HUMAN BEHAVIOR UNDERSTANDING THROUGH 3D DATA

A THESIS SUBMITTED TO
THE GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES
OF
MIDDLE EAST TECHNICAL UNIVERSITY

BY

ERKUT AKDAĞ

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR
THE DEGREE OF MASTER OF SCIENCE
IN
ELECTRICAL AND ELECTRONICS ENGINEERING

DECEMBER 2015

Approval of the thesis:

HUMAN BEHAVIOR UNDERSTANDING THROUGH 3D DATA

submitted by **ERKUT AKDAĞ** in partial fulfillment of the requirements for the degree of **Master of Science in Electrical and Electronics Engineering Department, Middle East Technical University** by,

Prof. Dr. M. Gülbin Dural Ünver
Dean, Graduate School of **Natural and Applied Sciences** _____

Prof. Dr. Gönül Turhan Sayan
Head of Department, **Electrical and Electronics Engineering** _____

Prof. Dr. Uğur Halıcı
Supervisor, **Electrical and Electronics Eng. Dept., METU** _____

Examining Committee Members:

Prof. Dr. Gözde Bozdağı Akar
Electrical and Electronics Engineering Dept., METU _____

Prof. Dr. Uğur Halıcı
Electrical and Electronics Engineering Dept., METU _____

Prof. Dr. Ferda Nur Alpaslan
Computer Engineering Dept., METU _____

Assoc. Prof. Dr. İlkay Ulusoy Parnas
Electrical and Electronics Engineering Dept., METU _____

Assist. Prof. Dr. Tolga İnan
Electrical and Electronics Engineering Dept., TEDU _____

Date: _____

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

Name, Last name : Erkut AKDAĞ

Signature :

ABSTRACT

HUMAN BEHAVIOR UNDERSTANDING THROUGH 3D DATA

Akdağ, Erkut

M.S., Department of Electrical and Electronics Engineering
Supervisor: Prof. Dr. Uğur Halıcı

December 2015, 69 pages

In the human action recognition area, so far 2D action recognition has been studied extensively. Recently some studies, understanding human actions in 3D is emerging due to development of devices collecting 3D data. In this thesis, a new human behavior recognition method, that we call silhouette flows, is proposed for 3D data sequences of depth map. The method proposed in this thesis constitutes two steps, which are the feature extraction and classification. In feature extraction part, motion features are extracted from the 3D binary depth data in order to discern possibilities for action within the environment. For this purpose, the 3D depth data is projected on to cartesian planes in order to obtain silhouettes in frontal, top and side views and then optical flow vector fields on these planes over each frame of the video are computed. After finding these flow vectors, averages are prepared according to the motion vector values separately for negative and positive values for each frame of each plane. In order to recognize various human behaviors, each frame in video is divided into some meaningful blocks. According to the significant motion blocks, the final motion feature vector is obtained. Then, this motion feature vector is given to the SVM classification system and the results are investigated. All experiments are conducted on depth map data “MSR Action3D Dataset”. This dataset includes

twenty human actions depth map sequences recorded with Microsoft Kinect depth sensors for ten different people. The experimental results are quite successful and the proposed method outperformed in some test the other methods existing in literature for the same data.
Keywords: Optical flow, SVM classifier, action recognition, 3D data, depth map.

ÖZ

3 BOYUTLU VERİLERDEN İNSAN DAVRANIŞI ANLAMA

Akdağ, Erkut

Yüksek Lisans, Elektrik ve Elektronik Mühendisliği Bölümü
Tez Yöneticisi: Prof. Dr. Uğur Halıcı

Aralık 2015, 69 sayfa

İnsan hareketi tanıma alanında, bugüne kadar 2 boyutlu hareket tanıma yoğun çalışılmıştır. Son zamanlarda, 3 boyutlu veri toplayan cihazların geliştirilmesiyle, 3 boyutlu insan hareketlerini anlama çalışmaları ortaya çıkmaktadır. Bu tezde, 3 boyutlu derinlik haritası veri dizileri için siluet akı olarak adlandırdığımız yeni bir insan davranışı tanıma yöntemi önerilmektedir. Bu tezde önerilen yöntem, özellik çıkarma ve sınıflandırma olmak üzere iki aşamadan oluşmaktadır. Özellik çıkarma bölümünde, hareket özellikleri 3 boyutlu ikili derinlik verilerinden, eylemin çevreyle etkileşimini fark etme amacıyla elde edilir. Bu amaçla, karşıdan görünüş, yukarıdan görünüş ve yandan görünüş siluetlerini elde etmek için 3D derinlik verileri kartezyen düzlemleri üzerine yansıtılır ve daha sonra bu düzlemlerde her videonun her çerçevesi içinde optik akış vektör alanları hesaplanır. Bu akış vektörleri bulunduktan sonra, her düzlemin her çerçevesi için ayrı ayrı pozitif ve negatif hareket vektör değerlerine göre ortalamalar hazırlanır. Çeşitli insan davranışlarını tanımak amacıyla, videoda her çerçeve bazı anlamlı bloklara ayrılır. Bu anlamlı hareket bloklarına göre, son hareket özelliği vektörü elde edilir. Sonra, bu hareket özelliği vektörü SVM sınıflandırma sistemine verilir ve sonuçlar incelenir. Tüm deneyler derinlik harita verileri "MSR Action3D Veri Seti" üzerinde yürütülür. Bu veri kümesi on farklı insan için Microsoft Kinect derinlik sensörleri

ile kaydedilen yirmi tane insan hareketinin derinlik harita dizilerini içerir. Deneysel sonuçlar oldukça başarılıdır ve önerilen yöntem bazı testlerde aynı veri seti için literatürde mevcut diğer yöntemleri geride bırakmıştır.
Anahtar Kelimeler: optik akı, destekçi vektör makinesi sınıflandırıcı, hareket tanıma, 3 boyutlu veri, derinlik haritası

To my mother…

ACKNOWLEDGEMENTS

It is a pleasure for me to express my sincere gratitude to my thesis supervisor Prof. Dr. Uğur Halıcı for his belief, patience, encouragement and guidance throughout the study.

I would also like to thank my company ASELSAN Inc. for supporting this thesis work and encouraging us to make scientific studies for the problems that we face off during the work.

I am also grateful to my family for their love, trust and support throughout my life.

TABLE OF CONTENTS

LIST OF FIGURES

FIGURES

LIST OF TABLES

TABLES

LIST OF ABBREVIATIONS

2-D : Two Dimensional

3-D : Three Dimensional

MSR : Microsoft Research

4-D : Four Dimensional

STOP : Space Time Occupancy Pattern

ROP : Random Occupancy Pattern

DMM : Depth Motion Map

HOG : Histogram of Gradients

BMI : Binary Motion Image

CNN : Convolutional Neural Network

HOJ3D : Histogram of 3D Joint Locations

LDA : Linear Discriminant Analysis

HMM : Hidden Markov Model

PCA : Principle Component Analysis

NBNN : Naive Bayes Nearest Neighbour

LOP : Local Occupancy Pattern

SVM : Support Vector Machine

FEM : Feature Extraction Method

FEM1 : Feature Extraction Method1

FEM2 : Feature Extraction Method2

FEM3 : Feature Extraction Method3

AoS : Average of Silhouette

AoSD : Average of Silhouette Difference

AoSF : Average of Silhouette Flow

SD : Silhouette Difference

LIBSVM : A Library for Support Vector Machines

CHAPTER 1

INTRODUCTION

1.1.Motivation

Human behavior understanding is an active research topic in computer vision. Its development began in the early 1980's. Rich literature about human action recognition can be found in a wide range of fields including computer vision, pattern recognition, machine learning and signal processing. While researches have mainly focused on understanding human behaviors from a single camera in the past decade, human action recognition researches are progressed on depth cameras instead of single camera in recent years.

During the recent years a wide range of applications using human activity recognition has been introduced such as assisted living, advanced human-computer interaction, gesture based interactive games, movies, 3D TV and animation, intelligent driver assistance systems, video surveillance, sport motion analysis and video annotation.

Instead of 2D dataset, 3D depth map dataset sequences used in experiments makes the task of action segmentation easier. It gives more information for human action recognition compared to 2D data. The researches on 3D dataset have recently started to emerge and the research on human action recognition using 3D data is quiet limited. These are the main reasons of motivation for this thesis.

1.2.Available Datasets for 3D Action Recognition

In recent years, the technology of action recognition has entered a new phase with release of the low-cost depth cameras like Microsoft Kinect [1]. These depth cameras provide 3D depth data as well as color image sequences in real time, which makes it possible to explore the fundamental solution for traditional problems in human action classification. There are a number of limited datasets containing depth information for human action recognition.

The publicly available datasets "DailyActivity3D" and "MSR Action3D" datasets for 3D action recognition are both constructed by Microsoft Research Group (MSR) [2]. Although there are some other datasets such as (RGBD-HuDaAct[3], CAD-60[4], UTKinect Action[5]), the two datasets by MSR mentioned above are the ones mostly used and cited in the literature. These datasets are explained in more detail in the following with emphasize on the MSR Action3D dataset since it is the one used in this thesis study.

DailyActivity3D dataset [2] is a daily activity dataset captured by a Kinect device. In this dataset, the entire captured scene is kept. There are 16 activity types such as drink, eat, read book, call cell phone etc. Totally ten subjects perform an activity twice in standing position and in sitting position and there is a sofa in all scenes. This dataset is designed to cover human's daily activities in the living room. Most of the activities involve the humans-object interactions. Thus this dataset is challenging.

The MSR Action3D dataset [6] is an action dataset of depth sequences captured by a depth camera. This dataset contains twenty actions: high arm wave, horizontal arm wave, hammer, hand catch, forward punch, high throw, draw x, draw tick, draw circle, hand clap, two hand wave, side boxing, bend, forward kick, side kick, jogging, tennis swing, tennis serve, golf swing, pick up & throw. Each action was performed by ten subjects for two or three times. The number of frames is changing

depending on action and it also varies for different recordings of the same action. The frame rate is 15 frames per second and depth map size is 320×240 pixels. Altogether, the dataset has 23797 frames of depth map for 402 action samples, resulting in about 60 frames per sequence on the average. The actions in the dataset were chosen to cover various movements of arms, legs, torso and their combinations, and the subjects were advised to use their right arm or leg if an action is performed by a single arm or leg. Although the background of this dataset is clean, this dataset is challenging because many of the actions in the dataset are highly similar to each other.

Figure 1.1-Figure 1.3 are showing ten depth sequences of tennis serve, pickup & throw, and golf swing actions used in dataset.

Figure 1.1: Examples of the sequences of depth files for tennis serve action

Figure 1.2: Examples of the sequences of depth files for pickup & throw action

Figure 1.3: Examples of the sequences of depth files for golf swing action

1.3.Objective and Scope of the Thesis

Recent studies taking advantage of 3D information have been showing advanced performances compared to the traditional 2D video-based researches [1, 7 and 8]. This thesis is focused on the human action recognition by utilizing sequences of only depth maps without any additional information. The depth map provides 3D shape information of the body. With the help of this information, distinguishing the actions can be more efficient rather than the 2D silhouettes captured from a single view. Although depth maps have some benefits, there is also drawback of them that is; the amount of the data to be processed is increased by using depth maps. Effective and efficient use of depth information is a key to develop a computationally efficient algorithm for human action recognition based on the sequences of depth maps. In this study the MSR Action3D dataset is considered.

1.4.Contribution of the Thesis

A new human action recognition method to operate on depth maps is proposed. This method, called, silhouette flows, has two basic stages which are feature extraction and classification. In the feature extraction part, which is the main contribution of thesis, firstly the silhouettes in front, top and side views are extracted by projecting the depth information on the related Cartesian planes, which are x-y, x-z and y-z planes respectively. While it is possible to extract the exact silhouette in the frontal

view, the silhouettes in the top and side views do not correspond to exact body silhouettes but they are the areas obtained by filling the backside of the partial contour available in these views due to lack of information in depth data for the obscured parts of the body. Optical flow is a fine technique for feature extraction from successive video frames. In our case, since the data is depth sequence, the optical flow values are calculated for each plane separately considering the silhouettes on these planes. Calculated optical flow values are examined and separated by being positive or negative values. Afterwards, average of optical flow is estimated. The classification part is straightforward. Support vector machine (SVM) is well accepted methodology for human action recognition and this technique is used also in this thesis.

As mentioned previously, the proposed method is evaluated on MSR Action3D dataset[6]. Our extensive experimental results show that the proposed method is able to achieve quite successful recognition accuracy and competes the state of the art methods.

1.5.Organization of the Thesis

The rest of this thesis is organized as follows. In Chapter 2, a literature survey on 3D human action recognition is presented and background information on fundamental techniques that are used in the thesis are provided. In Chapter 3, our proposed method is presented in detail with emphasize on feature extraction stage. In Chapter 4, experimental results obtained on MSR Action3D dataset is presented and compared with the results of the methods evaluated on the same dataset in literature. Finally, Chapter 5 concludes the thesis study and explains future work.

CHAPTER 2

THEORETICAL BACKGROUND

In this chapter, a literature survey is provided and the fundamental techniques which are used in this study are explained. In Section 2.1, the literature survey on human behavior understanding using 3D data is mentioned. In Section 2.2, Lucas Kanade optical flow method is explained. Finally, in Section 2.3, support vector machine (SVM) classifier is presented.

2.1.Literature Survey on Human Behavior Understanding Using 3D Data

Only a limited datasets recorded by depth sensors are publicly available for researchers. These datasets include different actions and activities performed by different volunteer subjects. The dataset used in this thesis study is the MSR Action3D dataset, which is also the one used in literature mostly. In this survey, human action recognition methods which are evaluated on MSR Action3D dataset [6] are reviewed.

Microsoft Kinect is the depth camera used in recording the videos in the MSR Action3D dataset. Microsoft Kinect automatically delivers the skeleton information beside the depth maps of people. Therefore the researches based on the datasets extracted with this device, commentated human action systems either using skeleton joints or depth map.

2.1.1. Human Action Recognition using depth maps

Li et al. [6] introduced a method that recognizes human actions from depth sequences. The study aims to develop a method that does not require joint tracking. This method uses 3D contour points instead of 2D silhouette. There are three orthogonal Cartesian plane projections of depth maps. In each frame, a specified numbers of points along the contours of all three projections are sampled. These samples are used in order to develop "a bag of points" model. Since dynamics of the actions are modeled in action graph, this model shows a set of salient postures that correspond to the nodes of an action graph. MSR Action3D dataset was created and firstly used by these authors. The human action recognition rate of this research is 74.4%.

In order to enhance some of the issues and to obtain better recognition accuracies, Vieira et al. [9] proposed space–time occupancy patterns (STOP). This method portrays the sequence of depth maps. In this method, the space and time axes are divided into multiple segments for embedding each action sequence in a multiple 4D grid. A saturation scheme consisting of points on a silhouette or some parts of body with motion was proposed in order to develop the role of spare grids. For classification part, a nearest neighbor classifier using the cosine distance was employed. Experiments are conducted on MSR Action3D dataset and the results show that STOP features produce better recognition accuracy for human action classification with respect to [6]. The disadvantage of this method is that setting the parameter for dividing sequences into the grids are done empirically.

Wang et al. [10] proposed a new method based on random occupancy pattern (ROP) features for addressing the noise and occlusion issues in action recognition. In this method, 4D shape is constructed from 3D action sequences. According to this method, randomly sampled 4D sub-volumes of different sizes and at different

locations using a weighted sampling scheme constitute the ROP features. In order to select the most discriminative features, an elastic-net regularized classification is modeled. These selected features are robust to noise and less sensitive to occlusions. While in real implementation 4D sub-volumes are used, 3D sub-volumes are shown in the illustration. For classification part, SVM classifier is used. Experiments are performed on the MSR Action3D dataset and the results show that this method outperforms previous methods by Li et al. [6] and Vieira et al. [9].

Yang et al. [11] proposed the new action recognition method which has ability to extract additional motion and shape information by using 3D depth maps. According to this system, three orthogonal Cartesian planes projections are obtained firstly for each 3D depth maps in the sequence. In order to generate each projection, the differences of consecutive depth frames are thresholded and then depth motion map (DMM) is obtained. For the feature extraction part, histogram of oriented gradients (HOG) is applied to each Cartesian plane projection. After this process, the futures obtained for each plane are concatenated and DMM-HOG descriptor is generated. In classification part, SVM classifier is used. The disadvantage of this method is that motion maps do not provide directional velocity information between the frames.

Dobhal et al. [12] proposed a new method based on binary motion image (BMI). On the idea of 2D representation of action video sequence, the image sequences are combined into a single image called BMI. BMI demonstrates the flow of motion of action and it is invariant to holes, shadows and partial occlusions. For classification part, they employed the Convolutional Neural Network (CNN) classifier. This classification method not only extracts meaningful features automatically but also introduces invariance to distortion. Experiments are performed on 2D Weizmann dataset. They extended their proposed 2D method to 3D depth maps using MSR Action3D dataset by extracting three BMI projections namely the front view, the

side view and top view. In order to obtain one single image for each action, three calculated BMIs are superimposed and normalized. Then this single image is fed into CNN classifier for training and testing part. According to the experimental results, they believe that BMI is sufficient for human action recognition and it has shown to be invariant to speed of the action performed in addition to the aforementioned variations.

2.1.2. Human Action Recognition using skeleton joints

In the sense of human action recognition using skeleton joints, Xia et al. [5] proposed the method based on 3D skeleton joints. In this method, histogram of 3D joint locations (HOJ3D) reveals the 3D human postures. In their representation, 3D space is partitioned into bins using a modified spherical coordinate system. For this purpose, 12 manually selected joints were used to build a compact representation of the human posture. While votes of 3D skeleton joints were cast into neighboring bins, Gaussian weight function is used in order to make the representation more robust. In feature extraction part, linear discriminant analysis (LDA) was carried out for dimension reduction in order to choose most dominant and discriminative features. These features were clustered into a fixed number of posture vocabularies which represent the prototypical poses of actions. Lastly, in human action classification part, the discrete Hidden Markov Model (HMM) was applied to extracted visual words for training and testing. Experimental results were obtained by using the MSR Action3D dataset. The method has the disadvantage that relying only on the hip joint might potentially decrease recognition accuracy because of the noise in estimation of hip joint location.

Yang et al. [13] also proposed a new method based on skeleton joints. The authors mentioned that skeleton joints have some advantages such as computationally inexpensive, more compact, and distinctive compared to depth maps. In the light of these benefits, they proposed the Eigen joints–based action recognition method. In

this system, three different features were extracted by using the skeleton joints. These features include posture (Fcc), motion features (Fcp) and offset features (Fci). Fcp encode the spatial and temporal characteristics of skeleton joints. Fci calculate the difference between a current pose and the initial one. In next step, principle component analysis (PCA) is applied to joint differences for obtaining the Eigen joints. As a classifier Naive-Bayes-Nearest- Neighbor (NBNN) classifier is used for human action recognition. Moreover, they found that there are a specific number of frames for successful recognition rates of human actions. Therefore, short sequence of 15-20 frames is sufficient for human action recognition according to their experimental results on the MSR Action3D dataset. In feature extraction part, the offset feature is calculated assuming that the initial skeleton pose is neutral. This is actually a disadvantage for this method, since the initial skeleton pose is not always neutral.

Wang et al. [14] proposed a method based on skeleton joint approach. In this method, skeleton and point cloud information are utilized. Skeleton information is not enough since some actions differ due to object interaction. To enhance this insufficient skeleton information, a novel method called actionlet ensemble model is introduced to refer each action and capture intra-class variance via occupancy information. Interactions between humans and objects are characterized by Local Occupancy Patterns (LOP) at each joint. The LOP features are computed based on the 3D point cloud around a particular joint. In this method, The Fourier Temporal Pyramid features are generated, and then the feature vectors are concatenated and Short Fourier Transform is applied to these feature vectors. In classification part, SVM classifier is applied. Experiments are conducted on MSR Action3D dataset and also on a new dataset called MSR Daily Activity3D. Experiments showed that their proposed method has more successful performance compared to [12] and [13] methods.

2.2.Lucas Kanade Optical Flow Method

In optical flow calculations, several assumptions are made and the most basic one of them is image brightness constancy. This assumption is from a short interval t1 to t2, position of object may change, the illumination and reflectivity will remain constant [15]. This is

$$f(x + \Delta x, y + \Delta y, t + \Delta t) \approx f(x, y, t) \tag{2.1}$$

Where $f(x, y, t)$ is the intensity of the image at time t and position (x, y); $\Delta x, \Delta y$ is the change in position and Δt is the change in time. If the Taylor series expansion is applied to the left hand side, equation will be as follows:

$$f(x + \Delta x, y + \Delta y, t + \Delta t) = f(x, y, t) + \frac{\partial f}{\partial x}\Delta x + \frac{\partial f}{\partial y}\Delta y + \frac{\partial f}{\partial t}\Delta t + h.o.t \tag{2.2}$$

Where $\frac{\partial f}{\partial x}, \frac{\partial f}{\partial y}, \frac{\partial f}{\partial t}$ are the partial derivatives of the image function in the x, y and t dimensions. It can be ignored the higher order terms (h.o.t.). Once substituting Equation 2.1 to Equation 2.2, the optical flow equation is obtained as follows.

$$\nabla I \cdot v + I_t = 0 \tag{2.3}$$

Where $\nabla I = (I_x, I_y)$ is the spatial gradient, $v = (u, \vartheta) = (\Delta x, \Delta y)$ is the optical flow vector and I_t is the temporal gradient. Because, it is considered a single time displacement between two frames, $\Delta t = 1$ and thus disappear. Using derivative operators, three gradients are easy to calculate. From this equation it can said that when flow vector is applied to the spatial gradient of the image it will be exactly canceled by temporal gradient.

During calculation of optical flow fields a lot of assumptions are made. Assume that objects are illuminated uniformly and there are no occlusions or transparencies. Mainly there exist two separate and widely known techniques for optical flow, which are Horn and Schunk and Lucas Kanade. In Horn and Schunk technique, the derivatives are used in order to calculate the first constraint on the flow vector, and then solve for the orthogonal component using a global method of minimizing a smoothness constraint. In Lucas and Kanade technique [17], a local method is used in order to calculate the flow vector using the constraints of a neighborhood around the pixel [16]. Lucas Kanade method uses a weighted least squares method to approximate the optical flow at pixel (x,y).

$$E_V = \sum_{p\in\varphi} W^2(p)[\,\nabla I(p)\cdot v + I_t(p)] \tag{2.4}$$

Where $\nabla I(p)$ and $I_t(p)$represents the spatial gradient and temporal gradient at neighboring pixel p respectively. v is the optical flow vector for pixel (x, y) and W(p) is the weight associating with neighboring pixel. For each pixel, optical flow vectors are found on the surrounding neighborhood φ of size n, where each neighbor is represented as p_i.

Lucas Kanade method supports for the flow vector local rather that global like Horn and Schunk technique. While it is needed to several iterations on Horn and Schunk [18], in Lucas Kanade method there is no several iterations due to being global technique. Since Lucas Kanade is performing consistently and robustly, it is easy to implement it, this method is chosen.

2.3.Support Vector Machine (SVM) Classifier Method

In machine learning, support vector machines are supervised learning models with associated learning algorithms that analyze data and recognize patterns, used or classification and regression analysis.

The support vector machine constructs a hyper plane or set of hyper planes in a high-or infinite-dimensional space, which can be used for classification, regression or other tasks. Intuitively, a good separation is achieved by the hyper plane that has the largest distance to the nearest training-data point of any class (so-called functional margin), since in general the larger the margin the lower the generalization error of the classifier.

In addition to performing linear classification, SVMs can efficiently perform a non-linear classification using what is called the kernel trick, implicitly mapping their inputs into high-dimensional feature spaces.

The goal in support vector machine is to design a hyper plane that classifies all training vectors in two classes. There can be different hyper planes. The best choice will be the hyper plane that leaves the maximum margin from both classes.

The margin is this distance between the hyper plane and closest elements from this hyper plane z1 and z2 margin. z2>z1 so the margin in z2 is higher than so choose this hyper plane.

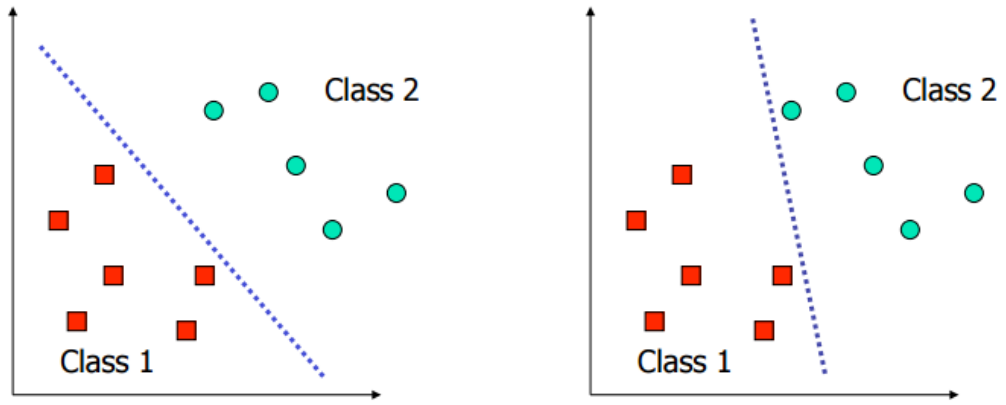

Figure 2.1: The figure of bad decision boundaries

The decision boundary should be as far away from the data of both classes as possible not like the Figure 2.1. For this purpose, the margin “m” should maximize.

Distance between the origin and the line equation are given in Equation 2.5.

$w^T x = k \; is \; \frac{k}{\|w\|}$ (2.5)

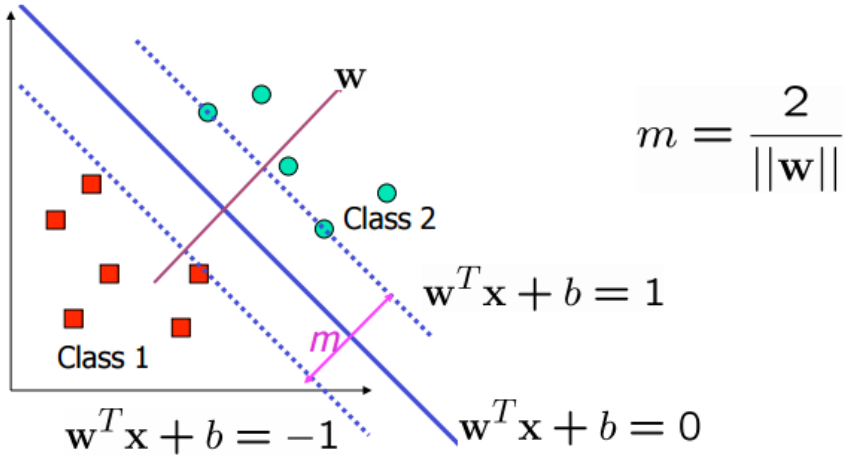

Figure 2.2: Choosing the decision boundary representation

Let {x1, ..., xn} be our data set and let yi Ɛ {1,-1} be the class label of xi and the decision boundary should classify all points correctly as shown in Equation 2.6.

$y_i(w^T x_i + b) \geq 1, \; \forall i$ (2.6)

The decision boundary can be found by solving the following constrained optimization problem in Equation 2.7.

Minimize $\frac{1}{2}\|w\|^2$ subject to $y_i(w^T x_i + b) \geq 1, \; \forall i$ (2.7)

This is constrained optimization problem. In order to solve it, there is a need for some new tools.

In order to minimize $\frac{1}{2}\|w\|^2$ subject to $y_i(w^T x_i + b) \geq 1, \; \forall i$ it should be used Lagrangian method. The Lagrangian method is given in Equation 2.8.

$L = \frac{1}{2} w^T w + \sum_{i=1}^{n} \alpha_i \left(1 - y_i(w^T x_i + b)\right)$ Note that $\|w\|^2 = w^T w$ (2.8)

Setting the gradient of L with respect to w and b to zero we have following equation.

$$w + \sum_{i=1}^{n} \alpha_i(-y_i)x_i = 0 \rightarrow w = \sum_{i=1}^{n} \alpha_i y_i x_i \qquad \sum_{i=1}^{n} \alpha_i y_i = 0 \tag{2.9}$$

If we substitute $w = \sum_{i=1}^{n} \alpha_i y_i x_i$ to L, we have the following equations.

$$L = \frac{1}{2}\sum_{i=1}^{n} \alpha_i y_i {x_i}^T \sum_{j=1}^{n} \alpha_j y_j x_j + \sum_{i=1}^{n} \alpha_i \left(1 - y_i\left(\sum_{j=1}^{n} \alpha_j y_j {x_j}^T x_i + b\right)\right) \tag{2.10}$$

$$L = \frac{1}{2}\sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_i \alpha_j y_i y_j {x_i}^T x_j + \sum_{i=1}^{n} \alpha_i - \sum_{i=1}^{n} \alpha_i y_i \sum_{j=1}^{n} \alpha_j y_j {x_j}^T x_i - b\sum_{i=1}^{n} \alpha_i y_i \tag{2.11}$$

$$L = -\frac{1}{2}\sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_i \alpha_j y_i y_j {x_i}^T x_j + \sum_{i=1}^{n} \alpha_i \qquad \text{note that } \sum_{i=1}^{n} \alpha_i y_i = 0 \tag{2.12}$$

This is a function of α_i only. It is known as dual problem. If we know w, we know allα_i, if we know all α_i, we know w. The original problem is known as the primal problem. The objective function of the dual problem needs to be maximized.

Therefore the dual problem is becoming as in Equation 2.13.

$$W(\alpha) = \sum_{i=1}^{n} \alpha_i - \frac{1}{2}\sum_{i=1,j=1}^{n} \alpha_i \alpha_j y_i y_j {x_i}^T x_j \text{ subject to } \alpha_i \geq 0 \text{ and } \sum_{i=1}^{n} \alpha_i y_i = 0 \tag{2.13}$$

This is a quadratic programming problem. A global maximum of α_i can always be found. w can be recovered by follow equation.

$$w = \sum_{i=1}^{n} \alpha_i y_i x_i \tag{2.14}$$

CHAPTER 3

PROPOSED ACTION RECOGNITION METHOD

Proposed human action recognition method is basically based on feature extraction and classification main stages. The general block diagram of the proposed algorithm is shown in Figure 3.1.

In feature extraction stage, three different methods are carried out in order to achieve successful human action recognition accuracies. 3D depth sequences are input for the feature extraction stage. After the 3D depth sequence dataset is processed, firstly x-y plane (frontal view) depth information is obtained for each frame in all episodes. This frontal plane data is projected on to x-z (top view) and z-y (side view) Cartesian planes with the help of acquired frontal view depth information. These sections of feature extraction stage are common for all methods. At the end of feature extraction part, distinctive feature vectors are attained according to three different feature extraction methods. Afterwards, these feature vectors are fed into classification stage. The output of classification stage is the label of the action recognized.

The best feature extraction among these methods is reached by comparing the human action recognition rates which are presented in detail in the next chapter.

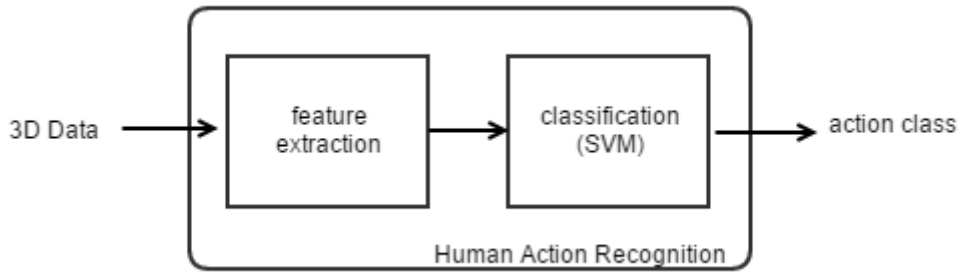

Figure 3.1: The general block diagram of proposed method

3.1. Matrix Notation for Representation of Data

Action, subject and episode parameters are defined as follows.

$Action_i = a_i$ where "i" is the action number; i=1, 2…20

$Subject_j = s_j$ where "j" is the subject number; j=1, 2…10

$Episode_k = e_k$ where "k" is the episode number; k=1, 2, 3

The episode e_k of the subject s_j of the action a_i is represented as $F(a_i, s_j, e_k)$ matrix.

The matrix notation for action i, subject j, all episodes are represented as in Equation 3.1. In Equation 3.2, the matrix represents the action i, for all subjects and all episodes. If we consider all actions, all subjects and all episodes the matrix becomes as given in Equation 3.3.

$$Fa_is_j = F(a_i, s_j, :) = \begin{bmatrix} a_is_je_1 \\ a_is_je_2 \\ a_is_je_3 \end{bmatrix} = 3\ episodes\ of\ subject\ j\ and\ action\ i \quad (3.1)$$

$$Fa_i = F(a_i, :, :) = \begin{bmatrix} Fa_is_1 \\ Fa_is_2 \\ \dots \\ Fa_is_9 \\ Fa_is_{10} \end{bmatrix} = 3\ episodes\ of\ 10\ subjects\ of\ action\ i \quad (3.2)$$

$$F = F(:,:,:) = \begin{bmatrix} Fa_1 \\ Fa_2 \\ \dots \\ Fa_{19} \\ Fa_{20} \end{bmatrix} = 3\ episodes\ of\ 10\ subjects\ of\ 20\ actions \quad (3.3)$$

3.2. Feature Extraction

In feature extraction part, a 3D depth file of each episode is the input which is also the whole system. At the end of this part, the feature vectors are extracted in order to supply to the classification part. Three different feature extraction methods (FEM) are proposed and examined in feature extraction part and the optimal one is chosen for our final action recognition method by conducting several experiments. These methods are consisting of various types of steps but they have also some common sections. For instance, processing of 3D depth data to obtain silhouettes in orthogonal planes and finding the active blocks for the convenience of operations in classification part are mutual for each FEM proposed. These FEMs will be explained in detail separately below.

3.2.1. Feature Extraction Method 1: Average of Silhouettes (AoS)

The block diagram of the feature extraction method 1 (FEM1) is shown in Figure 3.2.

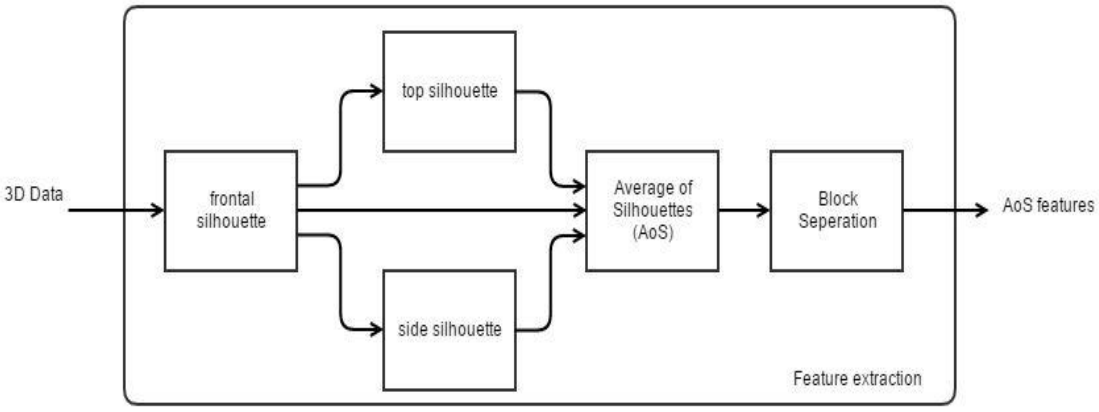

Figure 3.2: Block diagram for FEM1: Average of silhouettes (AoS)

First feature extraction method consists of reading the 3D depth files, silhouette extraction, average of silhouettes and block separation parts. Firstly, 3D depth files of dataset are read separately for each episode of related actions and subjects. Depth information of x-y plane that obtained for all episodes after processing the 3D depth files will be used for other plane projections. In all three planes, frontal, top and side view silhouettes are extracted, averages of these silhouettes are calculated and these averages are given to the block separation part. At the end of feature extraction stage, the future vectors that we call AoS (average of the silhouettes) features which will be given to the classification stage are attained.

3.2.1.1. Extracting Silhouettes

The 3D depth file is consisting of depth information for specific action, subject and episode. Each depth file should be converted to the meaningful data. Each video has variety number of frames; however, all frames have 320*240 pixels in x-y plane. Firstly x-y projection should be found from the depth file, and x-z and z-y plane projections are achieved, since depth files give the depth information of each pixel for x-y plane. Barely, as depth information cannot give the complete 3D information, as explained later, some other processes are needed in these planes for attaining the future vectors.

If the depth value of one pixel is zero there is no data. If the pixel value is smaller, it means the part of subject related this pixel is closer to the camera. On the other hand, if the pixel value is larger, which means the part of subject related this pixel is far away from the camera. Depth values are scanned in the boundary of human body for each frame of each episode in all dataset. Maximum depth value and minimum depth value except from "0" is found after scanning the all dataset in the boundary of human body. These values are minimum 290 and maximum 649. In order to give some margin, minimum and maximum values are switched to 200 and 800. Then the x-y plane data is set to "0" if background, and to "1" otherwise and frontal silhouette frames are extracted. At the end of this operation, the silhouette in the frontal plane is obtained. One example of x-y plane image, that is frontal silhouette, found from depth file and boundary of human body is shown in Figure 3.3.

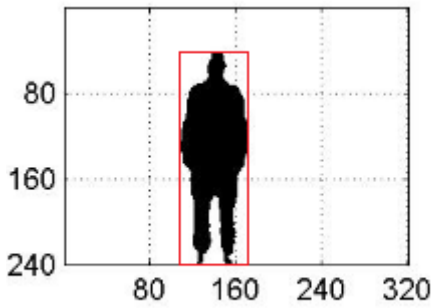

Figure 3.3: The example of frontal silhouette (x-y plane image)
and boundary of human body

All planes are representing the different views. The representations in x-y plane (frontal view), x-z plane (top view) and z-y (side view) plane are corresponding to frontal view, top view and side view respectively. The resolution of top view is 320x800 pixels and side view is 800x240 pixels.

For the projection to other planes, maximum and minimum depth values except from "0" are found for each frame by using the x-y plane depth information. The pixel corresponding to minimum depth value is chosen as reference point in x-z or z-y planes (other planes), after scanning rows or columns according to the related projections.

For example for the projection of x-z plane from x-y plane, x axis is common and we sweep the y axis of x-y plane. In x axis there are 320 columns. For each of 320 columns, all of the 240 rows are checked and the minimum depth value (i.e. closest to the camera) is determined. In this way, finally a vector involving minimum depth values for each column is obtained. These depth minimums for columns in x-y plane (frontal view) are used to find the silhouette in the x-z plane (top view). For this purpose, firstly for each column i in the x-z plane, the pixel corresponding to the depth minimum in the x-y plane is determined. Let the pixel in column i, has minimum value at row j, and the value is d_i that corresponds the z axis in x-z plane. Then the pixel corresponding to this minimum is the one in the position (i, d_i) in the x-z plane. Then all the pixels in positions (i, k) in x-z plane, k$\geq d_i$ are marked as belonging to the silhouette. This is repeated for each column i, in the x-y plane and the top view silhouette is obtained. In a similar way for each row, depth minimums are determined in the frontal view silhouette and the side view silhouette is obtained.

Some examples for silhouettes in frontal, top and side views are given as in Figure 3.4-Figure 3.9. All examples are given for selected frames of action 10, hand clap action, subject 1, and episode 1. Corresponding pixels for the head and hands of human are shown from x-y plane (frontal view) to related projected planes x-z plane, z-y plane (top and side views). Although z axis has 800 pixels, for representation 400 pixels of them are shown in Figure 3.4-Figure 3.6.

It should be noted that while the silhouette in the frontal view is exact, the silhouettes in the top and side views are only reflecting the correct silhouette on the side close to the camera. This is due to lack of information on the occluded portion of the body in depth data.

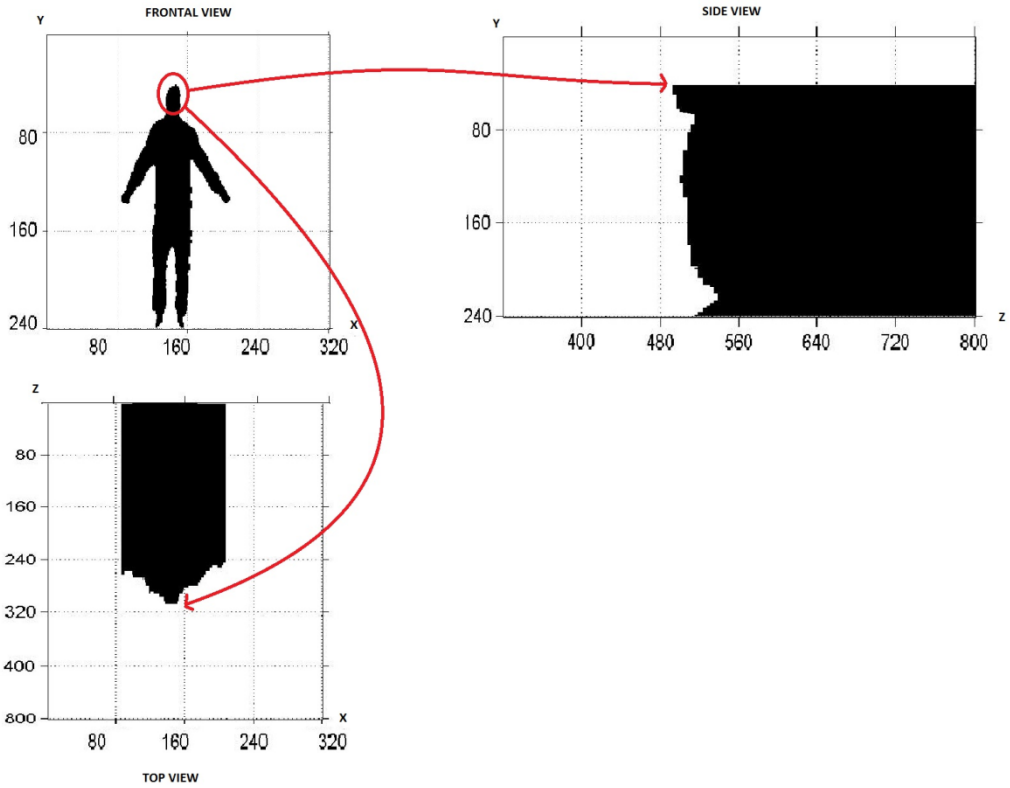

Figure 3.4: Frontal, top and side view silhouettes of frame 9
for hand clap action, subject1, episode1

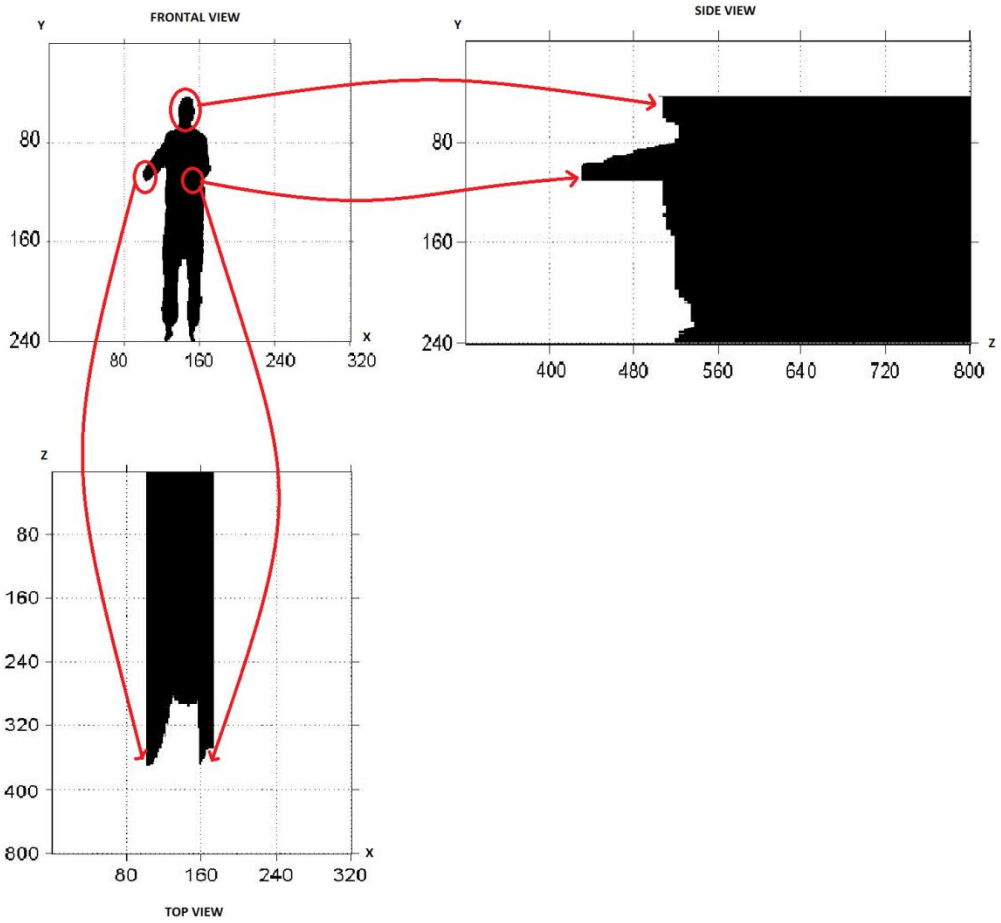

Figure 3.5: Frontal, top and side view silhouettes of frame 17
for hand clap action, subject1, episode1

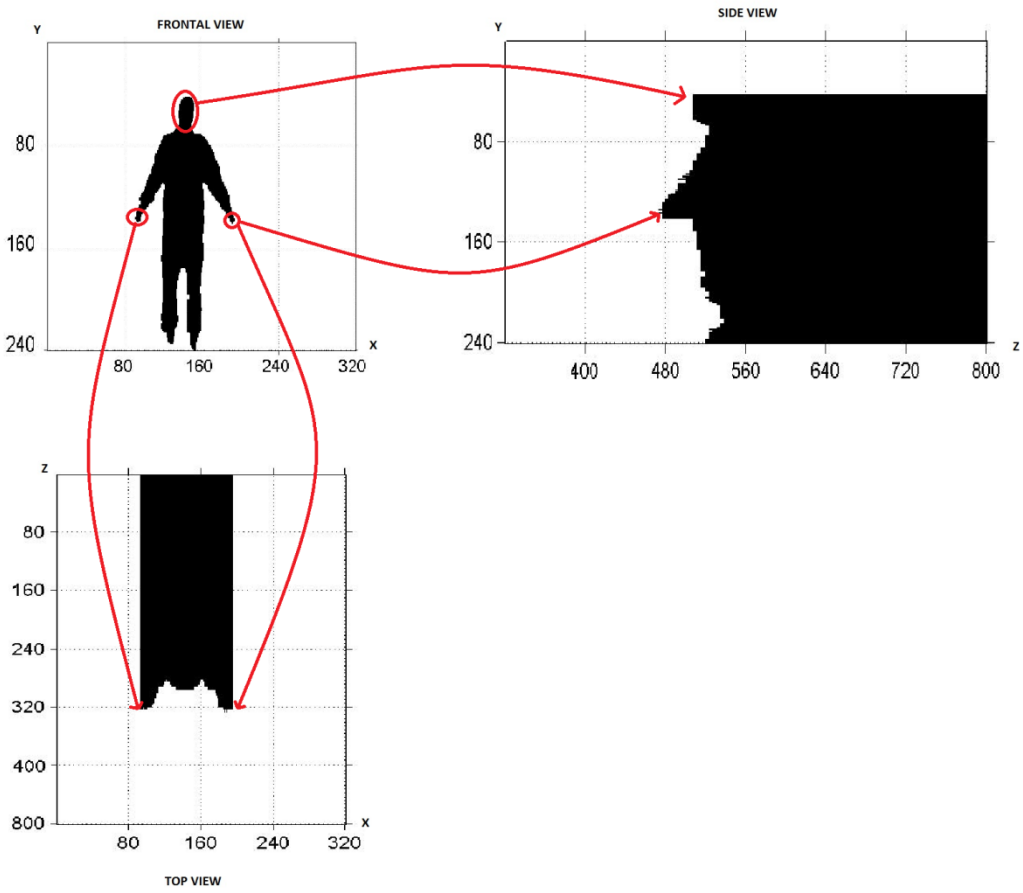

Figure 3.6: Frontal, top and side view silhouettes of frame 25
for hand clap action, subject1, episode1

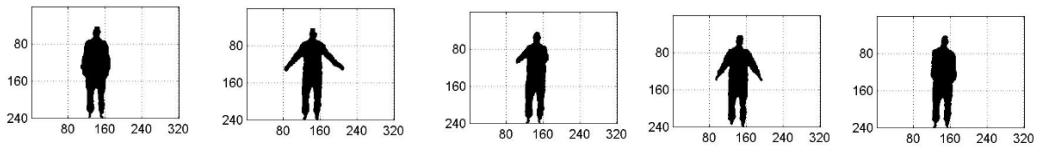

Figure 3.7: Silhouettes of frontal view for hand clap action, subject1, episode 1

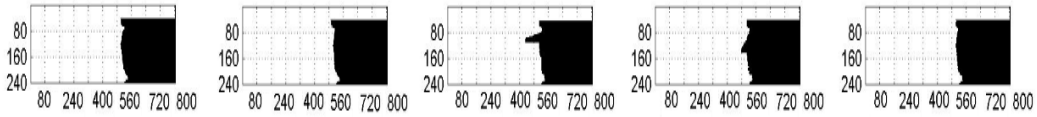

Figure 3.8: Silhouettes of side view for hand clap action, subject1, episode 1

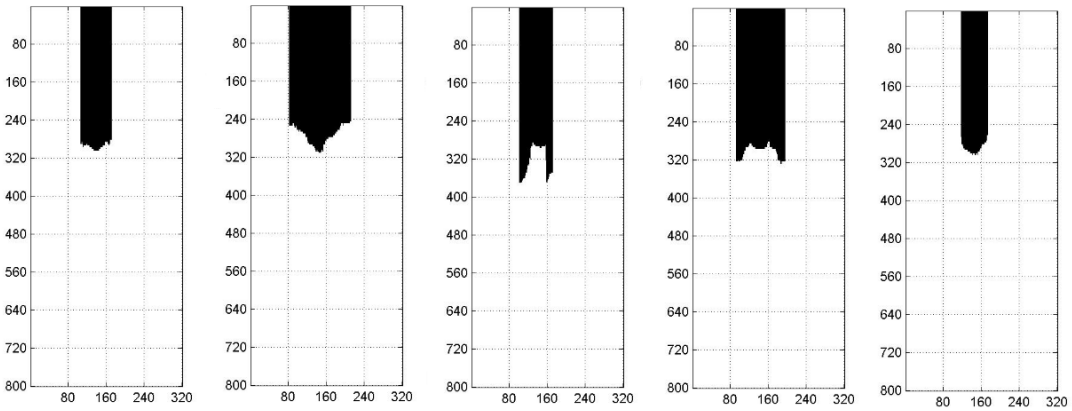

Figure 3.9: Silhouettes of top view for hand clap action, subject1, episode 1

3.2.1.2. Average of Silhouettes (AoS)

After the silhouettes for frontal, top and side views are obtained; each consecutive silhouette matrixes in episode should be summed up and should be averaged with dividing considering frame numbers. This process is repeated for all actions, subjects, episodes and views. In the following, the calculation for the frontal view is explained. For the other views, the calculations are done in the similar manner. In the end, averages of action matrixes for each episode are obtained.

How to find averages of silhouettes from frames is given in Equation 3.4.

$$AoS_{x-y}A_iS_jE_k = \frac{\sum_{f \in frames} S_{x-y}(a_i, s_j, e_k, f)}{N_{frame}(a_i, s_j, e_k, f)} \tag{3.4}$$

where $S_{x-y}(a_i,\ s_j, e_k, f)$ is the silhouette data extracted for frontal view, frame f, and corresponding episode and N_{frame} is the # of frames for corresponding episode.

After finding the averages, there are three huge matrixes for frontal, top and side view silhouettes.

In order to conduct experimental results, these three matrixes are combined into one matrix. While combining them, block separation method is used for the purpose of simplification in classification part.

Some frontal view AoS examples are given for forward punch action (Action5), side boxing action (Action12), side kick action (Action 15), and tennis serve action (Action 18) in Figure 3.10 and Figure 3.11.

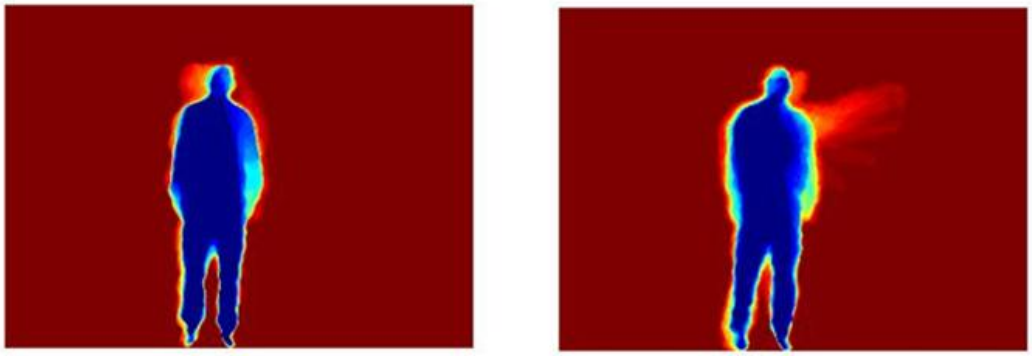

Figure 3.10: Subject1, episode1 for forward punch action (left) and side boxing action (right) AoS examples

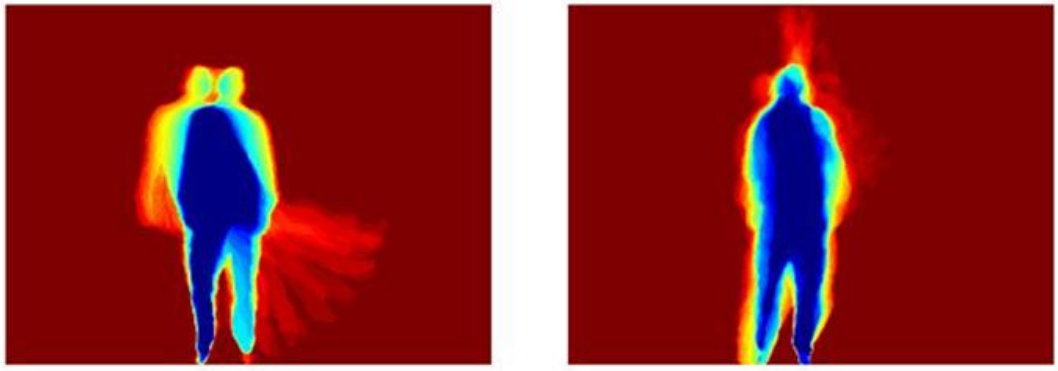

Figure 3.11: Subject1, episode1 for side kick action (left) and tennis serve action (right) AoS examples

3.2.1.3. Block Separation

By examining the AoS for frontal, top and side view silhouettes, the meaningful blocks are found and final incorporated matrix of three Cartesian planes is obtained. With the help of block separation, the active blocks where most of the actions are occurring are obtained for each view by eliminating which have values close to zero.

80*80	B1 80*80	B2 80*80	80*80
80*80	B3 80*80	B4 80*80	80*80
80*80	80*80	80*80	80*80

Figure 3.12: The frontal view (320*240) active blocks

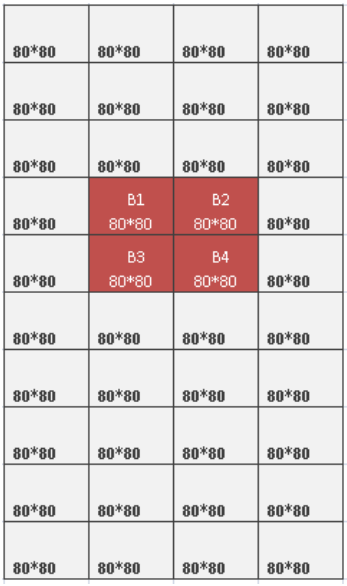

Figure 3.13: The top view (320*800) active blocks

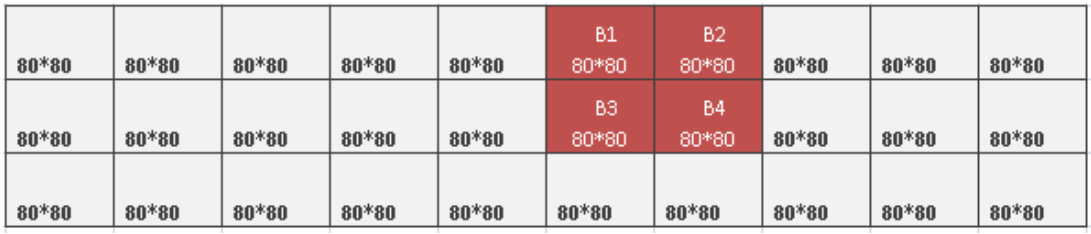

Figure 3.14: The side view (800*240) active blocks

For each view there are 4 active blocks separately. The dimension of each active block is 80*80 and these blocks are shown according to the corresponding view in Figure 3.12-Figure 3.14. All pixels in active blocks are used as a feature. Therefore, each active block has 80*80= 6400 features. Since there are four blocks for one projected view 6400*4=25600 features exist. This feature number is only for one view. For all three views it should be multiplied by 3. Totally there are $25600 * 3 = 76800$ features exist for three different views.

Since we have 20 actions, 10 subjects, 3 episodes, totally 20*10*3=600 samples exist. Eventually, the final matrix for all dataset has dimension (600*76800) and this feature vector is input for classification part.

3.2.2. Feature Extraction Method 2: Average of Silhouette Difference (AoSD)

The block diagram of the feature extraction method 2 (FEM2) is shown in Figure 3.15.

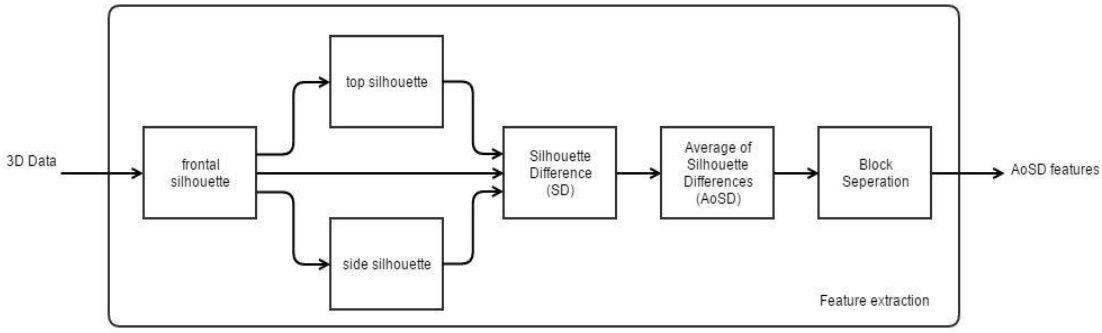

Figure 3.15: Block diagram for FEM2: Average of silhouette differences (AoSD)

Second feature extraction method consists the same steps of silhouette extraction as explained in FEM1. Then, consecutive frames are examined and differences of silhouettes are calculated for each plane. After finding these differences, averages of these silhouettes are calculated and these average matrixes are given to the block separation part. The block separation part is done in the same way with in FEM1. At the end of feature extraction method, the feature vectors that we call AoSD (average of the silhouette differences) which will be given to the classification stage are attained.

3.2.2.1. Silhouette Differences (SD)

After frontal, top and side view silhouettes are obtained, it is needed to find the motion existence in consecutive frames. For this purpose, silhouettes in consecutive frames are examined. If the pixel values in consecutive silhouette images are different the pixel value on the silhouette difference (SD) image is set to 1 to indicate existence of motion, otherwise it is set to 0.

Finally, we have (frame number -1) SD matrixes for each episode. This process should be done for frontal, top and side view silhouettes for all dataset.

3.2.2.2. Average of Silhouette Difference (AoSD)

After we obtain all silhouette differences, we need to obtain their averages. For this purpose, all obtained SD matrixes are summed up and divided by their number (frame number-1) for each episode. After this operation, one average matrix is obtained per each view, and totally there exists three matrixes (average silhouette difference image) for one episode. Some frontal view AoSD examples are given for forward punch action (Action5), side boxing action (Action12), side kick action (Action 15), and tennis serve action (Action 18) in Figure 3.16 and Figure 3.17.

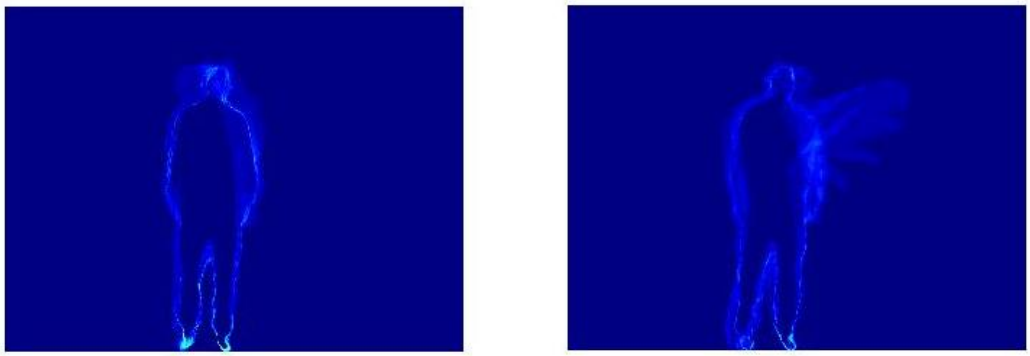

Figure 3.16: Subject1, episode1 for forward punch action (left) and side boxing action (right) AoSD examples

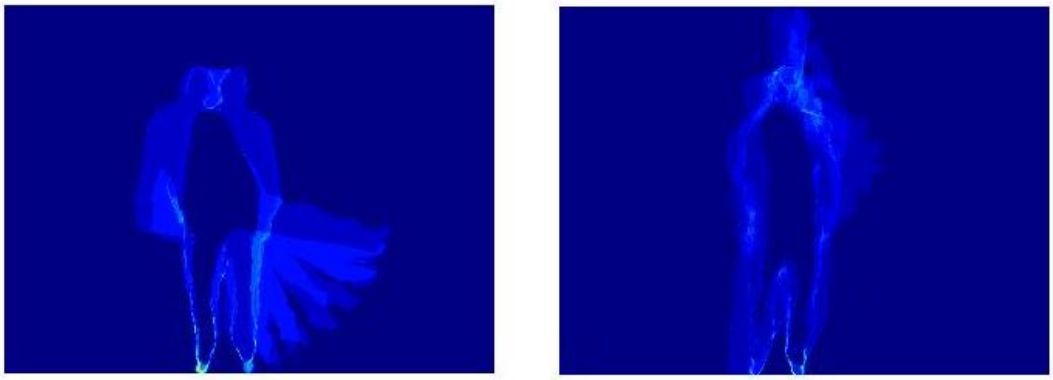

Figure 3.17: Subject1, episode1 for side kick action (left) and tennis serve action (right) AoSD examples

There are several matrixes and we should combine them for the ease of calculation. While combining all matrixes to one matrix, block separation method that mentioned before is applied and final matrix is reached for classification part. In the end of this method, the feature vector has dimension (600*76800) which is same as in FEM1.

3.2.3 Feature Extraction Method 3: Average of Silhouette Flows (AoSF)

The block diagram of the feature extraction method 3 (FEM3) is shown in Figure 3.18. This method provides us to most successful experimental results, therefore this is chosen as optimal method.

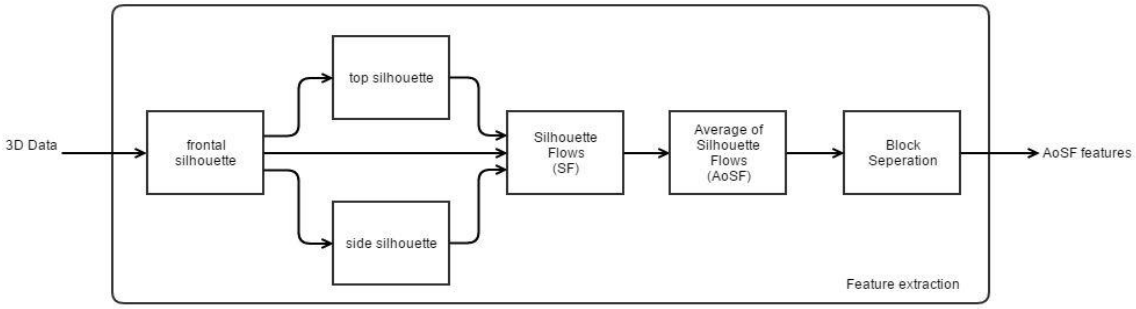

Figure 3.18: Block diagram for FEM3: Average of silhouette flows (AoSF)

Third feature extraction method consists the same steps of silhouette extraction as explained in FEM1. Then consecutive frames are examined and silhouette flows are calculated for each plane. After finding these silhouette flows, averages of silhouette flows are calculated and these average matrixes are given to the block separation part. The block separation part is done in the same way with in FEM1. At the end of feature extraction method, the feature vectors that we call AoSF (average of the silhouette flows) which will be given to the classification stage are attained.

3.2.3.1. Silhouette Flow (SF)

After silhouettes are obtained, we need to find the motion feature vectors using optical flow method. For this purpose, Lucas Kanade method is used as optical flow method which is implemented in Dollar toolbox [19].

It should be noted that Lucas Kanade method is in fact developed for gray level images, but here our images, i.e., silhouettes are binary images. We call the resulting images showing Vx and Vy values for each pixel as silhouette flows (SF).

When the related optical flow function is applied to frontal silhouette, Vx and Vy values of x and y axis motion is obtained for each pixel. Vx and Vy values can be positive or negative in accordance with motion types and motion directions. Some examples for this optical flow operation are given in Figure 3.19-Figure 3.31 for different actions which are forward punch action (Action5), side boxing action (Action12), side kick action (Action 15), and tennis serve action (Action 18).

In these figures, left columns show consecutive silhouettes in frontal view, while the right columns showing the flows calculated on these silhouettes. Vx values are shown at the top and Vy values at the bottom of the right column.

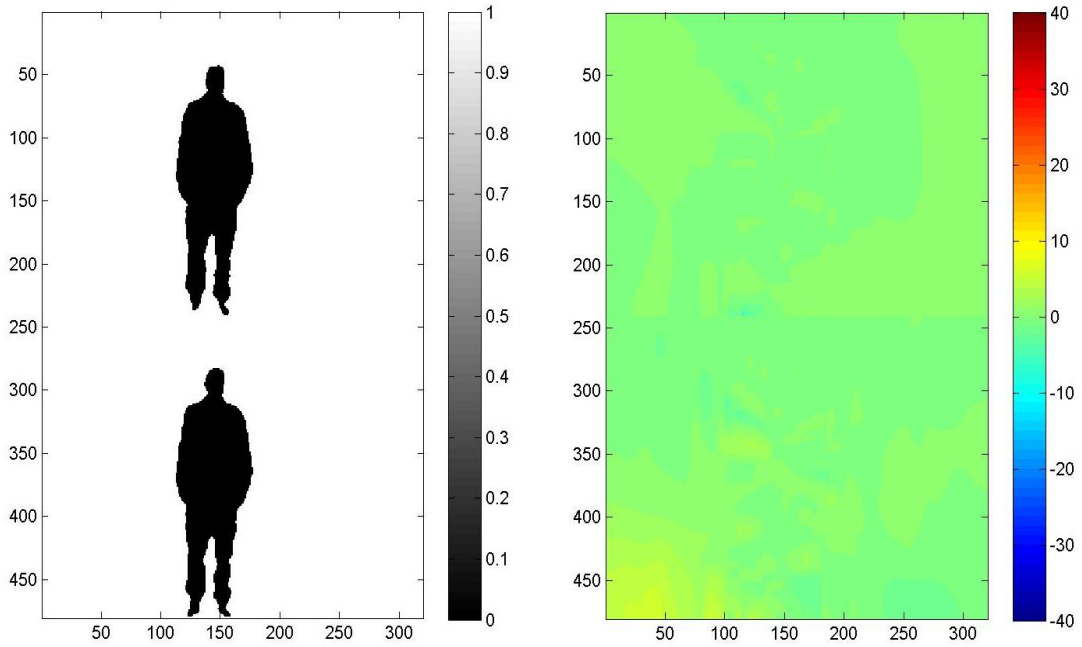

Figure 3.19: Forward punch action subject1, episode1, optical flow for frame 1&2

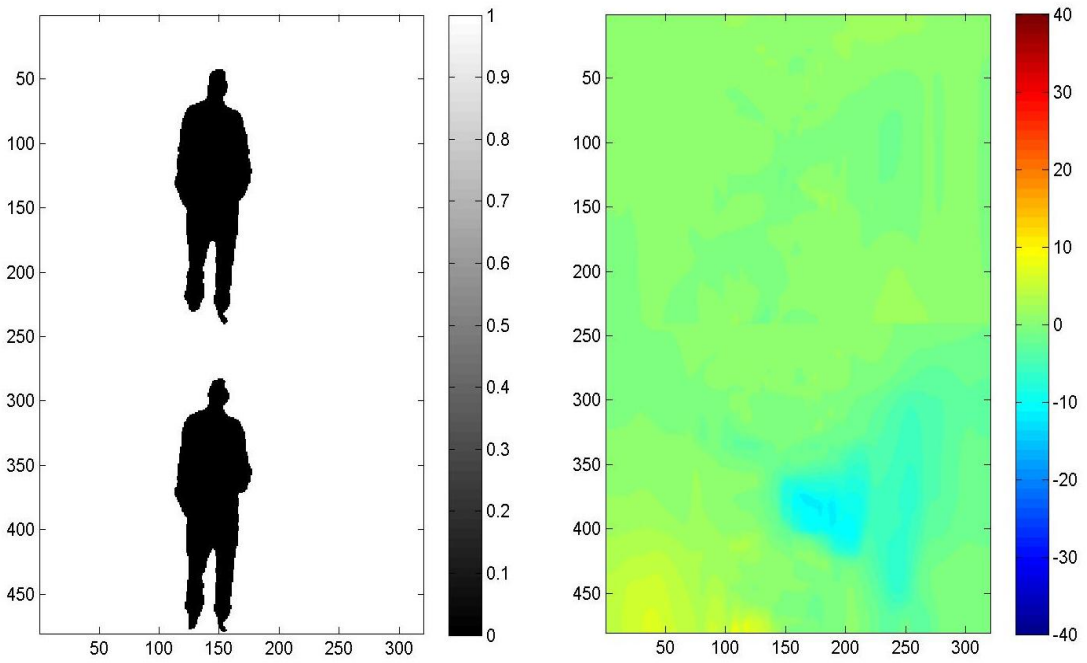

Figure 3.20: Forward punch action subject1, episode1, optical flow for frame 14&15

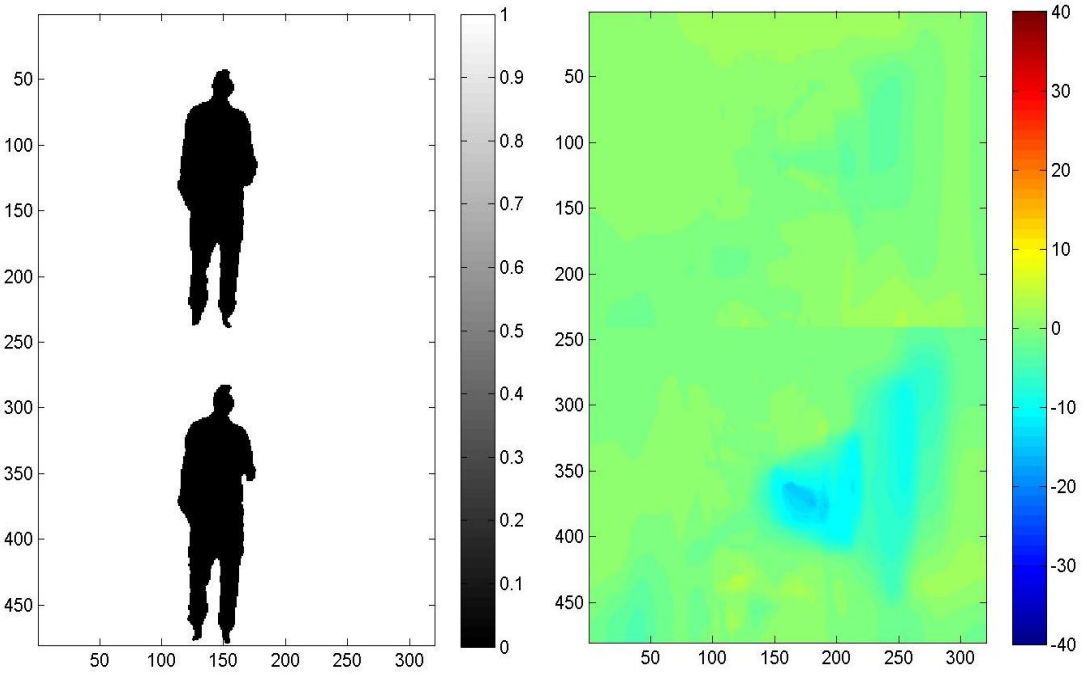

Figure 3.21: Forward punch action subject1, episode1, optical flow for frame 15&16

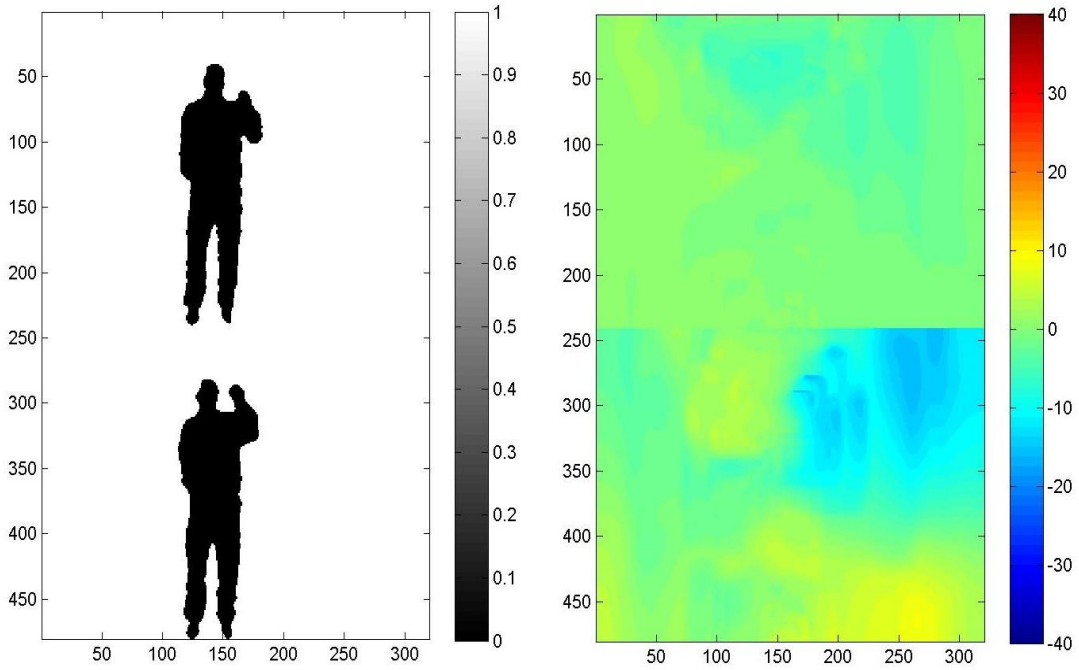

Figure 3.22: Forward punch action subject1, episode1, optical flow for frame 20&21

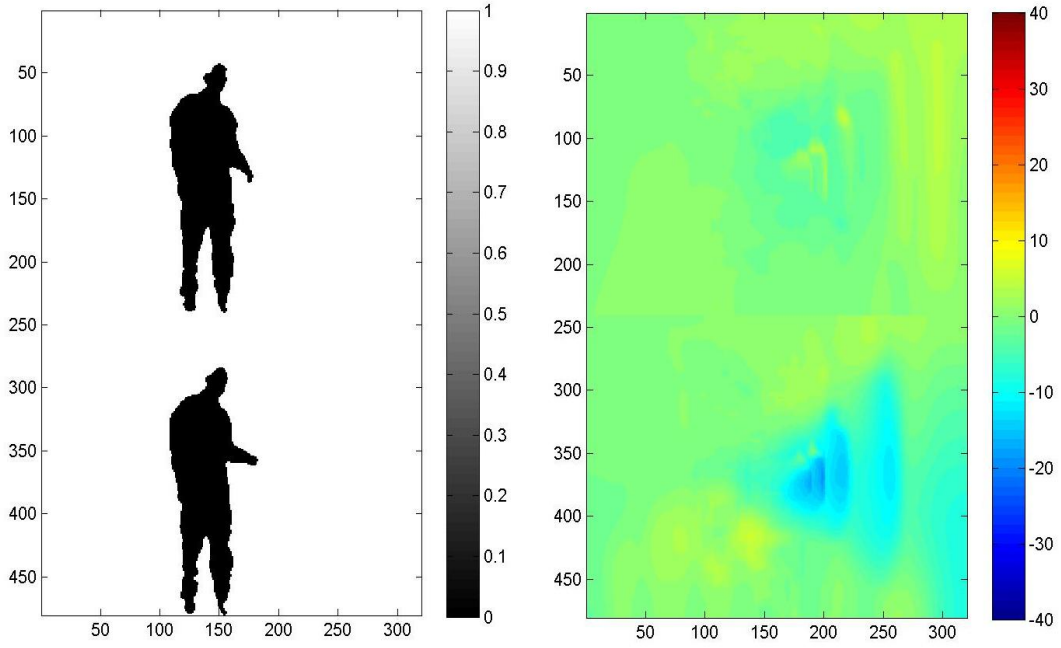

Figure 3.23: Side boxing action subject1, episode1, optical flow for frame 15&16

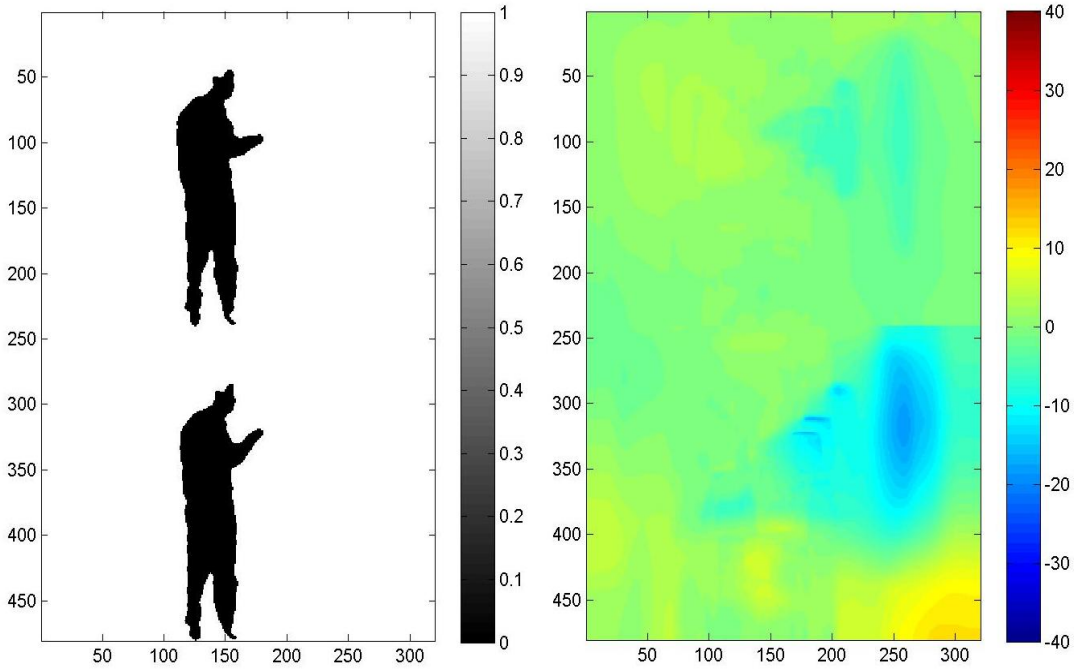

Figure 3.24: Side boxing action subject1, episode1, optical flow for frame 18&19

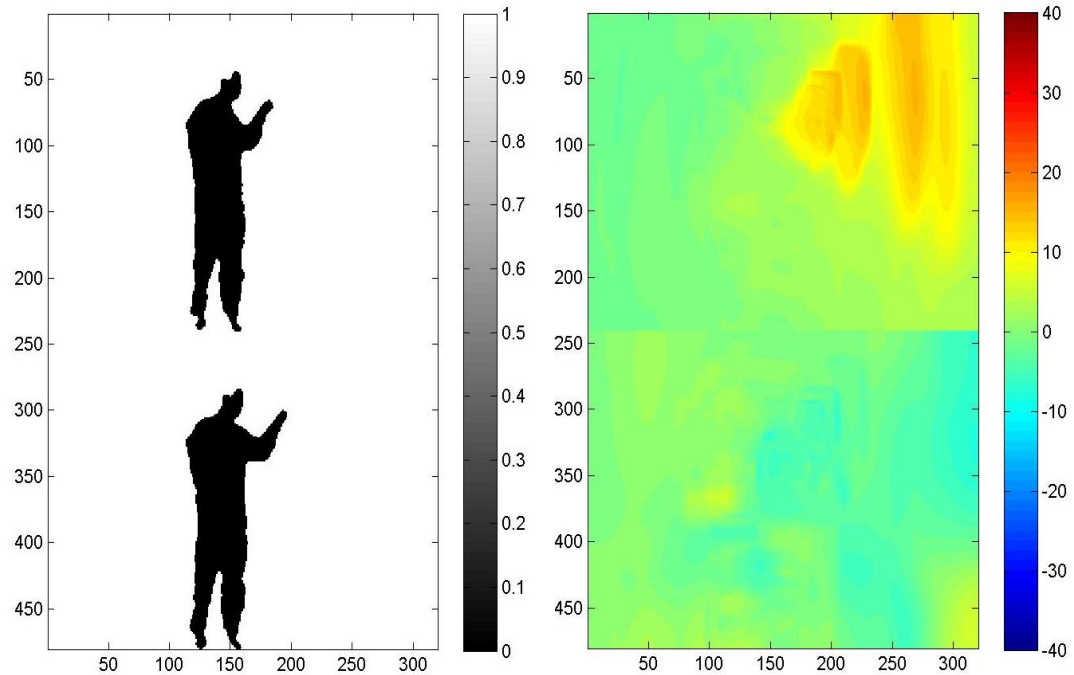

Figure 3.25: Side boxing action subject1, episode1, optical flow for frame 21&22

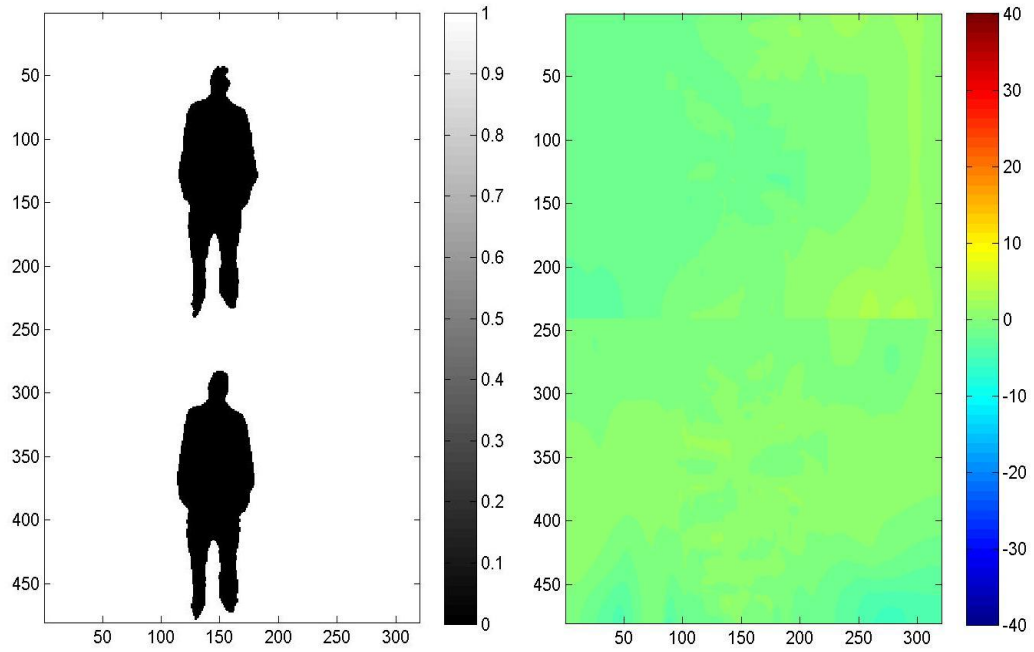

Figure 3.26: Side kick action subject1, episode1, optical flow for frame 1&2

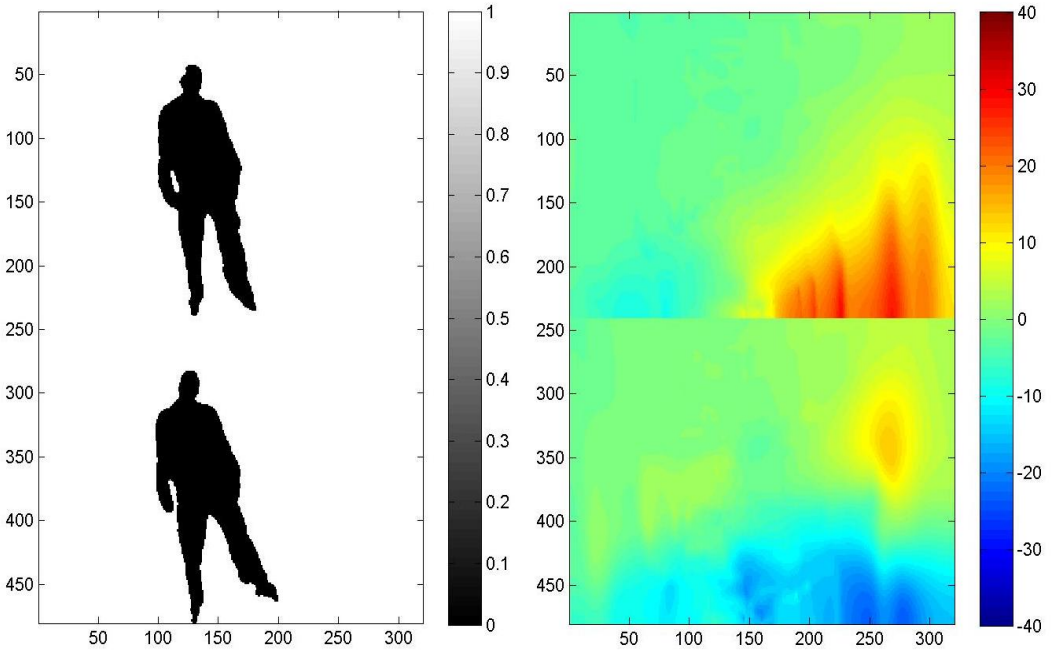

Figure 3.27: Side kick action subject1, episode1, optical flow for frame 9&10

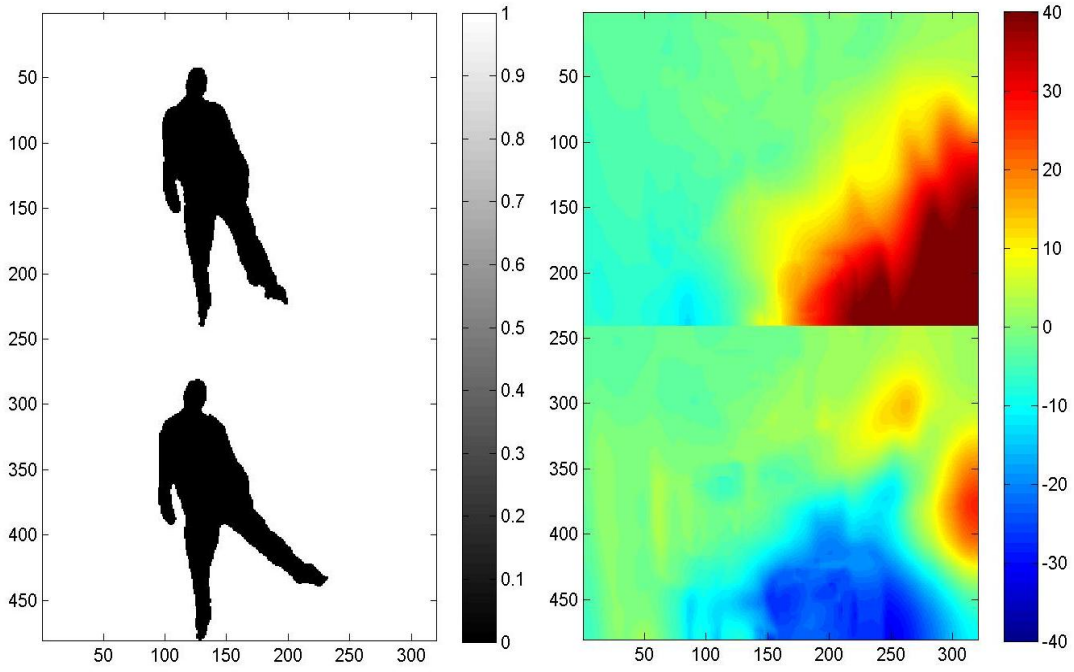

Figure 3.28: Side kick action subject1, episode1, optical flow for frame 10&11

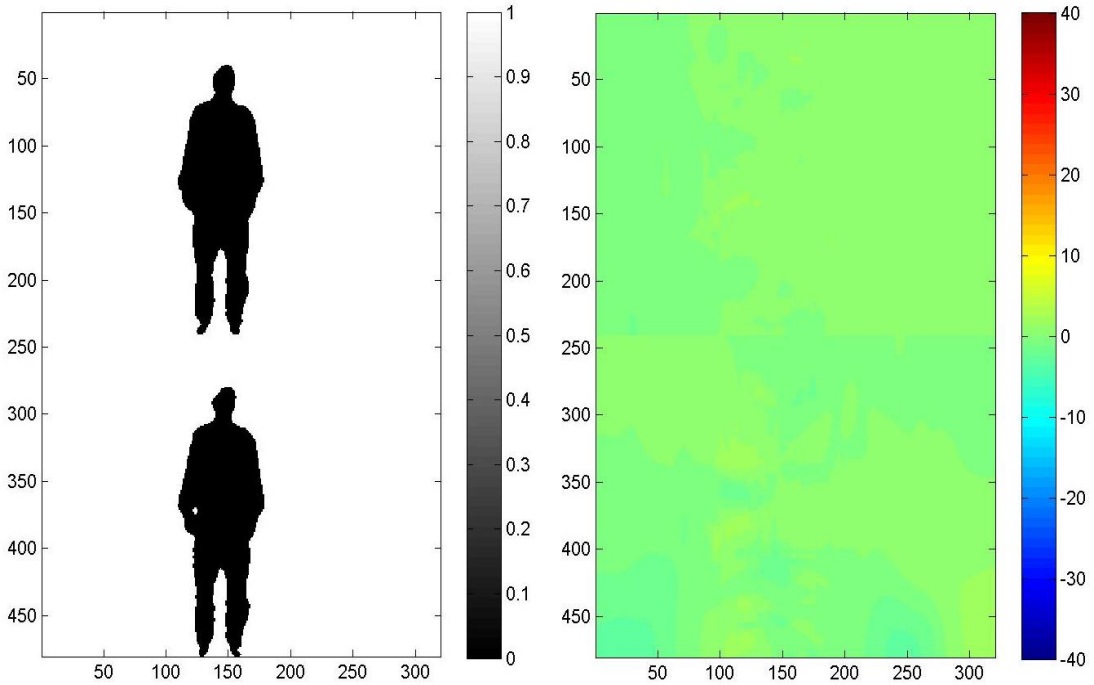

Figure 3.29: Tennis serve action subject1, episode1, optical flow for frame 1&2

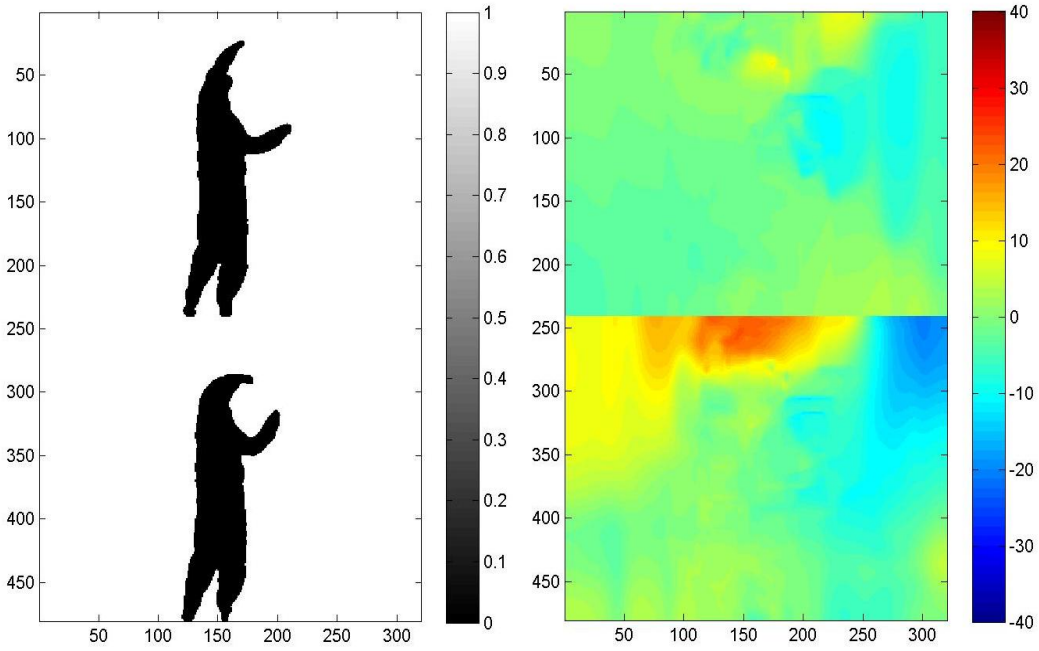

Figure 3.30: Tennis serve action subject1, episode1, optical flow for frame 20&21

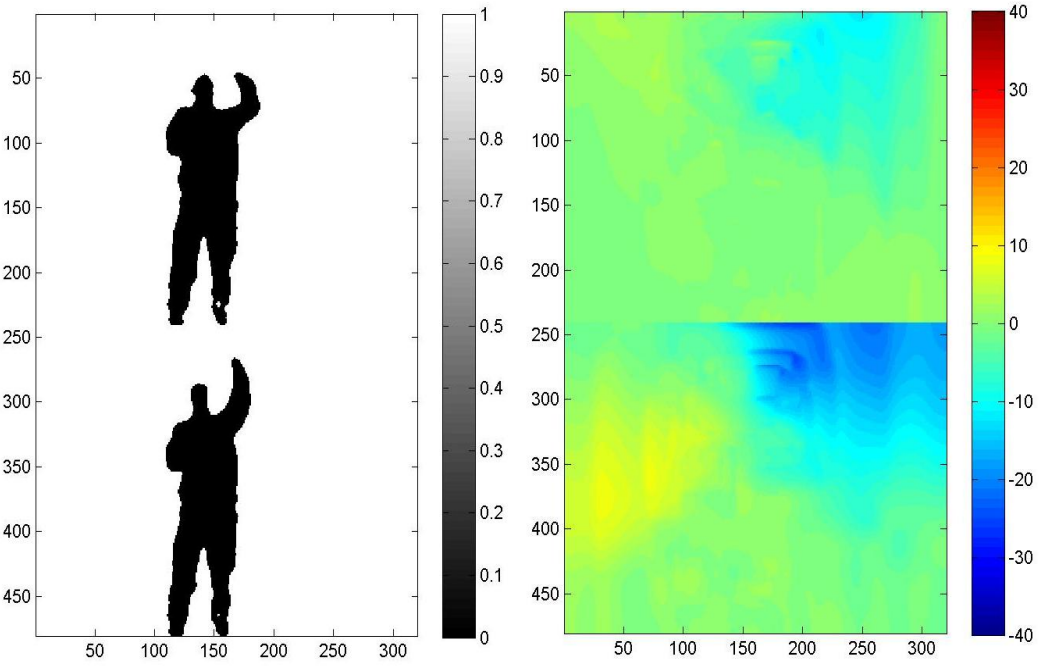

Figure 3.31: Tennis serve action subject1, episode1, optical flow for frame 28&29

In Equation 3.4, the matrix is showing silhouette flow values for action i, subject j, episode k, frame f.

$$SF_{x-y}a_i s_j e_k = SF_{x-y}(a_i, s_j, e_k, f, :) = \left[SF_{x-y,V_x}(a_i, s_j, e_k, f) \quad SF_{x-y,V_y}(a_i, s_j, e_k, f)\right] \tag{3.4}$$

3.2.3.2. Average of Silhouette Flows (AoSF)

After optical flow motion vectors are calculated for each consecutive frame, totally (frame number-1) silhouette flow matrixes are recorded. This operation is applied for each frontal, top and side view for all dataset. Some of calculated optical values of Vx and Vy are positive while some of them are negative. These values are grouped depending on positive or negative. At the end of this categorization, 4 different features are obtained as Vx pos, Vx neg, Vy pos, Vy neg from 2 motion vectors Vx and Vy. After this partitioning, it is needed to find the average of these values. For each video, the values of Vx pos, Vx neg, Vy pos, Vy neg are summed separately and calculate overall sum. That means, (number of frames-1) matrixes should be added consecutively and overall sum matrix is achieved for these 4 properties. These extracted optical flow sum matrix is used for calculating the average. How to average the optical flow values being positive or negative are given in Equation 3.5- Equation 3.8.

$$AoS_{x-y,V_x-pos}a_i, s_j, e_k = \frac{\sum_{f\in frames}[SF_{x-y,V_x}(a_i,s_j,e_k,f)>0]}{N_{x-y,V_x-pos}} \tag{3.5}$$

$$AoS_{x-y,V_x-neg}a_i, s_j, e_k = \frac{\sum_{f\in frames}[SF_{x-y,V_x}(a_i,s_j,e_k,f)<0]}{N_{x-y,V_x-neg}} \tag{3.6}$$

$$AoS_{x-y,V_y-pos}a_i, s_j, e_k = \frac{\sum_{f\in frames}[SF_{x-y,V_x}(a_i,s_j,e_k,f)>0]}{N_{x-y,V_y-pos}} \tag{3.7}$$

$$AoS_{x-y,V_y-neg}a_i, s_j, e_k = \frac{\sum_{f\in frames}[SF_{x-y,V_x}(a_i,s_j,e_k,f)<0]}{N_{x-y,V_y-neg}} \tag{3.8}$$

where N_{x-y,V_x-pos} is the # of frames in frontal silhouette flow of Vx, that has positive value, N_{x-y,V_x-neg} is the # of frames in frontal silhouette flow of Vx, that has negative value, N_{x-y,V_y-pos} is the # of frames in frontal silhouette flow of Vy, that has positive value, N_{x-y,V_y-neg} is the # of frames in frontal silhouette flow of Vy, that has negative value and where "[ ]" has value 1 if the condition inside is true, 0 otherwise.

According to the silhouette flow values being positive or negative, another matrix is obtained as follows:

$$AoS_{x-y}a_i, s_j, e_k = \begin{bmatrix} AoS_{x-y,V_x-pos}(a_i, s_j, e_k) \\ AoS_{x-y,V_x-neg}(a_i, s_j, e_k) \\ AoS_{x-y,V_y-pos}(a_i, s_j, e_k) \\ AoS_{x-y,V_y-neg}(a_i, s_j, e_k) \end{bmatrix} \quad (3.9)$$

Some frontal view AoSF examples for different features (Vx_pos, Vx_neg, Vy_pos, Vy_neg) are given for side boxing action (Action12) in Figure 3.32 and Figure 3.33.

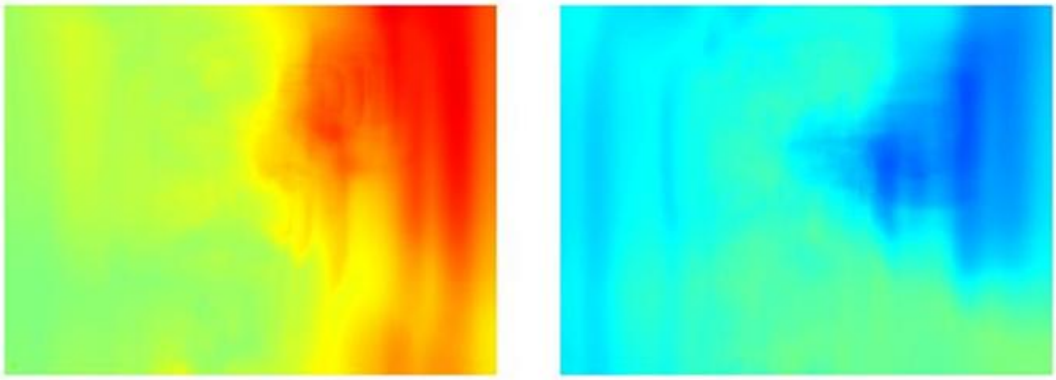

Figure 3.32: Subject1, episode1 for side boxing action Vx_pos (left) and Vx_neg (right) AoSF examples

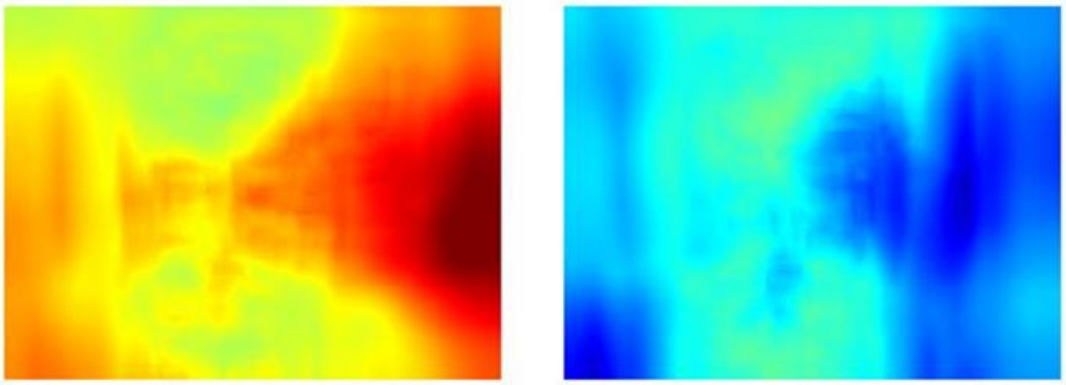

Figure 3.33: Subject1, episode 1 for side boxing action Vy_pos (left) and Vy_neg (right) AoSF examples

In addition to obtain the average of silhouette flow values, active blocks should be found and same block separation method should be done as mentioned in FEM1. Each block is chosen as 80*80 dimensions exactly same as first proposed method. Since there are four different type features, each block has 80*80*4= 25600 features. Moreover, as four blocks exist in each view there are 25600*4=102400 features. This feature number is only for one view. Totally $102400 * 3 = 307200$ features exist for frontal, top and side views. In the end of this method, the found matrix has dimension (600*307200) and this feature vector is input for classification part.

3.3. Action Classification

For the classification of actions SVM classifier is used. Each feature map which are obtained by different feature extraction methods, is fed into a multi class linear SVM that is implemented by using open source library, LIBSVM [20]. In our case there are 20 actions so 20 classes.

In human action classification part, different kinds of matrixes are generated according to the needed test subsets. Then using these matrixes, experiments are performed in various ways which will be explained in next chapter.

By comparing the human action recognition accuracies, it was decided to which proposed feature extraction method is more efficient. After the comparison of experimental results the third feature extraction method, which is the one extracting AoSF performed best as shown in the next chapter and it is chosen for our proposed method.

CHAPTER 4

EXPERIMENTAL RESULTS

In this chapter, the experimental results obtained by the proposed method on MSR Action3D dataset is presented and compared with the results available on the same dataset in literature. In first part of this chapter, MSR Action3D dataset is described. In the second part, experimental settings are defined. In the third part, performance measure of our proposed method is given based on human action recognition result accuracies. At the end of this chapter, timing requirements are investigated for different experimental settings.

4.1.Dataset

In order to evaluate our proposed method, MSR Action3D dataset [6] dataset is used. This dataset comprises of twenty actions. Each action is executed by different ten subjects. While some subjects were performing each action for two times, the others performed for three times. Totally 567 sequences exist, each one includes depth and skeleton joints for this dataset. 10 sequences are not valid in this dataset since the skeletons were either missing or wrong as explained by the authors in [2].

Subjects were fronting the camera during data recording. The actions were chosen to cover various movements of arms, legs, torso and their combinations. Moreover, if an action is performed by a single arm or leg, this is the right arm or leg. The data were recorded as binary depth files with the help of a depth sensor. The frame rate of depth files is 15 frames per second. The size of a depth map is 320x240 pixels. The first value of binary depth file is the total number of frames for the related video,

second and third values of binary depth file are image sizes and the other values are rest of the data.

The Table 4.1 shows the number of existing videos for all actions and subjects. There would be three videos for each subject doing action if data were complete. For some subjects there is no data collected for some actions, as indicated by 0 in the table, for some others data is collected for 2 episodes, indicated by 2 in the table.

Table 4.1: Number of videos for all actions and subjects

Action names	**action /subject**	**S1**	**S2**	**S3**	**S4**	**S5**	**S6**	**S7**	**S8**	**S9**	**S10**
High arm wave	**A1**	3	3	3	0	3	3	3	3	3	3
Horizontal arm wave	**A2**	3	3	3	0	3	3	3	3	3	3
Hammer	**A3**	3	3	3	0	3	3	3	3	3	3
Hand catch	**A4**	3	3	2	0	3	3	3	3	3	3
Forward punch	**A5**	3	3	3	0	3	2	3	3	3	3
High throw	**A6**	3	3	3	0	3	2	3	3	3	3
Draw X	**A7**	3	3	3	2	3	3	2	3	3	3
Draw tick	**A8**	3	3	3	3	3	3	3	3	3	3
Draw circle	**A9**	3	3	3	3	3	3	3	3	3	3
Hand clap	**A10**	3	3	3	3	3	3	3	3	3	3
Two hand wave	**A11**	3	3	3	3	3	3	3	3	3	3
Side boxing	**A12**	3	3	3	3	3	3	3	3	3	3
Bend	**A13**	3	3	3	3	3	3	3	3	3	3
Forward kick	**A14**	3	3	3	3	3	3	3	3	3	3
Side kick	**A15**	3	2	0	0	0	3	3	3	3	3
Jogging	**A16**	3	3	3	3	3	3	3	3	3	3
Tennis swing	**A17**	3	3	3	3	3	3	3	3	3	3
Tennis serve	**A18**	3	3	3	3	3	3	3	3	3	3
Golf swing	**A19**	3	3	3	3	3	3	3	3	3	3
Pick up & throw	**A20**	3	3	3	3	3	3	3	3	3	3

4.2.Experimental Settings

For comparisons of the results we obtained, the studies in the literature employing the MSR Action3D dataset are considered. However, the number of instances used in some studies is unclear. Many authors have compared their experimental results with Li et al. according to ten subjects [5, 14]. Wang et al. [13] described the dataset

as made up of 402 sequences. For the sake of clarity, this mistake is stated at [2]. It is explained 10 sequences out of the 567 are not used since the skeletons of these sequences are either missing or too erroneous. Therefore, the dataset is eventually composed of actually 557 sequences. As a consequence, it is very difficult to confirm whether these works use 402, 557 or 567 samples as it is not clear sure whether the authors are aware of these key aspects concerning the dataset, or if those are only naive text mistakes. Moreover, the missing information concerning the number of instances prevents to make a fair comparison between different methods. Regarding the experimentation method used by many authors working with the MSR Action3D dataset, it is worth to mention that there is a lack of agreement.

In the paper by Li et al. [6] where the dataset was firstly presented, three tests are performed: 1/3, 2/3 and cross-subject test (CrSub). In the first two tests, 1/3 and 2/3 of the instances are respectively used as training samples and the rest as testing samples. In the third test, half of the subjects are used for training and the remainder for testing. However, it is not described which instances or subjects are actually used in each partition of the dataset. In this study, we assume that the 1/3 means to split the dataset using the one of the episodes of each action performed by each subject for training, and to use the remaining 2 episodes for testing. The same is assumed for the 2/3. We call the tests "Test1" for 1/3 and "Test2" for 2/3 in [6] for our experiments.

Also, there exists the same conflict for the cross-subject test. It is not clearly mentioned which instances are used for training and which one testing. Any half of the 10 subjects can be used for training, e.g. 1, 2, 3, 9 and 10; and the remainder for testing, i.e. 4, 5, 6, 7 and 8. Since it is not clear which instances are used, each researcher is free to interpret anything, thereby comparing different methods where a distinct methodology has been used for the experimentation. However, this is not desirable when to compare and decide which method performs better.
In the CrSub test employed by Li et al. [6] the samples for subjects 1, 3, 5, 7 and 9 are used for training, whereas actors 2, 4, 6, 8 and 10 are used for validation. While

some authors use the mentioned settings for their training and validation sets, other authors use subjects 1-5 for training and 6-10 for validation. In our experiments, we call "TestA" for subject 1,2,3,4,5 are used training, others as testing, while we call "TestB" for subject 1,3,5,7,9 as training and others as testing.

In order to facilitate a fair comparison for Test1, Test2 and CrSub tests with state of works, we follow the same experimental settings as [1] to split 20 action categories into three subsets as listed in Table 4.2. This was due to the high computational cost of dealing with the overall dataset. Most of the papers working with MSR Action3D dataset have also used the same setting.

Table 4.2: Actions subsets used in our cross subject test, Test1, Test2

Action Set1 (AS1)		**Action Set2 (AS2)**		**Action Set3 (AS3)**	
actions	**action definition**	**actions**	**action definition**	**actions**	**action definition**
A2	horizontal arm wave	**A1**	high arm wave	**A6**	high throw
A3	hammer	**A4**	hand catch	**A14**	forward kick
A5	forward punch	**A7**	draw x	**A15**	side kick
A6	high throw	**A8**	draw tick	**A16**	jogging
A10	hand clap	**A9**	draw circle	**A17**	tennis swing
A13	bend	**A11**	two hand wave	**A18**	tennis serve
A18	tennis serve	**A12**	side-boxing	**A19**	golf swing
A20	pickup&throw	**A14**	forward kick	**A20**	pickup&throw

In addition to Test1, Test2 and CrSub tests in the literature, we also defined action set complete (AS-C) test and performed the following experimental protocol in our experiments.

Let AS-C= $\{A_8 - A_{14}, A_{16} - A_{20}\}$ which is the subset containing 12 actions for which the data is complete, that is there are exactly 3 videos of each subject for each action.

0/3 Test

Leave-one-out is repeated

for each subject S_m m=1,2,…10

with

TrainingSet0/3(S_m) = {F(a_i, : , :) - F(a_i, S_m , :) | a_i ∈ AS-C}

TestSet0/3(S_m) = {F(a_i, S_m , :) | a_i ∈ AS-C}

1/3 Test

Leave-one-out is repeated

for each subject S_m m=1,2,…10

for each episode e_n n=1,2,3

with

Training Set1/3(S_m)= TrainingSet0/3 U {F(a_i, S_m , e_n) | a_i ∈ AS-C}

TestSet1/3(S_m)= TestSet0/3-{F(a_i, S_m , e_n) | a_i ∈ AS-C}

2/3 Test

Leave-one-out is repeated

for each subject S_m m=1,2,…10

for each episode e_n n=1,2,3

with

Training Set2/3(S_m)= F(a_i, : , :) - F(a_i, S_m , e_n) | a_i ∈ AS-C}

TestSet2/3(S_m)= {F(a_i, S_m , e_n) | a_i ∈AS-C}

Consequently, we conduct AS-C, Test1, Test2 and CrSub tests in our experiments. AS-C test is performed on the 12 actions for which the data is complete. As for each subset AS1, AS2 and AS3, there are three different tests, i.e. Test1, Test2 and CrSub test. In Test One, 1/3 of the subset is used as training the rest as testing; in Test Two, 2/3 of the subset is used as training and the rest as testing; in cross subject test, half subjects are used for training and the rest ones used for testing. For CrSub test, also we define 2 tests as “TestA” and “TestB” according to which subjects used as training and others for testing.

4.3.Performance Measure

Firstly we conducted the experiments for 0/3, 1/3 and 2/3 tests on the action set AS-C according to the protocol we defined in the previous section. In 0/3 test on AS-C there are twelve actions, ten subjects and three videos per each subject. All data is 240 videos. There is no missing data. %90 of data is used for training and %10 of data is used for test. This means, 9 subjects are used for training and 1 subject is used for test. The experimental results for 0/3 test for the FEM1, FEM2, FEM3 are given in Table 4.3.

Table 4.3: Human action recognition rates (%) for FEM1, FEM2, FEM3 for 0/3 Test on action set ASC

Test	**FEM1(AoS) 0/3 Test**	**FEM2(AoSD) 0/3 Test**	**FEM3(AoSF) 0/3 Test**
S1	91.66	97.33	97.33
S2	91.66	91.66	97.33
S3	91.66	91.66	91.66
S4	83.33	91.66	91.66
S5	86.11	86.11	91.66
S6	55.55	55.55	55.55
S7	91.66	91.66	97.22
S8	73.67	75.00	75.00
S9	91.66	97.22	97.22
S10	86.11	91.66	86.11
Average	**84.31**	**86.95**	**88.07**

1/3 Test on ASC regards one video for training and two videos for testing of three episodes in twelve actions again. This experiment is performed for all subjects three times according to leave one out technique. The experimental results for 1/3 test for the FEM1, FEM2, FEM3 are demonstrated in Table 4.4, Table 4.5 and Table 4.6.

Table 4.4: Human action recognition rates (%) for FEM1 for 1/3 Test on action set ASC (1 in train 2 in test)

	FEM1(AoS) 1/3 Test			
subject	$e_n = e_1$	$e_n = e_2$	$e_n = e_3$	**Avg**
S1	91.67	91.67	95.83	93.05
S2	87.5	87.5	91.67	88.88
S3	83.33	79.17	83.33	81.94
S4	100	100	100	100
S5	91.67	95.83	87.5	91.66
S6	95.83	100	100	98.61
S7	100	100	100	100
S8	95.83	83.33	100	93.05
S9	100	95.83	95.83	97.22
S10	95.83	100	95.83	97.22
Average				**94.16**

Table 4.5: Human action recognition rates (%) for FEM2 for 1/3 Test on action set ASC (1 in train 2 in test)

	FEM2(AoSD) 1/3 Test			
subject	$e_n = e_1$	$e_n = e_2$	$e_n = e_3$	**Avg**
S1	100	91.67	95.83	95.83
S2	95.83	100	95.83	97.22
S3	79.17	79.17	83.33	80.56
S4	100	100	100	100
S5	100	83.33	91.67	91.67
S6	100	100	100	100
S7	100	100	100	100
S8	95.83	83.33	100	93,05
S9	100	95.83	95.83	97.22
S10	91.67	100	95.83	95.83
Average				**95.14**

Table 4.6: Human action recognition rates (%) for FEM3 for 1/3 Test on action set ASC (1 in train 2 in test)

	FEM3(AoSF) 1/3 Test			
subject	$e_n = e_1$	$e_n = e_2$	$e_n = e_3$	**Avg**
S1	95.83	91.67	100	95.83
S2	95.83	100	95.83	97.22
S3	83.33	87.5	91.67	87.5
S4	95.83	100	95.83	97.22
S5	95.83	91.67	100	95.83
S6	100	95.83	100	98.61
S7	95.83	95.83	100	97.22
S8	95.83	87.5	100	94.44
S9	100	95.83	95.83	97.22
S10	100	100	100	100
Average				**96.10**

2/3 Test on ASC regards two videos for training and one video for testing of three episodes in twelve actions. This experiment is performed by three times according to leave one out technique. All experimental results for 2/3 test for the FEM1, FEM2, FEM3 are shown in Table 4.7, Table 4.8 and Table 4.9.

Table 4.7: Human action recognition rates (%) for FEM1 for 2/3 Test on action set ASC (2 in train 1 in test)

	FEM1(AoS) 2/3 Test			
subject	$e_n = e_1$	$e_n = e_2$	$e_n = e_3$	**Avg**
S1	100	100	91.67	97.22
S2	91.67	100	91.67	94.44
S3	83.33	91.67	100	91.67
S4	100	100	100	100
S5	83.33	100	100	94.44
S6	100	100	100	100
S7	100	100	100	100
S8	100	91.67	100	97.22
S9	100	100	91.67	97.22
S10	91.67	100	100	97.22
Average				**96.94**

Table 4.8: Human action recognition rates (%) for FEM2 for 2/3 Test on action set ASC (2 in train 1 in test)

	FEM2(AoSD) 2/3 Test			
subject	$e_n = e_1$	$e_n = e_2$	$e_n = e_3$	**Avg**
S1	100	100	91.67	97.22
S2	91.67	100	100	97.22
S3	91.67	83.33	100	91.67
S4	100	100	100	100
S5	91.67	100	100	97.22
S6	100	100	100	100
S7	100	100	100	100
S8	100	91.67	100	97.22
S9	100	100	91.67	97.22
S10	91.67	100	100	97.22
Average				**97.50**

Table 4.9: Human action recognition rates (%) for FEM3 for 2/3 Test on action set ASC (2 in train 1 in test)

	FEM3(AoSF) 2/3 Test			
subject	$e_n = e_1$	$e_n = e_2$	$e_n = e_3$	**Avg**
S1	100	91.67	100	97.23
S2	91.67	100	100	97.23
S3	91.67	91.67	91.67	91.67
S4	100	100	100	100
S5	100	100	100	100
S6	100	100	100	100
S7	100	91.67	100	97.23
S8	100	91.67	100	97.23
S9	100	100	91.67	97.23
S10	100	100	100	100
Average				**97.70**

All these experiments results are summarized and variance values are also given in table Table 4.10. FEM3 gives the best human action recognition rates comparing to other feature extraction methods FEM1 and FEM2. Because of these successful results in this experiment with the CrSub, Test1, Test2 experiment results given later we consider the FEM3 as our proposed method. In Table 4.10, 0/3 test gives the lower recognition rates for some subjects, especially subject6 and subject8. However, in 1/3 test and 2/3 test these recognition rates are increasing sharply since some data of these subjects can be erroneous in 0/3 test while data is correct in other tests. Moreover, 1/3 test gives the lower rates compared to 0/3 test for some subjects, because for some data the used set is decreasing the rates and recognition rate changes. Also, these changes in recognition rates can be raised from the scaling issue based on the length and height of subjects.

Table 4.10: Human action recognition rates (%) for all tests on action set ASC

	FEM1(AoS)			**FEM2(AoSD)**			**FEM3(AoSF)**		
subject	**0/3 Test**	**1/3 Test**	**2/3 Test**	**0/3 Test**	**1/3 Test**	**2/3 Test**	**0/3 Test**	**1/3 Test**	**2/3 Test**
S1	91.66	93.05	97.22	97.33	95.83	97.22	97.33	95.83	97.23
S2	91.66	88.88	94.44	91.66	97.22	97.22	97.33	97.22	97.23
S3	91.66	81.94	91.67	91.66	80.56	91.67	91.66	87.5	91.67
S4	83.33	100	100	91.66	100	100	91.66	97.22	100
S5	86.11	91.66	94.44	86.11	91.67	97.22	91.66	95.83	100
S6	55.55	98.61	100	55.55	100	100	55.55	98.61	100
S7	91.66	100	100	91.66	100	100	97.22	97.22	97.23
S8	73.67	93.05	97.22	75.00	93,05	97.22	75.00	94.44	97.23
S9	91.66	97.22	97.22	97.22	97.22	97.22	97.22	97.22	97.23
S10	86.11	97.22	97.22	91.66	95.83	97.22	86.11	100	100
Average	**84.31**	**94.16**	**96.94**	**86.95**	**95.14**	**97.50**	**88.07**	**96.10**	**97.70**
Variance	**121.79**	**29.26**	**6.87**	**145.36**	**30.95**	**5.32**	**160.98**	**10.34**	**5.86**

According to the test sets defined in Table 4.2, other experiments are performed in accordance with studies in literature. As for each subset, there are three different tests, i.e. Test1, Test2 and CrSub. In CrSub test; half subjects are used for training and the rest ones used for testing. While some authors use subjects 1-5 for training and 6-10 for validation, which we called “TestA”, other authors [21, 22] use the mentioned settings for subjects 1-3-5-7-9 for training and 2-4-6-8-10 for validation, which we called “TestB”. According to our proposed different feature extraction methods, which are AoS (FEM1), AoSD (FEM2) and AoSF (FEM3), the comparisons of TestA and TestB for CrSub test experimental results are given in Table 4.11, while for Test1, Test2 tests experimental results are given in Table 4.12. According to these experimental results FEM3 gives the better recognition rates comparing to other methods FEM1 and FEM2, thus it is chosen as proposed method.

Table 4.11: Comparison of human action recognition rates (%) for different feature extraction methods (FEM1, FEM2, and FEM3) for CrSub test

	CrSub			**CrSub**		
Method	**TestA**			**TestB**		
	AS1	**AS2**	**AS3**	**AS1**	**AS2**	**AS3**
FEM1	72.88	55.46	66.38	65.09	67.56	60.71
FEM2	78.81	77.31	82.35	89.62	77.47	81.25
FEM3	82.35	79.15	93.45	90.76	82.98	91.70

Table 4.12: Comparison of human action recognition rates (%) for different feature extraction methods (FEM1, FEM2, and FEM3) for Test1, Test2

	Test1			**Test2**		
Method						
	AS1	**AS2**	**AS3**	**AS1**	**AS2**	**AS3**
FEM1	95.71	88.46	94.07	98.64	96	95
FEM2	95	92.5	95.25	93.75	92.5	97.5
FEM3	96.25	93.70	95.50	95.80	98.80	98.80

Table 4.13: Comparison of human action recognition rates (%) for CrSub test

	Cross Subject Tests							
Set	**Li et al. 2010 [6] TestB**	**Xia et al. 2012 [5] TestB**	**Yang et al. 2014 [14] TestB**	**Vieira et al. 2012 [9] TestB**	**Oreifej et al. 2013 [21] TestA**	**Rahmani et al. 2014 [22] TestA**	**ours TestA**	**ours TestB**
AS1	72.9	87.98	74.5	84.70	-	-	82.35	**90.76**
AS2	71.9	**85.48**	76.1	81.30	-	-	79.15	82.98
AS3	79.2	63.46	**96.4**	88.40	-	-	93.45	91.70
Average	74.7	78.97	82.33	84.80	88.89	**90.9**	84.98	**88.48**

The CrSub test results are given for two different test settings TestA and TestB in Table 4.13. For AS1 set, our experimental result for Test B outperforms the other results in literature. Moreover if we look for the average values of the AS1, AS2 and AS3 TestB results, our recognition rate is also more successful than the other researches. In addition, we compete with the state of the arts for TestA. However our recognition rates are not successful like TestA, since our proposed method gives the lower recognition rates when first 5 subjects are used for training.

While in Test1, 1/3 of the subset is used as training the rest as testing; in Test2, 2/3 of the subset is used as training and the rest as testing; if data is missing for a subject for an action this subject is not considered while average performance is calculated. Test1 and Test2 results are given with the other test results in literature [5, 6, 12, 23] in Table 4.14. Our recognition rates are more successful than the others for AS2 and AS3 sets of Test2; however our other results also can compete with the state of the arts.

Table 4.14: Comparison of human action recognition rates (%) for Test1 and Test2

	Test1, Test2				
Set	**Li et al. 2010 [6]**	**Lu et al. 2012 [5]**	**Yang et al. 2012 [23]**	**Vieira et al. 2012 [12]**	**ours**
Test1					
AS1	89.50	**98.50**	94.70	96	96,25
AS2	89.00	**96.70**	95.40	95	93.70
AS3	96.30	93.50	97.30	**97.5**	95.50
Test2					
AS1	93.40	**98.60**	97.30	98	95.80
AS2	92.90	97.20	98.70	97	**98.80**
AS3	96.30	94.90	97.30	98.50	**98.80**

4.4.Timing Requirements

Experiments are performed on laptop DELL 8 GB RAM, Intel I5 3317U @1.7GHz processor and time requirements during the processes are investigated. For CrSub, Test1 and Test2 test procedures, the time requirements for classifications are given in this section.

Moreover, 80*80 blocks sizes stated in the third chapter are reduced by applying subsampling on the AoSF blocks explained in Chapter3. The block sizes after subsampling and the number of AoSF features per episode is given in Table 4.15. The numbers of features are found from block size x 4 blocks x 3 views x 4 feature per pixel. One example is illustrated of subsampling size 16 for 80*80 block size in Figure 4.1.

Table 4.15: Subsample size and feature number table

Subsample size	**Block size**	**#of blocks**	**# of views**	**# of feature per pixel**	**Feature number**
1	80*80	4	3	4	307200
2	40*40	4	3	4	76800
4	20*20	4	3	4	19200
8	10*10	4	3	4	4800
16	5*5	4	3	4	1200
80	1*1	4	3	4	48

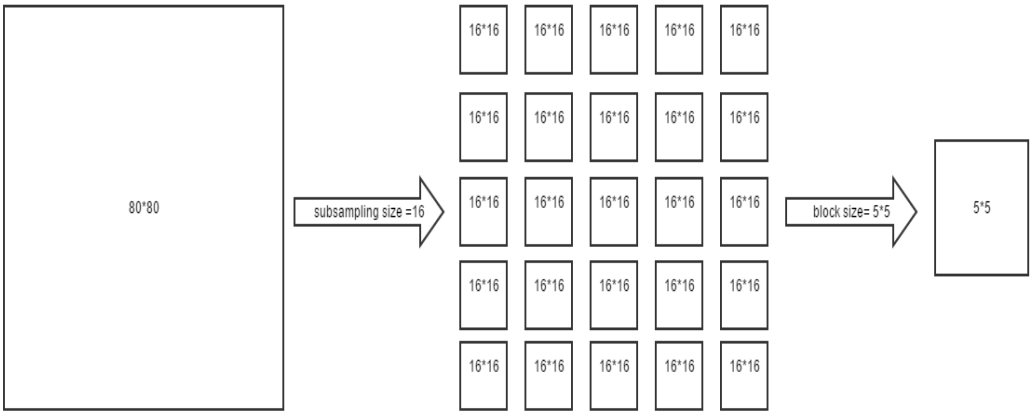

Figure 4.1: Example of subsampling size 16

The time requirements for AoSF features with respect to feature number after subsampling for CrSub TestA &TestB are given detail in Table 4.16. When feature number decreases, the time requirements also decrease sharply.

Table 4.16: Time requirements (sec) for AoSF features with respect to feature number after subsampling for CrSub tests, TestA and Test B

	CrSub time requirements (sec)					
Feature number	**TestA**			**TestB**		
	AS1	**AS2**	**AS3**	**AS1**	**AS2**	**AS3**
307200	20.663	22.854	20.702	25.765	26.460	24.620
76800	4.562	5.271	4.557	5.840	5.653	5.612
19200	1.210	1.355	1.163	1.506	1.464	1.379
4800	0.313	0.361	0.333	0.752	0.491	0.474
1200	0.131	0.115	0.105	1.726	0.176	0.206
48	0.054	0.073	0.079	1.546	0.070	0.068

The time requirements for AoSF features with respect to feature number after subsampling for CrSub TestA &TestB are shown in Figure 4.2-Figure 4.3 respectively.

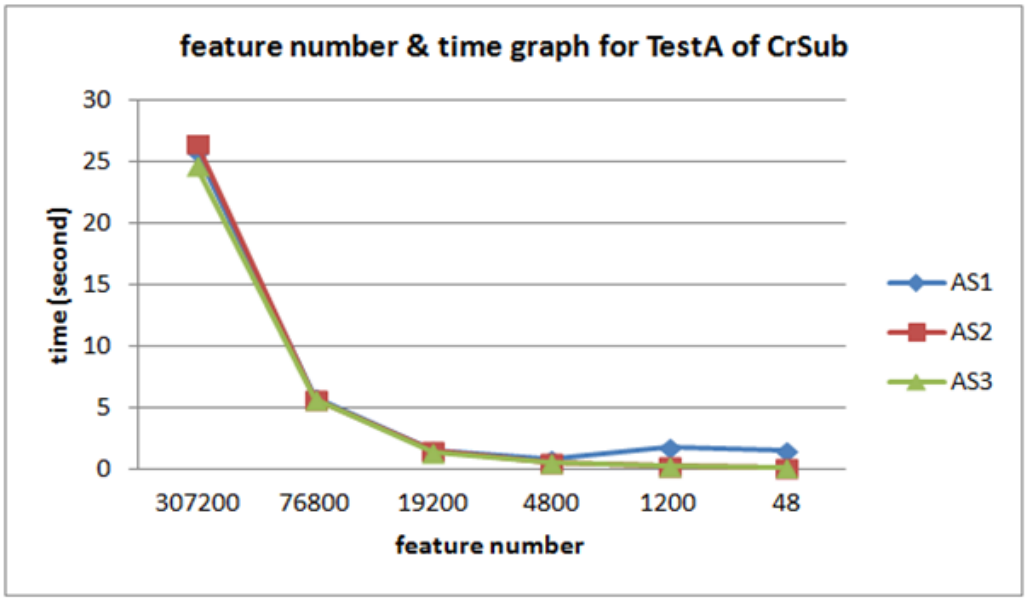

Figure 4.2: Time requirements (sec) for classification of AoSF features with respect to feature number after subsampling for CrSub test, TestA

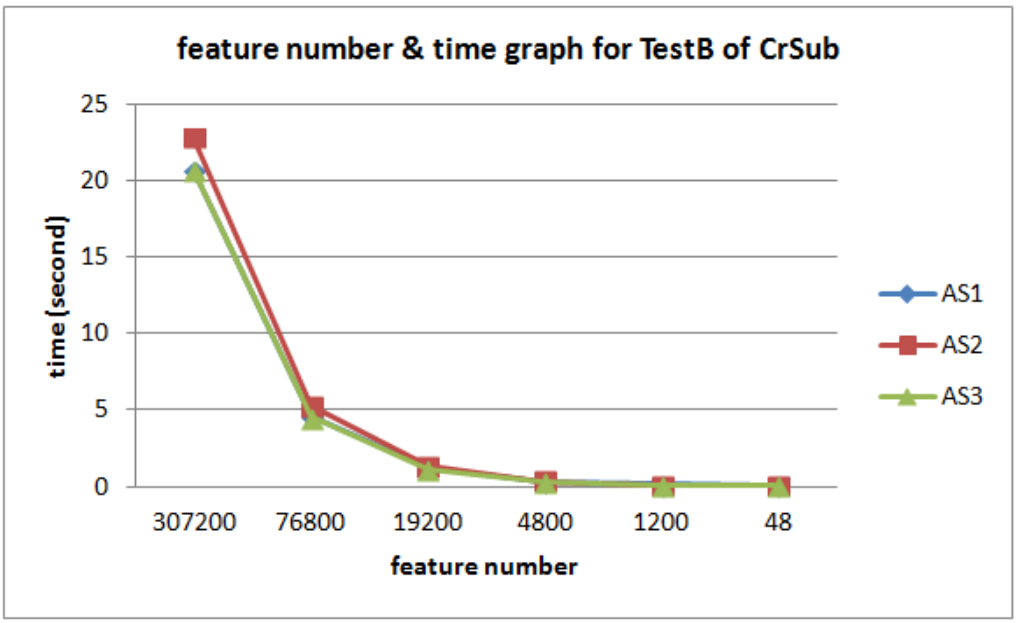

Figure 4.3: Time requirements (sec) for classification of AoSF features with respect to feature number after subsampling for CrSub test, TestB

Human action recognition rate analysis for AoSF features with respect to feature number after subsampling for CrSub test is given detail in Table 4.17. While feature number lessens, recognition rates are approximately stable until the minimum feature number value 48. However, if the enhancement in timing requirements is considered, this small decrease in recognition rate can be acceptable.

Table 4.17: Human action recognition rates (%) for AoSF features with respect to feature number after subsampling for CrSub tests, TestA and TestB

	CrSub recognition rates (%)					
Feature number	**TestA**			**TestB**		
	AS1	**AS2**	**AS3**	**AS1**	**AS2**	**AS3**
307200	81.36	78.15	92.43	90.56	82.88	91.07
76800	81.36	78.15	92.43	90.56	82.88	91.07
19200	81.36	77.31	92.43	90.56	82.88	91.07
4800	81.36	77.31	92.43	90.56	81.98	91.07
1200	82.20	76.47	92.43	90.56	81.98	91.07
48	77.96	65.54	89.92	85.85	76.58	88.40

The human action recognition rates for AoSF features with respect to feature number after subsampling for CrSub, TestA, TestB are shown in Figure 4.4 and Figure 4.5 respectively.

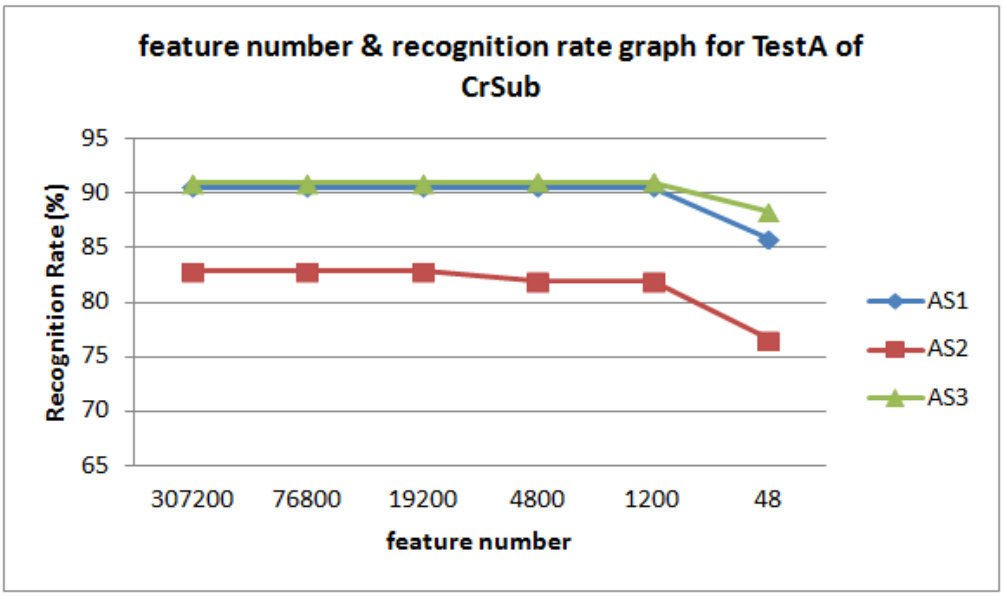

Figure 4.4: Human action recognition rates (%) for AoSF features with respect to feature number after subsampling for CrSub TestA

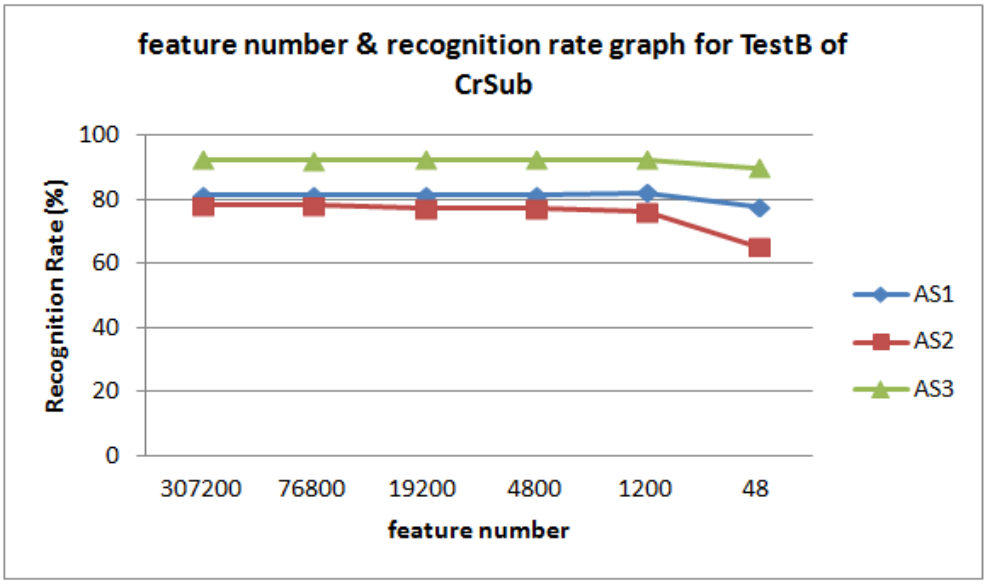

Figure 4.5: Human action recognition rates (%) for AoSF features with respect to feature number after subsampling for CrSub TestB

The time requirements for AoSF features with respect to feature number after subsampling for Test1 and Test2 are given detail in Table 4.18. When feature number decreases, the time requirements also decrease sharply.

Table 4.18: Time requirements (sec) for AoSF features with respect to feature number after subsampling for Test1 and Test2

	Time requirements (sec)					
Feature number	**Test1**			**Test2**		
	AS1	**AS2**	**AS3**	**AS1**	**AS2**	**AS3**
307200	32.31	36.55	34.75	41.88	44.22	40.45
76800	7.59	7.15	8.03	9.52	9.99	9.39
19200	1.78	1.77	1.61	1.98	2.10	1.91
4800	0.60	0.50	0.54	0.55	0.57	0.51
1200	0.34	0.17	0.15	0.14	0.15	0.14
48	0.04	0.06	0.05	0.05	0.05	0.04

To sum up, feature number can be decreased in order to enhance time requirements, if we consider the time requirements for CrSub, Test1 and Test2 tests. Moreover, human action recognition rates are approximately same for different feature numbers until the minimum one, so we can decrease the future number conveniently and robustly.

CHAPTER 5

CONCLUSION

It is worth to mention that the goal of this work is to classify actions only using raw depth maps without any additional information such as skeleton joint information or optical video. In literature, the most of the 3D human action recognition methods are based on skeleton joints data and depth maps are used seldom. In this thesis a new and effective method that we call silhouette flows is proposed for 3D human action recognition by using only depth map sequences.

The method proposed in this thesis constitutes two steps: feature and classification. The novelty of the method lies in the feature extraction part, in which motion features are extracted by using optical flow vector fields calculated on silhouettes in frontal, top and side views over each frame of 3D sequence. After these flow vectors are obtained, averages are prepared according to the motion vector values separately for negative and positive values for each frame of each plane. In order to recognize various human behaviors, each frame in video is divided into some meaningful blocks. According to the significant motion blocks, the final motion feature is obtained. Then, these motion features are given to the SVM classification system and the results are investigated.

In order to justify the proposed method, its performance is examined on the publicly available MSR Action3D dataset and compared with the performances the methods in literature evaluated on the same dataset. It should be mentioned that although these methods that we examined are using the same data set, there might be a mismatch on the number of samples and validation methods used by most of these

studies. The lack of information in these papers about how they split the dataset into training and validation sets has led to a lot of confusion. Therefore, a fair comparison is not possible since the experiments cannot be reproduced exactly as they are conducted in these studies. Thus, in this work we have tried to clear up the existing confusion. This may enable to improve future comparisons and increase the awareness of the need of clarifying experimental settings.

After all the validation methods in these papers are reviewed, we decide to use the "Cross Subject Test", "Test1", "Test2". Also we define the "AS-C (action set complete) Test" as another validation method. In cross subject test, it is considered all possible splits of the dataset for action recognition within subject tests and two different combinations of using 5 subjects for training and the remaining 5 for testing. In Test1, 1/3 of the instances are used for training samples and the rest as testing samples, while 2/3 of the instances are used for training samples and the rest as testing samples in Test2. AS-C test is experienced on the complete action data, which means there is no erroneous or missing part in these data. Experiments showed that the silhouette flows method that we proposed achieves quite successful results on the challenging MSR Action3D dataset and also competes the methods available in literature in most of the cases.

As future work, instead of 4 specific flow features we plan to investigate some other issues in feature extraction part. For instance, after the optical flow part, magnitude and phase components can be calculated for each action. These magnitude and phase components can be used as a feature by considering in three dimensional Cartesian planes. Moreover, behalf 4 specific flow features in two orthogonal directions for positive and negative values for each of the 3 orthogonal planes as proposed in this thesis, it may be enough to use a combined feature showing the size and angle of the flow in 3D.

Moreover, we also plan to investigate how some other classification methods affect our results. As a classification method, deep learning techniques can be applied and the experimental results could be observed and compared to SVM.

REFERENCES

[1] Z. Zhang, “Microsoft Kinect Sensor and Its Effect”, MultiMedia, IEEE, 2012, pp. 4-10.

[2]Microsoft Research. MSR Action Recognition Datasets, http://research.microsoft.com/enus/um/people/zliu/ActionRecoRsrc, last accessed on 23.12.2015.

[3] B. Ni, G. Wang and P. Moulin, “RGBD-HuDaAct: A color-depth video database for human daily activity recognition”, Computer Vision Workshops (ICCV Workshops) IEEE International Conference, 2011, pp. 1147-1153.

[4] J. Sung, C. Ponce, B. Selman and A. Saxena, “Unstructured human activity detection from RGBD images”, Robotics and Automation (ICRA) IEEE International Conference, 2012, pp. 842-849.

[5] L. Xia, C. Chen, and J. K. Aggarwal, “View invariant human action recognition using histograms of 3D joints”, Computer Vision and Pattern Recognition Workshops (CVPRW) IEEE Computer Society Conference, 2012, pp. 20-27.

[6] W. Li, Z. Zhang, and Z. Liu ,“Action recognition based on a bag of 3D points”, Computer Vision and Pattern Recognition Workshops (CVPRW) IEEE Computer Society Conference, 2010, pp. 9-14.

[7] D. Weinland, R. Ronfard and E. Boyer, “A survey of visionbased methods for action representation, segmentation and recognition”, Computer Vision and Image Understanding, 2011, pp. 224-241.

[8] L. Chen, H. Wei and J. Ferryman, “A survey of human motion analysis using depth imagery”, Pattern Recognition Letters, 2013, pp. 1995-2006.

[9] A.W. Vieira, E.R. Nascimento, G.L. Oliveira, Z. Liu, and M.F Campos, “Stop: Space-time occupancy patterns for 3D action recognition from depth map sequences”, Progress in Pattern Recognition, Image Analysis, Computer Vision and Application, 2012, pp. 252-259.

[10] J. Wang, Z. Liu, J. Chorowski, Z. Chen, and Y. Wu, “Robust 3D action recognition with random occupancy patterns”, Computer Vision, ECCV, 2012, pp 872-885.

[11] X. Yang, C. Zhang, and Y. Tian, “Recognizing actions using depth motion maps-based histograms of oriented gradients” , ACM International Conf. on Multimedia, 2012.

[12] T. Dobhal, V.Shitole, G. Thomas and G. Navada, “Human Activity Recognition using Binary Motion Image and Deep Learning”, Second International Symposium on Computer Vision and the Internet VisionNet’15, 2015, pp. 178-185.

[13] X. Yang and Y. Tian, “Effective 3D action recognition using Eigen joints”, Journal Of Visual Communication and Image Representation, 2014, pp 2-11.

[14] J. Wang, Z. Liu, Y. Wu, and J. Yuan, “Mining actionlet ensemble for action recognition with depth cameras”, Computer Vision and Pattern Recognition (CVPR) IEEE Conference, 2012, pp. 1290- 1297.

[15] P. O'Donovan, "Optical Flow: Techniques and Applications", 2005.

[16] S.S. Beauchemin and J.L. Barron, "The computation of optical flow", 1996.

[17] B.D. Lucas and T. Kanade, "An iterative image registration technique with an application to stereo vision", Proc 7th Intl Joint Conf on Artifical Intelligance (IJCAI), 1981, pp.674-679.

[18]B.K.P. Horn and B.G. Schunk, "Determining optical flow", Artifical Intellegence , 1981, pp.185-203.

[19]P.Dollar, "Piotr's Computer Vision Matlab Toolbox", http://vision.ucsd.edu/~pdollar/toolbox/doc/, last accessed on 23.12.2015.

[20] C. Chang and C. Lin, "LIBSVM: A Library for Support Vector Machines", ACM Transactions on Intelligent Systems and Technology (TIST), 2011.

[21] O. Oreifej and Z. Liu, "HON4D: Histogram of Oriented 4D Normals for Activity Recognition from Depth Sequences," Computer Vision and Pattern Recognition, Computer Vision and Pattern Recognition (CVPR) IEEE Conference, 2013, pp. 716-723.

[22] H. Rahmani, A. Mahmood, D. Huynh, A. Mian, "Action Classification with Locality-Constrained Linear Coding.", Pattern Recognition (ICPR), 22nd International Conference , 2014, pp.3511-3516.

[23] Y. Xiaodong, and Y. Tian, "Eigenjoints-based action recognition using naive-bayes-nearest-neighbor", Computer Vision and Pattern Recognition Workshops (CVPRW) IEEE Computer Society Conference, 2012, pp. 14-19.