is the ground-truth depth (in centimeter) obtained from [33] of the blob. Figure 3.11 shows the consistence of the two curves over 948 frames. Therefore, it shows the good performance of our 3D pose estimation method.

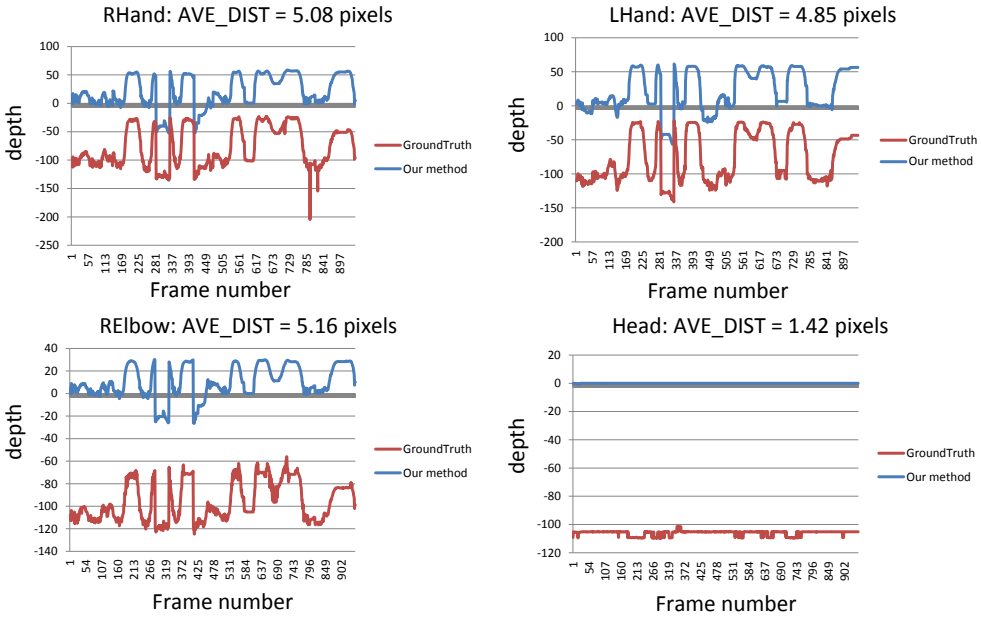


Figure 3.11: AVE_DIST and depth for Head, REblow, RHand, LHand.

Section 2: View-Invariant 3D Human Pose Estimation

Due to large variations of human poses in different viewpoints, self-occlusion, body orientation and depth ambiguity, the 3D pose estimation via monocular camera in an arbitrary viewpoint is considerably more challenging. In our proposed view-invariant 3D human pose estimation system, the shape, color and time continuity information are explored and integrated in a designed process. The occlusions including hand-hand, hand-body and foot-foot occlusion are also detected and handled. Then two phases of 3D pose estimation are proposed to reconstruct 3D poses.

3.2.a System Overview

The overview of the view-invariant 3D human pose estimation system, with monocular video inputs, consists of three stages as shown in Figure 3.12. In Stage 1, the background model is built and the segmentation is performed to extract the foreground object and 2D features, including silhouette, skin, edge, and motion. Besides, the orientation of the human body is estimated based on the scale change and the trajectory of the tracked object. Subsequently, in Stage 2, 2D body parts are tracked based on the information of shape, color and time continuity, which are integrated by a proposed fusion scheme. In the last stage, the 3D model poses are first coarsely estimated by the locations of 2D body parts and the orientation of the body. The 3D poses can then be refined by searching 3D poses in high dimensions. Additionally, the tracking of occluded body parts is also handled. Finally, the video with 3D body pose estimation is generated.

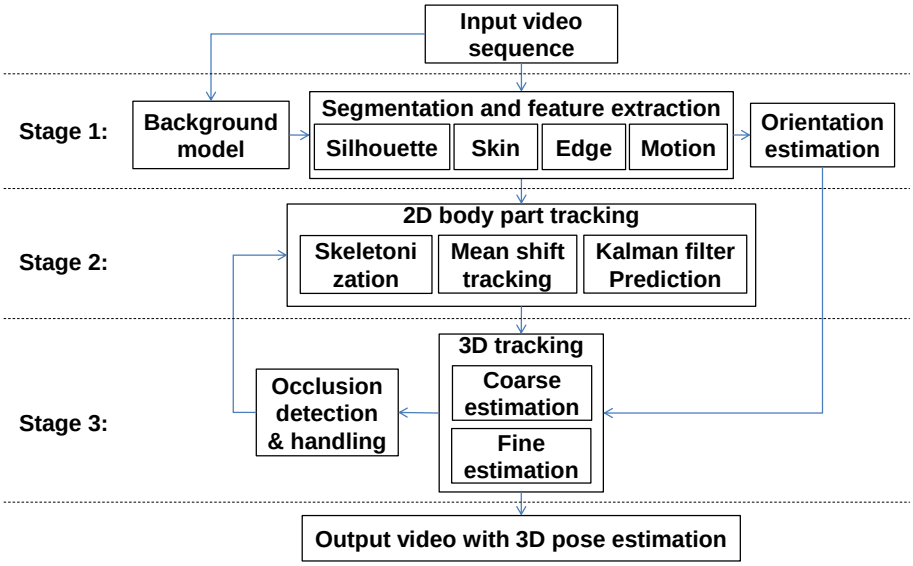


Figure 3.12: System overview of the view-invariant 3D human pose estimation.

3.2.b Segmentation and Feature Extraction

(i) Segmentation with Shadow Removal

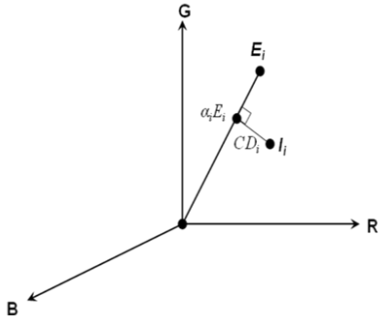


Figure 3.13: In 3D RGB space, E_i denotes the expected color value of the *i*th pixel in the background model, while I_i denotes the current color value of the pixel in the current frame [34].

In order to extract the foreground objects, we employ a robust statistical background subtraction with shadow removal approach [34], in which a Gaussian

mixture model for each pixel of the background is first constructed. More specifically, the *i*th pixel in the N background frames is represented based on the expected color value, the standard deviation, the variation of the brightness distortion α_i^*, the variation of the chromaticity distortion CD_i over N background frames, as shown in Figure 3.13.

The *i*th pixel of the current frame is classified into one of the four categories, F, B, S and H. It belongs to the foreground category (F) if the chromaticity is quite different from the expected value in the background; to the background category (B) if both of chromaticity and brightness are similar to the corresponding pixel in the background; to the shadow category (S) if chromaticity is similar but the brightness is lower than the corresponding pixel in the background; and to the highlighted background (H) if chromaticity is similar but the brightness is higher than the corresponding pixel in the background.

However, in some case with small chromaticity distortion but large brightness distortion, the pixel will be misclassified as shadow (S). In order to avoid the misclassification, we introduce an extra criterion; that is, the pixel with large difference, i.e., $d_i = \|I_i - E_i\| > \tau_d$, is classified as the foreground pixel (F). The classification procedure is shown in Equation 3.3, where $\tau_d, \tau_{CD}, \tau_{\alpha 1}, \tau_{\alpha 2}$ are predefined thresholds, and $\hat{\alpha}_i^*, \hat{CD}_i$ are the normalized brightness distortion and the normalized chromaticity distortion [34].

Equation 3.3:

$$
\begin{aligned}
&\textit{if} \quad \hat{CD}_i > \tau_{CD} \quad \textit{or} \quad d_i > \tau_d \\
&\quad \textit{category}(i) = F; \\
&\textit{else} \quad \textit{if} \quad \hat{\alpha}_i^* < \tau_{\alpha 1} \quad \textit{and} \quad \hat{\alpha}_i^* > \tau_{\alpha 2} \\
&\quad \textit{category}(i) = B; \\
&\textit{else} \quad \textit{if} \quad \hat{\alpha}_i^* < 0 \\
&\quad \textit{category}(i) = S; \\
&\textit{else} \\
&\quad \textit{category}(i) = H;
\end{aligned}
$$

In Figure 3.14, a classified frame is illustrated with the foreground pixels in white color, the background pixels in black color, the shadow pixels in red color and the highlighted background pixels in green color. Most shadows near the two feet are detected and the silhouette of the foreground object is roughly segmented. Subsequently, the morphological operations and median filter can be further applied to obtain a smooth silhouette of the foreground object.

Figure 3.14: The left image is original frame. The right image is the classified frame as four categories, F (white), B (black), S (red), H (green).

(ii) Feature Extraction

After the foreground is segmented, the skin pixel detection as defined earlier in Section 3.1.b.ii is applied on the foreground image to generate the skin image. And the Canny edge algorithm [35] is also applied on the gray level image of the segmented foreground to generate the edge image. The motion image is obtained by thresholding the absolute value of the gray-level difference between pixels in the current frame and the corresponding pixels in the previous frame. The silhouette, skin, edge and motion images are shown in Figure 3.15.

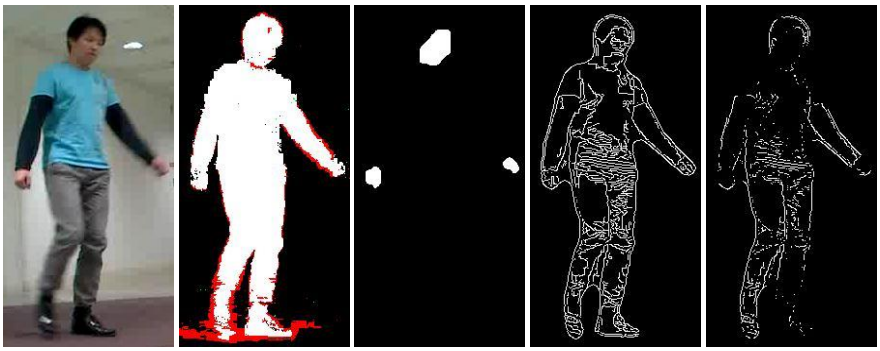


Figure 3.15: The 2D features from left to right are original image, silhouette, skin, edge and motion image.

(iii) Orientation Estimation

The 2D (angular) orientation of the moving human body is estimated based on the scale change and the trajectory of the centroid of the human body. The scale change is used to decide whether the body moves towards or away from the camera, while the trajectory change of the centroid is used to decide the horizontal movement of the body. Moreover, the trajectory and the scale changes are smoothed by a Kalman filter [36], based on the state vector $\vec{x} = \{x, y, s, \Delta x, \Delta y, \Delta s\} \in R^6$, which represents translation in x and

y coordinates, scale, velocity of x and of y, and the scale change individually, and the corresponding measurement vector $\vec{z} = \{x, y, s\} \in R^3$.

However, if the temporal orientation of the body is directly estimated by the change of the scale and the change of the centroid, it might change abruptly frame-by-frame. Therefore, in order to smooth the change of the orientation, the exponentially weighted moving average (EWMA) is applied. The temporal angular orientation θ_k^{Kalman} of the body in the k-th frame is computed by scale change and centroid movement after Kalman filter, and θ_k^{EWMA} represents the smoothed value after EWMA, as shown in Equation 3.4.

Equation 3.4: $$\theta_k^{EWMA} = \alpha\theta_k^{Kalman} + (1-\alpha)\theta_{k-1}^{EWMA}$$

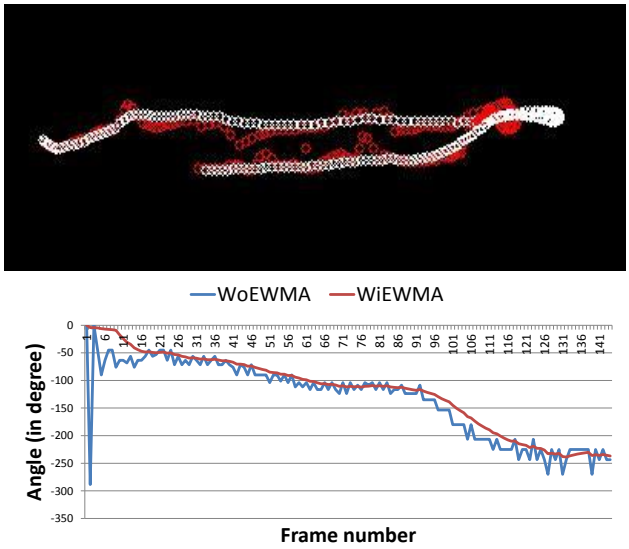


Figure 3.16: (a) The upper figure shows the trajectory of the centroid of the body along the 2D image space. The white curve is the trajectory after Kalman filter, while the red curve is the trajectory without Kalman filter. (b) The lower figure shows the orientation changes along the index of frames. The blue curve denotes orientation without EWMA, while the red curve denotes the one with EWMA smoothing.

Figure 3.16(a) shows the body trajectory of a video sequence in the 2D image space, where the white/red curves are the trajectories with/without Kalman filtering. Figure 3.16(b) shows the orientation of the body along the index of the frames, where the red/blue curves are the orientation curves with/without EWMA smoothing. The trajectory is smoothed by a Kalman filtering in Figure 3.16(a), while the change of the orientation is smoothed by EWMA in Figure 3.16(b).

3.2.c 2D Body Part Tracking

In this section, the 2D body part tracking at Stage 2 in Figure 3.12 is described in details. The 5 blobs of the body parts, Head, RightHand, LeftHand, RightFoot and LeftFoot, are tracked frame-by-frame based on shape, color and temporal information.

Compared with the color information, the shape information is more stable and resistant to the change of the illumination and the various colors of the clothes. Hence, the skeleton of the body is extracted by a skeletonization process and used to locate the body-part blobs. However, the skeleton cannot always locate the body-part blobs in any orientation of the body. For example, when hands are close to the torso and two feet are close to each other, the skeleton of the frame shows only one vertical line. Therefore, the color of the body-part blobs can assist to estimate the locations of the blobs. The mean-shift tracking algorithm [29]-[31] is thus applied to track the blobs based on the color information.

Nevertheless, when the blobs are badly occluded or hidden, neither shape nor color information can be useful to locate the blobs. Therefore, the temporal information is further considered to track the 2D blobs. In the proposed system, the Kalman filter [36] is again applied to predict the locations of the 2D blobs.

Finally, a fusion method is designed to integrate the 2D blob (Head, RightHand, LeftHand, RightFoot and LeftFoot) candidates, which come from the skeletonization scheme, the mean-shift tracking, and the Kalman filtering prediction separately, to provide the tracked locations of the 2D blobs.

(i) Skeletonization Scheme

Inspired by the work in [37], we designed an effective skeletonization scheme to extract the shape information of the object, and obtain the connected skeleton of the object with 1-pixel width.

First, the distance transform (DT) [38] is employed in the segmented silhouette to obtain the DT map. Then for a given pixel p, the thinness parameter (TP_p) [39], which is the difference of DT_p and the mean of the neighbor DT_{q_i} (MNT_p), is computed by Equation 3.5, where $\{q_i\}_{i=1\ldots8}$ are the 8-connected neighbors of the pixel p.

Equation 3.5:
$$TP_p = DT_p - MNT_p = DT_p - \frac{\sum_{i=1}^{8} DT_{q_i}}{8}$$

Moreover, we employ a thinning process [39], in which the pixel p is classified as the skeleton pixel if $TP_p > \tau_{TP}$ with the predefined threshold τ_{TP}, to generate a

disconnected skeleton (DS) map with scattered skeleton pixels. Subsequently, the morphological closing operation is applied on the DS map to enhance the property of the connection. After the closing operation, an $m \times m$ window is used on the DS map to remove the outliers, i.e., the pixel in the DS map will be removed if there is only one pixel within the window, and the DS^* map is thus generated.

In order to further obtain a connected skeleton (CS) map, the k-means algorithm [40] is employed to cluster the skeleton pixels in the DS^* map into k=12 groups. Because the performance of k-means is sensitive to the initial seeds, a set of 12 3D points, selected from our estimated 3D pose reconstruction in the previous frame, are projected into the 2D frame based on the orientation of the body, and these projected 2D points are set as the initial k-means seeds for the current frame. The locations of these 12 selected seed points in the 3D model are shown in Figure 3.17 (one for head, 3 for torso, 4 for arms/forearms and 4 for upper/lower legs).

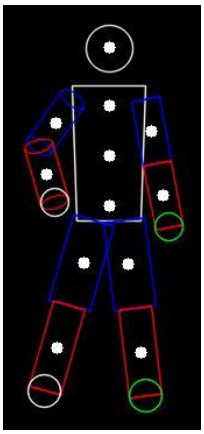


Figure 3.17: The 12 points of the 3D model with white dots in the previous frame are projected into the 2D frame, and set as the initial seeds for k-means.

After the skeleton pixels in the DS^* map are clustered into k=12 groups, we employ a minimal spanning tree (MST) algorithm on each group separately to form k undirected acyclic connected components $\{CC_i\}_{i=1...12}$ with k centroids $\{o_i\}_{i=1...12}$. To further connect these k connected components into a skeleton, MST is applied to obtain a direct acyclic graph (DAG) as described in [37]. However, if we directly use the k centroids of the k connected components as nodes, the structure of the connected skeleton might be wrong by connecting the wrong connected components (i.e., body parts). For example, the connected component of the forearm is supposed to be connected to the arm, but in certain poses, the forearm is connected to the upper leg, because the distance between the centroid of the forearm and the one of the upper leg is smaller than the distance between the centroid of the forearm and the one of the arm. In order to mitigate the impact caused by the above error, we propose a modified minimal spanning tree (MMST), on top of the original skeletonization scheme [37], where the weight of connecting two nodes $\{o_i, o_j\}$ is the shortest distance between the cluster CC_i and the cluster CC_j, as defined in Equation 3.6. In the MMST, when an edge with minimal weight is formed between o_i and o_j, the nodes o_i and o_j are not directly connected; instead, the nearest neighbors, p_i^* in CC_i and p_j^* in CC_j, as defined in Equation 3.6, are connected. The MMST algorithm can thus maintain the original segment shape of the 12 acyclic connected components and generate a single connected component graph, which is the desired connected skeleton

(CS) map. Furthermore, the candidates for 2D blobs (Head, RHand, LHand, RFoot, LFoot) are detected at the end points of the connected skeleton map.

Equation 3.6:

$$Weight(o_i, o_j) = d(p_i^*, p_j^*) \le d(p_i, p_j), \quad for \quad any \quad p_i \in CC_i, \quad any \quad p_j \in CC_j$$

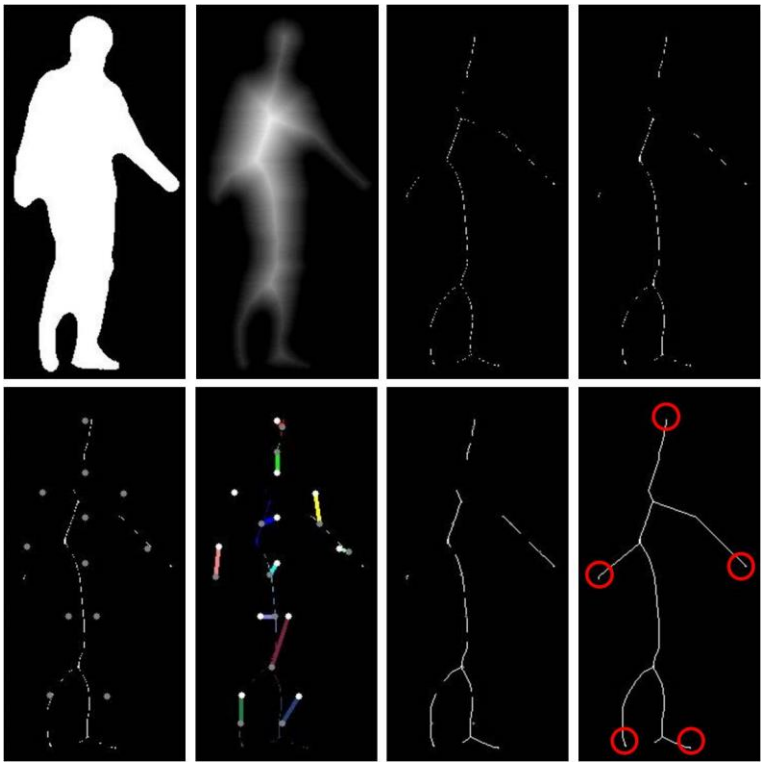


Figure 3.18: In the first row, from left to right order, the images are the silhouette, the DT map, the DS map, the DS^* map. In the second row, from left to right, the images are the DS^* map with 12 initial seeds, the DS^* map with the shift trajectories of 12 seeds, the 12 connected components in the DS^* map after MST, the CS map after MMST with the blob candidates in red circles.

The maps produced in the skeletonization procedure are shown in Figure 3.18. The images in the first row from left to right are the silhouette of the object, the distance transform (DT) map, the disconnected skeleton (DS) map (i.e., the DT map after thinning process) and the DS^* map (i.e., the DS map after morphological closing and isolation

removal). And the images in the second row from left to right are the DS^* map with 12 initial seeds for k-means, the DS^* map with the shift trajectories of 12 seeds after k-means, the 12 acyclic connected components (CC_i) after MST, and the connected skeleton (CS) map after MMST. The candidates for 2D blobs are shown in red circles in the CS map.

(ii) Mean-Shift Tracking of Body-Part Blobs

Since the skeletonization scheme cannot always locate the body-part blobs in any orientation of the body, the mean-shift tracking algorithm is thus applied to track the blobs based on the color values. In a mean-shift tracking algorithm [29]-[31], the color histogram with m bins of the target model, $\hat{q}$, of a tracked blob is built in the first frame based on Equation 3.7, as shown in Figure 3.19. Note that $\{x_i^*\}_{i=1,...,n} \in R^2$ are the normalized pixel locations of the target model in a tracked blob. The indicator function b, $R^2 \to \{1...m\}$, maps the pixel value at the location, x_i^*, into the index of its bin. δ is Kronecker delta function with $\delta[s]=1$ if $s=0$, otherwise $\delta[s]=0$. C is the normalization term, and the kernel function k is an Epanechniknov kernel, as defined in Equation 3.8.

Equation 3.7:
$$\hat{q}_u = C\sum_{i=1}^{n} k\left(\left\|x_i^*\right\|^2\right)\delta[b(x_i^*)-u], C=\frac{1}{\sum_{i=1}^{n} k\left(\left\|x_i^*\right\|^2\right)}$$

Equation 3.8:
$$k(x)=\begin{cases}1-x, & if \quad 0\le x\le 1\\ 0, & if \quad x>1\end{cases}$$

Figure 3.19: The target models of the color of 5 blobs are shown in red circles.

Moreover, in a similar way, the color histogram of the candidate model, $\hat{p}(y)$, with m bins of a tracked blob is built in the subsequent frames, as defined in Equation 3.9, where $\{x_i\}_{i=1,\ldots,n_h} \in R^2$ are the pixel locations of the target candidate with the center y in a tracked blob. The scale of the candidate model, n_h, is defined by the bandwidth *h*.

Equation 3.9:

$$\hat{p}_u(y) = C_h \sum_{i=1}^{n_h} k\left(\left\|\frac{y - x_i}{h}\right\|^2\right)\delta[b(x_i) - u], C_h = \frac{1}{\sum_{i=1}^{n_h} k\left(\left\|\frac{y - x_i}{h}\right\|^2\right)}$$

The similarity of the target model, $\hat{q}$, and the candidate model, $\hat{p}(y)$, is measured by the Bhattacharyya coefficient [41], as defined in Equation 3.10. The Bhattacharyya coefficient, $\rho[\hat{p}(y), \hat{q}]$, is maximized by moving the old center, y_{old}, of the candidate model to a new location, y_{new}. The mean shift tracking procedure [30] is applied to find the new center of the target candidate, y_{new}, as defined in Equation 3.11, where $g(x) = -k'(x)$ is the first derivative of the kernel $k(x)$, and w_i is the weight of the *i*th pixel, whose detailed derivations can be seen in [31].

Equation 3.10: $$\rho[\hat{p}(y),\hat{q}] = \sum_{u=1}^{m}\sqrt{\hat{p}_u(y)\hat{q}_u}$$

Equation 3.11: $$y_{new} = \frac{\sum_{i=1}^{n_h} x_i w_i g\left(\left\|\frac{y_{old}-x_i}{h}\right\|^2\right)}{\sum_{i=1}^{n_h} w_i g\left(\left\|\frac{y_{old}-x_i}{h}\right\|^2\right)}$$

$$where \quad w_i = \sum_{u=1}^{m}\sqrt{\frac{\hat{q}_u}{\hat{p}_u(y_{old})}}\delta[b(x_i)-u]$$

Additionally, since the color histogram of blobs might gradually change with the change of the orientation of the body, the mean shift tracking might fail to track blobs if the target models are only built from the initial frame. Therefore, we update the target models of the tracked blobs when the similarity between the target model and the corresponding candidate model is high, i.e., $\rho[\hat{p}(y),\hat{q}] > \tau_{update}$.

(iii) Kalman Filter Prediction

Even though the shape and color information are analyzed every single frame, however, in a video sequence, time continuity can be further exploited, especially when shape and color fail to track the blobs. Here, the Kalman filter [36] is again applied for tracking blobs.

In a tracked blob, the state vector $\vec{x} = \{x, y, \Delta x, \Delta y\} \in R^4$ represents the x translation, the y translation, the x velocity and the y velocity of the tracked blob, and the corresponding measurement vector is $\vec{z} = \{x, y\} \in R^2$. Both of the process and measurement covariance matrices are identity matrices. The state transition matrix A and

measurement matrix H are defined in Equation 3.12. An example of the trajectories of the 5 body part blobs are shown in Figure 3.20. Besides, the x velocity and y velocity of each blob can be further used for occlusion detection and handling.

Equation 3.12:

$$A = \begin{bmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, H = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}$$

Figure 3.20: The trajectories of the 5 body part blobs, including head, right/left hand, and right/left foot.

(iv) 2D Fusion Scheme for Body Part Tracking

In order to accurately locate the 2D body-part blobs $\{B_i\}_{i\in\{head,rhand,lhand,rfoot,lfoot\}}$, we design a fusion scheme, as shown in Figure 3.21, to integrate the shape, color and temporal information, which are exploited respectively by skeletonization, mean shift tracking and Kalman filter prediction. After the locations of the 2D blobs in the current frame are identified, the occlusion detection and handling are applied.

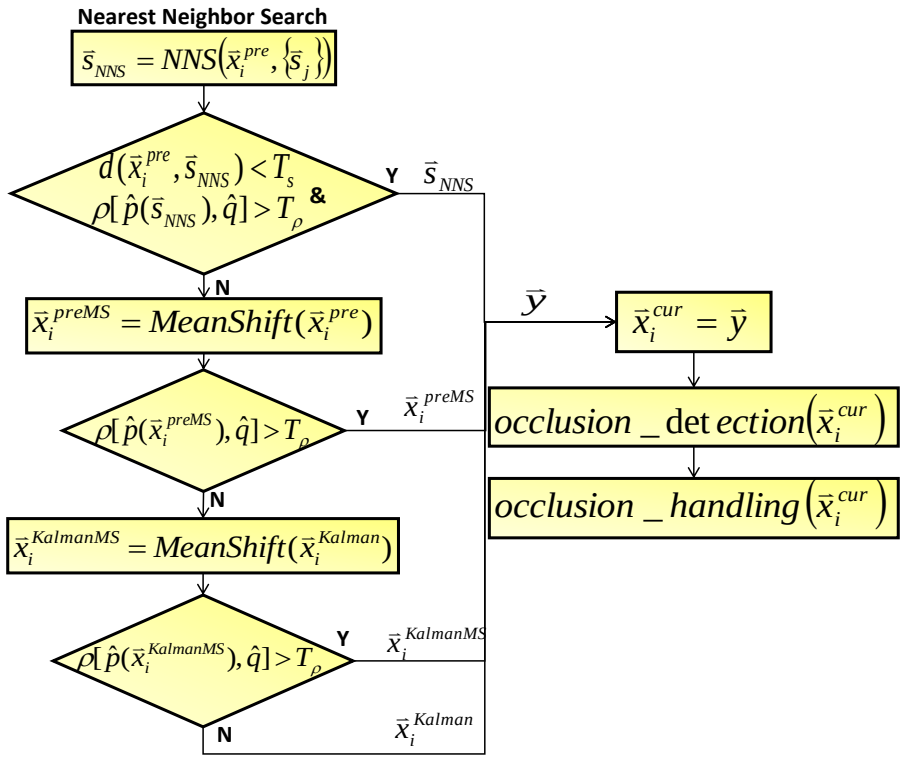


Figure 3.21: The flowchart of the fusion of shape, color and temporal information.

For a given blob B_i, the location of B_i is $\vec{x}_i^{pre} \in R^2$ in the previously tracked frame. The goal is to look for the location of B_i, $\vec{x}_i^{cur} \in R^2$, in the current frame. Let $\{\vec{s}_j\} \in R^2$ be the candidate locations for 2D blobs extracted from the end points of the skeleton map in the current frame, and let $\vec{x}_i^{Kalman}$ be the candidate location of the *i*th blob from Kalman filter prediction.

Step 1: The nearest neighbor search (NNS) is first applied on $\vec{x}_i^{pre}$ over $\{\vec{s}_j\}$ to obtain the nearest end point, $\vec{s}_{NNS}$, from $\vec{x}_i^{pre}$. If the distance $d(\vec{x}_i^{pre}, \vec{s}_{NNS})$ between $\vec{x}_i^{pre}$ and $\vec{s}_{NNS}$ is small ($< T_s$), and the color similarity $\rho[\hat{p}(\vec{s}_{NNS}), \hat{q}]$, as defined in Equation

3.10, between the target model $\hat{q}$ and the candidate model $\hat{p}(\vec{s}_{NNS})$ centered at $\vec{s}_{NNS}$ is large ($>T_\rho$), $\vec{s}_{NNS}$ is assigned to the location of the *i*th blob, i.e., $\vec{x}_i^{cur} = \vec{s}_{NNS}$.

Step 2: If the nearest neighbor candidate $\vec{s}_{NNS}$ from the skeleton map does not pass the distance and color similarity criteria with thresholds T_s, T_ρ, the mean shift tracking algorithm is applied on $\vec{x}_i^{pre}$ to generate the mean shift candidate location $\vec{x}_i^{preMS}$. If the color similarity of the target model $\hat{q}$ and the candidate model $\hat{p}(\vec{x}_i^{preMS})$ centered at $\vec{x}_i^{preMS}$ is large ($>T_\rho$), then $\vec{x}_i^{preMS}$ is assigned to the location of the *i*th blob, i.e., $\vec{x}_i^{cur} = \vec{x}_i^{preMS}$.

Step 3: If the mean shift candidate $\vec{x}_i^{preMS}$ does not pass the threshold T_ρ, the mean shift tracking algorithm is applied on the Kalman predicted candidate $\vec{x}_i^{Kalman}$ to generate the Kalman-MeanShift candidate $\vec{x}_i^{KalmanMS}$. If the color similarity of the target model $\hat{q}$ and the candidate model $\hat{p}(\vec{x}_i^{KalmanMS})$ centered at $\vec{x}_i^{KalmanMS}$ is large ($>T_\rho$), $\vec{x}_i^{KalmanMS}$ is assigned to the location of the *i*th blob, i.e., $\vec{x}_i^{cur} = \vec{x}_i^{KalmanMS}$. Otherwise, $\vec{x}_i^{Kalman}$ is assigned to the location of the *i*th blob, i.e., $\vec{x}_i^{cur} = \vec{x}_i^{Kalman}$.

After the location of *i*th blob in the current frame is determined as $\vec{x}_i^{cur}$ from either $\vec{s}_{NNS}$, $\vec{x}_i^{preMS}$, $\vec{x}_i^{KalmanMS}$ or $\vec{x}_i^{Kalman}$, the occlusion detection and handling mechanism is applied to $\vec{x}_i^{cur}$, which is described in Section 3.2.d.iii.

3.2.d 3D Pose Estimation and Occlusion Handling

In this section, the 3D pose estimation and the occlusion detection and handling at Stage 3 in Figure 3.12 are discussed in details.

The 3D human body model, the world coordinate and the camera coordinate are shown in Figure 3.22. The 3D human body model consists of head, torso and four limbs. Head is represented by a circle and torso by a rectangular cuboid, and each limb is represented by two cylinders for upper limb and lower limb. There are 25 degrees of freedom (DOFs) for poses (global translation, rotation and joint angles) and 15 DOFs for shapes (length and width of each body part). In the world coordinate, the depth is represented in the y-direction. The camera faces the y-direction in the world coordinate system.

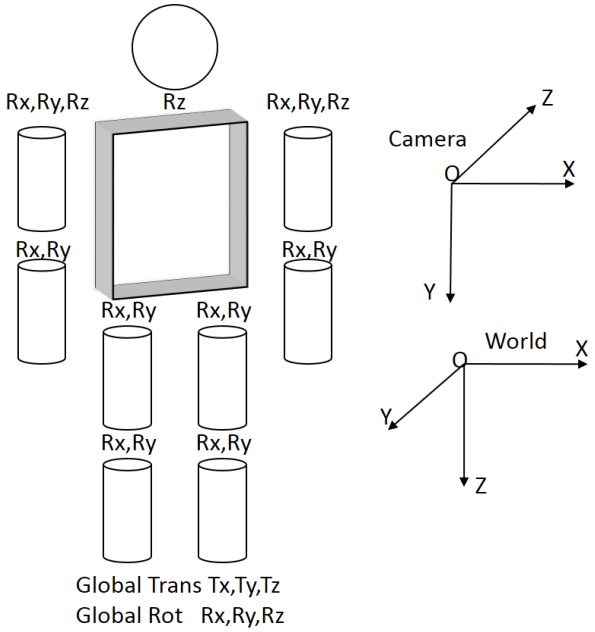


Figure 3.22: The 3D human body model and the world and camera coordinate.

(i) 3D Coarse Pose Estimation

After 2D body part tracking, the locations of the 2D body part blobs are estimated, and the centroid $\bar{o}_{body} \in R^2$ of the 2D body is obtained. Besides, the global rotation (θ_{Gz}) of the 3D model along the Z coordinate in the 3D world is estimated by the scale of the 2D body and the movement orientation from trajectory of the Kalman filter measurement, as described in Section 3.2.b.iii. Moreover, human kinematics is applied to reduce some DOFs. For example, the proportion of the shape of the 3D body parts is fixed to reduce the 15 DOFs for the shapes as 1 DOF, i.e., the scale of the whole body. In addition, four limbs are estimated independently.

Without loss of generality, take the right arm as an example. The right arm with 4 DOFs, θ_{Ux}, θ_{Uy}, θ_{Lx} and θ_{Ly}, is shown in Figure 3.23. θ_{Ux}/θ_{Uy} is the rotation angle of the arm with respect to the X/Y coordinate, while θ_{Lx}/θ_{Ly} is the rotation angle of the forearm with respect to the X/Y coordinate. l_{UA}/l_{LA} is the length of the arm/forearm, and (x_r, y_r, z_r) is the position of the right hand in 3D space, where (x_r, y_r) can be obtained by the 2D body parts tracking for the right hand blob, while z_r (depth) is unknown.

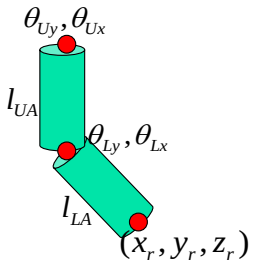


Figure 3.23: The configuration of the right arm.

Moreover, the default location of the right hand in the 3D model is $(0,0,l_{LA})$. After a series of scaling, translation, rotation matrix operations for the right hand, it will reach (x_r, y_r, z_r), as shown in Equation 3.13, where S_c is the scale matrix, W_{2c} is the world-to-camera transformation matrix, T_{Gi}/R_{Gi} is the translation/rotation matrix with respect to the q coordinate ($q \in \{X, Y, Z\}$), R_{Ux}/R_{Uy} is the rotation matrix by θ_{Ux}/θ_{Uy} with respect to the X/Y coordinate, R_{Lx}/R_{Ly} is the rotation matrix by θ_{Lx}/θ_{Ly} with respect to the X/Y coordinate, T_{Lz} is the translation matrix moving the origin from the elbow to the shoulder, and T_{Ux} and T_{Uz} are the translation matrices moving the origin from shoulder to the center of the torso. Some matrices are defined in Equation 3.14.

Equation 3.13:

$$S_C W_{2C} T_{Gz} T_{Gy} T_{Gx} R_{Gz} R_{Gx} R_{Gy} T_{Uz} T_{Ux} R_{Ux} R_{Uy} T_{Lz} R_{Lx} R_{Ly} \begin{bmatrix} 0 \\ 0 \\ l_{LA} \\ 1 \end{bmatrix} = \begin{bmatrix} x_r \\ y_r \\ z_r \\ 1 \end{bmatrix}$$

Equation 3.14:

$$S_C = \begin{bmatrix} scale & 0 & 0 & cx \\ 0 & scale & 0 & cy \\ 0 & 0 & scale & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad W_{2C} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix},$$

$$T_{Ux} = \begin{bmatrix} 1 & 0 & 0 & -\frac{l_s}{2} \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad T_{Lz} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & l_{UA} \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad T_{Uz} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & -\frac{l_T - W_{UA}}{2} \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

where l_S is the length of the shoulder, l_T is the height of the torso, W_{UA} is the width of the arm, and (cx, cy) is the centroid of the 2D body.

In order to obtain coarse 3D pose estimation, we set θ_{Lx}, θ_{Ly}, z_r to be the same values in the previous frame. Therefore, Equation 3.13 can be rewritten as Equation 3.15. Based on direct inverse kinematics, the 2 DOFs, θ_{Ux} and θ_{Uy} can be obtained by Equation 3.16.

Equation 3.15:

$$R_{Ux}R_{Uy}T_{Lz}R_{Lx}R_{Ly}\begin{bmatrix}0\\0\\l_{LA}\\1\end{bmatrix}=T_{Ux}^{-1}T_{Uz}^{-1}R_{Gz}^{-1}W_{2C}^{-1}S_C^{-1}\begin{bmatrix}x\\y\\z\\1\end{bmatrix}=T_{Ux}^{-1}T_{Uz}^{-1}R_{Gz}^{-1}\begin{bmatrix}\frac{x-cx}{scale}\\\frac{-z}{scale}\\\frac{y-cy}{scale}\\1\end{bmatrix}=T_{Ux}^{-1}T_{Uz}^{-1}R_{Gz}^{-1}\begin{bmatrix}\Delta x\\\Delta z\\\Delta y\\1\end{bmatrix}$$

Equation 3.16:

$$\begin{bmatrix}\cos\theta_{Uy} & 0 & \sin\theta_{Uy} & 0\\ \sin\theta_{Ux}\sin\theta_{Uy} & \cos\theta_{Ux} & -\sin\theta_{Ux}\cos\theta_{Uy} & 0\\ -\cos\theta_{Ux}\sin\theta_{Uy} & \sin\theta_{Ux} & \cos\theta_{Ux}\cos\theta_{Uy} & 0\\ 0 & 0 & 0 & 1\end{bmatrix}\begin{bmatrix}l_{LA}\sin\theta_{Ly}\\ -l_{LA}\sin\theta_{Lx}\cos\theta_{Ly}\\ l_{LA}\cos\theta_{Lx}\cos\theta_{Ly}+l_{UA}\\1\end{bmatrix}=\begin{bmatrix}\Delta x\cos\theta_{Gz}+\Delta z\sin\theta_{Gz}+\frac{l_s}{2}\\ -\Delta x\sin\theta_{Gz}+\Delta z\cos\theta_{Gz}\\ \Delta y+\frac{l_T-W_{UA}}{2}\\1\end{bmatrix},$$

$$\theta_{Uy}=\sin^{-1}\left(\frac{m_1v_3\pm|v_1|\sqrt{v_1^2+v_3^2-m_1^2}}{v_1^2+v_3^2}\right),\ \theta_{Ux}=\sin^{-1}\left(\frac{m_3v_2\pm|P|\sqrt{P^2+v_2^2-m_3^2}}{P^2+v_2^2}\right)$$

where $P=v_3\cos\theta_{Uy}-v_1\sin\theta_{Uy}$, $\vec{v}=\begin{bmatrix}l_{LA}\sin\theta_{Ly}\\ -l_{LA}\sin\theta_{Lx}\cos\theta_{Ly}\\ l_{LA}\cos\theta_{Lx}\cos\theta_{Ly}+l_{UA}\\1\end{bmatrix}=\begin{bmatrix}v_1\\v_2\\v_3\\1\end{bmatrix}$, $\vec{m}=\begin{bmatrix}\Delta x\cos\theta_{Gz}+\Delta z\sin\theta_{Gz}+\frac{l_s}{2}\\ -\Delta x\sin\theta_{Gz}+\Delta z\cos\theta_{Gz}\\ \Delta y+\frac{l_T-W_{UA}}{2}\\1\end{bmatrix}=\begin{bmatrix}m_1\\m_2\\m_3\\1\end{bmatrix}$

(ii) 3D Fine Pose Estimation

The ambiguities of the 3D poses, such as the occlusions of the body parts, different viewing perspectives and change of the illumination, greatly increase the difficulty of 3D pose estimation. Moreover, searching the optimized 3D model in high dimensions (40 dimensions in our 3D model) can take lots of computation as well.

Therefore, the 2D body part blob tracking and the coarse 3D pose estimation are applied in every frame to generate rough 3D model poses, so as to greatly decrease the 3D model search space, and to reduce the probability to be stuck on incorrect local minima while searching for the best 3D pose. Now, the best 3D model poses can be achieved by searching the local optimization, instead of searching a global optimization. Also considering the time performance, an efficient local optimization algorithm, the downhill simplex [25], is employed on searching 3D model poses by minimizing the cost function, which measured the difference between 2D features and 3D model projections as defined in Equation 3.2. The process is same as the Section 3.1.c.ii. After fine estimation, the best 3D poses are reconstructed.

(iii) Occlusion Detection and Handling

When viewed from some angular perspectives, the occlusions between hands, between feet or between hands and torso can easily happen. Therefore, a mechanism to detect and resolve the occlusions is critically needed.

Case 1: *Hand-hand (HH) occlusion or foot-foot (FF) occlusion*

Usually, the color of the left hand (foot) and the right hand (foot) are very similar; therefore, after the occluded hands (feet) separate, incorrect associations sometimes happen.

In a tracked frame, the case of HH occlusion is detected if the distance between the right hand blob and the left hand blob is small, as described in Equation 3.17. In a

similar way, the case of FF occlusion is detected if the distance between the right foot blob and the left foot blob is small.

Equation 3.17:

$$flag(B_{rhand}) = flag(B_{lhand}) = \begin{cases} HHOCCLUSION & ,if \quad d(B_{rhand}, B_{lhand}) < \tau_{HH} \\ ACTIVE & ,otherwise \end{cases}$$

$$flag(B_{rfoot}) = flag(B_{lfoot}) = \begin{cases} FFOCCLUSION & ,if \quad d(B_{rfoot}, B_{lfoot}) < \tau_{FF} \\ ACTIVE & ,otherwise \end{cases}$$

Equation 3.18:

$$score1 = \vec{v}_{rfoot}^{vel} \cdot \vec{v}_{rfoot}^{cand} + \vec{v}_{lfoot}^{vel} \cdot \vec{v}_{lfoot}^{cand}$$

$$score2 = \vec{v}_{rfoot}^{vel} \cdot \vec{v}_{lfoot}^{cand} + \vec{v}_{lfoot}^{vel} \cdot \vec{v}_{rfoot}^{cand}$$

$$if \quad score1 > score2 - \rho$$

$$\vec{x}_{rfoot}^{cur} = \vec{x}_{rfoot}^{pre}; \quad \vec{x}_{lfoot}^{cur} = \vec{x}_{lfoot}^{pre};$$

$$else$$

$$\vec{x}_{rfoot}^{cur} = \vec{x}_{lfoot}^{pre}; \quad \vec{x}_{lfoot}^{cur} = \vec{x}_{rfoot}^{pre};$$

For the FF occlusion case, when the foot is occluded in the previous frame, but changes into non-occluded in the current frame, it means the occluded feet start to be separated. The right foot velocity vector $\vec{v}_{rfoot}^{vel} \in R^2$ and the left foot velocity vector $\vec{v}_{lfoot}^{vel} \in R^2$, obtained from Kalman filter prediction for each blob, are then considered here as the movement directions of the feet to resolve the ambiguity. Let the locations of the right and left foot in the previous frame be $\vec{x}_{rfoot}^{pre}, \vec{x}_{lfoot}^{pre} \in R^2$ respectively. And let the candidates locations of the right and left foot in the current frame be $\vec{x}_{rfoot}^{cand}, \vec{x}_{lfoot}^{cand} \in R^2$ respectively. Then candidate vectors for the right foot and left foot are $\vec{v}_{rfoot}^{cand} = \vec{x}_{rfoot}^{cand} - \vec{x}_{rfoot}^{pre}$ and $\vec{v}_{lfoot}^{cand} = \vec{x}_{lfoot}^{cand} - \vec{x}_{lfoot}^{pre}$. If the direction of the candidate vectors and one of the velocity vectors are not consistent, the labeling of the feet will be exchanged. The FF occlusion

procedure is described in Equation 3.18, where ρ is a small constant to avoid the case of small difference between *score1* and *score2*.

An example of occlusion handling is illustrated in Figure 3.24. The white circles represent the right hand and the right foot, while the green circles denote the left hand and the left foot. In the left image (the previous frame), the flags for the right foot and left foot are FFOCCLUSION. In the right image (the current frame), the flags for the right foot and left foot are changed into ACTIVE. The *score1* and *score2* are calculated based on the velocity vectors and the candidate vectors in Equation 3.18. The velocity vector of the right foot is towards the right side of the image; therefore, the blob candidate on the right side is supposed to be the right foot blob. Consequently, the candidate locations of the right foot and left foot are exchanged in the right image (the current frame).

Figure 3.24: A foot-foot occlusion handling example. The white circles are the right hand and right foot, while the green circles are the left hand and left foot. The feet here denotes the ankles.

Case 2: *Hand-body (HB) occlusion*

In some viewing perspectives, hands are occluded by the body (torso). In this case, both the skeleton and the color tracking fail to locate the hands. If the hands are occluded by body over many frames, the time continuity will fail to locate the hands as well. Therefore, the occluded hands are assumed to be located around the corresponding hips.

The HB occlusion is detected based on the information of the depths of the hands $z_{rhand}^{pre3D}, z_{lhand}^{pre3D} \in R$, the depth of the centroid of the 3D model $z_{cbody}^{pre3D} \in R$ and the velocities of the hands $\vec{v}_{rhand}^{vel}, \vec{v}_{lhand}^{vel} \in R^2$, obtained from Kalman filter prediction for each blob. Take the left hand as an example, and assume the locations of the left hand blob and the 2D whole body centroid in the previous frame to be $\vec{x}_{lhand}^{pre}, \vec{o}_{body}^{pre} \in R^2$. The left-hand-body vector is $\vec{u}_{lhb}^{pre} = \vec{o}_{body}^{pre} - \vec{x}_{lhand}^{pre}$, and the distance between the left hand and the centroid of the whole body is $d_{lhb}^{pre} = \left\| \vec{o}_{body}^{pre} - \vec{x}_{lhand}^{pre} \right\|$. The HB occlusion is detected if the depth of the left hand is behind the depth of the body $\left(z_{lhand}^{pre3D} < z_{cbody}^{pre3D} \right)$, the left hand is close to the body $\left(d_{lhb}^{pre} < \tau_{HB} \right)$, and the direction of the left hand is pointing towards the body $\left(\vec{u}_{lhb}^{pre} \cdot \vec{v}_{lhand}^{vel} \geq 0 \right)$. If the HB occlusion has already been detected in the previous frame, the direction of the left hand is unnecessary towards the body. Once the HB occlusion is detected, the hand blob candidates from the skeletonization scheme and from the mean shift tracking are out of the consideration, because neither shape nor color can provide the information of the location of the hand; that is, only the hand blob candidate

from the Kalman filter is considered. The HB occlusion detection procedure is described in Equation 3.19.

Equation 3.19:
$$\begin{aligned} &if \quad z_{lhand}^{pre3D} < z_{cbody}^{pre3D} \quad and \quad d_{lhb}^{pre} < \tau_{HB} \\ &\quad if \quad \vec{u}_{lhb}^{pre} \cdot \vec{v}_{lhand}^{vel} \geq 0 \quad or \quad flag\left(B_{lhand}^{pre}\right) == HBOCCLUSION \\ &\quad\quad flag\left(B_{lhand}^{cur}\right) == HBOCCLUSION; \end{aligned}$$

3.2.e Experimental Results

The proposed system is tested on several video sequences on the public well-known dataset including HumanEva [42] and IXMAS [59] and the self-recorded dataset. For performance evaluation, we choose walking video sequences on HumanEva, which allow us to test whether the orientation of the object can be well measured when the object gradually turns around, whether the 2D body parts can be well tracked when occlusions happen, and also whether the 3D poses can be well estimated and reconstructed. Moreover, the proposed method is also compared with other 3 state-of-the-art 3D pose estimation methods through monocular video sequences on HumanEva II benchmark. Furthermore, computation efficiency is provided.

In order to evaluate the performance of our proposed method, the 13 3D joints, including Head, RHand, REblow, RShoulder, LHand, LEblow, LShoulder, RFoot, RKnee, RHip, LHip, LKnee and LFoot, are estimated frame-by-frame. Figure 3.25 shows the skeleton and the location of 13 joints on the left side, and the corresponding 3D model on the right side.

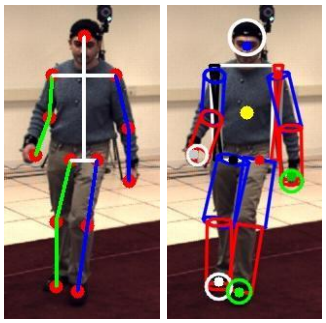


Figure 3.25: The left figure shows the skeleton and the locations of the 13 joints. And the right figure shows the corresponding 3D model.

(i) Accuracy of Estimated 3D Human Poses

In order to measure the accuracy of the estimated 3D human poses by the proposed system, the two public well-known datasets for human actions, HumanEva [42] and IXMAS [59], and the self-recorded dataset are applied to our proposed system for performance evaluation in terms of the estimation accuracy.

Case 1: *HumanEva dataset*

The widely used HumanEva [42] dataset is first considered because the dataset provides the 3D human poses by a commercial motion capture system (MoCap), Vicon Peak [42]. The estimated 3D human poses by MoCap are considered as ground truths. In the monocular video sequence from Subject 1 of Camera 3(S1/C3), 503 frames of S1/C3 with frame size 640x480 are analyzed to model 3D human body. For the qualitative evaluation, the snapshots are shown in Figure 3.26, where the right hand and right foot are denoted in white circles, left hand and left foot in green circles, the right shoulder and right hip in black dots, the left shoulder and left hip in red dots, the right side of the torso in black cross, and left side of the torso in red crosses.

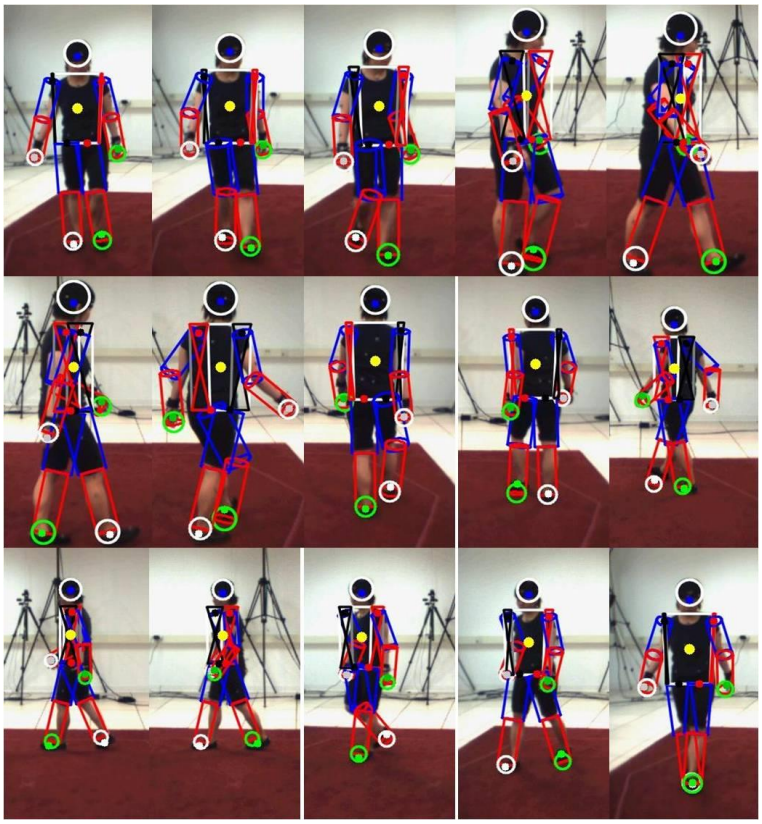


Figure 3.26: The snapshots of the 3D human model with the estimated poses.

Moreover, the frame-by-frame comparisons for the S1/C3 HumanEva walking video sequence with the ground truths and with the proposed estimated system of left shoulder and left hip in X, Y, Z coordinate are provided in Figure 3.27. The consistence of the ground-truth curve in blue line and the estimated curve in red line further shows the stable and favorable performance of the proposed system.

Table 3.2: The Mean Error for X, Y, Z Coordinates (in pixels)

	Mean_X	Mean_Y	Mean_Z
Average of 13 joints	7.95	4.78	8.58

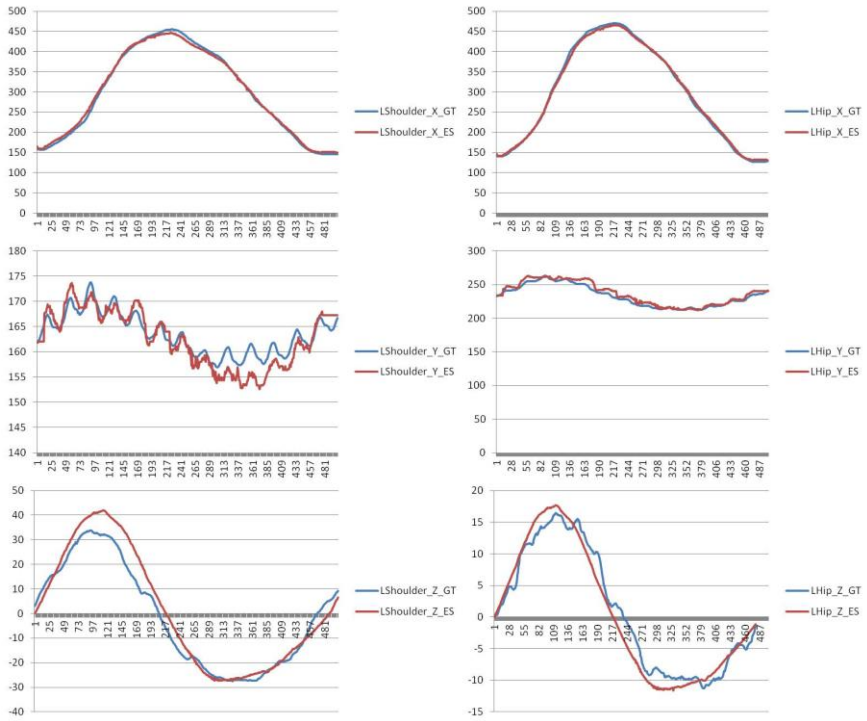


Figure 3.27: Frame-by-frame comparison of left shoulder (left column) and left hip (right column) in X (1st row), Y (2nd row), Z (3rd row) coordinate. The blue curve is the ground-truth motion and the red curve is the estimated motion.

Furthermore, for the quantitative evaluation, the average of mean errors of 13 3D joints (as defined in Figure 3.25) in X, Y, Z coordinates are shown in Table 3.2. The mean error (in pixels) is the mean of the absolute difference between the estimated poses and the ground-truth over frames in X, Y, Z coordinate. Compared to the frame size 640x480 pixels, the mean errors in Table 3.2 are small and it shows the good performance of the proposed system.

Case 2: *IXMAS dataset*

The IXMAS [59] dataset is another well-known human actions dataset, where 12 actors perform 13 types of actions, including “nothing”, “check_watch”, “cross_arm”,

“scratch_head”, “sit_down”, “get_up”, “turn_around”, “walk”, “wave”, “punch”, “kick”, “point” and “pick_up”. The IXMAS [59] dataset also provides 5 different view angles with camera0 ~ camera4 and frame size, 390x291. The camera3 is applied in our experiments.

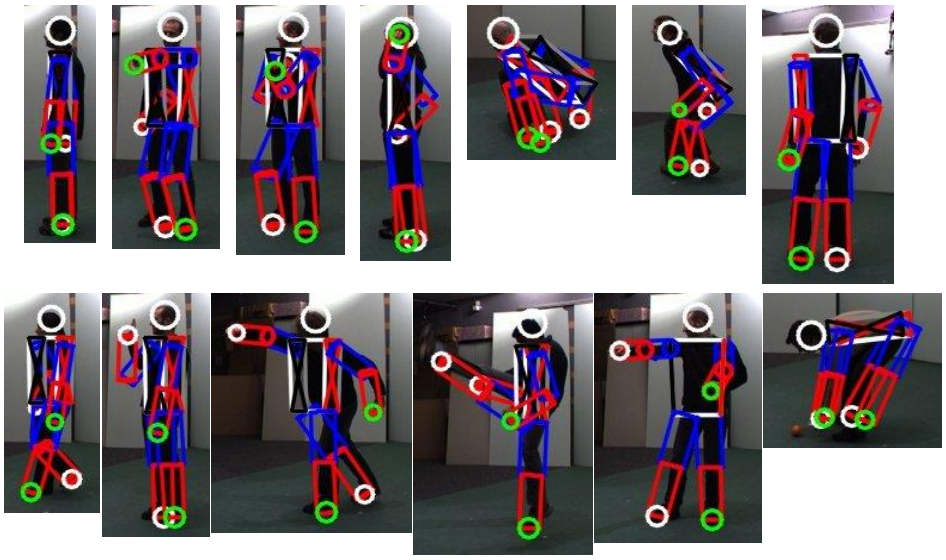


Figure 3.28: The snapshots of the 3D pose estimation results of the 13 types of actions in IXMAS for camera3 in different actors. The upper row is “nothing”, “check_watch”, “cross_arm”, “scratch_head”, “sit_down”, “get_up” and “turn_around”. The lower row is “walk”, “wave”, “punch”, “kick”, “point” and “pick_up”.

For the qualitative evaluation, 260 videos (13 types of actions x 12 actors) with frame size, 390x291, with camera3 in IXMAS dataset are applied to our proposed system to estimate 3D human poses. The snapshots of the 3D pose estimation results of the 13 types of actions for different actors are shown in Figure 3.28. Moreover, for the quantitative evaluation, the mean errors (the average of absolute difference between the estimated 3D joints and the ground-truth 3D joints in L2-Norm in pixel) for 13 3D joints as defined earlier are shown in Figure 3.29. The ground-truth 3D joints are provided with

manual label. The upper figure in Figure 3.29 shows the average of 3D mean errors of 13 3D joints for 13 types of actions. The 3D mean errors range from 3 pixels to 7 pixels for different types of actions. The mean errors do not vary largely in terms of different types of actions. On the other hand, the lower figure in Figure 3.29 shows the average of 3D mean errors of 13 types of actions for 13 3D joints. The 3D mean errors range from 2 pixels to 10 pixels for different 3D joints. The mean errors are small at head and the joints nearby the torso, while those are large at elbows, knees and the end points of limbs.

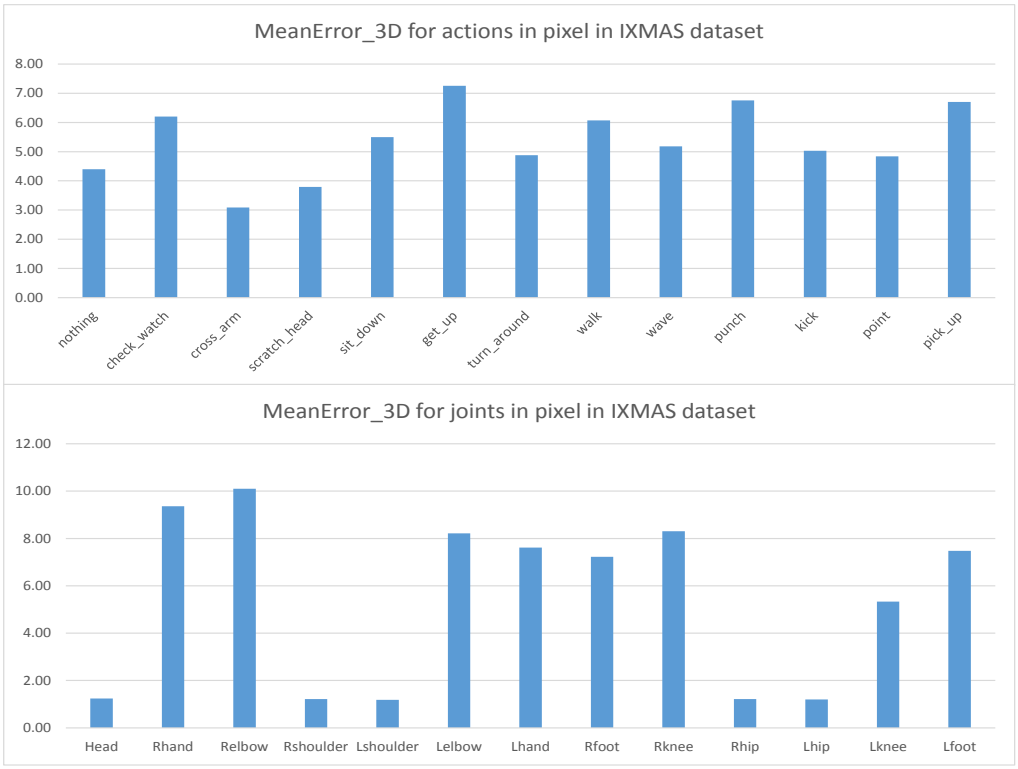


Figure 3.29: The upper figure shows the 3D mean error in pixel for 13 types of actions. The lower figure shows the 3D mean error in pixel for 13 joints.

Furthermore, the comparison of 2D and 3D mean errors between HumanEva [42] dataset and IXMAS [59] dataset is shown in Table 3.3. Mean_3D denotes the average of

mean errors of the 13 3D joints in both datasets, while Mean_2D only considers the X and Y coordinates. The Normalized Mean_2D and Normalized Mean_3D are defined as in Equation 3.20. The performance of IXMAS dataset is slightly better than that of HumanEva dataset because some actions in IXMAS are more steady such as "nothing". Both dataset shows good performance in our proposed system.

Equation 3.20:
$$Normalized \quad Mean\_2D = Mean\_2D / \sqrt{Width^2 + Height^2}$$
$$Normalized \quad Mean\_3D = Mean\_3D / \sqrt{Width^2 + Height^2}$$

Table 3.3: Comparison of Mean Errors on HumanEva and IXMAS

	Mean_2D	Mean_3D	Frame Width	Frame Height	Normalized Mean_2D	Normalized Mean_3D
HumanEva [42]	9.28	12.64	656	490	0.0113	0.0154
IXMAS [59]	4.55	5.36	390	291	0.0094	0.0110

Case 3: *Self-recorded dataset*

We also tested the proposed system on the self-recorded real-world video sequences. In these video sequences, the conditions are not so well-controlled, such as the change of the illumination, the unstable aperture and focus of the camera and the shadow effect. The segmentation with shadow removal, the Kalman filter prediction and the occlusion detection and handling are employed to overcome the issues. Two monocular self-recorded videos are experimented, including the walking sequence (SR1) and the parking-lot sequence (SR2). Moreover, in order to experiment on different types of actions by different actors, 4 different types of actions, including waving, throwing,

boxing and kicking, with 4 actors for self-recorded videos (SR3) are also applied to our proposed system.

The experiment on SR1 includes 145 frames with frame size 640x480. The 2D mean errors of 5 body parts (Head, RHand, LHand, RFoot, LFoot) are shown in Table 3.4, and the average of the 2D mean error is 7.44 pixels. The snapshots of the 3D human pose estimation results for SR1 are shown in Figure 3.30. The upper-row figures in Figure 3.30 show the skeleton results. The 13 joints locations are denoted in red circle and the orientation in the cyan arrow. The right arm and right leg are shown by green lines, while the left arm and left leg are shown by blue lines. The yellow circle is the centroid of the body. Besides, the lower-row figures in Figure 3.30 show the corresponding reconstructed 3D model with right hand and right foot in white circle, left hand and left foot in green circle, the right shoulder and right hip in black dot, the left shoulder and left hip in red dot, the right side of the torso in black cross, and left side of the torso in red cross. Besides, the cyan arrow denotes the moving direction of the object. The horizontal direction denotes the movement in the right/left direction, while the vertical direction denotes the movement towards/away from the camera. The Table 3.4 and Figure 3.30 show the subject in SR1 is well tracked.

Table 3.4: 2D Mean Error on SR1 Video (in pixels)

	Head	RHand	LHand	RFoot	LFoot
Mean (std)	4.92 (3.99)	5.59 (4.94)	7.15 (5.05)	7.93 (5.25)	11.63 (5.69)

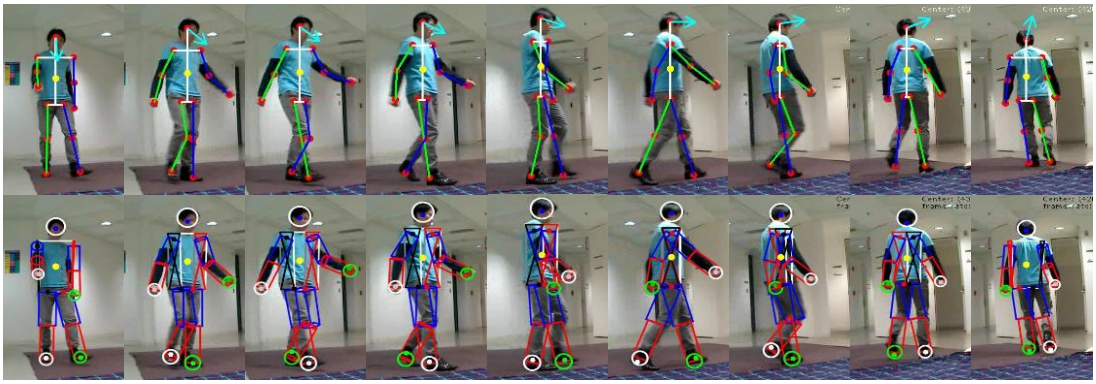


Figure 3.30: The snapshots of the 3D pose estimation results on SR1. The upper-row figures show the skeleton results, while the lower-row figures show the corresponding reconstructed 3D model.

In the SR2 video, 500 frames are analyzed with frame size 640x480. The snapshots of the 3D human pose estimation results are shown in Figure 3.31. Although the illumination condition is even worse in the parking lot, the person in SR2 is still well tracked and 3D human poses are well estimated. The SR2 is also to demonstrate the application to surveillance cameras in our proposed system.

Figure 3.31: The snapshots of the 3D pose estimation results on SR2.

In SR3 experiment, 16 self-recorded videos (4 types of actions x 4 actors) are applied to our proposed system. The snapshots of the 3D pose estimation results on 4 types of actions by four different persons are shown in Figure 3.32. It shows our proposed method can be applied not only to walking video sequences, but also to various types of action sequences.

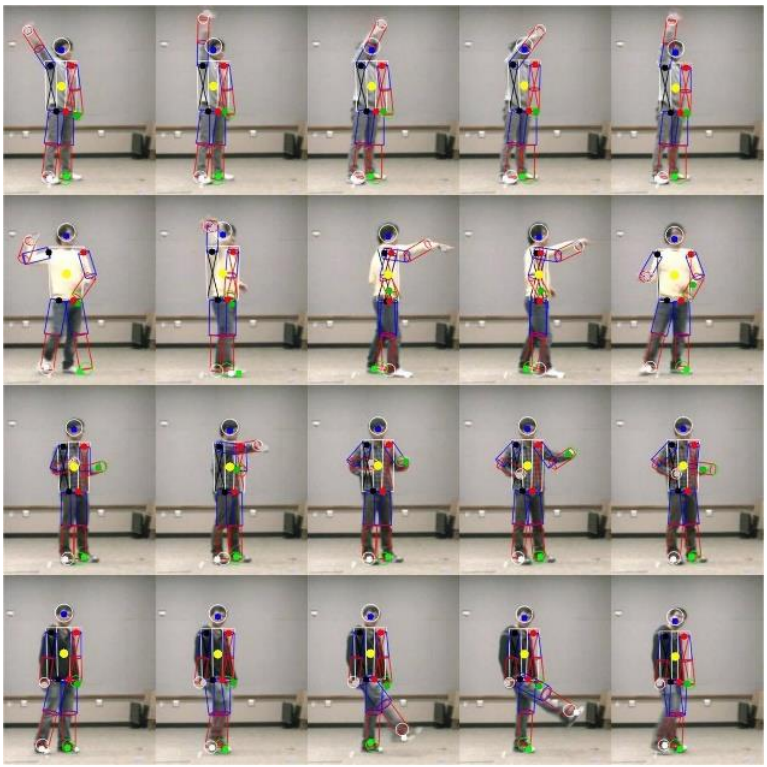


Figure 3.32: The snapshots of the 3D pose estimation results on SR3 videos.
From top row to the bottom row, they are waving, throwing, boxing and kicking by four actors.

(ii) Comparison with Other Methods

In this subsection, the experimental results are compared with other state-of-the-art methods [60], [61], [11]. In [60], Rogez et al. use spatio-temporal 2D-models to fit shape-skeleton features. In [61], Rogez et al. apply random forests classifiers on HOG

features. In [11], Andriluka et al. apply SVM classifiers on appearance models. However, the color information is not considered in [60] and [61], and the shape information is not considered in [11]. In our proposed method, we jointly take into account the appearance (color), shape (skeleton) and temporal (time continuity) information for 3D pose estimation. The resulting 3D skeletons with 3D joints for [60], [61], [11], and our proposed system are shown in Figure 3.33. The definition of the 13 joints (Head, R/LHand, R/LElbow, R/LShoulder, R/LFoot, R/LKnee, R/LHip) are the same, except the additional two joints (Neck and Center of Hip) used in [61] and [11]. Therefore, the locations of the 13 joints are comparable.

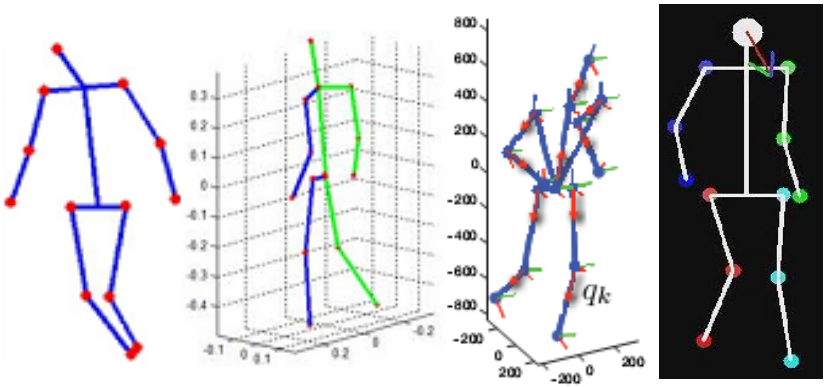


Figure 3.33: The 3D skeletons with 3D joints individually for [60], [61], [11], and our proposed system from left to right.

The quantitative performance comparison is shown in Table 3.5. Two monocular video sequences in HumanEva II dataset with frame size 656x490 are experimented on [60], [61], [11] and our proposed method, including the videos on Subject 2 of Camera 1 (S2/C1) and Subject 2 of Camera2 (S2/C2), where 350 frames for each video are analyzed and estimated the 3D human poses. The 13 joints of estimated 3D coordinates

are projected onto image plane, and the 2D mean error (Mean) and standard deviation (Std) of the average of the 13 joints are used for evaluation in Table 3.5. As shown in Table 3.5, our proposed method achieves better performance than [60], [61], [11]. Three main reasons for our better performance are discussed below. First, we consider more information including appearance, shape and time, and design a scheme to effectively fuse them. Second, neither of [60], [61], [11] directly use a 3D model, which is used in the analysis via synthesis framework as we did. In our proposed system, applying a 3D model is easier to handle occlusion and incorporate kinematic constraints into the system. Thirdly, in [60], [61], [11], due to the lack of the 3D model, a probabilistic model is needed for training. The size of the training dataset may also influence the performance. However, in our proposed system, we can achieve better performance without the training phase.

Table 3.5: Comparison of 2D Mean Errors on HumanEvaII (in pixels)

Subject/Camera	Mean (Std) [60]	Mean (Std) [61]	Mean (Std) [11]	Mean (Std) Proposed
S2/C1	16.96 (4.83)	12.98 (3.5)	10.49 (2.7)	**10.43** (2.68)
S2/C2	18.53 (5.97)	14.18 (4.38)	10.72 (2.44)	**9.82** (2.18)

Moreover, for the qualitative evaluation, the snapshots of the 3D human pose estimation results of S2/C1 and S2/C2 are shown in Figure 3.34 and Figure 3.35 respectively. The arrows, lines circles and colors in Figure 3.34 and Figure 3.35 have the same definition as the ones in Figure 3.30. The snapshots show the subject is well tracked

and the poses are well estimated during the change of the orientation of the body, the change of the body scale, the change of the poses in body parts and the self-occlusions.

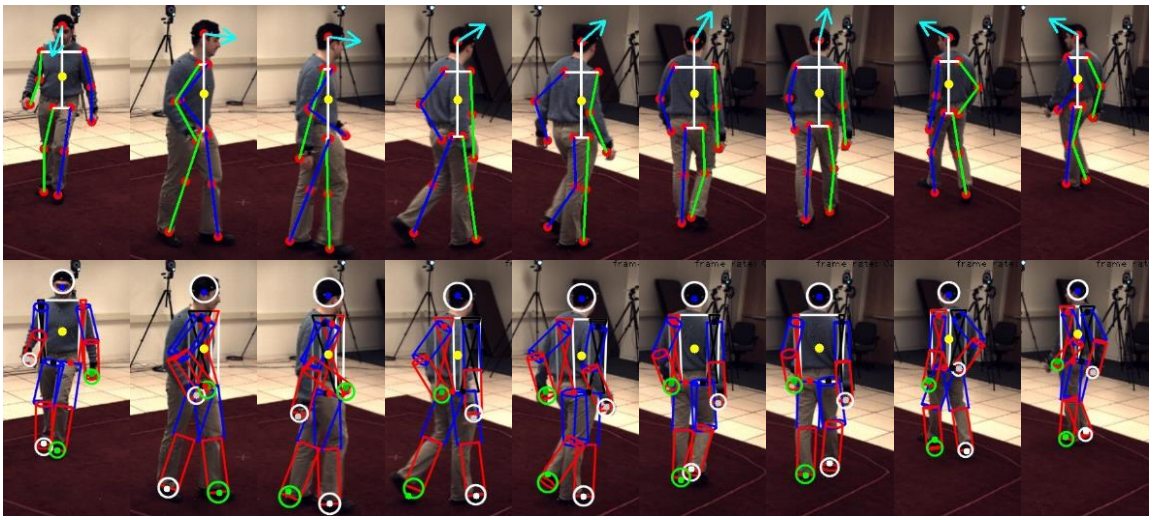


Figure 3.34: The snapshots of the 3D pose estimation results on HumanEva II S2/C1. The upper-row figures show the skeleton results, while the lower-row figures show the corresponding reconstructed 3D model.

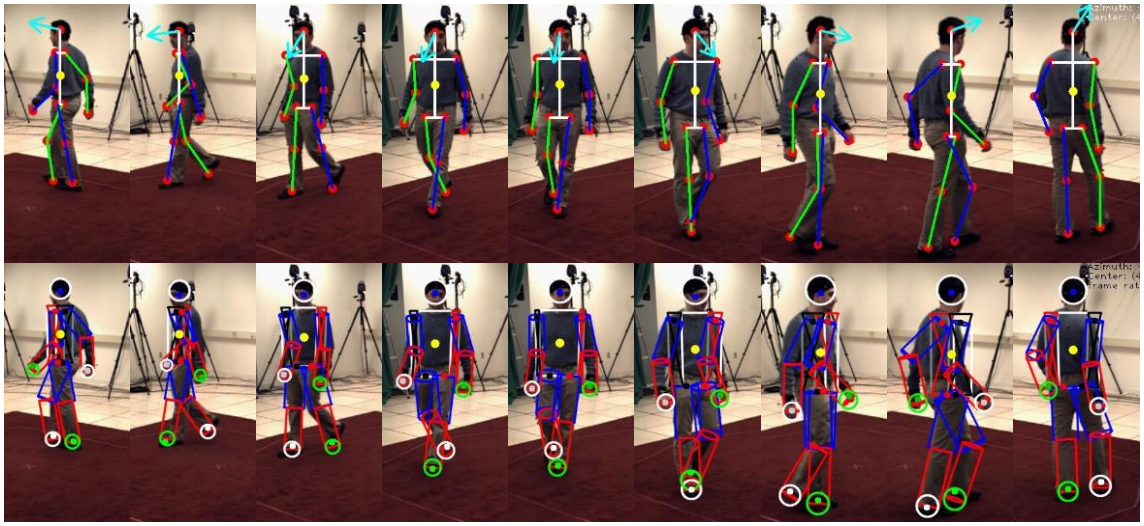


Figure 3.35: The snapshots of the 3D pose estimation results on HumanEva II S2/C2. The upper-row figures show the skeleton results, while the lower-row figures show the corresponding reconstructed 3D model.

(iii) Computation Evaluation

For computation evaluation, the average seconds per frame of S1/C3, S2/C1, S2/C2, SR1 and SR2 video sequences are shown in Table 3.6. The method in [12] takes about 2-3 minutes for the RVM and about 20 minutes for shape context extraction and clustering. The method using multilevel structured models in [17] takes an average of 5 minutes per frame. Compared with [12] and [17], our proposed method only takes less than 2 seconds for each frame, which is a significant improvement in time. Moreover, compared with [60], [61] and [11], our proposed method reduces the burdensome training process, which might take several hours to several days to learn extensive dataset. In addition, our proposed system is implemented in unoptimized C++ codes and runs on a laptop (CPU Intel Core 2 Duo T8100 2.1 GHz, RAM 3 GB, Windows 7). We believe the proposed system can be implemented as a real-time application with a careful code optimization and the use of graphic hardware such as OpenMP and GPGPU (General-Purpose computation on Graphics Processing Units).

Table 3.6: Computation Complexity Performance

	S1/C3	S2/C1	S2/C2	SR1	SR2
Seconds/frame	1.55	1.21	1.72	1.77	1.83

Section 3: Constrained Multiple Kernel Tracking for Human Limbs

In the above two proposed system, the 2D blobs of the 2D body part tracking only consist of head, left/right hand and left/right foot, without involving joint blobs such as elbows and knees. In order to stabilize our proposed 3D pose estimation system, not only hands/feet but also elbows/knees and the while trunk of the limb are involved into the 2D body part tracking. Therefore, our goal is to track a non-rigid object composed of several rigid small objects.

Human body parts are suitable to be formulated multiple kernels because the body is composed of several rigid objects connected with joints. The multiple kernel tracking [43], [44] methods have been proposed to track non-rigid objects with some constraints. In [43], the object is modeled by unconstrained multiple kernels with Sum of Squared Differences (SSD), where the Newton-style iteration is adopted for tracking the object. Moreover, Fan et. al. [44] propose a multiple collaborative kernel tracking (MCKT) method, where a human limb is modeled by 3 kernels with two equality constraints. But in [44], the kernels' regions are too small to effectively represent the motions of the human limbs. Besides, in [43], [44], the linearization of the objective function could cause loss of accuracy and result in unstable tracking performance.

Here, we propose to use the color information of whole chunk limb in the multiple kernels for human limbs tracking. Moreover, the inequality constraints are applied to control the angle between the arm/forearm or upper/lower legs in varying poses of human movement.

3.3.a Problem Formulation

We propose a method to model and track limbs. First, each upper limb is divided into the arm, the elbow, and the forearm. On the other hand, the lower limb is divided into the upper leg, the knee, and the lower leg. Without loss of generality, we will mainly use upper limb for algorithmic discussions and illustrations. For example, one side of the arm and one side of the forearm are bound to the joint as shown in Figure 3.36. The location $\vec{a} = (a_x, a_y)$ of the joint is in charge of the translation of the whole upper limb. The rotations of the arm and the forearm are controlled by θ_1 and θ_2 separately. The lengths, $2l_1$ and $2l_2$, of the arm and the forearm are obtained in the initialization of the first frame (the reference model). Therefore, the locations of the centroids of the arm/forearm are $\vec{a} + \vec{v}_1$ and $\vec{a} + \vec{v}_2$ separately, where $\vec{v}_1 = (l_1 \cos\theta_1, l_1 \sin\theta_1)$ and $\vec{v}_2 = (l_2 \cos\theta_2, l_2 \sin\theta_2)$.

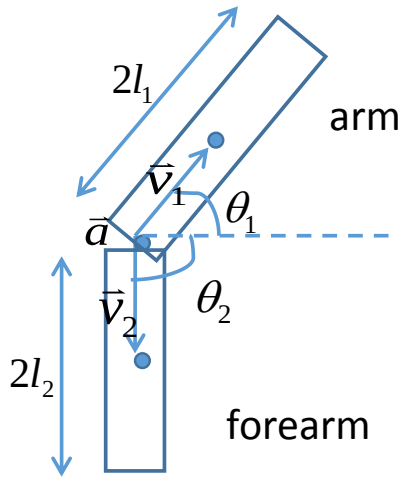


Figure 3.36: Illustration of the arm and forearm in an upper limb.

The color of the arm/forearm is used as the key feature for tracking. First, the color of the arm/forearm is weighted by a kernel function. In our implementation, we adopt Epanechnikov kernel [30] as shown in Equation 3.21. The m-bin histogram $\left(\hat{q}_{j,u}\right)$ of the reference model is formulated as Equation 3.22 [31], where $\left\{\bar{x}^*_{j,i}\right\}_{i=1,\ldots,n}^{j=1,2}$ is the normalized pixel locations in the region of the arm/forearm, centered at $\bar{0}$. The indicator function, b: $R^2 \rightarrow u \in \{1,\ldots,m\}$, maps the location of a pixel into the index of its histogram bin. δ is Kronecker delta function with $\delta[s]=1$ if $s=0$, otherwise $\delta[s]=0$. C_j is the normalization constant.

In a similar way, the *m*-bin histogram $\hat{p}_u\left(\vec{a}+\vec{v}_j\right)$ of the candidate model is formulated as Equation 3.23, where the arm/forearm is centered at $\vec{a}+\vec{v}_j$ with bandwidth h.

Equation 3.21:
$$k(x)=\begin{cases}1-x & if \quad 0\le x\le 1\\ 0 & if \quad x>1\end{cases}$$

Equation 3.22:
$$\hat{q}_{j,u}=\frac{1}{C_j}\left\{\sum_{i=1}^{n}k\left(\left\|\vec{x}^*_{j,i}\right\|^2\right)\delta\left[b\left(\vec{x}^*_{j,i}\right)-u\right]\right\}$$
$$\text{, where } C_j=\sum_{i=1}^{n}k\left(\left\|\vec{x}^*_{j,i}\right\|^2\right),\ j\in\{1,2\}$$

Equation 3.23:
$$\hat{p}_u\left(\vec{a}+\vec{v}_j\right)=\frac{1}{C_j^h}\left\{\sum_{i=1}^{n_h}k\left(\left\|\frac{\vec{a}+\vec{v}_j-\vec{x}_{j,i}}{h}\right\|^2\right)\delta\left[b\left(\vec{x}_{j,i}\right)-u\right]\right\}$$
$$\text{, where } C_j^h=\sum_{i=1}^{n_h}k\left(\left\|\frac{\vec{a}+\vec{v}_j-\vec{x}_{j,i}}{h}\right\|^2\right),\ j\in\{1,2\}$$

(i) Objective Function

The Bhattacharyya coefficient [41] is adopted to measure the similarity between the reference model histogram and the candidate model histogram based on Equation 3.24.

Equation 3.24: $\rho_j\left[\hat{q}_{j,u}, \hat{p}_u\left(\vec{a}+\vec{v}_j\right)\right]=\sum_{u=1}^{m}\sqrt{\hat{q}_{j,u}\hat{p}_u\left(\vec{a}+\vec{v}_j\right)}$

After combining Equation 3.21-Equation 3.24, the degree of the similarity of arm/forearm becomes Equation 3.25.

Equation 3.25: $\rho_j=\sum_{u=1}^{m}\sqrt{\frac{\left\{\sum_{i=1}^{n}\left(1-\left\|\vec{x}_{j,i}^{*}\right\|^2\right)\delta\left[b\left(\vec{x}_{j,i}^{*}\right)-u\right]\right\}\left\{\sum_{i=1}^{n_h}\left(1-\left\|\frac{\vec{a}+\vec{v}_j-\vec{x}_{j,i}}{h}\right\|^2\right)\delta\left[b\left(\vec{x}_{j,i}\right)-u\right]\right\}}{C_jC_j^h}}$

Therefore, the objective function f is formulated to maximize the degrees of similarity from both arm and forearm, as given in Equation 3.26. Because $\vec{a}=\begin{bmatrix}a_x\\a_y\end{bmatrix}, \vec{v}_j=\begin{bmatrix}l_j\cos\theta_j\\l_j\sin\theta_j\end{bmatrix}$, the objective function f can be represented as a 4-variable function in Equation 3.27 with constraint Equation 3.28, where β_{pre} denotes the angle between the arm and the forearm in the previous frame, that is, $\theta_2^{previous}-\theta_1^{previous}$. τ is a predefined constant to denote the maximum angle for the limb bending in one frame.

Equation 3.26: $f\left(\vec{a},\vec{v}_1,\vec{v}_2\right)=\sum_{j=1}^{2}\rho_j\left[\hat{q}_{j,u}, \hat{p}_u\left(\vec{a}+\vec{v}_j\right)\right]$

Equation 3.27: $f\left(a_x,a_y,\theta_1,\theta_2\right)=\sum_{j=1}^{2}\rho_j\left[\hat{q}_{j,u}, \hat{p}_u\left(\begin{bmatrix}a_x+l_j\cos\theta_j\\a_y+l_j\sin\theta_j\end{bmatrix}\right)\right]$

Equation 3.28: $\beta_{\min} = \max\left(0, \beta_{pre} - \tau\right) \le \theta_2 - \theta_1 \le \min\left(\beta_{pre} + \tau, \pi\right) = \beta_{\max}$

Hence, the tracking problem becomes finding the maximum of the objective function with inequality constraints as described in Equation 3.29.

Equation 3.29: $\vec{y}^* = \left(a_x^*, a_y^*, \theta_1^*, \theta_2^*\right) = \underset{a_x, a_y, \theta_1, \theta_2}{\arg\max}\ f\left(a_x, a_y, \theta_1, \theta_2\right)$
subject to $\theta_2 - \theta_1 - \beta_{\max} \le 0$, $\theta_1 - \theta_2 + \beta_{\min} \le 0$

3.3.b Gradient Projection

To solve the optimization problem, the gradient projection method [10] is adopted. The general problem is described as Equation 3.30, where A is an $m \times n$ matrix, $m < n$, $\vec{y}$ is an *n*-dim vector, and $\vec{b}$ is an *m*-dim vector.

Equation 3.30: **maximize** $f\left(\vec{y}\right)$
subject to $A\vec{y} - \vec{b} < \vec{0}$

Based on the derivations in [45], the gradient projection search vector $\vec{u}$ and the projection matrix P is defined in Equation 3.31 and Equation 3.32, where $\nabla f\left(\vec{y}\right)$ is the gradient of the objective function f at $\vec{y}$.

Equation 3.31: $\vec{u} = -P\nabla f\left(\vec{y}\right)$

Equation 3.32: $P = I - A^T\left(AA^T\right)^{-1}A$

For the inequality constraints problem, $A = A_r$ (an $r \times n$ matrix, $r \le m$) denotes the *r* active constraints, that is, $A_r\vec{y}^k - \vec{b}_r = \vec{0}$ at the current point $\vec{y}^k$. Then the next search point is $\vec{y}^{k+1} = \vec{y}^k + \alpha\vec{u}^{k+1}$, where $\vec{u}^{k+1} = -P_r\nabla f\left(\vec{y}^k\right)$, $P_r = I - A_r^T\left(A_rA_r^T\right)^{-1}A_r$, and α can be found through line search algorithm or set as a constant.

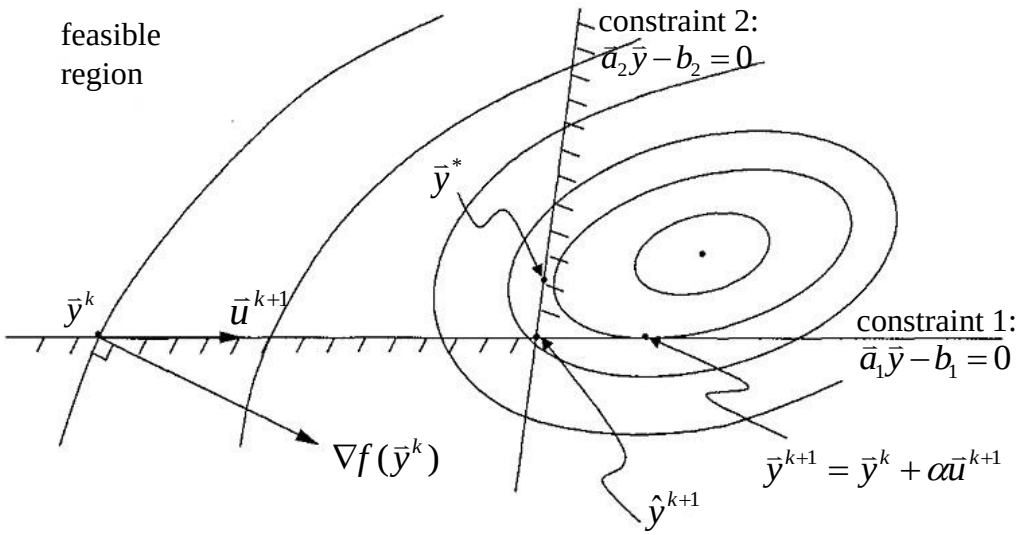


Figure 3.37: Illustration of the gradient projection with inequality constraints [45].

As shown in Figure 3.37, during the search for $\bar{y}^{k+1}$ along the direction $\bar{u}^{k+1}$, if an additional constraint is encountered along $\bar{u}^{k+1}$, intersecting $\bar{y}^{k+1}=\bar{y}^k+\alpha\bar{u}^{k+1}$ at $\bar{y}^{k+1}$, set $\bar{y}^{k+1}=\hat{y}^{k+1}$, add the new constraint to the active set, with associated matrix A_{r+1}, and the projection matrix becomes P_{r+1}. The optimal solution is reached at $\bar{y}^*$, when $P\nabla f\left(\bar{y}^*\right)=0$.

3.3.c Tracking Mechanism

The first frame of the video sequence is used to initialize the reference model. The position $\bar{a}^{ref}=\left(a_x^{ref},a_y^{ref}\right)$ of the limb and the centroids, $\bar{a}^{ref}+\bar{v}_1^{ref}$ and $\bar{a}^{ref}+\bar{v}_2^{ref}$, of the arm (or upper leg) and forearm (or lower leg) are computed based on the selected region. Hence, the length $2l_j$ and the angle θ_j^{ref} (see Figure 3.36) can be estimated. Moreover, the color histogram of the reference model, $\{\hat{q}_{j,u}\}$, is computed based on Equation 3.22.

In the tracking stage, the best limb pose $\vec{y}^{prev} = \left(a_x^{prev}, a_y^{prev}, \theta_1^{prev}, \theta_2^{prev}\right)$ in the previous frame is used as the initial point $\vec{y}^0$ to compute the color histogram of the candidate model, $\left\{\hat{p}_u\left(\vec{a}+\vec{v}_j\right)\right\}$, in the current frame based on Equation 3.23. The gradient projection method with inequality constraints is applied to find the optimal limb pose $\vec{y}^*$ in the current frame.

Equation 3.33: $\nabla f = \left(\frac{\partial f}{\partial a_x}, \frac{\partial f}{\partial a_y}, \frac{\partial f}{\partial \theta_1}, \frac{\partial f}{\partial \theta_2}\right)$

, where

$$\frac{\partial f}{\partial a_p} = \sum_{j=1}^{2}\sum_{u=1}^{m}\sqrt{\frac{\hat{q}_{j,u}}{\hat{p}_u\left(\vec{a}+\vec{v}_j\right)}}\frac{\left\{\sum_{i=1}^{n_h} Q_{j,i,p}\,\delta\left[b\left(\vec{x}_{j,i}\right)-u\right]\right\} - \hat{p}_u\left(\vec{a}+\vec{v}_j\right)\left\{\sum_{i=1}^{n_h} Q_{j,i,p}\right\}}{2C_j^h}$$

$$\frac{\partial f}{\partial \theta_j} = \sum_{u=1}^{m}\sqrt{\frac{\hat{q}_{j,u}}{\hat{p}_u\left(\vec{a}+\vec{v}_j\right)}}\frac{\left\{\sum_{i=1}^{n_h} R_{j,i}\,\delta\left[b\left(\vec{x}_{j,i}\right)-u\right]\right\} - \hat{p}_u\left(\vec{a}+\vec{v}_j\right)\left\{\sum_{i=1}^{n_h} R_{j,i}\right\}}{2C_j^h}$$

$$Q_{j,i,p} = \frac{2\left(a_p + v_{j,p} - x_{j,i,p}\right)}{h^2},\ p \in \{x, y\}$$

$$R_{j,i} = \frac{2\left(a_y v_{j,x} - a_x v_{j,y} + x_{i,x} v_{j,y} - x_{i,y} v_{j,x}\right)}{h^2}$$

$$\vec{a} = \begin{bmatrix} a_x \\ a_y \end{bmatrix}, \vec{v}_j = \begin{bmatrix} v_{j,x} \\ v_{j,y} \end{bmatrix} = \begin{bmatrix} l_j \cos\theta_j \\ l_j \sin\theta_j \end{bmatrix},\ j \in \{1,2\}$$

In the gradient projection method, $\vec{y}^*$ is reached, with an initial point $\vec{y}^0$, by iteratively moving the current point $\vec{y}^k$ to the next point $\vec{y}^{k+1}$ along the gradient projection search vector $\vec{u}^{k+1}$. The gradient of the objective function f is computed as Equation 3.33.

Finally, the selected region in the reference model is translated and rotated to the best estimated human limb pose based on $\vec{y}^*$. The contour of the transformed region is labeled and highlighted on the current frame.

3.3.d Experimental Results

We implement the proposed method for human limb tracking, and conduct experiments on the Brown HumanEva I dataset [42] and self-recorded videos. Besides, we also compared our method with MCKT [44].

(i) Qualitative Results

In Figure 3.38 (Subject 1), the human upper limb (arm and forearm) is tracked. The red region represents the arm, while the green region represents the forearm, with the white dot denoting the elbow. Moreover, in Figure 3.39 (Subject 2), the lower limb (upper leg and lower leg) is tracked. The red region denotes the upper leg, while the green region denotes the lower leg, with the white dot representing the knee. As shown in Figure 3.39, the lower limb can be well tracked, even if the upper leg is occluded by the hand at some frames. Furthermore, we also conduct experiments on the self-recorded video, where the illumination variations increase the difficulty of tracking the limbs. As shown in Figure 3.40 (Subject 3), the upper limb can still be well tracked in the self-recorded video sequence.

Figure 3.38: Snapshots for upper limb tracking (Subject 1).

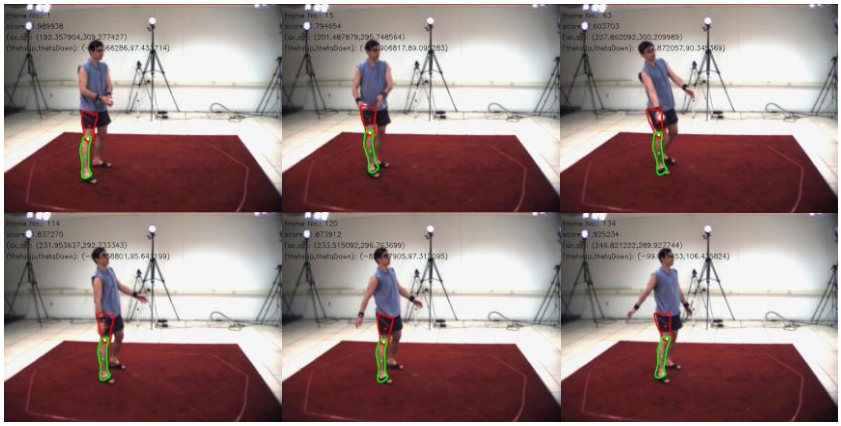


Figure 3.39: Snapshots for lower limb tracking (Subject 2).

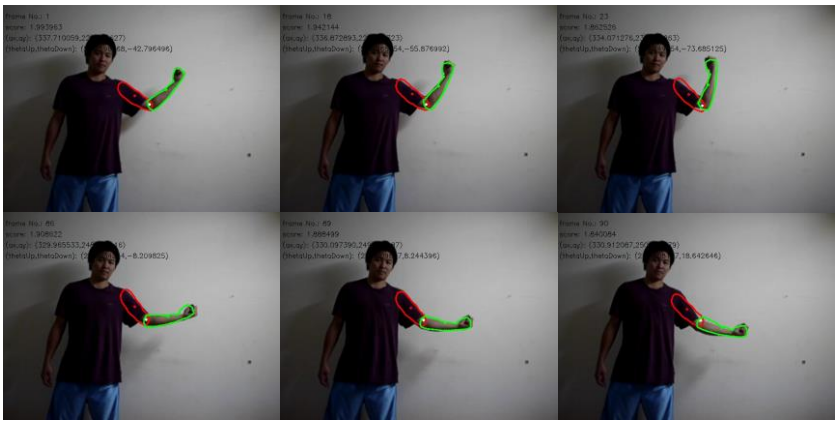


Figure 3.40: Snapshots for upper limb tracking (Subject 3).

Three methods, individually MCKT (Multiple Collaborative Kernel Tracking [44]), WoC (the proposed method without constraints) and WC (the proposed method with inequality constraints), for Subject 1 are compared as shown in Figure 3.41.

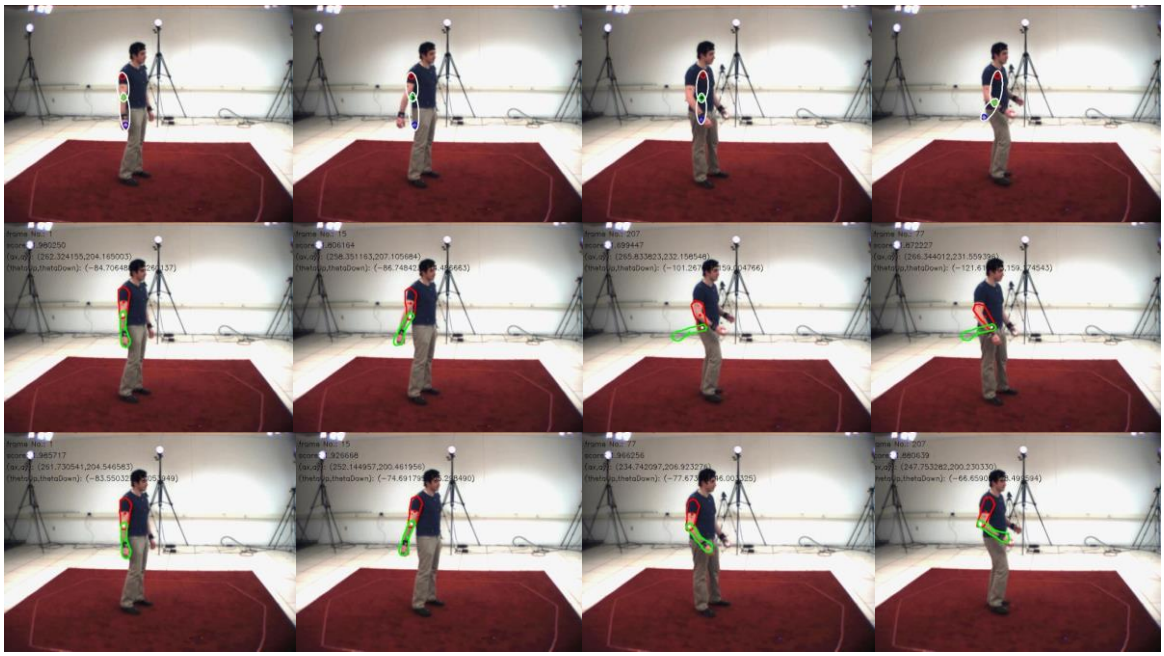


Figure 3.41: Comparison for Subject 1: the first row shows the results of MCKT [43]; the second row shows the results of the proposed method without constraints (WoC); the third row shows the results of the proposed method with inequality constraints (WC).

(ii) Quantitative Results

As shown in Table 3.7, three subjects in the video sequences are evaluated for the tracking performance with 210 frames for Subject 1, 162 frames for Subject 2, and 90 frames for Subject 3. The frame size is 640x480 for all the subjects. In our evaluations, DIST_AVE denotes the average distance (in pixels) over all the frames between the locations of the tracked and ground-truth joints, such as elbow and knee. Moreover, AND_AVE denotes the average over all video frames of all the normalized overlapped

area between the ground-truth limb area and the tracked limb area by the area of the ground-truth limb. The higher AND_AVE means better performance. Furthermore, AND_STD denotes the standard deviation of all the normalized AND areas over all video frames. Small AND_STD indicates better consistency of tracking performance. The comparison of the whole video sequence for Subject 1 with 3 methods is shown in Figure 3.42. The result is published in [47].

Table 3.7: Performance Evaluation for 3 Subjects

		DIST_AVE	AND_AVE	AND_STD
Subject 1	**MCKT**	19.348	0.284	0.141
	WoC	29.962	0.414	0.183
	WC	**7.388**	**0.806**	0.069
Subject 2	**MCKT**	39.831	0.090	0.213
	WoC	14.980	0.572	0.126
	WC	**12.625**	**0.810**	0.087
Subject 3	**MCKT**	44.439	0.669	0.192
	WoC	23.198	0.716	0.178
	WC	**11.240**	**0.842**	0.062

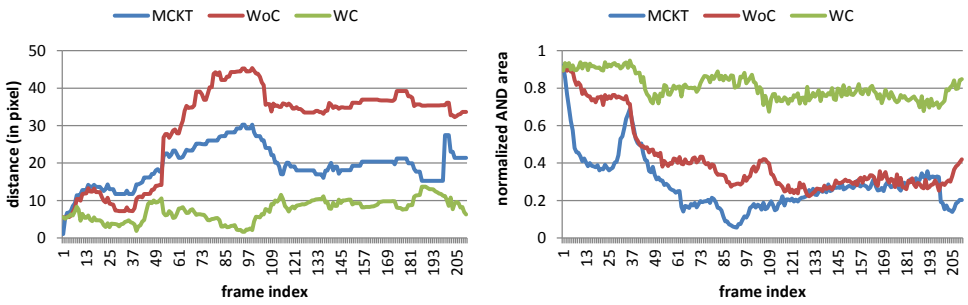


Figure 3.42: Comparison for Subject 1: The left figure shows the distance between the tracked and ground-truth joints, while the right figure shows the normalized AND (overlapped) area.

Section 4: Discussion

3.4.a Limitations

There are mainly 3 types of limitations in our proposed 3D human pose estimation system.

First, the elevation angle of views is limited. The larger the elevation angle of views, the less the shape and appearance information of the body. For example, in the top view of the human body, the only information is the head of the human body. It is extremely difficult to obtain the shape of body, the color of clothes, and the body configuration for monocular video sequences.

Second, the accurate initialization of 3D human pose estimation for one single frame is lacking. Our proposed method deals with the 3D human tracking and pose estimation, but does not deal with the initialization of 3D human pose estimation. The issue of initialization of 3D human pose estimation means to perform 3D pose estimation independently on each frame. Therefore, it is very difficult to recover once the tracking is lost. Although we designed a tracking recovery mechanism, it only works when the degree of tracking lost is small. If the tracking loss is heavy or complete, it is much more difficult to recover it back without the initialization for a single frame.

Third, we only deal with the self-occlusion of the body parts, but do not deal with the object occlusion. For example, if the human body is occluded by a machine for the

lower body part, the proposed method will fail to accurately estimate the 3D poses of the human body.

3.4.b Potential Extensions

There are mainly 3 types of potential extensions for our proposed 3D human pose estimation system.

First, the proposed method can be extended to 3D human pose estimation for multiple persons with some interactions with each other. Our proposed method smoothly deals with 3D human pose estimation on single human body. Combined with additional handling on the occlusions of different body parts from different persons, our proposed method can easily deal with the 3D human pose estimation for multiple persons with some interactions with each other.

Second, the proposed method can be applied in multiple cameras system. Our proposed method favorably performs 3D human pose estimation on monocular video sequences. Combined with the additional camera calibration, the estimated 3D poses from each camera can be integrated to generate a more reliable 3D human pose estimation system.

Third, the proposed method can be integrated with a depth camera. The depth camera has not only R, G, B, the color information, but also D, the depth information. With the additional depth information, the issue of the occlusion can be further eased. Moreover, a depth camera, RGB-D, can be used to resolve the initialization and provide

the solution to the lost tracking. For example, one of the well-known depth cameras is Microsoft Kinect. Kinect algorithms provide the 3D human pose estimation independently for each frame. Our proposed tracking and occlusion handling algorithms can easily be integrated with the Kinect algorithms to provide a more reliable 3D human pose estimation system.

Chapter 4 – Human Action Recognition

Human action recognition is useful for many applications, such as surveillance, entertainment and healthcare. In this chapter, we proposes a system to recognize both single and continuous human actions, from monocular video sequences, based on 3D human pose estimation and cyclic hidden Markov models (CHMMs). First, for each frame in a monocular video sequence, the 3D coordinates of joints of the human object with actions of multiple cycles are extracted using the proposed 3D human modeling technique as described in Chapter 3. The 3D coordinates are then converted into a set of geometrical relational features (GRFs) for dimensionality reduction and increase of discrimination. For further dimensionality reduction, the k-means clustering is applied to those GRFs to generate clustered feature vectors. These vectors are used to train CHMMs separately for different types of actions based on the Baum-Welch reestimation algorithm. For recognition of continuous actions, which are concatenated from several distinct types of actions, a graphical model is used to systematically concatenate different separately trained CHMMs. The experimental results show the effective performance of our proposed system in both single and continuous action recognition problems.

The overview of the proposed system is shown in Figure 4.1. The inputs of the system are monocular video sequences. In the first phase, the human object is segmented from the video frames, and the 3D human poses are estimated with the corresponding 3D coordinates of body joints based on the proposed 3D human pose estimation technique, whose details are described in Chapter 3. In the second phase, the estimated 3D

coordinates of body joints are converted into one-dimensional feature vectors based on GRF conversion [63] and k-means clustering [64]. In the third phase, one-dimensional feature vectors are used to train CHMMs, one model for one type of action, and then the designed graphical model is used to recognize continuous human actions concatenated from different types of human actions by switching CHMMs. Finally, the recognized human actions are created.

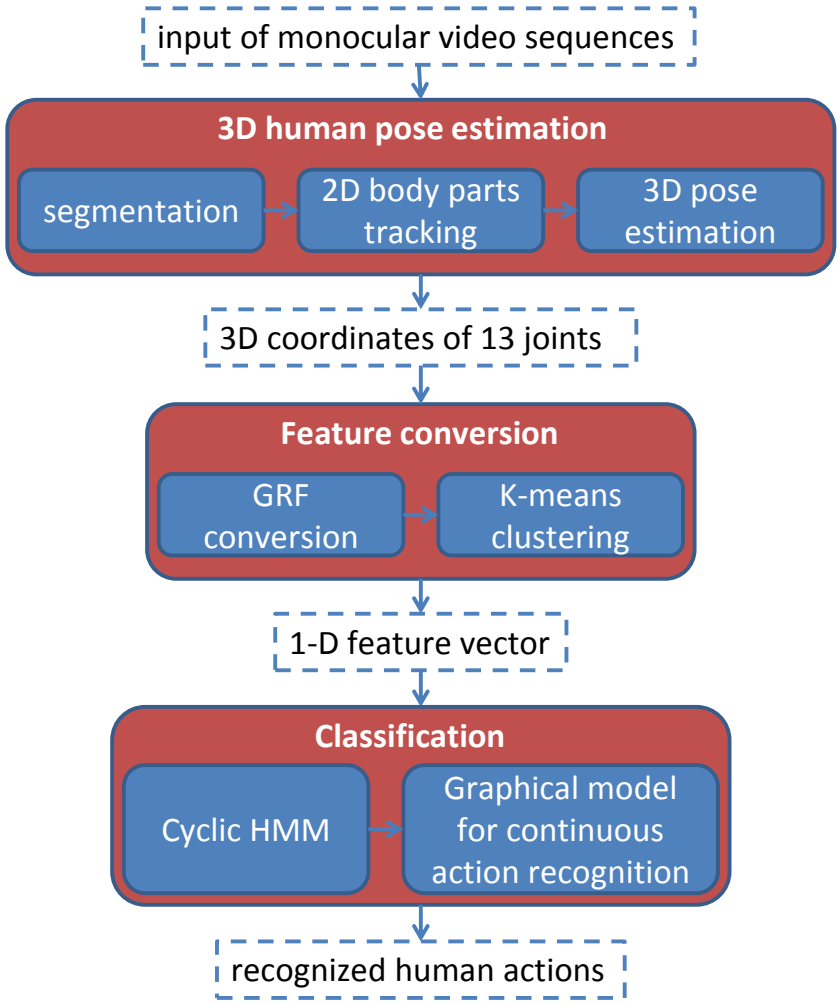


Figure 4.1: The overview of the propose system for human action recognition.

Section 1: Feature Conversion

The 3D coordinates of 13 human joints as shown in Figure 4.2 are extracted from the estimated human poses. These 3D coordinates are further converted into one symbol at each frame for dimensionality reduction and discrimination increase. Two operations are considered individually: geometrical relational feature (GRF) conversion [62], [63] and k-means clustering [64]. The dimensionality reduction is shown in Figure 4.3.

After the input video sequences are converted into one-dimensional feature vectors, these feature vectors of different types of actions with various repeated cycles are used to train the corresponding cyclic HMMs (CHMMs) [65], with one action being modeled by one CHMM, based on the Baum-Welch algorithm [66]. Subsequently, a switching graphical model is designed to switch different CHMMs for a long video sequence, which is concatenated from different types of human actions, and to recognize the continuous combined human actions based on the maximal likelihood of the observation sequence, computed by forward/backward and Viterbi algorithms [66].

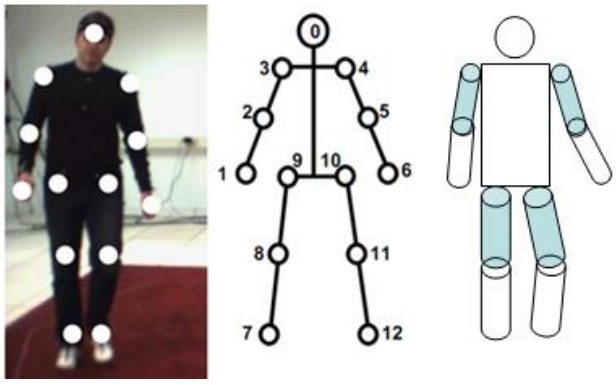


Figure 4.2: Human body model. From left to right, it shows a human image, the corresponding 13-point joints model, and the corresponding 3D human model.

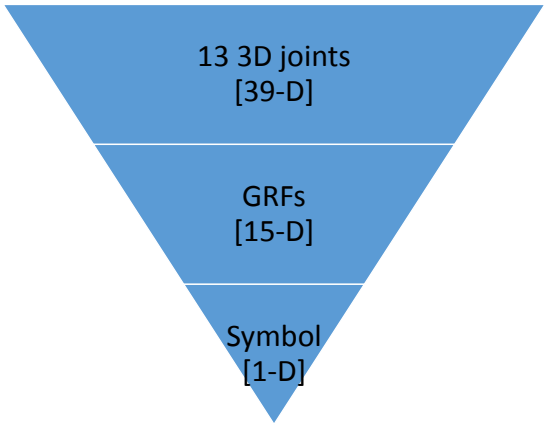


Figure 4.3: The flowchart of dimensionality reduction.

4.1.a GRF Conversion

The GRF descriptor defines the degree of the relative position of the body parts in 15 dimensions, as described in Table 4.1. Two types of features are included in the GRF descriptor, respectively distance-related features (F1~F9) and angle-related features (F10~F15). The features of F1~F4 define the geometrical relation between a point and a plane. Take an example of F1, which is the measurement of the distance between the right hand and the plane formed by right shoulder, left pelvis and right pelvis. Let p1, p2, p3, p4 respectively be the centroids of right shoulder, left pelvis, right pelvis and right hand. As defined in Equation 4.1, F1 is the inner product of the vector (p4 – p3) and normal vector n , which is the normal vector of the plane formed by p1, p2, p3, computed by the cross product of the vectors (p1 – p3) and (p2 – p3). The features of F5 and F6 define the geometrical relation between two vectors. One vector is defined as (R-Shoulder, R-Hand) or (L-Shoulder, L-Hand), and the other vector is a unit vector, defined

as (middle of L-Pelvis and R-Pelvis, Head). The inner product of these two vectors is used to provide the signed distance to indicate the degree of R/L-Hand above or below R/L-Shoulder. The feature of F7 is defined as the distance between the centroid of the body and the lowest foot in vertical direction (Y coordinate). F7 might have large changes in the action of sit_down and get_up. The feature of F8 is defined as the distance between R-Foot and L-Foot in vertical direction (Y coordinate). The feature of F9 is defined as the accumulated distance between the centroid of the body at the current frame and the centroid of the body at the 1st frame. F9 can provide a hint that the body acts always at the original location, or acts with some movement, such as spinning versus circling. Moreover, the angle-related features of F10~F13 measure the degree of the angle in π unit of a body part. Take an example of F10, which is the measurement of the angle between the upper and the lower right arm, as defined in Equation 4.2, where n1, n2 are respectively the unit vector of the upper and the lower right arm, and F10 is the arccosine of the inner product of two unit vectors n1, n2. The feature of F14 is defined as the degree of the angle of the body bending vertically (X coordinate). The feature of F15 is defined as the change of the angle of the body rotation horizontally (Y coordinate) between the previous frame and the current frame. The effectiveness of the definition of GRFs will be proved in experiments later.

Equation 4.1: $$F_1 = \langle n, (p_4 - p_3) \rangle, n = \frac{(p_1 - p_3) \times (p_2 - p_3)}{\|(p_1 - p_3) \times (p_2 - p_3)\|}$$

Equation 4.2: $$F_{10} = \arccos(\langle n_1, n_2 \rangle)$$

Table 4.1: The Detailed Description of GRFs

Feature	Description
$\mathbf{F_{1,2}}$	Signed distance between R/L-Hand and the plane defined by R/L-Shoulder, L-Pelvis, R-Pelvis
$\mathbf{F_{3,4}}$	Signed distance between R/L-Foot and the plane defined by L-Shoulder, R-Shoulder, R/L-Pelvis
$\mathbf{F_{5,6}}$	Signed distance between vector of R/L-Hand and R/L-Shoulder, and unit vector of the middle of two Pelvises and Head
$\mathbf{F_{7}}$	Distance between the centroid of the body and the lowest foot in Y coordinate
$\mathbf{F_{8}}$	Distance between R-Foot and L-Foot in Y coordinate
$\mathbf{F_{9}}$	Accumulated distance between the centroid of the body at current frame and the centroid of body at 1st frame
$\mathbf{F_{10,11}}$	Angle between upper and lower R/L-Arm
$\mathbf{F_{12,13}}$	Angle between upper and lower R/L-Leg
$\mathbf{F_{14}}$	Angle of the body bending vertically in X coordinate
$\mathbf{F_{15}}$	Angle change of body rotation horizontally between the previous frame and the current frame in Y coordinate

4.1.b k-means Clustering

The 15-dimensional GRF vectors are further clustered into k centroids (codewords) by the k-mean algorithm. Each GRF vector can be represented by one of the k codewords, i.e., a symbol, based on the nearest centroid. Therefore, each frame is eventually converted into one symbol out of k possible values, and the input monocular video sequence can now be represented as a one-dimensional feature vector. In our experiments, the k value is set as 64.

Section 2: Classification Algorithm

4.2.a Introduction to Hidden Markov Model

An HMM is specified by three terms [66], $\Phi = (\pi, \quad A, \quad B)$. The first term (π) is the initial probability of hidden states. The second term (A) is the transition matrix, which specifies a transition probability from one hidden state to another hidden state. The third term (B) is the observation matrix, which specifies the probability of the observed symbol given a hidden state. As addressed in [67], three types of problems of HMMs are addressed.

- *The evaluation problem*: Given a model $Y=\Phi$ and an observation sequence X, what is the probability (likelihood), $P(X \mid \Phi)$, of X given the specified model Φ ? This problem can be efficiently solved by the forward/backward algorithm.
- *The decoding problem*: Given a model Φ and an observation sequence X, what is the most likely underlying hidden state sequence that produces the observation sequence? This problem can be efficiently solved by the Viterbi algorithm.
- *The learning problem*: Given a model Φ and an observation sequence X, how can we adjust the model parameters $\Phi = (\pi, \quad A, \quad B)$ to maximize the conditional probability (likelihood) $\prod_X P(X \mid \Phi)$? This problem can be efficiently solved by the Baum-Welch reestimation algorithm.

HMMs have been popularly used to model time-sequential data such as speech and video. Bitar et al. [68] employ acoustic parameters (Aps) as a signal representation of speech in a HMM recognition for speech recognition. Moreover, Yamato et al. [50] train

HMMs to recognize actions of different tennis strokes by the Baum-Welch reestimation algorithm.

4.2.b Cyclic Hidden Markov Model

An HMM is a doubly stochastic process with an underlying hidden stochastic process and an observed stochastic process. Many human actions exhibit the quasi-periodicity, such as walking and waving, where the repeated movements are not identical in each cycle and the number of cycles is also indefinite without a predefined value. This motivates our use of the cyclic HMM (CHMM), which is an HMM which has left-to-right structure with a return transition from the last state to the first state to have capability to model the actions with multiple cycles, as shown in Figure 4.4 [65]. Each CHMM is trained by the corresponding type of action sequences with variable multiple cycles in the training dataset. The action sequences in testing dataset can be recognized by finding the maximal likelihood based on the forward/backward algorithm.

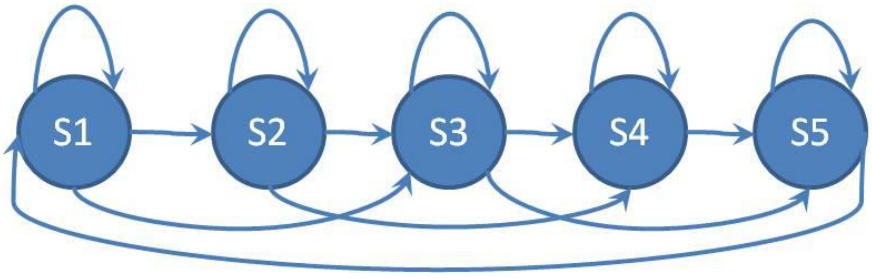


Figure 4.4: The transition graph of hidden states for a CHMM.

4.2.c Graphical Model for Continuous Action Recognition

Based on the trained single-action CHMMs, a graphical model, based on the Graphical Model Tool Kits [69], is designed for recognition of continuous human actions, which are concatenated from several different human actions.

The template model of an HMM is shown in Figure 4.5 (a). For an HMM, the hidden state nodes S are represented by the blue circles, while the observed nodes X are denoted by the red circles. And an arrow shows the dependency between two nodes. In our implementation, there are 5 hidden states and $k = 64$ observation symbols in a CHMM.

Moreover, to switch CHMMs for different types of actions, a graphical model is designed for recognition of continuous combined human actions. The template model of a switching graphical model is shown in Figure 4.5 (b). Besides state and observation nodes, there are 3 other hidden nodes, including ActIndex, ActID and the delta Δ nodes. The ActIndex nodes denote the index of the concatenation of actions. In our implementation, 6 actions are concatenated; hence, ActIndex is at $\{0 \sim 5\}$. The ActID nodes specify the type of actions, which could be boxing, kicking, throwing, or waving action in our implementation, and take the value of $\{0 \sim 4\}$ individually. The delta Δ nodes serve as the switching nodes to decide when to switch from one action to another action based on the value of 0 or 1. The dash arrows in Figure 4.5 (b) represent the dependency of switching parent nodes. For example, in Chunk, the delta Δ node and the ActID node are the switching parents of the state node. The probability relationship

between switching parent and its child is shown in Equation 4.3. In the *i*th frame, if the delta is 0, the current transition probability is specified by $P(S_i \mid S_{i-1})$ of the current cyclic HMM. On the other hand, if the delta is 1, the current transition probability is specified by $P(S_i \mid ActID_i)$, which is the initial state probability of the next cyclic HMM with the $ActID_i$. These template models are implemented through GMTK (Graphical Model Tool Kits) [69] in our system.

Equation 4.3:

$$P(S_i|S_{i-1},ActID_i) = \sum_{\Delta} P(S_i,\Delta|S_{i-1},ActID_i)$$
$$= P(S_i|S_{i-1},ActID_i,\Delta=0)P(\Delta=0) + P(S_i|S_{i-1},ActID_i,\Delta=1)P(\Delta=1)$$
$$= P(S_i|S_{i-1},\Delta=0)P(\Delta=0) + P(S_i|ActID_i,\Delta=1)P(\Delta=1)$$

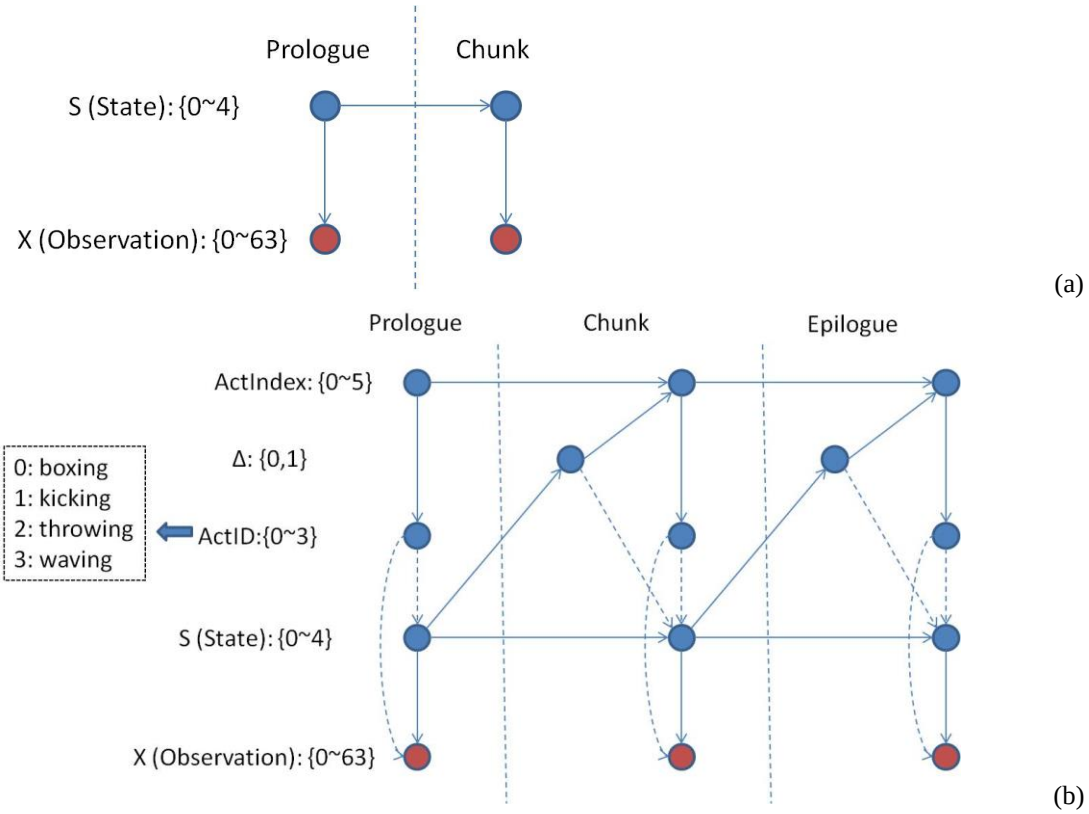


Figure 4.5: (a) the template model for an HMH, (b) the template model for a switching graphical model.

Section 3: Recognition of Single Human Action

The experiment of single-action recognition is performed on two different datasets, the self-recorded dataset and the IXMAS dataset [59]. Each type of human actions is trained as one single-action CHMM. And then each video sequence in a test dataset is recognized by the trained single-action CHMMs through computing the maximal likelihood. Moreover, the evaluations for classification algorithms are also provided in Section 4.3.c.

4.3.a Self-Recorded Dataset

In the self-recorded dataset, the same as 3.2.e.i, 4 persons {PersonA, PersonB, PersonC, PersonD} performed 4 different types of actions {Boxing, Kicking, Throwing, Waving} with 5 cycles in the self-recorded videos. Besides, a standing action is also recorded from videos. Therefore, totally there are 100 monocular video sequences with a single cycle. The representative snapshots of the video sequences after 3D human modeling for {Boxing, Kicking, Throwing, Waving} by {PersonA, PersonB, PersonC, PersonD} are shown in Figure 3.32.

Each action sequence (short sequence) is formed by the same type of actions with various numbers of cycles (1-5). For each type of actions for each person, 15 variable-cycle sequences are formed from one 5-cycle (1-2-3-4-5), two 4-cycle (1-2-3-4, 2-3-4-5), three 3-cycle (1-2-3, 2-3-4, 3-4-5), four 2-cycle (1-2, 2-3, 3-4, 4-5), and five 1-cycle (1, 2, 3, 4, 5) action sequences. Therefore, in total 300 (15 short sequences x 4 persons x 5

actions) short sequences, with variable cycles (1-5 cycles), can be formed. Two experiments are performed for single-action recognition.

The first experiment is the recognition generalization of the unknown person, i.e., every single-action CHMM is trained by actions from 3 persons, and recognized by the actions performed by the remaining 4th person. Therefore, 225 short sequences (15 short sequences x 3 persons x 5 actions) are used for training, i.e., 5 CHMMs for 5 actions. 75 short sequences (15 short sequences x 1 person x 5 actions) are used for recognition. There are four sub-experiments for 4 different persons in the testing dataset, as shown in Figure 4.6. The overall confusion matrix is shown in Table 4.2 with the average recognition rate 92.7%. The table shows the boxing action performed by one person is more similar to the throwing action by the other 3 persons, than to the boxing action by the other 3 persons. Similarly, the waving action performed by one person is more similar to the throwing action by the other 3 persons, than to the waving action by the other 3 persons.

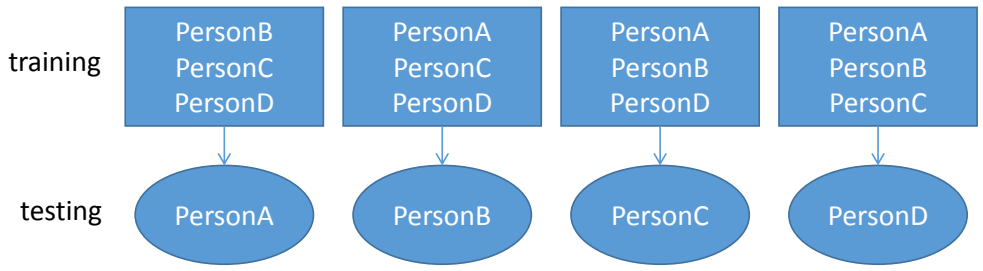


Figure 4.6: The 4 sub-experiments in unknown person recognition for the 4 different persons.

The second experiment is the cross-validation for the mixture of persons. For each type of action, out of 60 short sequences, 48 sequences are randomly selected for training,

and the remaining 12 sequences are for testing in each CHMM, that is, in total, 240 sequences for training and 60 sequences for testing. There are 5 independent sub-experiments are performed for cross-validation. The confusion matrix for all the 5 sub-experiments is shown in Table 4.3, which shows the perfect recognition rate (100%) for the mixture of persons.

Table 4.2: Confusion Matrix for Unknown Person Recognition

All	Boxing	Kicking	Throwing	Waving	Standing
Boxing	86.7%	0	11.7%	1.6%	0
Kicking	0	98.3%	1.7%	0	0
Throwing	0	0	95%	5%	0
Waving	0	0	16.7%	83.3%	0
Standing	0	0	0	0	100%

Table 4.3: Confusion Matrix for Mixture of Persons

All	Boxing	Kicking	Throwing	Waving	Standing
Boxing	100%	0	0	0	0
Kicking	0	100%	0	0	0
Throwing	0	0	100%	0	0
Waving	0	0	0	100%	0
Standing	0	0	0	0	100%

4.3.b IXMAS Dataset

To further justify the effectiveness of our proposed system, we also experiment on a more complex data, the IXMAS dataset [59], the same as 3.2.e.i, where 11 types of actions are performed by 12 actors, including “check_watch”, “cross_arm”,

“scratch_head”, “sit_down”, “get_up”, “turn_around”, “walk”, “wave”, “punch”, “kick” and “pick_up”. The snapshots of the 11 types of actions are shown in Figure 4.7. IXMAS dataset has 5 different view angles with camera0~camera4. To ensure monocular video sequences, the experiments are performed on the camera3 in IXMAS dataset.

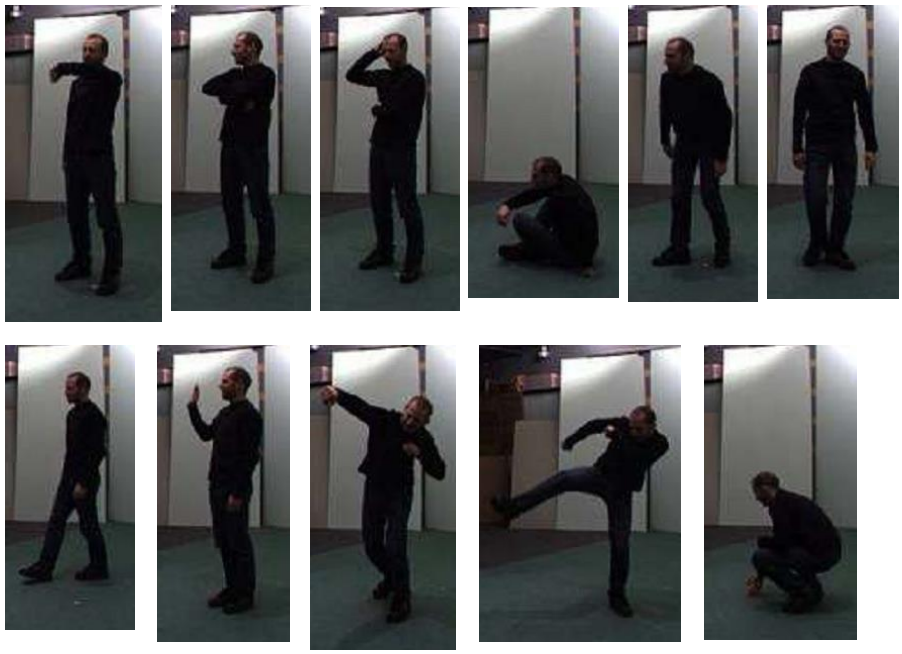


Figure 4.7: The snapshots of the 11 types of actions in IXMAS.

First, the estimated 3D coordinates are converted into GRFs as features and CHMM as the recognition algorithm (exactly the same as that used in Section 3.2.e.i). In order to show the effectiveness of the GRFs, we also compare the recognition performance based on features of the absolute 3D coordinates of joints and relative 3D coordinates of joints. The absolute 3D coordinates are the 3D joints with respect to the world’s origin, while the relative 3D coordinates are the 3D joints with respect to the centroid of the human body. Moreover, we also compare the proposed method with 3

other up-to-date methods using IXMAS dataset. The first method is proposed by Weinland et al. [59] to represent 3D exemplars, the 3D occupancy grids obtained from the projections of multiple cameras, as features and HMMs as the recognition algorithm. The second method is proposed by Junejo et al. [70] to represent SSM plus HoG and optical flow (OF) as features and SVMs as the recognition algorithm. The third method is proposed by Laptev et al. [71], [72] to represent space-time interest points (STIPs) as features and SVMs as the recognition algorithm. The comparison of the recognition rate is shown in Table 4.4, and the confusion matrix of the proposed method is shown in Table 4.5. As shown in Table 4.4, our proposed method of GRFs with 91.7% outperforms the performance of absolute 3D coordinates with 74.2% and relative 3D coordinates with 78.6%. It shows the more effective and discriminative representation of the GRFs. Besides, the proposed method of GRFs + CHMM with 91.7% also outperforms 3D exemplar + HMM [59] with 80.5%, SSM_HoG_OF + SVM [70] with 71.2%, and STIP + SVM [71], [72] with 85.5%. The improved performance of the proposed method most likely is contributed from three main factors. The first factor is the use of 3D pose estimation, which allows the invariance of viewing perspectives, as well as mitigates the effects caused by the illumination changes and the body occlusions. The second factor is the effectiveness of the conversion of the GRFs, which not only decreases the dimensionality to result in better CHMM training, but also increases the discrimination of the features. The third factor is the use of cyclic HMMs, which enable the same actions to repeat variable number of times.

Table 4.4: Comparison of Recognition Rates over IXMAS

Method	Recognition Rate (%)
3D exemplar + HMM [59]	80.5
SSM_HoG_OF + SVM [70]	71.2
STIP + SVM [71], [72]	85.5
absolute 3D + CHMM	74.2
relative 3D + CHMM	78.6
GRF + CHMM	**91.7**

Table 4.5: Confusion Matrix of the Proposed Method in IXMAS

Single_Action (%)	check_ watch	cross_ arm	scratch_ head	sit_ down	get_ up	turn_ around	walk	wave	punch	kick	pick_ up
check_watch	92	0	0	0	0	0	0	0	8	0	0
cross_arm	0	92	0	0	0	0	0	0	8	0	0
scratch_head	0	0	92	0	0	0	0	0	8	0	0
sit_down	0	0	0	100	0	0	0	0	0	0	0
get_up	0	0	0	0	100	0	0	0	0	0	0
turn_around	0	0	0	0	0	75	25	0	0	0	0
walk	0	0	0	0	0	0	100	0	0	0	0
wave	8	0	8	0	0	0	0	83	0	0	0
punch	0	0	0	0	0	0	0	0	92	8	0
kick	0	0	0	0	0	0	0	0	0	100	0
pick_up	0	0	0	17	0	0	0	0	0	0	83

4.3.c Evaluation for Classification Algorithms

With the same sequential data, that is, GRF sequences, CHMM is compared with the Dynamic Time Warping (DTW) on IXMAS dataset. The confusion matrices of the DTW with GRFs and CHMM with GRFs are respectively shown in Table 4.6 and Table 4.5. The comparison of the DTW and CHMM is provided in Table 4.7, in which DTW with 68.2% and CHMM with 91.7% recognition rates. It shows our proposed GRFs is much more appropriate for CHMM than for DTW.

Table 4.6: Confusion Matrix of DTW with GRFs in IXMAS

Single_Action_DTW (%)	check_ watch	cross_ arm	scratch_ head	sit_ down	get_ up	turn_around	walk	wave	punch	kick	pick_ up
check_watch	83	0	0	0	0	0	0	17	0	0	0
cross_arm	0	75	17	0	0	0	0	8	0	0	0
scratch_head	8	8	50	0	0	0	0	8	8	17	0
sit_down	8	0	0	83	8	0	0	0	0	0	0
get_up	0	0	0	0	100	0	0	0	0	0	0
turn_around	0	8	0	0	0	75	8	0	0	0	8
walk	8	0	0	0	0	8	75	0	0	8	0
wave	8	0	17	0	0	0	0	75	0	0	0
punch	8	8	8	0	0	0	8	17	42	8	0
kick	8	8	17	0	0	0	8	17	25	17	0
pick_up	0	0	0	0	8	8	8	0	0	0	75

Table 4.7: Comparison of DTW and CHMM in IXMAS

	GRFs + DTW	GRFs + CHMM
Recognition rate (%)	68.2	**91.7**

Section 4: Recognition of Continuous Human Actions

In the experiment of continuous-action recognitions, different action types are combined to form long video sequences. The long video sequences with distinct types of actions are recognized by the designed switching graphical model, which is concatenated from different types of trained single-action CHMMs. The experiments are performed on two datasets, self-recorded dataset and IXMAS dataset, which are the same as Section 4.3.

4.4.a Self-Recorded Dataset

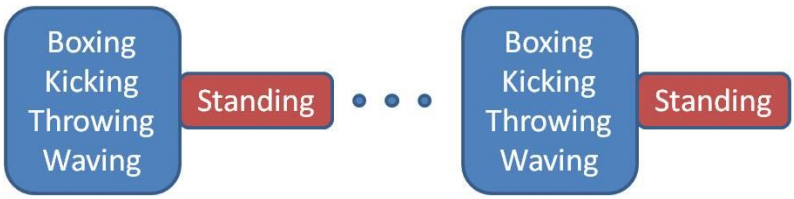


Figure 4.8: The formation of the long sequences (continuous combined actions).

In the experiment of recognition of continuous actions, with each continuous action sequence (long sequence) being concatenated from different actions performed by the same person, each long sequence is formed by 6 randomly chosen actions from {Boxing, Kicking, Throwing, Waving} short sequences (of 1-5 variable cycles) from the same person. Additionally, 6 standing actions are inserted right after each action by the same person. The formation of the long sequences is shown in Figure 4.8. Therefore, there are totally 4096 (4x4x4x4x4x4) long sequences for each person. We randomly sampled 100 out of 4096x4 long sequences for testing in each sub-experiment. Based on the designed switching graphical model, which dynamically switches CHMMs, each out

of the 100 randomly sampled long sequences is recognized as the underlying 6 different actions. Totally 600 actions are recognized in each sub-experiment.

Based on the standing action sequences being used or not, and the standing model being used no not, 7 different sub-experiments are conducted as in Table 4.8. In sub-experiment 1, no additional standing sequences are appended at each of 6 continuous actions. In sub-experiment 2, 3~5 frames of standing action are appended at each of 6 continuous actions. The overall recognition rates for sub-experiment 1 and 2 are 70.5% and 65.75%. Moreover, we consider adding standing frames, combined the one of {Boxing, Kicking, Throwing, Waving} actions, and trained together in either Boxing, Kicking Throwing, or Waving CHMMs. That is 3~5 extra standing frames or 7~10 extra standing frames individually in sub-experiment 3 or sub-experiment 4. The overall recognition rates for sub-experiments 3 and 4 are 67% and 66%. Furthermore, the extra standing frames are trained independently in a standing CHMM. In sub-experiment 5, 6 and 7, individually 30~40 standing frames, 7~10 standing frames and 3~5 standing frames are used to train a standing CHMM. With the explicitly standing CHMMs appended right after each action, the recognition rates for sub-experiment 5, 6, 7 almost achieve 100%. A conclusion is drawn that the recognition rate can reach almost 100% if the standing frames are used to train the standing CHMM, and are inserted between actions, no matter the number of the standing frames is 3~5 or 30~40. On the other hand, the insertion of standing frames only in the test sequences can degrade the recognition performance slightly, when the training is done without the insertion of standing frames.

Table 4.8: Sub-experiments for Continuous Actions Recognition

	Standing sequences	Standing CHMM	Recognition Rate
Sub-exp. 1:	0 frame	No	70.5%
Sub-exp. 2:	3~5 frames	No	65.75%
Sub-exp. 3:	3~5 frames	No, but trained in other CHMMs	67%
Sub-exp. 4:	7~10 frames	No, but trained in other CHMMs	66%
Sub-exp. 5:	30~40 frames	Yes, trained in Standing CHMM	100%
Sub-exp. 6:	7~10 frames	Yes, trained in Standing CHMM	99.55%
Sub-exp. 7:	3~5 frames	Yes, trained in Standing CHMM	100%

4.4.b IXMAS Dataset

In the experiments of continuous-action recognition in IXMAS, two sub-experiments are performed.

In the first sub-experiment, each long sequence is formed by concatenating 4 actions, with each action being randomly selected from one of the 11 action types in IXMAS dataset as shown in Figure 4.9, resulting in a total of 14641 (11x11x11x11) possible long sequences to be compared against. We randomly chose 100 out of 14641

long sequences for testing with 10-fold cross validation. The confusion matrix is shown in Table 4.9. The average recognition rate is 86.4%, which is still a satisfactory recognition rate.

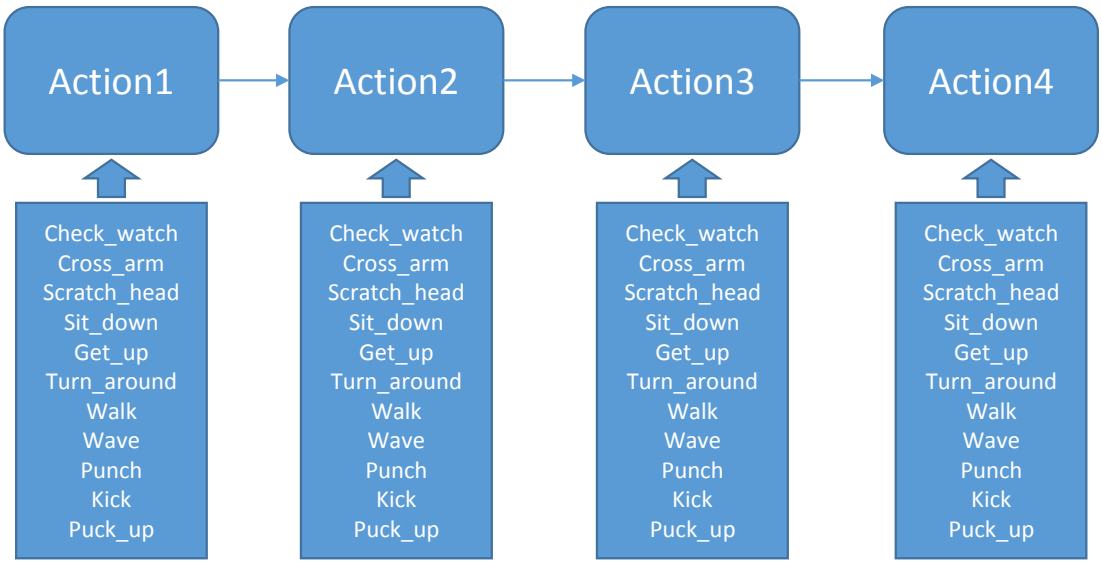


Figure 4.9: Each action is selected from 11 types of actions.

In the second sub-experiment, each long sequence is formed by concatenating 2, 3 or 4 actions, with each actions again being selected from the 11 types of actions in IXMAS dataset. Compared to the first sub-experiment, the long sequence is no long fixed to 4 actions, rather 2, 3 or 4 actions, making the recognition more difficult. Totally, there are 16093 (11x11x11x11+11x11x11+11x11) possible long sequences to be compared against. We also randomly chose 100 out of 16093 long sequences for testing with 10-fold cross validation. Because the number of types of actions is variable, Levenshtein distance (i.e., edit distance) [73] is applied for measuring the error distance between a ground-truth sequence and a recognized sequence. The Levenshtein distance is defined as

the minimum number of edits (insertion, deletion, or substitution) between two sequences. For example, the Levenshtein distance is 1 between sequence A={a,b,c} and sequence B={a,c} with only one edit (deletion of b in A). The recognition rate for the 2/3/4 actions is defined in Equation 4.4. The confusion matrix is shown in Table 4.10 with the average recognition rate, 85.1%. For comparison, the results of the recognition rates for continuous 4 actions and continuous 2/3/4 actions are shown in Table 4.11. The recognition rate of the 2/3/4 actions with 85.1%, which is 1.3% less than the recognition rate of the 4 actions with 86.4%. Therefore, it shows the recognition rate is still satisfactory even with the variable number of action types are concatenated.

Table 4.9: Confusion Matrix of the Continuous Actions Recognition in 4actions over IXMAS Dataset

4actions (%)	check_ watch	cross_ arm	scratch_ head	sit_ down	get_ up	turn_ around	walk	wave	punch	kick	pick_ up
check_watch	88.8	0.9	0.7	1.0	0.1	3.6	0.3	0.5	2.0	1.8	0.4
cross_arm	1.3	84.6	0.4	1.4	0.4	2.1	0.7	0.8	5.5	2.0	0.6
scratch_head	3.0	2.5	82.5	0.5	0.5	1.5	0.6	1.2	5.3	1.3	1.0
sit_down	2.5	1.9	0.5	86.0	0.1	1.6	0.2	1.0	1.6	2.7	1.9
get_up	1.2	0.1	0.3	0.3	96.5	0.3	0.5	0.2	0.3	0.2	0.1
turn_around	3.2	0.7	0.4	0.4	0.3	84.7	6.2	0.7	0.8	2.4	0.2
walk	0.8	0.5	0.1	0.8	0.3	5.5	88.9	0.6	0.4	1.5	0.6
wave	6.1	0.8	8.8	0.7	0.2	3.8	1.0	76.6	0.7	1.0	0.5
punch	2.7	0.8	0.5	0.5	0.3	2.1	1.6	0.7	88.0	2.3	0.5
kick	1.6	1.1	0.8	0.5	0.3	2.8	0.9	1.2	1.6	88.6	0.6
pick_up	1.3	0.3	0.6	6.2	0.4	0.9	0.9	0.2	1.8	2.0	85.3

Equation 4.4:

$$\text{Recognition rate} = 1 - \frac{\text{Sum of edit distance for all sequences}}{\text{Sum of actions for all truth sequences}}$$

Table 4.10: Confusion Matrix of the Continuous Actions Recognition in 234actions over IXMAS Dataset

234actions (%)	check_ watch	cross_ arm	scratch_ head	sit_ down	get_ up	turn_ around	walk	wave	punch	kick	pick_ up
check_watch	88.0	1.0	0.4	0.7	0.5	3.6	1.1	1.0	2.3	0.9	0.8
cross_arm	1.9	79.9	1.5	1.0	0.6	2.0	1.3	0.6	7.6	2.7	1.0
scratch_head	2.6	4.9	80.9	1.1	0.8	2.6	0.6	0.8	3.4	2.1	0.2
sit_down	2.1	1.4	0.1	83.6	0.6	2.7	1.4	0.6	0.7	3.2	3.4
get_up	0.9	0.5	0.4	0.2	94.5	0.6	0.3	0.2	0.8	0.9	0.5
turn_around	2.0	0.5	0.4	0.9	0.6	82.0	6.0	0.8	1.8	3.4	1.6
walk	1.7	2.0	0.1	0.8	0.3	6.9	85.2	0.3	0.7	1.7	0.2
wave	4.5	1.8	7.9	0.7	0.5	3.1	0.8	76.4	1.1	2.4	0.8
punch	2.2	1.7	1.3	1.3	0.1	2.3	0.7	1.0	86.4	1.6	1.5
kick	1.2	2.1	0.9	0.9	0.3	3.8	1.5	3.3	1.8	83.9	0.3
pick_up	0.7	0.4	0.1	10.0	0.5	1.3	2.1	0.5	1.3	3.2	79.8

Table 4.11: Recognition Rates for Continuous Actions Recognition

	Continuous 4 actions	Continuous 2/3/4 actions
Recognition rate	86.4	85.1

Section 5: Discussion

4.5.a Applications

The proposed method on human action recognition can be applied in 3 different scenarios including surveillance environment, entertainment environment and health care environment [3].

First, in surveillance environment, the proposed method can be used to support the security personnel to observe and to understand the activities of them, resulting in the recognizing the criminal, and detecting the suspicious activities as well. Most of the security surveillance systems are equipped with many cameras and require laborious humans monitoring on monitoring screens for video content understanding. By applying automatic human action recognition techniques to video-based surveillance systems, we can effectively reduce the workload of security staffs as well as systematically create an alert immediately when the security events are detected in order to prevent potentially dangerous situations. For example, it can improve the security environment by detecting various kinds of violent behaviors such as fighting, punching, stalking and loitering [86], [99], [112], [113], [114], [115].

Second, in the entertainment environment, the proposed method can also be used to recognize human actions in entertainment activities, such as sports [49], [50], [54], [76], dance [53], [74] and gaming [4], [9], [98], to enrich the lifestyle for community. One of the most interested leisure activities is playing video games. Our proposed real-

time 3D pose estimation system [26] can estimate 3D poses of not only the upper body part as in [9] but also the lower body part including knees and feet. A simple video game is implemented, i.e., a 3D avatar, which is real-time generated based on the derived 3D poses of the proposed system, is controlled to hit balls so as to get scores or to avoid attacks from flying balls.

Third, in health care environment, the proposed method can be used to analyze and understand the patients' actions, so as to facilitate health workers to diagnose, treat and care patients, resulting in improving the reliability of diagnosis, decreasing the working load for the medical personnel, shortening the hospital stay for patients, and improving the quality of life for patients. The proposed method can mainly be applied in daily life activity monitoring and rehabilitation. Daily life activity monitoring mainly focuses on learning and recognizing the daily life activities of seniors at home. The proposed systems are to provide seniors an opportunity to live safely, independently and comfortably. In order to accomplish this, most of proposed systems continuously capture the movements of individual senior or multiple seniors at home, automatically recognize their activities, and detect the gradual changes in baseline activities such as mobility functional disabilities and mental problems, as well as the urgent warning signs of abnormal activities such as falling down or stroke. Moreover, traditional rehabilitation systems often require patients to pay many visits to clinics for the physical therapy exercises and the scheduled evaluation until his/her full recovery of mobility function for daily activities. Such clinical visits can be avoided by using innovative rehabilitation

systems, which are home-centered and self-health care with the help of video-based activity recognition techniques. By continuously monitoring the daily activities and gaits, the early symptoms of some diseases can be timely detected so that the diagnosis and the intervention are more useful.

4.5.b Limitations

There are mainly two types of limitations in our proposed system for human action recognition.

First, some parameters are required to be predefined. The first predefined parameter is k, the number of the k clusters for all GRF vectors with the k-means algorithm. The second predefined parameter is N, the number of the hidden states for a cyclic HMM. Different scenarios and different sizes of the dataset might cause different choices of the predefined parameters, including k and N. Therefore, preprocessing might be needed to determine the predefined parameters, including k and N. In our experiments, the values of k and N are individually set as 64 and 5.

Second, the recognition rate for human actions highly depends on the accuracy of the 3D pose estimation. The GRFs vectors define the relational locations of body parts in the 3D world. If the results of the 3D pose estimation are incorrect, the relational locations of body parts will be wrong in the 3D world. This might cause ambiguity among various human actions and result in a low recognition rate. And this kind of problem could be eased if we could design a noise-tolerant feature representation.

4.5.c Potential Extensions

There are mainly two types of potential extensions in our proposed system for human action recognition.

First, the definition of GRF vectors is easily to extend to higher dimensions. The results of the 3D human pose estimation are the human model in the 3D world. We can use not only 13 joints of the 3D model, but also more joints such as the neck, the center of the chest and the center of hips. Then the dimensions of GRF can be expanded to more than 15, in order to describe the relational locations of body parts in detail. For example, the feature to define the distance between hands and hips could be useful to discriminate the action of the standing with arms akimbo from the action of pointing.

Second, the proposed system can be extended to abnormal actions recognition. By integrating the contexts and information of surrounding environments, we can differentiate abnormal or unusual activities from normal activities. For example, a deviation approach can be adopted to determine the activities are normal or abnormal. It builds a normal model as in background subtraction using given examples or previously observed data, and considers new observation as abnormal or unusual if they deviate too much from the trained model.

Chapter 5 – Conclusion and Future Work

Section 1: Conclusion

We propose a system to recognize both single and continuous combined human actions, based on 3D human pose estimation through monocular video sequences.

For 3D human pose estimation, the proposed system can successfully track 2D body parts and reconstruct the view-invariant 3D human poses. The shape, color and time continuity information of the moving objects are explored by skeletonalization, mean shift tracking and Kalman filter prediction, and a fusion scheme with extracted information is used to track the 2D body parts. The orientation of the object is also estimated by the scale change and moving trajectory of the object. Besides, the occlusions between hands, between feet and between hands and the body are effectively detected and handled.

Moreover, taking into account the human physiology and the scale and the orientation of the body, the locations of the 2D body parts can help to initialize the coarse 3D poses. Subsequently, the fine 3D poses are estimated by the downhill simplex algorithm, which searches the best 3D poses by minimizing the cost function, defined by the discrepancy between the 2D features and the 3D model.

Furthermore, the computation cost is also taken care. Without a training process, the 3D human poses are effectively and efficiently estimated through monocular video sequences with a very low computation cost.

The 3D pose estimation can be further improved in some way in our ongoing investigation. First, joints estimation, including elbows and knees, in 2D body parts tracking could increase the accuracy of the initialization of the 3D poses by fixing some DOFs. Also, the whole trunk of the limb tracking instead of the small blobs might increase the performance of the 2D body parts tracking.

After the 3D coordinates of human body are extracted from our proposed method for 3D pose estimation, the main task is to recognize both single and continuous combined human actions. With GRF conversion and k-means clustering, the 39-dimensional feature vectors are converted into 1-dimension feature vectors. The 1-dimensional feature vectors associated with one type of actions are used to train a CHMM, corresponding to the type of the action. Moreover, the switching graphical model is designed to switch CHMMs based on the long observation sequence.

The proposed system including 3D pose estimation and human action recognition is experimented on three different dataset, the self-recorded dataset, HumanEva dataset and IXMAS dataset. For 3D pose estimation, the proposed method is favorably compared with other three up-to-date methods in HumanEva dataset. For human action recognition, the proposed method is favorably compared with other three up-to-date methods in IXMAS dataset.

Section 2: Future Work

Although the recent video-based human action recognition has achieved encouraging performance, there are still some apparent performance issues that make it challenging for real-world deployment. It can be discussed as follows.

(i) Viewpoint issue remains the main challenge for human action recognition. In the real world action recognition systems, the video sequences are usually observed from arbitrary camera viewpoints; therefore, the applications require the viewpoint independent methods, i.e., the performance of systems needs to be invariant from different camera viewpoints. However, most recent algorithms are based on the constrained viewpoints, such as person has to be in front-view (i.e., face a camera) or side-view. Some effective ways to solve this problem have been proposed, such as using multiple cameras to capture different view sequences then combining them as training data or a self-adaptive calibration and viewpoint determination algorithm can be used in advance. Sophisticated viewpoint invariant algorithms for monocular videos should be the ultimate objective to overcome these issues.

(ii) Since most of moving human segmentation algorithms are still based on background subtraction, which requires a reliable background model, especially a background model which can be adaptively updated and can handle some moving background or dynamic cluttered background, as well as inconsistent lighting conditions. How to effectively deal with the dynamic cluttered background as

well as how to systematically understand the context (when, what, where, etc) should enable better and more reliable segmentation of human objects. Another important challenging research is how to handle the occlusion in terms of body-body part, human-human, human-objects, etc.

(iii) The natural human appearance can be changed due to many factors such as the walking surface conditions (e.g., hard/soft, level/stairs, etc.), clothing (e.g., long dress, short skirt, coat, hat, etc.), footgear (e.g., stockings, sandals, slippers, etc), object carrying (e.g., handbag, backpack, briefcase, etc) [110]. The change of human action appearance leads researchers to a new research direction, i.e., how to describe the activities that is less sensitive to appearance but still capture the most useful and unique characteristics of each action.

(iv) Unlike speech recognition systems, where the features are more or less unified to be the mel-frequency cepstral coefficients (MFCCs) for HMM classifiers, there are still no clear winners on the features for human action recognitions, neither the corresponding classifier designs. It can be expected that 3D viewpoint invariant modeling of human poses can be a good starting unification effort.

Finally, human action recognition tasks constitute the foundation for human behavior understanding, which requires the additional contextual information such as W5+ (who, where, what, when, why, how) [111]. The same actions may have different behavior interpretations depending on the context in which it is performed. More specifically, the “where” (place) context can provide the location information to be used

to detect abnormal behaviors. For example, lying down on the bed or a sofa is interpreted as taking a rest or sleeping, but at the irrelevant places such as floor in bathroom or kitchen, it can be interpreted as a falling or a stroke. Moreover, the “when” (time) context is also playing another important contextual role for behavior understanding. For example, a person usually watching TV after midnight can be regarded as insomnia. Another example is that a person will be detected as picking up stuffs if he/she squats and stands up soon. But if he/she squats for a longer period, there might be a motion difficulty due to osteoarthritis or senility. Furthermore, the number of repetitions of an action is also a good hint. For example, eating too many times or too little a day can be an early symptom of depression. The interaction between people or between person and objects is also a good context to identify meaning of the activity. For example, if a person is punching a punch-bag, he might be doing exercise. But if he is punching the wall, it can indicate anger or mental disorder.

Bibliography

[1] Thomas B. Moeslund, and Erik Granum, "A survey of computer vision-based human motion capture," Computer Vision and Image Understanding, 2001.

[2] Thomas B. Moeslund, Adrian Hilton, and Volker Krüger, "A survey of advances in vision-based human motion capture and analysis," Computer Vision and Image Understanding, Volume 104, Issues 2-3, November-December 2006.

[3] Shian-Ru Ke, Hoang L.U. Thuc, Yong-Jin Lee, Jenq-Neng Hwang, Jang-Hee Yoo, Kyoung-Ho Choi, "A Review on Video-Based Human Activity Recognition," Computers, vol. 2, no. 2, pp. 88-131, 2013.

[4] C. R.Wren, A. Azarbayejani, T. Darrell, and A. P. Pentland, "Pfinder: real-time tracking of the human body," Pattern Analysis and Machine Intelligence, IEEE Transactions on, vol.19, no.7, pp.780-785, Jul 1997.

[5] A. Nakazawa, H. Kato, and S. Inokuchi, "Human tracking using distributed vision systems," in International Conference on Pattern Recognition,1998.

[6] S. Iwasawa, K. Ebihara, J. Ohya, and S. Morishima, "Real-time estimation of human body posture from monocular thermal images," in Conference on Computer Vision and Pattern Recognition, 1997.

[7] M. K. Leung and Y. H. Yang, "First sight: A human body outline labeling system," Pattern Analysis and Machine Intelligence, IEEE Transactions on, 1995.

[8] D. M. Gavrila and L. S. Davis, "3-D model-based tracking of humans in action: a multi-view approach," Computer Vision and Pattern Recognition, 1996.

[9] Feifei Huo, E. Hendriks, P. Paclik, and A.H.J. Oomes, "Markerless human motion capture and pose recognition," 10th Workshop on Image Analysis for Multimedia Interactive Services, 2009. WIAMIS '09.

[10] Iat-Fai Leong, Jing-Jing Fang, and Ming-June Tsai, "Automatic body feature extraction from a marker-less scanned human body," Computer-Aided Design, Volume 39, Issue 7, July 2007.

[11] Mykhaylo Andriluka, Stefan Roth, and Bernt Schiele, "Monocular 3D pose estimation and tracking by detection," IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2010.

[12] Ankur Agarwal and Bill Triggs, "Recovering 3D Human Pose from Monocular Images," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 28, no. 1, pp. 44-58, Jan. 2006.

[13] S. Sedai, M. Bennamoun, and D. Huynh, "Context-Based Appearance Descriptor for 3D Human Pose Estimation from Monocular Images," dicta, pp.484-491, 2009 Digital Image Computing: Techniques and Applications, 2009.

[14] T. B. Moeslund and E. Granum, "3D human pose estimation using 2D-data and an alternative phase space representation," in Workshop on Human Modeling, Analysis and Synthesis at CVPR, Hilton Head Island, SC, June 2000.

[15] Mun Wai Lee and Isaac Cohen, "A Model-Based Approach for Estimating Human 3D Poses in Static Images," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 28, no. 6, pp. 905-916, June 2006.

[16] Mun Wai Lee and Ram Nevatia, "Body Part Detection for Human Pose Estimation and Tracking," IEEE Workshop on Motion and Video Computing, 2007. WMVC '07.

[17] Mun Wai Lee and Ramakant Nevatia, "Human Pose Tracking in Monocular Sequence Using Multilevel Structured Models," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 31, no. 1, pp. 27-38, Jan. 2009.

[18] Deva Ramanan, David A. Forsyth and Andrew Zisserman, "Tracking People by Learning Their Appearance," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 29, no. 1, pp. 65-81, Jan. 2007.

[19] M. Isard and A. Blake, "CONDENSATION-Conditional density propagation for visual tracking," in International Journal of Computer Vision, vol. 29, pp. 5-28, January 1998.

[20] Z. Tu and S.C. Zhu, "Image Segmentation by Data-Driven Markov Chain Monte Carlo," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 24, no. 5, pp. 657-673, May 2002

[21] M. Tipping, "The Relevance Vector Machine," In Neural Information Processing Systems, 2000.

[22] M. Tipping, "Sparse Bayesian Learning and the Relevance Vector Machine," J. Machine Learning Research, 1:211–244, 2001.

[23] G. Rogez, J.J. Guerrero, and C. Orrite, "View-invariant human feature extraction for video-surveillance applications," pp.324-329, IEEE AVSS, 2007.

[24] G. Rogez, C. Orrite, J. Mart´ınez, and J. E. Herrero, "Probabilistic Spatio-Temporal 2D-Model for Pedestrian Motion Analysis in Monocular Sequences," Conf. on Articulated Motion and Deformable Objects, pp.175-184, July 2006.

[25] W. Press, S. Teykolsky, W. Vetterling, and B. Flannery, “Numerical recipes in C++: the art of scientific computing,” Pearson Education, 1992.

[26] Shian-Ru Ke, LiangJia Zhu, Jenq-Neng Hwang, Hung-I Pai, Kung-Ming Lan, and Chih-Pin Liao, "Real-Time 3D Human Pose Estimation from Monocular View with Applications to Event Detection and Video Gaming," 7th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS), pp.489-496, 2010.

[27] Shian-Ru Ke, Jenq-Neng Hwang, Kung-Ming Lan, and Shen-Zheng Wang, “View-Invariant 3D Human Body Pose Reconstruction using a Monocular Video Camera,” IEEE International Conference on Distributed Smart Cameras (ICDSC), 2011.

[28] Robert J. Holt, Thomas S. Huang, Arun N. Netravali, and Richard J. Qian, “Determining articulated motion from perspective views: A decomposition approach,” Pattern Recognition, Volume 30, Issue 9, September 1997.

[29] Dorin Comaniciu, Visvanathan Ramesh, and Peter Meer, “Real-time tracking of non-rigid objects using mean shift”, CVPR, vol. 2, pp.2142, 2000.

[30] Dorin Comaniciu, Peter Meer, “Mean Shift: a robust approach toward feature space analysis”, Pattern Analysis and Machine Intelligence, IEEE Transactions on, 2002.

[31] Dorin Comaniciu, Visvanathan Ramesh, and Peter Meer, “Kernel-based object tracking,” Pattern Analysis and Machine Intelligence, IEEE Transactions on , vol.25, no.5, pp. 564- 577, May 2003.

[32] Saleh A. Al-Shehri, “A simple and novel method for skin detection and face locating and tracking”, Lecture Notes in Computer Science, vol. 3101, pp. 1-8, Berlin, 2004.

[33] J.F. Wang, P.L. Chen, C.P. Liao, Y.Y. Tsai, J. Huang, K.S. Wang, “Real time depth sensing with automatic determined depth range for human computer interactive applications”, IPCV, Las Vegas, NV, 12-15 July, 2010.

[34] T. Horprasert, D. Harwood, and L.S. Davis, “A Statistical Approach for Real-time Robust Background Subtraction and Shadow Detection”, IEEE ICCV, Frame-Rate Workshop, pp.1-19, 1999.

[35] John Canny, “A Computational Approach to Edge Detection,” Pattern Analysis and Machine Intelligence, IEEE Transactions on, vol.PAMI-8, no.6, pp.679-698, Nov. 1986.

[36] Greg Welch and Gary Bishop, "An Introduction to the Kalman Filter", Technical Report TR 95-041, Department of Computer Science, University of North Carolina at Chapel Hill, 1995.

[37] H. Sundar, D. Silver, N. Gagvani, S. Dickinson, and D. Silver Ý, "Skeleton Based Shape Matching and Retrieval," SMI, 2003.

[38] A. Rosenfeld and J. Pfaltz, "Distance Functions on Digital Pictures," PR, 1(1):33–61, 1968.

[39] N. Gagvani and D. Silver, "Parameter Controlled Volume Thinning," Graphical Models and Image Processing, 61(3):149–164, May 1999.

[40] T. Kanungo, D.M. Mount, N.S. Netanyahu, C.D. Piatko, R. Silverman, and A.Y. Wu, "An efficient k-means clustering algorithm: analysis and implementation," PAMI, 2002.

[41] T. Kailath, "The Divergence and Bhattacharyya Distance Measures in Signal Selection," IEEE Trans. Comm. Technology, pp.52-60, 1967.

[42] L. Sigal, A. Balan and M. J. Black, "HumanEva: Synchronized Video and Motion Capture Dataset and Baseline Algorithm for Evaluation of Articulated Human Motion," International Journal of Computer Vision, Vol. 87 (1-2), 2010.

[43] G.D. Hager, M. Dewan, and C.V. Stewart, "Multiple kernel tracking with SSD," Computer Vision and Pattern Recognition, 2004. CVPR 2004. Proceedings of the 2004 IEEE Computer Society Conference on, 2004.

[44] Zhimin Fan, Ying Wu, and Ming Yang, "Multiple collaborative kernel tracking," Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on, 20-25 June 2005.

[45] Jan A. Snyman, "Practical Mathematical Optimization," New York, Springer, pp. 81-88, 2005.

[46] Shian-Ru Ke, Hoang Le Uyen Thuc, Jenq-Neng Hwang, Jang-Hee Yoo, Kyoung-Ho Choi, "Human Action Recognition based on 3D Human Modeling and Cyclic HMMs," ETRI journal, 2013. (Accepted)

[47] Shian-Ru Ke, Jenq-Neng Hwang, Maryam Fazel, Shen-Zheng Wang, Hung-I Pai "Constrained Multiple Kernel Tracking for Human Limbs," IEEE International Symposium on Circuits and Systems, Seoul, Korea, May 20-23, 2012.

[48] Meinard Muller, Tido Roder, Michael Clausen, “Efficient content-based retrieval of motion capture data,” Association for Computing Machinery, Inc., pp. 677-685, 2005.

[49] Ying Luo, Tzong-Der Wu, Jenq-Neng Hwang, “Object-based analysis and interpretation of human motion in sports video sequences by dynamic bayesian networks,” Computer Vision and Image Understanding, Volume 92, Issues 2-3, November-December 2003.

[50] J. Yamato, J. Ohya, and K.Ishii, “Recognizing human action in time-sequential images using hidden Markov model,” in CVPR, pp. 379-385, Champaign, IL, June 1992.

[51] Ronald Poppe, “A survey on vision-based human action recognition,” Image and Vision Computing, Volume 28, Issue 6, June 2010.

[52] C. Schuldt, I. Laptev, and B. Caputo, “Recognizing human actions: a local SVM approach,” Proceedings of the 17th International Conference on Pattern Recognition, 2004. ICPR 2004.

[53] Moshe Blank, Lena Gorelick, Eli Shechtman, Michal Irani, and Ronen Basri, “Actions as Space-Time Shapes,” tenth IEEE International Conference on Computer Vision (ICCV'05) Volume 2, 2005.

[54] Yan Ke, R. Sukthankar, and M. Hebert, “Spatio-temporal Shape and Flow Correlation for Action Recognition,” IEEE Conference on Computer Vision and Pattern Recognition, 2007. CVPR '07.

[55] P. Dollar, V. Rabaud, G. Cottrell, and S. Belongie, “Behavior recognition via sparse spatio-temporal features,” vspets, pp.65-72, 2005 IEEE International Workshop on Visual Surveillance and Performance Evaluation of Tracking and Surveillance, 2005.

[56] Ashok Veeraraghavan, Amit K. Roy-Chowdhury, and Rama Chellappa, “Matching Shape Sequences in Video with Applications in Human Movement Analysis,” IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 27, no. 12, pp. 1896-1909, Dec. 2005.

[57] Somayeh Danafar, Alessandro Giusti, and Jürgen Schmidhuber, “Novel Kernel-Based Recognizers of Human Actions,” EURASIP Journal on Advances in Signal Processing, vol. 2010.

[58] Cheng-Hsien Lin, Fu-Song Hsu, and Wei-Yang Lin, “Recognizing Human Actions Using NWFE-Based Histogram Vectors,” EURASIP Journal on Advances in Signal Processing, vol. 2010.

[59] D. Weinland, E. Boyer, R. Ronfard, “Action Recognition from Arbitrary Views using 3D Exemplars,” IEEE 11th International Conference on Computer Vision (ICCV), pp.1-7, 14-21 Oct. 2007.

[60] G. Rogez, C. Orrite, and J. Martínez, “A spatio-temporal 2D-models framework for human pose recovery in monocular sequences,” Pattern Recognition, vol. 41, issue 9, September 2008.

[61] G. Rogez, J. Rihan, S. Ramalingam, C. Orrite, and P.H.S. Torr, “Randomized trees for human pose detection,” IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp.1-8, 23-28 June 2008.

[62] Hoang Le Uyen Thuc, Shian-Ru Ke, Jenq-Neng Hwang, Jang-Hee Yoo, and Kyoung-Ho Choi, “Human Action Recognition based on 3D Body Modeling from Monocular Videos,” Frontiers of Computer Vision Workshop, pp. 6-13, 2012.

[63] Hoang Le Uyen Thuc, Pham Van Tuan, and Jenq-Neng Hwang, “An Effective 3D Geometric Relational Feature Descriptor for Human Action Recognition,” IEEE RIVF, Vietnam, Feb.27-Mar.1, 2012.

[64] John A. Hartigan and Manchek A. Wong. "Algorithm AS 136: A k-means clustering algorithm." Applied statistics, pp. 100-108, 1979.

[65] Hoang Le Uyen Thuc, Shian-Ru Ke, Jenq-Neng Hwang, Pham Van Tuan, Truong Ngoc Chau, “Quasi-Periodic Action Recognition from Monocular Videos via 3D Human Models and Cyclic HMMs,” IEEE International Conference on ATC, Vietnam, Oct 10-12, 2012.

[66] L. Rabiner and B. Juang, “An introduction to hidden Markov models,” IEEE ASSP Magazine, vol.3, no.1, pp.4-16, Jan. 1986.

[67] Huang, Xuedong, Alejandro Acero, and Hsiao-Wuen Hon, “Spoken language processing,” Vol. 15. New Jersey: Prentice Hall PTR, 2001.

[68] N.N. Bitar and Carol Y. Espy-Wilson, “Knowledge-based parameters for HMM speech recognition,” IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP), vol.1, pp.29-32 vol. 1, 7-10 May 1996.

[69] Jeff Bilmes and Geoff Zweig, “The Graphical Models Toolkit: An Open Source Software System for Speech and Time-Series Processing,” IEEE ICASSP, 2002.

[70] I.N. Junejo, E. Dexter, I. Laptev, P. Perez, “View-Independent Action Recognition from Temporal Self-Similarities,” IEEE Transactions on PAMI, vol. 33, no. 1, pp. 172-185, 2011.

[71] I. Laptev, “On Space-Time Interest Points,” Int’l J. Computer Vision, vol. 64, nos. 2/3, pp. 107-123, 2005.

[72] I. Laptev, M. Marszałek, C. Schmid, and B. Rozenfeld, “Learning Realistic Human Actions from Movies,” Proc. IEEE CS Conf. Computer Vision and Pattern Recognition, 2008.

[73] Robert A. Wagner and Michael J. Fischer, “The string-to-string correction problem,” Journal of the ACM (JACM), vol. 21, no. 1, pp. 168-173, 1974.

[74] Eli Shechtman and Michal Irani, “Space-time behavior based correlation,” In Proc. CVPR, 2005.

[75] S. Kumari, S. and S.K. Mitra, “Human Action Recognition Using DFT,” Computer Vision, Pattern Recognition, Image Processing and Graphics (NCVPRIPG), 2011 Third National Conference on , vol., no., pp.239-242, 15-17 Dec. 2011.

[76] Wei-Lwun Lu, J.J. Little, “Simultaneous Tracking and Action Recognition using the PCA-HOG Descriptor,” Computer and Robot Vision, 2006. The 3rd Canadian Conference on, vol., no., pp. 6, 07-09 June 2006.

[77] Xin Lu, Qiong Liu, S. Oe, “Recognizing non-rigid human actions using joints tracking in space-time,” Information Technology: Coding and Computing, 2004. Proceedings. ITCC 2004. International Conference on, vol.1, no., pp. 620- 624 Vol.1, 5-7 April 2004.

[78] Paul Scovanner, Saad Ali, Mubarak Shah, “A 3-dimensional SIFT descriptor and its application to action recognition,” in Proceedings of the International Conference on Multimedia (MultiMedia’07), Augsburg, Germany, September 2007, pp. 357–360.

[79] K. Schindler and L. V. Gool, “Action snippets: How many frames does human action recognition require?” In Proc. CVPR (1-8), 2008.

[80] S. Danafar and N. Gheissari, “Action recognition for surveillance application using optic flow and SVM,” In Proc. ACCV, 2007.

[81] D.G. Lowe, “Object recognition from local scale-invariant features,” Computer Vision, 1999. The Proceedings of the Seventh IEEE International Conference on, vol.2, no., pp.1150-1157 vol.2, 1999.

[82] D.G. Lowe, “Distinctive Image Features from Scale-Invariant Keypoints”, IJCV, 2004.

[83] N. Dalal and B. Triggs, “Histograms of oriented gradients for human detection,” Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on, vol.1, no., pp.886-893 vol. 1, 25-25 June 2005.

[84] A. Dargazany and M. Nicolescu, “Human Body Parts Tracking Using Torso Tracking: Applications to Activity Recognition,” 2012 Ninth International Conference on Information Technology: New Generations (ITNG), pp.646-651, 16-18 April 2012.

[85] L.W. Campbell, D.A. Becker, A. Azarbayejani, A.F. Bobick, A. Pentland, “Invariant features for 3-D gesture recognition,” Proceedings of the Second International Conference on Automatic Face and Gesture Recognition, pp.157-162, 14-16 Oct. 1996.

[86] S. Sempena, N.U. Maulidevi, P.R. Aryan, “Human action recognition using Dynamic Time Warping,” Electrical Engineering and Informatics (ICEEI), 2011 International Conference on , vol., no., pp.1-5, 17-19 July 2011.

[87] Thi V. Duong, Hung H. Bui, Dinh Q. Phung, and Svetha Venkatesh, “Activity Recognition and Abnormality Detection with the Switching Hidden Semi-Markov Model,” Proc. IEEE CVPR, vol. 1, pp. 838-845, June 2005.

[88] M. Brand, N. Oliver, A. Pentland, “Coupled hidden Markov models for complex action recognition,” Computer Vision and Pattern Recognition, 1997. Proceedings, 1997 IEEE Computer Society Conference on, vol., no., pp.994-999, 17-19 Jun 1997.

[89] P. Natarajan, R. Nevatia, “Online, Real-time Tracking and Recognition of Human Actions,” Motion and video Computing, 2008. WMVC 2008. IEEE Workshop on, vol., no., pp.1-8, 8-9 Jan. 2008.

[90] Youtian Du, Feng Chen, Wenli Xu, “Human Interaction Representation and Recognition Through Motion Decomposition,” Signal Processing Letters, IEEE , vol.14, no.12, pp.952-955, Dec. 2007.

[91] K. P. Murphy, “Dynamic Bayesian Networks: Representation, Inference and Learning,” University of California, 2002.

[92] V. N. Vapnik, “Statistical Learning Theory,” Wiley, 1998.

[93] V. N. Vapnik, S. E. Golowich, and A. J. Smola, “Support vector method for function approximation, regression estimation and signal processing,” In M. C. Mozer, M. I. Jordan, and T. Petsche, editors, Advances in Neural Information Processing Systems 9. MIT Press, 1997.

[94] Homa Foroughi, Aabed Naseri, Alizera Saberi, and Hadi Sadoghi Yazdi, "An Eigenspace-Based Approach for Human Fall Detection Using Integrated Time Motion Image and Neural Network," Proc. IEEE Int. Conf. on Signal Processing, pp. 1499-1503, 2008.

[95] M.K. Fiaz, B. Ijaz, "Vision based human activity tracking using artificial neural networks," Intelligent and Advanced Systems (ICIAS), 2010 International Conference on , vol., no., pp.1-5, 15-17 June 2010.

[96] A. K. Jain, R. P.W. Duin, and Jianchang Mao, "Statistical pattern recognition: a review," Pattern Analysis and Machine Intelligence, IEEE Transactions on, vol.22, no.1, pp.4-37, Jan 2000.

[97] Andrew Y. Ng and Michael I. Jordan, "On Discriminative vs. Generative classifiers: A comparison of logistic regression and naive Bayes," Advances in Neural Information Processing System (NIPS), 2002.

[98] Robert Bodor, Bennett Jackson, and Nikolaos Papanikolopoulos. "Vision-based human tracking and activity recognition." In Proc. of the 11th Mediterranean Conf. on Control and Automation, vol. 1. 2003.

[99] PC Ribeiro, J Santos-Victor, "Human Activity Recognition from Video: modeling, feature selection and classification architecture," Human Activity Recognition and Modelling, 2005.

[100] C. Stauffer, W.E.L. Grimson, "Learning patterns of activity using real-time tracking," Pattern Analysis and Machine Intelligence, IEEE Transactions on , vol.22, no.8, pp.747-757, Aug 2000.

[101] J. Ben-Arie, Zhiqian Wang, P. Pandit, S. Rajaram,"Human activity recognition using multidimensional indexing," Pattern Analysis and Machine Intelligence, IEEE Transactions on , vol.24, no.8, pp. 1091- 1104, Aug 2002.

[102] Wonjun Kim, Chanho Jung, and Changick Kim, "Spatiotemporal Saliency Detection and Its Applications in Static and Dynamic Scenes," IEEE Transactions on CSVT, vol.21, no.4, pp.446-456, April 2011.

[103] B. D. Lucas and T. Kanade, "An iterative image registration technique with an application to stereo vision," International Joint Conf. on Artificial Intelligence, pp. 674-679, 1981.

[104] Jianbo Shi and C. Tomasi, "Good features to track," IEEE Conference on CVPR, pp.593-600, Jun 1994.

[105] Hirokatsu Kataoka and Yoshimitsu Aoki, “Symmetrical Judgment and Improvement of CoHOG Feature Descriptor for Pedestrian Detection,” IAPR Conference on Machine Vision Applications, Nara, Japan, P.536-539, June 2011.

[106] W. Gilks, S. Richardson, and D. Spiegelhalter, “Markov Chain Monte Carlo in Practice,” Chapman and Hall, 1996.

[107] Song-Chun Zhu, Rong Zhang, and Zhuowen Tu, “Integrating bottom-up/top-down for object recognition by data driven Markov chain Monte Carlo,” IEEE Conference on CVPR, vol. 1, pp.738-745, 2000.

[108] L. Rabiner and B. Juang, “Fundamentals of speech recognition,” Prentice Hall, 1993.

[109] Umeda, M. "Recognition of multi-font printed Chinese characters." In Proc. 6th ICPR, pp. 793-796, 1982.

[110] Davrondzhon Gafurov, “A Survey of Biometric Gait Recognition: Approaches, Security and Challenges,” Presented at the NIK-2007 Conference, 2007.

[111] Maja Pantic, Alex Pentland, Anton Nijholt and Thomas S. Huang, “Human Computing and Machine Understanding of Human Behavior: A Survey,” Artificial Intelligence for Human Computing, 2007.

[112] Wei Niu, Jiao Long, Dan Han, and Yuan-Fang Wang. "Human activity detection and recognition for video surveillance." In Multimedia and Expo, 2004. ICME'04. 2004 IEEE International Conference on, vol. 1, pp. 719-722. IEEE, 2004.

[113] Nicolas Moënne-Loccoz, François Brémond, and Monique Thonnat. "Recurrent Bayesian network for the recognition of human behaviors from video." In Computer Vision Systems, pp. 68-77. Springer Berlin Heidelberg, 2003.

[114] Weiyao Lin, Ming-Ting Sun, Radha Poovandran, and Zhengyou Zhang. "Human activity recognition for video surveillance." In Circuits and Systems, 2008. ISCAS 2008. IEEE International Symposium on, pp. 2737-2740. IEEE, 2008.

[115] Thi Thi Zin, Pyke Tin, Takashi Toriu, and Hiromitsu Hama. "A Markov random walk model for loitering people detection." In Intelligent Information Hiding and Multimedia Signal Processing (IIH-MSP), 2010 Sixth International Conference on, pp. 680-683. IEEE, 2010.

Vita

Shian-Ru Ke was born in Taiwan. He received his Bachelor of Science degree in the Department of Civil Engineering from National Central University in 2003. After that, he joined Communications and Multimedia Laboratory at National Taiwan University, and obtained his Master of Science degree in the Department of Computer Science and Information Engineering from National Taiwan University in 2005. In 2008, he joined Information Processing Laboratory at University of Washington and pursued his Ph.D. degree. He got a Master of Science degree in the Department of Electrical Engineering from University of Washington in 2010. In 2014, he earned a Doctor of Philosophy degree in the Department of Electrical Engineering at University of Washington. His research interests are computer vision, machine learning, and video/image processing.