Edith Cowan University

Copyright Warning

The Detection of Abnormal Events in Activities of Daily Living using a Kinect Sensor

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

Bachelor of Computer Science Honours

By: Laurence Da Luz

Faculty of Computing, Health and Science

Edith Cowan University

Perth, Western Australia

Supervisor(s): Dr Martin Masek, A/Prof Peng Lam

Date of submission: 29 October 2012

Table of Contents

Abstract 4
1. Introduction 5
1.1 Background to the study 5
1.2 The Purpose of the Study 7
1.3 The Significance of the Study 7
1.4 Contributions 8
1.5 Structure of Thesis 9
1.6 Summary 10
2. Review of the Literature 11
2.1 Review of Existing Techniques 11
2.1.1 Data Acquisition / Sensors 11
2.1.2 Activity Recognition Techniques and Methods 13
2.2 Review of Related Concepts 18
2.2.1 Distance Measures 18
2.2.2 Busy/Idle 19
2.2.3 Weighting System for ADL Detection 20
2.3 Summary 22
3. Research Approach 24
3.1 Collecting Training Data 24
3.2 Pre-Processing 26
3.3 Feature Extraction 27
3.4 ADL Signature Generation 30
3.5 Classification of ADLs 32
3.6 Abnormality Detection 34
3.7 Presentation of ADLs 36
3.8 Summary 38
4. Experiments and Results 39
4.1 Data 40
4.2 Leave one out Testing Approach 45

4.3 Experiments 46
4.3.1 ADL Classification using Individual Video Data 47
4.3.2 ADL Classification using Combined Video Data 48
4.3.3 ADL Classification using ADL Variation Video Data 51
4.3.4 Abnormality Detection 52
4.3.5 Feature Comparison 55
4.3.6 Distance Measure Comparison 57
4.4 Summary 59
5. Conclusion 60
References 63
Appendix A: ADL Recordings 66
Appendix B: Results for Classifying Individual Video Data 74
Appendix C: Results for Classifying Combined Video Data 81
Appendix D: Results for Classifying ADL Variation Video Data 84
Appendix E: Results for Abnormality Detection 85
Appendix F: Results for Feature Comparison 93
Appendix G: Results for Distance Measure Comparison 99

Abstract

The growing elderly population presents a challenge on the resources of carers and assisted living communities. This has led to various projects in remote automated home monitoring in order to keep the elderly in their own home environment longer. These have the promise of alleviating the strain on support services, and the benefits of keeping people in their existing familiar community environment. Such monitoring typically involves a myriad of sensors attached to the environment and person, so as to acquire rich enough data to determine the actions of the person being monitored.

In this research, an algorithm based around the Microsoft Kinect, its 3D camera and person detection features, is presented for monitoring activities of daily living. The Kinect was originally released as a device to improve human interaction in gaming for the Xbox 360 game console. In this approach, training Data is obtained and then pre-processed using Kinect's in build person recognition. Collection of training data is necessary is because the process in which Activities of Daily Living (ADLs) are completed varies from person to person, and therefore no generic template to recognise ADLs exists. The research attempts to infer representations of ADLs through features related to the spatial position of the person in the field of view of the Kinect camera, which is divided into a grid of several data points. Once this representation or "ADL signature" is obtained, ADLs are classified live in unknown data using a combination of distance measures, and abnormal events are detected amongst these ADLs using an automatically generated threshold value. This system has the potential to replace the various sensors in the camera's field of view, and provide a system that accurately analyses the behaviours of the elderly to provide doctors and carers with valuable observational data.

1. Introduction

1.1 Background to the study

Many countries are facing an increasing elderly population. In Australia, a combination of longer life expectancy and decreased birth rate is expected to see the percentage of population aged 70 or older increase from 9% in 2007 to 13% by 2021 and 20% by 2051 (Commonwealth of Australia, 2008). As people age their function decreases, with 67.5% of Australians aged 75 and over affected by some kind of disability (Connell, Grealy, Olver & Power, 2008), putting increasing pressure on care and support services. As a result, the concept of "aging in place" has become an important problem to solve.

Aging in place allows the elderly to remain and interact with the community that they are familiar with and gives them a continued sense of independence, while also easing the burden that is seen on the limited capacity home care institutions. To solve this problem, and allow the elderly to live a normal life within the community, the need arises for more intelligent infrastructures within the home. Specifically, there is a need for systems that can monitor the day to day tasks of the elderly and also recognise when something has gone wrong.

In monitoring a person's behaviour, it can be useful to first establish normal patterns of behaviour. In a home setting, this behaviour can be decomposed into a number of essential activities, known as Activities of Daily Living (ADLs). ADLs refer to self-care activities that are necessary for an individual's everyday living; including but not limited to the activities associated with grooming, feeding and sleeping. This research is focused on the monitoring and recognition of ADLs.

Researchers are currently using several different types of sensors and methods in an attempt to detect and recognise specific activities within a home. Some of the sensors that have been used include video streams (Ermis, Saligrama, Jodoin, & Konrad, 2008), wearable sensors (Stikic, Huynh, Van Laerhoven, & Schiele, 2008), as well as data received from a residential power line (Noury, Berenguer, Teyssier, Bouzid, & Giordani, 2011). Some of the problems faced when using these devices for abnormality detection include limited fidelity, dependence on lighting conditions and intrusiveness to daily living.

As the need for more sensor-based applications has increased, so has the number of different types of sensors that are available for public use. One of these is the Microsoft Kinect, originally released as a way to improve human interaction in gaming for the Xbox 360 game console. The Kinect is a motion sensing device that contains both a colour image camera and a depth camera. The depth sensor on the Kinect means uses its own (infrared) light source, meaning that issues due to variable and lighting conditions are minimised. The Kinect has been designed specifically for indoor use and as of January 2012, a Kinect sensor can be found in over 18 million homes (Takahashi, 2012). With the Kinect already in such a large number of homes, adoption of a system that utilises it will be more plausible than the task of implementing other sensor based systems.

The eventual goal of this project is to support the elderly who are living in their own home, and aims to achieve this by focusing on the monitoring and detection of abnormalities within ADLs. By monitoring ADLs, the aim is to be able to provide a system that accurately analyses the behaviours of the elderly and can provide doctors and carers with valuable observation data, as well as an alert system that

can detect behaviour that is out of place such as behaviour arising from a person suffering from a heart attack or stroke.

1.2 The Purpose of the Study

This project is focused on the monitoring and detection of abnormalities within activities of daily living (ADLs), specifically the aims of this study are:

- To investigate and develop techniques for the detection of abnormal events in Activities of Daily Living within an indoor environment using data collected via the Kinect.
- To obtain and generate from the Kinect, data that uniquely represents different Activities of Daily Living.
- To use the acquired data to detect and classify abnormal events.
- To evaluate the effectiveness of the developed approach.

1.3 The Significance of the Study

This section discusses the significance and overall importance of the research. The significance of this research, as well as the core benefits, are represented in the following categories:

- Expanding current knowledge
- Novel application of Kinect sensors
- Community benefits

This research contributes to expanding current knowledge in abnormality detection within Activities of Daily Living. Existing techniques and methods are analysed and evaluated, and techniques

are developed using a Kinect sensor as a means of obtaining data, classifying ADLs and detecting abnormalities.

The use of a Kinect is significant because there is currently limited knowledge in applying the Kinect to the problem of abnormality detection and activity recognition. Current research in this area deals with using sensors such as a regular video stream, impulses from the power line, and wearable sensors. However, many of these types of sensors can be intrusive or have high setup costs. With the Kinect already in over 18 million homes, any beneficial software developed using this sensor will be highly accessible to the community.

This research may prove to be beneficial for the elderly community, as a cheap and effective way to increase safety through automated monitoring. Currently, the majority of home care monitoring is performed through home visits. However, this method is costly and can only be done in set intervals. This research may lead to a decrease in the need for these home visits, which in turn would mean less costly monitoring and a higher feeling of independence for the elderly.

1.4 Contributions

The product of this research is a system that can: learn based on ADL example videos, provide carers with a continuous profile of activity, and includes the automated detection of abnormal activities. The contributions that are presented in this work include:

- A novel representation of ADLs using features based on the depth and shape of a person. These features include the average depth, variance of this depth, busy fraction, aspect ratio and form factor.
- A model of ADLs, used for systematic training and evaluation.

- The investigation of features in which simple features, in terms of computational complexity, are used to provide an efficient and robust method of generating a person's ADL signature.
- The investigation of distance measures in feature space, with the implementation of a voting system to employ the use of multiple distance measures.
- A continuous monitoring system based on chronograms/spatio-temporal graphs to provide carers with valuable observational data.
- Abnormality detection using an automated threshold.

1.5 Structure of Thesis

Chapter One – Introduction presents the background and purpose of the study, states the significance and contributions, and provides the structure of thesis.

The rest of this thesis is organised as follows:

Chapter Two – Review of the Literature outlines the core ideas based around the research by providing a review of both existing techniques and related concepts.

Chapter Three – Research Approach outlines the techniques developed in this study, and details the procedures that were used in each module.

Chapter Four – Experiments and Results presents an evaluation on the developed approach.

Chapter Five – Conclusion provides final discussion and related future work.

1.6 Summary

To allow the elderly to live a normal life within the community, the need arises for more intelligent infrastructures within the home. This project is focused on the monitoring and detection of abnormalities within activities of daily living (ADLs) and this research investigates and develops techniques for the detection of abnormal events in ADLs using data collected via the Kinect. It contributes a system that can learn based on ADL example videos, automatically detect abnormalities, and provide carers with a continuous profile of activity. The system is beneficial for the elderly community, as a cheap and effective way to increase safety through automated monitoring.

2. Review of the Literature

This literature review will outline the core ideas of abnormality detection and activity recognition. It has been structured to allow existing techniques to be analysed, providing a capture of the current state of play. This is followed by a technical review on some of the key components involved that are relevant to this research.

2.1 Review of Existing Techniques

This chapter presents the current state of the art within the field of activity recognition and abnormality detection. It is split into two main sub-sections, the first describing the use of different sensors for data acquisition, followed by a look into activity recognition techniques and methods.

2.1.1 Data Acquisition / Sensors

Several types of sensors have been used in the field of activity recognition and abnormality detection. Some of the sensors that have been used include video streams, wearable sensors, as well as data received from a residential power line. This section reviews the use of these sensors, and then looks at value of using the Kinect sensor for this research.

Researchers have used wearable sensors in an attempt to obtain accurate data from people being monitored. An example of this type of sensor involves an RFID chip embedded into a wearable bracelet (Stikic, et al., 2008). This system involves adding RFID tags to many common household objects or areas and works by recording any time the bracelet comes into contact with these objects. This means the sensor can detect when a person comes into contact with objects such as a light switch or broom etc, from which activity recognition can be determined. These types of wearable sensors have provided

researchers with promising results, however the problem with using these attachable sensors is that people often find them to be a burden or can forget to wear them at all necessary times. This problem is especially significant when dealing with the elderly, specifically with those who may be suffering from dementia or any other disorder which may reduce full memory function.

To deal with this problem, and to minimise the level of human interaction needed to monitor ADLs, a sensor that monitors the electrical impulses of a residential power line has been implemented by some researchers (Berenguer, Giordani, Giraud-By, & Noury, 2008; Noury, et al., 2011). In this system, the sensor detects electrical impulses when an electrical appliance is turned on or off. These impulses are unique to each electrical appliance which allows the researchers to track when the various appliances or lights are in use. This data is then used to determine the actions of a person, as well as the ADLs that are occurring. Drawbacks of this are that only the usage of electrical appliances and fixtures can be monitored. There are many activities where the only electrical component is to turn on a light in a particular room, and if enough natural light is present that might also not occur.

A common set of sensors that are being used in the field of activity recognition and abnormality detection are standard cameras (Debard et al., 2011; Ermis, et al., 2008). Cameras are cheap and easily accessible within a household, and when used to monitor a person over a long period of time they do not impose a burden on the subject that is seen with the wearable sensors that were previously discussed. Using a standard RGB video requires the use of processing techniques, such as background subtraction (Elgammal, Duraiswami, Harwood, & Davis,

2002; Zhang & Lu, 2001), to separate the people in the foreground of the image from the background of the image.

Smart homes that embed an array of sensors into the construction of the home have been created to obtain the location of a person as they complete various activities. The Georgia Institute of Technology *Aware Home* Project (Rowan, et al., 2001) is equipped with multiple in-floor pressure sensors that are triggered when a person walks over them, and the Massachusetts Institute of Technology *House_n* project (Tapia & Larson, 2004) contains several "state-change" sensors that indicate change in appliances, lights, and doors. Drawbacks of embedded sensors are that they cannot be easily added to an existing home, and may involve high installation costs.

This research will be using the Kinect as a means of data acquisition; this means the data received from this device may lead to more accurate results in a range of conditions. As well as the ability to obtain standard RGB images like that of a video camera, it is the Kinect's infrared sensor that separates it from many other types of sensors. The infrared sensor allows the Kinect to obtain the distance between itself and any object within its range. Unlike embedded sensors, the Kinect can be easily added to an existing environment without installation cost.

2.1.2 Activity Recognition Techniques and Methods

Different types of techniques and methods have been developed to assist automated activity recognition, and in turn abnormality detection. A recent survey of these techniques gives an overview that presents them as a tree-structured taxonomy (Aggarwal & Ryoo, 2011).

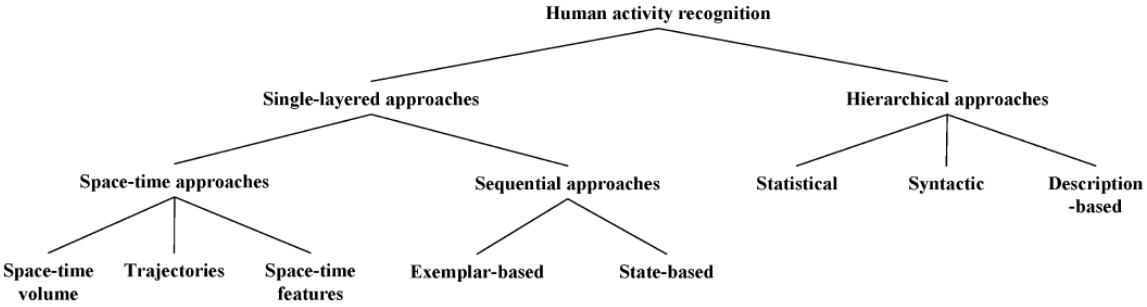


Figure 1: Tree-structured taxonomy as presented by Aggarwal & Ryoo (2011).

This classification splits the techniques and methods into **Single-layered approaches** and **Hierarchical approaches.** A Single-layered approach is one that recognises a sequence of actions directly from a video, and matches them to a suitable class. Due to its focus on sequences, this approach is better suited for short activities such as human gestures, which can implement a predetermined training sequence. A Hierarchical approach differs in that it uses the recognition of several smaller actions to recognise a larger or more complex activity and represent it a higher level. For example, "Cooking" would be considered a relevant complex ADL, in which it could be recognised through the smaller actions such as "turning on the stove" and "opening the fridge".

2.1.2.1 Single-Layered approaches

The single-layered approach is further categorized into **space-time approaches** and **sequential approaches**.

Space-Time Approaches

Space-time approaches are those which model video data as a 3-D volume, using time as the added dimension. This is done using the 2-D

frames from a video, and linking them along a time axis in order to obtain the 3-D Volume. In its simplest form, known as *space time volume,* the 3-D volume that is formed is a representation of the activity taking place during the selected period. This often only includes the foreground of each frame, to separate the area of interest from the rest of the image. Other forms of space-time approaches include *space-time trajectories, and space-time features.* Trajectory-based approaches are those which track the path of a person's joints, e.g. (Rao & Shah, 2001), which means as a person moves throughout a video, the position of their joints are recorded into a 3-D representation. The current issue with a trajectory-based approach is that it requires the 3-D modelling of body parts, which itself is still a current problem undergoing research. Feature-based approaches deal with 3-D volume, like that obtained in the space time volume approach, but involve the extraction of specific features from this volume which can be used to identify certain activities (Wong, Kim, & Cipolla, 2007). Feature based approaches can be applied to full frames and do not require the foreground of the frame to be identified first. However, this approach is limited to simple activities and does not perform well when used for complex tasks (Aggarwal & Ryoo, 2011).

When dealing with the task of activity recognition, the volumes, trajectories or features are analysed and directly compared using one of the following methods: *Template matching, neighbour-based (discriminative), and statistical modelling*. Template matching is the process of first creating a model that represents an activity (this model is based on training data and can incorporate volumes, trajectories or features) and then matching this model against one that is created from unknown or test data. Neighbour-based matching deals with first obtaining a set of sample volumes or trajectories, and then using this

set to estimate a match with part or all of an unknown video. This differs to Statistical Modelling, which involves the use of probability distribution function to match videos. These methods have been used by researchers in conjunction with various space-time approaches.

Rao & Shah (2001) employed a trajectory based approach which used template matching for activity recognition. Their work was based on tracking the position of hand movements, which was used to create the trajectories. Subsequently, Wong *et al.* (2007) utilised a space-time feature approach in which statistical modelling was used to detect a series of human actions, facial expressions and hand gestures. A space-time feature approach was also used by Laptev *et al*. (2008). However, in their approach a neighbour-based matching algorithm was used and the focus of their work was to address recognition in realistic scenes used in a variety of movies.

Sequential Approaches

Sequential approaches differ to space-time approaches in that they represent human activity as a sequence of features, and recognise these activities by searching for this sequence. Sequential approaches are further categorised into exemplar-based and state model-based. In Exemplar-based approaches, sequences are matched directly to the training data through the use of a direct template sequence. Whereas, State model-based deals with creating a model based on the training data that can be used to determine the likelihood of an activity occurring, but does not involve directly matching training sequences.

Vaswani *et al*. (2003) used an exemplar-based approach to detect abnormalities in surveillance footage of people leaving an airplane. In their approach they represented the group of people as a whole and measured the change in the shape of the group, not the individual

people. Hao *et al*. (2006) also use an exemplar-based approach in which they create a system that can locate specific actions within video sequences. Natarajan & Nevatia (2007) present a state model-based approach that uses coupled hidden Semi-Markov models in which they develop an efficient activity recognition algorithm.

2.1.2.2 Hierarchical approaches

The hierarchical approach is further categorised into **statistical**, **syntactic** and **description based** approaches.

Statistical Approaches

Statistical approaches use multi-layered graphs to categorise sequential activities. The first layer is usually similar to the single-layered sequential approach in that actions are recognised from a sequence of features. From here the next layer then generates the probability of how likely it is that the activity matches the unknown input. The activity is then recognised using an estimation based on which sequence has the highest probability. An example of a hierarchical statistical approach can be seen in the work of Nguyen *et al*. (2005), which involves a system for human actions that uses Hidden Markov Model to detect activities. Dai *et al*. (2009) is another example of a hierarchical statistical approach, however this work focuses on group interactions and uses a Bayesian model to recognise activities.

Syntactic Approaches

Syntactic approaches are those which model human activity as a string of actions. Similar to the statistical approach, the actions are the small tasks which make up an activity and must be recognised first. An example of a hierarchical syntactic approach can be seen in the work of Joo & Chellappa (2006), who developed a system using attribute grammars to detect abnormalities in a business parking lot. A syntactic

approach was also used in work to monitor people interacting with puzzle games (Minnen, Essa, & Starner, 2003), to which sub actions were used to recognise more complex activities.

Description Based

In description based approaches the structure of activities is characterised and sub activities are detected based on a predefined activity relationship model. Therefore in this approach, recognition is done by searching for the sub activities that satisfy the relationship model. A system created by Ryoo & Aggarwal (2008) demonstrates how a description based approach can be used to recognise group interactions as well as interactions between multiple groups, e.g. Two groups fighting. A description based approach has also been used as a way to construct plots for a video (Gupta, Srinivasan, Jianbo, & Davis, 2009), in which a logical storyboard is created based off the activities recognised in a baseball game.

2.2 Review of Related Concepts

This section presents a technical review of concepts that are relevant to this research. It is split into three main sub-sections, the first describing the different types of relevant distance measures, followed by a review of the Busy/Idle technique for characterising pixels, and finally a review of a weighting system for ADL detection.

2.2.1 Distance Measures

In image classification, distance measures are often used to determine similarities between data using a numerical value. Euclidian distance is a commonly used measure that provides the difference in squared distance of two values. This is calculated using the following formula, where the distance between points A and B is calculated as:

$$DISTANCE(A,B) = \left\{ \sqrt{\sum_{i=1\ldots N} (A_i - B_i)^2} \right\}$$

$$A = (A_1, A_2, \ldots, A_n)$$

$$B = (B_1, B_2, \ldots, B_n)$$

Equation 1

It is important to note that when using multiple factors to determine Euclidean distance, each factor should be normalised to the same scale to ensure they have the same weighting on the formula. This means that if the data is on separate scales, the scale with the highest numerical values will have the largest impact on the formula.

Manhattan distance differs in that the distance is calculated as the sum of the absolute difference between each feature. This is shown in the following equation, where the distance between points A and B is calculated as:

$$DISTANCE(A,B) = \left\{ \sum_{i=1\ldots N} |A_i - B_i| \right\}$$

$$A = (A_1, A_2, \ldots, A_n)$$

$$B = (B_1, B_2, \ldots, B_n)$$

Equation 2

Manhattan distance also requires each factor to be normalised to provide the same weighting on the formula.

2.2.2 Busy/Idle

Ermis *et al.*(2008) proposes busy/idle rates to characterise the behaviour profile of a given pixel from foreground objects. The relevant idea behind this is that a series of pixels are represented as a series of values being either 1 (busy) or 0 (idle), separated as those that undergo motion and those that are a part of the background image

respectively. Ermis *et al.* use these busy/idle values to create a time series for each individual pixel, showing when activity in each pixel has occurred over time.

This research utilises the busy/idle rate in the form of obtaining the 'busy' fraction, which represents the amount of time each pixel is active. This will be used, in part, when obtaining a numerical representation of Activities of Daily Living.

2.2.3 Weighting System for ADL Detection

Noury *et al.* (2011) presents a method of detecting activities of daily living using data received from a residential power line. This process involves using a sensor box that is linked to the power line, which can detect the electrical impulse that occurs with the use of appliances and lights within the household. Using these electrical impulses, the system is able to identify which appliances/lights are being used and can track this over time.

The process can be split up into 5 phases:

1. **Generate electrical signatures**

The first phase involves obtaining unique signatures that can be used to identify each appliance. These unique signatures are obtained from the electrical impulse that occurs when appliances are turned on and off. This phase involves a learning period where the system matches each signature to each electrical appliance.

2. **Associate relationships**

In this phase, relationships are formed between daily activities and the electrical appliances detected, as well as the rooms in which they are

located. The following table gives an example of ADLs being matched to specific rooms and appliances:

Table 1: ADLs are matched to specific rooms and appliances (Noury, et al., 2011).

ADL	Room Location	Electrical Appliance
Feeding (cooking)	Kitchen	Light, fridge, heating, furnace, boiler, dish-washer…
Toilets	Toilets (WC)	Lighting, heating
Grooming	Bathroom	Lighting, heating, hair-dryer
Others	Any room	Lighting

3. Associate Weight Factor

This phase involves associating a weight factor, being a number from 0 - 3, for each device depending on how often it is used in each ADL. For example, the kitchen light is never used with the activity of grooming, so would be given a 0. However, it is nearly always used during the activity of breakfast and so in this case would be given a 3.This data is used to form the following matrix:

Table 2: Representation of ADLs through the weight each device has on each ADL (Noury, et al., 2011).

	ADL j			
Electrical device i	breakfast	…	grooming	WC
Kitchen light	$p_{ii} = 3$		0	0
Coffee machine	3		0	0
Toilet light	0		3	0
…	…	…	…	…

This matrix represents the weight of each electrical device (i) on each activity of daily living (j). Note that this matrix is different for each person.

4. Select ADL

At any given time, the system will elect the activity which maximizes the quantity using the following formula:

$$ADL(ti) = Max\left\{\sum_{i=1\ldots N} P_{ij} * r_i\right\}$$

$$ti = time$$

$$P_{ij} = Data\ from\ above\ matrix$$

$$r_i = 1\ if\ the\ electrical\ activity\ was\ performed, else\ 0$$

Equation 3

5. Derive chronogram signal

Finally, a chronogram signal is created for each activity using the data obtained from the above formula, showing time for the whole day and not a specific moment. These are then used to create a spatio-temporal graph, displaying all user activity throughout a day.

This research utilizes and extends parts of the method proposed by Noury, but does so with a focus on using a Kinect sensor to obtain data.

2.3 Summary

Several types of sensors have been used in the field of activity recognition and abnormality detection, including video streams, wearable sensors, as well as data received from a residential power line. Techniques and methods for activity recognition can be classified into Single-layered approaches and Hierarchical approaches, and these methods deal with the recognising a range of activities from simple actions through to high level activity sequences. This research investigates the use of a Kinect sensor, and deals with Noury *et al*'s

work for classifying ADLs by investigating and adapting this approach using data obtained via a Kinect sensor.

3. An Approach for Detecting Abnormality in ADLs

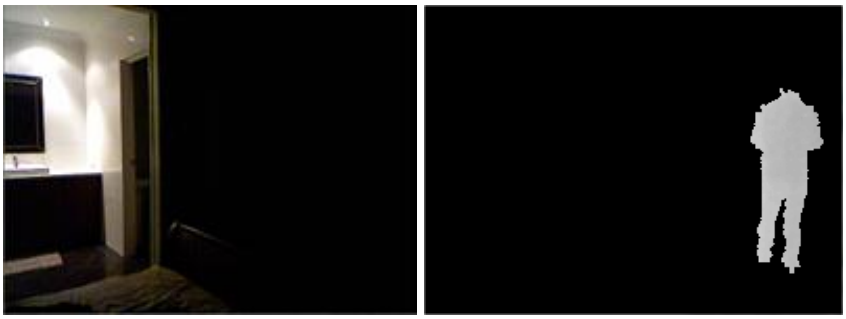
This chapter presents the techniques developed in this study. It details the procedure that was used, and provides an analysis of the data within each module. The approach consists of the following steps: *Data Collection, Pre-Processing, Feature Extraction, Signature Generation, Classification of ADLs, Abnormality Detection, and Presentation of ADLs.* In this approach, training Data is obtained and pre-processed using Kinect's in build person recognition. Collection of training data is necessary because the process in which ADLs are completed varies from person to person, and therefore no generic template to recognise ADLs exists. The research attempts to infer representations of ADLs through features related to the spatial position and orientation of the person in the field of view of the Kinect camera, which is divided into a grid of several data points. Once this representation or "ADL signature" is obtained, ADLs are classified live in unknown data using a combination of distance measures, and abnormal events are detected amongst these ADLs using an automatically generated threshold value.

3.1 Collecting Training Data

This section outlines the collection of training data. Specifically, it discusses the details of Kinect as a sensor, and presents the process involved with data collection for this research.

Collecting training data involves obtaining recordings of a specific subject completing several key ADLs using the Kinect. The Kinect is a sensor device that includes a depth camera, where the intensity of each pixel corresponds to the distance away from the camera. In addition, the Kinect also includes a colour video camera and microphone; both the video and depth camera output 30 frames a second. The 3D camera relies on an inbuilt infrared laser projection system, and thus is

tolerant to varying lighting conditions. The Kinect also includes functionality for person detection. The robustness is demonstrated in Figure 2, where a person not visible in the colour camera image is successfully detected using the built in person detection software from the depth image.



(a) **(b)**

Figure 2: Two images of the same scene. (a) from the Kinect's colour camera and (b) 3D depth camera image after processing using the built-in person detection software. Due to the lighting conditions the person is not visible in the colour image, but is detected in the 3D depth image due to the provision of an inbuilt infrared laser light source.

In this approach, both the depth image and a *binary mask of people* segmented from the depth image as provided by the Kinect software are utilised. This is because the research attempts to infer the current ADL through features related to the spatial position and orientation of the person in the field of view of the Kinect camera. Collection of training data is necessary is because the process in which ADLs are completed varies from person to person, and therefore no generic template to recognise ADLs exists. The process for the collection of training data includes the following steps:

Step 1: Setup Kinect/s within home. Note that the Kinect should be positioned so that the majority of interactive objects are within the view of the Kinect (see Figure 3).

Step 2: Record continuous stream for the length of the training period, obtaining both the *depth image* and the *binary mask of people*.

Step 3: Manually split and annotate videos into separate Activities of Daily Living.

The output produced from the data collection process is a set of annotated training videos for each ADL, containing both the depth image and binary mask of people.

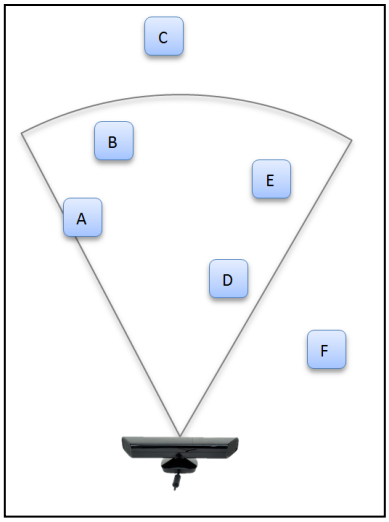


Figure 3: Room layout example. Blocks A – F represent locations or objects at certain locations that will be interacted with.

3.2 Pre-Processing

As a pre-processing step, the depth image is masked using the person detection data, so that only the depth information relating to the person detected is utilised, shown in Figure 4, with non-person related pixels set to black and the brightness of the remaining pixels corresponding to the distance of the person away from the camera. This provides 3D location information of the person in each frame in the depth camera's local coordinate system along the x (image column), y (image row) and z (pixel intensity) axes.

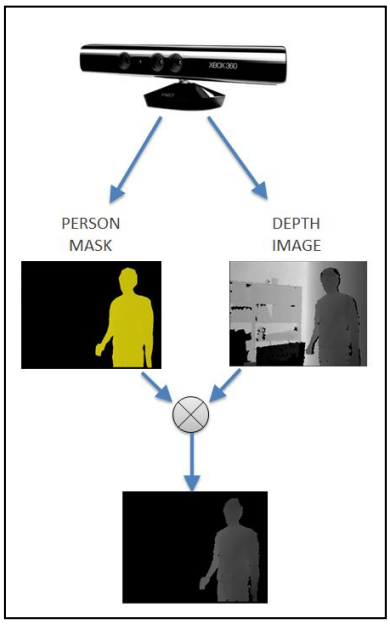


Figure 4: Obtaining a masked depth data image, showing only the depth data of the person detected

To allow for more efficient storage and processing, data reduction is made by sampling only one depth image frame per second from the 30 frames per second produced by the Kinect. Therefore, the output produced from pre-processing is a set of masked depth data, reduced to one frame per second. This data reduction is especially necessary when dealing with a stream of live data, in which the system will need to constantly store and process large amounts of data.

3.3 Feature Extraction

This section outlines the process of extracting features from the masked depth data. Specifically, it provides detail on *splitting the image into blocks* and *calculating each feature.* The features used in this research are depth, variance of depth, busy fraction, aspect ratio and form factor.

From the masked depth data, statistical information regarding the person's position over a set of image frames corresponding to either an

identified or candidate ADL is obtained. To reduce the amount of data, rather than do this per-pixel, the image pixels in each frame are divided into *N* blocks. The average intensity (depth) of the pixels inside each block is used for further processing, with features computed from this across a series of frames. In this approach, the 320 x 240 pixel depth image was divided into 25 regions of 64 x 48 pixels each. This reduces each image frame from 76800 pixels to 25 data points. This data reduction allows for more efficient storage and processing, and the approach is scalable to various hardware capabilities by adjusting the pixel block size and frame sample rate.

In this research, for each pixel block time series, i, a vector, W_i, containing multiple features is computed. These features are: the average depth value, A_i, over the sequence of frames, the variance of this depth, V_i, and the 'busy' fraction, B_i, corresponding to the proportion of frames in the sequence where a person is present in the pixel block. The aspect ratio of the bounding box of the detected player is also computed, R_i, as well as the form factor of the shape of the detected person, F_i.

The average depth and variance of this depth are obtained using the pixel intensity from the masked image, which indicates the distance of each pixel from the Kinect. The busy fraction is calculated from the sequence's busy/idle rate. A pixel block is labelled as "busy" when a person is present in it. The busy fraction is then calculated as the proportion of busy frames in the sequence.

Aspect ratio and form factor are features based on the shape of the person in the masked image and form a simple measure of the person's orientation. Since a person may occupy several image regions, when extracting these features, a pixel block cannot be analysed directly.

Instead, the full frame of the image is used to obtain these features. Once a feature value is calculated from the complete image, any pixel block that has been labelled as busy in the image is given this value. In a small number of cases, the detected person in the masked image is split into multiple blobs (shapes) by the Kinect's person finding algorithm (e.g. the head may be disconnected from the body). If this is the case, the blob with the largest area is used when extracting these features.

Aspect ratio is calculated by first obtaining the bounding box of the detected person in the image (see Figure 5), in which the feature is calculated as:

$$Aspect\ Ratio = \frac{Width\ of\ Bounding\ Box}{Height\ of\ Bounding\ Box}$$

Equation 4: Calculating Aspect Ratio

Using aspect ratio as a feature allows the system to differentiate between when a person is standing, sitting or lying down as the shape of the bounding box changes in each scenario.

Figure 5: The red box indicates the bounding box of the person in the image. The aspect ratio feature is calculated using the width and height of the bounding box.

Form factor is a shape descriptor that changes based on the changing perimiter of a shape. This feature is used to differentiate objects that contain the same area but differ in actual shape (see Figure 6).

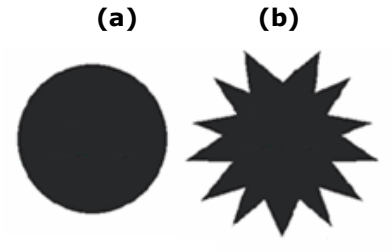


Figure 6: Shapes (a) and (b) both contain the same area; however the measured form factor changes as the perimeter increases. Form factor for shape (a) and (b) are 0.93 and 0.16 respectively.

Form factor is calculated using the following formula:

$$Form\ Factor = \frac{4\pi\ .Area}{Perimeter^2}$$

Equation 5: Calculating Form Factor

3.4 ADL Signature Generation

An ADL signature refers to the unique representation of an ADL that can be used for classifying against unknown data. This section outlines the techniques used in creating a representation of each ADL using the various features that are extracted from the masked depth data. This section details the *normalisation* and *grouping* of each feature.

Each feature vector variable is scaled to be in the range between 0 and 1. To do so, the average depth values are divided by the maximum range of the Kinect, which is 4000mm. The values obtained from the variance of depth are divided by the maximum variance found in the training data, where any test data that exceeds this limit is capped to 1. The 'busy' fraction and form factor are already obtained in the range

of [0 – 1], therefore no further normalisation is required for these factors. The aspect ratio values are divided by the maximum ratio that can be obtained from the 320 x 240 image, in this case being the maximum width of 320.

In order to build a profile of descriptor values for each ADL, these descriptors are calculated for data corresponding to specific ADLs in the training data. As variations exist in performing a particular ADL, multiple recordings of each ADL need to be analysed to produce an average descriptor value. Feature vectors, $W_{ij} = \{A_{ij}, V_{ij}, B_{ij}, R_{ij}, F_{ij}\}$, can be obtained for each pixel block $i \in [1,N]$ and ADL $j \in [1,K]$, where there are N pixel blocks and K ADLs. An example of a grouping of descriptor vectors to pixel blocks and ADLs is shown in Table 3.

Table 3: A feature vector, W_{ij}, is determined for each pixel block i in each ADL frame sequence j.

	ADL (j)			
Pixel block (i)	Cooking (1)	Cleaning (2)	...	(K)
1	$W_{1,1} = \{A_{1,1}, V_{1,1}, B_{1,1}, R_{1,1}, F_{1,1}\}$	$W_{1,2} = \{A_{1,2}, V_{1,2}, B_{1,2}, R_{1,2}, F_{1,2}\}$		$W_{1,K}$
2	$W_{2,1} = \{A_{2,1}, V_{2,1}, B_{2,1}, R_{2,1}, F_{2,1}\}$	$W_{2,2}$		$W_{1,K}$
....				
N	$W_{N,1}$	$W_{N,2}$		$W_{1,K}$

$A : Average\ Depth$ $R : Aspect\ Ratio$

$V : Variance\ of\ Depth$ $F : Form\ Factor$

$B : Busy\ Fraction$

Once constructed, this table of feature vectors represents the unique signature for each ADL and can now be used to compare against unknown data.

3.5 Classification of ADLs

This section focuses on activity recognition and classification, in which the goal is to be able to recognise specific ADLs as they are being performed live. It outlines the techniques used to test live data against the signature vectors of each ADL. Specifically, it provides details on the *live video buffer, the distance measures* and *the voting system* that are used to classify ADLs.

The live data is obtained from the Kinect that has already been setup in the home, and the system requires a buffer of the live data to accurately determine region activity that has previously occurred (see Figure 7). The system detects an appropriate buffer size based on the size of the training videos. This is done by attaining the maximum, minimum and average length of the training videos, and using them to create three different buffer sizes respectively. At each recognition point, all three buffer sizes are tested to obtain separate classification results to which a voting system is used to determine the most probable classification.

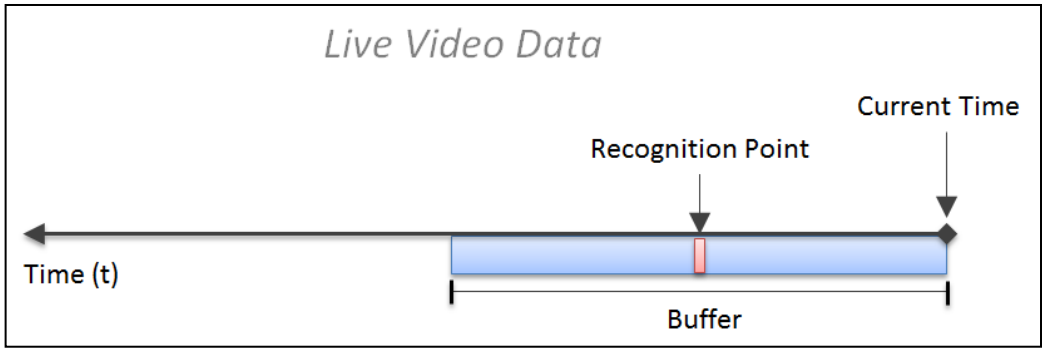


Figure 7: The use of a buffer for live video data.

A buffer is necessary because ADLs are performed over a period of time, therefore the system will detect an ADL through a sequence of

image frames and not using a single frame. As shown in Figure 7, the system will select the ADL that occurred at "Recognition Point".

To determine similarity between ADLs and an unknown stream of live data, the approach uses a voting system consisting of three different similarity measures; these being Euclidian distance, Manhattan distance, and Mahalanobis distance. To classify new data from the Kinect at a particular point in time, t, the previous data obtained from the buffer (individually for each of the three buffer sizes) is used to determine average depth, $AR_i(t)$, variance of depth, $VR_i(t)$, busy fraction, $BR_i(t)$, aspect ratio, $RR_i(t)$, and form factor, $FR_i(t)$ for each pixel block i.

In Euclidian distance, the ADL for the current time's recognition point, $ADL(t)$, is classified as ADL j so as to minimise the sum of Euclidian distances between feature vectors of pixel blocks of ADL j and the feature vectors of the unknown activity, shown in the following equation:

$$ADL(t) = \min_{j\in[1,K]} \sum_{i=1}^{N} \sqrt{(A_{ij} - AR_i(t))^2 + (V_{ij} - VR_i(t))^2 + (B_{ij} - BR_i(t))^2 + (R_{ij} - RR_i(t))^2 + (F_{ij} - FR_i(t))^2}$$

Equation 6: Euclidean Distance

Manhattan distance differs in that the distance is calculated as the sum of the absolute difference between each feature. This is shown in the following equation:

$$ADL(t) = \min_{j\in[1,K]} \sum_{i=1}^{N} |A_{ij} - AR_i(t)| + |V_{ij} - VR_i(t)| + |B_{ij} - BR_i(t)| + |R_{ij} - RR_i(t)| + |F_{ij} - FR_i(t)|$$

Equation 7: Manhattan Distance

Mahalanobis distance is also used as a distance measure of the developed approach, and as it takes into account the correlation between the data, Mahalanobis distance is calculated using the average depth, aspect ratio, and form factor features only. This is shown in the following equation:

$$ADL(t) = \min_{j \in [1,K]} \sum_{i=1}^{N} \sqrt{(x-u)^T S^{-1} (x-u)}$$

$$x = \left(AR_i(t), RR_i(t), FR_i(t)\right)$$

$$u = \left(A_{ij}, R_{ij}, F_{ij}\right)$$

$$S = Covariance\ Matrix\ of\ [A_{ij}, R_{ij}, F_{ij}]$$

Equation 8: Mahalanobis Distance

Once all three distance measures are separately calculated, a voting system is then used to determine the most likely ADL that occurred at that point. The voting system selects the ADL that has been classified by the majority of the distance measures. In the event where all three distance measures have calculated different classification results, no ADL is selected. Once the classified ADL is obtained from the majority vote, this classification process is repeated for each buffer size and the voting system is used on each buffer size to determine the final classification. The benefit of a voting system is that it allows the incorporation of three distance measures, all working differently, to provide input into the final result.

3.6 Abnormality Detection

This section outlines the techniques used to detect abnormalities amongst the ADLs. Specifically, it details the use of an automatically

determined threshold value and how this value is generated. In the context of this research, normal activities refer to those that have been monitored as training data. Abnormal activities are classified as events that are significantly different to, and therefore do not match, the activities monitored during the training phase.

To detect abnormalities, the system applies a generated threshold value when measuring the difference between the training data ADLs and the unknown ADL being classified by the system. That is, after an ADL has been classified by the system, the Euclidean distance between the feature vector of the current ADL and the feature vector of the class that it has been classified as, is compared to the threshold value. If the distance exceeds the threshold value, the system is said to have detected an abnormal event.

The threshold value is generated during the training of the system. To generate this value the system compares each individual ADL training video against the combined feature vector table that has been created; this is to obtain an average distance between each training video and the representation of its ADL. The system calculates the mean of these distances. This is used as an expected value of distance from an unknown video to the particular ADL. To account for the likely variation of an unknown video, the system allows a degree of sensitivity to abnormal events. An activity is flagged as abnormal if the distance is more than *n* standard deviations from the expected value.

3.7 Presentation of ADLs

This section outlines the final output of the approach. In detail, it provides information on both the *chronograms,* which represent the time in which each ADL has been detected, and the *spatio-temporal graphs,* displaying the combined chronograms for all activities mapped over several days. These graphs are useful in that they provide a representation of the behaviours of a person, and can be used by both doctors and carers as observational data.

To present the detection results, a chronogram signal for each activity is created using the classification method described in previous sections. These chronograms separately represent the time in which each ADL has been detected within the unknown data. An example of this, shown in Figure 8, demonstrates ADL 1 has been detected in 3 instances within the test video. The x-axis shows the time, in this case over a 24 hour period with the y-axis providing a value of 0 or 1 depending on whether or not the activity was detected.

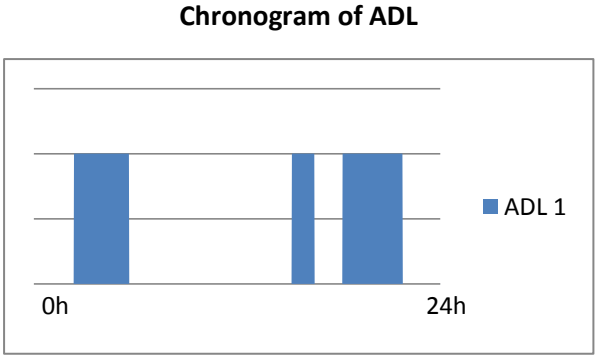


Figure 8: Example of chronogram graph, showing the time interval over 24 hours. In this example, ADL 1 is detected on 3 separate occasions over the 24 hour period.

Following this, these chronograms can be combined to create a spatio-temporal graph that displays occurrences of all activities. The x-axis of the spatio-temporal graph can be used to represent the time within

each day (24 hours), while the y-axis is used to display the number of days. This information, when gathered over several weeks/months, forms the "signature" of a person's activity as demonstrated in Figure 9.

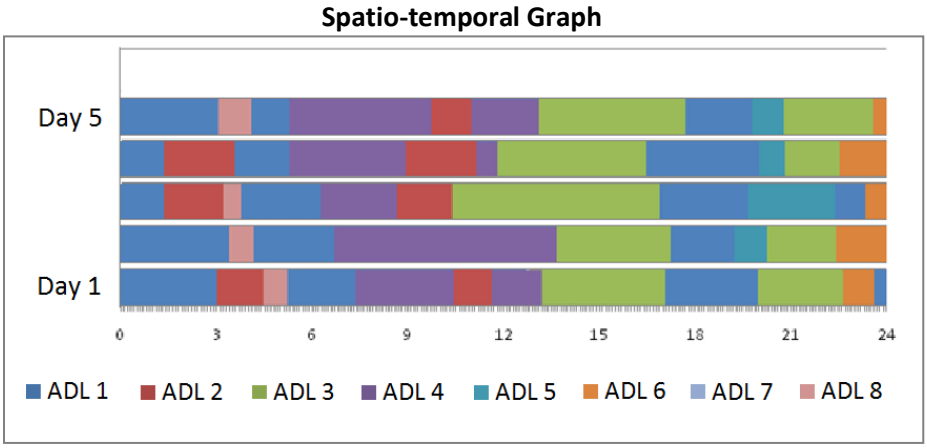


Figure 9: Example of a spatio-temporal graph over 5 days. This represents the "signature" of an individual. X-axis shows the time interval over 24 hours.

The spatio-temporal graph provides an overview of a person's activity and can provide valuable observational data over a long-term. It can allow doctors or carers to pick up "warning" signs about a person, things such as restlessness at night, missed meals or a change in a person's overall behaviour and lifestyle. Currently, the majority of home care monitoring is performed through home visits. However, this method is costly and can only be done in set intervals. The spatio-temporal graph over a long-term could supplement home visits and provide a less costly monitoring alternative and a higher feeling of independence for the person being monitored.

3.8 Summary

The process in which ADLs are completed varies from person to person, and therefore no generic template to recognise ADLs exists. For this reason, the collection of training data is necessary to create an approach that is suitable for all people. In this research, ADLs are represented as unique signatures that are formed through features related to the spatial position and orientation of the person in the field of view of the Kinect camera. Using this ADL signature, ADLs are classified live in unknown data using a combination of distance measures, and abnormal events are detected amongst these ADLs using an automatically generated threshold value. The recognition of ADLs is used to form spatio-temporal graphs, providing an overall profile of a person's activity. Over a long-term, these graphs provide valuable observational data that could provide a less costly monitoring alternative by supplementing home visits.

4. Experiments and Results

This chapter presents the experiments and results of the research. Several experiments were carried out in order to evaluate the effectiveness of the developed algorithm. In the developed approach, the intended order of process includes using the Kinect to record a continuous stream for the length of the training period, and manually splitting and annotating this video into separate Activities of Daily Living as part of training the system. Following this, after the system has created representations for each ADL, live data can be processed continuously from a live stream to obtain information on the monitored person. To allow a controlled and systematic approach to the experiments in this research, a live data stream is not used. Instead, ADLs are each individually recorded following a set of pre-determined scripts based on the movements and interactions around certain locations in a room. By obtaining both the training data and test data as individual recordings, experiments can be setup in a controlled environment and multiple recordings can be joined together to form larger test sequences that imitate a continuous stream. This chapter is arranged into the following sections: *Data, Leave one out Testing Approach,* and *Experiments. Data* outlines the techniques associated with obtaining all video recordings that were used in this research, specifically detailing the process of generating ADL data. *Leave one out Testing Approach* explains the procedure behind allocating training and test data for each experiment, while *Experiments* provides discussion and results on each experiment.

4.1 Data

To test the proposed approach, data associated with various ADLs is generated by performing a set of scripted ADL-like activities in a room setup for the experiment. Figure 10 shows this setup, which consists of 6 different activity nodes (A-F) with nodes A, C, D, E being within the Kinect's field of view and points B and F located outside this range.

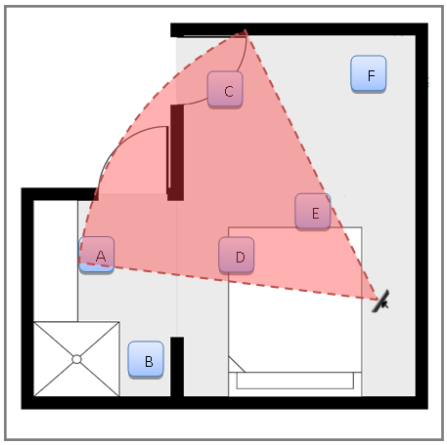


Figure 10: Room layout, showing the location of each activity node and the position of the Kinect.

In this research, an ADL is modelled as a sequence of time duration of a person's location at and between a set of defined activity nodes. For example, a 'cooking' ADL may be comprised of time spent at the activity nodes comprising of stove, fridge, sink etc. This will differ from the ADL of cleaning which may only involve time spent at the sink. These "generic activity sequences" consisted of several activity points, both in and out of the field of view of the Kinect.

To generate training data for ADLs in this experiment, generic activity sequences were performed using scripts based on the sequence maps shown in Figure 11. Each generic activity sequence consists of elements similar to the following generic script:

GENERIC SCRIPT

Start at point **[x_i].**

1. Interact here for **[y]** seconds.
2. Move to new **[x_i]** in **[z]** seconds, interact here for **[y]** seconds.

… Repeat until end of ADL

Finish ADL recording.

$[x_i] \in$ {activity node: A – F }

[y] = Time in seconds spent at activity node.

[z] = Time in seconds spent moving between activity nodes.

Figure 11 shows the sequence maps for each ADL used in this experiment. Using ADL 1 as an example, it can be seen that node E is the starting node and is visited 3 times during this sequence. [E_1, E_2, E_3] indicates the amount of time spent in each visit to E where E_n represents the time duration spent at E in the n^{th} visit. ADL 1 and ADL 4 incorporate the same set of activity nodes, but differ in the time spent at each activity node as well as in the order of activities that occurred. In ADL 1, node C is visited once and sums to 30 seconds, D is visited twice and sums to 75 seconds, and E is visited three times and sums to 270 seconds. Whereas in ADL 4, nodes C, D, and E are all visited twice and sum to 90 seconds, 230 seconds and 60 seconds respectively. Full details of each ADL video that was recorded are provided in *Appendix A.*

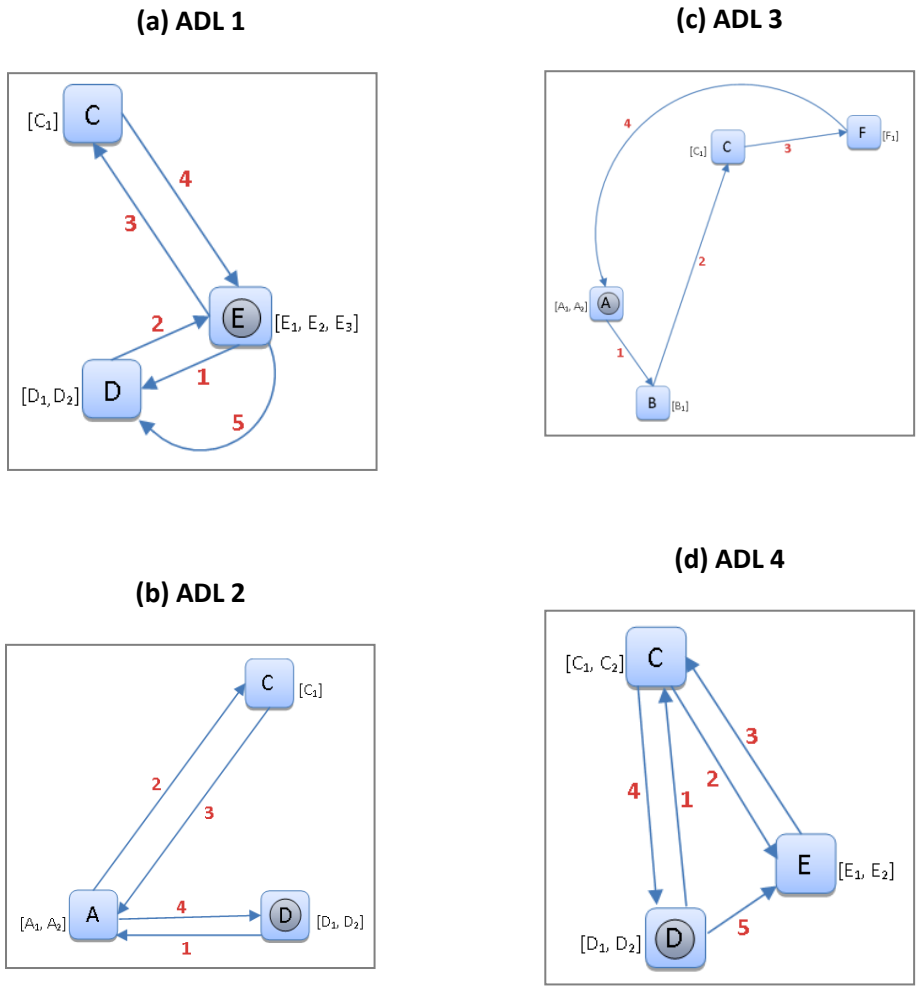


Figure 11: Each activity point (blue box) represents a person completing an activity at a certain location and also shows how many times an activity is completed at each point. The red numbers represent the sequence of movement between each activity. Note that each location point and transition has a dwell time. The node with a shaded circle indicates that it is the starting point for the ADL.

Note that each ADL is recorded several times under a variety of conditions, including different lighting conditions and clothing. The scripts used to create these recordings were set as a general guideline for each ADL recording, and in actual recordings, there are variations (in terms of time durations) to imitate realistic activities and accurately test the effectiveness of the developed system. In addition to the above

mentioned ADLs, other variations of the ADL were also recorded, in which the sequence that the activity nodes are visited is modified. An example of a modified ADL script is shown in Figure 12.

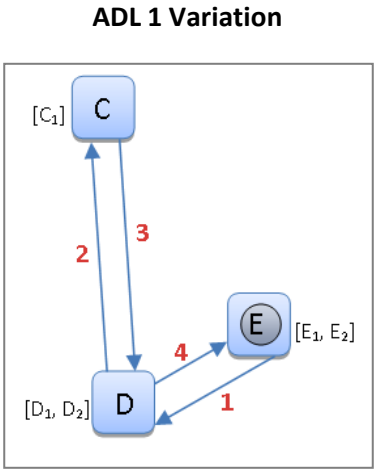
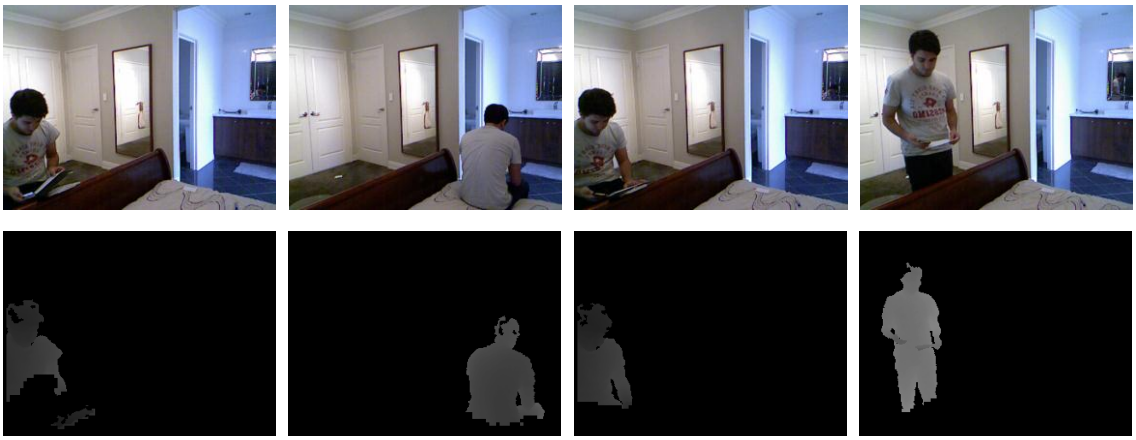


Figure 12: Sequence map showing a variation of ADL 1. Overall time spent at each point is similar, however the order and number of times each node is visited is significantly different.

Varied ADLs share similarities to the original ADLs in terms of the general amount of time spent at each activity node, however the order in which each node is visited and the number of movements made between each node are significantly different. These modified ADLs are obtained to test how effective the developed approach is when dealing with a different sequence of activity nodes for the same ADL.

Figure 13 displays several frames taken from recordings of ADL 1, ADL 2, and ADL 3 respectively. The figure shows samples of both regular colour images as well as their corresponding masked depth image frame. For complete annotations of each ADL video used in this research see *Appendix A,* which includes both the initial scripts and exact details of each video.

(a) ADL 1



(b) ADL 2

(c) ADL 3

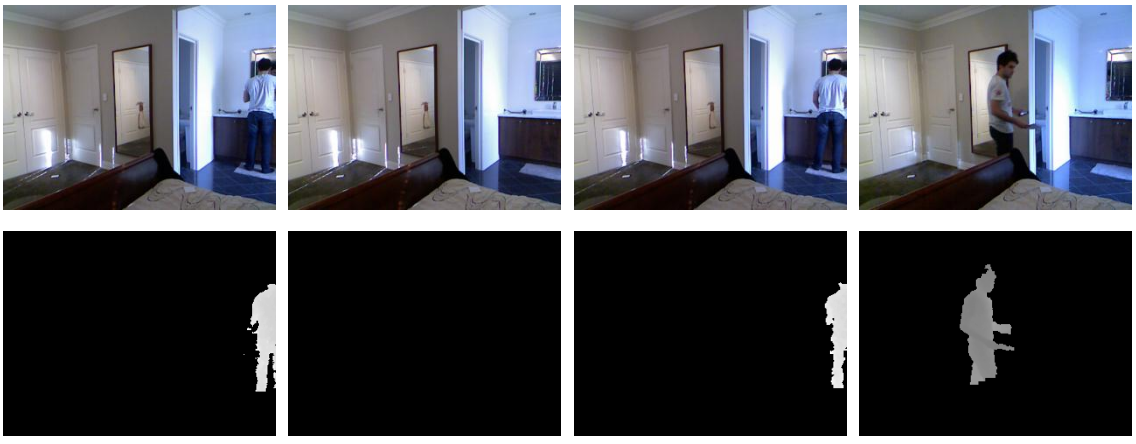


Figure 13: Sample frames taken from a recording of (a) ADL 1, (b) ADL 2, and (c) ADL 3. Below each colour frame is its respective masked depth image. See *Appendix A* for full details on each ADL recording.

4.2 Leave one out Testing Approach

The dataset used for experimenting in this research consists of four classes of ADLs, each with three to five video sequences. To ensure that the dataset is utilised effectively, the research follows the testing approach of "leave one out". The leave one out approach is a method of testing where a video sequence is not used as part of the training data and therefore becomes the test video. Unlike the generic approach of splitting a dataset into halves of training data and testing data, which is intended for larger datasets, this approach allows each video to be systematically tested against all other videos, where in each run the video being tested is excluded from the training data.

Some of the experiments in this research combine several ADL recordings into one continuous stream. To create these combined test videos, one recording from each ADL is selected and all are linked together. The leave one out approach also applies to these experiments, in that any recording that is used to form the test video is not included into the training data.

4.3 Experiments

A number of experiments were carried out to evaluate the proposed approach. These included ADL classification in which *Individual, Combined,* and *ADL Variation* Video Data is tested; as well as experiments focused on *Abnormality Detection, Feature Comparison,* and *Distance Measure Comparison.*

The purpose of the individual classification experiment is to evaluate the classification approach when identifying single ADLs from separate individual test videos. Combined video testing involves joining several ADL videos to create a single test video; this is to evaluate how the system reacts to the transitions between each ADL. Experiments were conducted using the ADL variation data, to evaluate whether the order in which the activity nodes are visited will affect the performance of the classification approach. *Abnormality Detection* is evaluated by testing unlabelled, and therefore abnormal, ADLs against an automatically generated threshold value in a combined video test. Lastly, comparison experiments are used to evaluate features and distance measures individually. All annotations of video data that was used in the following experiments can be seen in *Appendix A.*

4.3.1 ADL Classification using Individual Video Data

The aim of this experiment is to evaluate the classification approach using four classes of ADLs, to see if it is able to identify each ADL correctly when tested individually.

The dataset used in this experiment consisted of four ADLs, each with three video sequences. The video sequences used were Video 1, Video 2 and Video 3 for each ADL, as described in *Appendix A.* A total of 12 runs of detecting ADLs from test videos were conducted. In the first run, one video sequence from ADL 1 is left out (to be used as the test video) and the remaining video sequences are used to generate the feature vector table. This process is repeated 11 more times by systematically substituting the video that is left out with one of the remaining videos.

The system's output for each of these tests can be found in *Appendix B.* In each run, the proposed algorithm correctly identified each of the test videos with respect to its corresponding ADL. In terms of ADL 1 and ADL 4, this is significant as it demonstrates that the algorithm is able to differentiate between ADLs that comprised of the same set of activity nodes but varied in the time duration and order of visits at the different activity nodes.

Figure 14 displays the results of a test on *ADL 1 – Video 1,* where the algorithm has correctly classified the unknown ADL as ADL 1. In the resulting spatio-temporal graph, the length of the test video is represented on the x-axis in seconds. The empty space at the beginning and end of the graph is due to the system using a buffer, in that classification begins once enough data has been obtained for the size of the buffer.

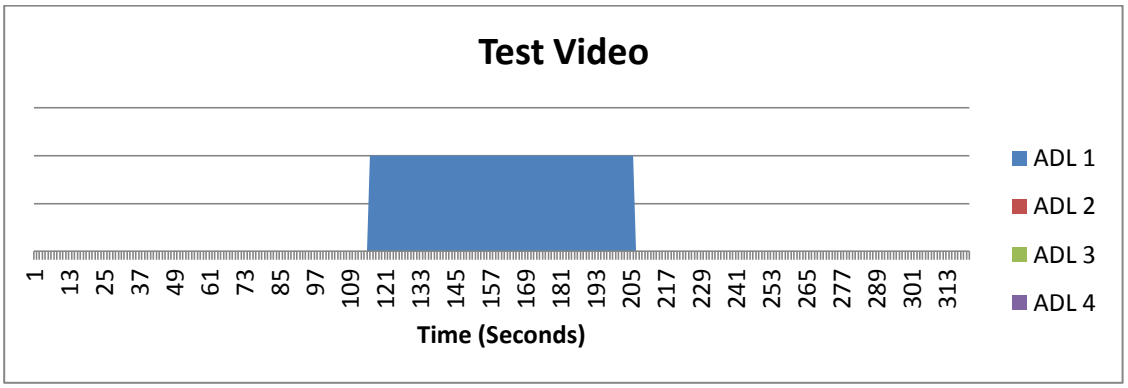


Figure 14: Spatio-temporal diagram showing recognition of ADL 1 in *test case: ADL 1 – Video 1*. The algorithm correctly classifies the unknown ADL as ADL 1.

4.3.2 ADL Classification using Combined Video Data

While the previous experiment demonstrates the system's robustness for classifying ADLs as single inputs, a crucial aspect of the developed approach is that data is analysed as a continuous stream. Therefore, the aim of this experiment is to evaluate the classification approach when tested on a video that contains several ADLs.

The experiment involved joining several ADL recordings into a single test video, to test how the system reacts during the transitions between each ADL. The dataset used in this experiment was the same as the previous experiment, consisting of four ADLs, each with three videos being Video 1, Video 2, and Video 3 respectively. The unknown sequences being tested were created by linking together a video from each ADL into one stream. To compare against ground truth, separate spatio-temporal diagrams for each of the unknown sequences were created by manually annotating the unknown video. All remaining videos in the dataset were used as training videos in each case.

Figure 15 shows the resulting spatio-temporal diagram for one of the combined test videos, which was created using a separately recorded

video of each ADL, in the order of: *ADL 1 – ADL 3 – ADL 2 – ADL 4.* In this test, the developed approach was able to correctly identify each ADL, and has done so with typically clean transitions between each ADL.

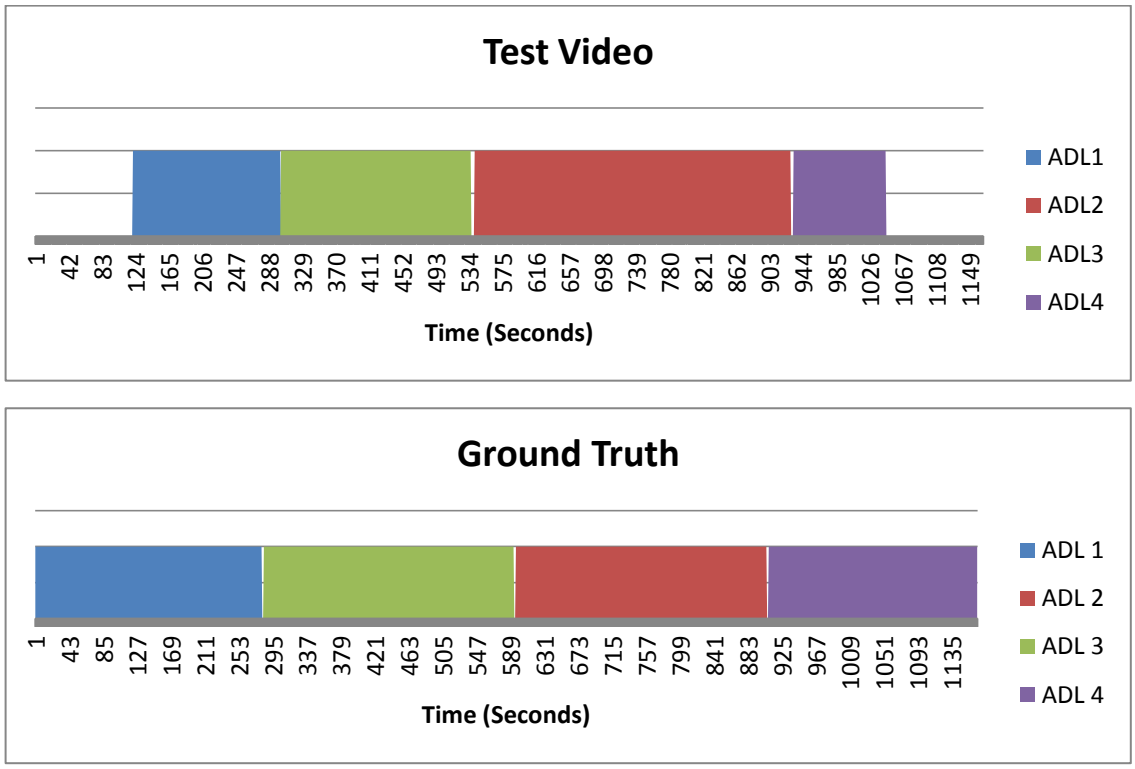


Figure 15: Spatio-temporal diagram showing recognition for the combined *test case: ADL 1 – ADL 3 – ADL 2 – ADL 4* (Test Video). The algorithm correctly identifies each ADL against the ground truth. The spatio-temporal diagram containing the ground truth was manually obtained by annotating the test video.

All test cases in this experiment presented results that correctly classified each ADL for the majority of the ADLs duration, with only one test showing minor misclassification during a transition. For the complete set of tests from this experiment, see *Appendix C.* Figure 16 displays the results from the *combined test case: ADL 2 – ADL 1 – ADL 4 – ADL 3*. In this test, the ADLs are correctly classified at most points, though the transition between ADL 2 and ADL 1 is momentarily

classified as ADL 4. A transition between ADLs is the period of time after an ADL ends and a new ADL is beginning, and therefore the buffer contains data from both ADLs. It is important that the developed approach is robust to transitions, as the approach is intended to use a live and continuous stream of data in which transitions will occur often.

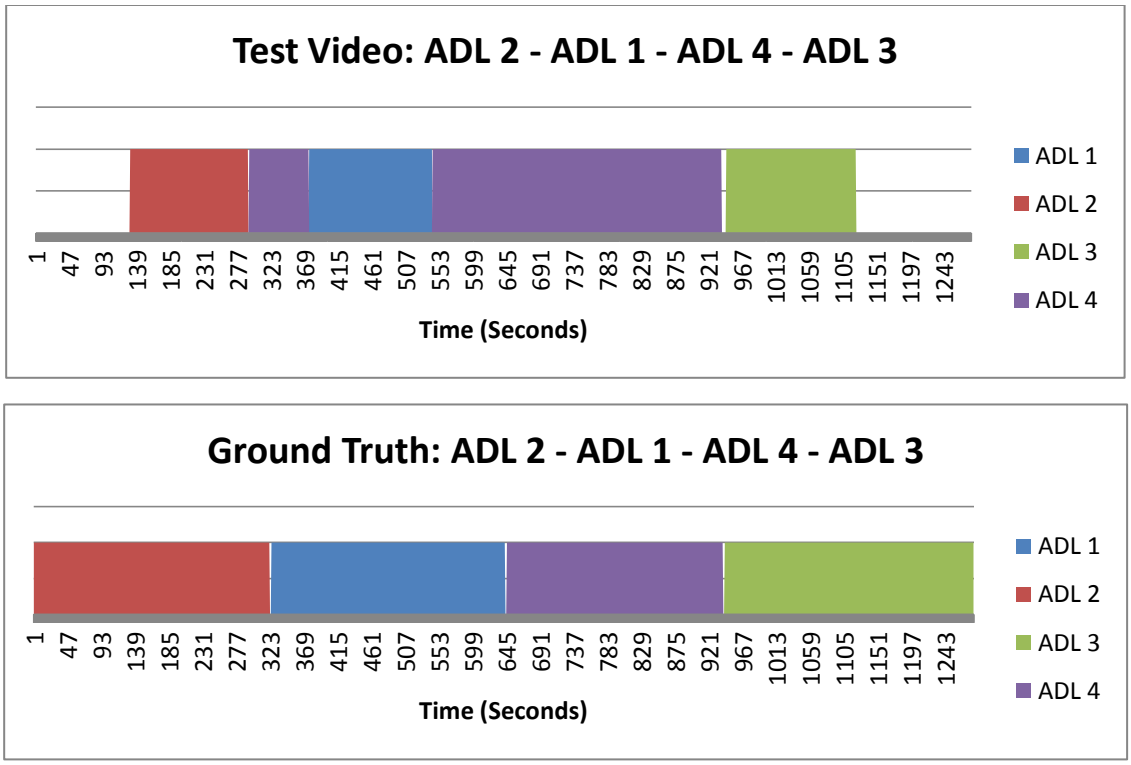


Figure 16: Spatio-temporal diagram showing recognition for the combined *test case: ADL 2 – ADL 1 – ADL 4 – ADL 3*. The algorithm correctly identifies each ADL at the appropriate times except during the transition between ADL 2 and ADL 1 to which a momentary misclassification occurs.

In each resulting spatio-temporal graph, gaps that effectively split each ADL during the transition phase can be seen. This is credited to the distance measure voting system, in which each distance measure has presented different classification results during these transitions, and therefore no ADL is classified at these points.

4.3.3 ADL Classification using ADL Variation Video Data

This experiment aims to evaluate the classification approach against ADLs that contain a modified sequence order; these being recordings based on the ADL variation scripts.

The training set used for this experiment consisted of four ADLs, each with three videos. All videos in the training set are recordings based off standard scripts for each ADL. The unknown video tested in this experiment was *ADL 1 – Video 5,* as defined in *Appendix A.* This test video is a variation recording of ADL 1 that differs from the standard ADL 1 in terms of sequence between each activity location.

The results of this experiment show that the ADL variation video was correctly classified as its respective ADL, see *Appendix D* for full details. Figure 17 shows the resulting spatio-temporal graph from this test,

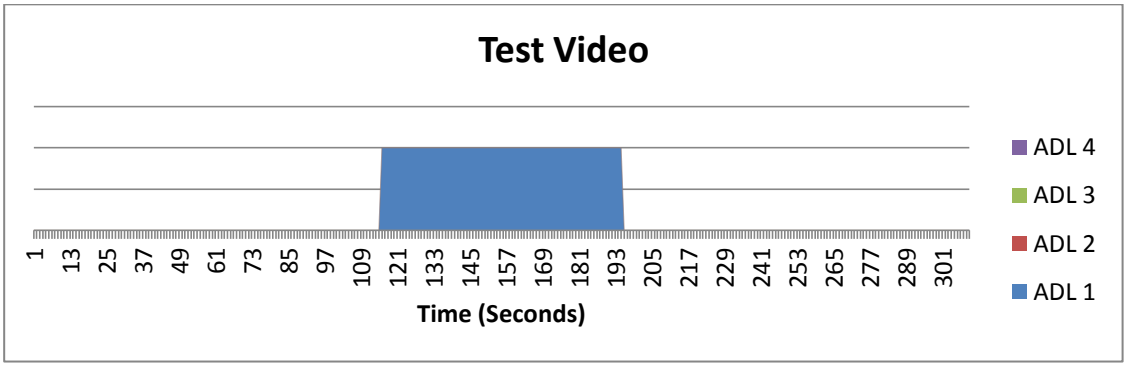


Figure 17: Spatio-temporal diagram showing recognition of ADL 1 in *test case: ADL 1 Variation Video*. The test video is a variation recording of ADL 1 that differs from the standard ADL 1 in terms of sequence between each activity location. The algorithm correctly classifies the unknown ADL as ADL 1.

This is significant for the reason that, while the varied ADL shares similarities to the original ADLs, the order in which the ADL was completed varies. This robustness to varying sequence is important when dealing with ADLs, as a person does not have a strict order of

process when competing ADLs. An example of this can be seen with the ADL of "Cooking", which may consist of time spent around the fridge, stove and sink. The order in which these location points are interacted with will vary from day to day and should not cause a misclassification.

4.3.4 Abnormality Detection

The aim of this experiment is to evaluate the abnormality detection approach presented in this research by testing unlabelled ADLs against the systems generated threshold value. Ideally, since samples associated with the unlabelled ADL are not included into the samples in the generation of the feature vector table during the training phase, the unlabelled ADL should be flagged as abnormal.

The dataset used in this experiment consisted of four ADLs, each with three video sequences. The video sequences used were Video 1, Video 2 and Video 3 for each ADL, as described in *Appendix A.* All video data from a single ADL was left out of training for each run to imitate an abnormal activity, and the test data for each test was a combined video containing a video from each ADL in the order of *ADL 1 - ADL 3 - ADL 2 - ADL 4.* In these tests, the generated threshold value, being the mean distance, plus one standard deviation was used detect abnormalities.

The results for this experiment can be seen in *Appendix E,* to which an abnormality test was carried out for each ADL. In each of these tests, the system correctly flagged the unknown activity as abnormal except for one case. The case that did not detect the abnormality was using ADL 4 as the unlabelled video, and could be attributed to its similarity with other ADLs, to which the misclassification provided a below threshold distance.

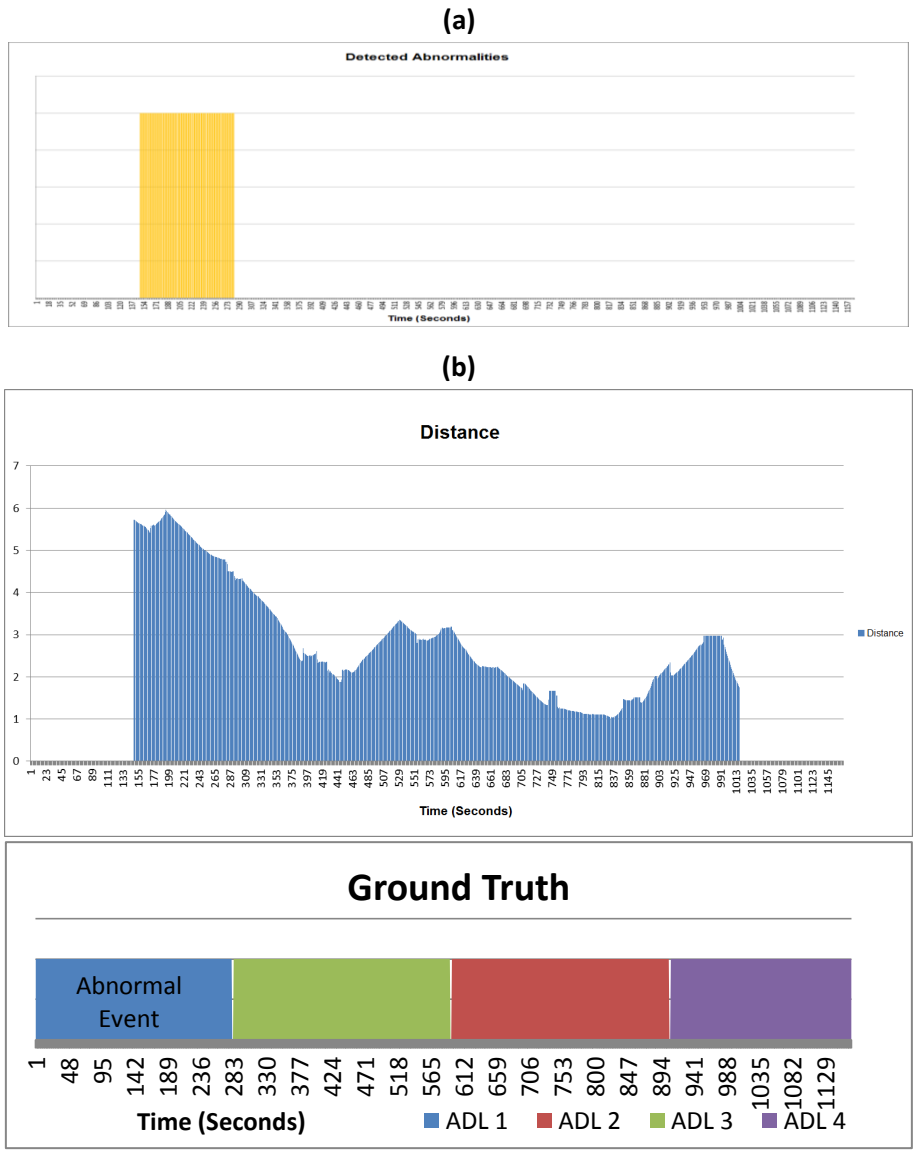


Figure 18: Results for *ADL 1 Abnormality Test,* showing (a) the points of classification in the test video that have been flagged as abnormal, and (b) the distance between an ADL signature and the unknown video at each point of classification. In this test, ADL 1 is the abnormal event as annotated in the ground truth. The automatically generated threshold value for this test was 4.528, and all distances above this are flagged as abnormal.

Figure 18 is a graphical representation of results for *ADL 1 Abnormality Test,* in which the system is tested for abnormalities against a test video containing ADL 1 as an abnormal event. The approach correctly detects the abnormal event at the appropriate time (a) using an

automatically generated threshold value. Figure 18 also shows the distance between the ADL signature and the unknown video at each point of classification (b), where the distance during the abnormal event is significantly higher than the rest of the recording.

It should be noted that a threshold value could also be used to effectively separate detected ADLs. To demonstrate this, an experiment that was tested in a previous section without abnormal events (*Combined Test 3* in *Appendix C)* can be used as an example. Figure 19 presents the distance between the ADL signature and the unknown video at each point of classification for this test. In this test all ADLs were classified correctly and the graph illustrates that the distance between the selected ADL and unknown ADL is significantly higher during the transitions between each activity.

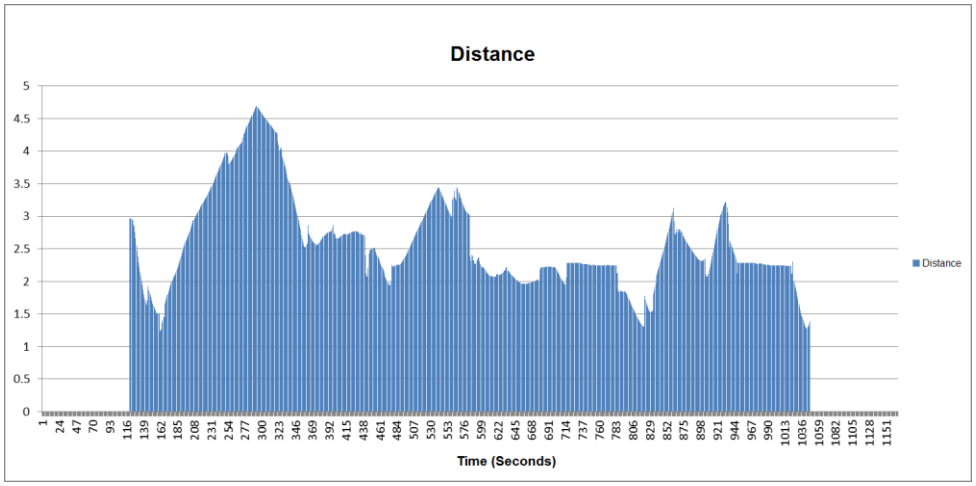


Figure 19: The distance between an ADL signature and the unknown video at each point of classification from a system that has trained with all ADLs, and has correctly classified each ADL. The distance is higher during the transitions between each ADL.

4.3.5 Feature Comparison

The developed approach utilises five features, they are depth, variance of depth, busy fraction, aspect ratio, and form factor. The aim of this experiment is to evaluate each feature when used individually in the classification algorithm, to gain a better understanding of the effect they each have in the developed approach.

The dataset used in this experiment consisted of four ADLs, each with three video sequences. The video sequences used were Video 1, Video 2 and Video 3 for each ADL, as described in Appendix A. The training data in each test consisted of Video 1 and Video 2 of each ADL. The test data was a combined video containing a recording (video 3) from each ADL in the order of ADL 1 - ADL 3 - ADL 2 - ADL 4. In these tests, the developed approach was first tested using all features, and then modified as to only include one feature in each run. The test cases were: *all features, depth only, variance of depth only, busy fraction only, aspect ratio only,* and *form factor only.*

The resulting spatio-temporal graphs for each run can be seen in *Appendix F.* These results show that the tests *using all features*, *variance of depth only*, and *busy fraction only* successfully classified each ADL with minimal error. While the tests *depth only*, *aspect ratio only*, and *form factor only* managed to classify ADLs but with a degree of error.

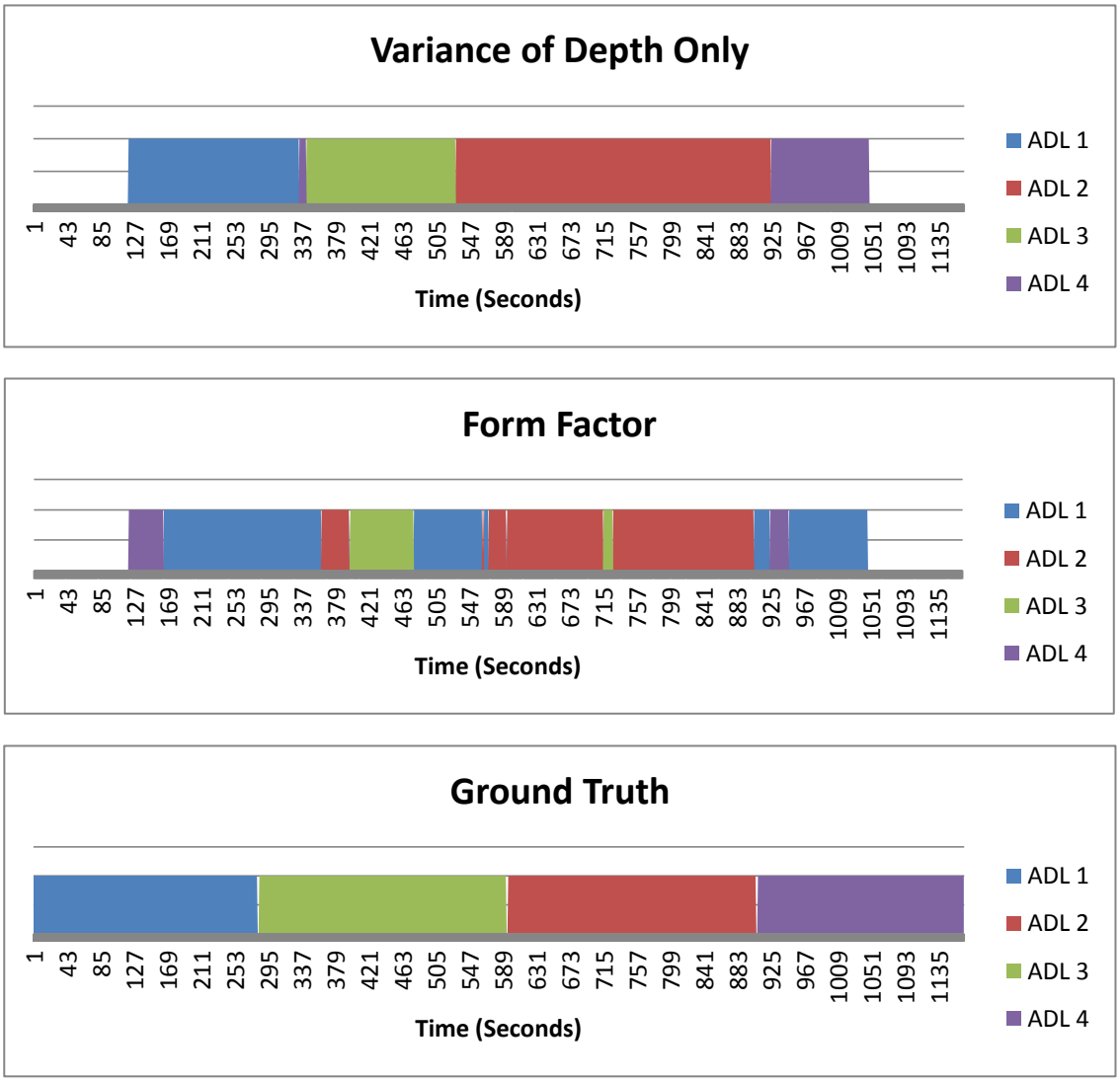


Figure 20: Spatio-temporal result of test video *ADL 1 – ADL 3 – ADL 2 – ADL 4* when using only variance and only form factor as the only feature in the classification approach respectively. Ground Truth shows the manually annotated spatio-temporal graph of the test video.

Figure 20 is the output of the *Variance of Depth Only* and *Form Factor Only* test respectively. In this experiment, using form factor as the only classification feature has performed the poorest; this degree of misclassification may be expected when only using a single feature in the classification approach. This experiment illustrates the value in

using multiple features in order to appropriately create an ADL signature representation.

4.3.6 Distance Measure Comparison

The developed approach incorporates the use of three different types of distance measures for classification; the aim of this experiment is to evaluate each distance measure when used individually in the classification algorithm, to gain a better understanding of the effect they each have in the developed approach. These distance measures are *Euclidean distance*, *Manhattan distance*, and *Mahalanobis distance*.

The dataset used in this experiment consisted of four ADLs, each with three video sequences. The video sequences used were Video 1, Video 2 and Video 3 for each ADL, as described in Appendix A. The training data in each test consisted of Video 1 and Video 2 of each ADL. The test data was a combined video containing a recording (video 3) from each ADL in the order of ADL 1 - ADL 3 - ADL 2 - ADL 4. In these tests, the developed approach was first tested using all distance measures (voting system), and then modified as to only include one distance measure in each run.

The resulting spatio-temporal graphs for each run can be seen in Figure 21, with full details provided in *Appendix G*. These results show that all tests were able to correctly classify each ADL; however Mahalanobis distance has done so with a degree of error during the transitions between each ADL, see Figure 21.

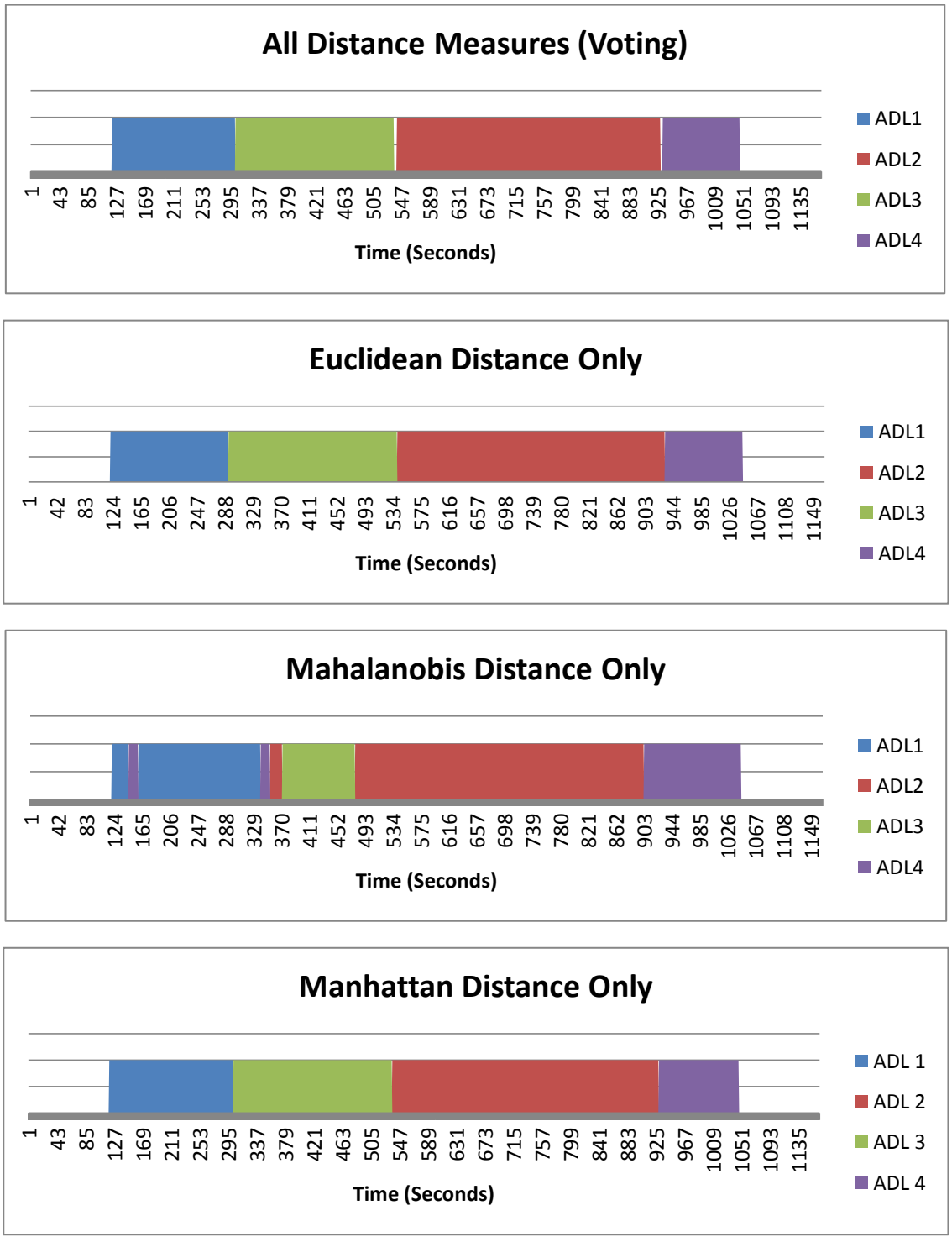


Figure 21: Spatio-temporal results for each test case in this experiment, displaying the differences in the distance measures used in this approach.

This experiment illustrates the value of using multiple distance measures as means of classification. In Figure 21, *All Distance Measures (Voting)* demonstrates the developed approach when using all three distance measures combined in a voting system, while the others demonstrate the system when using each distance measure individually. Specifically, it is during the transitions between each ADL that can cause the largest misclassification when using a single distance measure; as is seen when using *Mahalanobis Distance* Only.

4.4 Summary

Several experiments were carried out in order to evaluate the effectiveness of the developed algorithm. To allow a controlled and systematic approach to the experiments in this research, ADLs are each individually recorded following a set of pre-determined scripts based on the movements and interactions around certain locations in a room. Using these individually recorded videos, the leave one out testing approach was used to systematically test against all videos, where in each run the video being tested is excluded from the training data. The experiments in this research illustrate the robustness of the developed approach, in that it correctly classified both normal and varied ADL types. The approach has fittingly classified multiple ADLs from a combined stream input, and has done so with minimal misclassification during the transitions between each ADL. The experiments demonstrate the use of an automatically generated threshold value in detecting abnormal events, and evaluate the combined use of both features and distance measures.

5. Conclusion

This research has presented an investigation which explored a novel application of the Microsoft Kinect as a sensor for detecting abnormalities through the monitoring and recognition of Activities of Daily Living. Unlike existing work in recognition of ADLs using video recordings, which involved large amount of computation, the proposed approach attempts to use functionalities associated with Kinect to develop feature vectors that can be calculated very easily, thus making the task for detecting ADLs tractable when applied to real-time monitoring.

The ADLs in this study have been modelled as a sequence of time duration at a person's location at and between a set of defined activity nodes. This representation is aimed at modelling a set of structured activities in a structured environment (i.e. residence of an elderly person). Using the example of cooking as an ADL, this type of structured activity involves a fixed set of spatial locations in the residence and on an everyday basis would most likely involve very similar sequences and duration for an elderly person. In terms of facilitating ADL recognition, the representation for characterising an ADL should be robust against fine variations such as different sequencing of activities within an ADL or different time durations within each task. The feature descriptors used here are: the average depth value, *Ai*, over the sequence of frames, the variance of this depth, *Vi*, and the 'busy' fraction, *Bi*, corresponding to the proportion of frames in the sequence where a person is present in the pixel block. The aspect ratio of the bounding box of the detected player is also computed, R_i, as well as the form factor of the shape of the detected person, F_i. While these descriptors may appear to be rather "*high-level"*, they are robust to variations within individual sequences of images for each ADL.

The product of this research is a system that can learn based on ADL example videos, provide carers with a continuous profile of activity, and automatically detect abnormal events. This system can also scale to monitoring using multiple Kinect sensors, with each additional sensor simply adding its pixel blocks to the input data. As such, multiple rooms can be monitored, or multiple Kinect's used to monitor the same visual space to increase the richness of the data available for that particular room. The experiments in this research illustrate the robustness of the developed approach, in that it correctly classified both normal and varied ADL types, and was able to detect abnormal events using an automatically generated threshold value. The approach has fittingly classified multiple ADLs from a combined stream input, and has done so with minimal misclassification during the transitions between each ADL. Multiple features and distance measures are also evaluated, and demonstrate the value in combining these techniques to improve classification results.

Future work would involve modelling more complex ADLs which comprised of a higher number of activity nodes, either captured by a single Kinect or from multiple Kinects. To improve detection of ADLs, additional feature descriptors into the signature of ADLs could be explored, as well as to examine different similarity measures for matching. Future work may also include further research into the automatically generated threshold value, and how it could be applied to more effectively handle transitions between ADLs. The ultimate aim of this research is to develop a set of techniques which can be employed to detect abnormalities through characterising ADLs of an elderly person in their home, during the training phase over several weeks. The resulting "signature" of this person's activity in the form of a spatio-temporal graph obtained from the training phase can be used for

real-time monitoring. By processing recordings of ADLs from different homes, the corresponding ADL "signature" of its occupant can be captured.

References

Aggarwal, J., & Ryoo, M. S. (2011). Human activity analysis: A review. *ACM Computing Surveys (CSUR), 43*(3), 16-17.

Berenguer, M., Giordani, M., Giraud-By, F., & Noury, N. (2008, 7-9 July). *Automatic detection of activities of daily living from detecting and classifying electrical events on the residential power line.* Paper presented at the e-health Networking, Applications and Services, 2008. HealthCom 2008. 10th International Conference, 29-32.

Commonwealth of Australia. Department of Health and Ageing. (2008) *Ageing and aged care in Australia.* Retrieved from http://www.health.gov.au/internet/publications/publishing.nsf/Content/CA25774C001857CACA2574BE001A6E06/$File/Ageing_and_Aged_Care.pdf

J. Connell, C. Grealy, K. Olver and J. Power, Comprehensive scoping study on the use of assistive technology by frail older people living in the community, Urbis for the Department of Health and Ageing, 2008

Dai, P., Di, H., Dong, L., Tao, L., & Xu, G. (2009). Group Interaction Analysis in Dynamic Context. *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on, 39*(1), 34-42. doi: 10.1109/tsmcb.2008.2009559

Debard, G., Karsmakers, P., Deschodt, M., Vlaeyen, E., Van den Bergh, J., Dejaeger, E., . . . Vanrumste, B. (2011). Camera Based Fall Detection Using Multiple Features Validated with Real Life Video. *Workshop Proceedings of the 7th International Conference on Intelligent Environments 10*, 441-450.

Elgammal, A., Duraiswami, R., Harwood, D., & Davis, L. S. (2002). Background and foreground modeling using nonparametric kernel density estimation for visual surveillance. *Proceedings of the IEEE, 90*(7), 1151-1163.

Ermis, E. B., Saligrama, V., Jodoin, P. M., & Konrad, J. (2008, 7-11 Sept). *Abnormal behavior detection and behavior matching for networked*

cameras. Paper presented at the Distributed Smart Cameras, 2008. ICDSC 2008. Second ACM/IEEE International Conference, 1-10.

Gupta, A., Srinivasan, P., Jianbo, S., & Davis, L. S. (2009, 20-25 June). *Understanding videos, constructing plots learning a visually grounded storyline model from annotated videos.* Paper presented at the Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference, 2012-2019.

Hao, J., Drew, M. S., & Ze-Nian, L. (2006). *Successive Convex Matching for Action Detection.* Paper presented at the Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference, 1646-1653.

Laptev, I., Marszalek, M., Schmid, C., & Rozenfeld, B. (2008, 23-28 June). *Learning realistic human actions from movies.* Paper presented at the Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference, 1-8.

Minnen, D., Essa, I., & Starner, T. (2003, 18-20 June). *Expectation grammars: leveraging high-level expectations for activity recognition.* Paper presented at the Computer Vision and Pattern Recognition, 2003. Proceedings. 2003 IEEE Computer Society Conference, 626-632.

Mynatt, E. D., Rowan, J., Craighill, S., & Jacobs, A. (2001). *Digital family portraits: supporting peace of mind for extended family members.* Paper presented at the Proceedings of the SIGCHI conference on Human factors in computing systems, 333-340.

Natarajan, P., & Nevatia, R. (2007, Feb). *Coupled Hidden Semi Markov Models for Activity Recognition.* Paper presented at the Motion and Video Computing, 2007. WMVC '07. IEEE Workshop, 10-10.

Nguyen, N. T., Phung, D. Q., Venkatesh, S., & Bui, H. (2005, 20-25 June). *Learning and detecting activities from movement trajectories using the hierarchical hidden Markov model.* Paper presented at the Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference, *2,* 955-960.

Noury, N., Berenguer, M., Teyssier, H., Bouzid, M. J., & Giordani, M. (2011). Building an Index of Activity of Inhabitants From Their Activity on the

Residential Electrical Power Line. *Information Technology in Biomedicine, IEEE Transactions on, 15*(5), 758-766.

Rao, C., & Shah, M. (2001). *View-invariance in action recognition.* Paper presented at the Computer Vision and Pattern Recognition, 2001. CVPR 2001. Proceedings of the 2001 IEEE Computer Society Conference, *2,*316-322.

Ryoo, M. S., & Aggarwal, J. K. (2008, 8-9 Jan). *Recognition of High-level Group Activities Based on Activities of Individual Members.* Paper presented at the Motion and video Computing, 2008. WMVC 2008. IEEE Workshop, 1-8.

Seong-Wook, J., & Chellappa, R. (2006, 17-22 June). *Attribute Grammar-Based Event Recognition and Anomaly Detection.* Paper presented at the Computer Vision and Pattern Recognition Workshop, 2006. CVPRW '06. Conference, 107-107.

Stikic, M., Huynh, T., Van Laerhoven, K., & Schiele, B. (2008, Jan. 30 -Feb. 1). *ADL recognition based on the combination of RFID and accelerometer sensing.* Paper presented at the Pervasive Computing Technologies for Healthcare, 2008. PervasiveHealth 2008. Second International Conference, 258-263.

Takahashi, D. (2012). Xbox 360 surpasses 66M sold and Kinect passes 18M units Retrieved 25/5/2012, 2012, from http://venturebeat.com/2012/01/09/xbox-360-surpassed-66m-sold-and-kinect-has-sold-18m-units/

Tapia, E., Intille, S., & Larson, K. (2004). Activity recognition in the home using simple and ubiquitous sensors. *Pervasive Computing*, 158-175.

Vaswani, N., Roy Chowdhury, A., & Chellappa, R. (2003, 18-20 June). *Activity recognition using the dynamics of the configuration of interacting objects.* Paper presented at the Computer Vision and Pattern Recognition, 2003. Proceedings. 2003 IEEE Computer Society Conference, 633-640.

Wong, S. F., Kim, T. K., & Cipolla, R. (2007). *Learning motion categories using both semantic and structural information,* 1-6.

Zhang, D., & Lu, G. (2001). Segmentation of moving objects in image sequence: A review. *Circuits, systems, and signal processing, 20*(2), 143-183.

Appendix A: ADL Recordings

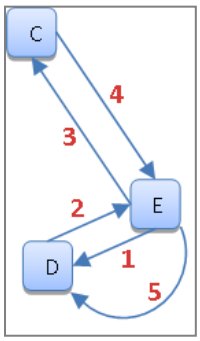
The following provides information on both the initial scripts used for each ADL recording, and the exact data manually extracted from each test video, to show the differences in each test video.

ADL 1

Initial Script

Estimated time: 400 seconds.

Start at point **E.**



1. Interact here for **30** seconds.
2. Move to **D** in **5** seconds, interact here for **25** seconds.
3. Move to **E** in **5** seconds, interact here for **120** seconds.
4. Move to **C** in **5** seconds, interact here for **30** seconds.
5. Move to **E** in **5** seconds, interact here for **120** seconds.
6. Move to **D** in **5** seconds, interact here for **50** seconds.

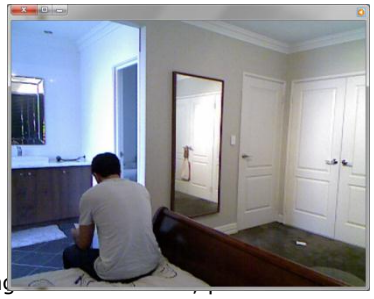
Finish ADL recording.

- Total time at point **C: 30** seconds
- Total time at point **D: 75** seconds
- Total time at point **E: 270** seconds

ADL 1 - Video 1

Exact Time: 5 minutes 21 seconds **(321 seconds)**

Start at point **E.**



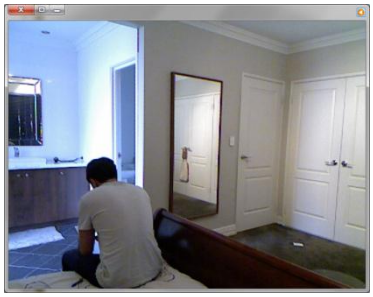
1. Interact here for **23** seconds.
2. Move to **D** in **4** seconds, interact here for **14** seconds.
3. Move to **E** in **4** seconds, interact here for **90** seconds.
4. Move to **C** in **4** seconds, interact here for **24** seconds.

5. Move to **E** in **6** seconds, interact here for **103** seconds.
6. Move to **D** in **6** seconds, interact here for **43** seconds.

Finish ADL recording.

ADL 1 - Video 2

Exact Time: 3 minutes 59 seconds **(239 seconds)**



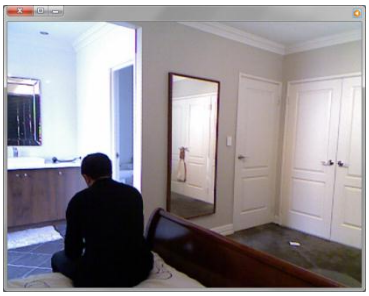
Start at point **E.**

1. Interact here for **22** seconds.
2. Move to **D** in **4** seconds, interact here for **15** seconds.
3. Move to **E** in **4** seconds, interact here for **70** seconds.
4. Move to **C** in **4** seconds, interact here for **16** seconds.
5. Move to **E** in **4** seconds, interact here for **66** seconds.
6. Move to **D** in **4** seconds, interact here for **29** seconds.

Finish ADL recording.

ADL 1 - Video 3

Exact Time: 3 minutes 59 seconds **(282 seconds)**



Start at point **E.**

1. Interact here for **22** seconds.
2. Move to **D** in **4** seconds, interact here for **17** seconds.
3. Move to **E** in **5** seconds, interact here for **80** seconds.
4. Move to **C** in **4** seconds, interact here for **21** seconds.
5. Move to **E** in **4** seconds, interact here for **81** seconds.
6. Move to **D** in **6** seconds, interact here for **38** seconds.

Finish ADL recording.

ADL 1 - Video 4

Exact Time: 5 minutes 10 seconds **(310 seconds)**

Start at point **E.**

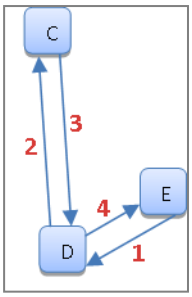
1. Interact here for **20** seconds.
2. Move to **D** in **5** seconds, interact here for **23** seconds.
3. Move to **E** in **7** seconds, interact here for **94** seconds.
4. Move to **C** in **5** seconds, interact here for **21** seconds.
5. Move to **E** in **7** seconds, interact here for **89** seconds.
6. Move to **D** in **6** seconds, interact here for **39** seconds.

Finish ADL recording.

Initial Script (modified)

Estimated time: 400 seconds.

Start at point **E.**

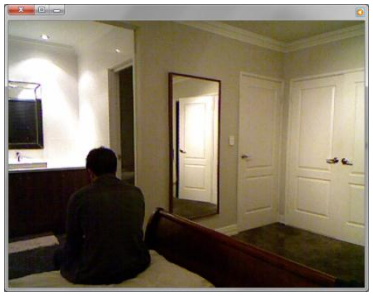


7. Interact here for **120** seconds.
8. Move to **D** in **5** seconds, interact here for **50** seconds.
9. Move to **C** in **5** seconds, interact here for **30** seconds.
10. Move to **D** in **5** seconds, interact here for **25** seconds.
11. Move to **E** in **5** seconds, interact here for **150** seconds

Finish ADL recording.

ADL 1 - Video 5

Exact Time: 4 minutes 54 seconds **(294 seconds)**



Start at point **E.**

1. Interact here for **81** seconds.
2. Move to **D** in **6** seconds, interact here for **40** seconds.
3. Move to **C** in **5** seconds, interact here for **23** seconds.
4. Move to **D** in **5** seconds, interact here for **20** seconds.
5. Move to **E** in **5** seconds, interact here for **109** seconds.

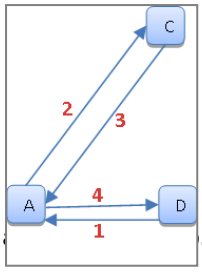
Finish ADL recording.

ADL 2

Initial Script

Estimated time: 400 seconds.

Start at point **D.**



1. Interact here for **120** seconds.
2. Move to **A** in **5** seconds, interact here for **30** seconds.

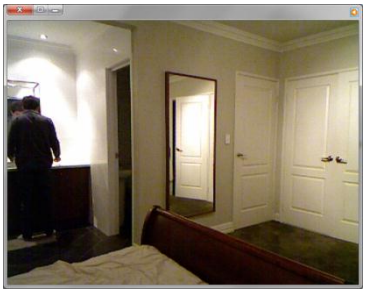
3. Move to **C** in **10** seconds, interact here for **60** seconds.
4. Move to **A** in **10** seconds, interact here for **60** seconds.
5. Move to **D** in **5** seconds, interact here for **100** seconds.

Finish ADL recording.

- Total time at point **A: 30** seconds
- Total time at point **C: 60** seconds
- Total time at point **D: 220** seconds

ADL 2 - Video 1

Exact Time: 5 minutes 23 seconds **(323 seconds)**



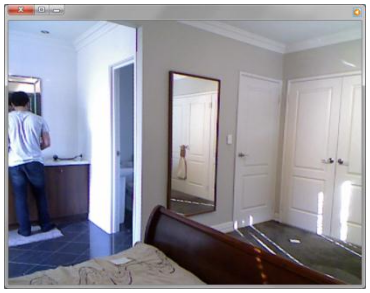
Start at point **D.**

1. Interact here for **87** seconds.
2. Move to **A** in **5** seconds, interact here for **26** seconds.
3. Move to **C** in **6** seconds, interact here for **51** seconds.
4. Move to **A** in **7** seconds, interact here for **51** seconds.
5. Move to **D** in **4** seconds, interact here for **86** seconds.

Finish ADL recording.

ADL 2 - Video 2

Exact Time: 5 minutes 58 seconds **(358 seconds)**



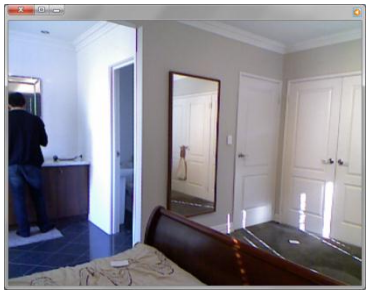
Start at point **D.**

1. Interact here for **108** seconds.
2. Move to **A** in **6** seconds, interact here for **28** seconds.
3. Move to **C** in **7** seconds, interact here for **54** seconds.
4. Move to **A** in **8** seconds, interact here for **56** seconds.
5. Move to **D** in **6** seconds, interact here for **85** seconds.

Finish ADL recording.

ADL 2 - Video 3

Exact Time: 5 minutes 13 seconds **(313 seconds)**



Start at point **D.**

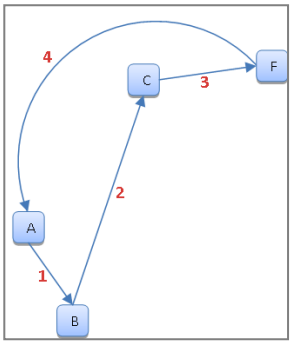
1. Interact here for **82** seconds.
2. Move to **A** in **5** seconds, interact here for **20** seconds.
3. Move to **C** in **6** seconds, interact here for **43** seconds.
4. Move to **A** in **7** seconds, interact here for **55** seconds.
5. Move to **D** in **6** seconds, interact here for **89** seconds.

Finish ADL recording.

ADL 3

Initial Script

Estimated time: 390 seconds.



Start at point **A.**

3. Interact here for **60** seconds.
4. Move to **B** in **5** seconds, interact here for **150** seconds.
5. Move to **C** in **10** seconds, interact here for **60** seconds.
6. Move to **F** in **5** seconds, interact here for **60** seconds.
7. Move to **A** in **10** seconds, interact here for **30** seconds.

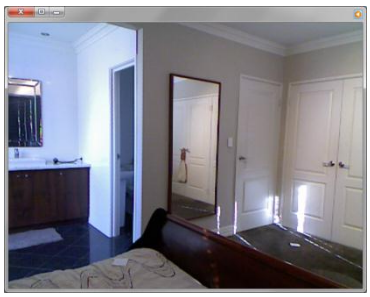
Finish ADL recording.

- Total time at point **A: 90** seconds
- Total time at point **B: 150** seconds
- Total time at point **C: 60** seconds
- Total time at point **F: 60** seconds

-

ADL 3 - Video 1

Exact Time: 5 minutes 41 seconds **(341 seconds)**



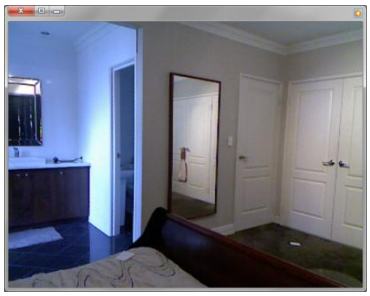
Start at point **A.**

1. Interact here for **54** seconds.
2. Move to **B** in **3** seconds, interact here for **139** seconds.
3. Move to **C** in **7** seconds, interact here for **48** seconds.
4. Move to **F** in **4** seconds, interact here for **52** seconds.
5. Move to **A** in **5** seconds, interact here for **29** seconds.

Finish ADL recording.

ADL 3 - Video 2

Exact Time: 5 minutes 47 seconds **(347 seconds)**



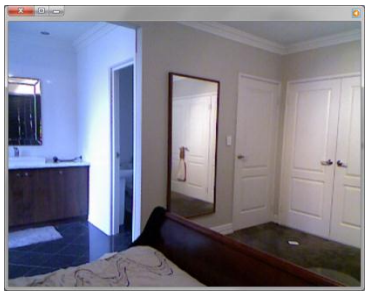
Start at point **A.**

1. Interact here for **50** seconds.
2. Move to **B** in **3** seconds, interact here for **136** seconds.
3. Move to **C** in **4** seconds, interact here for **56** seconds.
4. Move to **F** in **3** seconds, interact here for **60** seconds.
5. Move to **A** in **6** seconds, interact here for **29** seconds.

Finish ADL recording.

ADL 3 - Video 3

Exact Time: 5 minutes 12 seconds **(312 seconds)**



Start at point **A.**

1. Interact here for **43** seconds.
2. Move to **B** in **4** seconds, interact here for **111** seconds.
3. Move to **C** in **6** seconds, interact here for **53** seconds.
4. Move to **F** in **4** seconds, interact here for **55** seconds.
5. Move to **A** in **5** seconds, interact here for **31** seconds.

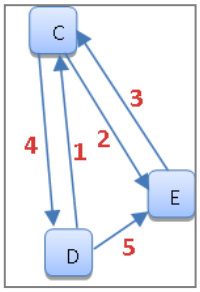
Finish ADL recording.

ADL 4

Initial Script

Estimated time: 405 seconds.

Start at point **D.**



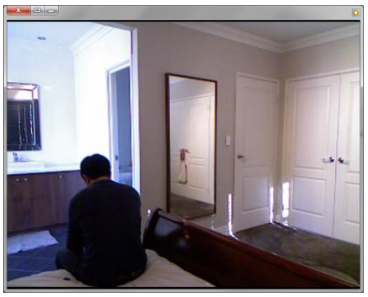
1. Interact here for **120** seconds.
2. Move to **C** in **5** seconds, interact here for **60** seconds.
3. Move to **E** in **5** seconds, interact here for 3**0** seconds.
4. Move to **C** in **5** seconds, interact here for **30** seconds.
5. Move to **D** in **5** seconds, interact here for **110** seconds.
6. Move to **E** in **5** seconds, interact here for **30** seconds.

Finish ADL recording.

- Total time at point **C: 90** seconds
- Total time at point **D: 230** seconds
- Total time at point **E: 60** seconds

ADL 4 - Video 1

Exact Time: 4 minutes 57 seconds **(297 seconds)**



Start at point **D.**

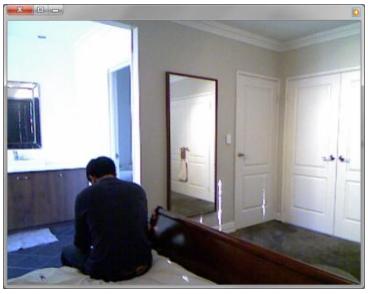
1. Interact here for **70** seconds.
2. Move to **C** in **6** seconds, interact here for **47** seconds.
3. Move to **E** in **6** seconds, interact here for **20** seconds.
4. Move to **C** in **5** seconds, interact here for **24** seconds.
5. Move to **D** in **5** seconds, interact here for **85** seconds.
6. Move to **E** in **6** seconds, interact here for **23** seconds.

Finish ADL recording.

ADL 4 - Video 2

Exact Time: 5 minutes 9 seconds **(309 seconds)**

Start at point **D.**



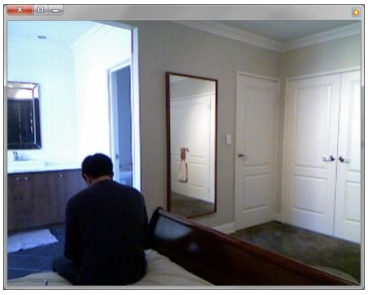
1. Interact here for **87** seconds.
2. Move to **C** in **5** seconds, interact here for **46** seconds.
3. Move to **E** in **6** seconds, interact here for **22** seconds.
4. Move to **C** in **6** seconds, interact here for **19** seconds.
5. Move to **D** in **4** seconds, interact here for **82** seconds.
6. Move to **E** in **5** seconds, interact here for **23** seconds.

Finish ADL recording.

ADL 4 - Video 3

Exact Time: 4 minutes 20 seconds **(260 seconds)**

Start at point **D.**



1. Interact here for **72** seconds.
2. Move to **C** in **5** seconds, interact here for **37** seconds.
3. Move to **E** in **5** seconds, interact here for **16** seconds.
4. Move to **C** in **4** seconds, interact here for **19** seconds.
5. Move to **D** in **3** seconds, interact here for **71** seconds.
6. Move to **E** in **6** seconds, interact here for **22** seconds.

Finish ADL recording.

Appendix B: Results for Classifying Individual Video Data

The tests in this section use the following videos as training input except for the video being tested in each case:

ADL 1: Video 1, Video 2, Video 3
ADL 2: Video 1, Video 2, Video 3
ADL 3: Video 1, Video 2, Video 3
ADL 4: Video 1, Video 2, Video 3

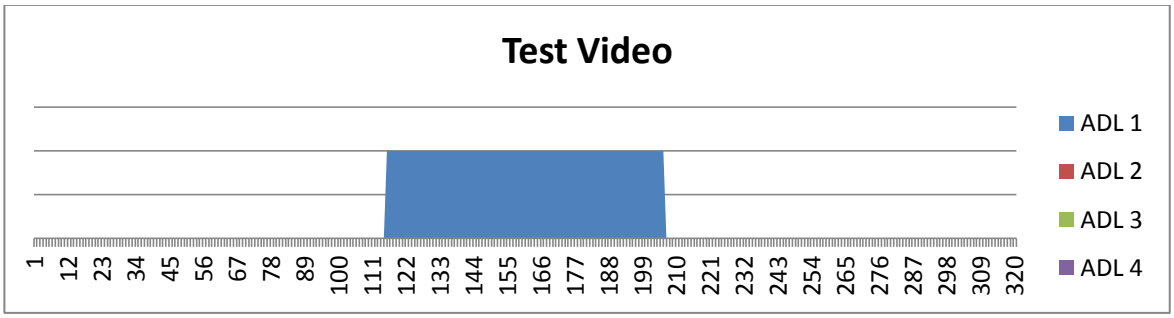
The resulting spatio-temporal graphs for each experiment represent the activities that were recognised at certain times in a test video. The x-axis refers to time in seconds.

ADL 1

Test Video: ADL 1 - Video 1

Training Data:

ADL 1:		Video 2	Video 3
ADL 2:	Video 1	Video 2	Video 3
ADL 3:	Video 1	Video 2	Video 3
ADL 4:	Video 1	Video 2	Video 3

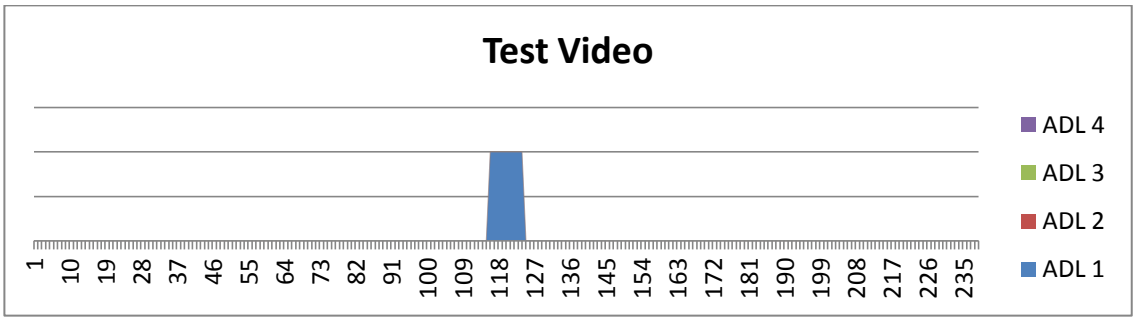


Ground Truth: ADL 1

Test Video: ADL 1 - Video 2

Training Data:

ADL 1:	Video 1		Video 3
ADL 2:	Video 1	Video 2	Video 3
ADL 3:	Video 1	Video 2	Video 3
ADL 4:	Video 1	Video 2	Video 3

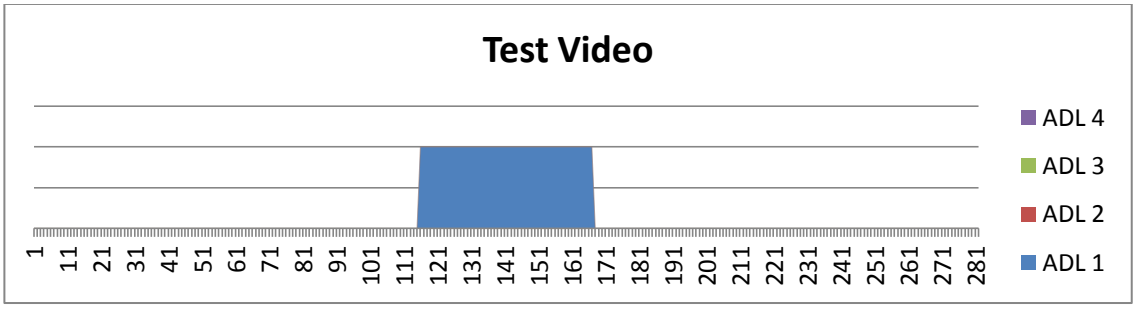


Ground Truth: ADL 1

Test Video: ADL 1 - Video 3

Training Data:

ADL 1:	Video 1	Video 2	
ADL 2:	Video 1	Video 2	Video 3
ADL 3:	Video 1	Video 2	Video 3
ADL 4:	Video 1	Video 2	Video 3



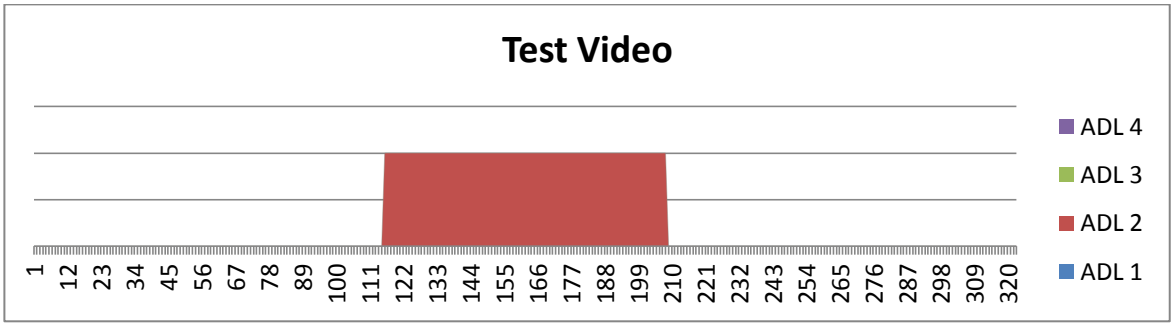
Ground Truth: ADL 1

ADL 2

Test Video: ADL 2 - Video 1

Training Data:

ADL 1:	Video 1	Video 2	Video 3
ADL 2:		Video 2	Video 3
ADL 3:	Video 1	Video 2	Video 3
ADL 4:	Video 1	Video 2	Video 3

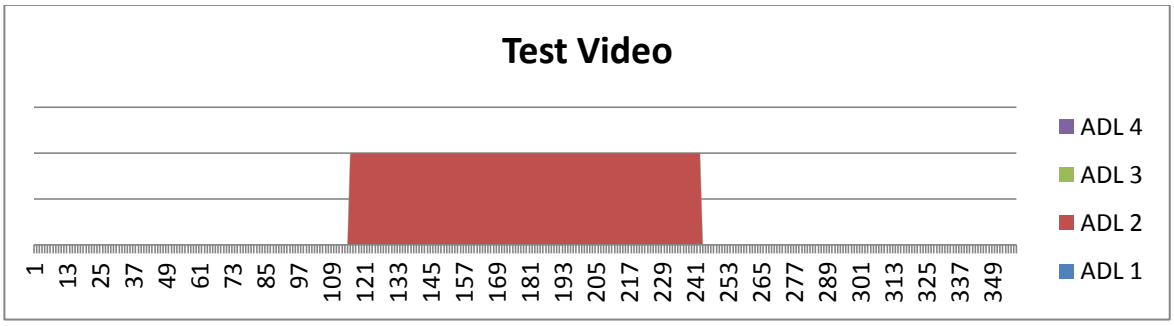


Ground Truth: ADL 2

Test Video: ADL 2 - Video 2

Training Data:

ADL 1:	Video 1	Video 2	Video 3
ADL 2:	Video 1		Video 3
ADL 3:	Video 1	Video 2	Video 3
ADL 4:	Video 1	Video 2	Video 3

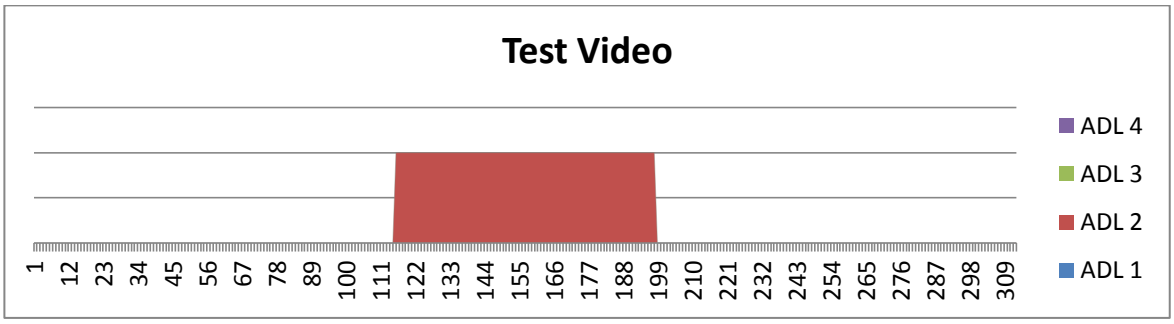


Ground Truth: ADL 2

Test Video: ADL 2 - Video 3

Training Data:

ADL 1:	Video 1	Video 2	Video 3
ADL 2:	Video 1	Video 2	
ADL 3:	Video 1	Video 2	Video 3
ADL 4:	Video 1	Video 2	Video 3



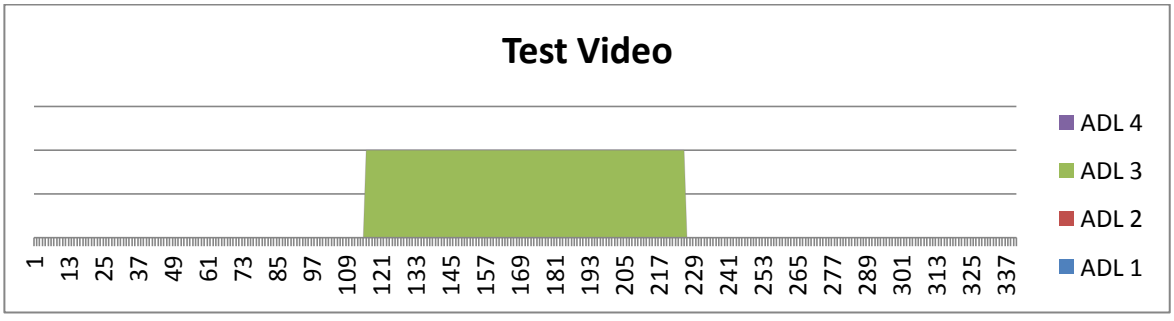
Ground Truth: ADL 2

ADL 3

Test Video: ADL 3 - Video 1

Training Data:

ADL 1:	Video 1	Video 2	Video 3
ADL 2:	Video 1	Video 2	Video 3
ADL 3:		Video 2	Video 3
ADL 4:	Video 1	Video 2	Video 3

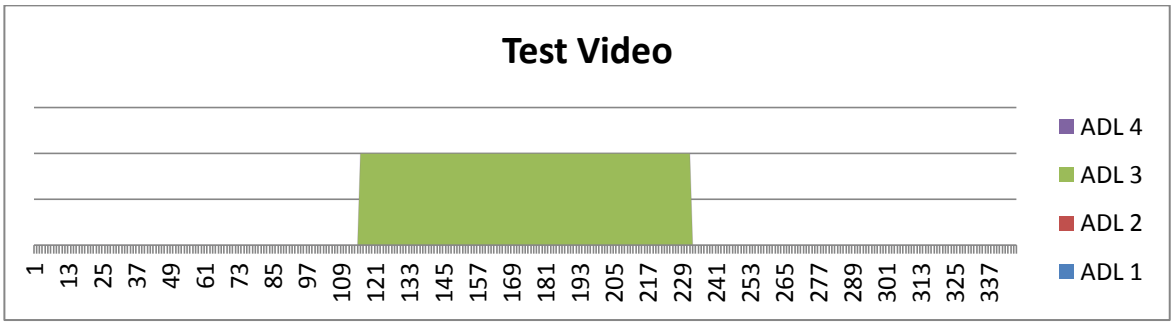


Ground Truth: ADL 3

Test Video: ADL 3 - Video 2

Training Data:

ADL 1:	Video 1	Video 2	Video 3
ADL 2:	Video 1	Video 2	Video 3
ADL 3:	Video 1		Video 3
ADL 4:	Video 1	Video 2	Video 3

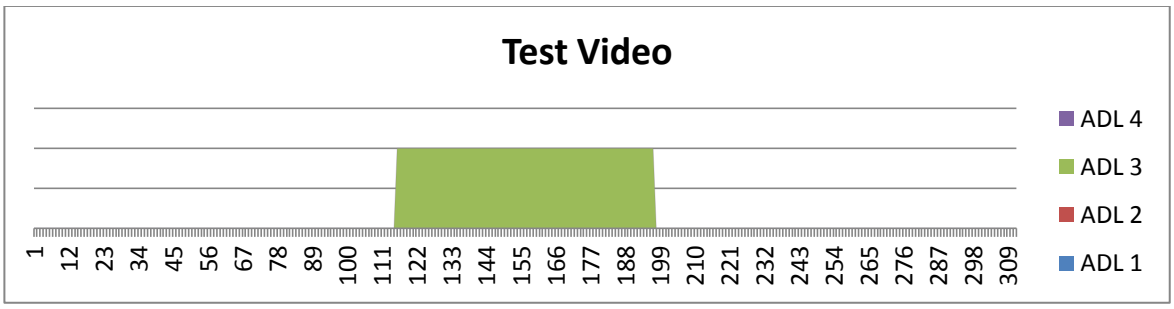


Ground Truth: ADL 3

Test Video: ADL 3 - Video 3

Training Data:

ADL 1:	Video 1	Video 2	Video 3
ADL 2:	Video 1	Video 2	Video 3
ADL 3:	Video 1	Video 2	
ADL 4:	Video 1	Video 2	Video 3



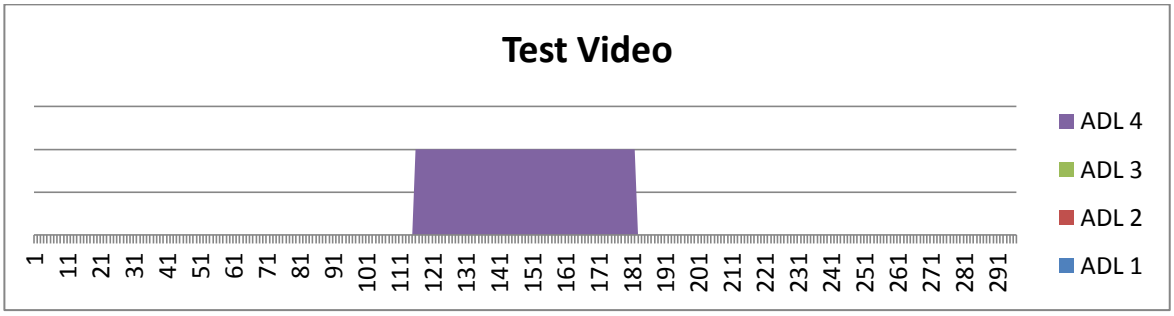
Ground Truth: ADL 3

ADL 4

Test Video: ADL 4 - Video 1

Training Data:

ADL 1:	Video 1	Video 2	Video 3
ADL 2:	Video 1	Video 2	Video 3
ADL 3:	Video 1	Video 2	Video 3
ADL 4:		Video 2	Video 3

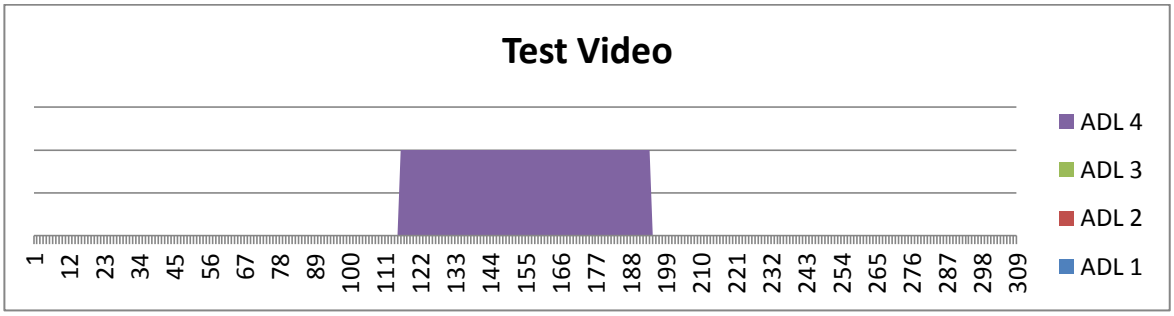


Ground Truth: ADL 4

Test Video: ADL 4 - Video 2

Training Data:

ADL 1:	Video 1	Video 2	Video 3
ADL 2:	Video 1	Video 2	Video 3
ADL 3:	Video 1	Video 2	Video 3
ADL 4:	Video 1		Video 3

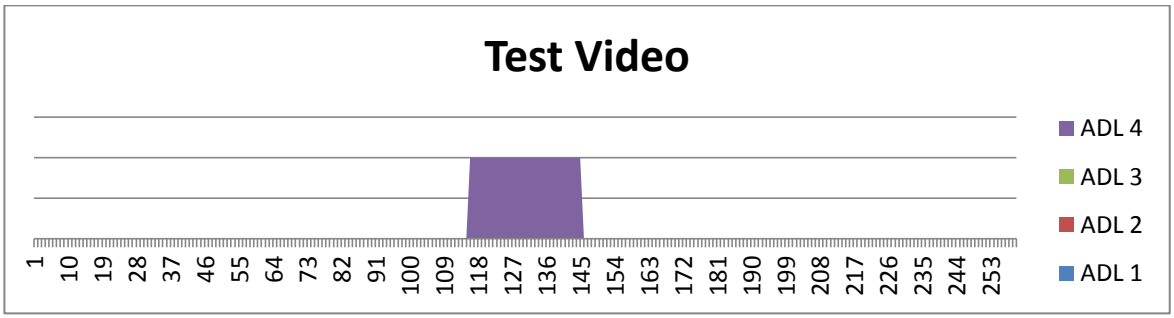


Ground Truth: ADL 4

Test Video: ADL 4 - Video 3

Training Data:

ADL 1:	Video 1	Video 2	Video 3
ADL 2:	Video 1	Video 2	Video 3
ADL 3:	Video 1	Video 2	Video 3
ADL 4:	Video 1	Video 2	



Ground Truth: ADL 4

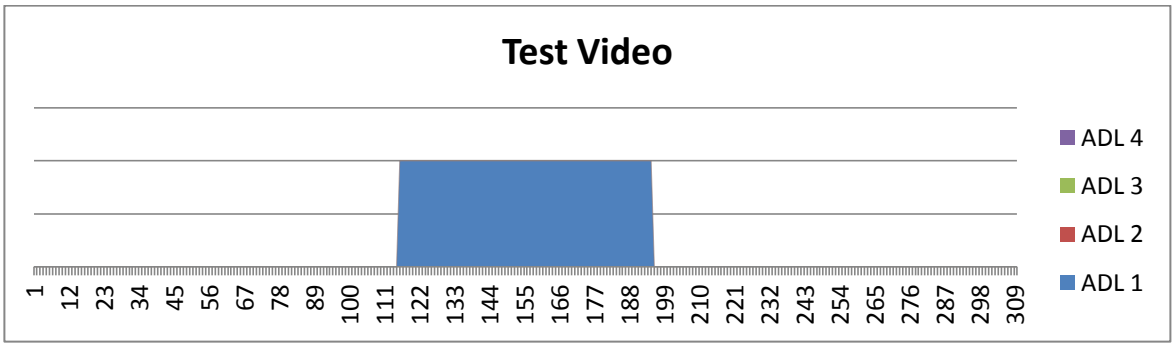
The following video test uses the poor lighting recording of ADL 1:

ADL 1 – Video 4

Test Video: ADL 4 - Video 1

Training Data:

ADL 1:	Video 1	Video 2	Video 3
ADL 2:	Video 1	Video 2	Video 3
ADL 3:	Video 1	Video 2	Video 3
ADL 4:	Video 1	Video 2	Video 3



Ground Truth: ADL 1

Appendix C: Results for Classifying Combined Video Data

The following test videos are created by linking together multiple ADLs into a combined test video. Note that the videos used as test data are not included into training the system. The dataset used for these tests are:

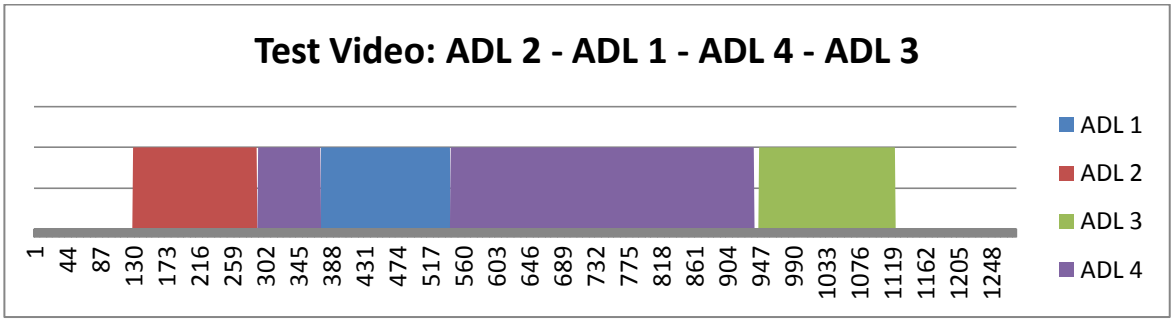
ADL 1: Video 1, Video 2, Video 3
ADL 2: Video 1, Video 2, Video 3
ADL 3: Video 1, Video 2, Video 3
ADL 4: Video 1, Video 2, Video 3

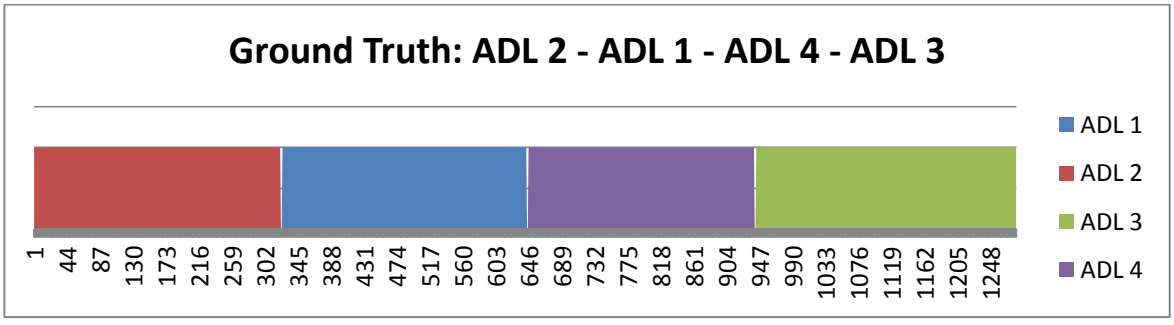
Combined Test 1

Test Video: ADL 2 - ADL 1 - ADL 4 - ADL 3

Training Data:

ADL 1:	Video 2	Video 3
ADL 2:	Video 2	Video 3
ADL 3:	Video 2	Video 3
ADL 4:	Video 2	Video 3



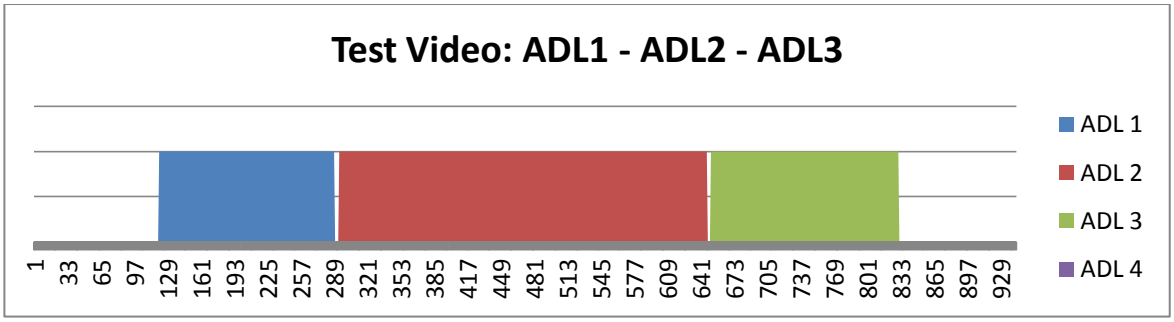


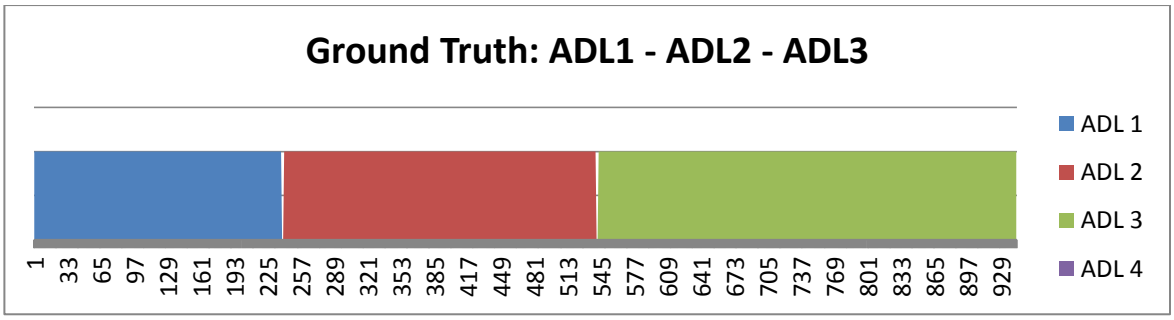
Combined Test 2

Test Video: ADL 1 - ADL 2 - ADL 3

Training Data:

ADL 1:	Video 1	Video 3
ADL 2:	Video 1	Video 3
ADL 3:	Video 1	Video 3
ADL 4:	Video 1	Video 3



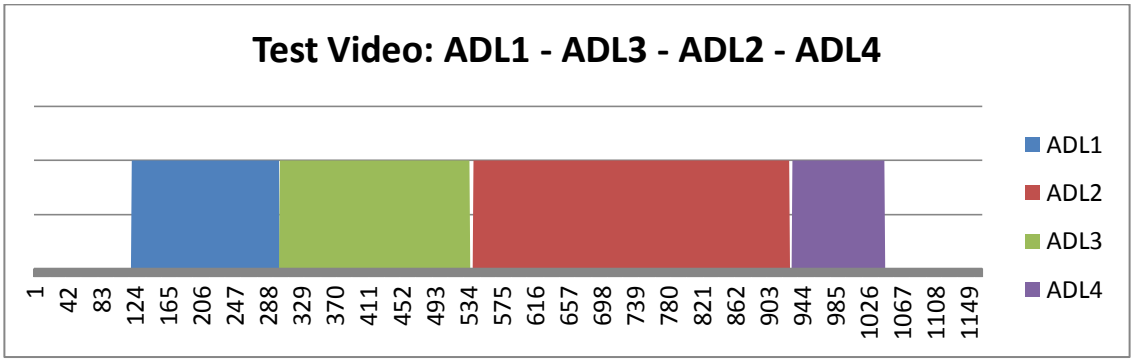


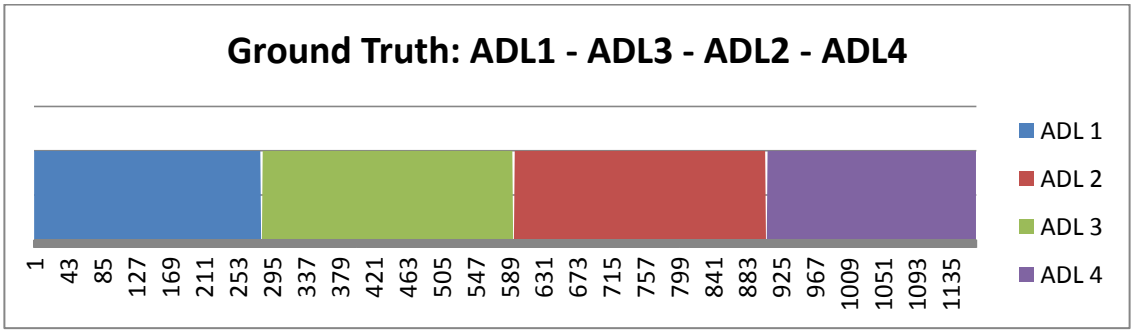
Combined Test 3

Test Video: ADL 1 - ADL 3 - ADL 2 - ADL 4

Training Data:

ADL 1:	Video 1	Video 2
ADL 2:	Video 1	Video 2
ADL 3:	Video 1	Video 2
ADL 4:	Video 1	Video 2





Appendix D: Results for Classifying ADL Variation Video Data

In the following test, an ADL variation video is tested against standard ADLs from its class. The ADL Variation video is not included into the training phase for this test.

The dataset used for these tests include:

ADL 1: Video 1, Video 2, Video 3, Video 5
ADL 2: Video 1, Video 2, Video 3
ADL 3: Video 1, Video 2, Video 3
ADL 4: Video 1, Video 2, Video 3

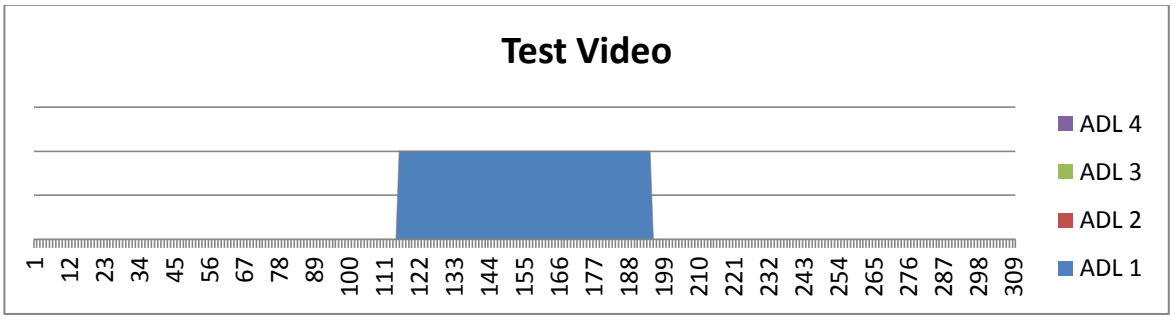
Note that *ADL 1: Video 4* is an ADL variation recording.

ADL 1 Variation Test

Test Video: ADL 1 - Video 5

Training Data:

ADL 1:	Video 1	Video 2	Video 3
ADL 2:	Video 1	Video 2	Video 3
ADL 3:	Video 1	Video 2	Video 3
ADL 4:	Video 1	Video 2	Video 3



Ground Truth: ADL 1

Appendix E: Results for Abnormality Detection

In each of the test cases presented here, video data from an ADL is left out of the training phase and then tested upon to evaluate the abnormality detection.

The dataset used for these tests include:

ADL 1: Video 1, Video 2, Video 3
ADL 2: Video 1, Video 2, Video 3
ADL 3: Video 1, Video 2, Video 3
ADL 4: Video 1, Video 2, Video 3

The video being tested in each of these cases is a combined video containing a sequence of each ADL. The threshold value being used is in this experiment is automatically generated as one standard deviation from the mean.

ADL 1 Abnormality Test

Test Video: ADL 1 - ADL 3 - ADL 2 - ADL 4

Training Data:

ADL 2:	Video 1	Video 2	Video 3
ADL 3:	Video 1	Video 2	Video 3
ADL 4:	Video 1	Video 2	Video 3

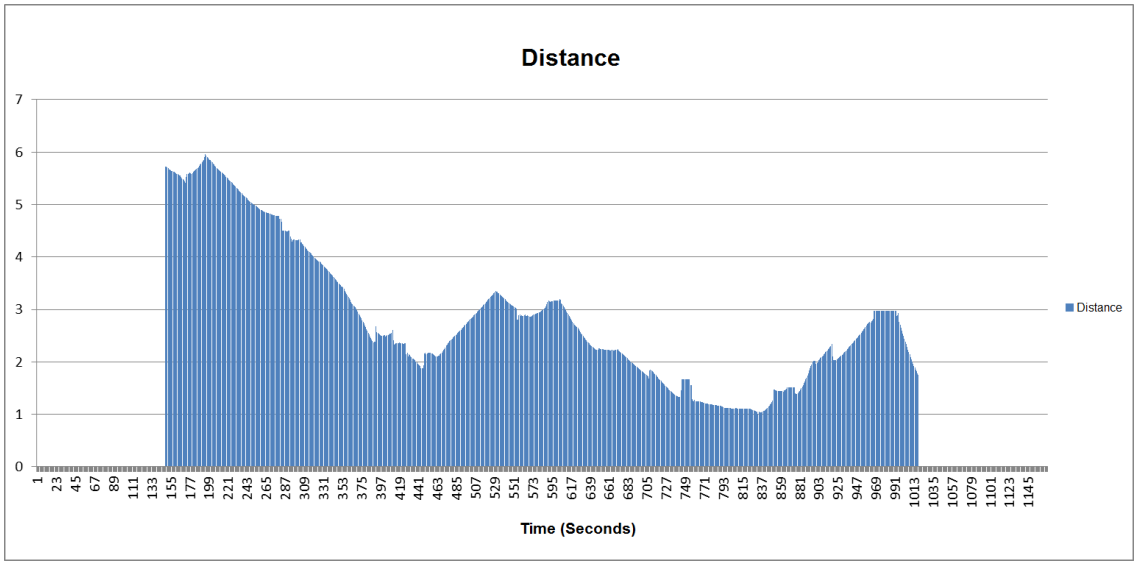
Generated Threshold Values:

Mean: 2.44971540267795

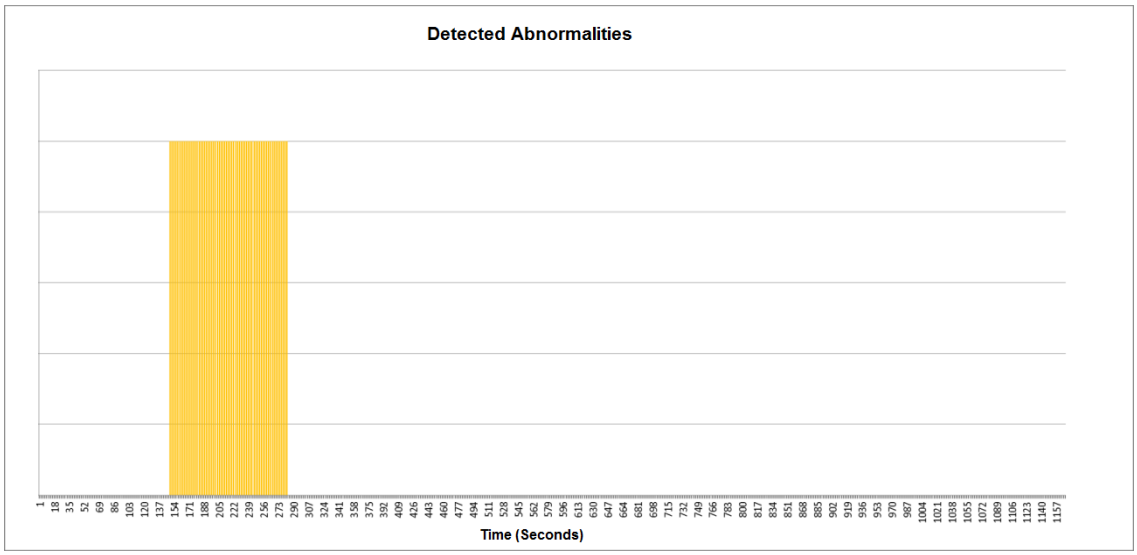
SD: 2.0789585370367

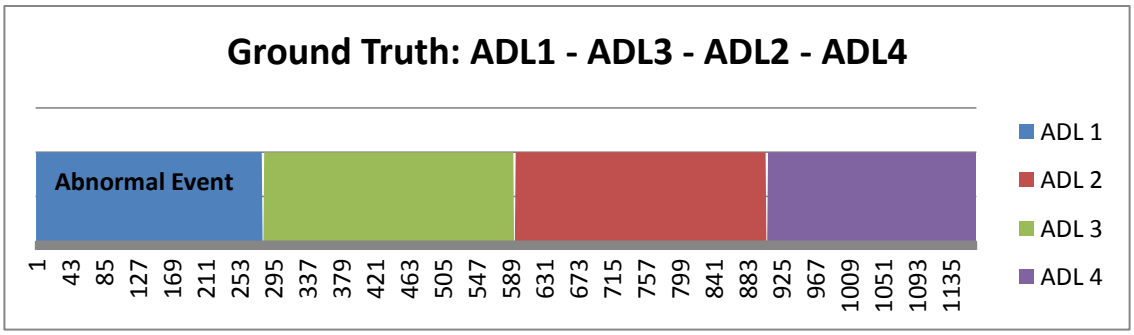
Threshold: 4.52867393971465

Distance Values over test video:



Detected Abnormalities within test video:





Ground Truth: ADL 1 is the abnormal event.

ADL 2 Abnormality Test

Test Video: ADL 1 - ADL 3 - ADL 2 - ADL 4

Training Data:

ADL 1:	Video 1	Video 2	Video 3
ADL 3:	Video 1	Video 2	Video 3
ADL 4:	Video 1	Video 2	Video 3

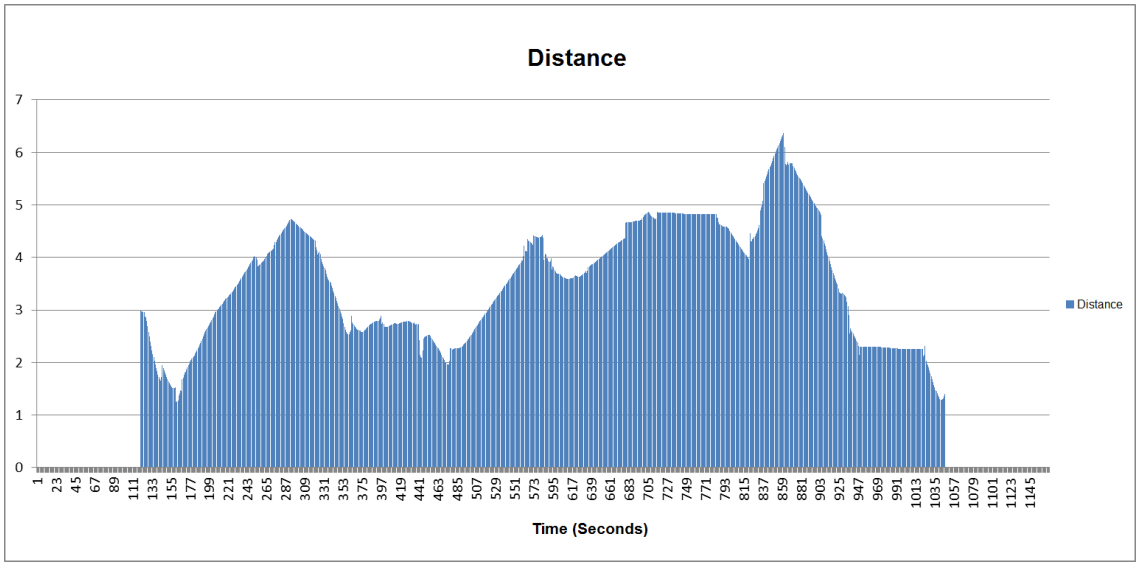
Generated Threshold Values:

Mean: 4.16378221624737

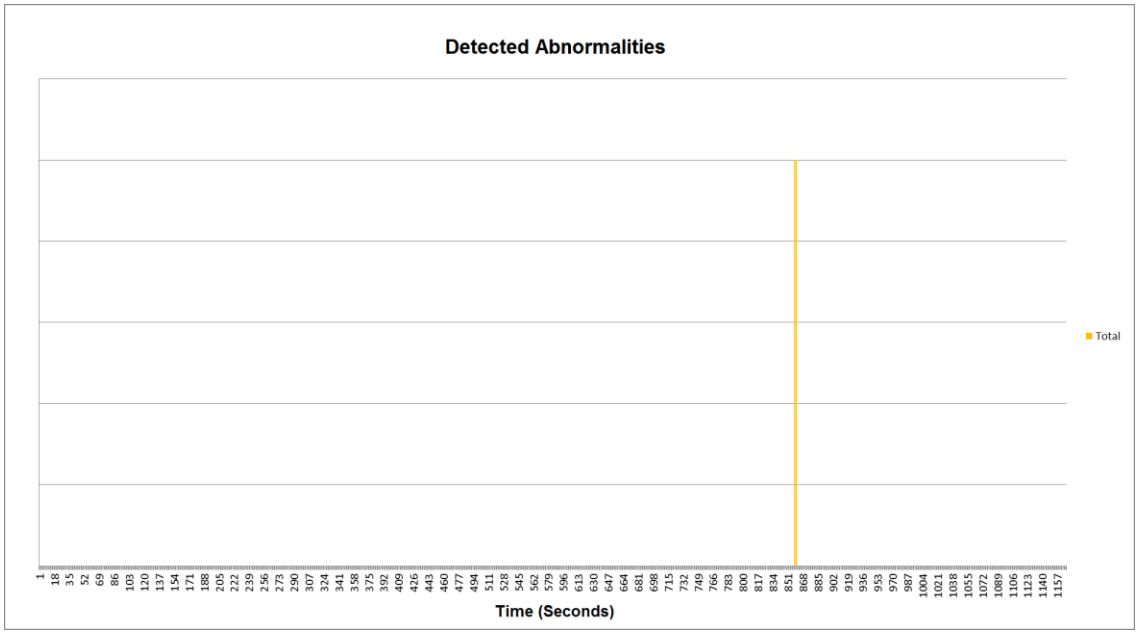
SD: 2.06376548537808

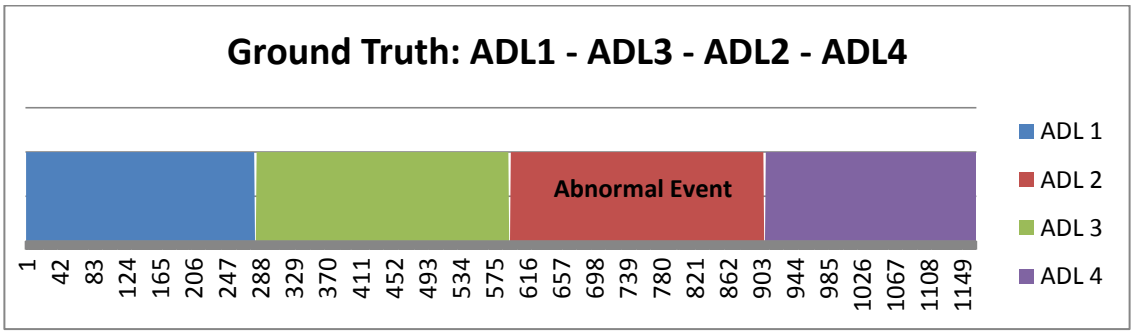
Threshold: 6.22754770162545

Distance Values over test video:



Detected Abnormalities within test video:





Ground Truth: ADL 2 is the abnormal event.

ADL 3 Abnormality Test

Test Video: ADL 1 - ADL 3 - ADL 2 - ADL 4

Training Data:

ADL 1:	Video 1	Video 2	Video 3
ADL 2:	Video 1	Video 2	Video 3
ADL 4:	Video 1	Video 2	Video 3

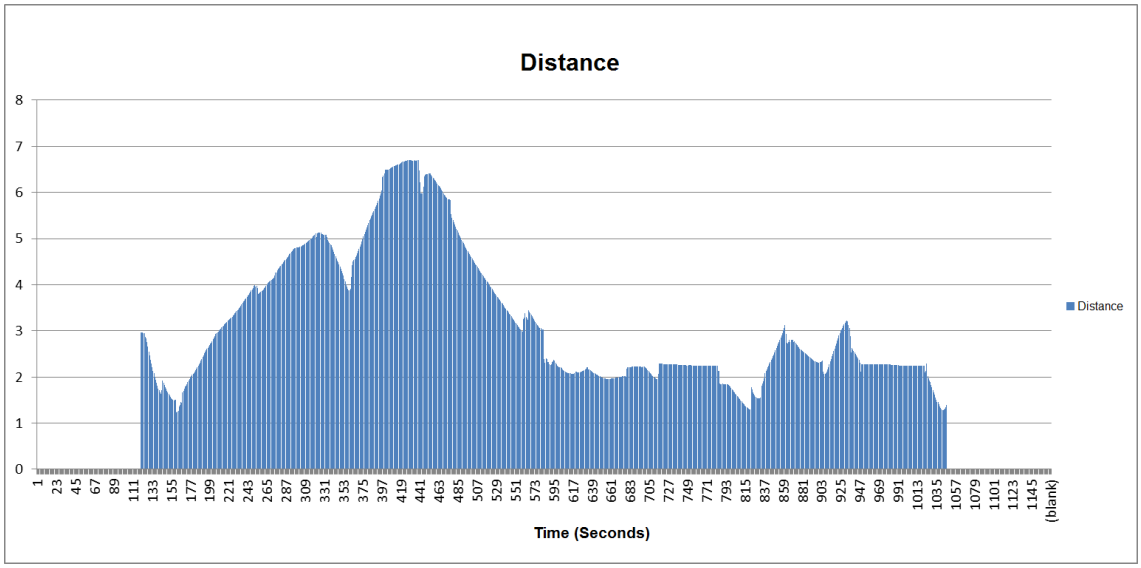
Generated Threshold Values:

Mean: 4.02850066613328

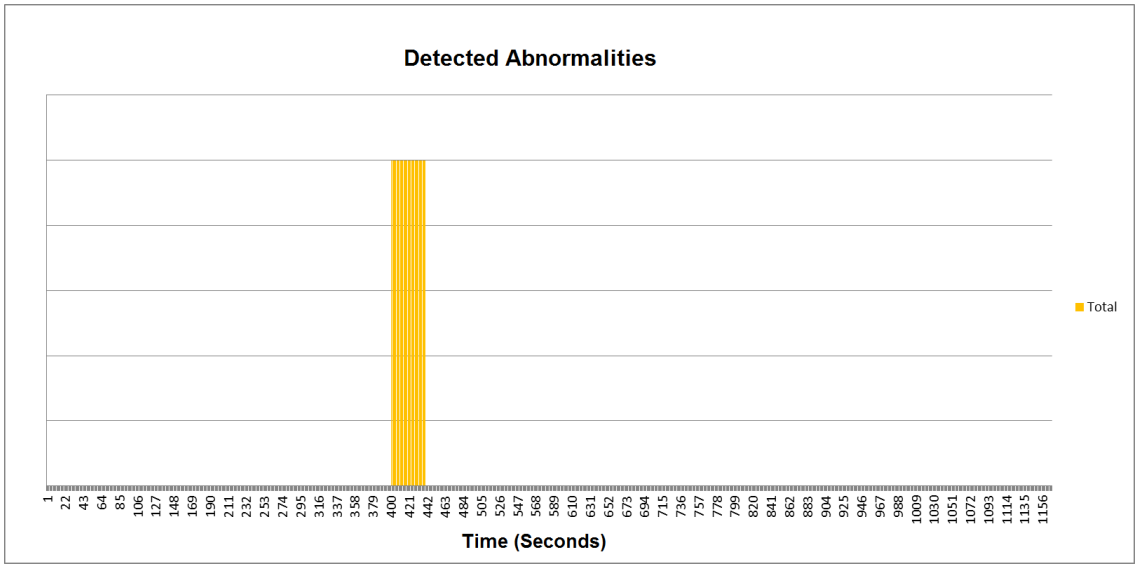
SD: 2.38720267475533

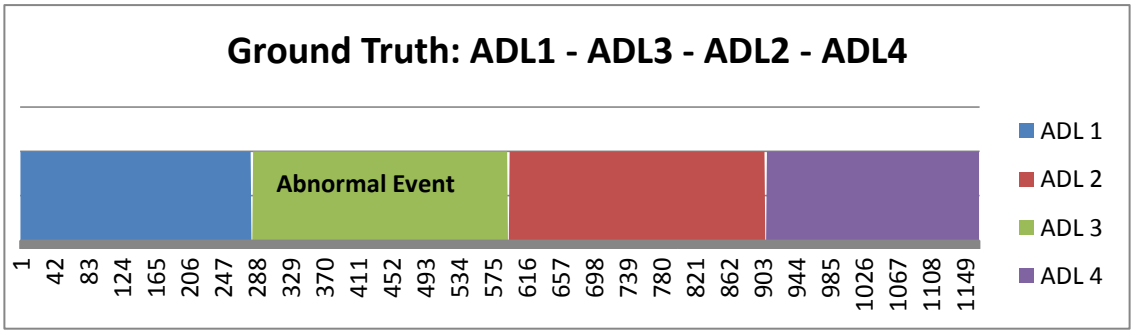
Threshold: 6.41570334088861

Distance Values over test video:



Detected Abnormalities within test video:





Ground Truth: ADL 3 is the abnormal event.

ADL 4 Abnormality Test

Test Video: ADL 1 - ADL 3 - ADL 2 - ADL 4

Training Data:

ADL 1:	Video 1	Video 2	Video 3
ADL 2:	Video 1	Video 2	Video 3
ADL 3:	Video 1	Video 2	Video 3

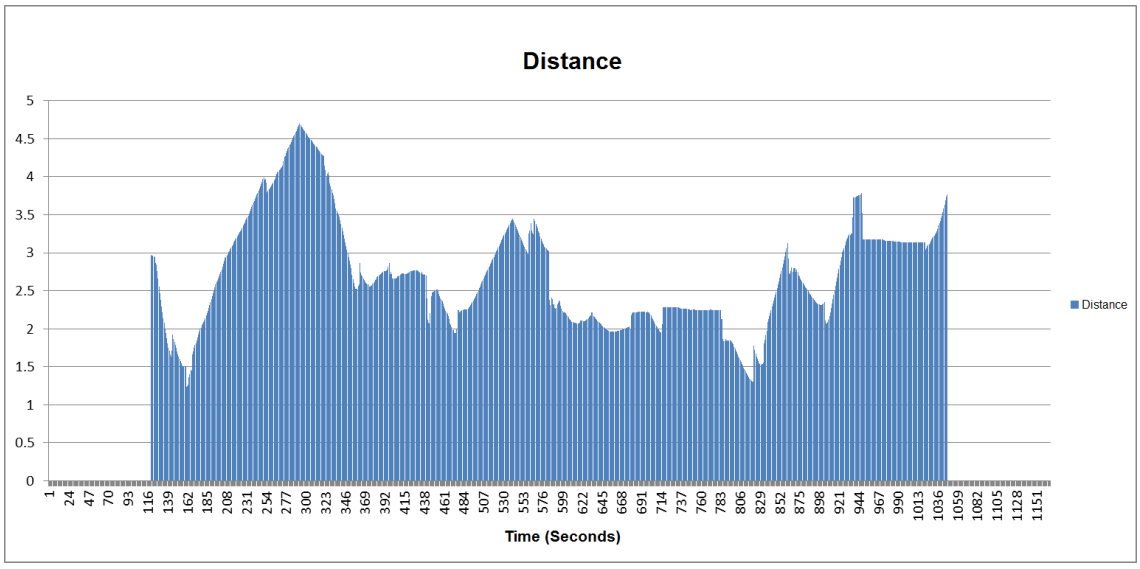
Generated Threshold Values:

Mean: 2.67673152009117

SD: 2.39376666609464

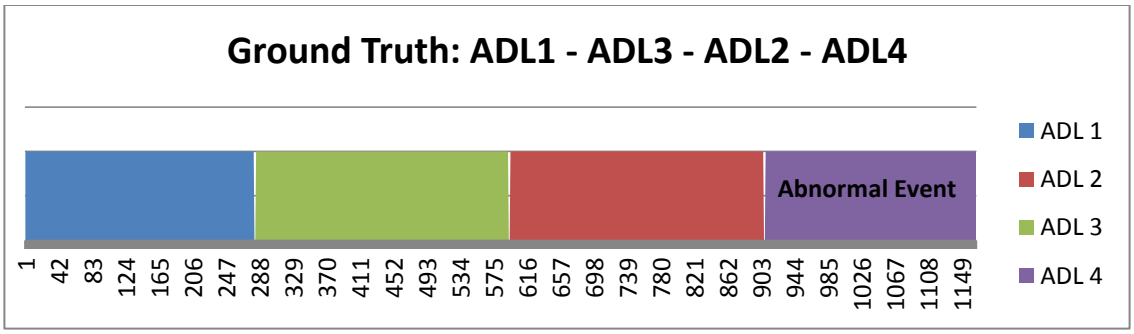
Threshold: 5.07049818618581

Distance Values over test video:



Detected Abnormalities within test video:

No Detected Abnormalities.



Ground Truth: ADL 4 is the abnormal event.

Appendix F: Results for Feature Comparison

In the following tests, the algorithm used to create the feature vector table is modified to test each feature individually. Note that the following tests use Euclidean distance as the only form of distance measure. The dataset used for these tests include:

ADL 1: Video 1, Video 2, Video 3
ADL 2: Video 1, Video 2, Video 3
ADL 3: Video 1, Video 2, Video 3
ADL 4: Video 1, Video 2, Video 3

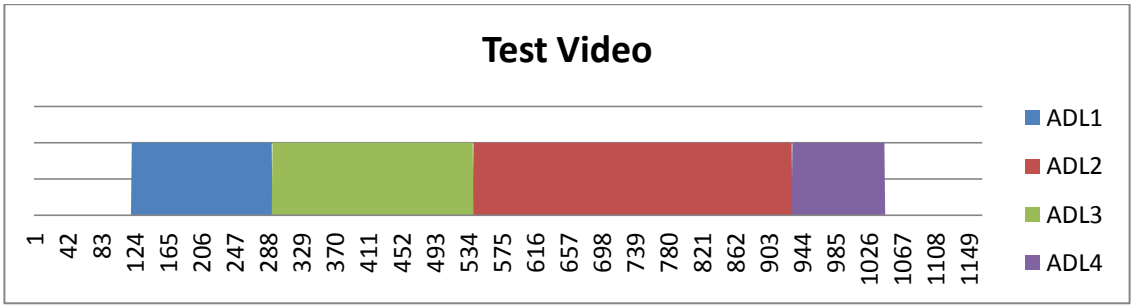
The video being tested in each of these cases is a combined video containing a sequence of each ADL.

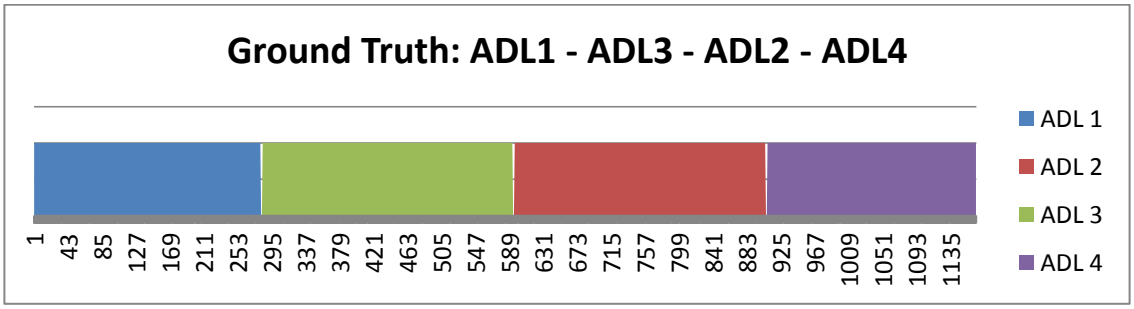
Depth, Variance, Busy Fraction, Aspect Ratio, Form Factor

Test Video: ADL 1 - ADL 3 - ADL 2 - ADL 4

Training Data:

ADL 1:	Video 1	Video 2
ADL 2:	Video 1	Video 2
ADL 3:	Video 1	Video 2
ADL 4:	Video 1	Video 2



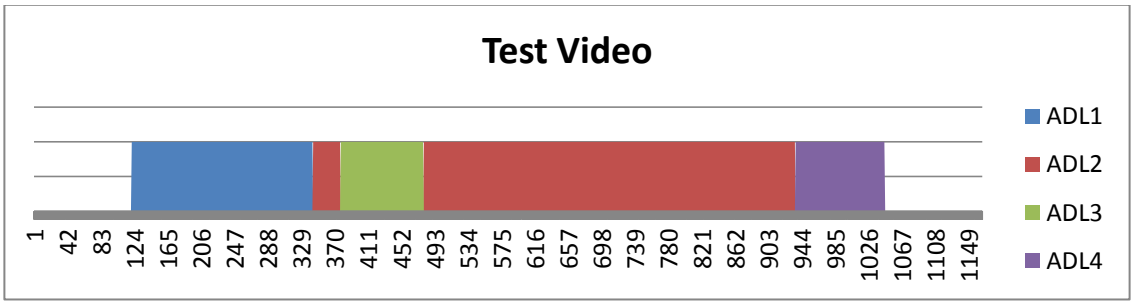


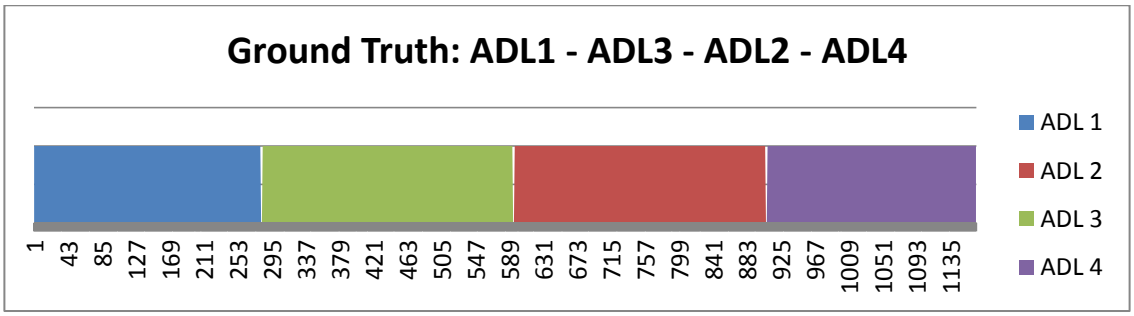
Depth Only

Test Video: ADL 1 - ADL 3 - ADL 2 - ADL 4

Training Data:

ADL 1:	Video 1	Video 2
ADL 2:	Video 1	Video 2
ADL 3:	Video 1	Video 2
ADL 4:	Video 1	Video 2



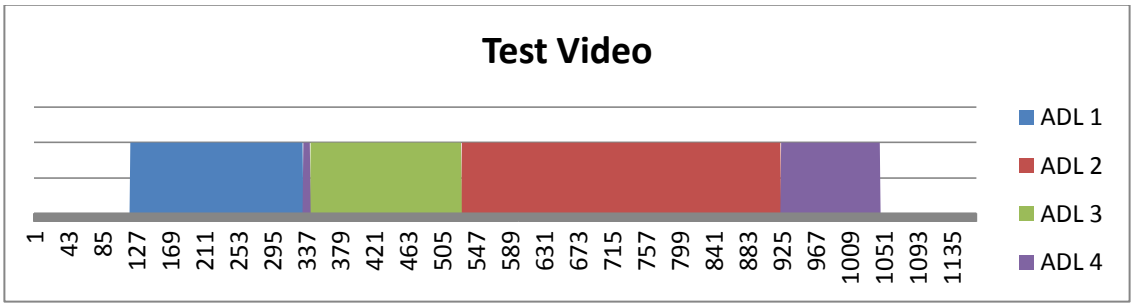


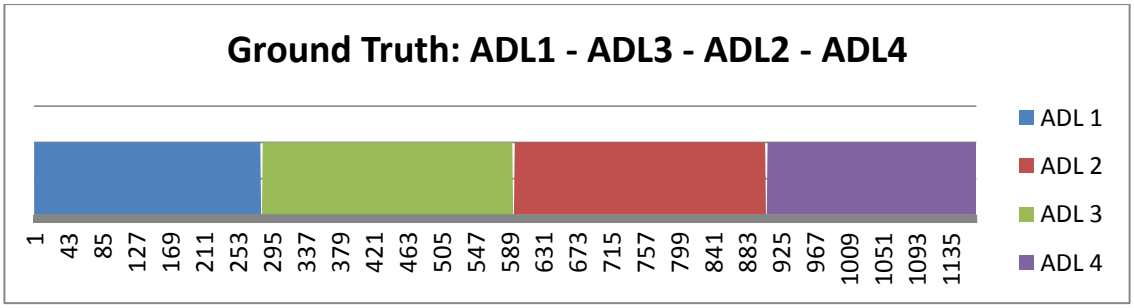
Variance Only

Test Video: ADL 1 - ADL 3 - ADL 2 - ADL 4

Training Data:

ADL 1:	Video 1	Video 2
ADL 2:	Video 1	Video 2
ADL 3:	Video 1	Video 2
ADL 4:	Video 1	Video 2



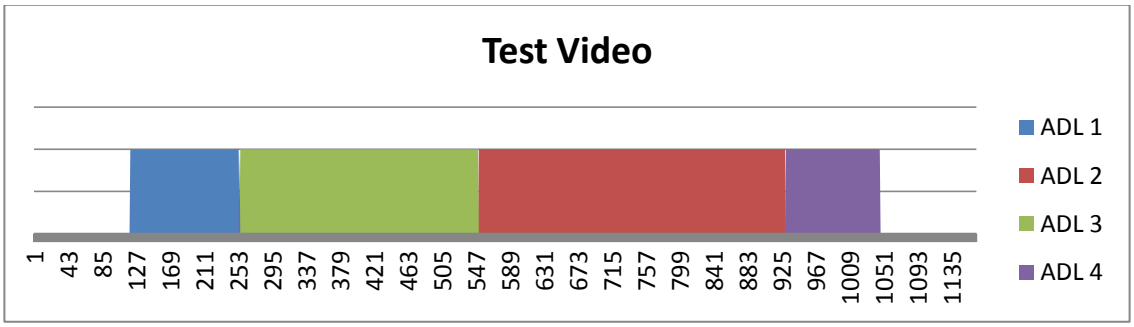


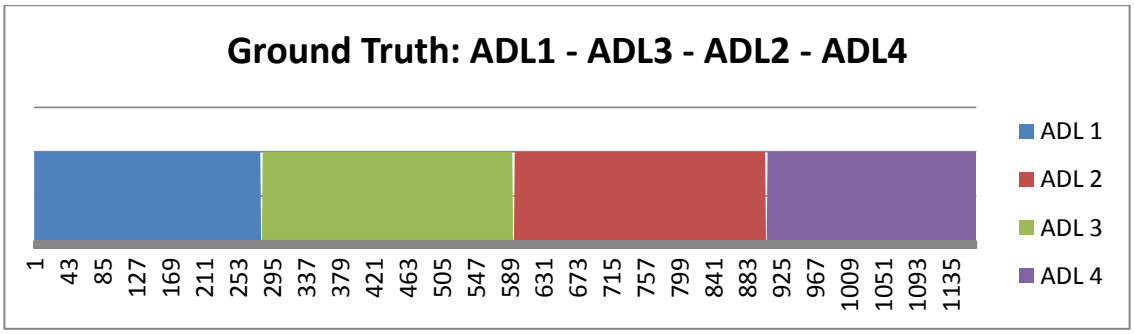
Busy Fraction Only

Test Video: ADL 1 - ADL 3 - ADL 2 - ADL 4

Training Data:

ADL 1:	Video 1	Video 2
ADL 2:	Video 1	Video 2
ADL 3:	Video 1	Video 2
ADL 4:	Video 1	Video 2



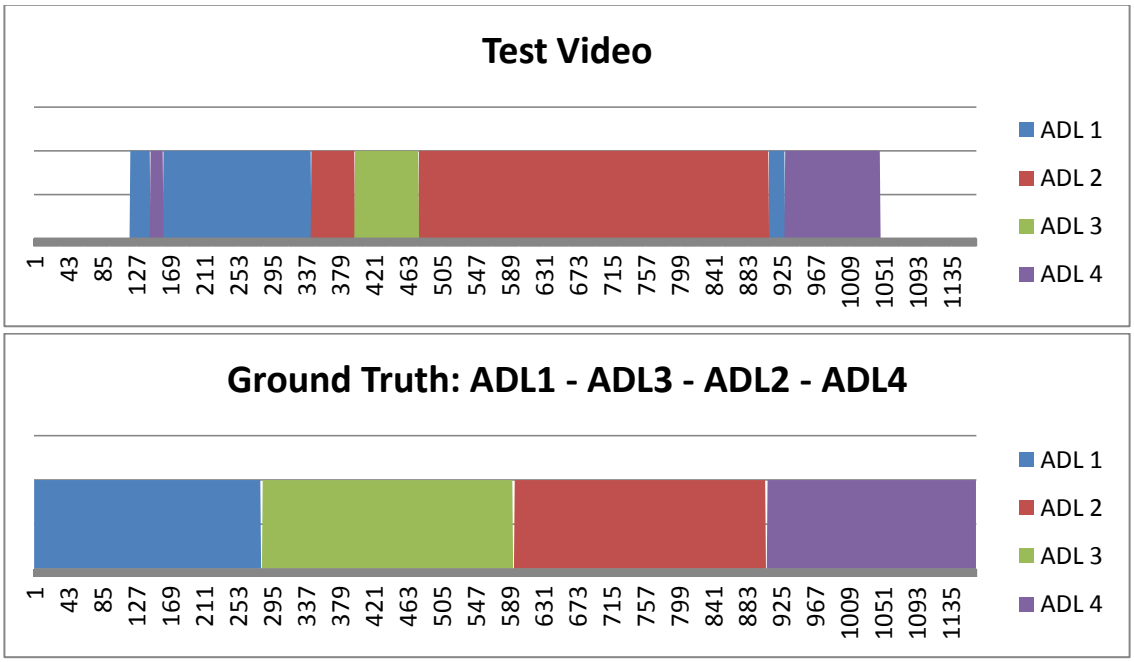


Aspect Ratio Only

Test Video: ADL 1 - ADL 3 - ADL 2 - ADL 4

Training Data:

ADL 1:	Video 1	Video 2
ADL 2:	Video 1	Video 2
ADL 3:	Video 1	Video 2
ADL 4:	Video 1	Video 2

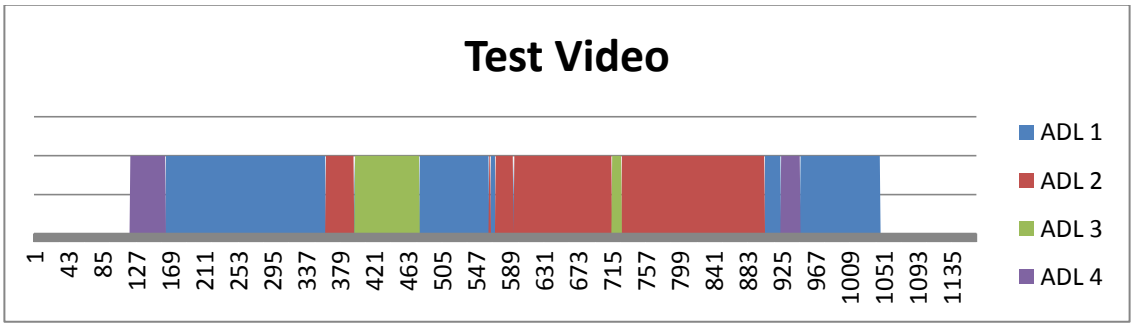


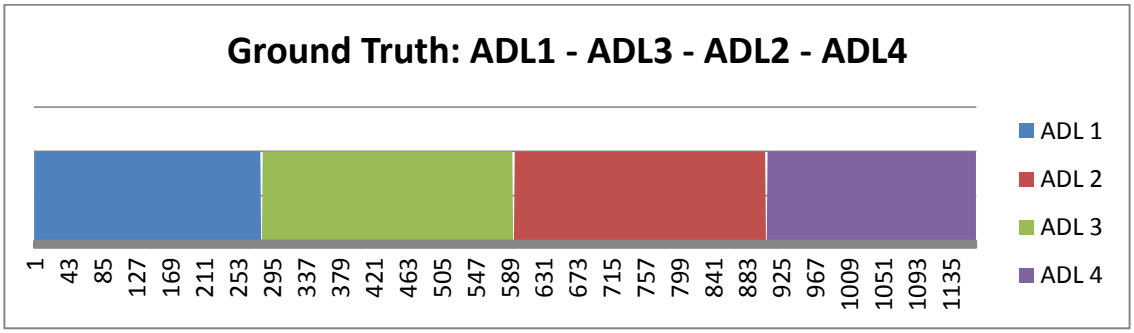
Form Factor Only

Test Video: ADL 1 - ADL 3 - ADL 2 - ADL 4

Training Data:

ADL 1:	Video 1	Video 2
ADL 2:	Video 1	Video 2
ADL 3:	Video 1	Video 2
ADL 4:	Video 1	Video 2





Appendix G: Results for Distance Measure Comparison

In the following tests, the classification process is change to test each distance measure individually. The following tests use all features. Note however that Mahalnobis distance excludes variance and busy fraction during calculation. The dataset used for these tests include:

ADL 1: Video 1, Video 2, Video 3
ADL 2: Video 1, Video 2, Video 3
ADL 3: Video 1, Video 2, Video 3
ADL 4: Video 1, Video 2, Video 3

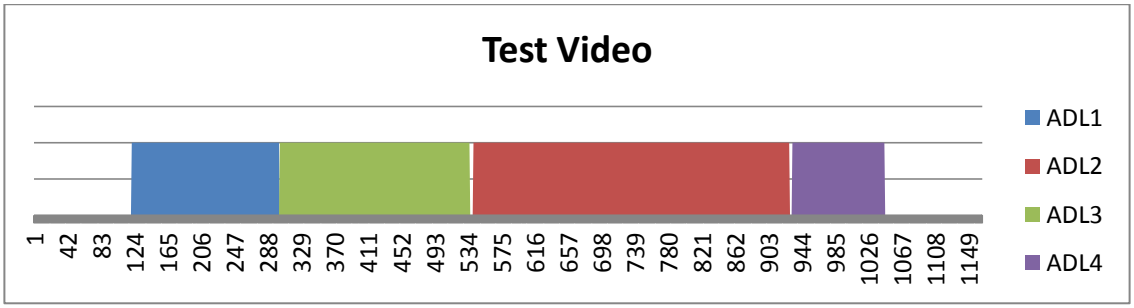
The video being tested in each of these cases is a combined video containing a sequence of each ADL.

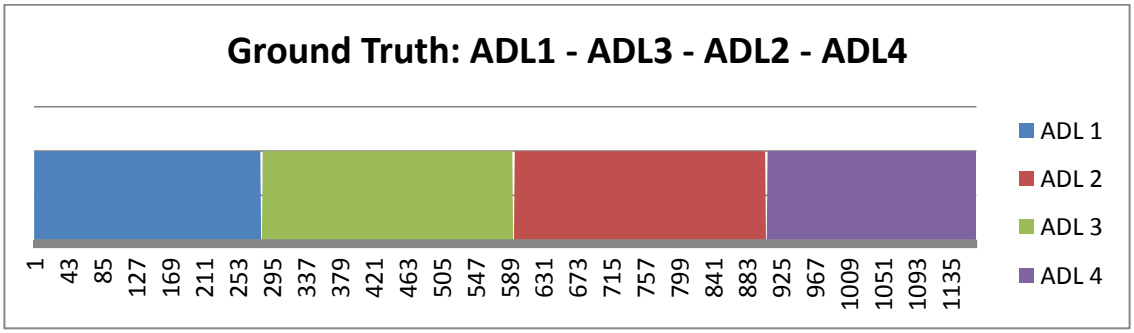
Euclidean, Manhattan, Mahalanobis (voting)

Test Video: ADL 1 - ADL 3 - ADL 2 - ADL 4

Training Data:

ADL 1:	Video 1	Video 2
ADL 2:	Video 1	Video 2
ADL 3:	Video 1	Video 2
ADL 4:	Video 1	Video 2



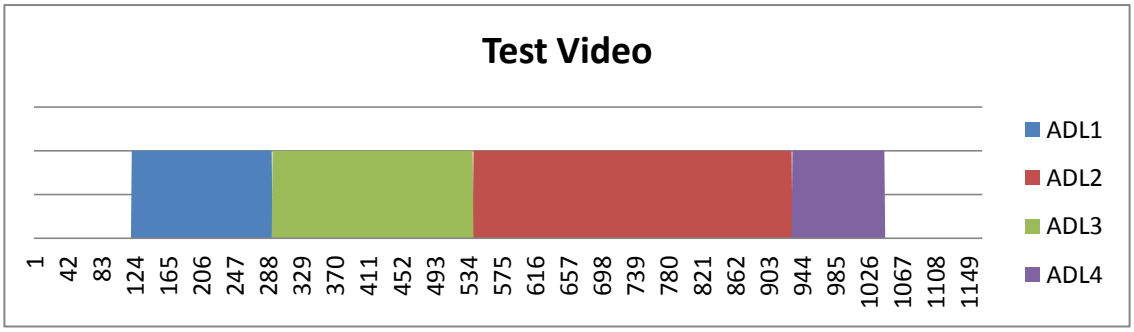


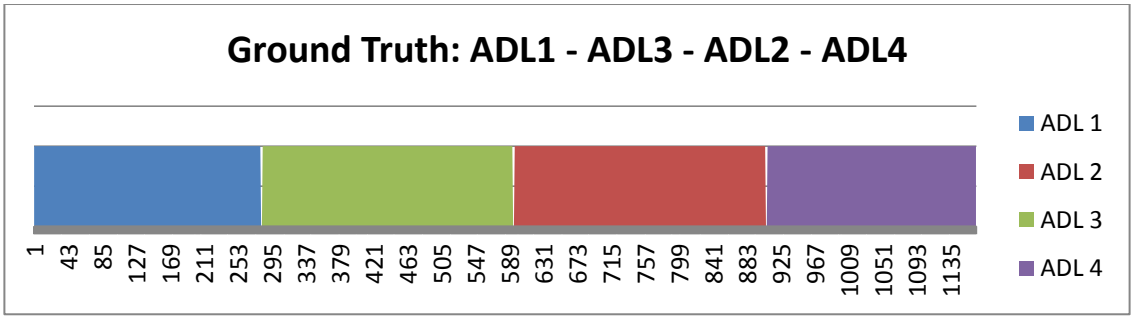
Euclidean Only

Test Video: ADL 1 - ADL 3 - ADL 2 - ADL 4

Training Data:

ADL 1:	Video 1	Video 2
ADL 2:	Video 1	Video 2
ADL 3:	Video 1	Video 2
ADL 4:	Video 1	Video 2



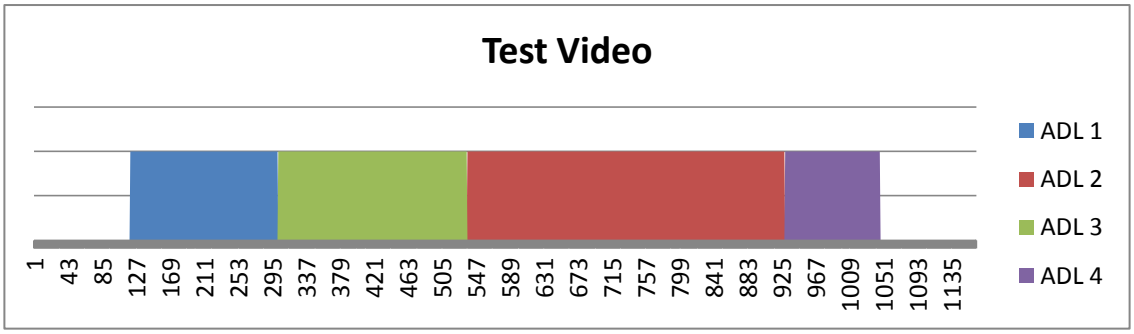


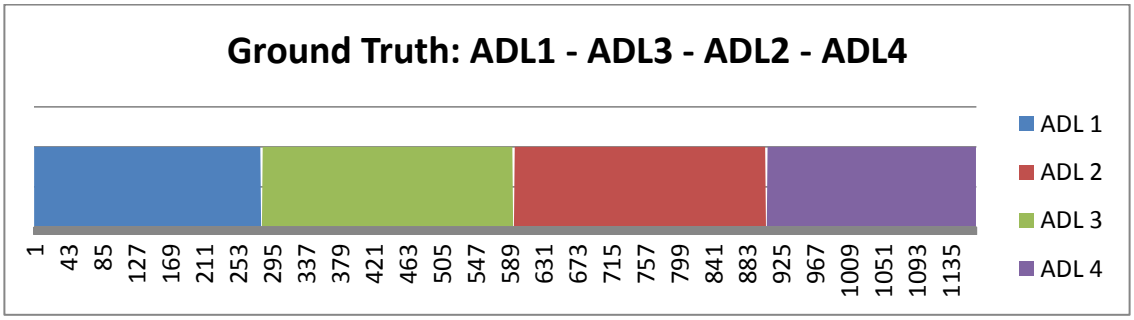
Manhattan Only

Test Video: ADL 1 - ADL 3 - ADL 2 - ADL 4

Training Data:

ADL 1:	Video 1	Video 2
ADL 2:	Video 1	Video 2
ADL 3:	Video 1	Video 2
ADL 4:	Video 1	Video 2





Mahalanobis Only

Test Video: ADL 1 - ADL 3 - ADL 2 - ADL 4

Training Data:

ADL 1: Video 1 Video 2
ADL 2: Video 1 Video 2
ADL 3: Video 1 Video 2
ADL 4: Video 1 Video 2