ACTION RECOGNITION IN CONTINUOUS DATA STREAMS USING FUSION OF DEPTH AND INERTIAL SENSING

by

Neha Dawar

APPROVED BY SUPERVISORY COMMITTEE:

____

Dr. Nasser Kehtarnavaz, Chair

____

Dr. Mehrdad Nourani

____

Dr. Carlos Busso

____

Dr. Nicholas Gans

Copyright 2018

Neha Dawar

All Rights Reserved

To my beloved family

ACTION RECOGNITION IN CONTINUOUS DATA STREAMS USING FUSION OF DEPTH AND INERTIAL SENSING

by

NEHA DAWAR, B.TECH, MS

DISSERTATION

Presented to the Faculty of

The University of Texas at Dallas

in Partial Fulfillment

of the Requirements

for the Degree of

DOCTOR OF PHILOSOPHY IN

ELECTRICAL ENGINEERING

THE UNIVERSITY OF TEXAS AT DALLAS

December 2018

ACKNOWLEDGMENTS

Firstly, I would like to express my deep gratitude to my advisor, Dr. Nasser Kehtarnavaz, for his unwavering support, guidance, and encouragement throughout this research work. Without his enlightening comments and suggestions, this work could not have been achieved. I sincerely thank my committee members, Dr. Mehrdad Nourani, Dr. Carlos Busso, and Dr. Nicholas Gans, for their valuable feedback and insightful discussions during my dissertation proposal that allowed me to improve my dissertation work.

I would like to thank my labmates and friends for their help during this research. I would particularly like to thank my dearest friend and mentor, Fatemeh Saki, for all the technical discussions, great advice and encouragement throughout the course of my research. I am also grateful to my friends, Daud Khan, Ashwin Baral, Abhishek Sehgal, and Sumit Jha, for their wonderful friendship during these years.

Finally, my deepest appreciation goes to my family for their immense love and support. They have always motivated me to stay focused and be dedicated to my work. I am highly grateful to them for their patience and their trust in me.

November 2018

ACTION RECOGNITION IN CONTINUOUS DATA STREAMS USING FUSION OF DEPTH AND INERTIAL SENSING

Neha Dawar, PhD
The University of Texas at Dallas, 2018

Supervising Professor: Dr. Nasser Kehtarnavaz

Human action or gesture recognition has been extensively studied in the literature spanning a wide variety of human-computer interaction applications including gaming, surveillance, healthcare monitoring, and assistive living. Sensors used for action or gesture recognition are primarily either vision-based sensors or inertial sensors. Compared to the great majority of previous works where a single modality sensor is used for action or gesture recognition, the simultaneous utilization of a depth camera and a wearable inertial sensor is considered in this dissertation. Furthermore, compared to the great majority of previous works in which actions are assumed to be segmented actions, this dissertation addresses a more realistic and practical scenario in which actions of interest occur continuously and randomly amongst arbitrary actions of non-interest. In this dissertation, computationally efficient solutions are presented to recognize actions of interest from continuous data streams captured simultaneously by a depth camera and a wearable inertial sensor. These solutions comprise three main steps of segmentation, detection, and classification. In the segmentation step, all motion segments are extracted from continuous action streams. In the detection step, the segmented actions are separated into actions of interest and actions of non-interest. In the classification step, the detected actions of interest are classified. The features considered include skeleton joint positions, depth motion maps, and statistical attributes of acceleration and angular velocity inertial signals. The classifiers considered include maximum entropy Markov model, support vector data description, collaborative representation classifier, convolutional neural network, and long short-term memory network. These solutions are applied to the two applications of smart TV hand gestures and transition movements for home healthcare monitoring. The results obtained indicate the effectiveness of the developed solutions in detecting and recognizing actions of interest in continuous data streams. It is shown that higher recognition rates are achieved when fusing the decisions from the two sensing modalities as compared to when each sensing modality is used individually. The results also indicate that the deep learning-based solution provides the best outcome among the solutions developed.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1

INTRODUCTION

The subject of human gesture or action recognition has been extensively studied in the literature spanning many human-computer interaction applications such as gaming, visual surveillance, health monitoring and assistive living. Gesture or action recognition involves automatic detection and recognition of human gestures or actions by a computer program based on data acquired from a sensor.

Different sensors have been utilized to perform gesture or action recognition. The sensors used are primarily either vision-based sensors, including conventional video cameras as well as depth cameras, or inertial sensors. In the great majority of action recognition solutions presented in the literature, either a vision sensor or an inertial sensor is used. It is well-established that there are limitations associated with an individual sensing modality when operating under realistic conditions. For instance, video cameras are sensitive to lighting conditions, camera position, and pose privacy issues. Depth cameras, e.g., Microsoft Kinect, have a limited field of view and are sensitive to ambient light. Inertial sensors, which incorporate accelerometers and gyroscopes, generate inertial signals that are dependent on their placement on the body and suffer from inertial drift during prolonged hours of operation. In general, under realistic operating conditions, there is no single sensing modality that can cope with various real-world situations that occur in practice. Furthermore, low inter-class variations and high intra-class variations of the actions performed for a particular application pose challenges for a single modality sensing approach. Consequently, there has been a growing interest in using fusion of different sensing modalities to achieve more robust human action or gesture recognition under realistic operating conditions. The scope of this dissertation involves the fusion of the two sensing modalities of a depth camera and a wearable inertial sensor.

Most existing works on action or gesture recognition involve recognizing actions or gestures that are segmented into a single action, that is when the start and the end of an action are already known or manually identified. In practice, detection and recognition of actions of interest need to be carried out within continuous action streams that contain actions of interest occurring randomly amongst arbitrary actions of non-interest. For example, when a person performs a smart TV hand gesture in the middle of a random set of actions, such a gesture is not segmented, and it is far more challenging to detect and recognize it as compared to the situations in which this action or gesture is manually segmented and recognition is carried out based on the segmented action.

To detect actions of interest in continuous action streams, it is first required to segment all actions, both actions of interest and actions of non-interest. Then, the segmented actions are to be detected or labelled as actions of interest or actions of non-interest. Finally, the labelled actions of interest need to be classified. Based on the fusion of the two differing sensing modalities of a depth camera and a wearable inertial sensor, this dissertation aims at addressing the more realistic and challenging problem of human action or gesture recognition when actions occur continuously and randomly among actions of non-interest. In essence, the novelty or contribution of this dissertation lies in the development of computationally efficient solutions to detect and recognize actions of interest in continuous action streams using fusion of information from a depth camera and a wearable inertial sensor. The gestures and actions examined in this dissertation include recognition of hand gestures in the smart TV application and recognition of body transition movements in the home healthcare monitoring application.

The chapters of this dissertation are organized based on three journal and four conference papers that have already been published, with each chapter or paper addressing the dissertation objective from a different perspective. Each chapter provides an abstract, an introduction, the methodology developed, a discussion of the results obtained, and a conclusion.

More specifically, Chapter 2 presents an approach to detect and recognize actions of interest in continuous action streams using skeleton joint positions that are obtained from a depth camera and inertial signals that are obtained from a wearable inertial sensor. First, an initial decision about an action being an action of interest and its classification is made by skeleton joint positions, and then inertial signals are used to remove false detections.

Chapter 3 discusses a detection and recognition approach using skeleton joint positions and depth images. Detection is performed using skeleton joint positions based on a one-class classifier to identify a segmented action as an action of interest or an action of non-interest. Then, recognition of detected actions of interest is performed using two collaborative representation classifiers, one operating on skeleton joint positions and the other operating on depth images.

Chapter 4 provides a more general approach to the ones developed in Chapters 2 and 3 where both of the sensing modalities of depth camera and inertial sensor are used to perform detection and recognition in parallel. Fusion is conducted for both detection and recognition. The first fusion is carried out for detection by filtering out or discarding the actions of interest that are detected by just one of the two sensing modalities. The second fusion is carried out on the recognition outcomes of the two sensing modalities. The application of the smart TV gestures is considered in this chapter.

Chapter 5 targets the data flow synchronization aspects of running in real-time the fusion system covered in Chapter 4 on a modern laptop.

Chapter 6 introduces a deep learning–based sensing fusion system to detect and monitor transition movements between body states as well as falls for the home healthcare monitoring application. The fusion system developed detects and recognizes actions of interest in continuous action streams.

Chapter 7 presents the effect of data augmentation on the outcome of the deep learning-based approach covered in Chapter 6 by examining three different datasets.

Chapter 8 brings the approaches presented in the previous chapters under one most effective solution towards addressing the main theme of this dissertation. This chapter presents a deep learning-based sensing fusion solution to detect and recognize actions of interest in continuous action streams. A convolutional neural network (CNN) is used for depth images along one path and a CNN+LSTM (long short-term memory) network is used for inertial signals along another path to perform detection and recognition. A decision-level fusion is then performed on the outcomes of the two paths.

Finally, Chapter 9 summarizes the dissertation contributions and states possible future extensions.

CHAPTER 2

REAL-TIME CONTINUOUS ACTION DETECTION AND RECOGNITION USING DEPTH IMAGES AND INERTIAL SIGNALS[*]

Authors- Neha Dawar, Chen Chen, Roozbeh Jafari, Nasser Kehtarnavaz

The Department of Electrical and Computer Engineering, EC33

The University of Texas at Dallas

800 West Campbell Road

Richardson, Texas 75080-3021

[*] ©(2017) IEEE. Reprinted, with permission, from (Neha Dawar, Chen Chen, Roozbeh Jafari, and Nasser Kehtarnavaz, "Real-time continuous action detection and recognition using depth images and inertial signals", Proceedings of 26th IEEE International Symposium on Industrial Electronics (ISIE), pp. 1342-1347, Edinburgh, June 2017.)

ABSTRACT

This chapter presents an approach to detect and recognize actions of interest in real-time from a continuous stream of data that are captured simultaneously from a Kinect depth camera and a wearable inertial sensor. Actions of interest are considered to appear continuously and in a random order among actions of non-interest. Skeleton depth images are first used to separate actions of interest from actions of non-interest based on pause and motion segments. Inertial signals from a wearable inertial sensor are then used to improve the recognition outcome. A dataset consisting of simultaneous depth and inertial data for the smart TV actions of interest occurring continuously and in a random order among actions of non-interest is studied and made publicly available. The results obtained indicate the effectiveness of the developed approach in coping with actions that are performed realistically in a continuous manner.

2.1 INTRODUCTION

Human action recognition is an extensively researched topic with a wide span of applications such as human-computer interaction, gaming, and rehabilitation. Different sensors or sensing modalities have been used for human action recognition including video cameras, depth cameras, and inertial sensors. For example, video camera images were used in [1] to perform human action recognition using 3D SIFT (Scale Invariant Feature Transform) based descriptors; depth camera images were used in [2] to conduct human action recognition using depth motion maps; and skeleton data from depth cameras were used in [3] to characterize different human actions. In [4-7], human action recognition was performed by using temporal and statistical features of inertial signals acquired from wearable inertial sensors. In [8-11], the data from both a depth camera and an inertial sensor were used simultaneously to achieve human action recognition with high accuracy.

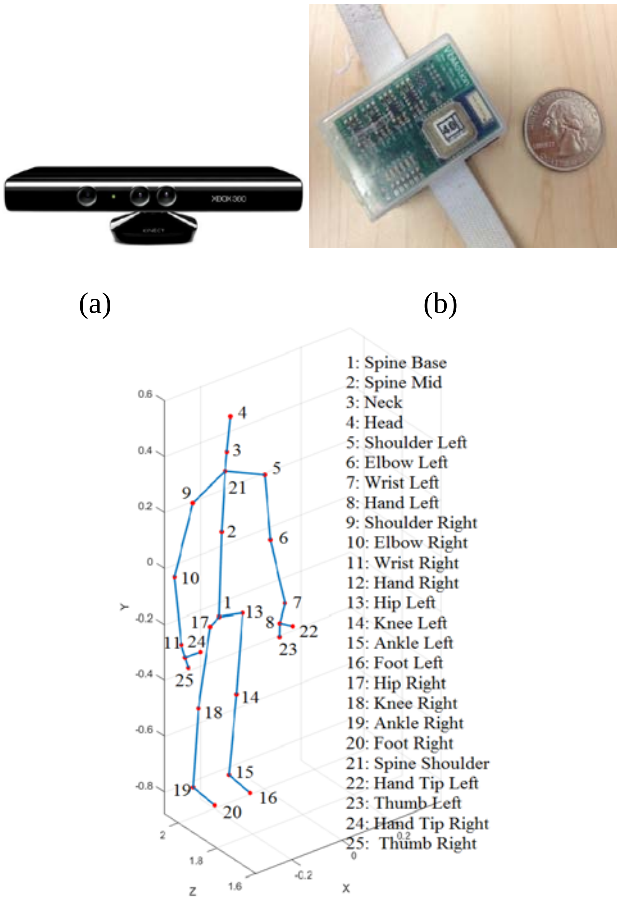
It is important to note that the bulk of the research on action recognition has involved recognizing actions that are segmented into single actions. That is to say, data to process contain only a single action of interest. However, in practice, in applications such as smart TV and gaming, one needs to deal with recognizing an action of interest in real-time when there is a continuous stream of activities by a subject. In such cases, it becomes more challenging to accurately detect and recognize actions of interest. Our objective in this chapter is to address this more challenging problem, that is to say, data to process are not segmented single actions but non-segmented actions of interest that appear continuously and in random order among actions of non-interest. Hence, in this work, both the detection and recognition processes are addressed within the same framework. First, actions of interest are separated from actions of non-interest and then the detected actions of interest are classified or recognized in real-time. In [12], a continuous recognition approach was discussed using only a depth camera, however, the continuous dataset used contained only actions of interest with no actions of non-interest randomly occurring between actions of interest.

This chapter presents an approach for real-time detection and recognition of actions of interest from a continuous data stream of activity by simultaneous utilization of a depth camera and an inertial sensor. Such data streams are considered to contain actions of interest randomly occurring among some arbitrary actions of non-interest. There are two main attributes that distinguish this work from the previous works on action recognition: (i) compared with the scenario of performing only actions of interest, a more realistic scenario of continuous activity is considered where actions of interest are performed continuously and in a random order among actions of noninterest, and (ii) a depth camera and an inertial sensor are used simultaneously for such a scenario.

The remainder of the chapter is organized as follows: Section 2.2 provides a description of the two differing sensor modalities used. Section 2.3 describes the continuous dataset collected and examined for the experiments. The details of the approach developed are then provided in Section 2.4 followed by the results and their discussion in Section 2.5. Finally, the chapter is concluded in Section 2.6.

2.2 SENSORS UTILIZED

The sensors used in the developed approach include a Kinect v2 depth camera and a wearable inertial sensor. The Kinect camera is a depth camera that is widely used for human action recognition. A picture of this camera is shown in Figure 2.1(a). The Kinect SDK [13], which is a publicly available software package, provides 3D spatial positions of 25 skeleton body joints that are derived from depth images. Figure 2.1(c) shows the skeleton joints that Kinect v2 generates from captured depth images. As will be explained in Section 2.4, our approach uses these joint


(a)

(b)

(c)

Figure 2.1. (a) Kinect depth camera (b) wearable inertial sensor (c) human body skeleton joints obtained by Kinect v2

positions to segment a continuous stream of activity into pauses and motions as a prelude to action recognition.

The wearable inertial sensor used is a small wireless body sensor discussed in [14]. A picture of this inertial sensor is shown in Figure 2.1(b). The sensor generates 3-axis acceleration and 3-axis angular velocity signals, which are wirelessly transmitted to a laptop via a Bluetooth link. It is worth mentioning that although wearing multiple inertial sensors on different parts of the body can increase the robustness of the system, due to the intrusiveness and thus the practicality aspect associated with wearing multiple inertial sensors, only one inertial sensor is used in this work.

2.3 CONTINUOUS ACTIONS DATASEST

Since the aim of this work is the recognition of some actions of interest from a continuous stream of activity, and given that there is no publicly available dataset that provides both a continuous and simultaneous stream of depth and inertial data, we have put together such a continuous dataset for the wrist actions involved in smart TV gestures. The dataset incorporates continuous streams of data that are simultaneously collected from the two differing modality sensors mentioned in Section 2.2. The smart TV gestures include ‘Waving a Hand’, ‘Flip to Left’, ‘Flip to Right’, ‘Counterclockwise Rotation’ and ‘Clockwise Rotation’.

For data collection, the subjects performed the actions in front of a Kinect v2 camera while wearing the wearable inertial sensor on their right wrist. The data from the two sensors were synchronized by using the time stamp scheme described in [9]. For training, the subjects were asked to perform a single action of interest at a time and both depth and inertial data were recorded simultaneously. During actual operation or testing, the subjects were asked to perform the actions of interest in a continuous manner while randomly performing actions of non-interest in-between the actions of interest. Examples of actions of non-interest included picking up a water bottle, drinking water, wearing a pair of glasses, etc. As a result, a typical data stream consisted of both actions of interest and actions of non-interest appearing in a random order. Note that the subjects were given complete freedom to choose their own actions of non-interest while staying within the field of view of the Kinect camera.

Two scenarios were considered for data collection. The first scenario was done in a subject-specific manner in which the training and testing data were collected for the same subject. During training, a subject was asked to repeat each of the 5 actions of interest 15 times. For testing, continuous sets of actions were performed 6 times. In each set, the subject performed the 5 actions of interest with several actions of non-interest conducted randomly in-between the actions of interest in a continuous manner.

The second scenario was done in a subject-generic manner, i.e., training and testing data were collected from 5 different subjects. For training, the 5 subjects were asked to perform each of the 5 actions of interest 5 times, resulting in a total of 125 data streams consisting of simultaneous depth and inertial data. For testing, each subject performed the actions of interest with some actions of non-interest conducted randomly in-between the actions of interest in a continuous manner. This continuous dataset is made available for public use and can be downloaded from http://www.utdallas.edu/~kehtar/UTD-CAD-Both.htm. It is worth noting that this dataset is different than the one we previously reported in [9], which includes segmented single actions.

2.4 DEVELOPED CONTINUOUS ACTION DETECTION AND RECOGNITION APPROACH

The approach developed in this chapter relies on breaking down a continuous stream of skeleton activity data into pauses and motions, similar to the approach reported in [15-17]. A variable length Maximum Entropy Markov Model (MEMM) classifier is used in order to detect the presence of an action of interest in continuous data streams. This classifier operates similar to a Hidden Markov Model (HMM) classifier but is computationally more efficient to enable real-time operation. The acceleration and rotation signals from the wearable inertial sensor are used to remove false positive cases or to improve the recognition outcome by using a Collaborative Representation Classifier (CRC) as discussed in [18]. A block diagram of the components of the developed approach appears in Figure 2.2. Note that the detection or segmentation task only uses the skeleton data, while both the skeleton and inertial data are used for the recognition task.

2.4.1 Training

Any continuous action is described as a sequence of pauses and motions. A training model similar to the one discussed in [17] is used to segment an activity sequence into pause and motion segments. In what follows, it is discussed how the model is modified in order to deal with a continuous data stream.

Pauses and motions from skeleton data are obtained by computing the length-invariant Normalized Relative Orientation (NRO) of the joints with respect to their rotating joints as follows:

$$F_{NRO}^{i} = \frac{L_i - L_j}{\left\| L_i - L_j \right\|} \tag{2.1}$$

where L_i and L_j denote 3D locations of the i^{th} and j^{th} joints, respectively, and j is the joint about which joint i rotates and $\|\cdot\|$ denotes the Euclidean distance.

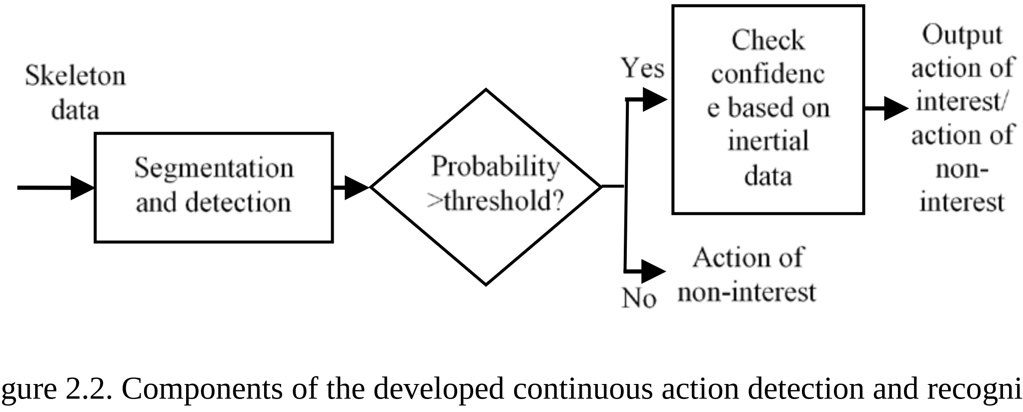

Skeleton data
Segmentation and detection
Probability >threshold?
Yes
No
Check confidenc e based on inertial data
Output action of interest/ action of non- interest
Action of non-interest

Figure 2.2. Components of the developed continuous action detection and recognition

Let $(F_1, F_2, \dots, F_t, \dots, F_n)$represent a sequence of NRO features, where $F_t = (F_{NRO}^1, F_{NRO}^2, F_{NRO}^3, \dots)_t$ indicates the NRO features of the joints at frame t. Based on a reference NRO (F_r), a so called potential energy at frame t is computed as follows:

$$PE(t) = \|F_t - F_r\|^2 \tag{2.2}$$

Then, the following potential difference at frame t is obtained:

$$PD(t) = PE(t) - PE(t-1) \tag{2.3}$$

If the potential difference of data frames becomes less than a very low value close to zero (for example, for the dataset collected in this chapter, 0.04 was found low enough to identify the start and end of all the motion segments), they are labeled as a pause segment, otherwise they are labeled as a motion segment. An example of pause and motion segments for the action 'Waving a Hand' is shown in Figure 2.3. The horizontal portions represent pause segments and varying portions represent motion segments.

Based on pause and motion segments, a codebook for pauses and motions is then set up which is used for action recognition via a variable-length MEMM classifier. Basically, an action is characterized by its sequence of motion segments.

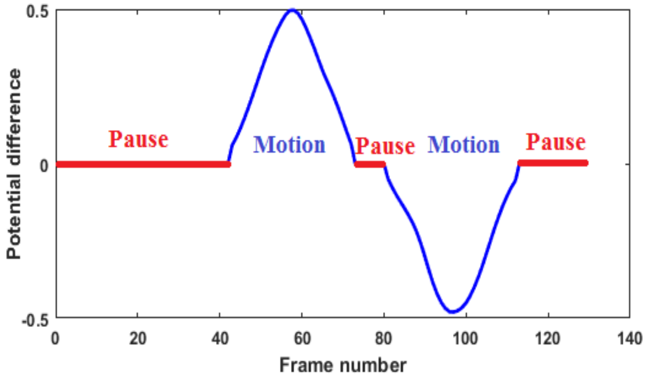


Figure 2.3. Pause and motion segments in the action 'Waving a Hand'

Similar motion segments occur in some actions of interest, for example the actions ‘Waving a Hand’ and ‘Flip to Left’ begin similarly, i.e., their first motion segments are similar. Hence, unlike [17], clustering is first applied to motion segments using a Gaussian Mixture Model (GMM) to group similar motion segments. The cluster representatives are used for action detection. The clustering is carried out by dividing each motion segment into three equal portions. Then, the averaged features from each of these three portions of motion segments are computed and used to cluster motion segments of an action into M clusters.

Next, the transition probabilities amongst the clusters are obtained. Since a pause segment is present between two motion segments, for every pair of motion clusters $MC1$ and $MC2$, the mean feature of the pause segments is obtained and stored in a so-called ‘dynamic cluster’ $DC(MC1, MC2)$. This way, a codebook of motion and pause clusters is generated from the training data.

2.4.2 Detection and Recognition

The task of continuous action detection and classification or recognition is carried out using likelihood probabilities. As the skeleton data is generated frame by frame, the corresponding NROs are generated, and potential differences are used to classify the segments as pause or motion segments. Whenever a motion segment ends, the likelihood probabilities of the motion segments are obtained for each motion cluster. The likelihood probability of a motion segment for a motion cluster m of an action n is obtained as follows:

$$D(n, m) = \left\| FM - MC_{m,n} \right\|^2 \tag{2.4}$$

$$P_{motion}(n,m) = \frac{1/_{D(n,m)}}{\sum_n \sum_m 1/_{D(n,m)}} \tag{2.5}$$

where FM denotes the feature vector corresponding to the three portions of a motion segment and $MC_{m,n}$ denotes the feature vector of the m^{th} motion cluster of the n^{th} action.

Similarly, the likelihood of a pause segment to lie between the motion clusters $m1$ and $m2$ of an action n is obtained as follows:

$$C(n,m1,m2) = \|FP - PC_n(m1,m2)\|^2 \tag{2.6}$$

$$P_{pause}(n,m1,m2) = \frac{1/_{C(n,m1,m2)}}{\sum_n \sum_{m1} \sum_{m2} 1/_{C(n,m1,m2)}} \tag{2.7}$$

where FP denotes the mean feature vector of a pause segment and $PC_n(m1,m2)$ represents the feature vector of the pause cluster associated with the motion clusters $m1$ and $m2$ of the n^{th} action. Once the likelihood probabilities are obtained, the variable-length MEMM classifier is used to assign the probability of a motion segment for each of the action classes. If the probability of a motion segment is greater than a set threshold, to be discussed later, the presence of an action of interest is indicated and the segment is assigned to the action class with the likelihood probability greater than the threshold.

2.4.3 Inertial Data

The probability threshold impacts the detection and recognition outcome for a continuous data stream. If the threshold is set too low, the rate of incorrect detections will be high, i.e., many actions of non-interest will be classified as actions of interest. On the other hand, if the threshold is set too high, actions of interest will be missed. It is therefore important to set the threshold such that the highest number of actions of interest are detected while minimizing the number of wrong detections.

The data from the inertial sensor are used to improve the detection accuracy by ruling out false detections. A CRC classifier is used here for this purpose. Whenever the likelihood probability of a particular motion segment using the skeleton data is greater than the threshold, the inertial data are used to verify the detection. Based on the CRC classifier, a residual error for each class is obtained and the segment is assigned to the class corresponding to the minimum error. If the detected action for a particular segment using the inertial data is the same as the one obtained using the skeleton data, that particular segment is considered to belong to that action, otherwise it is considered to be an action of non-interest.

2.5 RESULTS AND DISCUSSION

In this section, the results of the developed approach on continuous data streams are reported. Since one cannot match the sequence predicted in a continuous data stream to serve as the ground truth, the evaluation framework proposed in [19] is used here. That is, a true positive was flagged whenever an action was detected within a window of 4 frames from the ground truth, while a false positive was flagged when the predicted action lied outside the window of 4 frames or when the action classified did not match the ground truth. The actions in the dataset examined contained a maximum of 4 motion segments, hence the number of motion clusters per action was set to 4, or $M = 4$, for the experimentations reported below.

For each continuous data stream, the number of true positives, false positives and false negatives were found and the performance was evaluated based on F1 score discussed in [20-21]. This score is derived from the precision and recall indices that are defined as follows:

$$P = \frac{\#TP}{\#TP + \#FP} \tag{2.8}$$

$$R = \frac{\#TP}{\#TP + \#FN} \tag{2.9}$$

$$F1 = 2\frac{P \cdot R}{(P + R)} \tag{2.10}$$

where P denotes the precision index, R the recall index, $\#TP$ the number of true positives, $\#FP$ the number of false positives, and $\#FN$ the number of false negatives.

For the subject-specific scenario, the precision values, recall values and F1 scores that were obtained for different values of the probability threshold *p* with and without using the inertial data are listed in Tables 2.1, 2.2 and 2.3, respectively. As can be seen from these tables, for high threshold values, there were very few false positive detections resulting in a high value of precision. For such cases, some of the true positives were wrongly rejected when using the inertial data, which led to a drop in the recall value. Furthermore, many of the true positives could not be identified, hence the F1 score became low. As the threshold probability *p* was decreased, the F1 score and the recall value improved since more and more true positives were detected. However, upon further decreasing the threshold probability *p*, the number of true positives did not increase much but the number of false positives grew, which was reflected in the decrease in the F1 score. As can be observed from the tables, the precision values, recall values and F1 scores improved when both the skeleton and inertial data were used. Note that the improvement when using the inertial data was more for the precision values as compared to the recall values. This is because the inertial data was used for the purpose of rejecting false positives.

Table 2.1. Precision values for subject specific scenario

***p* values**	**Without Inertial**	**With Inertial**
p=0.60	85.2%	94.1%
p=0.55	82.4%	89.9%
p=0.50	80.5%	89.1%
p=0.45	74.7%	87.1%
p=0.40	64.0%	76.4%
p=0.35	51.6%	67.0%

Table 2.2. Recall values for subject specific scenario

***p* values**	**Without Inertial**	**With Inertial**
p=0.60	61.6%	61.2%
p=0.55	77.4%	75.1%
p=0.50	86.7%	84.5%
p=0.45	92.5%	93.5%
p=0.40	92.9%	96.1%
p=0.35	93.8%	97.1%

Table 2.3. F1 scores for subject specific scenario

***p* values**	**Without Inertial**	**With Inertial**
p=0.60	71.5%	74.2%
p=0.55	79.8%	81.9%
p=0.50	83.5%	86.7%
p=0.45	82.7%	90.2%
p=0.40	75.7%	85.1%
p=0.35	66.5%	79.3%

The precision values, recall values and F1 scores for the subject-generic scenario with different values of the threshold probability *p* are listed in Tables 2.4, 2.5 and 2.5, respectively. These tables show the results with and without using the inertial data.

The best performance in both the scenarios was observed at the threshold probability of *p*=0.45, and at this threshold, the improvement in the F1 score by using the inertial data was about 8% for the subject specific scenario and 2% for the subject generic scenario. Due to large variations of the same actions associated with different subjects, in general, the subject specific scenario is recommended for any practical deployment.

Table 2.4. Precision values for subject generic scenario

***p* values**	**Without Inertial**	**With Inertial**
p=0.55	95.9%	100.0%
p=0.50	93.1%	97.7%
p=0.45	85.1%	92.0%
p=0.40	68.6%	78.5%
p=0.35	55.4%	68.5%
p=0.30	43.2%	53.2%

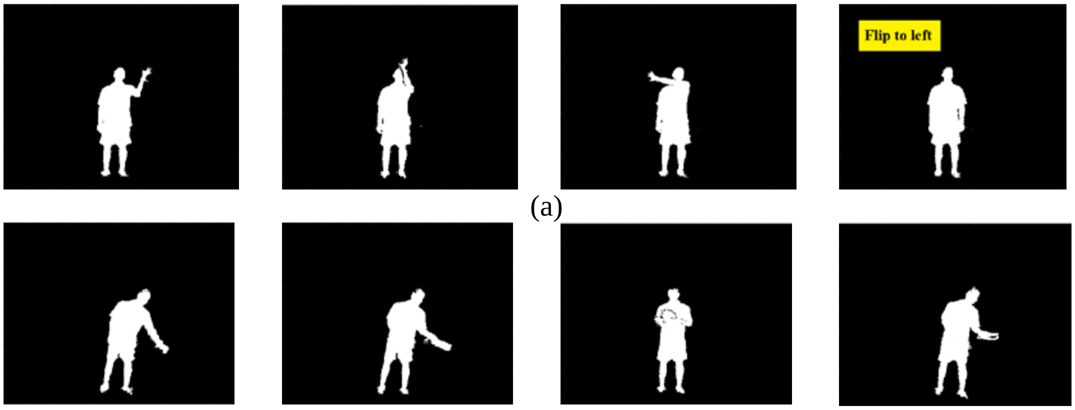
Table 2.5. Recall values for subject generic scenario

***p* values**	**Without Inertial**	**With Inertial**
p=0.55	56.8%	56.0%
p=0.50	70.8%	69.6%
p=0.45	82.4%	79.2%
p=0.40	89.2%	90.8%
p=0.35	93.2%	96.0%
p=0.30	95.6%	97.2%

Table 2.6. F1 scores for subject generic scenario

***p* values**	**Without Inertial**	**With Inertial**
p=0.55	71.3%	71.7%
p=0.50	80.4%	81.3%
p=0.45	83.7%	85.1%
p=0.40	77.5%	84.2%
p=0.35	69.5%	80.0%
p=0.30	59.6%	68.8%

An important point to note here is that action detection and recognition were performed continuously in real-time (30 frames per second). For each incoming depth image frame and inertial signals, the features were extracted and pause and motion segments were obtained in real-time. The classification was performed when a motion segment was completed. An example of the depth and inertial data for an action of interest and an action of non-interest in a test sequence is shown in Figures 2.4 and 2.5, respectively. The vertical lines in Figure 2.5 exhibit the segments associated with the actions of interest and actions of non-interest in the potential difference function. The other two graphs in this figure show the acceleration along the z-direction for an action of interest and an action of non-interest. A videoclip of the developed continuous action



(a)

(b)

Figure 2.4. (a) Snapshots of depth images from an action of interest ‘Flip to Left’ (b) an action of non-interest ‘picking up and reading a book’

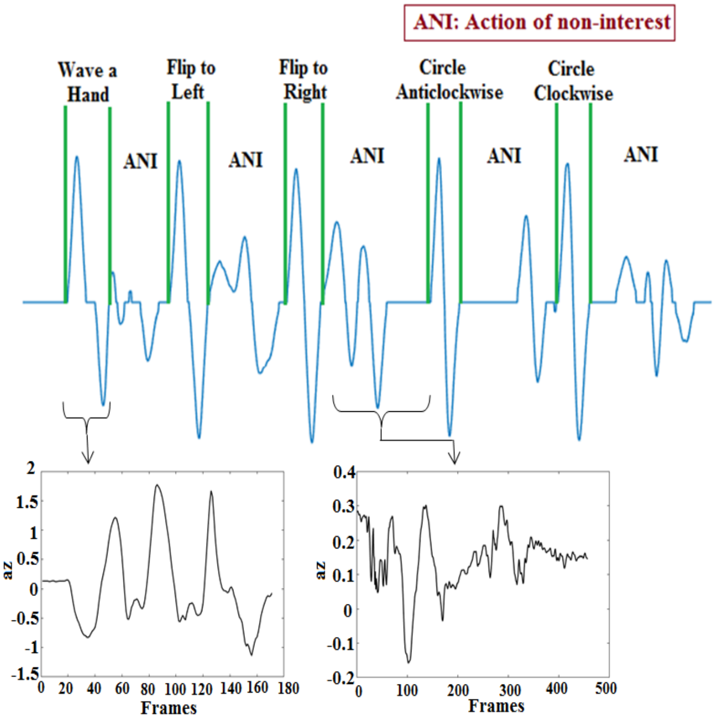


Figure 2.5. Potential difference (top curve) and acceleration along z-axis (bottom curves) in an activity sequence consisting of actions of interest and actions of non-interest

detection and recognition approach running in real-time can be viewed at www.utdallas.edu/~kehtar/ContinuousAction.avi.

2.6 CONCLUSION

In this chapter, a real-time action detection and recognition approach has been introduced which is capable of processing a continuous stream of depth images and inertial signals, a more challenging scenario than the conventional single action scenario normally reported in the literature. Continuous data implies actions of interest occur in a continuous manner while having actions of non-interest randomly located in-between them. The developed approach was applied to a continuous dataset for the smart TV application by simultaneously using a Kinect depth camera and a wearable inertial sensor. The results obtained show the effectiveness of the developed approach when activities are done continuously. In our future work, we plan to apply this approach to other applications or other sets of actions of interest.

2.7 REFERENCES

[1] S. Paul, S. Ali, and M. Shah, “A 3-dimensional sift descriptor and its application to action recognition,” *Proceedings of the 15th ACM International Conference on Multimedia*, pp. 357–360, September 2007.

[2] C. Chen, R. Jafari, and N. Kehtarnavaz, "Action recognition from depth sequences using depth motion maps-based local binary patterns," *Proceedings of the IEEE Winter Conference on Applications of Computer Vision*, pp 1092–1099, Waikoloa, HI, January 2015.

[3] J. Wang, Z. Liu, Y. Wu and J. Yuan, "Mining actionlet ensemble for action recognition with depth cameras," *Proceedings of Computer Vision and Pattern Recognition*, pp. 1290–1297, Providence, RI, June 2012.

[4] K. Altun, and B. Barshan, "Human activity recognition using inertial/magnetic sensor units," *Proceedings of International Workshop on Human Behavior Understanding*, Springer Berlin Heidelberg, pp. 38–51, August 2010.

[5] C. Chen, N. Kehtarnavaz, and R. Jafari, "A medication adherence monitoring system for pill bottles based on a wearable inertial sensor," *Proceedings of the 36th IEEE International Conference on Engineering in Medicine and Biology*, pp. 4983–4986, Chicago, IL, August 2014.

[6] M. Ermes, J. Parkka, J. Mantyjarvi, and I. Korhonen, "Detection of daily activities and sports with wearable sensors in controlled and uncontrolled conditions," *IEEE Transactions on Information Technology in Biomedicine*, vol. 12, no. 1, pp. 20–26, January 2008.

[7] E. Guenterberg, H. Ghasemzadeh, and R. Jafari, "Automatic segmentation and recognition in body sensor networks using a hidden Markov model," *ACM Transactions on Embedded Computing Systems*, vol. 11, no. S2, pp. 46:1–46:19, August 2012.

[8] C. Chen, R. Jafari, and N. Kehtarnavaz, "A survey of depth and inertial sensor fusion for human action recognition," *Multimedia Tools and Applications*, vol. 76, pp. 4405–4425, Feb 2017.

[9] C. Chen, R. Jafari, and N. Kehtarnavaz, “UTD-MHAD: A multimodal dataset for human action recognition utilizing a depth camera and a wearable inertial sensor,” *Proceedings of the IEEE International Conference on Image Processing*, pp. 168–172, Quebec City, Canada, September 2015.

[10] C. Chen, R. Jafari, and N. Kehtarnavaz, “A real-time human action recognition system using depth and inertial sensor fusion,” *IEEE Sensors Journal*, vol. 16, no. 3, pp. 773–781, February 2016.

[11] B. Delachaux, J. Rebetez, A. Perez-Uribe, and H. Mejia, “Indoor activity recognition by combining one-vs.-all neural network classifiers exploiting wearable and depth sensors,” *Proceedings of the 12th International Work-Conference on Artificial Neural Networks,* pp 216–223, June 2013.

[12] D. Huang, S. Yao, Y. Wang, and F. De La Torre, "Sequential max-margin event detectors," *Proceedings of European Conference on Computer Vision*, Springer International Publishing, pp. 410–424, September 2014.

[13] *Kinect for Windows*. Accessed: 2017. [Online]. Available: http://www.microsoft.com/en-us/kinectforwindows/

[14] Y. Yang, R. Jafari, S. Sastry, and R. Bajcsy, “Distributed recognition of human actions using wearable motion sensor networks,” *Journal of Ambient Intelligence and Smart Environments*, vol. 1, no. 2, pp. 103–115, January 2009.

[15] J. Shan, and S. Akella, "3D human action segmentation and recognition using pose kinetic energy," *Proceedings of the IEEE Workshop on Advanced Robotics and its Social Impacts*, pp. 69–75, September 2014.

[16] G. Zhu, L. Zhang, P. Shen, J. Song, L. Zhi and K. Yi, "Human action recognition using key poses and atomic motions," *Proceedings of the IEEE International Conference on Robotics and Biomimetics*, pp. 1209–1214, December 2015.

[17] G. Zhu, L. Zhang, P. Shen, and J. Song, "An Online Continuous Human Action Recognition Algorithm Based on the Kinect Sensor," *IEEE Sensors Journal*, vol. 16, no. 2, pp. 161, January 2016.

[18] C. Chen, R. Jafari, and N. Kehtarnavaz, "Improving human action recognition using fusion of depth camera and inertial sensors," *IEEE Transactions on Human-Machine Systems*, vol. 45, no. 1, pp. 51–61, February 2015.

[19] V. Bloom, D. Makris, and V. Argyriou, "G3d: A gaming action dataset and real time action recognition evaluation framework," *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*, pp. 7–12, June 2012.

[20] J. Davis, and M. Goadrich, "The relationship between Precision-Recall and ROC curves," *Proceedings of the 23rd International Conference on Machine Learning*, pp. 233–240, June 2006.

[21] C. Goutte, and E. Gaussier, "A probabilistic interpretation of precision, recall and F-score, with implication for evaluation," *Proceedings of European Conference on Information Retrieval*, Springer Berlin Heidelberg, pp. 345–359, March 2005.

CHAPTER 3

CONTINUOUS DETECTION AND RECOGNITION OF ACTIONS OF INTEREST AMONG ACTIONS OF NON-INTEREST USING A DEPTH CAMERA*

Authors – Neha Dawar, Nasser Kehtarnavaz

The Department of Electrical Engineering, EC33

The University of Texas at Dallas

800 West Campbell Road

Richardson, Texas 75080-3021

* ©(2017) IEEE. Reprinted, with permission, from (Neha Dawar, and Nasser Kehtarnavaz, “Continuous detection and recognition of actions of interest among actions of non-interest using a depth camera”, Proceedings of IEEE International Conference on Image Processing (ICIP), pp. 4227-4231, China, September 2017.)

ABSTRACT

This chapter presents a human action recognition approach using a depth camera for situations when actions of interest are performed in a continuous and random manner among actions of non-interest. The developed approach first performs detection of actions of interest by separating actions of interest from actions of non-interest in an on-the-fly manner and then classifies the detected actions of interest. Skeleton joint positions from depth images are used to achieve the detection of actions of interest. Recognition of detected actions of interest is then achieved by fusing the outcome of two classifiers, one classifier using skeleton joint positions and the other classifier using depth images. A continuous dataset consisting of actions of interest associated with the smart TV application is collected and made publicly available. The results obtained by applying the developed approach to this dataset indicate its effectiveness in detecting and recognizing actions of interest from continuous data streams.

3.1 INTRODUCTION

Human action recognition is an extensively researched topic in computer vision that has been utilized in many human-computer interaction applications. The literature includes a wide collection of papers involving the use of the Kinect depth camera for human action recognition, e.g., [1-5]. In these works, various features such as action graphs, random occupancy patterns (ROP), space-time occupancy patterns (STOP), depth motion maps (DMM), histogram of oriented gradients (HOG) have been extracted from depth images to achieve action recognition. 3D skeleton joint positions from depth images have also been used for action recognition, e.g., [6-7]. These joint positions are made available via the Microsoft Software Development Kit v2 [8]. Kinect v2 is capable of providing the 3D spatial locations of 25 skeleton body joints. In [9], both depth images and skeleton joint positions were used simultaneously for action recognition.

It is important to note that the considerable amount of research that has been conducted on human action or gesture recognition has focused primarily on recognizing actions that appear as single or isolated actions. It still remains a challenge to deal with continuous streams of activities composed of both actions of interest and actions of non-interest that appear in a random order. Continuous streams of activities constitute a more realistic scenario in many human-computer interaction applications such as smart TV and gaming.

A continuous action recognition approach using a depth camera was covered in [10]. However, the dataset examined only contained actions of interest. This chapter deals with a more challenging situation where both actions of interest and actions of non-interest occur continuously and in a random order. As a result, both the problems of action detection and action recognition are addressed at the same time to allow recognizing actions of interest among actions of non-interest in an on-the-fly manner. The developed approach first detects the presence of actions of interest from continuous data streams and then classifies them. The major contribution of this chapter is the development of a human action recognition approach, that is capable of dealing with recognizing actions of interest among actions of non-interest in continuous data streams by simultaneously using depth and skeleton information captured by a Kinect depth camera.

The rest of the chapter is organized as follows: The continuous dataset collected to analyze the developed approach is described in Section 3.2. Section 3.3 provides a detailed description of the approach. The experimental results are reported in Section 3.4, and the chapter is concluded in Section 3.5.

3.2 CONTINUOUS DATASET

This work involves the detection and recognition of actions of interest from a continuous data stream consisting of actions of interest and actions of non-interest that appear in a random order with respect to each other. Apart from the video datasets provided in [11-12] that are captured by video cameras, there is no publicly available dataset that provides continuous data streams from a depth camera. Hence, as part of this work, a dataset for the wrist actions involved in smart TV gestures was collected and is made publicly available. This dataset can be downloaded from this link http://www.utdallas.edu/~kehtar/UTD-CAD.htm.

The actions of interest for the smart TV application consist of ‘Waving a hand’, ‘Flip to Left’, ‘Flip to Right’, ‘Counterclockwise Rotation’, and ‘Clockwise Rotation’. For training, the subjects were asked to perform these actions of interest one action at a time. While for testing, the subjects were asked to perform these actions of interest continuously among various actions of non-interest in a random order. Subjects had the freedom to choose their own actions of non-interest. Example actions of non-interest included drinking water, eating snacks, stretching, walking around and reading a book.

Two scenarios were considered for data collection: subject-specific and subject-generic. For the subject-specific scenario, the training and testing were done by the same subject. For the training dataset, each of the actions of interest was done 15 times. For the testing dataset, the actions of interest were done randomly among actions of non-interest for 6 continuous data streams. For the subject-generic scenario, 5 different subjects were asked to repeat each of the actions of interest 5 times during the training. For the testing dataset, for each subject, a continuous data stream was collected which consisted of all the actions of interest randomly done among actions of non-interest.

3.3 DEVELOPED CONTINUOUS ACTION DETECTION AND RECOGNITION

The action detection and recognition approach developed in this chapter involves first segmenting and detecting actions of interest from a continuous data stream of skeleton joint positions and then classifying such detected actions of interest by utilizing both skeleton joint positions and depth images. The developed approach comprises three main steps: segmentation, detection and classification. The segmentation step involves identifying the presence of an action in a continuous data stream. These actions can be actions of interest or actions of non-interest. Next, in the detection step, a segmented action is labeled as an action of interest or an action of non-interest. This is achieved by using the method of support vector data description (SVDD) described in [13]. Then, in the classification step, the detected actions of interest are classified by using both the skeleton joint positions and depth images. A variable-length Maximum Entropy Markov Model (MEMM) classifier [14] is used for classification of skeleton information, while a Collaborative

Representation Classifier (CRC) [15] is used for classification of depth information. Basically, in this chapter, different existing techniques are integrated into a real-time approach to perform both detection and recognition of actions of interest among actions of non-interest performed in a continuous manner. A block diagram of the steps involved in the developed approach is shown in Figure 3.1. In what follows, more details of these steps are mentioned.

3.3.1 Segmentation Step

Segmentation of actions is achieved using skeleton joint positions via a technique similar to the one discussed in [14, 16]. The so called normalized relative orientations (NRO) of the joints are

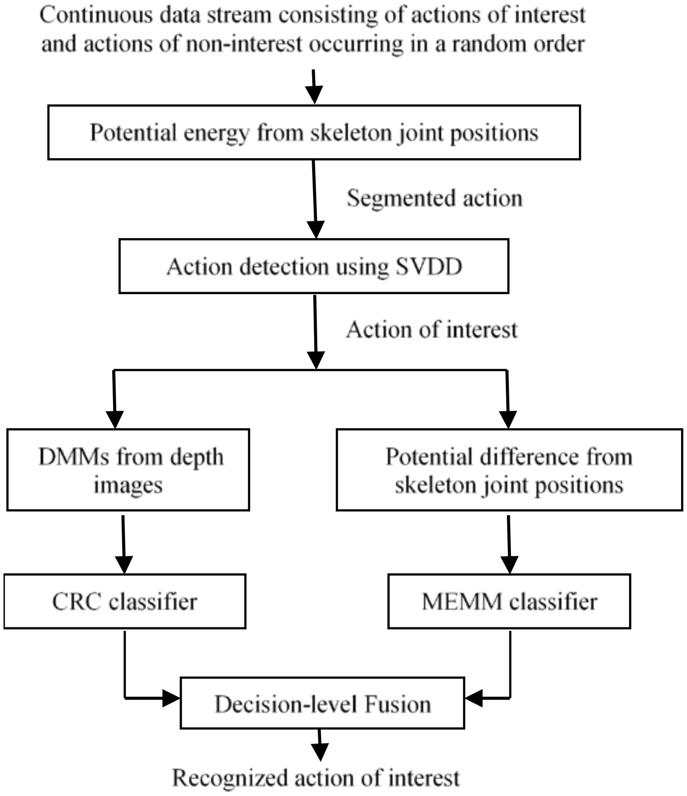

Continuous data stream consisting of actions of interest and actions of non-interest occurring in a random order
Potential energy from skeleton joint positions
Segmented action
Action detection using SVDD
Action of interest
DMMs from depth images
Potential difference from skeleton joint positions
CRC classifier
MEMM classifier
Decision-level Fusion
Recognized action of interest

Figure 3.1. Block diagram of the developed continuous action detection and recognition approach

extracted and used as features for segmentation. The NRO of a joint i is computed with respect to its rotating joint j as follows:

$$F_{NRO}^{i} = \frac{L_i - L_j}{\left\| L_i - L_j \right\|} \tag{3.1}$$

where L_i and L_j denote the respective 3D locations of joints i and j and $\|\cdot\|$ represents the Euclidean distance. Let $F_t = (F_{NRO}^{1}, F_{NRO}^{2}, F_{NRO}^{3}, \dots)_t$ be the vector of all the joints NROs at frame t. Based on a sequence of NRO feature vectors $(F_1, F_2, \dots, F_t, \dots)$, a potential energy function at the t^{th} frame is obtained as follows:

$$PE(t) = \|F_t - F_r\|^2 \tag{3.2}$$

where F_r denotes a reference NRO feature vector, which is considered here to be the first frame in the sequence. This potential energy function is compared to a user specified threshold. If the potential energy of the frames appears below this threshold, it is set to zero. This threshold is set experimentally. For example, for the dataset collected, a threshold in this range [1.05, 2.90] for the subject-specific scenario and a threshold in this range [3.70, 5.00] for the subject-generic scenario

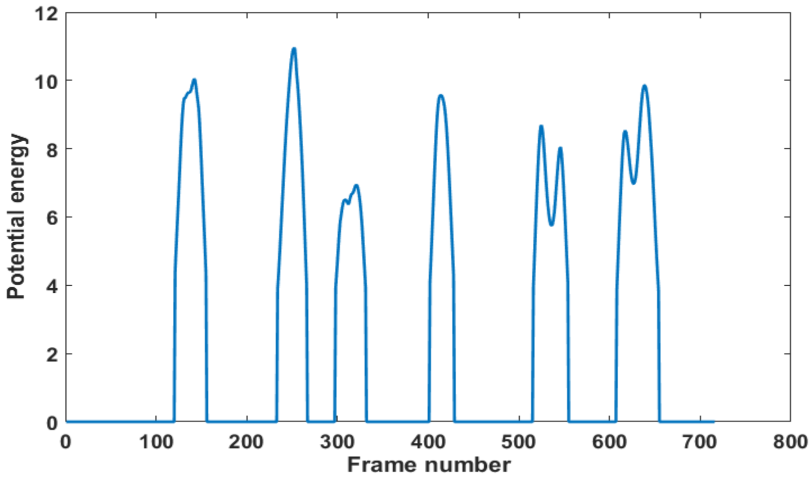


Figure 3.2. Potential energy of a continuous data stream

generated accurate detection. An example of the potential energy function for a continuous data stream is shown in Figure 3.2. As can be seen from this figure, consecutive frames with positive values are marked as a segmented action. Segmenting frames in this manner provides the start and end of an action, which are then used to detect whether that action is an action of interest or not.

3.3.2 Detection Step

Detection of actions of interest from a segmented action is done based on a one-class SVDD classifier. The basic concept behind SVDD is to find a spherical boundary enclosing all the data of interest. Consider a training dataset X consisting of N data samples x_m, $X = \{x_m, m = 1, \dots, N\}$. If R is the radius and a is the center of the smallest sphere encircling all the data samples, the following quantity is minimized in SVDD [13, 17]

$$H(R, a, \xi_m) = R^2 + \gamma \sum_{m=1}^{N} \xi_m \tag{3.3}$$

subject to the constraints

$$\|\phi(x_m) - a\|^2 \leq R^2 + \xi_m, \text{ for all } m \tag{3.4}$$

and

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

where ξ_m is a slack variable that penalizes outliers, γ is a parameter which controls a trade-off between volume and error and the notation ϕ indicates a nonlinear transformation to a higher dimensional kernel space. The interested reader is referred to [13] for more details on SVDD.

Once an action is considered from a continuous data stream, the potential energy of that action is divided into three equal portions and the average NROs from these three portions are used as X

in the above SVDD minimization. To examine a segmented action, the average NROs of its three equal portions are computed and mapped according to the nonlinear transformation. The distance of the feature vector from the center of the sphere is found. If this distance is less than the radius of the sphere, the corresponding action is considered to be an action of interest.

3.3.3 Classification Step

Next, the detected actions of interest are classified using a variable-length MEMM classifier for the skeleton information and a CRC classifier for the depth information. The operation of the MEMM classifier is similar to that of a Hidden Markov Model (HMM) classifier, but it is computationally more efficient than HMM.

For classification of skeleton data, a potential difference at frame t is computed from the potential energy function as follows:

$$PD(t) = PE(t) - PE(t-1) \tag{3.6}$$

If this potential difference is less than a value close to zero, segments of an action are labeled as pause segments, otherwise they are labeled as motion segments. For the dataset examined in this work, the value 0.04 allowed separating pause and motion segments. An example of the pause and motion segments for a segmented action ‘Flip to left’ is shown in Figure 3.3.

Based on pause and motion segments, a codebook is then setup, which is used to perform recognition. The details associated with the training of the MEMM classifier is provided in [16] with the difference that in this work, the clustering as part of the training is applied to motion segments, not pause segments. As a result, instead of considering motion segments between every pair of pause clusters, the mean of pause segments between every pair of motion clusters is

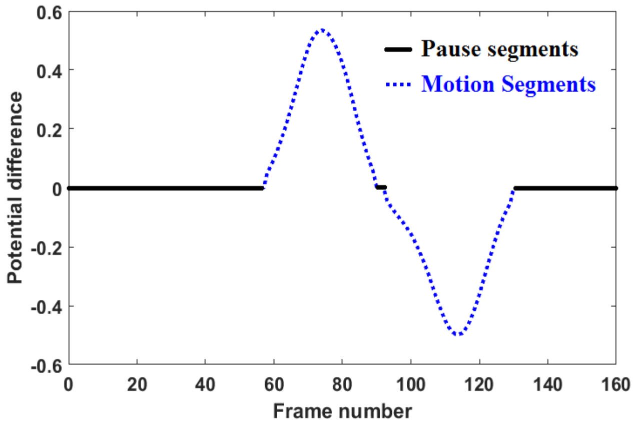


Figure 3.3. Pause and motion segments in the action ‘Flip to left’

considered. Recognition of a segmented action is then carried out based on likelihood probabilities as discussed in [16].

Classification of the depth data is done by first extracting depth motion maps (DMM) as described in [18]. DMMs are derived from 2D projection maps corresponding to the front, side and top views of 3D depth data. For a depth sequence of n frames, a DMM is obtained as follows:

$$DMM = \sum_{k=1}^{n-1} |map^{k+1} - map^{k}| \tag{3.7}$$

Similar to [19], a l_2-regularized CRC is utilized here to classify actions of interest based on DMMs. Finally, the decision-level fusion approach discussed in [20] is adopted using uniformly distributed classifier weights. The label of the segmented action is assigned to be the class with the largest probability.

3.4 RESULTS AND DISCUSSION

This section presents the results of the developed detection and recognition approach on the continuous dataset collected. Noting that there exists no approach in the literature that performs both detection and recognition of actions of interest when performed in a continuous and random manner among actions of non-interest, the evaluation of the approach developed in this chapter is carried out via commonly used recognition measures. The average duration of a continuous data stream in the examined dataset is about 40s with the actions of interest occupying 10s of this duration on average and the actions of non-interest occupying the remaining time of 30s.

The outcome of the segmentation and detection steps is reported in Table 3.1. For the subject-specific scenario, there were a total of 30 actions of interest in the 6 continuous data streams. In this scenario, all the 30 actions of interest were correctly detected, while there was only one action of non-interest that was wrongly detected as an action of interest. Similarly, all the 25 actions of interest in the subject-generic scenario were correctly detected, with only one action of non-interest wrongly detected as an action of interest.

For the overall detection and recognition results, since a sequence in a continuous data stream is unknown or cannot be matched to the ground truth, the evaluation of the overall approach was done using the widely used precision P, recall R and $F1$ score measures. These measures are defined as follows [21-22]:

$$P = {}^{\#TP}\big/_{(\#TP + \#FP)} \tag{3.8}$$

$$R = {}^{\#TP}\big/_{(\#TP + \#FN)} \tag{3.9}$$

$$F1 = {}^{2P \cdot R}\big/_{(P + R)} \tag{3.10}$$

Table 3.1. Outcome of the segmentation and detection steps

Scenario	Actions of interest correctly detected	Actions of non-interest detected as action of interest
Subject-specific	30/30	1
Subject-generic	25/25	1

where $\#TP$ denotes the number of true positives, $\#FP$ the number of false positives, and $\#FN$ the number of false negatives.

As mentioned in [23], whenever an action was found within a window of four frames, it was marked as a true positive, whereas the actions of non-interest wrongly detected as actions of interest or the actions of interest that were misclassified were marked as false positives. Actions of interest not detected or not correctly recognized were marked as false negatives.

The precision, recall and F1 score measures obtained for the subject-specific and subject-generic scenarios are reported in Tables 3.2 and 3.3, respectively. These tables also show the results of the situations when the classification was performed using the skeleton and depth information individually or separately. As can be seen from these tables, the values of the precision, recall and $F1$ score measures were increased when both the skeleton and depth information were used together due to correcting some of the misclassifications.

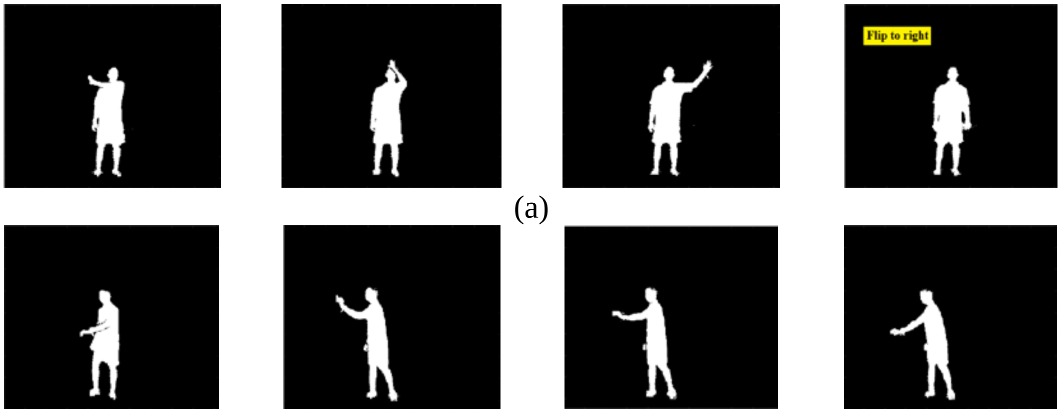
It is important to emphasize that detection was performed whenever an action was segmented in a continuous data stream and classification was performed whenever that action was labeled as an action of interest. An example of an action of interest and an action of non-interest from a test sequence is shown in Figure 3.4.

Table 3.2. Precision, recall and F1 score measures for subject-specific scenario

Modality used for classification	**Precision**	**Recall**	**F1 score**
Skeleton only	80.4%	83.1%	81.7%
Depth only	62.5%	64.5%	63.5%
Skeleton+Depth	85.6%	88.3%	86.9%

Table 3.3. Precision, recall and F1 score measures for subject-generic scenario

Modality used for classification	**Precision**	**Recall**	**F1 score**
Skeleton only	77.2%	80.2%	78.7%
Depth only	69.2%	72.0%	70.5%
Skeleton+Depth	86.8%	90.0%	88.3%



(a)

(b)

Figure 3.4. (a) Snapshots of depth images from an action of interest ‘Flip to Right’ (b) an action of non-interest ‘writing on a board’.

3.5 CONCLUSION

In this chapter, an action detection and recognition approach has been developed which is capable of dealing with continuous data streams captured by a depth camera. Such data streams for the smart TV application were collected which consisted of five actions of interest performed continuously and in a random order among various actions of non-interest. The results obtained indicate the effectiveness of the developed approach in separating actions of interest from actions of non-interest and classifying them in an on-the-fly manner. In our future work, it is planned to apply this approach to other applications involving different sets of actions of interest.

3.6 REFERENCES

[1] W. Li, Z. Zhang, and Z. Liu, "Action recognition based on a bag of 3d points," *Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition-Workshops*, pp. 9-14, June 2010.

[2] J. Wang, Z. Liu, J. Chorowski, Z. Chen, and Y. Wu, "Robust 3d action recognition with random occupancy patterns," *Proceedings of Computer Vision–ECCV 2012*, Springer Berlin Heidelberg, pp. 872-885, 2012.

[3] A. Vieira, E. Nascimento, G. Oliveira, Z. Liu, and M. Compos, "Stop: Space-time occupancy patterns for 3d action recognition from depth map sequences," *Iberoamerican Congress on Pattern Recognition*, Springer Berlin Heidelberg, pp. 252-259, September 2012.

[4] C. Chen, R. Jafari, and N. Kehtarnavaz, "Action recognition from depth sequences using depth motion maps-based local binary patterns," *Proceedings of the IEEE Winter Conference on Applications of Computer Vision*, Waikoloa, HI, pp. 1092-1099, January 2015.

[5] X. Yang, C. Zhang, and Y. Tian, "Recognizing actions using depth motion maps-based histograms of oriented gradients," *Proceedings of the 20th ACM International Conference on Multimedia*, pp. 1057-1060, October 2012.

[6] L. Xia, C. Chen, and J. Aggarwal, "View invariant human action recognition using histograms of 3d joints," *Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*, pp. 20-27, June 2012.

[7] X. Yang, and Y. Tian, "Eigenjoints-based action recognition using naive-bayes-nearest-neighbor," *Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*, pp. 14-19, June 2012.

[8] *Kinect for Windows*. Accessed: 2017. [Online]. Available: http://www.microsoft.com/en-us/kinectforwindows/

[9] J. Wang, Z. Liu, Y. Wu, and J. Yuan, "Mining actionlet ensemble for action recognition with depth cameras," *Proceedings of IEEE Computer Vision and Pattern Recognition (CVPR)*, pp. 1290-1297, June 2012.

[10] D. Huang, S. Yao, Y. Wang, and F. De La Torre, "Sequential max-margin event detectors," *Proceedings of European Conference on Computer Vision*, Springer International Publishing, pp. 410-424, September 2014.

[11] R. De Geest, E. Gavves, A. Ghodrati, Z. Li, C. Snoek, and T. Tuytelaars, “Online Action Detection”, *Proceedings of European Conference on Computer Vision*, Springer International Publishing, pp. 269-284, October 2016.

[12] M. Marszalek, I. Laptev, and C. Schmid, "Actions in context," *Proceedings of IEEE Computer Vision and Pattern Recognition*, pp. 2929-2936, June 2009.

[13] D. Tax, and R. Duin, "Support vector data description," *Machine Learning*, vol. 54, no. 1, pp. 45-66, January 2004.

[14] G. Zhu, L. Zhang, P. Shen, and J. Song, "An Online Continuous Human Action Recognition Algorithm Based on the Kinect Sensor," *IEEE Sensors Journal*, vol. 16, no. 2, pp. 161, January 2016.

[15] L. Zhang, M. Yang, and X. Feng, "Sparse representation or collaborative representation: Which helps face recognition?," *Proceedings of IEEE International Conference on Computer Vision*, pp. 471-478, November 2011.

[16] G. Zhu, L. Zhang, P. Shen, J. Song, L. Zhi and K. Yi, "Human action recognition using key poses and atomic motions," *Proceedings of IEEE International Conference on Robotics and Biomimetics*, pp. 1209-1214, December 2015.

[17] F. Saki, and N. Kehtarnavaz. "Online frame-based clustering with unknown number of clusters," *Pattern Recognition*, vol. 57, pp. 70-83, September 2016.

[18] C. Chen, K. Liu, and N. Kehtarnavaz, "Real-time human action recognition based on depth motion maps," *Journal of Real-Time Image Processing*, pp. 1-9, August 2013.

[19] C. Chen, R. Jafari, and N. Kehtarnavaz, “A real-time human action recognition system using depth and inertial sensor fusion,” *IEEE Sensors Journal*, vol. 16, no. 3, pp. 773-781, February 2016.

[20] C. Chen, R. Jafari, and N. Kehtarnavaz, "Fusion of depth, skeleton, and inertial data for human action recognition," *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 2712-2716, March 2016.

[21] J. Davis, and M. Goadrich, "The relationship between Precision-Recall and ROC curves," *Proceedings of the 23rd International Conference on Machine Learning*, pp. 233-240, June 2006.

[22] C. Goutte, and E. Gaussier, "A probabilistic interpretation of precision, recall and F-score, with implication for evaluation," *Proceedings of European Conference on Information Retrieval*, Springer Berlin Heidelberg, pp. 345-359, March 2005.

[23] V. Bloom, D. Makris, and V. Argyriou, "G3d: A gaming action dataset and real time action recognition evaluation framework," *Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*, pp. 7-12, June 2012.

CHAPTER 4

REAL-TIME CONTINUOUS DETECTION AND RECOGNITION OF SUBJECT-SPECIFIC SMART TV GESTURES VIA FUSION OF DEPTH AND INERTIAL SENSING[*]

Authors – Neha Dawar, Nasser Kehtarnavaz

The Department of Electrical Engineering, EC33

The University of Texas at Dallas

800 West Campbell Road

Richardson, Texas 75080-3021

[*] ©(2018) IEEE. Reprinted, with permission, from (Neha Dawar, and Nasser Kehtarnavaz, "Real-time continuous detection and recognition of subject-specific smart TV gestures via fusion of depth and inertial sensing", IEEE Access, vol. 6, pp. 7019-7028, January 2018.)

ABSTRACT

This chapter presents a real-time detection and recognition approach to identify actions of interest involved in the smart TV application from continuous action streams via simultaneous utilization of a depth camera and a wearable inertial sensor. Continuous action streams mean when actions of interest are performed continuously and randomly among arbitrary actions of non-interest. The developed approach consists of a detection part and a recognition part. In the detection part, two support vector data descriptor classifiers corresponding to the two sensing modalities are used to separate actions of interest from actions of non-interest in continuous action streams. The actions detected as actions of interest by both of the sensing modalities are then passed to the recognition part. In this part, actions of interest are classified by fusing the decisions from two collaborative representation classifiers, one classifier using skeleton joint positions and the other classifier using inertial signals. The developed approach is applied to the hand gestures in the smart TV application. The experimental results obtained indicate the effectiveness of the developed approach to detect and recognize smart TV gestures in continuous action streams.

4.1 INTRODUCTION

Human action recognition is finding its way into many consumer electronics products for gaming, video surveillance, content-based video retrieval, health monitoring, and assistive living. Information from various sensing modalities, such as RGB cameras, depth cameras, and wearable inertial sensors, have been used in the literature to recognize human actions. Many approaches utilizing information from image sequences captured by conventional RGB cameras have been proposed for action or gesture recognition, e.g., [1]–[3]. As noted in [4], the major limitations associated with using RGB cameras lies in the high computational demand when processing RGB images and in the challenges associated with obtaining 3D data. With the introduction of low-cost depth cameras, extensive research has been carried out using depth images and skeleton joint positions obtained from these cameras, e.g., [5]–[8]. Information from inertial signals acquired from wearable inertial sensors have also been widely used for action or gesture recognition, e.g., [9], [10].

It is important to note that in most existing works, action recognition is performed when actions of interest appear as segmented or isolated actions. That is to say, during the testing or operation phase, actions of interest are considered to be segmented from an activity sequence, or the start and end of the actions are considered known. A more realistic scenario constitutes performing action or gesture recognition from a continuous stream of activities where the actions or gestures of interest appear continuously and randomly among arbitrary actions of non-interest. It is a more challenging task to detect and recognize actions or gestures of interest from such continuous streams of activities. In [11], detection of actions from a continuous dataset was discussed, however, the dataset used only consisted of actions of interest with no randomly occurring actions of non-interest, and a person standing still was considered to be the only action of non-interest. In [12], [13], the detection and localization of actions from continuous image frames or video clips were carried out in an offline manner. Due to the high computational demand of processing RGB images, the real-time or on-the-fly operation of such approaches requires the use of dedicated image processing hardware. Here, no dedicated image processing hardware is utilized. Here, it is worth mentioning that recently there has been a considerable amount of increase in the use of convolutional neural networks (CNN) for action or gesture recognition, e.g., [14] –[16]. However, these approaches have not addressed the real-time aspect, which is the thrust of this work.

This chapter presents a real-time approach to detect and recognize actions or gestures of interest occurring in continuous action streams using both depth and inertial information. Continuous action streams constitute actions of interest occurring continuously and in a random order among arbitrary actions of non-interest. Skeleton joint positions from depth images obtained from a depth camera and inertial signals obtained from a wearable inertial sensor are used simultaneously to first separate actions of interest from actions of non-interest in continuous action streams and then to classify the detected actions of interest. Hence, the process of both detection and recognition are addressed at the same time in this approach in an online or real-time manner. To reduce the false detection of actions of non-interest, only the actions or gestures of interest that are detected by both skeleton joint positions and inertial signals are passed onto the recognition part. Recognition is performed by fusing the decision outcomes of two classifiers, one classifier using skeleton joint positions and the other using inertial signals. The simultaneous utilization of skeleton joint positions and inertial signals has allowed robust action or gesture recognition to be achieved in a computationally efficient or real-time manner.

In our previous work in [17], a detection and recognition approach was developed where the detection was done using only skeleton joint positions and inertial signals were used to verify the recognition outcome based on skeleton joint positions. In this chapter, detection and recognition are done in parallel for both skeleton joint positions and inertial signals. Furthermore, a different detection classifier is used here as compared to [17]. In our previous work in [18], only depth images were used to devise a detection and recognition approach.

The detection and recognition approach developed in this chapter is designed for subject-specific scenarios, meaning that the action recognition system is trained for a specific subject and then tested on the same subject while the subject carries out a continuous stream of activities. For commercial products, the subject-specific setting considered here constitutes a more realistic setting noting that there is a large intra-class variation of the same actions when they are performed by different subjects as part of arbitrary continuous streams of activities.

The depth camera used in this work is a Kinect v2, which captures depth images with a resolution of 512×424 pixels at a rate of approximately 30 frames per second. Its publicly available software package Kinect SDK [19] is capable of tracking the 3D spatial positions of 25 skeleton joints from the depth images generated by this camera.

The wearable inertial sensor that is used in this work is a small wireless body sensor discussed in [20]. The sensor generates 3-axis acceleration and 3-axis angular velocity at a sampling rate of 200Hz. These signals are wirelessly transmitted to a laptop via a Bluetooth link. The statistical features obtained from these signals are used for the detection of actions of interest from continuous action streams. It is worth pointing out that many other commercially available wearable inertial sensors can be used here in place of this inertial sensor to provide the same inertial signals.

The fusion of the outcomes of the two sensing modalities is conducted in both the detection and the recognition parts. In the detection part, the actions that are labeled as actions of interest by only one of the two sensing modalities are rejected. Only the actions of interest that are detected by both of the sensing modalities are labeled as actions of interest. A fusion in the recognition part is done by fusing the decisions made by two collaborative representation classifiers (CRC), one classifier operating on skeleton joint positions and the other classifier operating on inertial signals.

The major contributions of this work include: (i) real-time detection and recognition of actions or gestures of interest in realistic continuous action streams when actions or gestures of interest occur randomly and continuously among arbitrary actions of non-interest, and (ii) fusion of information from two differing sensing modalities of depth camera and inertial sensor to enable a more robust smart TV gesture detection and recognition compared to using a single sensing modality.

The remainder of the chapter is organized as follows. A description of the continuous dataset collected to analyze the developed approach is provided in Section 4.2. In Section 4.3, the details of the developed detection and recognition approach using the two differing sensing modalities are discussed. The experimental results and their discussions are then reported in Section 4.4. Finally, the chapter is concluded in Section 4.5.

4.2 CONTINUOUS DATASET

The aim of this work is to detect and recognize actions or gestures of interest associated with the smart TV application from continuous action streams using both skeleton joint positions and inertial signals. The datasets TVSeries [21] and Hollywood-2 [22] that appear in the literature provide continuous action streams from a video camera without any depth and inertial information. Hence, in order to study the performance of our approach, we have put together a dataset by simultaneously collecting continuous action streams from a depth camera and a wearable inertial sensor.

The wrist actions or gestures involved in the smart TV application were considered to put together this dataset. These wrist actions or gestures include: ‘Waving a Hand’, ‘Flip to Left’, ‘Flip to Right’, ‘Counterclockwise Rotation’ and ‘Clockwise Rotation’. Subjects were asked to wear the wearable inertial sensor on their right wrist while standing in front of the depth camera to perform these actions or gestures. The synchronization of the data from the depth camera and the inertial sensor was done based on the time stamp scheme described in [23]. Figure 4.1 illustrates the experimental setup for the collection of the continuous dataset.

The dataset was collected in a subject-specific manner, that is training and testing were carried out for the same subject. For training, a subject was asked to perform actions of interest one action at a time and the data were captured from the depth camera and the wearable inertial sensor

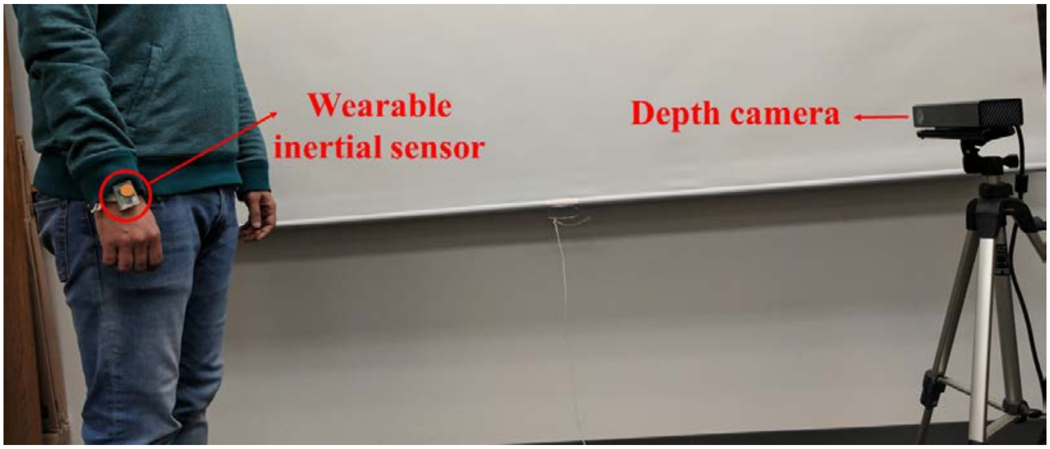
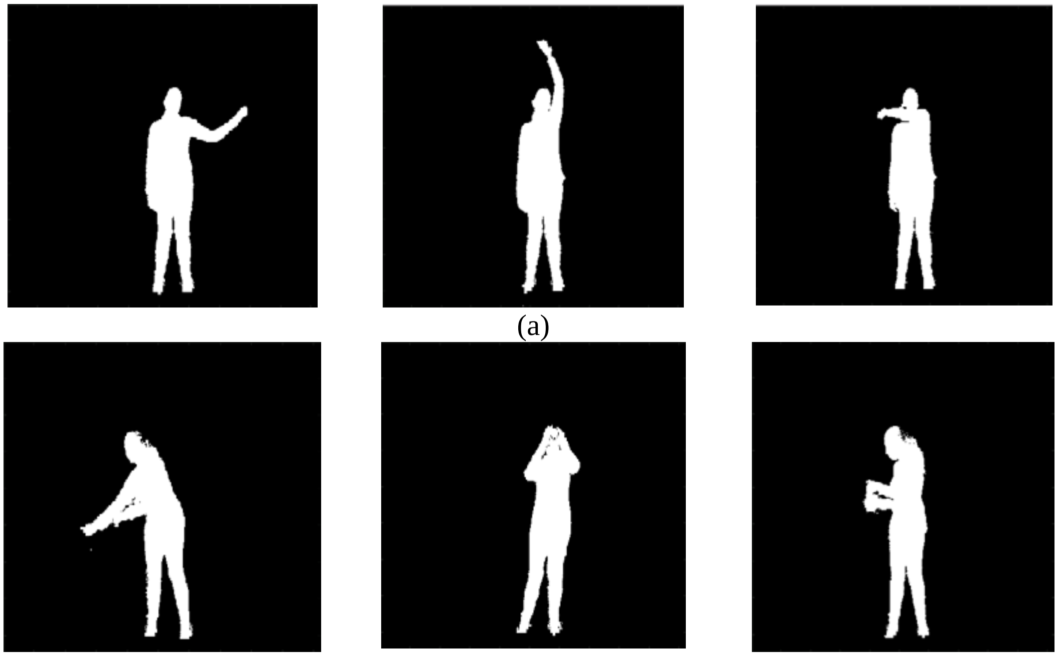


Figure 4.1. Experimental setup showing the depth camera and the wearable inertial sensor used

simultaneously. The training data involved 10 repetitions of each of the 5 actions of interest by a subject. For testing, the same subject was asked to perform the actions of interest among arbitrary actions of non-interest in a continuous and random order. A total of 5 such continuous action streams from a subject were considered to report the testing results. The data collection was repeated for 12 different subjects: 9 males and 3 females. No prior instructions were given to the subjects regarding the actions of non-interest and they had the complete freedom to choose their own actions of non-interest while staying in the field of view of the depth camera. Some actions of non-interest performed by the subjects included drinking water, reading a book, pointing at some object, writing on the board, etc. A typical duration of the collected continuous action streams is about 92s with the actions of interest occupying a total duration of about 16s out of 92s. Sample depth image frames with the background subtracted for an action or gesture of interest and some



(a)

(b)

Figure 4.2. Example background subtracted depth images from (a) different frames of an action of interest ‘Counterclockwise Circle’ and (b) actions of non-interest, from left to right: ‘Lifting a water bottle, drinking water and putting it back’

actions of non-interest are shown in Figure 4.2. This dataset is made available for public use and can be downloaded from this link: http://www.utdallas.edu/~kehtar/UTD-CAD-Specific.htm.

4.3 DEVELOPED DETECTION AND RECOGNITION APPROACH

The developed detection and recognition approach is based on the fusion of information from skeleton joint positions and inertial signals. The process begins by performing segmentation and detection based on skeleton joint positions and inertial signals separately. Only the actions of interest detected by both of the sensing modalities are considered for recognition. Action recognition is achieved by fusing the decisions of two CRC classifiers, one classifier using skeleton joint positions and the other using inertial signals. Hence, the processing pipeline can be considered to comprise the following main modules or components: segmentation and detection, and classification

Segmentation of actions using skeleton joint positions is carried out by using the so-called potential difference of the skeleton feature vectors [24]. The technique of support vector data description (SVDD) [25] is used to label the segmented actions as either actions of interest or actions of non-interest. In parallel or simultaneously, segmentation based on the inertial signals is carried out by using the acceleration difference signal. Detection of actions of interest from the segmented actions is carried out using SVDD.

The classification or recognition of detected actions of interest is performed using two l_2-regularized CRC classifiers [26] separately for skeleton joint positions and inertial signals. A decision fusion is then applied to the outcome of the two classifiers. A detected test action is assigned to the class that best approximates it. The entire processing pipeline is designed to be computationally efficient allowing it to run in real-time on a laptop platform with no dedicated

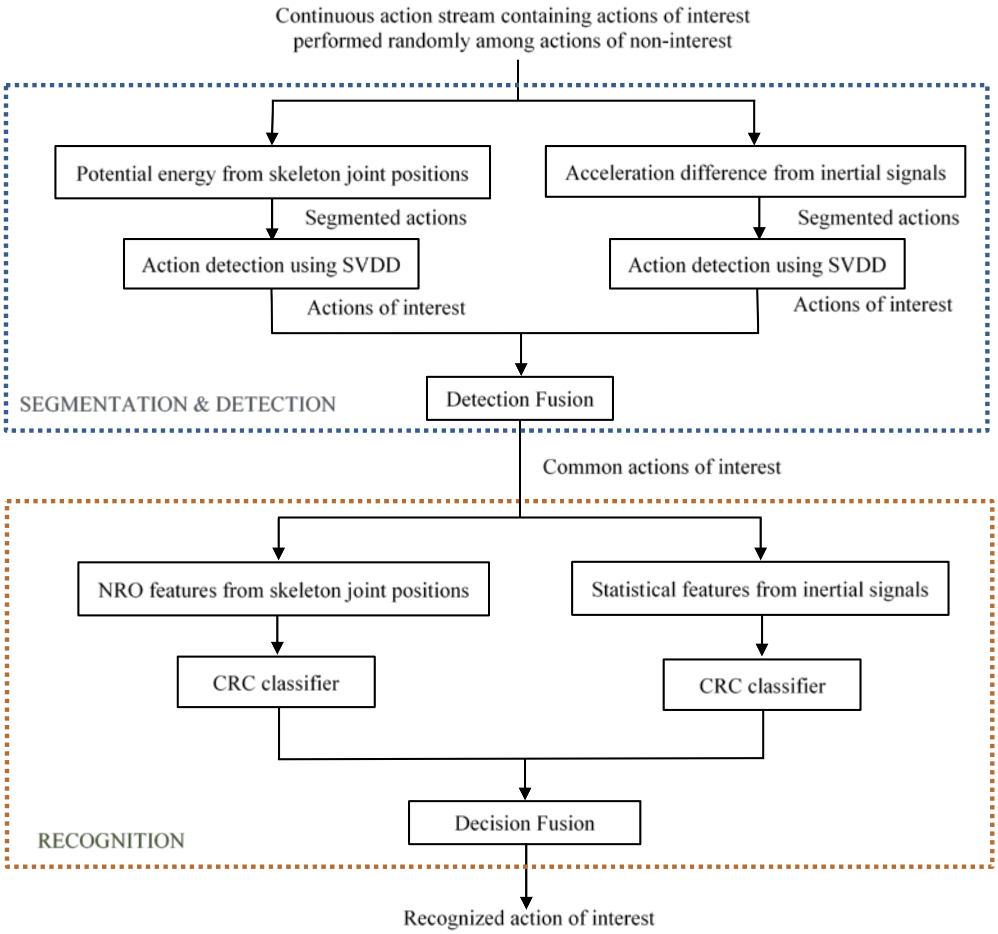

Continuous action stream containing actions of interest performed randomly among actions of non-interest
Potential energy from skeleton joint positions
Acceleration difference from inertial signals
Segmented actions
Segmented actions
Action detection using SVDD
Action detection using SVDD
Actions of interest
Actions of interest
Detection Fusion
SEGMENTATION & DETECTION
Common actions of interest
NRO features from skeleton joint positions
Statistical features from inertial signals
CRC classifier
CRC classifier
Decision Fusion
RECOGNITION
Recognized action of interest

Figure 4.3. Block diagram of the developed continuous action detection and recognition approach

image processing hardware. Figure 4.3 displays a block diagram of the developed detection and recognition approach. These blocks are discussed in more details in the subsections that follow.

4.3.1 Segmentation using Skeleton Joint Positions

To obtain features or a feature vector from skeleton joint positions, as proposed in [27], the Normalized Relative Orientation (NRO) of each joint, that is the relative position of each joint with respect to the joint about which it rotates, is computed. This makes the features obtained from the skeleton joints invariant to the camera position and the height of a subject. If L_i and L_j represent the respective 3D locations of a joint i and a joint j, the NRO of the joint i relative to its rotating joint j is computed as follows [24]:

$$F_{NRO}^{i} = \frac{L_i - L_j}{\|L_i - L_j\|} \tag{4.1}$$

where $\|\cdot\|$ denotes the Euclidean distance. For example, the NRO of the ankle joint is obtained with respect to the knee joint, the NRO of the knee joint is obtained with respect to the hip joint, and so on.

Let $(F_1, F_2, \dots, F_t, \dots, F_n)$ represent a sequence of NRO features, where $F_t = (F_{NRO}^{1}, F_{NRO}^{2}, F_{NRO}^{3}, \dots)_t$ indicates the NRO features of the joints at frame t. Based on a reference NRO F_r, a potential energy function at frame t is computed as follows:

$$PE(t) = \|F_t - F_r\|^2 \tag{4.2}$$

Here, the first frame in the sequence is used as the reference NRO.

The potential energy function is then thresholded. That is, if the potential energy function at a particular frame appears below a specified threshold value (discussed later in the parameter setting part of the results section), it is set to zero and the condition is considered to be a pause segment. Consecutive frames with zero value of potential energy are marked as pause segments while consecutive frames with positive values of potential energy are marked as action segments. An action segment appears in between pause segments. Whenever an action is identified, detection based on SVDD is activated to see whether that action is an action of interest or an action of non-interest. An example of a potential energy function exhibiting segmented actions in a continuous action stream is shown in Figure 4.4.

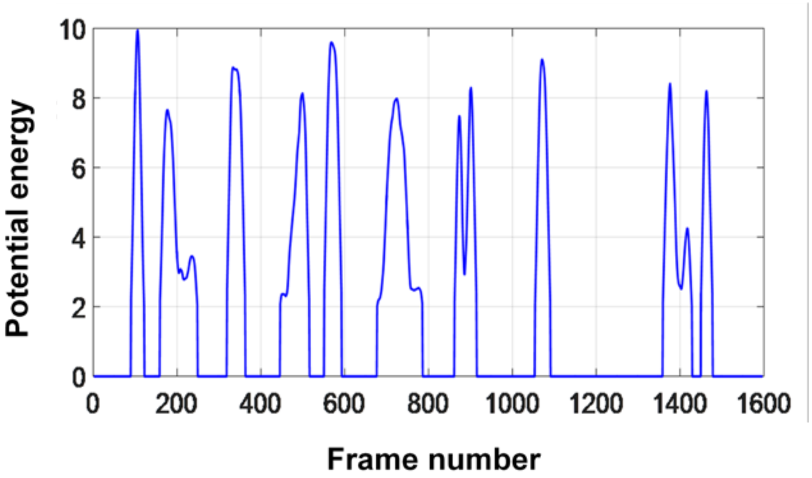


Figure 4.4. An example potential energy function of a continuous action stream

4.3.2 Segmentation using Inertial Signals

To obtain segmented actions from inertial signals, the acceleration signal is utilized. Acceleration at frame t is computed as follows:

$$a(t) = \sqrt{a_x(t)^2 + a_y(t)^2 + a_z(t)^2} \tag{4.3}$$

where $a_x(t), a_y(t), a_z(t)$ denote the 3D accelerations at frame t. Based on a reference acceleration $a(r)$, an acceleration difference function is obtained as follows:

$$AD(t) = a(t) - a(r) \tag{4.4}$$

Here, the acceleration at the first frame of a sequence is used as the reference. The acceleration difference function is then thresholded. Signal frames below a specified threshold value (discussed later in the parameter setting part of the results section) are identified as pause segments. Consecutive frames with non-zero acceleration difference make up an action segment. An example of an acceleration difference function for a continuous action stream is shown in Figure 4.5. Note that angular velocity difference, instead of acceleration difference, can also be used for the same purpose.

4.3.3 SVDD based Detection

Segmenting actions from a continuous action stream is followed by labeling them as actions of interest or actions of non-interest. This detection step is done based on a one-class SVDD classifier. SVDD allows finding a spherical boundary that encloses all the data of interest. Training an SVDD classifier involves finding the center and the radius of the spherical boundary. During testing, if a sample falls inside the boundary, it is regarded as data of interest.

Based on a training set X comprising N samples, $X = \{x_k | x_k \in \mathbb{R}^D, k = 1, \dots, N\}$, as discussed in [25], [28], the radius R and the center b of the spherical boundary enclosing all the samples are obtained by solving the following minimization problem:

$$\min_{R,b,\xi} \left(R^2 + \gamma \sum_{k=1}^{N} \xi_k \right) \tag{4.5}$$

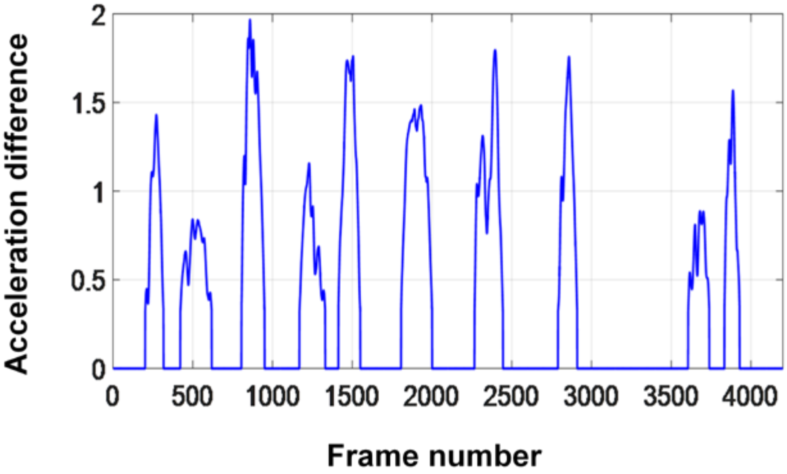


Figure 4.5. An example acceleration difference function of a continuous action stream

subject to the constraints

$$\|\phi(x_k) - b\|^2 \leq R^2 + \xi_k, \text{ for all } k \tag{4.6}$$

and

$$\xi_k \geq 0, \text{ for all } k \tag{4.7}$$

where ξ_k is a variable penalizing outliers, γ is a parameter controlling a trade-off between volume and misclassification error, and ϕ represents a nonlinear transformation to a higher dimensional kernel space. In [29], the solution of this minimization problem is provided using the Lagrange multiplier method.

For testing an unknown sample $y \in \mathbb{R}^D$, the distance of this sample from the center of the spherical boundary is computed. If the sample lies within the boundary of the sphere, it is accepted as data of interest, otherwise it is rejected. For action detection when using skeleton joint positions, the potential energy function is divided into three equal portions. The average NROs from these three equal portions are then used as X in the above SVDD minimization problem. For the inertial signals, the statistical features of *mean*, *variance*, *standard deviation* and *root mean square* for the acceleration and angular velocity signals along all the three directions, similar to the features used in [30], are used as X in the above SVDD minimization problem. Only those actions that are detected as actions of interest by both skeleton joint positions and inertial signals are considered further in the recognition part. This improves the detection accuracy as actions of non-interest wrongly detected as actions of interest by one of the two sensing modalities get ruled out in this manner.

4.3.4 Recognition via CRC

Recognition based on skeleton joint positions and inertial signals is done by using two CRC classifiers. A CRC classifier is a computationally efficient classifier that has proven effective in image processing applications [31]. To keep the computational complexity low, as discussed in [26], l_2-regularization is considered instead of the conventional l_1-regularization.

Let $Z = [z_1, z_2, \dots, z_N] \in \mathbb{R}^{d\times N}$ denote N training samples with $z_m \in \mathbb{R}^d$. A CRC classifier represents a test sample $w \in \mathbb{R}^d$ as a linear combination of all the training samples, that is

$$w = Z\alpha \tag{4.8}$$

where $\alpha \in \mathbb{R}^N$ represents a coefficient vector corresponding to the training samples. The test sample is classified by solving the following l_2-regularized minimization problem as discussed in [26], [32]:

$$\hat{\alpha} = \arg\min_{\alpha} \|w - Z\alpha\|_2^2 + \lambda\|\alpha\|_2^2 \tag{4.9}$$

where λ denotes a regularization parameter. The closed form solution of (9) is given by [32]

$$\hat{\alpha} = (Z^T Z + \lambda I)^{-1} Z^T w \tag{4.10}$$

where I denotes the identity matrix. If Z_c is the set of all the training samples belonging to a class $c \in [1, \dots, C]$ and $\hat{\alpha}_c$ is the corresponding coefficient vector, a reconstruction error $e_c(w)$ can be computed for each class as follows:

$$e_c(w) = \|w - Z_c\hat{\alpha}_c\|_2 \tag{4.11}$$

The test sample w is then assigned to the class having the least reconstruction error.

For recognition based on skeleton joint positions, the potential energy function is divided into N_S equal size temporal portions. The average NROs from these portions are used as Z in the CRC classification.

For recognition based on inertial signals, the acceleration and angular velocity signals are divided into N_I equal size temporal portions. The statistical features of *mean*, *variance*, *standard deviation*, and *root mean square* for the acceleration and angular velocity signals along all the three directions per temporal portion are used as Z in the CRC classification.

4.3.5 Decision Fusion

To take into consideration the decision made by each CRC classifier, a decision fusion is carried out. Due to its high computational efficiency, the logarithm opinion pool (LOGP) technique is used here similar to the fusion performed in [31], [33]. Each CRC classifier generates an error vector indicating the reconstruction error for each class. Let e^S and e^I represent the error vectors generated by the CRC classifiers based on the skeleton joint positions and inertial signals, respectively. Let $\beta \in [1, \ldots, C]$ represent the class label. As described in [32], the individual posterior probabilities $p_S(\beta|w)$ and $p_I(\beta|w)$ of the two classifiers for a test sample w can be obtained based on a Gaussian mass function as follows:

$$p_S(\beta|w) = \exp(-e^S), \quad p_I(\beta|w) = \exp(-e^I) \tag{4.12}$$

These posterior probabilities with the errors normalized to one are used in the LOGP technique to set up this probability [30]

$$P(\beta|w) = p_S(\beta|w)^{\alpha_S} \cdot p_I(\beta|w)^{\alpha_I} \tag{4.13}$$

where α_S and α_I denote weights associated with the two classifiers. Here, these weights are considered to be the same, that is $\alpha_S = \alpha_I = \frac{1}{2}$. The class label assigned to a test sample is considered to be the one which has the maximum probability $P(\beta|w)$, that is

$$label(w) = \arg \max_{\beta \in [1,\ldots,C]} P(\beta|w) \tag{4.14}$$

Note that the probabilities obtained in (12) and (13) are normalized by their sum over all the classes before they are used.

4.4 EXPERIMENTAL RESULTS AND DISCUSSION

This section presents the experimental results of the developed detection and classification modules on the continuous smart TV gesture dataset described earlier when using the two sensing modalities of depth camera and inertial sensor simultaneously. Note that fusion of these two differing sensing modalities is done both in the detection and the recognition or classification part. The next section provides procedures or guidelines for selecting the parameters used in the developed approach.

4.4.1 Parameter Setting

As mentioned earlier, during segmentation, the potential energy and acceleration difference functions are thresholded. Proper values of these thresholds can be determined by collecting a validation continuous action stream for each subject. The maximum potential energy value and the maximum acceleration difference value of the pause segments from the validation stream can be used to serve as the threshold to separate pause and action segments. In our experimentations, the threshold obtained in the above manner for the potential energy were found to be in the range [0.25, 3] and those for the acceleration difference were found to be in the range [0.15, 0.5].

As noted earlier, SVDD is used to define a spherical boundary enclosing the actions of interest performed by a specific subject. The samples that fall within this boundary are labeled as actions of interest, while the ones falling outside this boundary are rejected or considered to be actions of non-interest. To determine the center and the radius of the spherical boundary, a transformation of the sample points to a higher dimensional space is performed using a Gaussian kernel. The standard deviation parameter σ of the Gaussian kernel determines the smoothness of the boundary obtained by SVDD [28]. The value of σ is chosen per specific subject such that all the actions of interest in a continuous validation stream are detected and all the actions of non-interest are rejected. For the skeleton data, $\sigma = 3$ for subject 9 and $\sigma = 5$ for all the other subjects were found to separate actions of interest and actions of non-interest. For the inertial data, $\sigma = 1e6$ for subjects 1, 4 and 5, $\sigma = 1e5$ for subjects 2, 6, 7, 10 and 11 and $\sigma = 1e4$ for subjects 3, 8, 9 and 12 were found to separate actions of interest and actions of non-interest.

The values of N_S and N_I in the recognition part were determined as follows. For each subject, a sweep of values was examined for both the potential energy function and acceleration difference function. Appropriate values were obtained via a five-fold cross-validation over the training set. For each value of N_S and N_I, the training data for each subject was randomly divided into training and validation sets in a 4:1 ratio, and the recognition rate was obtained. This was repeated 100 times for each value of N_S and N_I. Then, the N_S and N_I values providing the highest recognition rate per subject were chosen. The average of the recognition rates over 100 trials across different values of N_S and N_I for one of the subjects is shown in Figure 4.6.

Another parameter is the regularization parameter λ of the CRC classifiers. The value of λ was chosen based on a five-fold cross validation over the training set per specific subject. Different values of λ were used in the CRC classifiers and the one generating the highest recognition rate was chosen. Figure 4.7 shows the recognition rates obtained for different values of λ for one of the subjects. The selected values of N_S, N_I and λ for the skeleton (λ_s) and inertial (λ_I) modalities per subject are indicated in Table 4.1.

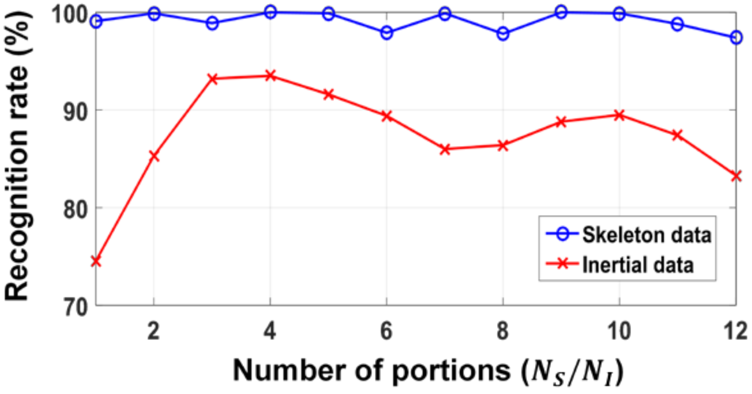


Figure 4.6. Average recognition rates of cross validation over different values of N_S and N_I for subject 3

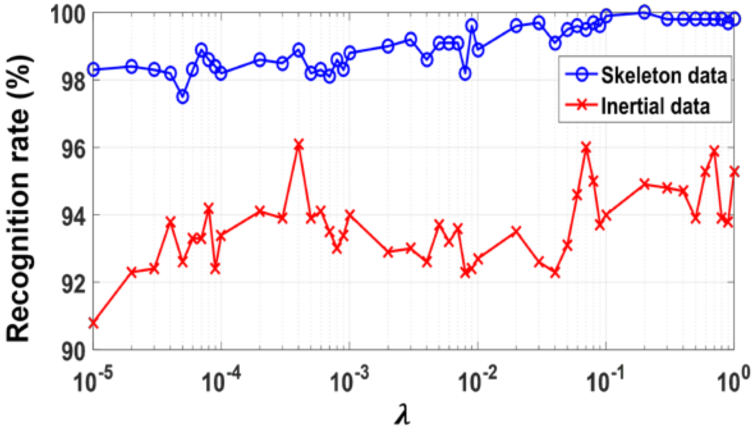


Figure 4.7. Average recognition rates of cross validation over different values of λ for subject 4

Table 4.1. Values of N_S, N_I, λ_S and λ_I chosen based on cross-validation per subject

Subject	$\boldsymbol{N_S}$	$\boldsymbol{N_I}$	$\boldsymbol{\lambda_S}$	$\boldsymbol{\lambda_I}$
1	1	2	4e-2	7e-5
2	6	2	3e-2	2e-5
3	4	4	2e-1	2e-5
4	6	3	2e-1	4e-4
5	4	2	2e-1	1e-5
6	6	6	9e-2	3e-5
7	7	3	1e-1	3e-5
8	7	2	1e-2	3e-5
9	7	3	1e-1	6e-5
10	4	3	1e1	9e-5
11	6	4	3e-5	1e-5
12	4	3	1e-1	5e-5

4.4.2 Detection Outcome

As described earlier, detection of actions of interest from a continuous action stream is carried out based on two SVDD classifiers, one operating on skeleton joint positions and the other on inertial signals. The actions detected as actions of interest by both the SVDD classifiers are considered to be the fusion outcome of the detection part. Table 4.2 reports the number of correctly detected actions of interest, referred to as true detections and the number of falsely detected actions of non-interest, referred to as false detections, for each subject by the SVDD classifiers corresponding to the skeleton joint positions and the inertial signals individually. Note that per subject there were a total of 25 actions of interest in the 5 continuous action streams. The fusion outcomes of the detection part are also reported in this table, which incorporates rejecting those actions that are detected by only one of the two classifiers. As can be observed from this table, the fusion in the detection part led to rejecting all the wrongly detected actions of non-interest. In other words, there

Table 4.2. Detection outcome based on skeleton joint positions only, inertial signals only and fusion of the two

	Skeleton Joint Positions Only		**Inertial Signals Only**		**Fusion of Skeleton Joint Positions and Inertial Signals**	
Subject	**TD**	**FD**	**TD**	**FD**	**TD**	**FD**
1	25	0	25	4	25	0
2	25	2	25	2	25	0
3	25	2	25	3	25	0
4	25	0	25	1	25	0
5	25	3	25	1	25	0
6	25	1	25	3	25	0
7	25	0	25	0	25	0
8	24	0	25	6	24	0
9	25	0	25	2	25	0
10	25	0	23	18	23	0
11	25	0	25	2	25	0
12	25	0	25	14	25	0

TD: True Detections, FD: False Detections

was no action of non-interest that was labeled as an action of interest when using both of the sensing modalities.

4.4.3 Continuous Action Recognition Outcome

In the recognition part, classification is performed on those actions of interest that are detected by both of the SVDD classifiers. The ground truth actions were identified by visual segmentation from the continuous action streams. Actions of interest correctly classified were marked as true positives. Actions of non-interest wrongly detected as actions of interest as well as misclassified actions of interest were marked as false positives. Actions of interest that were not detected and the ones that were not correctly classified were marked as false negatives. The overall performance of the approach is evaluated here based on the widely used measures of precision, recall, and $F1$ score [34]–[36]. Precision measures the fraction of true positives over all detections. Let $\#TP$

represent the number of true positives, $\#FP$ the number of false positives and $\#FN$ the number of false negatives, then precision P is defined as

$$P = \frac{\#TP}{\#TP + \#FP} \tag{4.15}$$

The recall measure R reflects the fraction of true positives detected among actions of interest and is defined as

$$R = \frac{\#TP}{\#TP + \#FN} \tag{4.16}$$

Considering the 25 actions of interest in the continuous action streams, the term $\#TP + \#FN$ for each subject was 25. The measure $F1$ score is the harmonic average of precision and recall, that is

$$F1 = 2\frac{P \cdot R}{(P + R)} \tag{4.17}$$

Per subject, the precision, recall and $F1$ scores were obtained using the skeleton joint positions and the inertial signals separately or individually, and by using the fusion of the two sensing modalities. These measures are reported in Table 4.3 for the subjects examined. A noticeable improvement can be seen in this table due to the fusion of the two sensing modalities compared to the situations when using a single or an individual sensing modality. Table 4.4 reports the overall precision, recall and $F1$ scores computed over all the subjects. As evident from this table, an overall improvement of more than 10% was gained in the $F1$ score when using the fusion of the two sensing modalities.

The recognition confusion matrices for the classification based on only the skeleton joint positions, only the inertial signals, and the fusion of the two are shown in Tables 4.5, 4.6 and 4.7, respectively. In these tables, WH denotes the gesture ‘Waving a Hand’, FL denotes the gesture

‘Flip to Left’, FR denotes the gesture ‘Flip to Right’, CCR denotes the gesture ‘Counterclockwise Rotation’ and CR denotes the gesture ‘Clockwise Rotation’. Note that the reported confusion

Table 4.3. Precision, recall, and F1 scores per subject

Subject	**Modality Used**	**Precision**	**Recall**	**F1 score**
1	Skeleton only	84.0%	84.0%	84.0%
	Inertial only	79.3%	92.0%	85.1%
	Skeleton & Inertial	96.0%	96.0%	96.0%
2	Skeleton only	55.5%	60.0%	57.6%
	Inertial only	92.5%	100%	96.1%
	Skeleton & Inertial	100%	100%	100%
3	Skeleton only	74.0%	80.0%	76.9%
	Inertial only	78.5%	88.0%	83.0%
	Skeleton & Inertial	88.0%	88.0%	88.0%
4	Skeleton only	92.0%	92.0%	92.0%
	Inertial only	69.2%	72.0%	70.5%
	Skeleton & Inertial	96.0%	96.0%	96.0%
5	Skeleton only	82.1%	92.0%	86.7%
	Inertial only	92.3%	96.0%	94.1%
	Skeleton & Inertial	100%	100%	100%
6	Skeleton only	80.7%	84.0%	82.3%
	Inertial only	78.5%	88.0%	83.1%
	Skeleton & Inertial	92.0%	92.0%	92.0%
7	Skeleton only	80.0%	80.0%	80.0%
	Inertial only	92.0%	92.0%	92.0%
	Skeleton & Inertial	100%	100%	100%
8	Skeleton only	83.3%	80.0%	81.6%
	Inertial only	74.2%	92.0%	82.1%
	Skeleton & Inertial	100%	96.0%	97.9%
9	Skeleton only	68.0%	68.0%	68.0%
	Inertial only	88.8%	96.0%	92.3%
	Skeleton & Inertial	96.0%	96.0%	96.0%
10	Skeleton only	72.0%	72.0%	72.0%
	Inertial only	45.2%	76.0%	56.7%
	Skeleton & Inertial	100%	92.0%	95.8%
11	Skeleton only	76.0%	76.0%	76.0%
	Inertial only	88.8%	96.0%	92.3%
	Skeleton & Inertial	96.0%	96.0%	96.0%
12	Skeleton only	76.0%	76.0%	76.0%
	Inertial only	61.5%	96.0%	75.0%
	Skeleton & Inertial	96.0%	96.0%	96.0%

Table 4.4. Overall precision, recall, and F1 scores across all the subjects

Modality Used	Precision	Recall	F1 score
Skeleton only	76.9%	78.7%	77.8%
Inertial only	76.3%	90.3%	82.8%
Skeleton & Inertial	96.6%	95.7%	96.2%

Table 4.5. Confusion matrix for classification using skeleton joint positions only (in %)

Action of interest	WH	FL	FR	CCR	CR
WH	74.1	6.9	17.3	1.7	-
FL	-	86.4	8.5	5.1	-
FR	5.0	1.7	91.6	1.7	-
CCR	3.3	8.3	15.0	66.7	6.7
CR	6.7	-	11.6	6.7	75.0

Table 4.6. Confusion matrix for classification using inertial signals only (in %)

Action of interest	WH	FL	FR	CCR	CR
WH	98.3	-	-	1.7	-
FL	1.7	91.5	3.4	-	3.4
FR	6.7	3.3	85.0	1.7	3.3
CCR	3.3	-	1.7	95.0	-
CR	8.4	3.3	-	3.3	85.0

Table 4.7. Confusion matrix for classification using fusion of skeleton joint positions and inertial signals (in %)

Action of interest	WH	FL	FR	CCR	CR
WH	98.3	1.7	-	-	-
FL	-	100	-	-	-
FR	1.7	-	95.0	3.3	-
CCR	1.7	-	1.7	93.3	3.3
CR	1.7	-	1.7	-	96.6

matrices correspond to the recognition after rejecting the actions that were not detected by both the sensing modalities, while the reported precision, recall and $F1$ scores correspond to the performance of the entire approach consisting of both the detection and recognition parts.

Noting that existing datasets that provide simultaneous data from depth and inertial sensors contain actions that are already segmented, it was not possible to apply the developed detection and recognition approach to them. In order to provide a comparison, the recognition part of the developed approach was applied to the UTD-MHAD dataset [23], which is a multimodal dataset consisting of 27 actions performed by 8 subjects. The data from odd subjects were used for training, and the data from even subjects were used for testing. Table 4.8 shows the comparison of the accuracies with two other computationally efficient approaches. As seen from this table, a higher accuracy is achieved when using the developed approach.

4.4.4 Processing Time

It is important to note that the developed action recognition system detects and recognizes actions of interest in an on-the-fly or real-time manner. The detection module is executed whenever the segmentation of an action gets done across a continuous action stream. The recognition module is executed whenever an action is detected as an action of interest. All these operations are carried out online as data samples are received in real-time from the depth camera and the inertial sensor. The implementation coding was done in MATLAB. The average processing time of the major components in these modules are listed in Table 4.9. These processing times are for a laptop equipped with 2.6GHz processor with 16GB RAM without using any dedicated image processing hardware. As noted in the table, the total processing time of the major components is only about 6.7ms, which meets the real-time frame processing rate of the depth camera at 30ms per frame. A video clip of the approach running in real-time can be viewed at this link: http://www.utdallas.edu/~kehtar/RTContinuousAction.avi.

Table 4.8. Recognition accuracy on UTD-MHAD dataset

Method	Accuracy (%)
ELC-KSVD [37]	76.2
Kinect and Inertial [23]	79.1
Developed approach	86.3

Table 4.9. Average processing times of different components or modules

Component	Processing Time (in ms)
Detection using depth camera	0.4/action
Detection using inertial sensor	1.2/action
Classification using depth camera	2.5/action
Classification using inertial sensor	2.6/action

4.5 CONCLUSION

In this chapter, a computationally efficient action or gesture detection and recognition approach has been introduced which is capable of dealing with continuous action streams consisting of actions of interest occurring continuously and randomly among arbitrary actions of non-interest. Two differing sensing modalities of a depth camera and a wearable inertial sensor are used simultaneously to enable a robust performance via fusion of the skeleton joints and inertial data for both detection and recognition. The developed approach has been applied to the actions of the smart TV application in a subject-specific setting. The reported experimental results have demonstrated the effectiveness of the approach in detecting and recognizing smart TV gestures from continuous action streams. As future work, we plan to apply this approach to other applications involving a different set of actions of interest.

4.6 REFERENCES

[1] S. Paul, S. Ali, and M. Shah, “A 3-dimensional sift descriptor and its application to action recognition,” *Proceedings of the 15th ACM International Conference on Multimedia*, pp. 357–360, September 2007.

[2] Z. Khan, and W. Sohn, “Abnormal human activity recognition system based on R-transform and kernel discriminant technique for elderly home care,” *IEEE Transactions on Consumer Electronics*, vol. 57, no. 4, November 2011.

[3] W. Lao, J. Han, and P. DeWith, “Automatic video-based human motion analyzer for consumer surveillance system,” *IEEE Transactions on Consumer Electronics*, vol. 55, no. 2, May 2009.

[4] C. Chen, R. Jafari, and N. Kehtarnavaz. “A survey of depth and inertial sensor fusion for human action recognition,” Multimedia Tools and Applications, pp. 1–21, December 2015.

[5] J. Wang, Z. Liu, J. Chorowski, Z. Chen, and Y. Wu, “Robust 3d action recognition with random occupancy patterns,” *Computer Vision–ECCV 2012*, Springer Berlin Heidelberg, pp. 872–885, 2012.

[6] C. Chen, R. Jafari, and N. Kehtarnavaz, “Action recognition from depth sequences using depth motion maps-based local binary patterns,” *Proceedings of the IEEE Winter Conference on Applications of Computer Vision*, Waikoloa, HI, pp. 1092–1099, January 2015.

[7] A. Jalal, M. Uddin, and T. Kim, “Depth video-based human activity recognition system using translation and scaling invariant features for life logging at smart home,” *IEEE Transactions on Consumer Electronics*, vol. 58, no. 3, August 2012.

[8] X. Yang, and Y. Tian, “Eigenjoints-based action recognition using naive-bayes-nearest-neighbor,” *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*, pp. 14–19, June 2012.

[9] J. Baek, and B. Yun, “A sequence-action recognition applying state machine for user interface,” *IEEE Transactions on Consumer Electronics*, vol. 54, no. 2, May 2008.

[10] C. Chen, N. Kehtarnavaz, and R. Jafari, “A medication adherence monitoring system for pill bottles based on a wearable inertial sensor,” *Proceedings of the 36th Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, Chicago, IL, pp. 4983–4986, August 2014.

[11] D. Huang, S. Yao, Y. Wang, and F. De La Torre, “Sequential max-margin event detectors,” *Proceedings of the European Conference on Computer Vision*, Springer International Publishing, pp. 410–424, September 2014.

[12] A. Gaidon, Z. Harchaoui, and C. Schmid, “Actom sequence models for efficient action detection,” *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3201–3208, June 2011.

[13] M. Jain, J. Van Gemert, H. Jégou, P. Bouthemy, and C. Snoek, “Action localization with tubelets from motion,” *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 740–747, 2014.

[14] S. Ji, W. Xu, M. Yang, and K. Yu, “3D convolutional neural networks for human action recognition,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 1, pp. 221-231, January 2013.

[15] B. Mahasseni, and S. Todorovic, “Regularizing long short term memory with 3D human-skeleton sequences for action recognition,” *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3054-3062, 2016.

[16] F. Ordóñez, and D. Roggen, “Deep convolutional and LSTM recurrent neural networks for multimodal wearable activity recognition,” *Sensors*, vol. 16, no. 1, pp. 115, January 2016.

[17] N. Dawar, C. Chen, R. Jafari, and N. Kehtarnavaz, “Real-Time Continuous Action Detection and Recognition Using Depth Images and Inertial Signals,” *Proceedings of IEEE 26th International Symposium on Industrial Electronics (ISIE)*, pp. 1342-1347, June 2017.

[18] N. Dawar, and N. Kehtarnavaz, “Continuous Detection and Recognition of Actions of Interest among Actions of Non-interest using a Depth Camera,” *Proceedings of IEEE International Conference of Image Processing (ICIP)*, pp. 4227–4231, Beijing, China, September 2017.

[19] *Kinect for Windows*. Accessed: 2017. [Online]. Available: http://www.microsoft.com/en-us/kinectforwindows/

[20] A. Yang, R. Jafari, S. Sastry, and R. Bajcsy, “Distributed recognition of human actions using wearable motion sensor networks,” *Journal of Ambient Intelligence and Smart Environments*, vol. 1, no. 2, pp. 103–115, 2009.

[21] R. De Geest, E. Gavves, A. Ghodrati, Z. Li, C. Snoek, and T. Tuytelaars, “Online Action Detection,” *Proceedings of the European Conference on Computer Vision*, Springer International Publishing, pp. 269–284, October 2016.

[22] M. Marszalek, I. Laptev, and C. Schmid, “Actions in context,” *Proceedings of the IEEE Computer Vision and Pattern Recognition*, pp. 2929–2936, June 2009.

[23] C. Chen, R. Jafari, and N. Kehtarnavaz, "UTD-MHAD: A multimodal dataset for human action recognition utilizing a depth camera and a wearable inertial sensor," *Proceedings of the IEEE International Conference on Image Processing*, pp. 168–172, September 2015.

[24] G. Zhu, L. Zhang, P. Shen, and J. Song, "An Online Continuous Human Action Recognition Algorithm Based on the Kinect Sensor," *IEEE Sensors Journal*, vol. 16, no. 2, pp. 161, January 2016.

[25] D. Tax, and R. Duin, "Support vector data description," *Machine Learning*, vol. 54, no. 1, pp. 45–66, January 2004.

[26] L. Zhang, M. Yang, and X. Feng, "Sparse representation or collaborative representation: Which helps face recognition?" *Proceedings of the IEEE International Conference on Computer Vision*, pp. 471–478, November 2011.

[27] G. Zhu, L. Zhang, P. Shen, J. Song, L. Zhi and K. Yi, "Human action recognition using key poses and atomic motions," *Proceedings of the IEEE International Conference on Robotics and Biomimetics*, pp. 1209–1214, December 2015.

[28] F. Saki, and N. Kehtarnavaz. "Online frame-based clustering with unknown number of clusters," *Pattern Recognition*, vol. 57, pp. 70–83, September 2016.

[29] V. Vapnik, Statistical Learning Theory, Wiley, New York, 1998.

[30] C. Chen, R. Jafari, and N. Kehtarnavaz, "Improving human action recognition using fusion of depth camera and inertial sensors," *IEEE Transactions on Human-Machine Systems*, vol. 45, no. 1, pp. 51–61, February 2015.

[31] C. Chen, R. Jafari, and N. Kehtarnavaz, "A real-time human action recognition system using depth and inertial sensor fusion," *IEEE Sensors Journal*, vol. 16, no. 3, pp. 773–781, February 2016.

[32] C. Chen, R. Jafari, and N. Kehtarnavaz, "Fusion of depth, skeleton, and inertial data for human action recognition," *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 2712–2716, March 2016.

[33] C. Chen, R. Jafari, and N. Kehtarnavaz, "Action recognition from depth sequences using depth motion maps-based local binary patterns," *Proceedings of the IEEE Winter Conference on Applications of Computer Vision*, pp. 1092–1099, January 2015.

[34] J. Davis, and M. Goadrich, "The relationship between Precision-Recall and ROC curves," *Proceedings of the 23rd International Conference on Machine Learning*, pp. 233–240, June 2006.

[35] C. Goutte, and E. Gaussier, “A probabilistic interpretation of precision, recall and F-score, with implication for evaluation,” *Proceedings of the European Conference on Information Retrieval*, Springer Berlin Heidelberg, pp. 345–359, March 2005.

[36] T. Landgrebe, P. Pavel, and R. Duin, “Precision-recall operating characteristic (P-ROC) curves in imprecise environments,” *Proceedings of the 18th International Conference on Pattern Recognition*, vol. 4, pp. 123–127, August 2006.

[37] L. Zhou, W. Li, Y. Zhang, P. Ogunbona, D. Nguyen, and H. Zhang, “Discriminative key pose extraction using extended lc-ksvd for action recognition,” *International Conference on Digital lmage Computing: Techniques and Applications (DlCTA),* pp. 1-8, November 2014.

CHAPTER 5

DATA FLOW SYNCHRONIZATION OF A REAL-TIME FUSION SYSTEM TO DETECT AND RECOGNIZE SMART TV GESTURES*

Authors – Neha Dawar, Nasser Kehtarnavaz

The Department of Electrical Engineering, EC33

The University of Texas at Dallas

800 West Campbell Road

Richardson, Texas 75080-3021

* ©(2018) IEEE. Reprinted, with permission, from (Neha Dawar, and Nasser Kehtarnavaz, "Data flow synchronization of a real-time fusion system to detect and recognize smart TV gestures", Proceedings of IEEE International Conference on Consumer Electronics (ICCE), pp. 1-4, Las Vegas, January 2018.)

ABSTRACT

This chapter presents the data flow synchronization aspects of running a fusion system on a modern laptop in real-time. The fusion system uses two differing modality sensors of a Kinect depth camera and a wearable inertial sensor to detect and recognize a number of actions of interest from continuous action streams. This system is utilized to detect and recognize smart TV gestures when they are performed in a random and continuous manner among various actions of non-interest. It is shown that the processing times associated with the components of the developed fusion system lead to the real-time operation of the system on modern laptops.

5.1 INTRODUCTION

Human action/gesture recognition is an extensively researched topic in the literature. Different modality sensors, such as video cameras, depth cameras, and inertial sensors, have been used for this purpose. For example, video cameras were utilized in [1-2], depth cameras were utilized in [3-5], and inertial sensors were utilized in [6-7] to achieve human action/gesture recognition. It is well known that one modality sensor or camera cannot cope with all possible situations that occur in practice. Thus, to achieve more robust recognition under realistic operating conditions, in our previous works in [8-10], the two different modality sensors of a depth camera and a wearable inertial sensor were used together or simultaneously to perform human action/gesture recognition. In applications such as smart TV and gaming, it is required to automatically segment/detect as well as recognize actions of interest from continuous streams of activities. In [11-12], we developed two approaches to detect and recognize actions of interest from continuous streams of activities. A continuous stream denotes a number of actions of interest occurring randomly amongst arbitrary actions of non-interest in a continuous manner. The thrust of this chapter is on the data flow synchronization aspects of the detection and recognition system developed in our previous works. Such synchronization aspects are critical for having an actual real-time working fusion system. The algorithm implemented in this system uses both skeleton joint positions obtained from a Kinect depth camera and inertial signals obtained from an inertial sensor in parallel to perform detection and recognition. Detection is performed by both the sensing modalities and the common actions of interest detected by both the modalities are then recognized based on a decision-level fusion technique via two classifiers: one classifier operating on features derived from skeleton joint positions, and the other classifier operating on features derived from inertial signals.

The actions of interest considered in this work are the five smart TV gestures of ‘Waving a Hand’, ‘Flip to Left’, ‘Flip to Right’, ‘Counterclockwise Rotation’ and ‘Clockwise Rotation’. A subject-specific scenario for testing is considered in this work, that is training and testing are carried out by the same subject.

The rest of the chapter is organized as follows: An overview of our continuous action detection and recognition approach is provided in Section 5.2. The synchronization and implementation aspects of this approach are then discussed in Section 5.3. Section 5.4 includes the results of the experimentations carried out. Finally, the chapter is concluded in Section 5.5.

5.2 OVERVIEW OF DETECTION AND RECOGNITION FUSION SYSTEM

The sensor modalities used in the developed fusion system include a Kinect v2 camera and a wearable inertial sensor. The Kinect camera is a depth camera that captures depth images at an approximate rate of 30 frames per second and a resolution of 512×424 pixels. This camera is connected to a laptop via a USB port. The software tool Kinect SDK [13] provides the 3D positions of 25 skeleton joints. The wearable inertial sensor used is a small wireless body sensor reported in [14] which generates 3-axis acceleration and 3-axis angular velocity signals at a sampling rate of 200Hz. These signals are transmitted to a laptop via a Bluetooth link.

The algorithm run by the system first detects actions of interest from continuous action streams using skeleton joint positions and inertial signals individually. The fusion at the detection stage then takes only those actions that are detected by both the sensing modalities to perform recognition. Recognition is performed by two collaborative representation classifiers (CRC) [8] using skeleton joint positions and inertial signals separately, and the decisions of the two classifiers are fused to recognize the detected actions of interest.

Initially, segmentation of any action (whether an action of interest or an action of non-interest) from continuous action streams is performed based on skeleton joint positions using normalized relative orientations (NRO) of the skeleton joints [15]. As described in [12], the NROs of the joints are used to obtain a potential energy function. This potential energy function is compared to a threshold and the frames with a potential energy greater than this threshold are marked as action segments and those below it are marked as pause segments.

In parallel, segmentation of actions from continuous action streams is performed based on a difference acceleration signal computed using a reference acceleration signal. The difference acceleration signal is compared to a threshold. Frames with difference acceleration greater than the threshold are marked as action segments and the ones with difference acceleration below it are marked as pause segments. The actions segmented using the two modalities are then grouped into actions of interest or actions of non-interest using two one-class support vector data descriptor (SVDD) classifiers [16-17], one classifier using NROs obtained from skeleton joint positions as the features, and the other using the statistical features of *mean*, *variance*, *standard deviation* and *root mean square* of the difference acceleration signal as the features. The actions that are detected as actions of interest by both the modalities are then passed onto the recognition stage.

For recognition of the detected actions of interest, two computationally efficient l_2-regularized CRC classifiers are used. One CRC classifier uses the joint NROs as the features for those frames marked as part of an action of interest. Another CRC classifier uses the statistical features of inertial signals as the features.

The recognition decision is made by a decision fusion of the two classifiers via the logarithm opinion pool (LOGP) technique [18]. The recognized action is considered to be the one which generates the maximum probability obtained by the LOGP technique.

5.3 IMPLEMENTATION ASPECTS OF THE FUSION SYSTEM

As mentioned earlier, the publicly available Kinect SDK allows access to various features of Kinect including 25 skeleton body joints. The OpenCV library in C++ was used here to track these joints. Capturing of inertial signals from the wearable inertial sensor was also performed in the same program.

Due to the availability of various toolboxes in MATLAB like SVDD toolbox, the coding of the developed approach was done in MATLAB. Hence, a data flow synchronization technique was required to import data in MATLAB as soon as they were captured in C++. The following subsections discuss the process of training the system and the synchronization technique adopted to allow the flow of data between the two coding parts written in MATLAB and C++.

5.3.1 Training Process

The training was performed in an offline manner by asking a subject under test to wear the inertial sensor on his/her right wrist and stand in front of the Kinect v2 camera to perform the smart TV gestures. The subject was asked to repeat each of the five actions of interest 10 times and the data from both the sensors were recoded. It is important to note that actions of non-interest were not needed to be performed by the subject during the training phase.

5.3.2 Synchronization Technique for Actual Operation

During the actual operation or testing, the same subject performs the actions of interest in between some arbitrary actions of non-interest continuously in front of the Kinect camera while wearing the inertial sensor on the wrist. The acquisition of data from both the sensors is done in C++. The skeleton joint positions at each frame are read as binary files and saved in a folder. In addition, each sample of the 3-axis acceleration and 3-axis angular velocity signals from the inertial sensor is stored in a text file.

The MATLAB code begins by looking for the first frame of the skeleton joint positions in the folder and the first sample of the inertial signals in the text file. The search across skeleton joint positions and inertial signals is done in parallel as the rate of capturing data from the two sensors is different. Whenever the MATLAB code finds the first frame of the skeleton joint positions in the folder, it imports it into the MATLAB environment and increments the frame counter to look for the next frame. In parallel, the sample counter is incremented whenever the MATLAB code finds the current sample of the inertial signal being processed. The block diagram of the data flow synchronization process is shown in Figure 5.1.

As the frame and sample numbers increase, the size of the folder containing the binary files corresponding to the skeleton joint positions and the size of the text file containing the inertial signal samples increase. Hence, the computational complexity of looking for the binary file corresponding to a particular frame and inertial readings for a particular sample also increases. In order to keep the computational complexity low for the purpose of the system being able to operate in real-time, the binary files in the folder are deleted after every 100th frame considering that they have already been copied by the MATLAB code. That is, when the binary file corresponding to

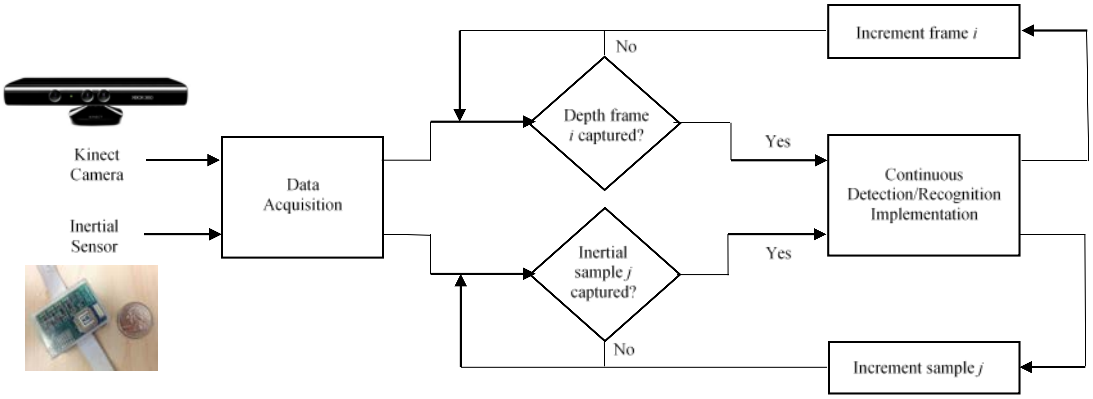


Figure 5.1. Block diagram of data flow synchronization

frame i is recorded in the folder, the file corresponding to frame $i - 100$ is deleted. This makes sure that the size of the folder containing these files never grows. Similarly, the size of the text file containing the inertial signal readings is maintained by deleting the reading corresponding to sample $j - 1000$ whenever the reading corresponding to sample j is recorded. The reason for keeping the buffer size for inertial signals high is that the sampling rate of the inertial sensor is higher as compared to the depth camera or sensor.

As the data from the two sensors are gathered in the MATLAB code, a moving average filter of window size 3 is applied to smoothen the data. The window size is kept small in order to avoid any processing delay. The filtered data are then used in the detection and recognition modules. There is a time stamp index value associated with each frame of the skeleton joint positions that provides the corresponding sample number of the inertial signals. Hence, while processing frame i, all the inertial samples from the index value at frame $i - 1$ to the index value at frame i are processed. The raw data in the MATLAB code are erased once the features from the skeleton joint positions at a particular frame and the inertial signals at a particular sample are gathered.

5.4 EXPERIMENTAL RESULTS

To evaluate the performance of the developed real-time continuous action detection and recognition fusion system, subjects under test were asked to perform continuous action streams consisting of actions of interest and actions of non-interest in front of a Kinect camera while wearing the inertial sensor. Each continuous action stream comprised the five actions of interest performed exactly once. A total of 30 such continuous action streams were collected and examined for the subject under test. The detection and classification accuracies of these action streams are reported in Table 5.1. Detection accuracy indicates the percentage of actions of interest detected from the continuous streams, while the classification accuracy indicates the percentage of correctly classified actions of interest among the detected actions. Of the 150 actions of interest in 30 continuous action streams performed, on average, 146 actions per subject were correctly detected resulting in a detection accuracy of 97.3%. The confusion matrix of the recognition rates obtained is shown in Table 5.2. In this table, WH denotes the gesture 'Waving a Hand, FL denotes the gesture 'Flip to Left', FR denotes the gesture 'Flip to Right', CCR denotes the gesture 'Counterclockwise Rotation' and CR denotes the gesture 'Clockwise Rotation'.

Table 5.3 shows the average processing times of the major components of the real-time fusion system. These times are reported for a laptop with 2.6GHz Intel Core i7 CPU and 12GB RAM. Note that the reported detection and classification times are not frame based as these modules are not executed per frame. Detection is performed whenever an action is segmented from a continuous action stream and classification is performed whenever the segmented action is detected as an action of interest. Also, note that the reported depth data flow time includes the time to extract depth and skeleton data from binary files and storing them in the correct format in

Table 5.1. Detection and recognition accuracies

	Accuracy
Detection	97.3%
Recognition	92.5%

Table 5.2. Confusion matrix of recognition rates (in %)

Action of interest	**WH**	**FL**	**FR**	**CCR**	**CR**
WH	100.0	-	-	-	-
FL	7.1	92.9	-	-	-
FR	3.4	-	96.6	-	-
CCR	3.3	-	3.3	93.4	-
CR	6.7	3.3	-	10.0	80.0

Table 5.3. Average processing times of the major components of the real-time fusion system

Component	**Average processing time (in ms)**
Depth data flow to MATLAB	14.1
Inertial data flow to MATLAB	8.9
Detection using depth data	1.6
Classification using depth data	4.1
Detection using inertial data	1.8
Classification using inertial data	2.8

matrices, which results in a higher processing time compared to the inertial data flow. The total processing time is noted to be 33.3ms, which meets the frame processing rate of about 30 depth image frames per second, thus generating a smooth real-time operation.

Figure 5.2 shows an example of a depth frame as it appears during the data acquisition and after the simultaneous data flow to the MATLAB code. A video clip of the system running in real-time can be viewed at http://www.utdallas.edu/~kehtar/SmartTVGestures-synchronization.avi.

5.5 CONCLUSION

This chapter has addressed the data flow synchronization aspects of running a fusion system in order to perform continuous human action/gesture detection and recognition in real-time. The

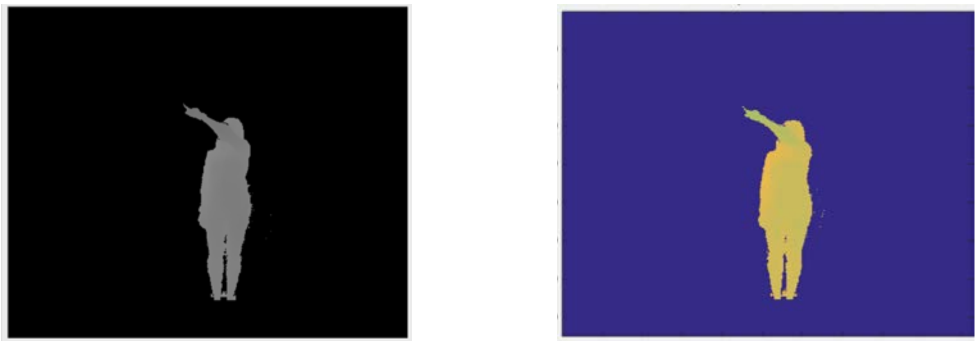


Figure 5.2. An example depth frame as it appears at the same time in the MATLAB and C++ code parts

fusion system uses a Kinect depth camera and a wearable inertial sensor. The developed system is applied to the smart TV application to detect and classify the five actions of interest in this application from continuous action streams consisting of these actions of interest performed randomly and continuously among arbitrary actions of non-interest. The data flow synchronization reported in this chapter has led to a smooth flow of data between the MATLAB and C++ code parts.

5.6 REFERENCES

[1] S. Paul, S. Ali, and M. Shah, “A 3-dimensional sift descriptor and its application to action recognition,” *Proceedings of the 15th ACM International Conference on Multimedia*, pp. 357–360, September 2007.

[2] Z. Khan, and W. Sohn, "Abnormal human activity recognition system based on R-transform and kernel discriminant technique for elderly home care," *IEEE Transactions on Consumer Electronics*, vol. 57, no. 4, November 2011.

[3] J. Wang, Z. Liu, J. Chorowski, Z. Chen, and Y. Wu, “Robust 3d action recognition with random occupancy patterns,” *Computer Vision–ECCV 2012*, Springer Berlin Heidelberg, pp. 872–885, 2012.

[4] R. Vemulapalli, F. Arrate, and R. Chellappa, “Human action recognition by representing 3d skeletons as points in a lie group,” *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 588-595, 2014.

[5] A. Jalal, M. Uddin, and T. Kim, "Depth video-based human activity recognition system using translation and scaling invariant features for life logging at smart home," *IEEE Transactions on Consumer Electronics*, vol. 58, no. 3, August 2012.

[6] J. Baek, and B. Yun, "A sequence-action recognition applying state machine for user interface," *IEEE Transactions on Consumer Electronics*, vol. 54, no. 2, May 2008.

[7] K. Altun, and B. Barshan, "Human activity recognition using inertial/magnetic sensor units," *Proceedings of the International Workshop on Human Behavior Understanding*, Springer Berlin Heidelberg, pp. 38–51, August 2010.

[8] C. Chen, R. Jafari, and N. Kehtarnavaz, "Improving human action recognition using fusion of depth camera and inertial sensors," *IEEE Transactions on Human-Machine Systems*, vol. 45, no. 1, pp. 51–61, February 2015.

[9] C. Chen, R. Jafari, and N. Kehtarnavaz, "A real-time human action recognition system using depth and inertial sensor fusion," *IEEE Sensors Journal*, vol. 16, no. 3, pp. 773–781, February 2016.

[10] C. Chen, R. Jafari, and N. Kehtarnavaz, "Fusion of depth, skeleton, and inertial data for human action recognition," *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 2712–2716, March 2016.

[11] N. Dawar, C. Chen, R. Jafari, and N. Kehtarnavaz, “Real-time continuous action detection and recognition using depth images and inertial signals,” *Proceedings of 26th IEEE International Symposium on Industrial Electronics (ISIE)*, pp. 1342–1347, June 2017.

[12] N. Dawar, and N. Kehtarnavaz, “Continuous detection and recognition of actions of interest among actions of non-interest using a depth camera,” *Proceedings of IEEE International Conference on Image Processing (ICIP)*, pp. 4227–4231, September 2017.

[13] *Kinect for Windows*. Accessed: 2017. [Online]. Available: http://www.microsoft.com/en-us/kinectforwindows/

[14] A. Yang, R. Jafari, S. Sastry, and R. Bajcsy, “Distributed recognition of human actions using wearable motion sensor networks,” *Journal of Ambient Intelligence and Smart Environments*, vol. 1, no. 2, pp. 103–115, 2009.

[15] G. Zhu, L. Zhang, P. Shen, and J. Song, "An online continuous human action recognition algorithm based on the Kinect sensor," *IEEE Sensors Journal*, vol. 16, no. 2, pp. 161, January 2016.

[16] D. Tax, and R. Duin, "Support vector data description," *Machine Learning*, vol. 54, no. 1, pp. 45–66, January 2004.

[17] F. Saki, and N. Kehtarnavaz. "Online frame-based clustering with unknown number of clusters," *Pattern Recognition*, vol. 57, pp. 70–83, September 2016.

[18] C. Chen, R. Jafari, and N. Kehtarnavaz, “Action recognition from depth sequences using depth motion maps-based local binary patterns,” *Proceedings of the IEEE Winter Conference on Applications of Computer Vision*, pp. 1092–1099, January 2015.

CHAPTER 6

A CONVOLUTIONAL NEURAL NETWORK-BASED SENSOR FUSION SYSTEM FOR MONITORING TRANSITION MOVEMENTS IN HEALTHCARE APPLICATIONS*

Authors – Neha Dawar, Nasser Kehtarnavaz

The Department of Electrical Engineering, EC33

The University of Texas at Dallas

800 West Campbell Road

Richardson, Texas 75080-3021

* ©(2018) IEEE. Reprinted, with permission, from (Neha Dawar, and Nasser Kehtarnavaz, "A convolutional neural network-based sensor fusion system for monitoring transition movements in healthcare applications", Proceedings of IEEE International Conference on Control & Automation (ICCA), pp. 482-485, Alaska, June 2018.)

ABSTRACT

This chapter presents a convolutional neural network-based sensor fusion system to monitor six transition movements as well as falls in healthcare applications by simultaneously using a depth camera and a wearable inertial sensor. Weighted depth motion map images and inertial signal images are fed as inputs into two convolutional neural networks running in parallel, one for each sensing modality. Detection and thus monitoring of the transition movements and falls are achieved by fusing the movement scores generated by the two convolutional neural networks. The results obtained for both subject-generic and subject-specific testing indicate the effectiveness of this sensor fusion system for monitoring these transition movements and falls.

6.1 INTRODUCTION

Among various population groups, recovering patients and elderly people require a high level of healthcare services, which is partially due to their mobility constraints. There has been a steady increase in healthcare costs for these population groups. As a result, there has been a shift in healthcare services from healthcare facilities to home environments [1]. An example of this shift is seen in human activity monitoring systems for healthcare applications. Deployment of monitoring systems in homes not only saves healthcare costs, but also provides elderly or patients with the independence of staying at home rather than staying in healthcare facilities [2]. An application of home-based monitoring systems involves fall detection. Falls, especially in elderly population, can be detrimental to their lives causing severe injuries and disabilities and may even lead to deaths [3]. Research has shown that 70% of the accidental deaths in the elderly population over the age of 75 are caused by falls [4]. Incorporating detection of falls in activity monitoring systems would lead to providing timely assistance and thus preventing life threatening delays.

Human action or gesture recognition plays a key role in activity monitoring systems for assistive living. Different types of sensors, e.g., vision and inertial [5-6], have been used to perform action or gesture recognition. In general, as noted in [7], a single modality sensing cannot deal with various realistic situations that occur in practice. Several action recognition sensor fusion systems have been developed by our research group in [8-12] by using two different modality sensors at the same time in order to achieve robust recognition under realistic operating conditions.

This work presents a monitoring system for the purpose of detecting or recognizing transition actions or movements, including falls, in healthcare applications. The actions of interest here include transitions between sitting, standing and lying down states. The system is designed to detect transitions occurring between these body states as well as the time spent in a particular state. To gain robustness, similar to our previous works, the information from a depth camera and a wearable inertial sensor are utilized at the same time or are fused together to detect transition movements as well as falls.

In previous fusion systems we have developed in [8-12], different classifiers including Maximum Entropy Markov Model (MEMM), Collaborative Representation Classifier (CRC) and Hidden Markov Models (HMM) were used to perform action or gesture recognition. Considering that Convolutional Neural Networks (CNNs) have recently gained widespread utilization for recognition applications [13], in this work we have developed a CNN-based sensor fusion system for detecting transition movements and falls.

The remainder of the chapter is organized as follows: Section 6.2 provides a description of the system setup and the dataset collected by it. Section 6.3 presents the details of the developed CNN-based sensor fusion system to perform monitoring of transition movements as well as falls. The results obtained by the system are discussed in Section 6.4. Finally, the chapter is concluded in Section 6.5.

6.2 SENSOR FUSION SYSTEM AND ITS DATASET

The inertial sensor used here is a small wireless wearable sensor reported in [14], which is capable of generating 3-axis acceleration signals and 3-axis angular velocity signals. Note that many other existing inertial sensors can be used in place of this inertial sensor. We have opted to use a depth camera instead of a video camera to avoid any privacy issue as no faces are identifiable in depth images. Basically, this sensor fusion system consists of a laptop computer, a Kinect v2 camera, and a wearable inertial sensor worn around the waist.

Since this system uses data from a Kinect depth camera and a wearable inertial sensor simultaneously, these exists no publicly available dataset where depth images and inertial signals for transition movements and falls are captured at the same time. Hence, to analyze the performance of the developed CNN-based sensor fusion system, a dataset was collected by the system. This dataset consists of the 6 transition actions or movements of ‘stand-to-sit’, ‘sit-to-stand’, ‘sit-to-lie’, ‘lie-to-sit’, ‘stand-to-lie’, ‘lie-to-stand’ as well as ‘fall’. Figure 6.1 shows an illustration of the sensor used and the transition movements between the body states monitored by the system.

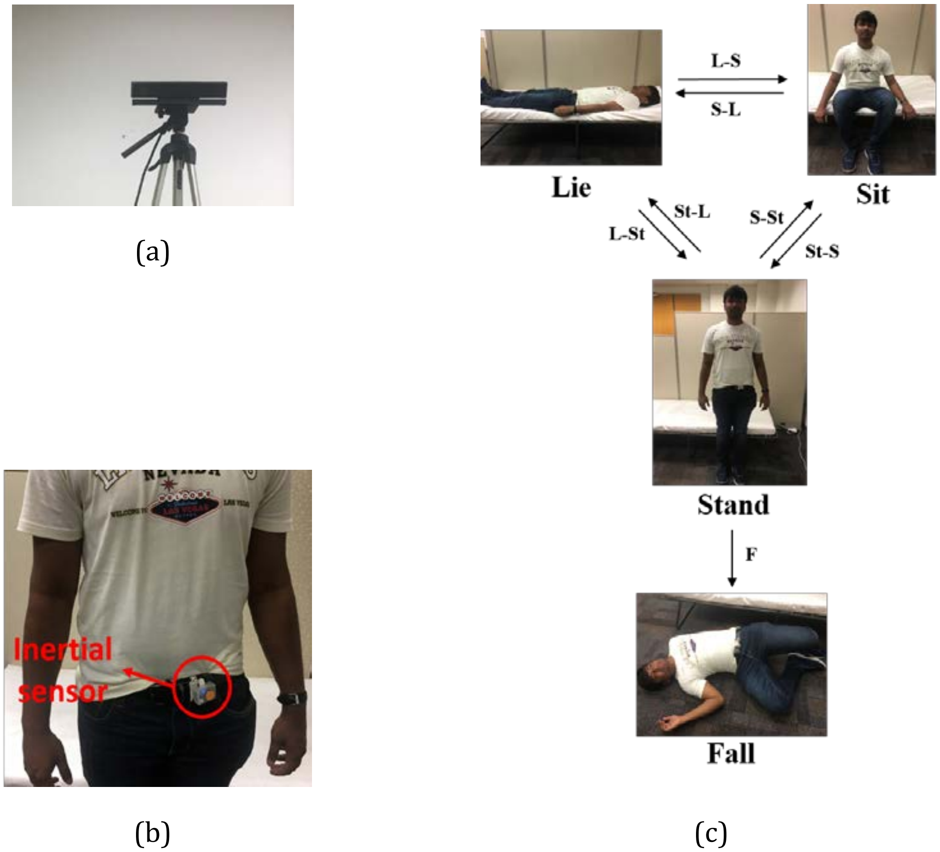


Figure 6.1. (a) Kinect depth camera, (b) wearable inertial sensor, and (c) illustration of the transistion movements and fall between the body states: St-S (stand-to-sit), St-L (stand-to-lie), S-St (sit-to-stand), S-L (sit-to-lie), L-S (lie-to-sit), L-St (lie-to-stand), F (fall)

A Kinect camera was installed at the corner of a room in which a bed had been placed. 12 different subjects were asked to wear the inertial sensor on their waist and perform the above mentioned 6 transition and fall actions. The depth images and inertial signals from the Kinect depth camera and the inertial sensor, respectively, were collected simultaneously and synchronized using the time stamp scheme discussed in [9]. The subjects were asked to repeat each of the 7 actions (6 transition movements plus fall) 10 times, resulting in a total of 70 transition movement samples per subject. The dataset is made available for public use and can be downloaded at this link www.utdallas.edu/~kehtar/UTD-Dataset-Transitions&Falls.htm.

6.3 CNN-BASED TRANSITION MOVEMENTS DETECTION

The developed CNN-based transition movement detection system incorporates two detection or recognition paths, one CNN for each sensing modality. The decision from each path is fused to reach a final detection outcome. The first step in each detection or recognition path involves generating appropriate images for the CNN of that path. For the depth camera path, so called weighted depth motion map (DMM) images as described in [15] are generated and used based on depth images captured by the depth camera. For the inertial sensor path, images are formed by row-wise stacking of signals that are translation and rotation invariant based on acceleration and angular velocity signals captured by the inertial sensor. The following subsections describe the formation of the images in the two CNN paths and the CNN architectures used.

6.3.1 Weighted DMM Images as First Path Images

Based on depth images, weighted DMM images are obtained by first projecting depth images onto three 2D projection maps corresponding to the front, side, and top views. In order to retain the temporal information in depth images, the projection maps are weighted as described in [15]. In this work, only the projection map corresponding to the front view is considered in order to gain computational efficiency. For a transition movement or action over N depth image frames, a weighted DMM is computed as follows:

$$DMM = \sum_{i=1}^{N-1} \left| map^{i+1} - map^{i} \right| * weight(i+1) \tag{6.1}$$

where map^{i} denotes the 2D projection map corresponding to the front view for the i^{th} depth image frame and N denotes the number of image frames in one action sequence. Motion areas of the projection map are weighted linearly in this work, that is the weight function of $weight(i) = i/N$ is considered. Weighting motion areas ensures that they appear brighter as the frame number in a transition movement or action sequence increases. Example weighted DMM images for the transition movements and fall are shown in Figure 6.2. Note that the size of the weighted DMM images is 512×424, which is the same as the size of the depth images captured by the Kinect depth camera. Weighted DMM images are resized to 100×100 to reduce the computational complexity for the CNN-based detection.

6.3.2 Inertial Signal Images as Second Path Images

For each transition movement, the acceleration and angular velocity signals are captured via the wearable inertial sensor. First, in order to make these signals invariant to the way the inertial sensor is worn on the body, the 18 heuristic orientation invariant transformation signals proposed in [16], 9 computed based on the overall acceleration and 9 computed based on the overall angular velocity, are concatenated row-wise to form a signal image as shown in Figure 6.3. Each row of this signal

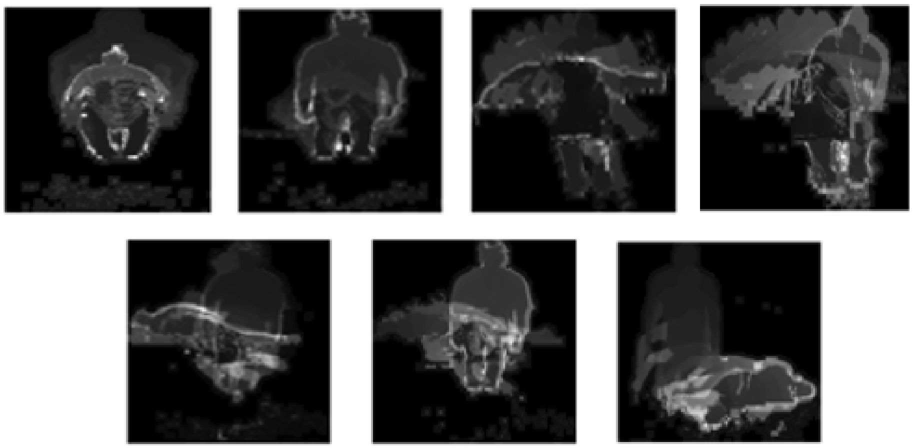


Figure 6.2. Example weighted DMM images of the transition movements and fall - from top left to botton right: ‘stand-to-sit’, ‘sit-to-stand’, ‘sit-to-lie’, ‘lie-to-sit’, ‘stand-to-lie’, ‘lie-to-stand’, and ‘fall’

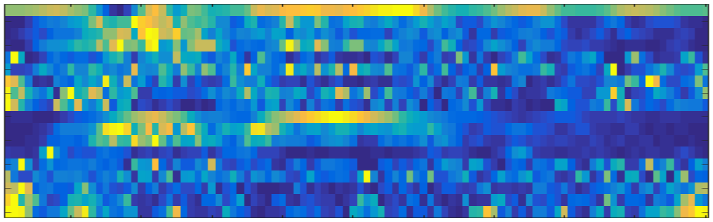


Figure 6.3. Example of a signal image generated from acceleration and angular velocity inertial signals

image is normalized to have values between 0 and 1. Since the length of each action sequence is different, in order to have the same size for all the images, the size of each signal image is mapped to a common size of 18 × 100. Note that the captured time series signals are converted to images here to form the CNN input.

6.3.3 CNN Architectures

The architecture of the two CNNs is different, one is designed for weighted DMM images and the other is designed for signal images. Each CNN provides scores for the transition movements and falls. These scores are combined to form a fused detection score. Figure 6.4 illustrates the CNN architectures in the two paths. Using the weighted DMMs, the first convolutional layer convolves the input images of size 100×100 with 32 filters of size 3×3. The convolution outputs are passed through a rectified linear unit (ReLU) activation layer. The second convolutional layer constitutes 32 filters of size 3×3 applied to the output of the ReLU layer. Another ReLU activation layer is used for mapping the output of the convolutional layer, followed by subsampling using a max pooling layer. The output of the pooling layer is then flattened, and a dense layer is used to map the output to a 1024×1 vector followed by a ReLU activation layer. The last layer is a fully connected layer, which uses a softmax activation to map the previous output into scores for the transition movements.

A similar architecture is used for the inertial signal images, except that the first two convolutional layers consist of 64 filters of size 7×7 and 64 filters of size 5×5, resepectively. There is an additional convolution layer comprising 64 filters of size 3×3 before the subsampling layer. Similar to the architecture used for the weighted DMMs, a dense layer is used to map the flattened output of the subsampling layer to a 1024×1 vector which is followed by a ReLU layer. Finally, the movement scores are obtained using a fully connected layer that uses a softmax activation.

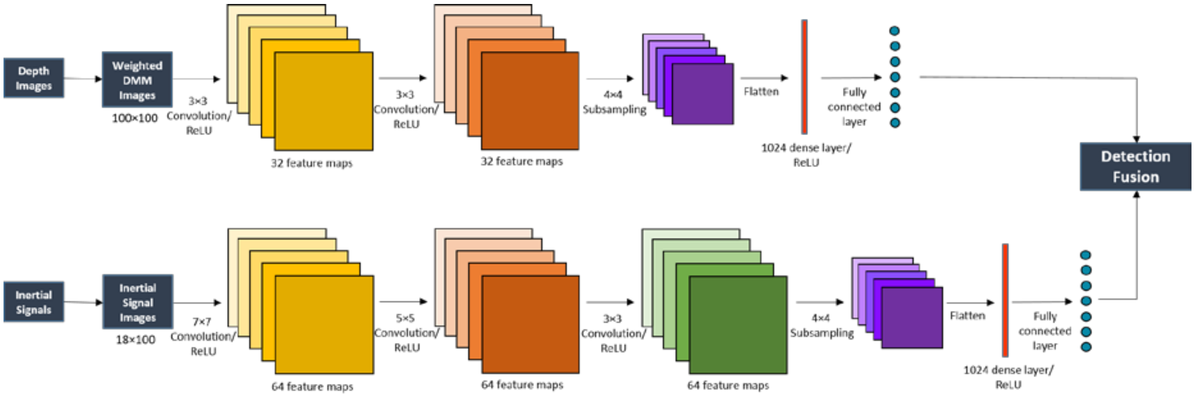


Figure 6.4. Developed CNN architectures of the sensor fusion system

The fusion of the two modalities is performed by accumulating the scores of the fully connected layers of the two CNNs. The movement with the maximum total score is then considered to be the detected movement.

6.4 EXPERIMENTAL RESULTS AND DISCUSSION

To examine the performance of the developed CNN-based sensor fusion system, we compared the outcome of the developed system to the outcome of our previously developed sensor fusion system discussed in [9]. Two different scenarios were considered in the experimentations. The first scenario involved a subject-generic testing based on the leave-one-out approach. In this scenario, the movements from one of the subjects were considered for testing and the movements from another subject were considered for validation and the movements of the remaining 10 subjects were used for training. This process was repeated for each subject, and the results were averaged. The second scenario involved a subject-specific testing in which training, validation and testing were done using the same subject. For each subject, 70% of the movement samples was used at random for training, 10% for validation, and the remaining 20% for testing. Again, this process was repeated for each subject and the results were averaged.

The weights of the CNN networks were learned by using a gradient descent optimizer with a learning rate of 1e-4. In order to control the overfitting of the CNN to the training data, the dropout scheme as explained in [17] was utilized. Note that the training data, especially in the subject-specific scenario, is normally insufficient for CNN training. Hence, the data augmentation scheme discussed in [18] was used in order to include some variations in the training data.

The detection outcome on the dataset stated earlier were compared with the fusion system introduced by Chen et al. in [9], in which depth images and inertial signals were also used at the same time by using two CRC classifiers. In order to provide a fair comparison, the samples used as validation in our CNN-based fusion system were used to obtain the parameter value λ of the CRCs. The detection or recognition accuracies of the two fusion systems for the subject-generic and subject-specific scenarios averaged over all the subjects are reported in Table 6.1. A noticeable improvement over the previously developed system can be seen in the detection accuracies when using the CNN-based system.

Table 6.1. Detection accuracies of transition movements and falls

Scenario	**Fusion System**	**Accuracy (%)**
Subject-generic	Chen *et al.* [9]	88.1
	CNN-based	96.5
Subject-specific	Chen *et al.* [9]	88.7
	CNN-based	97.6

Table 6.2. Confusion matrix exhibiting detection rates (in %) for the subject-generic scenario

	St-S	**St-L**	**S-St**	**S-L**	**L-S**	**L-St**	**F**
St-S	99.2	0.8	-	-	-	-	-
St-L	0.8	96.7	-	2.5	-	-	-
S-St	0.8	-	91.7	-	-	7.5	-
S-L	-	0.8	-	95.9	2.5	0.8	-
L-S	-	-	-	-	95.8	4.2	-
L-St	-	0.8	0.8	-	2.5	95.9	-
F	-	-	-	-	-	-	100.0

St-S: stand-to-sit, St-L: stand-to-lie, S-St: sit-to-stand,
S-L: sit-to-lie, L-S: lie-to-sit, L-St: lie-to-stand, F: fall

Table 6.3. Confusion matrix exhibiting detection rates (in %) for the subject-specific scenario

	St-S	**St-L**	**S-St**	**S-L**	**L-S**	**L-St**	**F**
St-S	95.8	4.2	-	-	-	-	-
St-L	-	100.0	-	-	-	-	-
S-St	-	-	91.7	-	-	8.3	-
S-L	-	4.2	-	95.8	-	-	-
L-S	-	-	-	-	100.0	-	-
L-St	-	-	-	-	-	100.0	-
F	-	-	-	-	-	-	100.0

St-S: stand-to-sit, St-L: stand-to-lie, S-St: sit-to-stand,
S-L: sit-to-lie, L-S: lie-to-sit, L-St: lie-to-stand, F: fall

The confusion matrices using the CNN-based system for the subject-generic and subject-specific scenarios are shown in Tables 6.2 and 6.3, respectively. In general and as seen in these tables, the subject-specific scenario generates higher detection rates compared to the subject-generic scenario due to the fact that data variation for the same subject is normally lower in the subject-specific scenario compared to the subject-generic scenario. Note that the subject-specific scenario is the scenario that is considered more suited for actual deployment in assistive living applications. As can be observed from the tables, the highest overlap in both subject-generic and subject-specific scenarios occurs between the transition movements of ‘lie-to-stand’ and ‘sit-to-stand’. This is due to the fact that the transition movement ‘lie-to-stand’ consists of the transition movements of ‘lie-to-sit’ and ‘sit-to-stand’ getting performed in series. Also, it is worth stating that as seen in the confusion matrices the transition movement ‘stand-to-lie’ and ‘fall’ do not overlap, although they may seem to be similar. There are three reasons for this: (i) lying in our case denotes lying on a bed while falls denote dropping on the floor from a standing position, (ii) a sudden change in acceleration normally appears in falls but not in lying down, and (iii) in lying down the transition occurs by first sitting and then lying whereas in falls no sitting occurs.

6.5 CONCLUSION

In this chapter, a CNN-based sensor fusion system to detect and thus monitor transition movements between body states as well as falls in healthcare applications has been developed which runs on any modern laptop computer. Images that are created from the data captured by a depth camera and a wearable inertial sensor are fed into two convolutional neural networks to achieve a robust detection of the transition movements and falls. The results obtained indicate the effectiveness of this sensor fusion system towards detecting the transition movements and falls.

6.6 REFERENCES

[1] A. Avci, S. Bosch, M. Marin-Perianu, R. Marin-Perianu, and P. Havinga, “Activity recognition using inertial sensing for healthcare, wellbeing and sports applications: A survey,” *Proceedings of 23rd International Conference on Architecture of Computing Systems (ARCS)*, pp. 1-10, 2010.

[2] S. Chernbumroong, S. Cang, A. Atkins, and H. Yu, “Elderly activities recognition and classification for applications in assisted living,” *Expert Systems with Applications*, vol. 40, no. 5, pp.1662-1674, 2013.

[3] J. Chen, K. Kwong, D. Chang, J. Luk, and R. Bajcsy, “Wearable sensors for reliable fall detection,” *Proceedings of 27th Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, pp. 3551-3554, 2005.

[4] G. Perolle, P. Fraisse, M. Mavros, and I. Etxeberria, “Automatic fall detection and activity monitoring for elderly,” *Proceedings of MEDETEL*, pp. 65-70, 2016.

[5] N. Dawar, and N. Kehtarnavaz, “Continuous detection and recognition of actions of interest among actions of non-interest using a depth Camera,” *Proceedings of IEEE International Conference of Image Processing (ICIP)*, pp. 4227–4231, Beijing, 2017.

[6] K. Altun, and B. Barshan, “Human activity recognition using inertial/magnetic sensor units,” *Proceedings of the International Workshop on Human Behavior Understanding*, Springer Berlin Heidelberg, pp. 38-51, 2010.

[7] C. Chen, R. Jafari, and N. Kehtarnavaz, “A survey of depth and inertial sensor fusion for human action recognition,” *Multimedia Tools and Applications*, pp. 1-21, 2015.

[8] C. Chen, R. Jafari, and N. Kehtarnavaz, “Improving human action recognition using fusion of depth camera and inertial sensors,” *IEEE Transactions on Human-Machine Systems*, vol. 45, no. 1, pp. 51–61, 2015.

[9] C. Chen, R. Jafari, and N. Kehtarnavaz, “UTD-MHAD: A multimodal dataset for human action recognition utilizing a depth camera and a wearable inertial sensor,” *Proceedings of the IEEE International Conference on Image Processing*, pp. 168-172, 2015.

[10] N. Dawar, C. Chen, R. Jafari, and N. Kehtarnavaz, “Real-time continuous action detection and recognition using depth images and inertial signals,” *Proceedings of IEEE 26th International Symposium on Industrial Electronics (ISIE)*, pp. 1342-1347, 2017.

[11] N. Dawar, and N. Kehtarnavaz, “Real-time continuous detection and recognition of subject-specific smart TV gestures via fusion of depth and inertial sensing,” *IEEE Access*, vol. 6, pp. 7019–7028, 2018.

[12] K. Liu, C. Chen, R. Jafari, and N. Kehtarnavaz, “Fusion of inertial and depth sensor data for robust hand gesture recognition,” *IEEE Sensors Journal*, vol. 14, no. 6, pp. 1898-1903, 2014.

[13] S. Ji, W. Xu, M. Yang, and K. Yu, “3D convolutional neural networks for human action recognition,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 1, pp. 221-231, 2013.

[14] A. Yang, R. Jafari, S. Sastry, and R. Bajcsy, “Distributed recognition of human actions using wearable motion sensor networks,” *Journal of Ambient Intelligence and Smart Environments*, vol. 1, no. 2, pp. 103-115, 2009.

[15] C. Chen, M. Liu, H. Liu, B. Zhang, J. Han, and N. Kehtarnavaz, “Multi-temporal depth motion maps-based local binary patterns for 3-D human action recognition,” *IEEE Access*, vol. 5, pp. 22590-22604, 2017.

[16] A. Yurtman, and B. Barshan, “Activity recognition invariant to sensor orientation with wearable motion sensors,” *Sensors*, vol. 17, no. 8, pp. 1838, 2017.

[17] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, “Dropout: A simple way to prevent neural networks from overfitting,” *Journal of Machine Learning Research*, vol. 15, no. 1, pp. 1929-1958, 2014.

[18] X. Cui, V. Goel, and B. Kingsbury, “Data augmentation for deep neural network acoustic modeling,” *IEEE/ACM Transactions on Audio, Speech and Language Processing (TASLP)*, vol. 23, no. 9, pp. 1469-1477, 2015.

CHAPTER 7

DATA AUGMENTATION IN DEEP LEARNING-BASED FUSION OF DEPTH AND INERTIAL SENSING FOR ACTION RECOGNITION[*]

Authors – Neha Dawar, Sarah Ostadabbas, Nasser Kehtarnavaz

The Department of Electrical Engineering, EC33

The University of Texas at Dallas

800 West Campbell Road

Richardson, Texas 75080-3021

[*] ©(2018) IEEE. Reprinted, with permission, from (Neha Dawar, Sarah Ostadabbas, and Nasser Kehtarnavaz, “Data augmentation in deep learning-based fusion of depth and inertial sensing for action detection”, IEEE Sensors Letters, October 2018.)

ABSTRACT

This chapter covers a deep learning-based decision fusion approach for action or gesture recognition via simultaneous utilization of a depth camera and a wearable inertial sensor. The deep learning approach involves using a convolutional neural network for depth images captured by a depth camera and a combination of convolutional neural network and long-short term memory network for inertial signals captured by a wearable inertial sensor, followed by a decision-level fusion. Due to the limited size of the training data, a data augmentation procedure is carried out by generating depth images corresponding to different orientations of the depth camera and by generating inertial signals corresponding to different orientations of the inertial sensor placement on the body. The results obtained indicate the positive impact of the decision-level fusion as well as the data augmentation on the recognition accuracies.

7.1 INTRODUCTION

In the past decade, human action or gesture recognition has been extensively studied in the literature using different sensing modalities involving video cameras, depth cameras, and inertial sensors, e.g., [1-3]. As discussed in [4], action recognition can reach higher accuracies under realistic operating conditions by fusing the two sensing modalities of depth and inertial. As an alternative to the classifiers reported in [5-6] for the fusion of depth and inertial sensing in action recognition, a deep learning-based approach is considered in this chapter. A convolutional neural network (CNN) is used to achieve recognition using depth images captured by a depth camera, and a combination of CNN and long-short term memory (LSTM) network is used to achieve recognition using inertial signals captured by a wearable inertial sensor. A decision-level fusion is then performed on the recognition outcomes of these networks.

The datasets examined in this work include the publicly available datasets provided in [6-8]. These datasets consist of synchronized data from a Kinect depth camera and a wearable inertial sensor described in [9]. The SmartTV dataset provided in [6] contains 5 hand gestures performed by 12 subjects. The Transition Movements dataset provided in [7] contains 7 body transition movements performed by 12 subjects. The UTD-MHAD dataset provided in [8] contains 27 actions consisting of a comprehensive set of movements performed by 8 subjects.

As discussed in [10], deep learning networks require a considerable amount of training data to perform effectively. Since the above datasets are rather limited in size for the purpose of adequately training a deep learning network, this work presents a data augmentation procedure to substantially increase the sizes of the datasets in order to achieve a more effective training of the networks.

In what follows, after describing an overview of the architectures of the deep learning networks used, it is discussed how the training set in the above datasets are augmented to increase the size of the training sets by nearly 100 times. A comparison in terms of recognition accuracy is then made in the results section between individual sensing modality and fusion decisions without and with the data augmentation.

7.2 DEEP LEARNING-BASED FUSION

In this section, a decision-level fusion using two deep learning networks, one for depth images and one for inertial signals, is described. A CNN network is used for depth images while a CNN+LSTM network is used for inertial signals. For the first network, weighted depth motions maps (WDMMs) [11] derived from depth images are used as inputs. For the second network, images formed by stacking inertial signals are used as inputs.

In order to cope with the limited size of the datasets, a data augmentation procedure is considered based on a simulator to generate more data reflecting different orientations of the depth camera and different orientations of the inertial sensor. In the subsection that follows, more details of the two deep learning architectures are stated followed by a section on the data augmentation procedure.

7.2.1 CNN and CNN+LSTM Architectures

An illustration of the layers of the CNN network used for depth images is shown in Figure 7.1(a). This network takes the WDMM images of a sequence, resized to 50×50 images, as its input. WDMM images are obtained by projecting depth images onto three orthogonal planes corresponding to the front, side and top views. To gain computational efficiency, only the projection images corresponding to the front view are used here. For a sequence of N depth images, let map^n represent the projection map corresponding to the front view at frame n. Then, WDMM is computed as follows [11]:

$$WDMM = \sum_{n=1}^{N-1} |map^{n+1} - map^n| * (n+1)/N \qquad (7.1)$$

As compared to the originally defined DMMs in [8], WDMMs assign higher weights to later movements in an action, thus allowing the temporal characteristics of the action to be captured. As a result, 2D convolution layers are used here to capture the spatial characteristics of WDMMs. As shown in Figure 7.1(a), the network uses 2 convolution layers, each of which are followed by a batch normalization layer, a Rectified Linear Unit (ReLU) layer and a subsampling or max pooling layer. The outputs of these layers are then passed onto a fully connected layer. A softmax function is used in the output layer generating a probability score for each class.

The architecture of the CNN+LSTM network used for inertial signals is shown in Figure 7.1(b). This network takes 8 time-series inertial signals as its input and passes them through 1D convolution and LSTM layers. The 8 time-series inertial signals consist of the 3-axis acceleration, the 3-axis angular velocity, the overall acceleration and the overall angular velocity signals. If a_x, a_y, a_z denote the 3-axis acceleration at a frame, the overall acceleration a is obtained as follows:

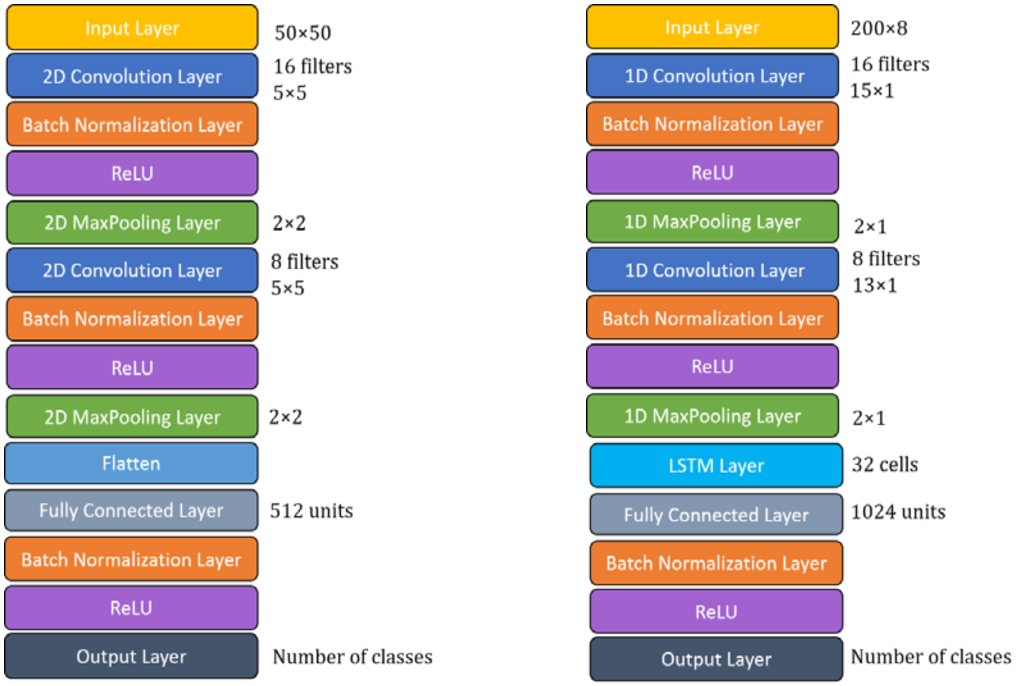
$$a = \sqrt{{a_x}^2 + {a_y}^2 + {a_z}^2} \qquad (7.2)$$

Similarly, if g_x, g_y, g_z denote the 3-axis angular velocity at a frame, the overall angular velocity g is obtained as follows:

$$g = \sqrt{{g_x}^2 + {g_y}^2 + {g_z}^2} \qquad (7.3)$$

The network is composed of a series of two 1D convolution layers and an LSTM layer. The convolution layers are followed by a batch normalization layer, ReLU layer and 1D max pooling layer. The output of these layers is passed onto a LSTM layer with 32 cells, followed by a fully connected layer. A softmax function is used in the output layer generating a probability score for each class.

The two networks are trained individually. For a test sequence, the class scores at the output layers of the CNN network operating on depth images and the CNN+LSTM network operating on inertial signals are multiplied to obtain a decision fusion score. The test sequence is assigned to the class with the highest fusion score.



(a) (b)

Figure 7.1. (a) CNN architectures for depth images, (b) CNN+LSTM architecture for inertial signals

7.3 DATA AUGMENTATION

As mentioned earlier, considering that deep learning networks require a considerable amount of data for their training, in order to cope with the limited size of the datasets, a data augmentation procedure is considered here. To perform augmentation on the depth data, different orientations or viewpoints of the camera are generated from the collected depth data. In other words, additional training data are generated by rotating the depth images in a way as if those images were captured from a different orientation or viewpoint of the depth camera. Different orientations or viewpoints of the camera are simulated by adding rotations about the three axes. Since a bounding box is placed around the WDMMs before resizing them, translating the depth camera would not lead to additional training data.

To imitate different viewpoints of the camera, a pixel (x, y) in a depth image is first converted to the 3D world coordinates (X, Y, Z) with Z denoting the depth value at the pixel (x, y) by using the following equations [12]:

$$X = Z(x - C_x)/f_x \tag{7.4}$$

$$Y = Z(y - C_y)/f_y \tag{7.5}$$

where (C_x, C_y) represents the center of the depth image and f_x and f_y are the focal lengths of the camera. For the Kinect v1 depth camera, $f_x = 580$ and $f_y = 580$ and for the Kinect v2 depth camera, $f_x = 365$ and $f_y = 365$.

Let the rotations of the camera about the three directions of x, y and z be represented by α, β and γ,respectively. Then, the transformation matrices can be written as follows [12]:

$$R_{Tx} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos(\alpha) & -\sin(\alpha) & Z \cdot \sin(\alpha) \\ 0 & \sin(\alpha) & \cos(\alpha) & Z \cdot (1 - \cos(\alpha)) \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{7.6}$$

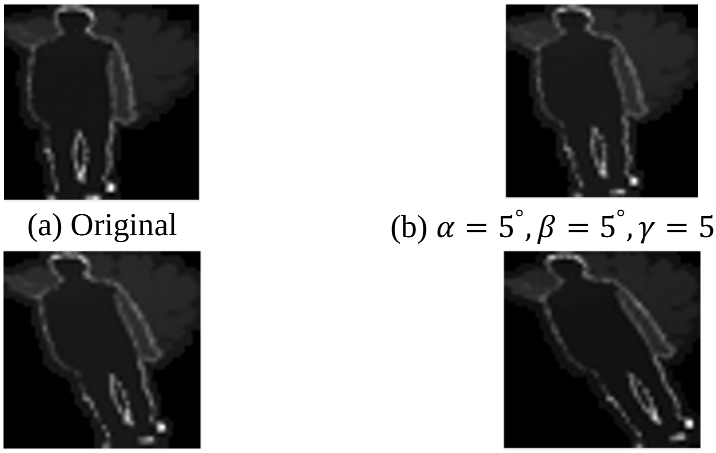
$$R_{Ty} = \begin{bmatrix} \cos(\beta) & 0 & \sin(\beta) & -Z \cdot \sin(\beta) \\ 0 & 1 & 0 & 0 \\ -\sin(\beta) & 0 & \cos(\beta) & Z \cdot (1 - \cos(\beta)) \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{7.7}$$

$$R_{Tz} = \begin{bmatrix} \cos(\gamma) & -\sin(\gamma) & 0 & 0 \\ \sin(\gamma) & \cos(\gamma) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{7.8}$$

The transformation to the new coordinates (X', Y', Z') after the rotations is then obtained as follows:

$$\begin{bmatrix} X' \\ Y' \\ Z' \\ 1 \end{bmatrix} = R_{Tx} R_{Ty} R_{Tz} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} \tag{7.9}$$

Data augmentation using WDMMs is achieved by considering random values of $\alpha,\ \beta,$ and γ. In addition to rotations about all the three axes, rotations about just one of the three axes and about any two of the three axes are also considered. To turn off the rotation about any of the axes, its corresponding rotation angle is set to zero, thus resulting in an identity transformation matrix for that axis. Figure 7.2 shows an original WDMM for the action ‘flip-to-left’ and examples of WDMMs generated for different viewpoints of the camera.



(a) Original

(b) $\alpha = 5^{\circ}, \beta = 5^{\circ}, \gamma = 5^{\circ}$

(c) $\alpha = 10^{\circ}, \beta = 10^{\circ}, \gamma = 10^{\circ}$

(d) $\alpha = 15^{\circ}, \beta = 15^{\circ}, \gamma = 15^{\circ}$

Figure 7.2. WDMM augmentation examples

A similar data augmentation procedure is carried out for the inertial signals. Different orientations of the inertial sensor placement on the body are considered. For a given inertial training sequence, new sequences are generated by considering different orientations of the sensor on the body. Let θ, φ, ρ represent the three axes rotation variations of the inertial sensor (called yaw, pitch and roll) from its original placement. Then, the transformation rotation matrices are obtained as follows [13]:

$$R_x = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\theta) & -\sin(\theta) \\ 0 & \sin(\theta) & \cos(\theta) \end{bmatrix} \tag{7.10}$$

$$R_y = \begin{bmatrix} \cos(\varphi) & 0 & \sin(\varphi) \\ 0 & 1 & 0 \\ -\sin(\varphi) & 1 & \cos(\varphi) \end{bmatrix} \tag{7.11}$$

$$R_z = \begin{bmatrix} \cos(\rho) & -\sin(\rho) & 0 \\ \sin(\rho) & \cos(\rho) & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{7.12}$$

Let a_x, a_y, a_z be the original 3-axis accelerations associated with a data sample. The new acceleration values a_x', a_y', a_z' after the rotations are obtained by using the following transformation equation:

$$\begin{bmatrix} a_x' \\ a_y' \\ a_z' \end{bmatrix} = R_x R_y R_z \begin{bmatrix} a_x \\ a_y \\ a_z \end{bmatrix} \tag{7.13}$$

The same transformation is considered for angular velocity signals. Similar to depth sequences, for randomly generated values of $\theta, \varphi,$ and ρ, new sequences are generated by rotations about just one axis, about all combinations of two axes and about all the three axes. Additional inputs to the deep learning networks are then formed from the newly generated signals. An example of 12° rotation of the inertial signals about all the axes is shown in Figure 7.3.

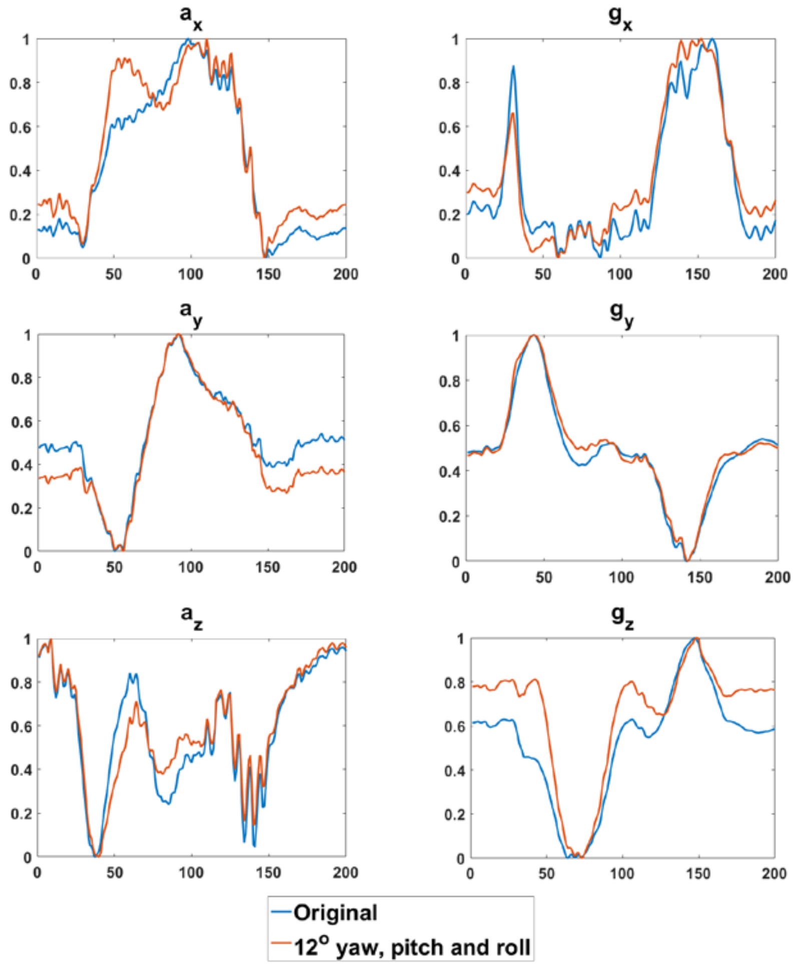


Figure 7.3. Inertial signals augmentation examples

7.4 EXPERIMENTAL RESULTS

The evaluation of the fusion approach was conducted by examining the three datasets: SmartTV, Transition Movements, and UTD-MHAD. The leave-one-out cross validation method was considered, in which the data from one of the subjects was removed from the training set and used for the validation. This process was repeated for all the subjects and the validation outcomes were averaged. The data augmentation was performed just on the training data, not the testing data. Both the collected training data and augmented training data were used for training the deep learning networks. For testing, the scores of the output layers of the two networks were multiplied in order to obtain the decision-level fusion scores. A test sequence was assigned to the class that generated the maximum fusion score. It is worth mentioning here that in addition to a decision-level fusion, a feature-level or data fusion was also considered in this work by inputting the data from the two sensors into a common network. It was found that the decision-level fusion by far was more effective than the feature-level or data fusion.

The recognition accuracies were obtained without and with the data augmentation as well as by using the individual sensing modalities. These accuracies are reported in Table 7.1. As can be seen from this table, the highest recognition accuracies were obtained by the fusion of depth and inertial sensing with the data augmentation. It is also worth noting that even when using an individual sensing modality, the data augmentation led to improvements of the recognition accuracies. The impact of the data augmentation was most noticeable for the UTD-MHAD dataset, which contains the data from the least number of subjects and most number of actions amongst the three datasets. It is worth pointing out that the accuracies reported in this table differ from the ones reported in [6] due to the following differences: (i) In [6], skeleton joint positions were used not the more generic depth images. Skeleton joints would work only for applications in which skeleton joints appear with no overlap. (ii) The testing done in [6] was performed in a subject specific manner, that is training and testing were performed on the same subject, while the testing done here was performed in a subject generic manner.

To provide an example of the performance improvement per action due to the data augmentation, the fusion recognition accuracies for the sport actions of the UTD-MHAD dataset are shown in

Figure 7.4 as a bar chart with and without using the data augmentation. The bar charts for the other datasets exhibit similar behavior and are not included here due to the lack of space.

7.5 CONCLUSION

In this chapter, a deep learning-based decision fusion approach for action recognition has been described based on depth images from a depth camera and inertial signals from a wearable inertial

Table 7.1. Recognition accuracies for different datasets when using individual sensing modalities versus decision fusion of the modalities

Dataset	Mode	# of training samples	Depth camera only	Inertial sensor only	Decision-level fusion
SmartTV	Without Augmentation	600	68.5%	75.0%	81.0%
	With Augmentation	63600	69.5%	80.0%	**86.8%**
Transition Movements	Without Augmentation	840	98.1%	89.8%	98.2%
	With Augmentation	89040	98.3%	93.8%	**99.1%**
UTD-MHAD	Without Augmentation	861	65.9%	52.6%	78.1%
	With Augmentation	91266	75.9%	81.5%	**89.2%**

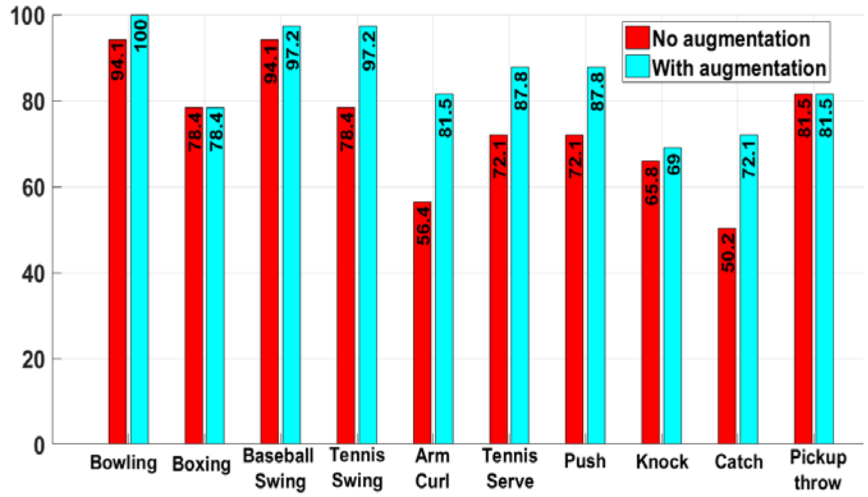


Figure 7.4. Bar chart showing fusion recognition accuracies for the sport actions in the UTD-MHAD dataset with and without the data augmentation

sensor. To deal with the limited size of the datasets studied, a data augmentation procedure has been developed which substantially increased the size of the datasets (by a factor of 10^2). The augmentation of depth images was achieved by mimicking different viewpoints of the depth camera, while the augmentation of inertial signals was achieved by imitating different orientations of the inertial sensor placement on the body. The results obtained by applying the developed fusion system to three publicly available synchronized depth and inertial datasets have shown that fusing the decisions of two deep learning networks, one operating on depth images and the other on inertial signals, improves the accuracy of recognition compared to the situations when individual sensing are used and is further improved by using the data augmentation procedure.

7.6 REFERENCES

[1] W. Lao, J. Han, and P. De With, "Automatic video-based human motion analyzer for consumer surveillance system," *IEEE Transactions on Consumer Electronics*, vol. 55, no. 2, pp. 591–598, May 2009.

[2] J. Wang, Z. Liu, Y. Wu and J. Yuan, "Mining actionlet ensemble for action recognition with depth cameras," *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1290–1297, June 2012.

[3] R. Xu, S. Zhou, and W. Li, “MEMS accelerometer based nonspecific-user hand gesture recognition,” *IEEE Sensors Journal*, vol. 12, no. 5, pp.1166–1173, May 2012.

[4] C. Chen, R. Jafari, and N. Kehtarnavaz. "A survey of depth and inertial sensor fusion for human action recognition," *Multimedia Tools and Applications*, pp. 1–21, December 2017.

[5] K. Liu, C. Chen, R. Jafari, and N. Kehtarnavaz, “Fusion of inertial and depth sensor data for robust hand gesture recognition,” *IEEE Sensors Journal*, vol. 14, no. 6, pp. 1898–1903, June 2014.

[6] N. Dawar and N. Kehtarnavaz, “Real-time continuous detection and recognition of subject-specific smart TV gestures via fusion of depth and inertial sensing,” *IEEE Access*, vol. 6, pp. 7019–7028, January 2018.

[7] N. Dawar and N. Kehtarnavaz, “A convolutional neural network-based sensor fusion system for monitoring transition movements in healthcare applications,” *Proceedings of IEEE International Conference on Control and Automation (ICCA)*, pp. 482-485, Anchorage, AK, June 2018.

[8] C. Chen, R. Jafari, and N. Kehtarnavaz, “UTD-MHAD: A multimodal dataset for human action recognition utilizing a depth camera and a wearable inertial sensor,” *Proceedings of IEEE International Conference on Image Processing*, pp. 168–172, Quebec City, Canada, September 2015.

[9] A. Yang, R. Jafari, S. Sastry, and R. Bajcsy, “Distributed recognition of human actions using wearable motion sensor networks,” *Journal of Ambient Intelligence and Smart Environments*, vol. 1, no. 2, pp. 103-115, 2009.

[10] S. Liu, Y. Yin, S. Ostadabbas, “In-Bed Pose Estimation: Deep Learning with Shallow Dataset,” 2018. [arXiv:1711.01005]

[11] C. Chen, M. Liu, H. Liu, B. Zhang, J. Han, and N. Kehtarnavaz, “Multi-temporal depth motion maps-based local binary patterns for 3-D human action recognition,” *IEEE Access*, vol. 5, pp. 22590-22604, October 2017.

[12] P. Wang, W. Li, Z. Gao, J. Zhang, C. Tang and P. Ogunbona, "Action recognition from depth maps using deep convolutional neural networks," *IEEE Transactions on Human-Machine Systems*, vol. 46, no. 4, pp. 498–509, August 2016.

[13] A. Yurtman, and B. Barshan, “Activity recognition invariant to sensor orientation with wearable motion sensors,” *Sensors*, vol. 17, no. 8, p.1838, August 2017.

CHAPTER 8

ACTION DETECTION AND RECOGNITION IN CONTINUOUS ACTION STREAMS BY DEEP LEARNING-BASED SENSING FUSION*

Authors – Neha Dawar, Nasser Kehtarnavaz

The Department of Electrical Engineering, EC33

The University of Texas at Dallas

800 West Campbell Road

Richardson, Texas 75080-3021

* ©(2018) IEEE. Reprinted, with permission, from (Neha Dawar, and Nasser Kehtarnavaz, "Action detection and recognition in continuous action streams by deep learning-based sensing fusion", IEEE Sensors Journal, September 2018.)

ABSTRACT

This chapter presents a deep learning-based sensing fusion system to detect and recognize actions of interest from continuous action streams that contain actions of interest occurring continuously and randomly among arbitrary actions of non-interest. The sensors used in the fusion system consist of a depth camera and a wearable inertial sensor. A convolutional neural network is utilized for depth images obtained from the depth sensor and a combination of convolutional neural network and long short term memory network is utilized for inertial signals obtained from the inertial sensor. Each sensing modality first performs segmentation of all actions and then detection of actions of interest for a particular application. A decision-level fusion of the two sensing modalities is carried out to achieve the recognition of the detected actions of interest. The developed fusion system is examined for two applications: one involving transition movements for home healthcare monitoring and the other involving smart TV hand gestures. The results obtained show the effectiveness of the developed fusion system in dealing with realistic continuous action streams.

8.1 INTRODUCTION

Human action or gesture recognition has enabled natural interfacing between humans and computers, and has already found its way into consumer electronics products. Many applications have benefitted from human action or gesture recognition. For example, human action recognition has been increasingly used for activity monitoring of the elderly population in home environments to address the steady increase in healthcare costs [1].

Different sensing modalities including RsGB cameras, e.g., [2], [3], depth cameras, e.g., [4], [5], and inertial sensors, e.g., [6], [7], have been mostly utilized individually for human action or gesture recognition. As discussed in our previous works [8]-[10], action or gesture recognition can be made more robust by fusing decisions from two differing modality sensors as compared to a single modality sensor.

In the great majority of works reported in the literature on action or gesture recognition, actions or gestures of interest are already segmented from action streams. To operate a human computer interaction system in a real-world setting, it is required that the actions of interest are detected from unseen continuous action streams in which they occur randomly and continuously amongst arbitrary actions of non-interest or no action. This real-world setting is by far a more challenging scenario as compared to the scenario where action streams are segmented manually such that segments contain only one action of interest. Detection of actions of interest from continuous action streams requires first segmenting all possible actions, regardless whether they are actions of interest or actions of non-interest, followed by identifying and classifying the actions of interest for a particular application. In our previous works [11]-[13], several fusion approaches were developed to detect and recognize smart TV gestures from continuous action streams by using skeleton joint positions obtained from a depth camera and inertial signals obtained from an inertial sensor. In [14], a data flow synchronization technique was developed to enable the real-time implementation of our fusion approaches.

Most of the previously developed fusion systems use handcrafted features together with classifiers such as Hidden Markov Model (HMM), Collaborative Representation Classifier (CRC), and Maximum Entropy Markov Model (MEMM) [11]-[15]. With the growing popularity of deep learning neural networks due to their high performance in various recognition tasks, in particular Convolutional Neural Networks (CNN) [16] and Long Short Term Memory (LSTM) networks [17], a CNN+LSTM-based fusion system to automatically detect and recognize actions of interest from continuous action streams has been developed in this work. The developed fusion system is used to detect actions of interest from continuous action streams for two applications including human body transition movements monitoring and smart TV hand gesture recognition. The actions of interest in the transition movements monitoring application involve transitions between the body states of sitting, standing and lying down. Considering the importance of fall detection monitoring for elderly and patients [18], in addition to the transition movements, falls are also monitored and detected here.

The fusion system developed in this chapter utilizes a depth camera and a wearable inertial sensor simultaneously to perform continuous action detection and recognition. Unlike video cameras, depth cameras do not provide identifying facial information thus avoiding any privacy concern. A continuous action dataset is also made available in this chapter for public use. This dataset consists of synchronized depth images and inertial signals associated with body transition movements as well as falls that are performed in a continuous and random manner in between various actions of non-interest. In addition to this dataset, our continuous action dataset (named UTD-CAD) in [13], which consists of smart TV hand gestures performed continuously and randomly in between various actions of non-interest is also examined here. Noting that training a CNN or LSTM network often requires very large datasets, a data augmentation step is carried out to address the limited size of the above continuous datasets for CNN and LSTM training.

Basically, this work constitutes the first attempt at developing a deep learning-based fusion system based on a depth camera and an inertial sensor for the purpose of detecting and recognizing actions of interest of an application that are performed continuously and randomly in between arbitrary actions of non-interest.

The rest of the chapter is organized as follows. An overview of related works appears in Section 8.2. Section 8.3 covers a description of the transition movements dataset collected for this study, which is provided for public use. Section 8.4 covers the details of the developed deep learning-based fusion system. The experimental results and their discussion are then presented in Section 8.5. Finally, the chapter is concluded in Section 8.6.

8.2 OVERVIEW OF RELATED WORKS

The bulk of research on action or gesture recognition involves the use of a single modality sensor. However, there are limitations associated with using a single modality sensor when performing action or gesture recognition in real-world settings [19] due to high intra-class variations and low inter-class variations in the actions performed for a particular application. No modality sensing can cope with such variations perfectly or flawlessly. Fusion is a way to address such limitations of using a single modality sensing. In [15], a fusion system using information from a depth camera and an inertial sensor was developed to achieve more robust gesture recognition. In [8], depth motion maps derived from depth images and statistical features derived from inertial signals were fused to achieve improved action recognition. The use of three data modalities of depth images, skeleton joint positions, and inertial signals was reported in [10].

Furthermore, in most action or gesture recognition approaches, actions or gestures are considered to be segmented actions or gestures with the start and end of the actions or gestures already known or manually identified. In [11], we reported an approach to detect and recognize actions of interest performed continuously and randomly amongst unknown actions of non-interest using skeleton joint positions obtained from a depth camera and inertial signals obtained from a wearable inertial sensor. Skeleton joint positions were used to perform detection and recognition while inertial signals were used to enhance the performance of recognition by removing false positives. In [12], we used skeleton joint positions to detect actions of interest from continuous action streams while recognition was achieved by fusing the outcomes of two CRC classifiers, one acting on skeleton joint positions and the other on depth images. In [13], [14], we reported a fusion approach at both detection and recognition stages based on skeleton joint positions and inertial signals. In these works, the detection of actions of interest from continuous actions streams was achieved using one-class Support Vector Data Descriptor (SVDD) classifiers and the classification of the detected actions of interest was achieved using CRC classifiers.

Recently, deep learning neural networks, in particular CNN and LSTM, have been increasingly used for action and gesture recognition based on a single modality sensor. For example, weighted hierarchical depth motion maps (WHDMM) were used in a three channel CNN in [20] to improve the recognition performance. In [21], a 3D CNN was used to learn the spatio-temporal features from raw depth sequences and it was combined with the feature vectors obtained from skeleton joint positions. In [22], both CNN and LSTM networks were considered for depth image sequences to achieve recognition. In [23], inertial signals from a set of body worn sensors were used and fed as images into a CNN network to recognize human activity. In [24], CNN and LSTM layers were combined to achieve action recognition using information from multiple wearable inertial sensors. In [25], shallow features of inertial signals were used along with deep features extracted by a CNN network to achieve recognition.

The work reported in this chapter differs from all of the previous works in the following manner. In comparison to single modality sensor solutions reported in the literature to perform action recognition, a fusion system is developed in this work by using CNN and LSTM networks to detect and recognize actions of interest from continuous action streams. Detection and recognition are performed for each of the two sensing modalities in parallel followed by a decision-level fusion. CNN is used to learn the spatio-temporal features from depth images, while both CNN and LSTM are used to learn the temporal features from inertial signals.

8.3 CONTINUOUS DATASETS

Considering the unavailability of a public domain continuous dataset where depth images and inertial signals are captured simultaneously, a dataset is collected in this work for the transition movements application and is made available for public use. The depth images in the dataset are captured by a Microsoft Kinect v2 depth camera at a rate of approximately 30 frames per second and a resolution of 512×424. Examples of background subtracted depth images captured by this camera are provided in [13]. The camera is connected to a laptop computer via a USB port. The inertial signals are captured by the wearable initial sensor reported in [26] at a rate of 200Hz. These signals consist of 3-axis acceleration and 3-axis angular velocity signals, which are transmitted via a Bluetooth link to the laptop computer running the fusion system software. The data from the two sensors are synchronized based on the time stamp scheme described in [27]. Basically, time stamps of depth image frames are used as reference and inertial signals samples with the time stamp closest to a particular depth image frame are aligned with that depth image frame.

To collect data, a bed was placed in a room that had the depth camera installed at the room corner near the ceiling. The inertial sensor was worn on the waist while performing the actions. The dataset collected consists of 6 transition movements between the body states of sitting, standing and lying down, as well as falling down, thus forming these actions of interest ‘stand-to-sit’, ‘sit-to-stand’, ‘stand-to-lie’, ‘lie-to-stand’, ‘sit-to-lie’, ‘lie-to-sit’, and ‘fall’. Figure 8.1 illustrates these transition movements between the body states. The continuous testing dataset was collected from 5 different subjects and a total of 5 continuous sets were collected from each subject resulting in a

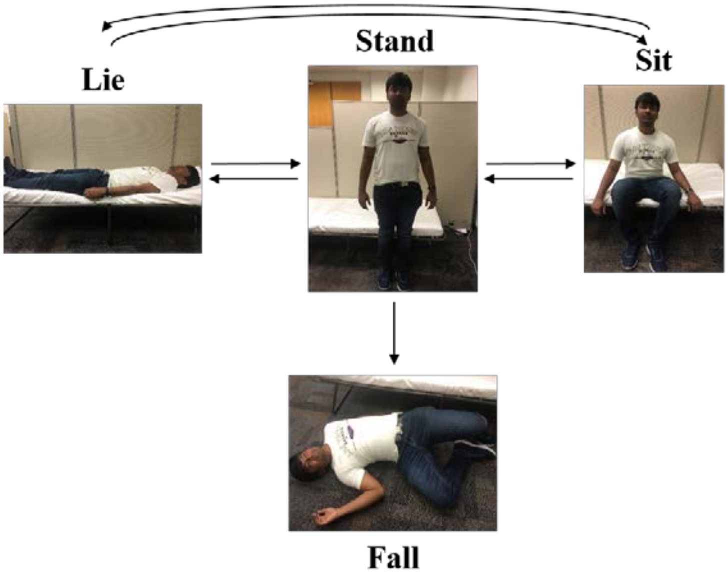


Figure 8.1. Illustration of transition movements between the body states as well as fall in the continuous transition movements dataset: stand-to-lie, lie-to-stand, stand-to-sit, sit-to-stand, lie-to-sit, sit-to-lie, and fall

total of 25 continuous testing sets. Each continuous set contains the above 7 actions of interest performed in a continuous and random manner in between arbitrary actions of non-interest such as stretching, reading a book, drinking water, eating, combing, etc. The subjects were given complete freedom to perform any actions of non-interest as per their choice.

The collected continuous dataset was used only for testing or the operation phase. Training of the neural networks was performed using the segmented transition movement and fall actions provided in [28]. This dataset consists of segmented action data from 12 subjects. The continuous transition movement dataset collected in this work is made available for public use at this link: www.utdallas.edu/~kehtar/UTD-Dataset-ContinuousTransitionMovements.htm.

For the smart TV application, the continuous dataset for the smart TV hand gestures in [13] is used here. The actions or gestures of interest in this dataset consist of 'waving a hand', 'flip to left', 'flip to right', 'counterclockwise rotation' and 'clockwise rotation'. For these gestures, the inertial sensor was worn on the wrist. This dataset contains 5 continuous gesture streams each containing the above 5 gestures from 12 subjects.

8.4 DEVELOPED DEEP LEARNING-BASED CONTINUOUS DETECTION AND RECOGNITION FUSION SYSTEM

The developed fusion system carries out detection and recognition for each of the two differing sensing modalities of a depth camera and an inertial sensor, followed by a decision level fusion. The depth camera path uses a CNN network, while the inertial sensor path uses a CNN+LSTM network to perform detection and recognition. Segmentation is carried out along each modality path. The segmented actions are then passed through the CNN or CNN+LSTM networks of their respective paths to detect actions of interest or actions of non-interest and then to classify the

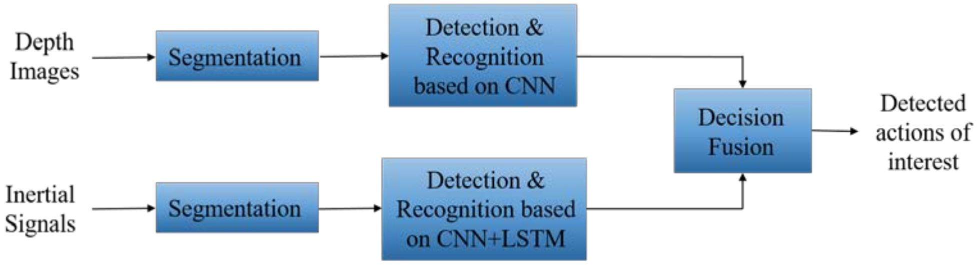

Depth Images
Segmentation
Detection & Recognition based on CNN
Inertial Signals
Segmentation
Detection & Recognition based on CNN+LSTM
Decision Fusion
Detected actions of interest

Figure 8.2. Block diagram of the detection and recognition fusion system

detected actions of interest. A decision-level fusion is conducted based on the output of the two paths. Figure 8.2 illustrates a block diagram of the overall detection and recognition fusion system. Note that detection here means identifying the segmented actions as actions of interest or as actions of non-interest, while recognition means classifying the detected actions of interest.

8.4.1 Segmentation

For segmentation of the transition movements in the depth sensor path, the centroid (c_x, c_y) of the background subtracted depth images is obtained as follows:

$$c_x = \frac{\sum_{i=1}^{N} x_i m_i}{\sum_{i=1}^{N} m_i}, \qquad c_y = \frac{\sum_{i=1}^{N} y_i m_i}{\sum_{i=1}^{N} m_i} \tag{8.1}$$

where (x_i, y_i) denotes a pixel location with m_i representing its intensity value. A sequence of centroids $C = (C^1, C^2, \dots, C^t, \dots)$ is then obtained where $C^t = (c_x, c_y)^t$ represents the centroid at frame t. A centroid difference C_d^t at t^{th} frame is then obtained as follows:

$$C_d^t = C^t - C^{t-1} \tag{8.2}$$

Noise related small fluctuations of centroid differences during no action are eliminated by setting centroid difference values below 5% level of the maximum centroid difference to zero. Frames with centroid differences above this level are used to denote the presence of movement or action. An example of centroid differences for a continuous action stream is shown in Figure 8.3.

A similar segmentation process is carried out to segment actions using the inertial signals. If g_x^t, g_y^t, g_z^t denote the 3D angular velocities at a frame t, the angular velocity G^t at this frame is obtained as follows:

$$G^t = \sqrt{{g_x^t}^2 + {g_y^t}^2 + {g_z^t}^2} \tag{8.3}$$

Let $G = (G^1, G^2, \dots, G^t, \dots)$ represent a sequence of angular velocities. An angular velocity difference G_d^t at frame t is then obtained as follows:

$$G_d^t = G^t - G^{t-1} \tag{8.4}$$

Noise related small fluctuations of angular velocity differences during no action are eliminated by setting angular velocity differences below 5% level of the maximum angular velocity difference to zero. Frames with angular velocity differences above this level are used to denote the presence of movements or actions. An example of angular velocity differences for a continuous action stream is shown in Figure 8.4.

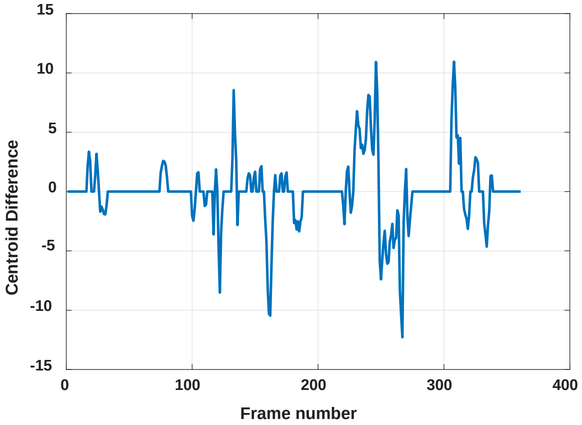


Figure 8.3. An example of centroid differences of a continuous action stream

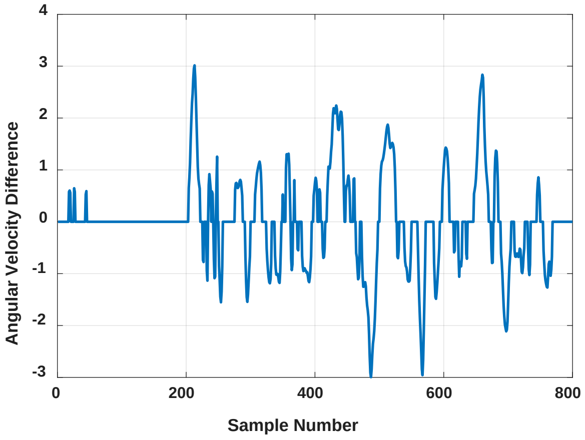


Figure 8.4. An example of angular velocity differences of a continuous action stream

For continuous smart TV hand gesture dataset, the same technique described in [13] is employed for segmentation. It is to be noted that the hand gestures involved in the continuous smart TV gestures dataset have the same starting and ending point. In other words, all the actions of interest start and end at more or less the same reference point. Hence, it is unnecessary to obtain the centroid or angular velocity differences for the purpose of capturing the peaks and dips of an action stream. Thresholding of the signals is sufficient to detect the start and end of the hand gestures. Furthermore, since the smart TV gestures are performed relatively close to the camera, skeleton joint positions are used as they provide more reliable information for segmentation instead of the depth image centroids considering that centroid positions do not change much when performing the hand gestures.

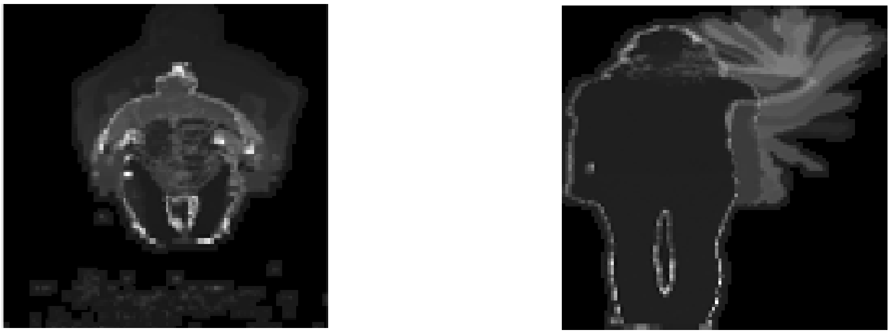
8.4.2 CNN Architecture for Depth Images

A two stream CNN is used here to detect and recognize actions based on depth images, that is there are two separate streams that use depth images to detect and recognize actions of interest from continuous action streams. The first stream takes raw background subtracted depth images as its input and uses 3D convolutional layers to obtain deep features from the segmented actions as was reported in [21]. The raw images are resized to a common size of 32×32 and a fixed number of frames is evenly extracted from each action sequence. These resized depth images are used as the input to the first stream. For the transition movements dataset, 15 depth images per action sequence are used as the input noting that the average length of each action of interest is 15 frames. Similarly, 25 depth images are used for the continuous smart TV hand gesture dataset.

As discussed in [25], when the dataset size is limited, one CNN stream alone is not able to capture the hierarchy of features in its entirety. Hence, another stream of CNN is used by considering handcrafted features of the segmented actions as its inputs. These features are the weighted depth motion map (DMM) images of the actions. To obtain the weighted DMM images, the depth images are projected onto three orthogonal planes corresponding to the front, side and top views. In order to keep the computational complexity low, only the projection onto the front view is utilized here. The projection map is weighted to obtain the DMM image. If map^{n} represents the front view projection map for n^{th} frame, for a sequence of N depth frames, the weighted DMM is computed as follows [29]:

$$DMM = \sum_{n=1}^{N-1} |map^{n+1} - map^{n}| * weight(n+1) \qquad (8.5)$$

The motion areas are weighted linearly, that is $weight(n) = n/N$. The advantage of using weighted DMMs instead of traditional DMMs [8] is that motion areas around later frames appear brighter than earlier frames, thus making it easier to differentiate between the reversed transition movements such as ‘sit-to-stand’ and ‘stand-to-sit’, which would otherwise have a similar DMM. An example of a weighted DMM image from the transition movements dataset and an example



(a) (b)

Figure 8.5. Examples of weighted DMM images: (a) action ‘stand-to-sit’ from the transition movements dataset, (b) action ‘waving a hand’ from the smart TV gesture dataset

from the smart TV gesture dataset are shown in Figure 8.5. The weighted DMM images are resized to 50×50 and used as the input to the second CNN stream.

The overall architecture of the two CNN streams is shown in Figure 8.6. The first CNN stream comprises two 3D convolutional layers. The first layer convolves the raw depth images with 16 convolution filters of size 5×5×5. The output is passed through a 3D subsampling layer employing max pooling. The second convolutional layer convolves the output of the pooling layer with 8 filters of size 5×5×5 and passes it to another 3D subsampling layer. The second CNN stream convolves the input weighted DMM images with 16 2D convolution filters of size 5×5. The outputs of the convolution layer are subsampled and passed onto another 2D convolution layer comprising 5 filters of size 5×5, followed by another subsampling layer using max pooling. Rectified linear unit (ReLU) activation is used at each 2D and 3D convolution layer. The output of each CNN stream is flattened and concatenated before passing it to a dense layer, which maps it to a 512×1 vector based on the ReLU activation. The last layer is a fully connected layer using a sigmoid activation, which generates scores for the output classes.

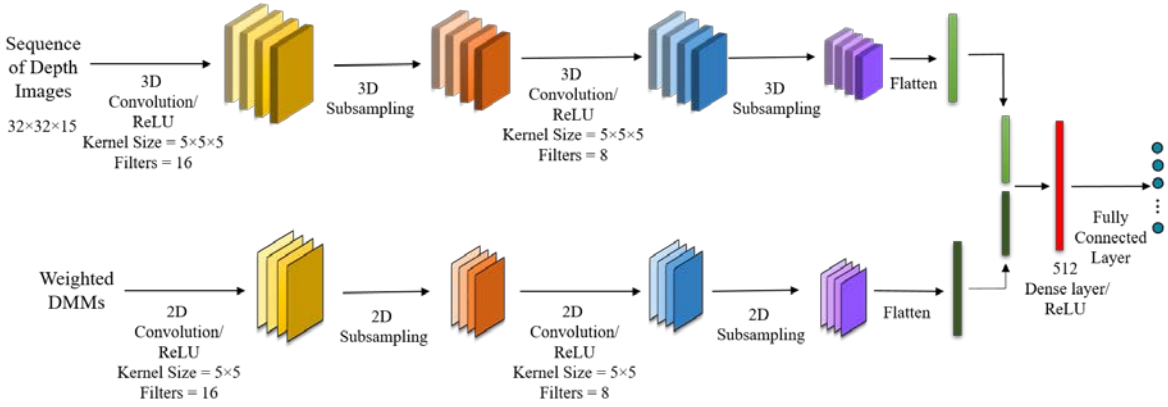

Sequence of Depth Images
32×32×15
3D Convolution/ ReLU Kernel Size = 5×5×5 Filters = 16
3D Subsampling
3D Convolution/ ReLU Kernel Size = 5×5×5 Filters = 8
3D Subsampling
Flatten
Weighted DMMs
2D Convolution/ ReLU Kernel Size = 5×5 Filters = 16
2D Subsampling
2D Convolution/ ReLU Kernel Size = 5×5 Filters = 8
2D Subsampling
Flatten
512 Dense layer/ ReLU
Fully Connected Layer

Figure 8.6. CNN architecture used for continuous action detection and recognition based on depth images

8.4.3 CNN+LSTM Architecture for Inertial Signals

An architecture similar to the above two-stream CNN architecture is used for inertial signals, except that the second stream directly uses the handcrafted features with no further feature extraction. The first stream uses CNN and LSTM layers and the second stream directly uses the handcrafted inertial features. The input to the first stream is 8 time-series signals corresponding to the 3-axis acceleration signals, the 3-axis angular velocity signals, the overall acceleration signal, and the overall angular velocity signal. The overall angular velocity is obtained by computing the angular velocity at each frame via Equation (8.3) and the overall acceleration is obtained using the acceleration A^t at each frame t of a sequence as follows:

$$A^t = \sqrt{{a_x^t}^2 + {a_y^t}^2 + {a_z^t}^2} \tag{8.6}$$

where a_x^t, a_y^t, a_z^t denote the 3D accelerations at frame t. These 8 time-series signals are sampled to obtain a total of 200 samples per sequence. These signals are normalized and sent to the CNN+LSTM stream as stacked inertial signal images of size 200×8.

The handcrafted features used in the second stream involve the statistical features of the inertial signals. The above 8 time-series signals are divided into 3 equal sized temporal segments and similar to [25], the statistical features of *mean*, *variance*, *standard deviation*, *root mean square*, *median*, *minimum*¸ and *maximum* of the segments of these signals, and *mean*, *variance*, *standard deviation*, and *root mean square* of the segments of their first derivatives are used as the handcrafted signals.

The overall architecture of the CNN+LSTM network used for the inertial signals is shown in Figure 8.7. The inertial signal images are first convolved with 16 1D filters of size 15 to obtain features from the time-series signals based on the ReLU activation. The output is subsampled using 1D max pooling and passed onto the LSTM layer with 64 cells. The output of the LSTM layer is then used along with the handcrafted features from the second stream to form a single vector. A dense layer maps this vector to a vector of size 512×1 based on the ReLU activation. The output of the dense layer is finally passed onto a fully connected layer that uses a sigmoid activation generating scores for the output classes.

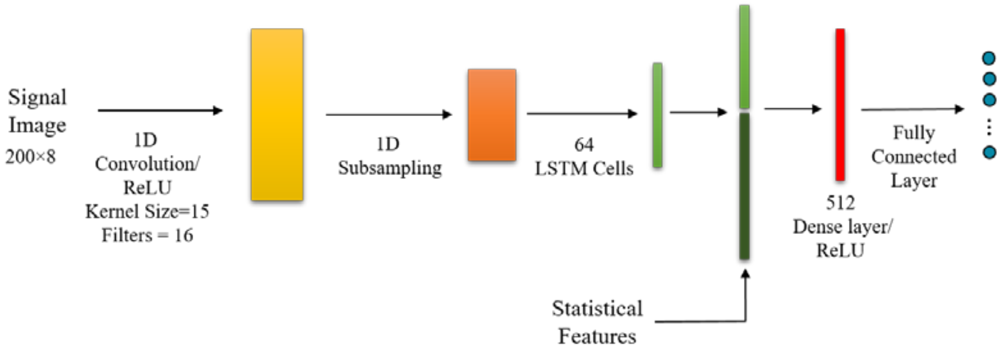


Figure 8.7. CNN+LSTM architecture used for continuous action detection and recognition based on inertial signals

For the continuous smart TV gesture dataset, another 1D convolution layer with 8 filters of size 9 and a subsampling layer is used before the LSTM layer to capture the entire dynamics of the hand gestures. Since the size of the time-series signals gets reduced further by adding another subsampling layer, only 16 LSTM cells are used here. The handcrafted features used for this dataset are the statistical features of *mean*, *variance*, *standard deviation*, *root mean square*, *median*, *minimum*¸ and *maximum* of the three equal sized segments of the 8 time-series signals.

Here, it is worth stating that different architectures and parameters were examined to reach the architectures utilized here by using a subset of the training data as the validation data. One to three convolution layers with different numbers and sizes of filters were considered. Different numbers of LSTM cells were also examined. The architectures reported above for depth images and inertial signals were found to be the most effective ones. Apart from examining different architectures for decision-level fusion, a feature-level or data fusion was also considered by passing the data from the two sensors to a common network. It was found that the decision-level fusion by far was more effective than the feature-level or data fusion.

8.4.4 Continuous Detection and Recognition

A technique similar to the one reported in [30] is adopted here to perform detection and recognition from continuously segmented actions. Given a segment S^k with S^k_{start} representing its starting point and S^k_{stop} its stopping point, an examination action set $A^k = \{A_k^0, A_k^1, \dots A_k^l, \dots, A_k^{K-1}\}$ is formed where A_k^l represents an action whose starting point is S^{k-l}_{start}. The stopping point of all the actions in A^k is S^k_{stop}. Hence, whenever a segment is obtained from a continuous action stream, it is examined along with *K-1* prior segments to detect the presence of an action of interest. Only the actions formed from these segments whose length lies within the range of actions of interest are examined further. Based on the number of segments that normally occur in the transitions movements actions of interest, $K = 5$ was found to work best for depth image segments and $K = 10$ was found to work best for inertial signal segments. Similarly, depth images with a length falling in the frame range of $\{8, 40\}$ and the ones segmented from inertial signals having a length falling in the sample range of $\{70, 750\}$ were found to work best. The experimentations reported in the next section are based on using these values.

The detection and recognition of the actions of interest are performed based on the output scores of the fully connected layer. Note that a softmax activation is not used here at the fully connected layers. The reason is that softmax activation would result in the output scores that add up to one. A sigmoid activation is used instead along with the mean squared error loss function. This ensures that all the classes are trained individually and this way the output scores at the fully connected layer do not need to add up to one. The main advantage of modifying the loss function and activation at the fully connected layer is that detection and recognition of actions of interest can be performed at the same time using the same network. This modification results in a low score throughout all the classes for most actions of non-interest. Also, an action with more than one high score class is indicative of the presence of actions of non-interest.

Based on the output scores at the fully connected layers in the two paths, an initial detection of actions of interest is performed. Based on the output scores of the depth image path, only the actions with scores >0.9 are labeled as potential actions of interest. Similarly, for the output scores in the inertial signal path, the actions which have the first scores >0.9 and the second scores <0.1 are labeled as potential actions of interest. Only the actions that qualify as potential actions of interest from the two paths are passed onto the next stage. The output scores from the two paths are then multiplied to obtain the fusion scores. The actions which have fusion scores >0.8 for exactly one class and fusion scores <0.1 for the rest of the classes are clearly indicative of actions of interest. Such actions are labeled as actions of interest and are classified or placed in the class with the highest score. Note that performing detection both before the fusion and during the fusion results in the rejection of the great majority of actions of non-interest.

8.4.5 Data Augmentation for Limited Datasets

Since training a CNN or LSTM network requires a very large amount of training data, a data augmentation step was performed in order to address the limited size of the dataset for training the CNN or LSTM networks. In case of depth images, the training samples were flipped, rotated and translated to generate multiple new training samples from a single sample. These operations were applied to both depth image sequences and weighted DMMs simultaneously to produce synchronized training samples. In case of inertial signals, white noise was added to random frames at the beginning or end or both at the beginning and end of the action streams. In addition to the data augmentation, a dropout ratio of 0.5 was used throughout the networks to control overfitting as discussed in [31].

8.5 EXPERIMENTAL RESULTS AND DISCUSSION

The effectiveness of the developed deep learning-based continuous action detection and recognition fusion system was examined on the two continuous datasets: continuous transition movements dataset and continuous smart TV hand gesture dataset. As mentioned earlier, the training of the CNN and CNN+LSTM networks was performed using the segmented datasets. Both the networks were trained individually from the two input layers to the fully connected output layer. The Adam optimizer was used to train the networks using the mean squared error loss function. For testing, the segmentation was carried out by using both of the sensing modalities and only the actions that qualified as potential actions of interest by both the modalities were passed onto the decision fusion stage to conduct the removal of false positives and to reach the final decision. The coding for both the training and testing of the developed continuous detection and recognition system was done in Python.

Here, it is worth mentioning that apart from the continuous datasets examined, all other existing datasets that provide simultaneous data from both a depth and an inertial sensor contain segmented or isolated actions, and thus it is not possible to test the performance of the detection and recognition system on these datasets. However, the developed fusion approach was compared with the existing fusion based recognition approaches in [32], [27], [13] by using the UTD-MHAD dataset [26]. The fusion of CNN for depth images and CNN+LSTM for inertial signals was performed by multiplying the scores of their fully connected layers and the assigned class label was considered to be the one with the highest score. The UTD-MHAD dataset is a multimodal dataset comprising 27 actions performed by 8 subjects. To provide a fair comparison with the approach in [27], the data from the odd numbered subjects were used for training, while the data from the even numbered subjects were used for testing. The results obtained using different approaches are reported in Table 8.1. As can be seen from this table, even with the limited size of the dataset, a higher accuracy was achieved with the developed deep learning-based fusion system. In order to see the effect of using two streams for recognition, the segmented data from the three datasets (UTD-MHAD, Continuous Transition Movements and Continuous Smart TV Gestures) were divided into a training and a validation set. The training sets were used to train the networks associated with single streams and the two networks associated with both streams. The recognition accuracies of the validation sets are reported in Table 8.2. As can be seen from this table, the use of both streams led to higher accuracies compared to single streams.

To examine the performance of the overall system on the two continuous datasets, the ground truth actions were manually identified from the continuous action streams by visual inspection. The performance evaluation was based on the widely used measures of precision, recall and $F1$ score [33]. First, the detected actions were marked as either true positives or false positives. The actions of interest detected within a window of five frames from the ground truth and correctly classified were marked as true positives. The actions with no overlap with the ground truth, or the ones misclassified were marked as false positives. The ground truth actions which were not detected by the system, or the ones detected but not correctly classified were marked as false negatives. Based on the number of true positives N_{TP}, the number of false positives N_{FP} and the number of false

Table 8.1. Recognition accuracy for UTD-MHAD dataset

Method	Accuracy (%)
ELC-KSVD [32]	76.2
Kinect and Inertial [27]	79.1
Skeleton Joints and Inertial [13]	86.3
Developed Deep Learning-based Fusion	92.8

Table 8.2. Recognition accuracies when using single streams versus both streams

Dataset	Using first stream only	Using second stream only	Using both streams
UTD-MHAD	87.4%	87.4%	92.8%
Continuous Transition Movements	98.1%	95.2%	99.3%
Continuous Smart TV Gestures	80.3%	75.7%	86.3%

negatives N_{FN} across all the continuous action streams, the measures of precision P, recall R and $F1$ scores were computed as follows [33]:

$$P = \frac{N_{TP}}{N_{TP} + N_{FP}} \tag{8.7}$$

$$R = \frac{N_{TP}}{N_{TP} + N_{FN}} \tag{8.8}$$

$$F1 = 2\frac{P \cdot R}{(P + R)} \tag{8.9}$$

The results obtained for these measures are discussed further in the subsections that follow.

8.5.1 Continuous Transition Movements Dataset

The continuous action streams in the continuous transition movements dataset were segmented using centroid differences of the depth images and angular velocity differences of the inertial signals. The testing was repeated for each subject by not using the subject in training. A threshold falling in the range [1, 2] that corresponded to 5% of the maximum value was applied to remove negligible centroid differences in the continuous action streams. A threshold of 0.5 that corresponded to 5% of the maximum value was applied to remove angular velocity differences in the continuous action streams.

The measures of precision, recall and F1 score obtained by the developed deep learning-based fusion system for all the subjects are reported in Table 8.3. This table also shows the overall or average precision, recall and F1 score obtained across all the subjects. This measure was also obtained by using a single modality of depth camera and inertial sensor and an improvement of more than 15% was achieved in F1 score when the fusion of the two modalities was used as compared to the cases where a single modality (either depth camera or inertial sensor) was used individually based on the same CNN or CNN+LSTM networks. This is due to the fact that the fusion system was able to reject most of the false positives that were detected by a single modality. An example showing the ground truth centroid difference signal and the detected actions is shown in Figure 8.8. The confusion matrix indicating the recognition performance of the fusion system is reported in Table 8.4 indicating an overall recognition accuracy of 98.8%. As indicated in this

Table 8.3. Precision, recall, and F1 score for the continuous transition movements dataset

	Precision	**Recall**	***F*1 Score**
Subject 1	91.3%	90.0%	90.3%
Subject 2	96.9%	88.6%	92.5%
Subject 3	94.2%	92.9%	93.5%
Subject 4	83.3%	100%	90.9%
Subject 5	86.1%	88.6%	87.3%
Average	90.4%	92.0%	90.9%

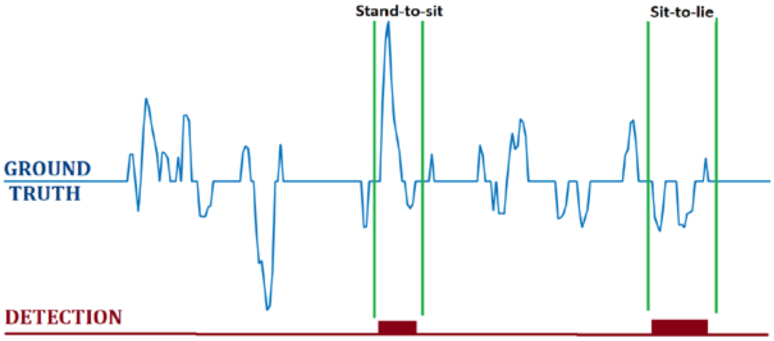


Figure 8.8. An example of detected actions of interest versus the ground truth for the centroid difference signal in a continuous action stream

Table 8.4. Confusion matrix for the continuous transition movements dataset (in %)

	St-S	**St-L**	**S-St**	**S-L**	**L-S**	**L-St**	**F**
St-S	100	-	-	-	-	-	-
St-L	-	96	-	4	-	-	-
S-St	-	-	100	-	-	-	-
S-L	-	4	-	96	-	-	-
L-S	-	-	-	-	100	-	-
L-St	-	-	-	-	-	100	-
F	-	-	-	-	-	-	100

St-S: stand-to-sit, St-L: stand-to-lie, S-St: sit-to-stand,
S-L: sit-to-lie, L-S: lie-to-sit, L-St: lie-to-stand, F: fall

table, most misclassifications occurred due to the fact that the action ‘lie-to-stand’ can be regarded as a combination of the actions ‘lie-to-sit’ and ‘sit-to-stand’ performed in series, and the action ‘stand-to-lie’ can be regarded as a combination of the actions ‘stand-to-sit’ and ‘sit-to-lie’ performed in series. As a result, these actions were sometimes misdetected.

8.5.2 Continuous Smart TV Gesture Dataset

As mentioned earlier, since the continuous smart TV gesture dataset consists of hand gestures, it was easier to segment these hand gestures using the skeleton joint positions and inertial signals via the technique described in [13]. Once the segmentation was done, the deep learning-based fusion system was used to identify the actions of interest in order to provide a comparison with the subject-specific results reported in [13]. The subject-specific scenario means the system is trained using the segmented data of the subject for whom testing is performed. The measures of precision, recall and $F1$ score were obtained and averaged for the 12 subjects in the dataset and compared to the results obtained by the SVDD and CRC-based continuous detection and recognition approach reported in [13]. Table 8.5 provides the comparison of the measures between the fusion approaches in [13] and the one developed here. It should be noted that the approach developed in [13] utilizes skeleton joint positions while the approach developed here utilizes depth images. Although skeleton joint positions are more informative, the use of depth images is more general purpose in terms of applicability to different action recognition applications since in practice skeleton joint positions appear overlapping in many action recognition applications. To allow proper tracking of the skeleton joints, the joints should be visible at all times with no overlap. In practice, however, overlapping occurs in many action recognition applications. The confusion matrix of the

Table 8.5. Precision, recall, and F1 score for the continuous smart TV gesture dataset

	Precision	**Recall**	***F*1 Score**
[13]	96.6%	95.7%	96.2%
Deep Learning Fusion	97.5%	96.5%	97.0%

Table 8.6. Confusion matrix for the continuous smart TV gesture dataset (in %)

	WH	**FL**	**FR**	**CCR**	**CR**
WH	100	-	-	-	-
FL	-	98.3	1.7	-	-
FR	-	1.7	96.6	1.7	-
CCR	-	1.7	-	93.3	5
CR	-	-	-	-	100

WH: waving a hand, FL: flip to left, FR: flip to right,
CCR: counterclockwise rotation, CR: clockwise rotation

recognition performance for the continuous smart TV dataset is provided in Table 8.6 indicating an overall recognition accuracy of 97.6%.

8.5.3 System Operation Processing Time

The times to process segments and obtain their handcrafted features were measured on a laptop computer running the fusion system with the depth camera connected to it via a USB port and the wearable inertial sensor connected to it via a Bluetooth link. This laptop was equipped with a 4.2GHz processor and 64GB RAM. It was found that the computation of the weighted DMMs from the actions obtained from the depth segments and the final scores of the CNN network took 94ms on average. Similarly, the formation of the handcrafted statistical features from the inertial segments and the final score computation using the CNN+LSTM networks took 3ms on average. As a result, the detection and recognition of actions of interest from the continuous action streams was made 100ms after the completion of an action. It is worth mentioning here that this time represents the algorithmic complexity of the system to perform continuous detection and recognition via a modern laptop without the need to use any additional dedicated processing hardware. Two video clips of the operation of the fusion system running in real-time on continuous action streams corresponding to the two applications considered can be viewed at these links: www.utdallas.edu/~kehtar/DeepLearningFusionSystem-TransitionMovements.avi and www.utdallas.edu/~kehtar/DeepLearningFusionSystem-SmartTV.avi .

8.6 CONCLUSION

In this chapter, a deep learning-based fusion system to detect and recognize actions of interest from continuous action streams has been developed. Continuous action streams reflect the way actions are performed in real-world situations, that is when actions of interest are performed continuously and randomly among arbitrary and unknown actions of non-interest. The system uses depth images from a depth camera and inertial signals from a wearable inertial sensor. Decision-level fusion is applied to the actions of interest that are detected by both of the modalities in order to reject actions of non-interest and classify the detected actions of interest. The developed fusion system has been examined for two applications: one involving transition movements for home healthcare monitoring and the other for smart TV hand gestures. The results obtained indicate the effectiveness of the developed fusion system in the detection and recognition of actions of interest in realistic continuous action streams.

8.7 REFERENCES

[1] A. Avci, S. Bosch, M. Marin-Perianu, R. Marin-Perianu, and P. Havinga, “Activity recognition using inertial sensing for healthcare, wellbeing and sports applications: A survey,” *Proceedings of 23rd International Conference on Architecture of Computing Systems (ARCS)*, pp. 1–10, 2010.

[2] N. Zerrouki, F. Harrou, Y. Sun, and A. Houacine, “Vision-based Human Action Classification Using Adaptive Boosting Algorithm,” *IEEE Sensors Journal*, vol. 18, no. 12, pp.5115–5121, May 2018.

[3] W. Lao, J. Han, and P. De With, "Automatic video-based human motion analyzer for consumer surveillance system," *IEEE Transactions on Consumer Electronics*, vol. 55, no. 2, pp. 591–598, May 2009.

[4] J. Wang, Z. Liu, Y. Wu and J. Yuan, "Mining actionlet ensemble for action recognition with depth cameras," *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1290–1297, June 2012.

[5] B. Ghojogh, H. Mohammadzade, and M. Mokari, “Fisherposes for human action recognition using Kinect sensor data,” *IEEE Sensors Journal*, vol. 18, no. 4, pp.1612–1627, February 2018.

[6] R. Xu, S. Zhou, and W. Li, “MEMS accelerometer based nonspecific-user hand gesture recognition,” *IEEE Sensors Journal*, vol. 12, no. 5, pp.1166–1173, May 2012.

[7] A. Wang, G. Chen, J. Yang, S. Zhao, and C. Chang, “A comparative study on human activity recognition using inertial sensors in a smartphone,” *IEEE Sensors Journal*, vol. 16, no. 11, pp.4566–4578, June 2016.

[8] C. Chen, R. Jafari, and N. Kehtarnavaz, "Improving human action recognition using fusion of depth camera and inertial sensors," *IEEE Transactions on Human-Machine Systems*, vol. 45, no. 1, pp. 51–61, February 2015.

[9] C. Chen, R. Jafari, and N. Kehtarnavaz, “A real-time human action recognition system using depth and inertial sensor fusion,” *IEEE Sensors Journal*, vol. 16, no. 3, pp. 773–781, February 2016.

[10] C. Chen, R. Jafari, and N. Kehtarnavaz, "Fusion of depth, skeleton, and inertial data for human action recognition," *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 2712–2716, March 2016.

[11] N. Dawar, C. Chen, R. Jafari, and N. Kehtarnavaz, “Real-time continuous action detection and recognition using depth images and inertial signals,” *Proceedings of 26th IEEE International Symposium on Industrial Electronics (ISIE)*, pp. 1342–1347, June 2017.

[12] N. Dawar and N. Kehtarnavaz, “Continuous detection and recognition of actions of interest among actions of non-interest using a depth camera,” *Proceedings of IEEE International Conference of Image Processing (ICIP),* pp. 4227–4231, September 2017.

[13] N. Dawar and N. Kehtarnavaz, “Real-time continuous detection and recognition of subject-specific smart TV gestures via fusion of depth and inertial sensing,” *IEEE Access*, vol. 6, pp. 7019–7028, January 2018.

[14] N. Dawar and N. Kehtarnavaz, “Data flow synchronization of a real-time fusion system to detect and recognize smart TV gestures”, *Proceedings of IEEE International Conference on Consumer Electronics (ICCE),* pp. 1–4, January 2018.

[15] K. Liu, C. Chen, R. Jafari, and N. Kehtarnavaz, “Fusion of inertial and depth sensor data for robust hand gesture recognition,” *IEEE Sensors Journal*, vol. 14, no. 6, pp. 1898–1903, June 2014.

[16] S. Ji, W. Xu, M. Yang, and K. Yu, “3D convolutional neural networks for human action recognition,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 1, pp. 221–231, 2013.

[17] J. Donahue, L. Anne Hendricks, S. Guadarrama, M. Rohrbach, S. Venugopalan, K. Saenko, and T. Darrell, “Long-term recurrent convolutional networks for visual recognition and description,” *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2625–2634, 2015.

[18] J. Chen, K. Kwong, D. Chang, J. Luk, and R. Bajcsy, “Wearable sensors for reliable fall detection,” *Proceedings of 27th Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, pp. 3551–3554, January 2006.

[19] C. Chen, R. Jafari, and N. Kehtarnavaz. "A survey of depth and inertial sensor fusion for human action recognition," *Multimedia Tools and Applications*, pp. 1–21, December 2017.

[20] P. Wang, W. Li, Z. Gao, J. Zhang, C. Tang and P. Ogunbona, "Action recognition from depth maps using deep convolutional neural networks," *IEEE Transactions on Human-Machine Systems*, vol. 46, no. 4, pp. 498–509, August 2016.

[21] Z. Liu, C. Zhang, and Y. Tian, “3D-based deep convolutional neural network for action recognition with depth sequences,” *Image and Vision Computing*, vol. 55, pp. 93–100, November 2016.

[22] C. Li, P. Wang, S. Wang, Y. Hou, and W. Li, "Skeleton-based action recognition using LSTM and CNN," *Proceedings of IEEE International Conference on Multimedia & Expo Workshops (ICMEW),* pp. 585–590, July 2017.

[23] J. Yang, M. Nguyen, P. San, X. Li, and S. Krishnaswamy, “Deep convolutional neural networks on multichannel time series for human activity recognition,” *Proceedings of 24th International Conference on Artificial Intelligence (IJCAI)*, pp. 3995–4001, July 2015.

[24] F. Ordóñez, and D. Roggen, “Deep convolutional and LSTM recurrent neural networks for multimodal wearable activity recognition,” *Sensors*, vol. 16, no. 1, pp. 115, January 2016.

[25] D. Ravi, C. Wong, B. Lo, and G. Yang, “A deep learning approach to on-node sensor data analytics for mobile or wearable devices,” *IEEE Journal of Biomedical and Health Informatics*, vol. 21, no. 1, pp. 56–64, January 2017.

[26] A. Yang, R. Jafari, S. Sastry, and R. Bajcsy, “Distributed recognition of human actions using wearable motion sensor networks,” *Journal of Ambient Intelligence and Smart Environments*, vol. 1, no. 2, pp. 103–115, 2009.

[27] C. Chen, R. Jafari, and N. Kehtarnavaz, “UTD-MHAD: A multimodal dataset for human action recognition utilizing a depth camera and a wearable inertial sensor,” *Proceedings of the IEEE International Conference on Image Processing*, pp. 168–172, September 2015.

[28] N. Dawar and N. Kehtarnavaz, “A convolutional neural network-based sensor fusion system for monitoring transition movements in healthcare applications”, *Proceedings of IEEE International Conference on Control and Automation (ICCA)*, pp. 482-485, June 2018.

[29] C. Chen, M. Liu, H. Liu, B. Zhang, J. Han, and N. Kehtarnavaz, “Multi-temporal depth motion maps-based local binary patterns for 3-D human action recognition,” *IEEE Access*, vol. 5, pp. 22590-22604, 2017.

[30] G. Zhu, L. Zhang, P. Shen, and J. Song, "An Online Continuous Human Action Recognition Algorithm Based on the Kinect Sensor," *IEEE Sensors Journal*, vol. 16, no. 2, pp. 161, January 2016.

[31] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, “Dropout: A simple way to prevent neural networks from overfitting,” *Journal of Machine Learning Research*, vol. 15, no. 1, pp. 1929-1958, 2014.

[32] L. Zhou, W. Li, Y. Zhang, P. Ogunbona, D. Nguyen, and H. Zhang, “Discriminative key pose extraction using extended lc-ksvd for action recognition,” *Proceedings of International Conference on Digital lmage Computing: Techniques and Applications (DlCTA)*, pp. 1-8, November 2014.

[33] C. Goutte and E. Gaussier, "A probabilistic interpretation of precision, recall and F-score, with implication for evaluation," *Proceedings of the European Conference on Information Retrieval*, Springer Berlin Heidelberg, pp. 345–359, March 2005.

CHAPTER 9

CONCLUSION AND FUTURE WORK

The thrust of this dissertation has been to detect and recognize actions of interest in continuous action streams by using a depth camera and a wearable inertial sensor within a decision fusion framework. More specifically, the major contributions of this dissertation include:

1- Compared to the great majority of existing action detection and recognition approaches where actions of interest are assumed segmented, a more realistic scenario is addressed in this dissertation where actions of interest take place continuously and randomly amongst arbitrary unknown actions of non-interest.

2- In the great majority of existing works on action detection and recognition, only a single modality sensor is used. To achieve a more robust action detection and recognition when operating under realistic conditions, two differing modality sensors of a depth camera and a wearable inertial sensor are used by fusing the decisions that are made based on each of these sensors.

3- Two continuous datasets containing synchronized data from a depth camera and an inertial sensor are collected and made available for public use.

4- After developing and examining a number of fusion approaches, it is found that the deep learning-based approach discussed in Chapter 8 provides the most effective solution compared to the other solutions discussed in the other chapters.

As possible future work, this research work can be extended in the following ways:

1- In addition to or in place of a depth camera, the fusion framework can be extended to video cameras. There is a wealth of detection and recognition techniques available in the literature for video images that can be added to the developed fusion framework to enhance the robustness of continuous action detection and recognition.

2- Adding a missing data capability to the framework to take into consideration situations when the data get missing from one of the sensors, for example when a subject moves out of the field of view of the camera, the decision made by that sensor is disabled and only the decision made by the other sensor is used.

3- Applying the developed fusion framework to other applications and tuning various components of the framework to those applications towards enhancing the robustness of detection and recognition under realistic operating conditions.

BIOGRAPHICAL SKETCH

Neha Dawar received her Bachelor of Technology degree in Communication and Computer Engineering from the LNM Institute of Information Technology, Jaipur, India in 2011, and her MS degree in Electrical Engineering from the University of Calgary, Canada in 2014. She is currently a PhD candidate in the Department of Electrical and Computer Engineering at The University of Texas at Dallas and is expected to graduate in December 2018. Her research interests include signal and image processing, computer vision and machine learning. Her research contributions have appeared in five journal papers and six conference papers. Based on her academic record, she was awarded the Jonsson School Graduate Study Scholarship in 2015 and the Jan P. Van der Ziel Graduate Fellowship in 2017 and 2018 by the Erik Jonsson School of Engineering and Computer Science at The University of Texas at Dallas.

CURRICULUM VITAE

Neha Dawar

Address: University of Texas at Dallas, Department of Electrical and Computer Engineering

Email: neha.dawar@utdallas.edu

EDUCATION

✓ **PhD Electrical Engineering (Signal and Image Processing)** Aug 2015—Dec 2018 (expected)
University of Texas at Dallas**,** Richardson, TX
Dissertation advisor: Prof. Nasser Kehtarnavaz
Dissertation title: Action Recognition in Continuous Data Streams Using Fusion of Depth and Inertial Sensing

✓ **MS Electrical Engineering (Computer Vision)** Sep 2012—Sep 2014
University of Calgary, Alberta, Canada
Thesis advisor: Dr. John Nielsen and Dr. Gérard Lachapelle
Thesis title: Computer Vision based Indoor Navigation Utilizing Information from Planar Surfaces

✓ **BS Electrical Engineering** July 2007—May 2011
LNM Institute of Information Technology, Jaipur, India

PROFESSIONAL EXPERIENCE

✓ Student Member, IEEE, 2013—present

✓ Member, IEEE Young Professionals, 2013—present

✓ Member, IEEE-Eta Kappa Nu, 2015—present

✓ Assistant Editor, IEEE IMPACT, 2015

TEACHING EXPERIENCE

✓ **Signals and Systems Laboratory**	**Fall 2015-Spring 2018**
✓ **Digital Image Processing**	**Fall 2017**
✓ **Signal and Systems**	**Fall 2015**
✓ **Computer Vision**	**Winter 2014**

RESEARCH INTERESTS

✓ Machine Learning
✓ Signal Processing
✓ Image Processing
✓ Computer Vision

PUBLICATIONS

Journal Papers (accepted/published)

1. **N. Dawar**, and N. Kehtarnavaz, “Data Augmentation in Deep Learning-Based Fusion of Depth and Inertial Sensing for Action Recognition,” *IEEE Sensors Letters*, 2018.
2. **N. Dawar**, and N. Kehtarnavaz, “Action Detection and Recognition in Continuous Action Streams by Deep Learning-Based Sensing Fusion,” *IEEE Sensors Journal*, vol. 18, no. 23, pp. 9660–9668, December 2018.
3. **N. Dawar**, and N. Kehtarnavaz, “Real-Time Continuous Detection and Recognition of Subject-Specific Smart TV Gestures via Fusion of Depth and Inertial Sensing”, *IEEE Access*, vol. 6, pp. 7019–7028, January 2018.
4. **N. Dawar**, T. Sharma, R. Darraji, and F. Ghannouchi, “Linearization of Radio Frequency Power Amplifiers Exhibiting Memory Effects using Direct Learning-based Adaptive Digital Predistortion,” *IET Communications*, vol. 10, no. 8, pp. 950-954, May 2016.
5. T. Sharma, R. Darraji, F. Ghannouchi, and **N. Dawar**, “Generalized Continuous Class-F Harmonic Tuned Power Amplifiers,” *IEEE Microwave and Wireless Components Letters*, vol. 26, no. 3, pp. 213-215, March 2016.

Conference Papers

1. **N. Dawar**, and N. Kehtarnavaz, “A Convolutional Neural Network-Based Sensor Fusion System for Monitoring Transition Movements in Healthcare Applications”, *Proceedings of IEEE International Conference on Control & Automation (ICCA)*, pp. 482-485, Anchorage, Alaska, June 2018.
2. **N. Dawar**, and N. Kehtarnavaz, “Data Flow Synchronization of a Real-Time Fusion System to Detect and Recognize Smart TV Gestures”, *Proceedings of IEEE International Conference on Consumer Electronics (ICCE)*, pp. 1-4, Las Vegas, January 2018.
3. **N. Dawar**, and N. Kehtarnavaz, “Continuous Detection and Recognition of Actions of Interest among Actions of Non-interest using a Depth Camera”, *Proceedings of IEEE International Conference of Image Processing (ICIP)*, pp. 4227-4231, Beijing, China, September 2017.
4. **N. Dawar**, C. Chen, R. Jafari, and N. Kehtarnavaz, “Real-Time Continuous Action Detection and Recognition Using Depth Images and Inertial Signals”, *Proceedings of IEEE 26th International Symposium on Industrial Electronics (ISIE)*, pp. 1342-1347, Edinburgh, June 2017.
5. **N. Dawar**, and J. Nielsen, “Indoor Navigation based on Computer Vision utilizing Information from Patterned Surfaces”, *Proceedings of the 27th International Technical Meeting of the Satellite Division of the Institute of Navigation (ION GNSS+ 2014)*, pp. 1652-1660, Tampa, Florida, September 2014.
6. J. Nielsen, V. Dehghanian, and **N. Dawar**, "GNSS Spoofing Detection Based on Particle Filtering," *Proceedings of the 26th International Technical Meeting of The Satellite Division of the Institute of Navigation (ION GNSS+ 2013)*, pp. 2997-3005, Nashville, TN, September 2013.

SELECTED HONORS AND AWARDS

✓	Best Teaching Assistant Award, The University of Texas at Dallas	2018
✓	Jan P. Van der Ziel Fellowship, The University of Texas at Dallas	2017-2018
✓	Jonsson School Graduate Study Scholarship, The University of Texas at Dallas	2015
✓	Outstanding Volunteer Award, IEEE Southern Alberta Section, Canada	2015
✓	Award of Excellence in Research, Faculty of Graduate Studies, University of Calgary, Canada	2013
✓	Graduate Student Research Scholarship, University of Calgary , Canada	Sep 2012-Aug 2014
✓	Director’s Gold Medal for excellence in academics, LNM Institute of Information Technology, India	2011