Delft University of Technology

Thesis

Action Recognition From Variable Viewpoints

Towards a safer living environment for elderly

Woohoo!

Date

4th of September 2016

Faculty

Delft University of Technology
Faculty of Mechanical Engineering
BioMechanical Engineering Specialization
Delft BioRobotics Lab

Report by

Dennis Priester

Supervisors

Dr. ir. Maja Rudinac
Prof. dr. ir. Pieter Jonker

Action Recognition From Variable Viewpoints

Towards a safer living environment for elderly

Dangerous action detected!

Abstract

The Dutch Central Bureau of Statistics expects the elderly population to grow from 2.4 million in 2012 to 4.7 million in 2041, putting intense pressure on health care budgets. As elderly get older and older, even more pressure on health care budgets will exist in the near future. Therefore, there is currently a focus on prevention: reduce incidents and allow people to live safely in their own homes for a longer time as this will greatly reduce healthcare costs. One approach to this goal is the development of an autonomous service robot, capable of assisting elderly persons in their daily activities and able to recognize dangerous actions or situations.

Although humans are seemingly capable of effortless action recognition, artificial systems employing human action recognition algorithms still have many difficulties doing so. This thesis proposes a new approach: Kinect RGB-D skeletal data is captured in a spatial temporal pattern, making use of 3D motion history. Using a dimensionality reduction technique called orthogonal class learning a novel representation is generated called motion history spatio temporal pattern or MH-STP. This allows the space and time information of the skeleton to be preserved and compressed in a compact feature. Using an action graph model for classification, real time recognition rates can be achieved by modeling probabilities of observed consecutive poses. This thesis aims to provide a novel method for successful, robust and reliable action recognition in an online setting.

The system was tested on the Microsoft Research Action 3D dataset as well as a novel dataset (12 actions, 9 subjects, 3 different angles, 3 instances per angle) recorded specifically for this thesis. Although the Microsoft dataset did not have high quality skeleton data, promising initial results were obtained (existing methods 68% - 90%, proposed method 83%). More extensive testing of the method's parameters on the novel data set led to a much better understanding of the MH-STP representation in both offline and online action recognition methods. This resulted in an excellent offline recognition rate on the novel data set in a more challenging cross subject situation (proposed method: 95%, existing up to 89%).

In the online situation the general approach was tested as well using a novel two-stage classifier that adds a secondary classification step on top of the action recognition framework. This allows (re)classification if the action graph result is not confident enough or fails to classify an observed test sequence. A very good recognition rate of 92% was observed in a 50% training cross subject test. Although this result was well above expectations, inter-action variance is a challenging factor, depending on the MH-STP grid size and the amount of available samples.

Acknowledgements

Finishing this thesis, which concludes my study at the Delft University of Technology, was quite a long walk. Yes, it took a while. But I am very happy I didn't take the bus as I met some wonderful people along the way and learned a lot in this time.

Of course, guidance along the road was given by a lot of people and I am especially grateful that my girlfriend, family and friends never stopped nagging me with the question "When are you going to graduate?" or "When are you *really* going to start working?" although I was already investing 40+ hours a week in my own company.

All and all, this work could not have been finished without the help of other people. First of all I would like to thank dr. ir. Maja Rudinac who allowed me to choose my own subject, believed I could do it, had to deal with me and still is willing to take part in the graduation commission. For this I am very grateful.

Secondly I would like to thank prof. dr. ir. Pieter Jonker for the support in the final months of working on this thesis. I gained quite some insights during our short talks, and this helped me a lot to finish my thesis. Furthermore I would like to thank the other commission members as well as the people who were so kind to offer their support: the awesome people of Daza Opticare for helping me generate a novel action dataset, Tim and Randy for being cool and relaxed during our discussions about results in my work that were just-not-perfect to me and last but certainly not least, my girlfriend Mandy who was understanding, kind or angry just at the right moments.

I would like to conclude this text with my sincerest hopes that in the not-so-far future, elderly people are given the certainty that they are not left alone helplessly in their homes. A system that could protect them from danger in their own homes without infringement of their privacy is in my opinion something that should be developed as soon as possible. Hopefully this work, even if it is an insignificant piece in the total path towards it, can contribute to this.

Delft, 4th of September 2016

Dennis Priester

7:15 A.M.

In the stark white kitchen, Alicia reads Billy Rae a list of chores she wants him to do today, including polishing the wood floors, cleaning the windows and changing the beds. Billy Rae, who was bought with a Southern-style voice synthesizer because it reminded Bill of his Texas roots, acknowledges the commands with a burst of "Your Cheatin' Heart," and Alma turns away, satisfied. When the Morrows first paid their $5,000 and brought the 4-foot-tall robot home two years ago, neither was sure they'd ever adjust to having it around. In fact, it had taken several weeks – and several near-catastrophes (such as the time Billy Rae pulled down the kitty chow, instead of the oat bran, and served it to them with milk for breakfast) – before they were able to input all the specific information the robot needed to function properly. Now Billy Rae is an irreplaceable part of their lives.

L.A. 2013, 3rd of April 1988, Los Angeles Times[1]

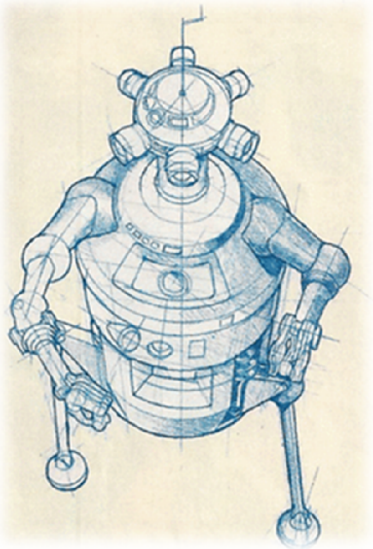

Figure 1 Sketch of a future household robot, as drawn in 1988.

Table of contents

Chapter 1

Action recognition: can it improve the lives of our elderly?

1.1 The future is here?

Although a lot of the predictions in the full L.A 2013 article as published 3rd of April 1988 [1] have come true, household robots are still not available in the vast majority of our homes. Apart from devices such as vacuum cleaners and lawn mowers, of course. We now know that the prediction on household robots was slightly optimistic but that does not mean there is no interest in these clever devices. On the contrary. There is a continuous increase in interest in the field of household robots, especially towards health care, leisure and personal service robots [2].

This is confirmed by looking at the market for 'simple' household robots, which rapidly increases. In 2009, according to the United Nations Economic Commission (UNEC) and International Federation of Robotics (IFR)[2], already stated that the market for these types of robots would exceed 17,1 billion dollars in 2010 and is estimated to be 51.7 billion dollars by 2025, a figure already estimated in 2007 by Bill Gates [3]. In more recent work late 2012, it is stated that the total service robotics market is expected to reach $46.18 billion by 2017 [4]. In December 2015, it was estimated that the market for service robots will annually grow by 23.5% in the coming 2015-2020 period [5].

The question remains: why can we still not buy our own robot with capabilities such as Billy Rae, R2D2 or Bender? Apart from technical difficulties such as 'how to physically build and power one', difficulties are numerous. To name a few: a robot should be able to respond correctly to its environment, interact with objects, interact with humans, communicate properly, know how to recognize environments, objects, humans, understand and perform tasks, and many more. Some examples are shown in Figure 2.

One of the challenges towards creating such an advanced system is action recognition, which is the topic of interest for this thesis. But why is action recognition such a challenging task? One's own experience with action recognition, supported by fictitious robot characters such as Disney's Wall-E, might make action recognition seem to be an easy task, as humans perform action recognition seemingly effortlessly and instantaneously. One could even characterize most types of recognition performed by humans as 'involuntary'. Unfortunately, although many researchers have worked on this vision based problem for decades[6], there is still no uniform approach available for the topic of action recognition. This is caused by the large variance in observable scenes, actors, possible actions and circumstances that exist. While humans excel in dealing with these variances, it is still quite hard to deal with these variances for an artificial system even if it is designed for a specific recognition problem. This is one of the reasons why a system such as Billy Rae the household robot is not available yet.

Figure 2 Three example application fields for action recognition: surveillance, gaming and healthcare.

Note however that artificial systems employing action recognition are not limited to household robots by far. Although these robotic systems are great subjects to help imagine the potential of action recognition systems, there are many more applications: action recognition can be used in all kinds of systems that benefit from the interpretation of a human's actions and activities. Because of the broad usefulness of action recognition, the field of application grows every year along with the development of new action and activity recognition techniques (e.g. [6]–[10]); long wished applications that seemed fiction before are becoming a reality because of these new techniques.

Examples of these applications can be found throughout many different branches in society, with the field of security as best known example. Other fields of applications are in health care, such as attempts to create fall detection algorithms [11]–[13]. Other examples of research fields are gaming [14]–[16], video annotation [17] and sports [18]. One great example of an action recognition is Microsoft's Kinect which uses an action recognition system as a control interface for games. As the field of focus for this thesis is health care, another great example is a still to be developed system that allows elderly to live safer, and longer, in their own homes by recognizing dangerous actions.

1.2 Research context: can action recognition help our elderly?

A system such as in the given example, which would protect elderly in their daily lives, would be most welcome. This is especially true as the aging population is putting heavy pressure on the available health care budgets in The Netherlands. Unfortunately this problem will keep increasing for the next twenty years (Figure 3). Causes for this phenomenon are the decline in fertility rates since the 1970s and an increased life expectancy of elderly, which reduces the number of working persons (contributing most to taxes) per retired person (using most of the health care budget) from little under four to less than two by the year 2030. [19], [20]

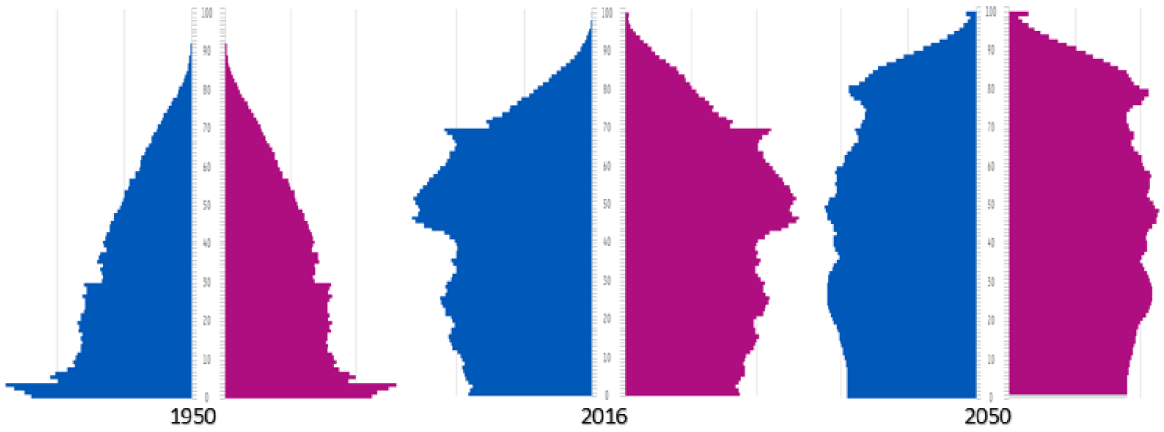

Figure 3 Past, present and expected population in the Netherlands. Source: CBS.

A large part of the elderly health care costs are related to elderly care centers and the aftermaths of fall incidents. In 2011 alone, 83.000 elderly people had to visit emergency health care due to fall incidents and the amount of hospitalizations increased to 43.000, which is a dramatic increase compared to a decade earlier. Furthermore, 2.165 elderly people (in most cases indirectly) lost their lives due to fall incidents. Some of these incidents occur in elderly care centers where help is close by. Unfortunately, accidents are occurring at home increasingly frequent, with the risk of waiting for help for hours, as elderly people are stimulated (forced) to keep living in their own homes due to budget cuts in elderly care centers. [21], [22]

Apart from the emotional and physical impact on the lives of the people that experience such an incident, the impact on the health care budget is significant. For the year 2011 it was estimated around 674 million euros [23] and this amount will rise yearly, as the elderly part of the population is expected to grow rapidly. Developments aimed at the safety of elderly therefore seem to be very promising and rewarding as investments into new techniques might reduce or prevent further increase of these fall incidents. This is especially true when these systems are used to allow elderly to live longer in their own homes independently.

An elderly health care system employing an action recognition module may monitor the living patterns of an elderly person and detect changes in this pattern, potentially indicating health problems long before these would be detected by family or nursing personnel. It could detect or prevent dangerous activities or actions in their day to day lives. It could alert neighbors, care takers or family members in case of dangerous situations or emergencies and thus provide a safer environment for those who need it. Your own home could become a safer living environment.

Apart from the safety aspect, there are other benefits of health care systems employing action recognition. For starters, workloads can be reduced for nursing personnel. Due to the budget cuts there is increasingly less personnel available in health care centers, especially during the nights as these shifts are more expensive. This means one nurse may have responsibility of 24 to 36 elderly people, and in the near future even up to 48 persons per night shift nurse [24]. With the current state of motion-detection / sound / infrared detector alarm systems, this inherently leads to a very large amount of alarms to be dealt with, of which the majority are false: people going to the toilet, an Alzheimer patient getting out of bed at 2AM, and a person making noises in his or her sleep. If a nurse would only be notified of events that require (potentially immediate) attention, the nurse would be able to prioritize the different events during the night, gaining control of situations more easily.

Another effect of action recognition on health care systems is that this will lead to a solid increase in privacy. Although these health care systems intuitively seem to harm the privacy (especially when called surveillance systems), this is far from the truth. First of all, good action recognition systems should be able to work autonomously; not a single person is needed to review data streams from inside the living environments of elderly. To put this in contrast: an increasing amount of elderly care centers starts to apply camera surveillance to tackle the before mentioned problem of prioritization, damaging the privacy of both elderly and nursing personnel. Secondly, nurses no longer need to frequently check patients in their rooms during the night and day.

1.3 Main issues and research question

Nonetheless, there are still no autonomous products such as Billy Rae available on the market due to the numerous difficulties of developing such a system. Apart from technical difficulties like processing power and methodological ones such as segmentation, the main difficulty of creating a successful action recognition system is the vast amount of situations that can be observed. This is caused by the fact that there are many different types of actions, different ways to perform an action and different conditions an action is performed and captured in. Also, not all persons have the same posture and wear the same clothes. Even the living rooms such a system should be able to operate in are far from similar to each other. On top of that, one type of action can be performed in several different ways, while on the other hand there can be multiple different actions that look a lot like each other. In short: variance makes sure there is no easy general approach to the challenge of action recognition.

Finally, another big problem is that classic action recognition systems that are able to detect and recognize multiple different actions require a very large database with reliable training samples. The problem is that in many occasions these training samples have to be manually labelled and segmented, which is a time consuming and thus expensive job, plus such systems are unable to recognize actions they are not trained for.

The notion that health care systems with action recognition possibilities do not yet exist despite the obvious benefits of these systems lead to the following research question:

> *Can we create an action recognition system that is capable of recognizing in-home daily actions of persons, independent of their orientation, posture and appearance, without requiring very large sets of training data?*

Having studied the literature extensively, a first answer to this question is that it is very likely possible to create such a system.

Although there is no general approach to this problem of action recognition, algorithms have been tailored to the specific needs of situations and environments in an effort of narrowing down a large portion of the possible variance [16], [25]–[29]. This process of creating a tailored action recognition algorithm consist (roughly) of five main parts: data acquisition, representation, features, training a classifier and classification, which will be discussed very briefly in this introduction.

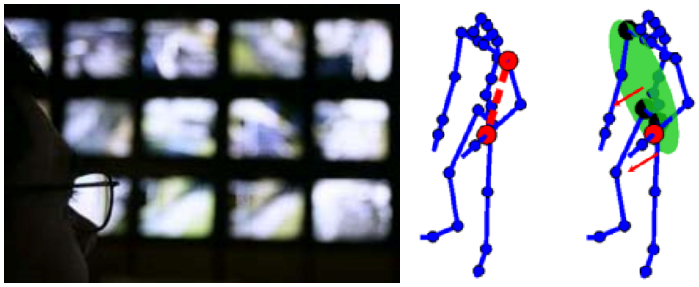

Figure 4 Left: do elderly really need to sacrifice privacy for safety? Right: two examples of a skeleton representation (blue) with in red/green examples of possible features such as distance between joints and speed of a joint perpendicular to a certain plane.

1.4 Contributions of this thesis

There are multiple options to perform data acquisition [30]–[33]; Some applications may require the use of 2D video data or 3D depth maps, some applications such as video annotation use readily available data as offline data stream while others such as surveillance require an online (live) data feed. Note that until the release of Microsoft's Kinect, focus was on using 2D video data, as 3D depth sensors were in general too expensive and inconvenient in use. After this moment, 3D sensors became very affordable and the field started shifting towards 3D data acquisition. Once the data is acquired, a suitable representation is formed. This is a different way of showing the data: maximizing the available information while reducing sources of variance. A good example is a skeleton representation: it is extracted from 3D depth data by a set of algorithms and accurately depicts the place and orientation of a person's body parts. Third part of the approach is the choice of a feature set: which properties of the representation can be used as to distinguish between different observed actions and how can these properties be obtained? Finally, a suitable classification method has to be chosen that is able to use the features extracted from the representation to predict or calculate to which class a newly observed sample likely belongs to. By using a certain amount of training samples, this classifier can be trained to do so properly.

Over the years, numerous different methods have been proposed to maximize the accuracy and speed of action recognition algorithms by developing all kinds of representations, feature sets and classification methods. To realize a well performing action recognition algorithm, this work makes use of a 4D space time pattern (STP) as it is an intuitive visual representation that is able to preserve spatial and temporal contextual information without reducing the flexibility needed to accommodate intra-action variations. In this representation space and time axes are divided into multiple segments to define a 4D grid, which allows easy transformation into a high dimensional vector. To give an idea on the high amount of features: a rough STP grid with e.g. 10x10x10x3 space time cells will already lead to a 3000 dimensional vector but an STP with a higher resolution such as 30x30x30x3 leads to a representation with 81.000 space time cells. It is important that this high dimensionality should be reduced by techniques such as principle component analysis (PCA) and comparable methods such as orthogonal class learning (OCL). This ensures the creation of a much lower dimensional feature vector that is constructed from the main information of the original high dimensional STP, which helps classifying the actions.

This work proposes two novelties in the field of action recognition:

1. The use of an intermediate skeleton representation, which is used in a motion history like fashion to create a novel 4D spatial-temporal pattern representation called MH-STP.

2. The application of this specific representation in a state of the art two-stage action recognition framework using action graphs that allows actions to be detected and recognized in an online setting without requiring very large data sets.

Figure 5 Left: example of 4D space time pattern created using depth data directly. Right: from scene to the STP representation as proposed in this Thesis.

The validation of the proposed novelties in both offline and online settings is tested on the publically available Microsoft Research Action 3D dataset and a novel dataset created specifically for this work to reduce the chance of biased exploration of the MH-STP.

The main novelty of this thesis is the proposal of a new method to form a new representation: an intermediate, normalized and interpolated skeleton representation is used in combination with a 3D-adapted version of a motion history imaging scheme [34](Figure 5). The new representation is therefore called motion-history spatiotemporal pattern or MH-STP. The method is inspired on a 2012 article [27] that proposes to build up an STP from point clouds directly, making use of a saturation scheme. The hypothesis is that this method is likely to achieve good results as it should be able to effectively reduce large sources of variance such as rotation, posture, appearance and depth noise while at the same time preserves location and time dependence. Also, the method should be less memory and computational demanding than the method that uses point clouds directly. Apart from presenting the used method in Chapter 4, the effects of changing the parameters of the algorithm are also explored and discussed in the Chapters 5 and 6. Hopefully, this improves the understanding of the various aspects of the representation.

This work also describes the recognition of actions using the new MH-STP representation in an online setting, which means that it tries to detect actions immediately after the occurrence instead of working with pre-recorded data as is the case in an offline setting. In the online case however the algorithm generates the MH-STP representation using short consecutive intervals of an action instead of using the complete action sequence at once. The low dimensional features extracted from these - short - MH-STP representations are in turn classified by a neutral pose classifier that is using a support vector machine (SVM)[35]. Using this SVM neutral pose classifier, action start and stop indicators within a sequence are detected. These indicators allow the interesting, non-neutral sequence of pose representations to be classified using an action graph. The benefit of using action graphs is that they require very few training samples compared to other methods. Once trained, an action graph returns the probability that a certain set of samples belongs to the trained action [25].

This thesis is focused on the development of the new representation. Therefore segmentation of the action samples is considered to be a solved problem. This is done by using available techniques such as background subtraction to determine the region of interest and by a time segmentation of the data samples prior to their use in the newly developed algorithms. Also, since there is no public data set of elderly performing actions available yet and creating one is difficult and very time consuming to arrange, the proposed novelty is not tested on data samples of actions performed by elderly people. It is therefore likely that the same system, when observing an elderly person whom is performing actions, behaves differently. This could require that the system needs to be fine-tuned and retrained in order to achieve similar recognition accuracies.

1.5 Thesis outline

With the background information, importance of the study, research question and novelty of this research introduced, this introduction comes to an end. The structure of this thesis is as follows: In Chapter 2, related work is presented as well as in depth information on spatiotemporal patterns. This is followed by a description of the used approach to the problem in Chapter 3. In Chapter 4, the representation and methods of offline and online action recognition are discussed after which in Chapter 5 a novel dataset is presented, as well as the experimental results of the offline action recognition algorithm is presented. The online action recognition results are then given in Chapter 6 and in Chapter 7 conclusions on the offline and online results are drawn. Finally, the possibilities in future work are given Chapter 8.

Chapter 2

Action recognition in literature

This Chapter not only describes related works that try to achieve action recognition, but is also used to offer background information and explain several different topics that were used within this research. This is necessary to gain knowledge and insight in the underlying problems and to help understand the offered solutions. In relation to the research question, questions such as 'what is an action' (Section 2.1), 'how do humans detect actions?' (Section 2.2), 'how is a computer able to recognize an action' (Section 2.3) are answered first. Afterwards, the challenges of action recognition (Section 2.4.1 / 2.4.2) and the difference between offline and online action recognition (§ 2.4.3) are discussed. This is followed by a short introduction to representations (Section 2.5) and features / dimensionality reduction (Section 2.6) which are in close relation to action recognition. Finally, 2D, 3D and the explicit use of time in 3D representations are discussed (Section 2.7, 2.8 and 2.9 respectively) followed by an overview of the most important articles (Section 2.10) and a brief summary of the most important findings from the related work (Section 2.11).

2.1 What is an action?

Detecting and recognizing actions in the broadest sense means that apart from only wanting to know 'what is in this scene' we also want to know 'what is happening?'. This introduces the variable: 'time'. Time lets us use the temporal information available in the scene which is very useful when discriminating between actions such as running and walking, since the movements are similar but the tempo varies. Time increases insight in interaction between objects, facilitates causality analysis and the detection of state changes. However, time is not always a necessity when detecting actions: a lot of actions can be recognized while looking only at one frame. The scenes in Figure 6 for instance depict actions straightforwardly: a girl running, making a 'thumbs up', playing soccer, and drinking water from a glass. Or are these actions much less straightforward to detect than it seems? It turns out it is slightly more complicated than that. To understand this, the following question needs to be answered: what exactly is an action and how can knowledge about this contribute to the overall problem of action recognition?

Figure 6 Different actions depicted in a few frames.

According to Webster's Revised Unabridged Dictionary[36], an action is described as "*A process or condition of acting or moving, as opposed to rest; the doing of something; exertion of power or force, as when one body acts on another; the effect of power exerted on one body by another".* This leads one to believe that an action must constitute physical movements. An action of a person can thus be seen as movement with or without interaction with its environment. A philosophical definition of action was formulated by Arthur Schopenhauer [37]. Schopenhauer states that an action is always performed by an actor and that there is a relationship between an actors will, the external world and its action. According to Schopenhauer, "*Action is a perceivable manifestation of the will under given circumstances and thus one's body is the instrument to furnish the will through actions in the perceivable world.*"

The previous two definitions combined can form basic boundaries of what an action is in the light of action recognition:

> *An action can be observed by movements and is the result of something planned by a 'will' to do so.*

The result of this definition is that 'natural actions' are excluded from scope, although it can be discussed if for example animal actions are in accordance with the definition or not. It is also clear that an elderly person whom is accidently stumbling over a loose stone and falls down is not doing this directly because her 'will' wants to. Therefore the definition is slightly adjusted:

> *An action can be observed by human movement and is the (potentially unexpected) result of something planned by a 'will' to do so.*

There are several scales of actions that fall within this definition. For instance, the woman drinking water in Figure 6 performs a different action per scale. On the smallest scale, an action could be moving one finger. One scale up would be a coordinated movement of body parts to 'grab an object', as well as lifting the glass, bringing the glass to the mouth, taking a sip of water, putting the glass back on the table and releasing the glass. Another scale up, the descriptive action could be 'drinking'. A descriptive action for the running woman could be 'running' and for the monk it could be 'making a thumbs up gesture'. Again zooming out, activities can be observed such as walking a marathon, being on holiday, playing soccer or having lunch.

The difference between these actions is timespan and the level of consciousness while performing these actions. This shows one major difficulty in action recognition: how do you know if there is enough information to tell what is happening, especially when the observed data has never been seen before? What scale of the action are we trying to perceive? When does an action start or stop? This problem of segmenting actions is out of scope of this thesis and therefore not addressed in this research. Instead, the problem of action recognition is simplified by only using data samples that contain one pre segmented action each. The actions in these samples start with a neutral position, followed by the action itself after which the action is ended in another neutral position.

2.2 How do humans detect actions?

Humans are well capable of observing actions of other humans. They can name what is happening on each of the action scales as mentioned in section 2.1 and are even able to infer other people's intentions from observed actions [38]. This is possible since the human brain relies on internal models of kinematics and dynamics of physical representations that are not only used to generate the motions needed to e.g. 'grab a cup', but also to recognize these motions from observations [38], [39]. This does not necessarily mean that temporal information such as movement is always needed to recognize actions. Studies show that still images captured in the process of movement, induce a perception of movement in the brain and therefore allow action recognition using implicit dynamic context [38]. This inspired some researchers to investigate the use of Action Points, which are similar (but not equal to) important key poses in action sequences [16].

Contextual information improves action recognition as well. When a soccer player is observed making a movement with his foot towards a ball, it is expected that he will kick the ball. Therefore it does not come as a surprise when the ball is kicked and shoots away. However, contextual information is not a prerequisite for action recognition as it turns out that the kicking of the ball is still recognizable when only the kinematics of the subject are shown [40]. The contextual information is nonetheless of use when looking at it from a causality perspective: an object was lying still and now it is soaring through the air, which strongly suggests it has been kicked as footballs do not start to fly all by themselves.

Causality is a very important source of information for humans. In the early nineties, a series of experiments by Spelke and Van de Valle [41] showed that infants aged 7.5 to 9.5 months old are already sensitive to a wide range of visual events derived from causal constraints of motion, which they tested using a variety of experiments. One of them is shown in Figure 7. They posed that at that young age three different principles are active in the understanding of motion, even well before the infants are able to reliably segment still objects: contact, cohesion and continuity. In short these principles can be explained as 'no action at a distance and no contact without an action', 'no splitting or fusing unless a series of other causal events causes this' and 'no gapped trajectories or contactless collisions' [41], [42]. It is clear that these principles are following physical rules. Remarkably, the study showed that infants act surprised when a tricked situation is shown where these rules are broken. This fact, that even infants at that young age are able to show fluent application of these principles, strengthens the idea that humans use internal physical models that help them recognize actions.

Concluding, although there are many events that are not causal such as gestures, contextual information and the knowledge of causality helps in recognizing a wide variety of tasks. Examples are grasping a glass of water or kicking the football: most of the information is encoded in the scene or can be deducted from the observed causal events. This helps humans to reconstruct the plans or intentions and thus the action performed by the actor through its will [43]. Although these principles are worth mentioning and of large importance for action recognition by humans, in this research however causality and contextual information is not used, as these topics would increase the complexity of the thesis too much.

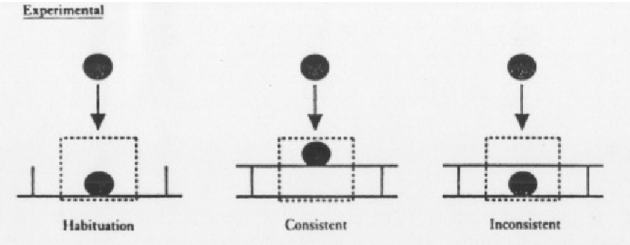

Figure 7 Experimental test showed to infants. The third situation is inconsistent with the three basic rules: the ball should not be able to move through the table.

2.3 How to detect an action with an artificial system?

Most works assume actions to be detected and readily segmented in time before action recognition is performed. This assumption facilitates action recognition, since the burden of detection and segmentation is removed from the recognition task [6]. Other researches primarily focussed on repetitive actions such as walking, cycling and different gymnastic exercises and they rely on the repetitive nature of the action for proper action detection. However, this might not always be very realistic. In an online setting for instance, action detection and recognition should preferably be performed simultaneously and an action should be classified directly, not after a number of repetitions. Otherwise, an action such as jumping is detected only after a few jumps, while you might want to detect the first one already.

The approach to the detection of an action depends on the observed scene. Detecting sign language gestures, a single person in a controlled environment performing one action at a time, or multiple persons minding their own business in the lobby of a crowded airport, require different approaches for action detection. Similar in these scenes is the presence of humans, and the fact that these humans, or at least parts of these humans, are moving: movement is a strong indicator of the occurrence of an action. These two action indicators (presence of humans and movement of these humans), require several preliminary processing steps, which can be found in almost any vision based approach. For instance, a region of interest can be found by separating the background from foreground by background subtraction. From this foreground the presence of humans can be detected and afterwards motion of the human could be followed in a series of captured observations. This section covers a selection of these methods that are in close relation to this work.

For background subtraction, many different techniques such as frame differencing can be used. This is a technique that compares two frames and tries to determine what has changed between them and then considers this as foreground. Unfortunately this has some limitations, in case of e.g. moving backgrounds. Fortunately, over the years more complex, often probabilistic methods have been developed: ranging from simpler mean filters and running Gaussian averages towards more complex methods such as Mixture of Gaussians, Kernel Density Estimation, image variation co-occurrences, Codebook Models, coherence matching methods and many more [44]–[46]. However, when the camera itself is moving these methods usually fail. In these cases, especially when 3D data is available but also when 2D information is used, trajectory estimation can be performed to compensate for ego movements. These methods usually employ pixel based models in probabilistic/statistical modelling, estimate 3D locations in some cases and most often rely on online-learned frameworks [47]–[49]. In this work background subtraction is not directly used, but it is embedded in the device that was used to record the datasets that are used for validation.

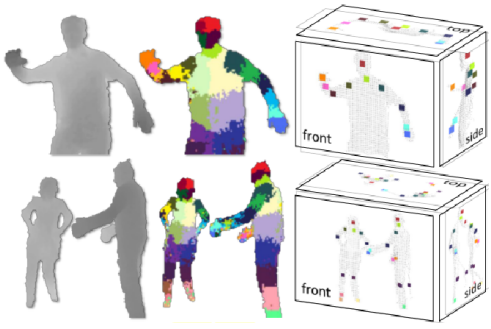

Figure 8 Depth image -> Body parts -> Joint estimations.

When it comes to human detection, this is a topic of interest for many researchers. Common 2D approaches involve the use of an accurate background model to extract interesting objects on the foreground and rely on either a body parts' based approach or an approach using detection windows. [50]–[52]. Human detection in the 3D case relies on range data (depth data) or 'point clouds' as input. The difficulty in using 3D depth data is that the appearance of a human body in the data is highly dependent on its body pose, distance to the sensor, self-occlusion and occlusion by other objects. The benefit is that this data is not sensitive towards colour or illumination changes and it can help to solve occlusion issues in some cases. One of the more recent developments in human detection from depth images uses a 3D object recognition approach to estimate body parts. In this research, the Microsoft Research group uses a large artificial training database allowing body parts to be detected invariant to pose, body shape, clothing etc. [53]. Their main contribution is to treat pose estimation as an object recognition task using body parts as an intermediate representation. Figure 8 shows an example of this technique.

Finally, the detection of (human) movement can occur using a wide variety of methods. Some are suitable for 2D or 3D data and some methods are especially useful for cameras that have ego movement. Examples of such methods are frame differencing, the well-studied method of optical flow [54]–[57] and motion saliency [58]. Especially interesting are the methods that focus on human motion detection: once a human is detected, its motions directly reveal information. Already in 1973, G. Johansson [59] stated that the motion of a living body can be well defined with much less information than is available in a 2D gray scale image (Figure 9). Of course, this is even much less information than is available in RGB-D recordings that are now used to detect actions in. Johansson showed that humans are able to recognize activities even when this information is in an extremely compact form: a few bright dots attached to the joints of a body is enough for a human to get a good sensation of the 3D movement in the scene and to recognize the actions. This is even true when multiple persons equipped with bright dots are in a scene at once.

Inspired by this idea, other compact representations were used, such as "star" skeletons [60] and extremities-tracking [61], as well as MoCap data which is directly recorded by tracking the 'bright dots' connected to the joints of the subject in 3D coordinates. MoCap data however has as drawback that it is only suitable for motion analysis in a controlled environment since it is an invasive method that requires a calibrated recording room. The other methods were based on 2D observations and although reasonable coordinates were estimated, there was no reliable and accurate method to obtain a solid joint representation.

This changed with the arrival of affordable RGB-D sensors. Using a variety of algorithms a skeleton representation can be extracted from the acquired depth maps. The skeleton representation is in fact a joint location and orientation representation, with accuracies ranging from ~2cm deviation when in frontal view to 5cm in 45° orientation[62]. In 90° orientation the report stated that Kinect often failed due to self-occlusion. However, more recent skeleton estimation methods such as proposed in [53] by J. Shotton are likely to perform better under identical situations.

Once the skeleton representation has been determined, detection of human movement is a matter of comparing joint locations between successive frames, effectively leading to an optical flow of skeleton joints. On top of that, apart from detecting that there is movement it is now also visible in what direction and with what speed the movement occurs.

This work makes use of the action indicators described, by using Microsoft's Kinect sensor. This sensor extracts a person performing an action from the background (only if it detects it as a human) and allows easy motion detection by tracking the skeleton joints, facilitating the detection when an action starts and stops. Although the algorithms in the sensor itself are complex, it uses algorithms that preprocess the data that is acquired first, relying on lower level operations such as the described background subtraction, human detection and motion detection.

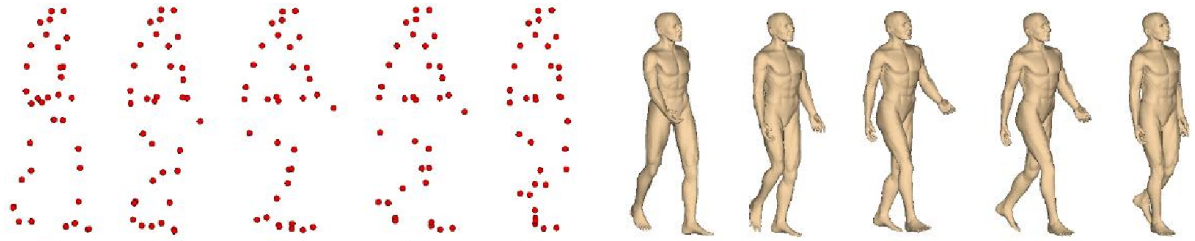

Figure 9 Showed by G. Johansson in 1973, humans are able to determine what is happening in an observation sequence when a sequence containing only a few bright dots are shown. [59] The images itself are not exactly true representatives of Johansson's but depict the general idea.

2.4 Important aspects of action recognition

As stated in the introduction, there is no uniform approach to the problem of action recognition due to the fact that there are many variances to take into account. This is due to different ways of recording that can be used, the fact that many different recording conditions exist and there is a lot of variance in appearance of the actor and performance of the action. This section discusses several approaches, pinpoint its problems, and gives a comparison of the state of the art. The focus of this chapter is limited: it does not focus on approaches that explicitly try to deal with differences in context such as interactions with other humans or objects, approaches that have to work in multiple different environments or works that try to perform action recognition while under the burden of heavy noise or partial occlusions.

2.4.1 Challenges

The notice that action recognition is still an unsolved problem brings up the following question: what are the common challenges in the field of human action recognition? Most of these challenges are related to variance since there are countless possibilities in purpose and application, circumstances and observed scene. Poppe [63] addresses these challenges in an extensive dissertation and splits them in four main groups: intra / inter class variety, environment and recording settings, spatial and temporal variation and evaluation criteria.

First of all, intra / inter class variances relate to how actions are performed: on the one hand, one type of action can be performed in several ways with large variety as can be seen in Figure 10. On the other hand there can be multiple different actions that look a lot like each other. These two situations seem to require opposing approaches and hence a good human action recognition algorithm needs to generalize well over the different ways that an action can be performed. However at the same time it should prevent that the algorithm will generalize over similar performed, but different actions.

Secondly, spatial and temporal properties of the data introduce some more variation in the recognition problem. As discussed, most researchers assume actions to be already well segmented in time before recognition occurs: it is known beforehand (often labelled manually) where the human is and when the action starts and ends. In reality however these ground truth spatial segmentation and temporal anchor points are to be determined with lower level pre-processing algorithms. Also, cameras record the analogous time in a discrete fashion causing actions to be depicted differently in each recording sequence. This all will lead to additional variation which a robust action recognition algorithm should be able to cope with.

Thirdly, environmental circumstances and recording settings introduce variation as well. Since action recognition is a vision based technique, difference in viewpoint, the used camera settings, appearance of a person and the environment (background, illumination, other foreground movements) all have their effect on the recognition process. Variations can be introduced by differences in the results of low-level pre-processing steps such as localization, segmentation and motion estimation and therefore also lead to variances in the results of high-level pre-processing steps such as joint estimation, pose estimation and action segmentation.

Another ongoing challenge in human action recognition is the limitation of available technical possibilities. Apart from technical challenges such as camera range, field of view and their performance under different circumstances, a major challenge is the continuous increase of required computer power. The reason for this is that techniques for data acquisition are continuously improving, leading to the acquisition of higher resolution image sequences recorded at higher frames per second, while at the same time containing more additional information such as depth maps. This especially is a problem when training with (very) large sample databases is required for proper training.

Although not all algorithms such as low and high level pre-processing steps are equally computationally expensive, the increase of available information presses on the available computing resources. Furthermore, as techniques proceed it becomes possible to successfully distinguish more and more types of actions in different situations, performed by different people. Once the amount of actions to be identified becomes large, generating reliable models or performing reliable classification might become too memory or computationally expensive. As an example, classification schemes like the action graph [64] that use Action-Specific Viterbri Decoding suffer from increasingly growing memory and computation requirements when the amount of actions to be classified grows. Luckily, in accordance with Moore's law, computing power is expected to keep increasing in the future, doubling each 2 years, allowing techniques that are now computationally prohibitive to be used.

Figure 10 Variances of the action 'jumping': a lot of variance between performers.

2.4.2 Temporal variance

One specific source of variance, temporal variance, requires additional attention. When an action is performed by moving the body parts in the exact same way, but at different speeds, this difference in the temporal domain might influence classification. This is especially true for methods that employ exemplar based recognition[8], [65]: when the similarity of an observed sequence with a set of template- or sample sequences is high enough the human action recognition system is able to deduct that a certain action is performed. A large temporal variation or a difference in sequence length however reduces the similarity and prevents proper recognition. To overcome this problem, a broadly applied technique, a distance measure originally developed for applications in the domain of speech recognition, is often used in the vision domain: Dynamic Time Warping (DTW).

Dynamic time warping as distance measure to match sequences was studied from the early 90's in several researches, trying to find an optimal non-linear match between two sequences by establishing correspondence between related events in both sequences. An example is given in Figure 11. Unfortunately, DTW is not a very suitable method to overcome this issue in this work as the method of DTW requires that key poses are available and these key poses are not yet known when trying to detect unknown actions. Still, the issue of temporal variance is partly dealt with in the proposed method in this thesis.

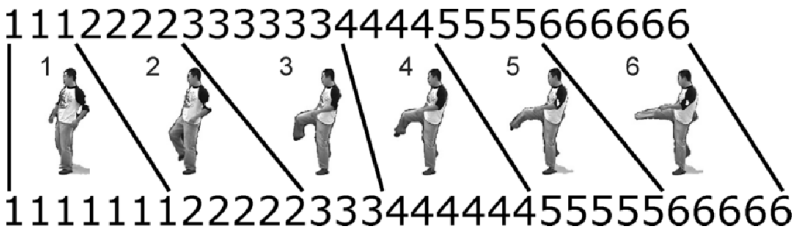

Figure 11 Reprint from [8]; stretch leg action sequence performed at different nonlinear execution rates.

2.4.3 Differences between offline and online recognition

Action recognition applications can be seen in both online and offline settings. Online processing is used to provide immediate information about the current scene. Offline processing on the other hand does not have the need to provide direct feedback, instead the data is processed much later. An online processing example is a security camera in a bank designed to detect armed robberies: immediate action is required in the event of a robbery. An offline processing example is video annotation to archive a soccer match: there is no need for a system here that immediately annotates all actions, processing can be done subsequently.

While trying to recognize actions or activities, offline processing has the benefit that it can use 'future' data, or in other words: data that was not available at the time when the action itself was not yet finished. Online processing can of course not access this future data and hence a time delay is introduced: only when the action is almost completely finished it is getting clear what action is performed. However, for some actions, data over a complete action is not always needed for accurate recognition. [16]

When it comes to learning or training a recognition system, there is also a difference in online and offline learning. Offline learning systems are trained in an initial training phase. After this phase their approximation function does not change. This type of training is often used when trying to construct a general, input independent target function. Although these trained systems excel in classifying the data sets that are alike their training set and they can partially account for noise, they are not really suitable for data that includes new events or changes over time, since the observations might diverge too much from the training set to allow proper recognition.

Online learning systems on the other hand are often (but not always) models of induction: incrementally learning from one instance at a time instead of learning whole batches. This is beneficial since in many practical applications, obtaining a representative set of training data is considered to be an expensive and time consuming business. This is especially true for end users: customers of a product that controls a television with custom gestures are not accepting a system that needs a ton of training data before they can use their own custom gesture commands for example. The key characteristic of online learning is that the system is able to refine its hypothesis over the target function (e.g. classification) with each new data instance obtained. This is called incremental learning, which makes them much more flexible than their offline counterparts.

Concluding, offline and online systems both have their benefits and whether a system should be fit for offline or online processing greatly depends on the purpose of the application. Both 2D and 3D human action recognition are suitable for offline and online processing, as this is just a representation that can be used by offline and online processes. When it comes to observing elderly in their home environment, a successful system will most likely contain both offline and online elements; the offline elements will provide a solid basis and the online elements provide accurate and real time recognition with the possibility to adapt to the scene that is changing over time, and perhaps even to changes in the behaviour of the centre of attention itself: the elderly person to be observed.

2.5 The important choice of representation

As mentioned in the introduction, a representation is a different way of showing acquired data: maximizing the available information while reducing sources of variance, or while optimizing for computations. Representations can be holistic or patch based, and some are using parts of both methods.

Holistic representations are very rich representations, potentially encoding much of the available visible information, and are in general not very susceptible to changes in appearance. However, their downside is that in general their success depends on the accurateness of low-level pre-processing steps such as background subtraction and segmentation since this step introduces noise if not performed accurately. Another downside is that as the scene is regarded as a whole, the system is more vulnerable to self-occlusions. Examples that were often used with 2D data are given in Figure 12.

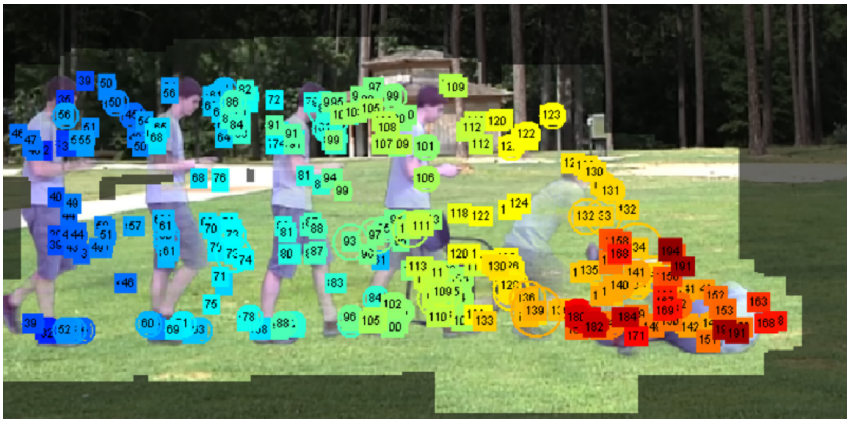

Figure 12 STIP features in the action sequence 'falling'. Method proposed in [9].

Patch based representations on the other hand try to describe the observed scene as a collection of independent pieces of information (patches). This class of representations is inspired by the idea that for some actions the entire body does not need to be observed for proper action recognition; instead the idea is that the main characteristics of an action are concentrated in small areas of the observed scene. Examples of patch based representations are spatial temporal information points (STIP)[9], space time cuboids (STC) [66] and bag of words classification (BOW).

For patch based representation the amount of extracted features usually varies from frame to frame. To ensure feature vectors used for training and classification still have the same size, histograms are often used. Unfortunately, this is a notorious disadvantage of using patch based representations: spatial and temporal information is usually lost in the process although researchers have proposed a fair amount of methods to overcome this issue by retaining the spatial information. [67] ,[68]. Benefits of patch based representation are that low level pre-processing is not always a requirement, reducing computational costs and errors induced by pre-processing. Another benefit is that, in contrast to holistic approaches, patch based approaches are less susceptible to noise and occlusions.

A more recent example of this principle is the 'bag of points' as introduced in a 2010 article by Wanqing Li et al.[25] In their work a depth map from the region of interest is extracted and projected onto three perpendicular planes (holisitic representation), after which the circumference of the projected silhouettes is sampled and back-projected in 3D space. This resembles a patch based representation. By 'bagging' these points, recorded poses are represented using only 1% of the available depth points.

Another example of a widely used representation is the skeleton representation as illustrated in Figure 13. It consists of a set of points in 3D space which correspond to the estimated location of the joints of a human test subject. This effectively removes many sources of variance. Using the skeleton representation differences in appearance and posture between elderly persons is of much less importance. The representation is used in many recent works [8], [53], [69], [70] which use the representation either directly or as intermediate representation to create their own, new representation such as in the 2013 article of Xiaodong Yang and YingLi Tian on Eigen joints [26]. Skeleton representations were initially estimated from 2D images but these were not very accurate. With 3D depth sensors, skeleton representation became easier and more accurate skeleton representations became available. More recently, Jamie Shotton of Microsoft research proposed a new method of obtaining a skeleton representation by using a patch based intermediate representation that uses holistic information of body parts to come to the final skeleton representation [71].

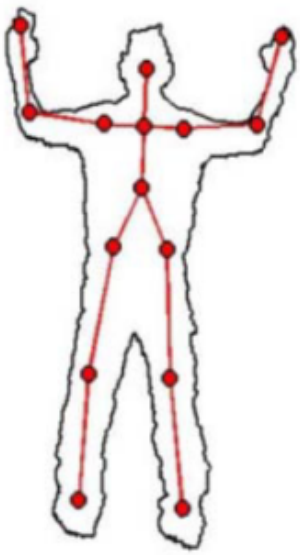

Figure 13 Skeleton representation example.

2.6 Advancements in 2D human action recognition

From the first scientific enquiry towards the current state of the art methods, human action recognition is following a rapidly paced development. This was even increased when cheap, reliable 3D depth sensors such as Microsoft's Kinect and the Asus Xtion Pro Live were released, which caused a shift of attention towards the use of 3D input data instead of 2D data. However, the fact that many new possibilities were coming into reach with the introduction of these new 3D sensors does not mean that research in the 2D domain has been done in vain: these works increased our understanding of a wide variance of aspects. After all, early research towards vision techniques that are based on 2D contributed a lot to come to the current state of the art of human action recognition. Also, the field of action recognition is and should not be limited to one type of input data and therefore 2D action recognition remains a field of interest even after the introduction of cheap 3D sensors. This section states some of the methods that are introduced and broadly applied.

2.6.1 Research through the years, covered in surveys

Most of these 2D action recognition methods are discussed in extensive surveys published in the last decade. In *A survey of advances in vision-based human motion capture and analysis* [72] by Moeslund, advances in Human Motion Capture and Analysis from 2000 to 2006 are discussed, covering a very large amount of papers from that period. The work mostly covers topics on pose estimation and tracking, involving low-level processing steps as discussed. In 2007, the work *The meaning of action: a review on action recognition and mapping* [73] by Krüger et al. gives specific attention to the topic of actions. It discusses different approaches for the representation, synthesis, recognition and understanding of actions without going too deep into details of lower level processing steps. The year after, in 2008, a survey is published in the magazine of IEEE Circuits and Systems Society: *Machine recognition of human activities: A survey* [7] by P. Turaga. Like Krügers work, it does not go in depth about current state of the art lower level processing steps, as it primarily focusses on high level modeling of activities and actions. It discusses both parametric and non-parametric action recognition approaches extensively, as well as different types of models such as graphical models and syntactic approaches and knowledge/logic based models.

The following years, in 2009 the extensive dissertation *Discriminative vision-based recovery and recognition of human motion* [63] and in 2010 *A survey on vision-based human action recognition* [6] by Poppe were published. These works are focused on both lower and higher level processing steps and form a complete overview of the state of the art in that time. The works give a lot of attention to methods of representation, pose estimation and classification and introduce pose-based classification as a particular point of interest. In *A survey of vision-based methods for action representation, segmentation and recognition* [72] published in 2011 by Weinland et al., different spatial action representations (body models, image models, sparse features) and temporal action representations (grammars, templates, key frames) are discussed as well as methods of action segmentation. Special attention in this work is given to view-independent action recognition, a topic of particular interest for this report. It discusses advancements in topics such as normalization, view invariance and exhaustive search for both 2D and 3D.

After this date, much less 2D action recognition surveys of importance are published. One last interesting survey dates from April 2014: A survey on still image based human action recognition [74]. This survey explores action recognition without the availability of temporal information. Especially useful is their table that shows which papers employ which high level cues and which cues are described for the presence of an action.

2.6.2 Methods in Literature

The most widely applied representation in 2D action recognition throughout the years is a holistic representation: the silhouette. As described this is a very rich representation that is not very susceptible to changes in appearance and its success depends on the accurateness of a preliminary segmentation process. Although this representation is not well suited for applications with a moving camera, it is a widely used representation. Bobick and Davis were one of the first to use silhouettes in their research to action recognition[34]. Silhouettes can be generated using several methods, of which the most well-known are based on foreground-background segmentation and the use of motion detection. The latter is used to generate binary motion energy images (MEI). Another representation introduced by Bobick and Davis is the motion history image(MHI). In this representation the pixels are an indication of the recency of the detected motion or silhouette motion. Examples of MEI and MHI are given in Figure 14.

Silhouettes, MEI and MHI can be used in the recognition process in many ways. Contours can be generated out of the silhouettes, facilitating e.g. the fitting of star skeleton models or the extraction of many types of other feature descriptors. Other approaches include, but are not limited to, matching silhouettes with silhouette templates, matching average silhouettes with its respective templates. With matching is meant to determine the 'distance' between the object and template using one of the available distance functions: the smaller the distance the better the match. Examples of distance functions are Euclidean distance, chamfer distance and Mahalanobis distance but basically any sensible definition can be used as a distance measure.

Silhouettes are also used in more complicated methods. For instance, silhouettes are used to make a system more robust to variance of viewpoints. For example, silhouettes extracted from multiple camera images can be used to create visual hulls [75] or 3D voxel representations such as the 'stacked' in time space time volume (STV). In the work *Action as Space-Time Shapes* [76], Gorelick et al. regarded human actions as a three dimensional shape created in space-time by stacking subsequent silhouettes on top of each other. They even showed that their method was robust to significant changes in scale and viewpoint, partial occlusions, and low quality input data. Related to STVs are motion history volumes (MHV) [77]: while STVs are generated by stacking silhouettes on top of each other MHV are generated by stacking motion information obtained from object-based optical flow. These types of constructed 3D surfaces allow for extraction of 3D points of interest but are also suitable for template matching as is done on the 2D silhouettes directly.

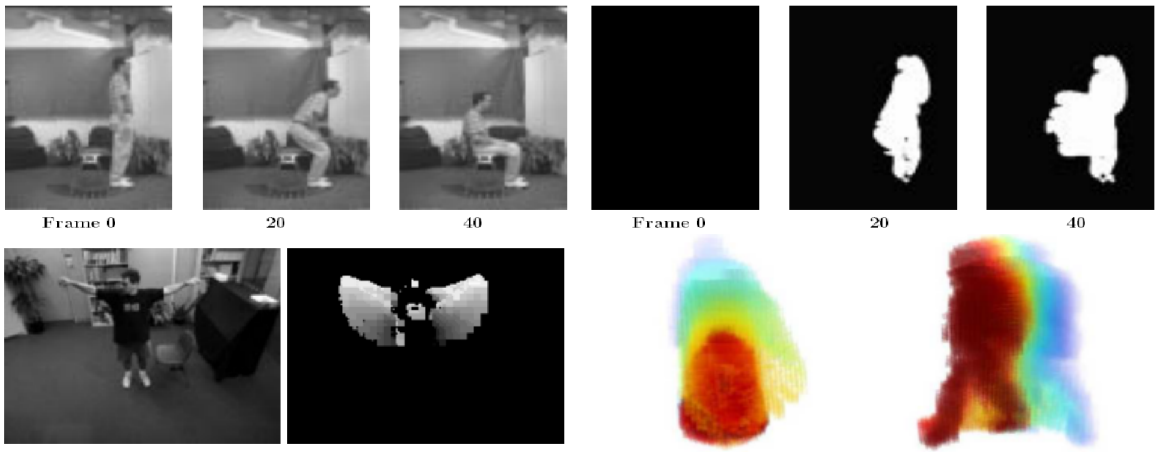

Figure 14 Top row plus lower left: the creation of MEI and MHI representations. MEI is a binary cumulative motion detection result, MHI indicates the time since the motion at a certain image position was detected. Lower right: example of a motion history volume for actions 'sit down' and 'walk'.

The other representation that is often used is the local / patch based representation which uses patches that are typically placed around so called 'spatial temporal interest points' or STIP. Laptev and Lindeberg [78] were among the first to propose a temporal extension towards already successful feature detectors in the fields of object recognition and image matching using a Harris-Laplace based space time interest point detector. STIP can be obtained directly from images or from intermediate representations such as STV or MHV. There exist many types of detectors that could extract these interest points such as HL-STIP, HES-STIP; some detect only a sparse set of features while others try to create very dense sets of feature points [79]. The theoretical principles and the different algorithms available for STIP based representations are outside the scope and thus will not be discussed in this report. In short, interest lays on these points with significant local variation of image intensities, assuming that these points are most informative for the recognition of human motion.

Since the amount of local descriptors often varies from frame to frame and sequence to sequence for the STIP representation, descriptors cannot be compared straightforwardly. To cope with this issue the bag of words models (BOW) can be used. A codebook is generated by putting each descriptor in separate bins, where each local descriptor is a code word. A BOW model is thus basically a histogram of code word frequencies and functions as an intermediate mid-level representation. BOW as an intermediate representation is used not only in methods where it facilitates STIP representations, it is also used as an intermediate representation for poses. This is shown in the work of Hatun [80] where an action is considered to be a document consisting of words, where each word corresponds to a certain pose in a frame. The work of Tahayna et al. The work of Tahayna [81] is another example of using a BOW representation to facilitate detecting and recognizing actions: STIP extracted from STV are put in a BOW representation which is used in a genetic algorithm based SVN-classifier. In this work special attention is given to the optimal representation of the BOW, as choices regarding dimension, selection and weighing of words turns out to be of crucial importance to the eventual classification performance.

Next to the use of silhouettes and local patches, a lot of attention has been given to 2D pose based approaches. In early works, focus within pose estimation was on tracking body parts and determining joint movements, such as in the work of Gavrila et al. [82] in 1995. These methods relied heavily on tracking individual body parts, which proved to be an extremely difficult task in realistic settings. Attention to pose recognition however never faded away completely due to the fact that this representation has several major advantages: they are invariant to changes of viewpoint, differences in appearance of the humans and contain only high level information. The first two advantages result in much lower intra class variance, the last advantage simplifies learning for the process of action recognition. Another benefit is a reduced sensitivity to occlusions.

Figure 15 A few examples of relational pose features, obtained from [83]. From left to right: Joint distance, distance joint/plane spanned by black dots, another type of joint/plane distance, velocity component in the direction of a certain other joint, velocity in the direction normal to a certain plane. Many options are possible.

After years of research in the field of pose estimation (e.g. [84], [85]), in 2011 Yao et al. published the article in the form of a question: *Does human action recognition benefit from Pose Estimation*? [83] The answer to this was *yes*, as they showed in their comparative work. The use of poses in action recognition is also demonstrated in the work of Poppe in 2009 [63] where a pose recovery approach is combined with a common spatial patterns classifier to recognize human actions.

Poses as representation for the human action recognition can be used in multiple ways. In general, poses are not used in action recognition by comparing their joint locations in space and time. Instead, relational pose features [86] are extracted for the classification process. Some examples of relational pose features are given in Figure 15. These features can be used in models employing hidden Markov Limb modelling, or one of many other models, many of which are discussed in [6].

In 2008, Zhang et al. present their *Expandable Data-Driven Graphical Modeling of Human Actions Based on Salient Postures* [64]. This work presents a graphical model for learning and recognizing human actions in a so called action graph. An example of an action graph is illustrated in Figure 16. An action graph has nodes that represent salient postures and weights that correspond to transitional probabilities between two salient postures, or nodes. An action can therefore be encoded as one or multiple paths within an action graph. The proposed method turns out to be an effective way to recognize actions, and states that it can easily be expanded to include new actions. Furthermore it is robust to noise and changes of viewpoint. This model, combined with advanced pose recognition, is currently the state of the art method in 2D (and 3D) action recognition.

Figure 16 Example of an action graph for the action 'run' together with 8 salient postures, potentially used in the action graph.

2.7 Advancements in 3D human action recognition

Using 3D information in the field of computer vision has always been a topic of interest, as it comes with a set of interesting benefits. Within the field of human action recognition, 3D information can provide additional information about the body shape. This helps to differentiate actions that would otherwise have an identical or similar 2D representation in a single recorded frame. This additional information resolves part of the ambiguity problems that exist in this field of research, since combining RGB with depth data leads to much richer representations which can be used for feature extraction and classification. Furthermore, the use of depth data removes some difficulties existing in lower level processing steps such as segmenting similar foreground/backgrounds and dealing with variances in appearance and illumination. This section discusses the different methods to obtain depth maps, different representations and several popular techniques used in 3D human action recognition.

2.7.1 Acquiring depth information

The recent advancements in the techniques to obtain 3D representations such as depth maps, facilitated the advancement of techniques within the field of 3D human action recognition to a great extent. Consumer-targeted RGB-D sensors are capable of obtaining depth maps more reliably in many situations with more than reasonable precision although this has not always been the case. Before the introduction of the Kinect other techniques to obtain depth maps were used which can be divided in roughly four different groups: stereo triangulation, structured light, coded aperture and time of flight (ToF). These groups are a topic of interest to this day. Most of these groups employ methods in the spatial or temporal domain, although some use features from both domains[33], [87].

Stereo triangulation [88] is one of the first methods used to generate depth maps. It is a technique in the spatial domain, employing two cameras set up with a small horizontal offset to acquire a pair of slightly different images at the same time. Stereo triangulation can be used as an active or passive method. Passive stereo triangulation is beneficial since it is non-invasive while it attempts to find matching image features between a set of images, usually without any prior knowledge on the observed scene. The benefit of active stereo matching is that it minimizes the difficulties in finding occurrences between the two images. This is done by replacing one of the cameras by a projector that projects patterns of (invisible) light on the observed scene, also called structured light. These projected structures are then observed by the camera of the system.

Structured light is a technique in the spatial domain and it appears in many forms. Fringe and Moiré based techniques [89] as well as several types of stripes and checkerboard patterns have been used as projection methods in order to estimate the 3D shape of objects. Recently, advancements in these techniques were discussed by S. Zhang in [32]. In general, to achieve real-time performance, the structured light is changed rapidly while the system monitors the deformation of the structured light, hereby calculating the shape of the object causing the deformation. Microsoft's Kinect also uses structured light [15], [31]. It uses a pattern of projected infrared-points, in the form of uniquely positioned groups of dots for every location in the projected structure. The Kinect's infra-red camera captures the projected dots and by retrieving the location of unique groups of dots within the recorded image the depth of the groups of points can then be calculated by stereo triangulation, unhindered by burdens as occurring with conventional passive stereo triangulation. Another example of a popular system using this method was Asus' Xtion Pro. Unfortunately, Apple bought and discontinued PrimeSense, and this product is therefore scarcely available.

The third group of techniques use coded aperture [30]. It is a passive method in the spatial domain that uses apertures that are occluded in a special pattern. With such a coded aperture and using a statistical model of images, both depth information and an all-focus image from a single photograph can be obtained. The resulting depth map is a layered representation of depth. This method is perhaps less suitable for human action recognition as it operates in relatively narrow depth ranges and depends on specific focus and exposure settings, making it less robust for different illumination settings.

Figure 17 Person with a projection of a sinusoidal grayscale banded image (left), the result after processing (center) and an example of Kinect's speckle structure (right).

Finally, ToF is an active system operating in the temporal domain. Typically, by measuring the time it takes for emitted light to be reflected back, depth can be estimated. This is not necessarily limited to emitting light, as other examples of ToF are radar and LIDAR. ToF systems using emitted light were in general extremely expensive and often had limited resolution; recently however products such as the D-Imager of Panasonic and the depth Sense from the Soft Kinetic came to market. These consumer-priced devices are capable of generating high resolution depth maps by combining (fusing) lower resolution ToF with stereo matching high resolution RGB data [33],[90] .

Another technique was revealed in August 2013 by MIT. It is a technique called Wi-Vi. Although it has almost nothing in common with the Kinect it is announced as the 'next generation Kinect that can see through walls' using Wi-Fi signals to enable the detection of moving objects (even through walls), track a human body and recognize simple gestures. However, there is not enough literature at this time to make any assumptions on the usability for accurate human action recognition, as the technique is premature

Figure 18 Upper row: example image from [30] showing an RGB scene(left), aperture coding pattern using a thin cardboard layer(center), and the resulting 'layered' depth map (right). Lower row: Kinect, depth Sense and Xtion.

2.7.2 Representations in 3D space time

Depending on the used hardware, a sequence of depth maps can be obtained, which in some cases comes together with a calibrated RGB image sequence of the same scene. This RGB data provides additional information. As described, a depth map is an image representation showing per pixel the depth distance of that particular image location to the sensor. Depth maps are a representation form that can be used for low level pre-processing steps such as background subtraction and motion detection using methods as well. The usefulness of this representation is shown in many works, for instance in the field of health care and elderly fall detection where Rougier et al. [12] and Zhang et al. [13] use depth maps to segment and locate a person from depth maps. A survey on human fall detection is available in [11].

Another commonly used representation is called 'Point Cloud', which is a collection of points with real world coordinates that have x, y and z coordinates relative to the sensor. A point cloud was until recently usually obtained by Lidar and other laser scanners, and functioned as a high quality representation of the real world. Another system that acquires point clouds is the Kinect. In fact, the earlier mentioned depth map is only a projection of the point cloud on the XY plane.

Depending on the hardware used and the detail that is needed in a representation, a point cloud may contain a very large set of 3D points. To help developers of 3D vision applications deal with these point clouds efficiently, a point cloud library or PCL was created [91]. PCL helps developers with low and medium level processing tasks including filtering, feature estimation, surface reconstruction, model fitting and segmentation. Other popular software packages offering a variety of drivers and tools are Microsoft Kinect SDK, OpenNI and NITE and LibFreeNect.

When observing a human being, acquiring low level representations such as depth maps and point clouds offers opportunities of obtaining a higher level representation: a 3D skeleton. This representation is also called a 'pose' representation, since the estimated skeleton determines the pose the observed human is in. Although 3D representations facilitate estimating the correct skeleton's joint angles and body part lengths from 3D information, 2D representations have also been used to estimate joint angles [82], as discussed in Paragraph 2.6.2. Another method to obtain motion of joints is using a motion capture systems using markers.

Fast and accurate extraction of skeletons is one of the core points of the internal Kinect software, and software packages such as NITE. Extracting poses is however not something that should be thought of lightly. For instance, problems that hinder the development of reliable, accurate, robust extraction of joint angles and location are e.g. (re)initialization problems, difficulties tracking the skeleton, large occlusions and limited resolution. This still leads to a lot of accuracy issues.

The skeleton representation is in most cases a holistic representation which is extracted from the available data in point clouds or depth maps, using a variety of low level processing techniques. In some cases however a skeleton is formed from a patch based representation. In these cases an intermediate representation is used: the body part representation. This however requires the recognition of body parts and their orientation, regardless of their appearance.

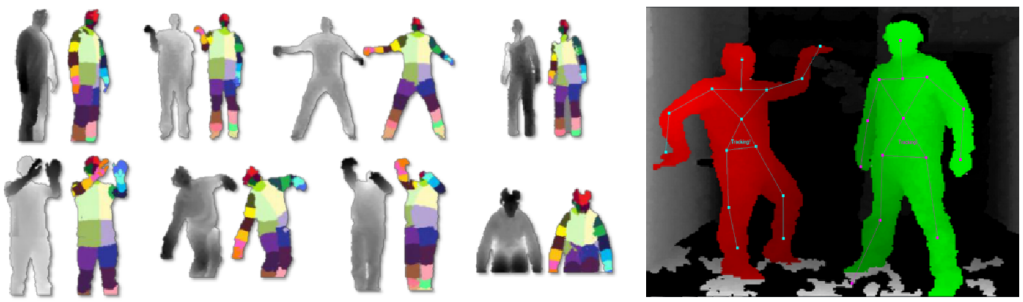

Figure 19 Left: body part labeling on real data as described in [53]. Right: Example of tracking (multiple) skeletons in the NITE & OpenNI framework.

A well-known, state of the art method is published by Shotton et al. in their work *Real-time human pose recognition in parts from single depth images* [53]. This work forms the core of Kinect's skeletal tracker, which is able to extract skeletons of humans observed in the scene (almost) regardless of their size, orientation and pose, including a wide variance of human body appearances. The biggest reason of success for their approach is that they employ a body part recognition algorithm as intermediate representation, combined with a very large sample database for training the system with synthesized data[92]. Once the body parts are recognized, the system proposes body joint location and angles based on confidence scores. Beneficial is that it runs very fast (200fps at consumer hardware) and it does not use temporal information; instead it extracts the skeleton from only one depth frame, eliminating the issues that would otherwise be involved with tracking and initialization problems occurring in other methods such as used by OpenNI that perform initialization of tracking the skeleton by requiring the subject to stand in a characteristic T-pose. Other recent works in the field of pose estimation are [71], [93], [94], the latter being an interesting article using pose refinement to obtain joint locations reliably.

2.7.3 Action recognition in 3D space time

The skeleton representation is the most used representation in action recognition nowadays. A well-estimated skeleton is especially useful since it can be made view invariant and it is robust to changes in appearance, occlusions and noise. Note that the reliable extraction of postures and the recognition of actions using the estimated poses are two separate research domains, although they are intertwined. Many researchers in the field of action recognition therefore assume skeletons or poses to be extracted reliably beforehand, allowing them to only focus on the task of human action recognition, without being 'bothered' by lower level process steps. This section discusses recent advances in pose based action recognition as well as a number of other techniques.

2.7.3.1 Pose based approaches

Once a reliable skeleton representation is estimated from the observed data, it can be used in many ways to perform action recognition. Pose based action recognition is not a new technique as it is used in the 2D domain as well, although obtaining a proper pose representation is often troublesome. The usefulness of 3D poses for view invariant action recognition is shown in an extensive dissertation by Poppe [63]. Some action recognition techniques try to find differences within the representations and are discriminative in nature by trying to find more compact and discriminative features. Others try to exploit the sequential nature of all the small movements that form an action. These methods often try to cope with temporal variations by using dynamic time warping (DTW) as explained in paragraph 2.4.2.

In the work of Poppe [63], a method to extract 3D joint information from single static 2D RGB camera images is proposed. In the second part of the dissertation an action recognition algorithm is presented based on common spatial patterns (CSP). More interesting is the combination of the two in Chapter 6 of the work, where human action recognition is performed based on recovered poses from 2D images. Promising results were shown and as suggestion for future work: the detection of new poses with unknown classes as a topic of interest.

There are quite a few recent works that are discriminative in nature and use joint locations. Only a few recent works will be mentioned to illustrate the general idea of discriminative methods. In the 2012 article by Xia, Chen and Aggarwal [95], 3D Joint Locations are projected in a special spherical coordinate system to obtain view invariance. Subsequently, they are clustered in histograms of orientated joint locations (HOJ3D) and clustered in visual words (BOW) representing the prototypical poses of the action. The temporal evolution of the visual words is then modelled using Hidden Markov Models (HMM). Results are promising and the system is suitable for running online in real-time.

In 2013, Theodorakopoulos et al. [96] propose a *Pose-based human action recognition via sparse representation in dissimilarity space*. In this work a skeleton is post processed to increase robustness to variance in viewpoint first, after which classification of actions occurs by extracting feature vectors containing dissimilarities to a set of prototypical actions and then using sparse representation based classification. Again, HMMs are used to model the temporal evolution of the extracted features. Yang et al. described in 2013 a novel action recognition method [26] using an action feature descriptor called an *Eigen joint* that combines action information such as static posture, motion property, and overall dynamics. A Naïve Bayes Nearest Neighbour classifier is used to classify different actions, as it is capable of dealing with a large number of different classes and is not prone to overfit. The main contribution of the paper is the new *Eigen joint* feature to be used in discriminative classification, created by taking the differences in both the spatial and temporal domain and using dimensionality reduction te4chnique called principle component analysis (PCA) to remove redundant and noisy features.

To illustrate works that try to exploit the sequential nature of actions a number different examples are given as well. In 2008, Zhang et al. present their *Expandable Data-Driven Graphical Modeling of Human Actions Based on Salient Postures* [64]. This work presents a graphical model for learning and recognizing human actions in a so called action graph. An action graph has nodes that represent salient postures and weights that correspond to transitional probabilities between two salient postures, or nodes. An action can therefore be encoded as one or multiple paths within an action graph. The proposed method turns out to be an effective way to recognize actions, and states that it can easily be expanded to include new actions. Furthermore it is robust to noise and changes of viewpoint. This model, combined with advanced pose recognition, is currently the state of the art method in 2D (and 3D) action recognition. An example of an action graph is given in Figure 20.

In 2012, Lin et al. propose a novel representation called action trait code (ATC) [97], used in combination with a graphical model to learn and recognize human actions. Action trait code uses the average velocity of body parts to form a feature vector describing the actions. The method is applicable in real-time and show very good results. Unfortunately, no further literature can be found using ATC, but in basis the method is 'just' a graphical method with a certain chosen feature used as intermediate representation. In the article published in 2012 by Jalal et al. [98] HMMs are trained using a codebook representation, exploiting the sequential nature of the actions by learning probabilities of transitions from one occurrence to another for different actions. Although the poses are represented in a body part based fashion, it could also function for normal skeleton based approaches. Superior recognition rates were obtained with +/- 20% higher recognition rates than conventional methods available at that date.

2.7.3.2 Bag of Points and STOP

Not all action recognition methods in the 3D domain use skeletons. In [25], Li, Zhang and Liu present their method which uses an action graph to model the actions explicitly and they use a bag of 3d points to characterize a set of salient postures. These salient postures are representing the nodes of the action graph. They project the dense 3D point clouds on different orthogonal planes and use a smart sampling scheme that samples only 1% of the original points to be used as a salient posture representation. A Haussdorf distance is used as similarity measure. The 3D coordinates belonging to these points are then fed into a GMM to capture the statistical distribution of these points effectively. While reducing the amount of information to be more computation friendly, still impressive recognition rates are achieved. A drawback to the method is that it is susceptible to changes in viewpoint and variance in appearance. Also, self-occlusions might lead to other orthogonal projections and therefore completely different sets of sampled points, hindering the classification phase.

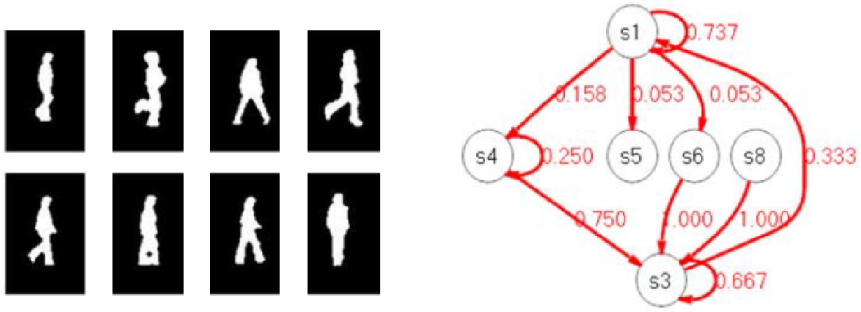

Figure 20 Action graph wih salient postures (left) as nodes.

Another method that does not require 3D joints to be estimated was proposed by Viera et al. in 2012. Their method, *Space-Time Occupancy Patterns for 3d action recognition from depth map sequences (STOP)* [27] looks a lot like the use of motion history images in the 2D domain for classification, but instead of a 3D (2D+time) representation they employ a 4D (3D+time) representation. In contrary to MHI classification however they do not use the direct history of the scene but rather the saturation of time blocks: the average occupation of points in a certain positioned block in space time. The claimed benefit is that their method preserves spatial and temporal contextual information while still being flexible enough to accommodate intra-action variations.

2.7.3.3 Key points, Action Points and action constraints

The discussed STOP method in paragraph 0 updates the action graph with new information as soon as a non-neutral pose is discovered. This is useful since this is a direct cue for the online action recognition algorithm that an action is starting to be performed, or as a cue to indicate a certain characteristic intermediate position in a sequence of movements corresponding to an action. These cues exist in other research as well, although usually they are not based on being non-neutral but rather on their discriminative nature: in sports such as tennis there exist many characteristic poses that are an occurrence indication of a certain action by their appearance alone. In the field of human action recognition, these cues are often called key poses [99]–[102], key postures or salient postures. A variance to this concept is given in [16] where the concept of Action Points is presented. Most works on human action recognition using key poses in the last decade had their focus in the 2D domain. However, since poses can be extracted in the 3D domain more reliably, key poses are discussed in this part of the report as it is expected that future research using key poses obtained from 3D data will lead to promising results in the near future.

Given a set of available actions, a human is in most cases able to recognize a certain action from a single frame, without examining the whole action sequence. This is the base behind key poses. Key poses often do not incorporate any temporal information and rely only on spatial information available in the scene. In an article by Baysal in 2010 [99], a key pose is defined as 'a set of frames that uniquely defines an action from all others'. For each to be classified action sequence all frames are labeled as one of the available trained actions by measuring the similarity with the key poses of these actions. After processing the entire sequence, it is classified to the action it received most evidence for.

This method requires action sequences to contain many key poses, as recognition rates are relatively low for low amounts of key poses. Although this work is not robust to variance of viewpoint, another work using key poses does try to cope with these variations [100]. It uses a database of synthesized key poses from different view angles to match new poses with. Furthermore it uses an action representation scheme called 'action net'. This allows the researchers to include three separate action constraints: the occurrence of key poses should be in a certain order, the transition between actions should not be arbitrary - you cannot walk without standing up first if you are sitting - and finally changes in orientation of the subject should be smooth. The last constraint is not explained but it is assumed that it means that key poses that differ too much are excluded.

Unfortunately, classical methods using key points are not very suitable for online action recognition. This is due to the fact that reliable recognition requires long action sequences to match many key poses, plus the proposed methods assume proper segmentation on forehand which is usually not the case in an online setting. An alternative, recently published method is published by Nowozin and Shotton in their work *Action Points: A Representation for Low-latency Online human action recognition* [16], where the concept of Action Points is proposed. In a very recent review, *Enhanced Computer vision with Microsoft Kinect Sensor,* published in October 2013 by Han and Shao. [31], the work is pinpointed to be the potential start of a research trend, as it offers recognition at extreme low latencies which facilitates online human action recognition.

Compared to key poses, action points are more closely related to the idea that an action can be characterized by a single frame. A major difference between them is that action points include temporal information: Apart from describing the pose itself only spatially, it also describes how the actor got into the pose. Action points only rely on information from the past and since they are designed to define a clear presence of an action, the action is detected and recognized as soon as Action point is recognized. An example on annotation is given in Figure 21. The proposed method in [16] is therefore a low-latency, real time, online action recognition system that does not require prior action segmentation.

One problem that is addressed in their work however is that current state of the art classification methods are not really suitable for classification using Action Points. Therefore the article proposes a performance measure and two new classification methods tailored for Action Points; the first using a variant of a HMM with an explicit 'firing' state marking the Action Point and the second using randomized trees to achieve direct classification. Results were promising and showed a direct relation of the accuracy – latency tradeoff. Downside of the method is that it is limited to actions that are momentary, voluntarily performed and discrete in nature. It is less suitable for continuous actions such as walking.

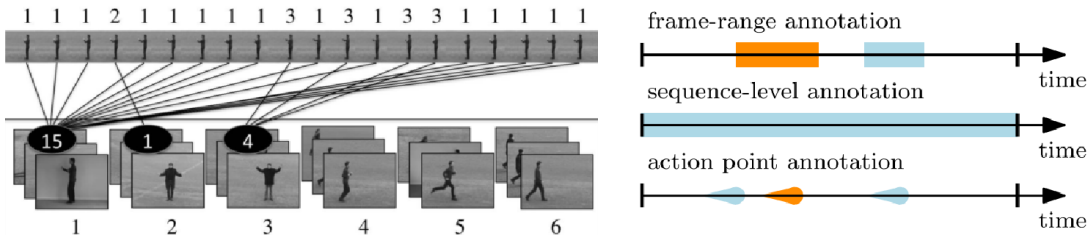

Figure 21 Left: classifying an action based on key pose majority voting. Right: annotation of action points.

2.8 Explicit use of time as 4th dimension by using a Space Time Pattern

Apart from action recognition in 3D, there also exist methods that in addition to the three spatial dimensions encode time in their representation explicitly as a 4th dimension. One of these methods is the space time pattern (STP). STOP, introduced in the previous section, is an example of how such an STP can be used. But what exactly is this STP? This section explains what an STP is and how it can be used in the classification process.

In essence, an STP is a method to capture the location of an observation in both space and time. As seen from its basis: a normal RGB camera captures the 2D locations of objects in one time instance and a video camera is capable of recording the locations of these objects during a certain amount of time frames. The STOP method and the novel method introduced in this work discretize the dimensions of space time into a certain amount of bins which are focused on the input data corresponding to a certain space time area only. For instance, a typical 3 dimensional STP 'space' is focused on a 2 x 2 x 2 m area, centered on e.g. a depth recording of the subject, to be able to capture the extremities of the subject's body. An STP can then be built by using multiple-frame recordings of the observed area, which can be placed in multiple 'time' parts. The resolution of the example shown in Figure 22 is one tenth of the area dimensions, and three time segments, leading to 10 x 10 x 10 x 3 = 3000 4D space time cells. The total set of these locations is called the space time pattern.

Mathematically, when using depth data or point cloud recordings, the construction of an STP in the STOP article can be described as follows; Consider a recorded action sequence a, consisting of F frames, with a total of N recorded 3 dimensional points in the entire sequence. Each frame f contains a large set of 3 dimensional points $p_n = (x_n, y_n, z_n)$ which means that each individual point in the sequence can be denoted with $p_n = (x_n, y_n, z_n, t_n)$, where t_n denotes the frame number of that particular 3D point n. The total recorded action sequence a can thus be described by the set of points $A = \{(x_n, y_n, z_n, t_n), n = 1..N\}$.

The STP itself can be described as a 4 dimensional grid B, partitioned in $x \cdot y \cdot z \cdot t = m$ space time cells where $x \cdot y \cdot z$ denote the amount of bins per space dimension and t the amount of time bins. A space time cell is denoted with c_i and the total set of space time cells is called a partition $C = \{c_1, \dots, c_m\}$. The amount of depth points in A belonging to the individual cells c_i of C can then be determined using the intersection $D_i = A \cap c_i$. The total set of depth point cells $D = \{D_1 \dots D_m\}$ can then be visualized, such as seen in Figure 22.

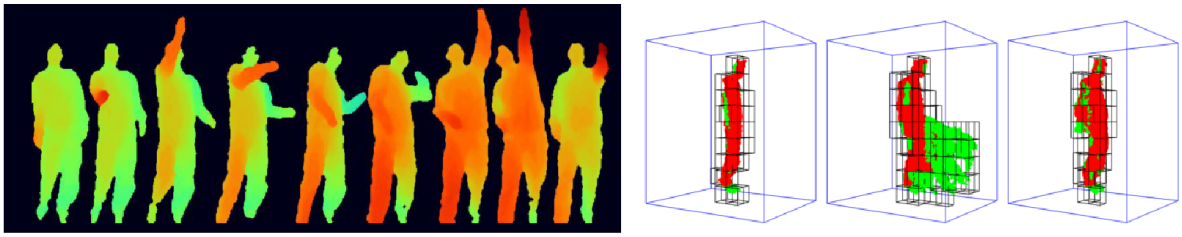

Figure 22 Left: tennis serve in a sequence of depthmaps. Right: an STP showing three space time segments or of the action 'forward kick' from the MSR action dataset. The time segments are created by adding multiple space recordings to each time segment, which leads to the shown presence of many (red) or few (green) points at the level of individual space time cells.

The described representation of $D = \{D_1 \dots D_m\}$ however is not necessarily created in an informative way. Although it does tell how many 3D depth points have been seen in all time cells for the entire sequence it has a preference to increase the importance of body parts that are clearly visible; a hand that remains in the same space time cell leads to a lower point count than a torso, which is showing much more points as it is both larger (more points) and contains less boundaries (less noise). This is unfortunate, as the hand is most likely equal or more informative about the observation than the torso. Furthermore the method is highly dependent on the amount of frames in the recorded action sequence: more frames leads to more points being added to the individual cells.

Even when normalizing over the frames to reduce point differences between different cells, the importance problem persists: cells containing large number of points still remain higher valued than cells with lower amounts of points in their cells. The STOP paper therefore introduced a saturation scheme and called the resulting values of each space time cell after some calculation the 'occupancy' of this position, hence the name space time occupancy pattern. The occupancy of a space time cell is defined in the formula shown in Figure 23. By choosing a certain value for p, both the static hand and torso then lead to a value of 1, while captured movements leading to lower amounts of points still remained distinguishable from non-moving body parts. The dependence of this value van be seen in Figure 23.

The STOP method is designed to be used in both offline and online settings. For offline recognition well segmented actions are fed into the system, which is trained with well-segmented training samples. Classification is done with a simple cosine distance classifier. Although the report shows very good results it remains questionable how well the system would perform in less ideal situations. For online recognition, very short sequences of depth maps are converted to lower dimensional PCA-STOP features using OCL as dimensionality reduction technique, to train a neutral pose classifier using SVM. As soon as a non-neutral position is detected, an action graph was used to perform classification, which performs well in online situations. However it remains unclear how STOP would perform in situations where a non-static camera is used, or in situations where the orientation of the action differs from sequence to sequence.

$$P(c_i) = \begin{cases} 1, if\ |D_i| \geq p \\ \frac{|D_i|}{p}, otherwise \end{cases}$$ [1]

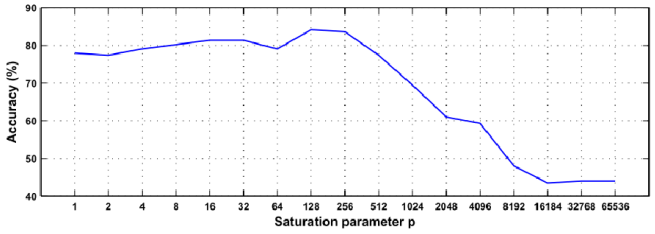

Figure 23 Empirically shown in [27], the accuracy rate depends on the saturation parameter.

[1] note: $|S_i|$ indicates the cardinality (number of elements) of the set of points described with S_i

2.9 Obtaining features from an STP

Regardless of using the STOP or the novel method that is proposed in this work, the way features are constructed for the training and classification processes is identical. The method is of holistic nature to preserve the spatial and temporal relations: the entire STP can therefore be seen as one large feature. To visualize this, recall that an STP is composed of a set of $x \cdot y \cdot z \cdot t = m$ space time cells where x, y, z, t are the amount of bins per dimension. The feature vector can then be seen as one large vector of dimension $[1 \times m]$ wherein all cell values $M(c_i), i = 1..m$ are placed. This results in feature vectors that have a very high dimensionality. For instance, an STP with 30 bins in x, y and z direction and 3 time segments has a very large dimensionality of 81.000.

This dimensionality however is problematic: apart from the fact that it will increase computation times dramatically, due to the curse of dimensionality the feature space tends to get extremely sparse and accurate recognition gets increasingly difficult. Also, due to the high dimensionality, classifiers that rely on distance measures loose performance, as dissimilarity measures such as Euclidean and Mahalanobis distance of samples are less effective in high dimensionalities. In general: the smaller the amount of training samples available, the lower the amount of features to be used for successful classification.

This problem can be overcome by realizing that the feature space is (in general) very sparse and that there exist strong relations between the available features. For instance, the captured motion of an arm being risen in the air results in a set of space time cells in C that are 'touched' by the set of skeleton pixels $S_f \cap c_i$ intersecting these space time cells. Not only the wrist joint touches space time cells, the elbow joint touches different ones as well. This implies there is a relation present between several cells during an action and therefore most of the features are likely to be redundant or non-informative (zero) and only some features are important to describe the recorded action accurately.

It is therefore very convenient that many techniques exist that are able to reduce the dimensionality of such a feature vector [103]–[105]. Probably the most well-known technique is principle component analysis (PCA). Using PCA, one is able to express the data in such a way as to highlight their similarities and differences[106], allowing to reduce the dimensionality. This leads to a mapping of the feature vector to a new, lower dimensional feature space, which effectively removes redundant and noisy information from the feature vector. As stated in the work of Burges [105], there are many different versions of PCA. An interesting variant of this method is a modified version of the PCA which is called orthogonal class learning (OCL). This method was introduced in 2012 by Oliveira et al. [107]. This method is especially useful when the original feature vectors are sparse and of high dimensionality compared to a relative low amount of available samples. OCL constructs a new base with a dimensionality equal to or lower than the amount of available samples, in which the original features are represented. Advantageous of OCL over classic PCA is that after the mapping towards a lower dimensionality, the samples belonging to different classes are always mapped orthogonal to each other which with some classifiers will lead to higher recognition rates.

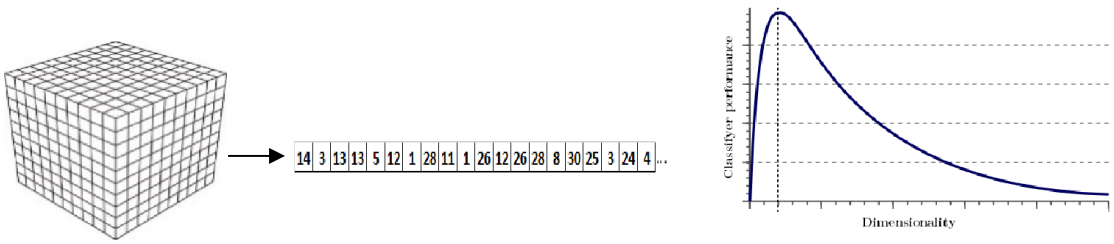

Figure 24 Left: STP with 1 time segment: already 1.000 dimensions! Right: curse of dimensionality depicted in a graph: although counterintuitive, increasing the amount of feature descriptions does not necessarily lead to higher recognition rates.

2.10 Comparing articles

Author / ref	Year	Dim	Mode	Main Representation	Method	View point Invariant	Actor Invariant	Scalable[1]	CPU
R. Poppe[63]	2009	2D	Offline and Online	Skeleton poses	Pose sequences, CSP	+	0		+
W. Li, Z. Zhang, Z. Liu[25]	2010	3D	Offline	Projected Silhouettes	Bag of points, action graph	+	-	-	+
S. Nowozin, J. Shotton [16]	2012	3D	Online, Real-time	Skeleton poses	Action Points, F-HMM	+ +	+ +	+	++
I. Theodorakopoulos et al. [96]	2013	3D	Offline	Skeleton poses	Classification in dissimilarity space, HMM	+ +	+ +		
X. Yang, Y.Tian[26]	2013	3D	Offline and Online	Skeleton poses	Eigen joints, Naïve Bayes NN	+ +	+ +	-	-
S. Lin et al.[97]	2012	3D	Offline and Online	Skeleton poses	Action trait code, action graph	+ +	+	-	
A. Jalal, S. Lee et al. [98]	2012	3D	Offline	Skeleton poses	Body parts feat. Extr., HMM	+	+		
A. Viera et al. [27]	2012	4D	Offline and Online	STOP-PCA	SVM, action graph	?	+ +	-	++

1 How suitable is the method when many classes are involved

2.11 Summary

Human action recognition as a field of research has been popular for decades and keeps gaining more interest to date. Over the years, many vision based techniques have been developed that can be used in support of this research field. Human action recognition operates in both 2D and 3D domains, the latter getting more and more interest since the introduction of reliable and accurate mass-consumer priced RGB-D. Depending on the domain of operation, different representations can be chosen, either holistic or (body) part based. The chosen representation is a careful balance between reducing variances and at the same time being sufficiently rich in high or low level information while making sure that the representation offers the data in such a way that it can well be used for the rest of the action recognition process.

From the representation of choice, features are extracted. These features can be very high dimensional and in these cases it is best practice to make use of dimensionality reduction techniques such as principle component analysis and orthogonal class learning. Depending on the representation, features can be built up in many ways and are used in both discriminative as well as generative classification methods. In short, discriminative methods compare a sample feature set with template/exemplar feature sets obtained from known classes and classify the sample to the class that seems to be most alike, depending on the distance function used. In generative methods different class models are trained and the probability that a certain class model would produce the sample feature set is determined. The sample is classified to a certain class if it has the highest probability.

In the current state of the art, skeletal feature vectors are most commonly used, either directly extracted or via intermediate representations such as histograms of orientated joints, bag of words or bag of points. These intermediate representations are used in generative classification of which methods employing graphical models such as the action graph are the most popular one, as they can be expanded easily, are suitable for real time classification and do not require many training samples. These models try to exploit the sequential nature of poses that build up actions by statistically determining the transitional nature from one graphical node to another. In these action graphs, the nodes resemble the intermediate pose representations.

Research so far aims at the recognition of actions from pre-segmented data. There is however a trend towards low-latency systems that are able to quickly recognize an action in real time. This is especially true in action recognition systems with an application in health care: dangerous situations need to be recognized without too much delay. Online systems are therefore a topic of interest, as lots of applications could benefit from real time recognition. The best methods currently available are action points [16], bag of points [25], Eigen Joints [26] and STOP [27].

Another point of attention is that developments towards e.g. the monitoring of patients and elderly require action recognition systems to work independently of their orientation and location. This too is a challenge for future research as most researchers still focus on fixed camera systems while it could well be that future systems would be autonomous machines moving around in our homes. Steps to correct or cope with ego movements are therefore a topic of interest in research as well although existing solutions are still far from perfect.

Chapter 3

New approach towards a safer living environment for elderly?

How to create an action recognition system that is able to detect and recognize potential dangerous actions of an elderly person? The third chapter of this thesis describes the approach of this work, which ultimately forms a possible answer to the research question and offers a contribution to a safer living environment for elderly in the near future. First let us investigate some details of the application field.

3.1 Background information; important aspects and challenges

Currently the elderly population with the highest risk to perform dangerous actions (of a certain 'zorg klasse') is mainly living in elderly care centers. Typically these people are having physical issues such as limited mobility or psychological issues such as Alzheimer's disease. In this group of mainly 65+ year old people, one of the biggest problems is the risk of falling. Fall incidents lead to very high medical costs due to the high risk of fractures which are often very expensive to treat [108]. However, falling is not the only dangerous action an elderly person can perform and what is dangerous might differ from one person to another.

As the elderly population increases the budgets of public health care will come under increased pressure and more and more elderly will be forced to stay in their own homes for a longer time, with a higher risk of performing dangerous actions. A system able to detect dangerous actions would improve the safety of these people in elderly care centers but especially in the private setting of their homes. It is therefore inevitable that an action recognition system will have to be extremely robust to the many different types of variance that is present between different home situations. Also an action recognition system should not be too dependent on its placement. For instance it is preferred that a system can be installed with great ease on any fixed place, that it functions independent of its viewing angle and the direction the elderly person is facing, but it should also be possible to place it on a moving platform such as our household robot of the future.

Two other aspects are of high importance as well. First of all, the system should be capable of very fast response times: the less time it takes between event and alert the better. Post processing overnight is obviously not an option. Secondly, every home, situation and person is different which causes actions to be performed in many different ways. This fact, in combination with the notice that there can be many different actions performed by the elderly makes it virtually impossible to re-train the system with a fixed dataset. Therefore, action recognition has to be performed in an online setting which deals with both of the above aspects; it should allow recognition of actions very fast and it should offer possibilities for creating a learning algorithm that can easily be adapted to situations or trained for new actions.

3.2 Boundaries of this research

Of course, an action recognition system needs a way to acquire data as interface to the real world, an operating system and platform. The purpose of this thesis however is not to make a fully functional product that is ready for use in elderly care centers or in our own homes; the main interest lies in the algorithms and methodology behind such a system. Therefore possibilities to create interfaces with health care systems nor any other aspect that are related to creating an actual marketable product are discussed. Only the impact of choices on required computational power, which is related strongly to power consumption and costs is investigated.

One inevitable factor to be discussed is the method of data acquisition used, as this is strongly connected to the used algorithms and methodology. As suggested in the literature in Chapter 2, the state of the art method lies in the use of RGB-D data as it solves many issues related to variance in appearance and adds another layer of information by including depth data. From these RGB-D data skeleton representations can be extracted, which also solve large parts of the issues related to variance. For this research it was decided to use Microsoft's Kinect, as it is by far the best device to capture the RGB-D data and is able to extract the skeleton representations using its hardware. The Kinect is able to record at 30 frames per second, in a 640x480 resolution.

Although the skeleton estimation algorithms are quite accurate and can be estimated very fast (nowadays even from one frame [53]) there are still some drawbacks. For instance, estimating skeletons from a downward / upward angle is not working very well. Best results are achieved when the subject is observed from a frontal view. Once a skeleton is fitted successfully the skeleton can be tracked when the subject e.g. rotates or performs movements in any direction. Obviously, there are also difficulties when the person is performing actions away from the Kinect (self-occlusion) or when part of the person is behind another person or large object such as a closet, a table or if the person is e.g. below a blanket. These problems can be solved partially by adding more data acquisition systems and fuse their data (might solve occlusion) plus it is expected that the skeleton estimation algorithms will improve continuously. In this research subjects are limited to perform actions without being occluded and only viewed from limited variation in angles (+/- 45°). The algorithms behind the skeleton representation are not improved nor discussed in this work but details can be read in [53].

Another benefit of using the Kinect is that it can be connected easily to Windows machines. This means that Matlab can be used as programming environment to acquire data (Kinect plugin) and to design and test algorithms. Although programming in C or C++ on Linux is a much more efficient and fast way to process the captured data, the Matlab environment is more user friendly and especially useful to test the working principles of the algorithms as it is very flexible and has many built-in functionalities. The device used to test the algorithms on is a Samsung Ativ book 8 with an Intel core i7 3635QM and 8GB of memory.

3.3 Action recognition in this work

With Kinect as data acquisition method the skeleton representation is readily available. However, the skeleton representation is not used as the final representation in this work but as an intermediate representation. The reason for this is that the related work showed in a non-skeleton context that very good results can be achieved by using spatial temporal patterns to preserve spatial and temporal contextual information. Furthermore this representation can easily be understood without any complex prior knowledge and it is robust to many sources of variance. This leads to the assumption that by combining two representations, skeleton and a spatial-temporal pattern, the benefits of both methods can be used. The skeleton representation is used as intermediate representation and thus its spatial and temporal context information will be preserved by capturing the skeleton movements of an action in a spatial-temporal pattern (Figure 5). Furthermore, to add temporal information explicitly to the new representation, we propose using an older technique commonly seen in 2D action recognition called 'motion history'. This technique can be added by using the consecutive frames of the action; the most recent skeleton positions will be recorded in the spatial temporal pattern with the highest values, leading to a motion-history spatial temporal pattern or MH-STP. Also, this allows the estimation of the skeletons movements between two consecutive captured frames, leading to a more detailed representation of the movement itself. This representation is the main novelty of this thesis.

In order to evaluate the assumption that the proposed representation will lead to promising results, a public dataset of actions is used, as well as a set of works that use the same dataset, for comparison. This way, results can be compared with the state of the art. A good sample set of 3D actions was found in the Microsoft Research Action 3D dataset, which was recorded with a beta version of the first generation Kinect and is used in quite a few woks. As this dataset did not contain recordings of actions performed under rotations a second, newly recorded dataset was constructed and used. The new data set contains 12 actions performed by 9 different subjects under three different angles. The first generation of this device is used to record the dataset, as the second version of the Kinect, which very likely implements the new and greatly improved skeleton recognition algorithm by Shotton [52], was not available at the time the dataset was recorded. This second dataset should prevent tailoring of the proposed method to one specific dataset and prevent biased results.

With the dataset determined, the question remains how to perform the evaluation of the representation in such a way that it is comparable to the state of the art. To do so, the basic testing methodology as used in the STOP article [26] is mimicked. This work uses the Microsoft Action 3D dataset, some type of STP as representation and a cosine distance classifier in an offline setting and reported state of the art recognition rates. However this thesis proposes an entirely different way of generating a representation. By staying close to the STOP testing method (amount of samples, test types, etc), a comparison of the representation types' performance can be made.

The action recognition is then performed by splitting the dataset into a testing and training set, after which training samples from one single performed action are used to create a representation of the particular action and the test set is classified using the set of trained action representation. If the results are promising it is tested how certain parameters influence the recognition rate and required computational power of the algorithm. This way, a basis is made for testing the representation in an online setting which is far more complex. The offline setting can therefore be seen as a preliminary testing environment.

The online action recognition will be tested on the same dataset, only this time the actions are not considered as a whole but as a series of short segments of the action. This is needed since the online setting restricts the use of future data. Although ideally the segments of the action should be automatically segmented based on a measure of informativeness this is not done in this research, as accurate segmentation of movements within an action is a difficult problem on its own. The segments of the action should not be too long to as actions can be performed quickly but not too short either as this removes most of the time related information from the representation. The actions in the data set are thus split in separate fragments of N frames long, which means a short-sequenced MH-STP representation will become available to the system every N frames.

These representations are then fed to a neutral pose classifier, realized with a support vector machine. This also shows the suitability of the representation in an SVM. If non-neutral representations - think of them as intermediate poses within the action - are detected they are used by a classifier in a consecutive manner, for as long no new neutral representation is detected. This classifier consists of a set of action graphs, each connecting to one type of class. The observed action is then classified to the action class which belongs to the action graph most likely able to generate the observed sequence of representations.

The offline recognition results will be compared to the state of the art and if the performance is as expected, the novel representation will be used as basis for the online action recognition framework. One downside of the online action recognition methodology as presented, is that for some observed actions the classifier may be indecisive. The used representation however allows the recreation of a full length action representation MH-STP from a series of individual action segments, after which a second classification similar to the offline case can occur. This way, a sample that would likely be not- or misclassified in the online classifier due to similarities, has a higher chance to be classified correctly in a consecutive classification step. Note that the second step requires extra processing power in the online run time but is not needed for every sample that is being classified.

3.4 Schematic overview

This section shows the approach used in this report in a schematic overview. First the initial investigation, then the offline action recognition and finally the online action recognition.

Initial investigation to the MH-STP's potential - schematic approach	
Non-optimized parameters are used for both representation generation as well as feature extraction, in order to get a quick idea on the potential of the used method.	
1. Aquire data	Depth map (point cloud) and skeleton data acquired from MSR Action 3D
2. Generate representation	Novel MH-STP method using a skeleton intermediate representation and a 3D adapted version of motion history in a sparial-temporal grid.
3. Obtain Features	Holistic conversion of MH-STP to a high dimensional feature vector, dimensionality reduced using orthogonal class learning.
4. Prepare data using a training set	Use a subset of the data to generate a set of features with known classes that can be seen as 'trained data'.
5. Perform classification	Use the data not used for training as test data set with unknown classes. Use a cosine distance similarity measure as offline classification method
6. Evaluate results	Evaluate the results of the initial action recognition test with the MH-STP. If promising results are achieved, start exploring the MH-STP feature in order to optimize

Figure 25 Schematic approach of the preliminary tests.

Exploring the MH-STP in an offline setting	
The goal of this part of the research is to explore every aspect of the MH-STP. By using multiple datasets and multiple test setups, influences of parameters on results are discussed.	
1. Aquire data	Depth map (point cloud) and skeleton data acquired from both MSR Action 3D and novel dataset
2. Generate representation	Novel MH-STP method using a skeleton intermediate representation and a 3D adapted version of motion history in a sparial-temporal grid. The generation of the representation is parameter dependent and thus the parameter influences need to be explored
3. Obtain Features	Holistic conversion of MH-STP to a high dimensional feature vector, dimensionality reduced using orthogonal class learning. The influence of the dimensionality is explored as well.
4. Prepare data using a training sets	Multiple tests are performed for verification of the representation. Cross instance (33% training and 66% training samples) as well as a challenging cros subject test. Subsets of actions are used for verification.
5. Perform classification	Use the data not used for training as test data set with unknown classes. Use a cosine distance similarity measure as offline classification method
6. Evaluate results	Evaluate the results of the offline recognition. The parameters that lead to the best results should be used in the online recognition case.
Exploring the MH-STP in an online setting	
The goal of this part of the research is to evaluate the performance in an online setting. A short version of the MH-STP is used, together with an neutral pose SVM and Action Graph classifier	
1. Aquire data	Depth map (point cloud) and skeleton data acquired acquired from novel dataset
2. Generate representation	Generate multiple short MH-STP representatons by using only small sections of an action sequence instead of the whole sequence. Multiple of these representations are created for one action and they form a short MH-STP sequence.
3. Obtain Features	Holistic conversion of all short MH-STP sequences to a high dimensional feature vector, dimensionality reduced using orthogonal class learning.
4. Determine neutral representations	The short MH-STP representations are classified to either neutral or non-neutral poses. This marks the start and end point of an action: neutral poses are not informative for the actions.
5. Training the action graph	Using a set of short non-neutral MH-STP sequences with known class, obtained from a training set, an action graph model is trained per class.
6. Perform classification	Use the data not used for training as test data set with unknown classes. Generate and find short non neutral MH-STP sequences and determine which Action Graph model has the highest likelihood to produce the observed short MH-STP sequence.
7. Perform classification (2)	In case the classification fails in making a conclusion, superposition the short MH-STP sequences and perform a secondary classification as is done in the offline situation.
8. Evaluate results	Critically look back at all obtained results and evaluate the overall performance of the representation.

Figure 26 Schematic overview of the approach towards researching the offline and online setting.

Chapter 4

Exploring the novel representation and online framework

This Chapter explores the methods of the novel MH-STP representation and framework as discussed in Chapter 3 and discusses the novelties in greater detail. In Section 4.1 the novel MH-STP representation is discussed, followed by the offline and online implementation in Section 4.2 and 4.3 respectively. After these sections on offline and online action recognition, the principles behind the idea to combine the offline and online method are discussed in Section 4.4. Finally, a summary of the methodology is given in Section 4.5.

4.1 A motion history STP using an intermediate skeleton representation.

The novel representation discussed in this thesis combines three existing techniques and tries to take the benefit from the positive aspects of each technique. The first technique is the use of a spatial temporal pattern, the second one is the use of a 3D adapted version of 2D-MHI and the third technique is the use of an intermediate skeleton representation. The separate ideas behind this combination are discussed in the following two paragraphs.

4.1.1 Using a 3D adapted version of 2D-MHI in an STP

Using an STP as representation can lead to very good recognition rates, as the STOP article in [27] shows. As discussed in the literature overview, an STP can be described as a 4 dimensional grid, partitioned in $x \cdot y \cdot z \cdot t = m$ space time cells where $x \cdot y \cdot z$ denotes the amount of bins per space dimension and t the amount of time bins. A typical algorithm that generates an STP representation does this by counting the amount of 3D data points over time and places these within the bins the points belong to, for an entire action sequence. Place and time information is readily available, but if the amount of bins is low the captured information on the action decreases.

Unfortunately the STP does not tell much about the movement within the individual time segments. A basic STP only counts the occurrence of depth points in the individual space time cells which are related to the residence time and size of the visible body in the observed area. Most movement information is stored in the difference between time segments of the STP. From this perspective, the method shows resemblance to the work of Bobick and Davis [34] on motion energy images (MEI): each time segment in an STP is in fact an individual 3D motion energy recording. In a way, the earlier discussed STOP method can thus be seen as a 3D version of the 2D MEI method, with a saturation parameter of 1: every space time cell that contains at least 1 point is assigned the value 1. (Recall the saturation parameter from Section 2.8).

Inspired by this apparent connection to a technique that dates back from 1997, it seemed plausible that the other renowned representation of Bobick and Davis, the Motion History Image (MHI), would be even more useful if it could be used in the STP. With such a representation, each space time cell could become a function of the observed motion at that particular location. As motion is a very important aspect of actions it is expected that this will lead to a more informative representation and hence higher accuracy. Also note that although the MEI and MHI representation are not very robust to variations in rotation in case of 2D recognition, a 3D version should be much more robust to this source of variation as the third dimension is now available, in contrary to projected views of reality in 2D MEI/MHI. An example is given in Figure 27.

In the 2D representation, the magnitude of the MHI is an indicator of the moment in time that a certain motion is observed, while the 2D gradient of a location in the captured MHI is an indicator of the direction and speed of the movement. In the MH-STP case however, it is expected that just the MHI magnitude per space time cell should suffice, as the spatial temporal preserving nature of the STP representation allows for the inter-cell difference in magnitude of neighboring cell locations to be captured in the extracted feature. This has as benefit that a more computationally intensive 3D gradient calculation for each location is not needed.

Figure 27 Left: variations of a 2 dimensional MEI representation generated of the action 'sitting' under different rotation angles: it is effected a lot by rotation variance. Right: impression of an MHI in a 3D STP.

As described earlier, the total set of N recorded 3D points can be described with $A = \{(x_n, y_n, z_n, t_n), n = 1..N\}$, the set of space time cells is denoted with $C = \{c_1, \dots, c_m\}$, the frame number with f and the total amount of frames denoted with F. For the use of MH-STP the total set of points can be described more conveniently as a combination of points obtained per frame

$$A = \{A_f\}, f = 1..F \text{ with } A_f = \{(x_{f,n}, y_{f,n}, z_{f,n}, t_{f,n}), n = 1..N_f\}.$$

Do note that A_f depicts the set of 3D points at frame f and N_f is the amount of acquired points in that frame. Next, the set of points $D_{i,f} = A_f \cap c_i$ can be determined which is the set of 3D points originating from a single frame that intersect with one space time cell. The MH-STP value at a space time cell can be described by:

$$P(c_{i,f}) = \begin{cases} 1, if\ |D_i| \geq p \\ 0, otherwise \end{cases}, p \geq 1\ (noise\ parameter)$$

$$M(c_{i,f}) = \begin{cases} Mhidelay & , if\ P(c_{i,f}) > 0 \\ 0 & , if\ M(c_{i,f-1}) = 0 \\ M(c_{i,f-1}) - 1 & , if\ M(c_{i,f-1}) > 0 \end{cases} \quad with\ M(c_i, 0) = 0$$

4.1.2 The use of an intermediate skeleton representation

The STP combined with the MHI is expected to lead to a better performance of the action recognition system in comparison with methods commonly used. However, the modification does not solve the problem of variance in rotation; In fact, as the discussed methods to build up the STP use the acquired depth data from the 3D sensor in a direct manner both the STOP method and the MHI variant seem to be very sensitive to variance. For instance, the following sources of variance are present: difference in clothing of the subjects are directly reflected in the depth points, the depth image of a person from a 45° angle differs a lot from a frontal acquired one, the location of the person relative to the grid center can vary and there is a lot of size difference in postures of people. This variance in acquired depth points is partially covered for by the binning-nature of spatial temporal patterns, but a lot of the variance cannot be compensated for using these methods.

Fortunately, there is a very suitable representation that is able to deal with the before mentioned variances in depth data when the subject of interest is a human being: the skeleton representation. It is able to efficiently deal with appearance variances as it is designed to estimate the joint locations and angles, which makes it a very suitable representation to compensate for location, rotation and size. Furthermore the skeleton representation is a state of the art method that is used in many applications successfully and still is topic of interest for many researchers, leading to believe that further improvements to the method are to be expected.

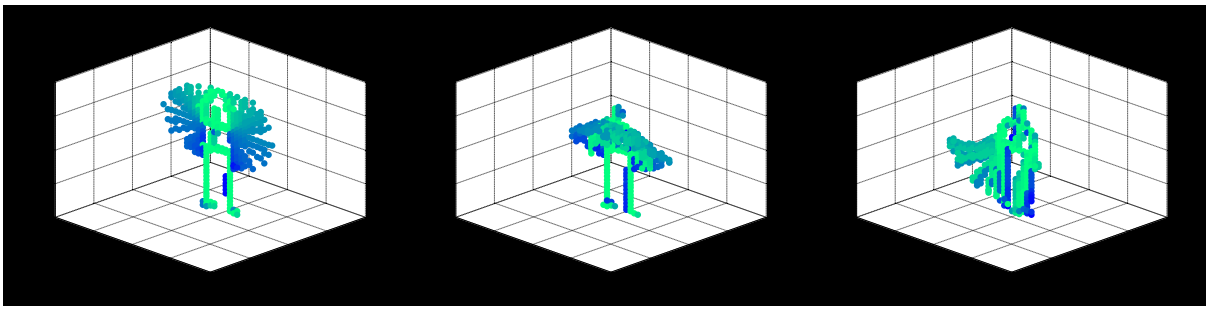

Figure 28 Three example actions (2 hand wave, horizontal wave, kick) depicted as skeleton enhanced MH-STP representations.

The idea of the novel MH-STP method is to use the skeleton representation as an intermediate layer between the raw input depth data and the algorithm that generates the STP, which is expected to increase the recognition accuracy. In this way, the MH-STP can be constructed in a rotation, location and appearance invariant manner which is done in such a way that it remains close to the original idea: create an intuitive, visually easy to understand representation. The MH-STP construction method is done as follows:

- Obtain a set of 3D data points
- Extract a skeleton representation
- For each 'limb' of the skeleton, construct a set of points by interpolating between the skeleton joint locations.
- Use both the joint locations J and interpolated limb points to construct the MH-STP in the same manner as in 0 but now the skeleton points S are used:

$$S_f = J_f \cup L_f \text{ where } J_f = \{(x_{f,n}, y_{f,n}, z_{f,n}, t_{f,n}), n = 1..N_f\}_{jointlocations}$$
$$L_f = \{(x_{f,n}, y_{f,n}, z_{f,n}, t_{f,n}), n = 1..N_f\}_{interp\_limbpoints}$$

With $D_{i,f} = S_f \cap c_i$

The use of the intermediate skeleton representation, plus the 2d motion history image method adaptation, is expected to lead to a very good accuracy in comparison to the state of the art.

4.2 Evaluating the representation: offline classification and test setup

To evaluate the new MH-STP representation, a cosine distance classifier is used to perform an offline classification. The cosine distance as dissimilarity measure is easy to understand and implement. Also, it turns out to be especially useful in case of high dimensional feature vectors. Especially if they are mapped to a lower dimensional space with Orthogonal Class Learning (OCL) as dimensionality reduction technique.

The reason why OCL is suitable for this is that cosine similarity is a measure of alignment between two feature vectors in space, in terms of the angle of orientation between them. If perfectly aligned, the returned value is 1 and in case two orientations are orthogonal the returned value is 0. Any other orientation difference leads to a value between 0 and 1, or between -1 and 0 in for vectors that are opposite to each other. The OCL dimensionality reduction method can provide suitably structured data for this classification method, as it maps high dimensional feature vectors into a lower dimensional space in which features corresponding to different classes are placed orthogonally to each other (see paragraph 2.9). This means that the similarity values as determined by the cosine distance classifier more likely return values around ~1 or ~0. This makes classifying easier as it is less likely to obtain arbitrary distance thresholds between 0 and 1 such as 0.4 and 0.6. The cosine similarity distance between two feature vectors f_c with classes $c = 1,2$ can be calculated with

$$d_{cosine} = \left(1 - \frac{f_1 f_2'}{\sqrt{(f_1 f_1') \cdot (f_2 f_2')}}\right)$$

This offline cosine distance classification works as follows: all distances from a tested feature vector to all trained sample points are calculated, and averaged per class. The class that has the most close by average distance result determines to which class the test sample is classified to.

To find out if the MH-STP representation is a suitable representation, the offline test as described in the STOP article (and others) is used for a good comparison. The STOP representation is then replaced with the new MH-STP representation. To get an idea on the effects of this new representation it is important to keep the amount of changes to the existing STOP testing method to a minimum. This means that the same dataset is used (MS Action 3D), the same classifier and the same test scheme. It is tried to remain as true to the original method as possible, but this turned out to be not always possible since some information on the STOP testing method was not available.

What stays the same in this evaluation?

- Basic source of input data (point clouds). As it is assumed that the Microsoft Research action dataset is acquired using a Kinect, it is likely that the Kinect's internal algorithms used the same point cloud data to create an intermediate skeleton representation as was used by the STOP paper. After enquiry, the authors explained to have used an early test version of the Kinect, which may have had an inferior data and skeleton acquisition method to the released version of the Kinect.
- The dimensionality reduction method.
- The offline classification is done by a cosine distance classifier.
- The testing scheme: which actions belong to the subsets, type of tests (33% training / 66% training / 50% cross subject training).

What is different/unknown in this evaluation?

- Quality of the MSR Action 3D dataset regarding the available skeleton representations is unclear.
- Exact used parameters in the STOP article are not known such as dimensionality.
- Exact used train and test scheme: although percentages of training / testing were mentioned, which training and testing sets were used and how many repetitions were used to generalize the results? (Or were maxima taken?).

4.3 Online classification method and test setup

Although the method for online classification in this thesis has an entirely different approach compared to its offline counterpart, it also shows a number of similarities to the offline classification method: the data acquisition and the way a representation is constructed are identical to the offline method. However, in this case the MH-STP representation is not created using an entire action sequence, but instead created using only a few frames of the action sequence. The representation could therefore be seen as a 'short' version of the MH-STP and an action is thus captured in a collection of consecutive MH-STP representations as shown in Figure 30.

And that is where the similarities between the offline and online recognition method end.

One big difference between the two situations is that in offline classification the actions are segmented nicely and are readily available. In the online case however, actions are observed in a live setting and there are no annotations available. This makes it more difficult to classify an action, because it is not known on forehand when an action has started or when it has ended.

The online action recognition process is as follows

1. An action is observed and every 5 frames an MH-STP representation is formed as a segment of the action. Every action therefore exists of multiple short-sequenced segments. Since a segment is built from only very few frames, it is possible to say that an action segment is similar to a 'pose'.
2. Every resulting segment is classified by a neutral pose classifier. A neutral pose is e.g. sitting or standing. The non-neutral segments between the neutral segments, such as an upward directed arm in a wave action, are assigned to a set of salient poses.
3. This set of salient poses is classified using an action graph as classification method, leading to either a correct classification, a misclassification or an indecisive classification. An indecisive classification is of course undesirable and therefore incorrect. This can happen when e.g. a sample is equally probable to be classified to one class as to another, or if an observed action sequence does not fit any of the action models. The cause of this is that the transition models of the individual action graphs might not be trained for a certain node transition.

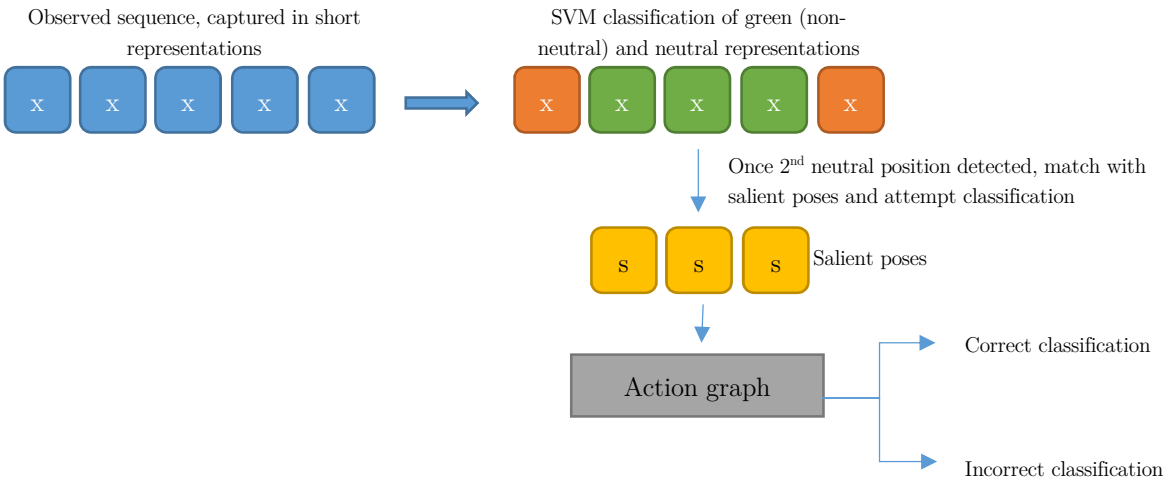

Figure 29 Schematic overview of online action recognition method.

4.3.1 Using SVM as neutral pose classifier

In this work, one problem to overcome is knowing when an action starts and stops by making the assumption that an action starts from and ends at a neutral pose. A neutral pose can be e.g. 'standing' or 'sitting'. An example is the neutral outer left representation in Figure 30, just before the double wave action starts. When the double wave action ends at the end pose, not shown in the figure, it is likely to be very similar to the initial neutral representation.

To detect these neutral poses an SVM classifier is used with a kernel using a standard radial basis function. This classifier is trained using a large set of neutral and non-neutral samples. Ideally, this training dataset is labeled automatically. From the total data set, the available segments of action 1 (stand still) and action 3 (sit still) can be considered neutral poses. Also, the first and last N segments of each action sequence are considered to be neutral. These segments can therefore be automatically labeled as neutral and non-neutral by using an automatic labeling process that verifies which part of the action the segment belongs to. Of course this automatic labeling process is not very refined: there is a big chance that segments might be classified incorrectly as neutral or non-neutral. This might negatively influence the neutral pose classifier. There are a couple of reasons why this could go wrong:

- The recorded full actions are pre-segmented. This introduces noise. The action performed by the actor could for instance already be over 3 segments before the end of the total recorded sequence. If only the last segment is marked as neutral, the two segments before that can be incorrectly marked as non-neutral segments.
- The action performed by the actor could have started much later than the recording of the action sequence started. If only the first segment is marked as neutral, the few segments after that, where the person was still standing still, could be incorrectly marked as non-neutral segments.
- The other way around could also happen: the recording is stopped just before the action is stopped. This will cause the last short segment to be incorrectly labeled as neutral.

Although it is very interesting to see how well the neutral pose classifier will perform with automatically-labeled data, a manually labeled data set is also constructed for comparison and to minimize the adverse effects of an SVM classification that would be based on inaccurate training data. Part of these neutral samples are obtained from the neutral actions standing and sitting still and part of the neutral samples are obtained by manually labeling 10.000 short action segments in the form of MH-STP representations This was a lot of work. The use of the neutral actions to obtain neutral samples is done since it allows very easy extraction of a large volume of neutral representations. The manual labeling is done since short neutral MH-STP representations captured just before and after the action itself are likely to be different from the neutral MH-STP representations. The reason for this is that segments captured at the beginning and end of a non-neutral action might contain a little bit of data from the very end or beginning of the recorded actions and therefore a little variation in the MH-STP due to movement at the start and end of the actions may be accounted for.

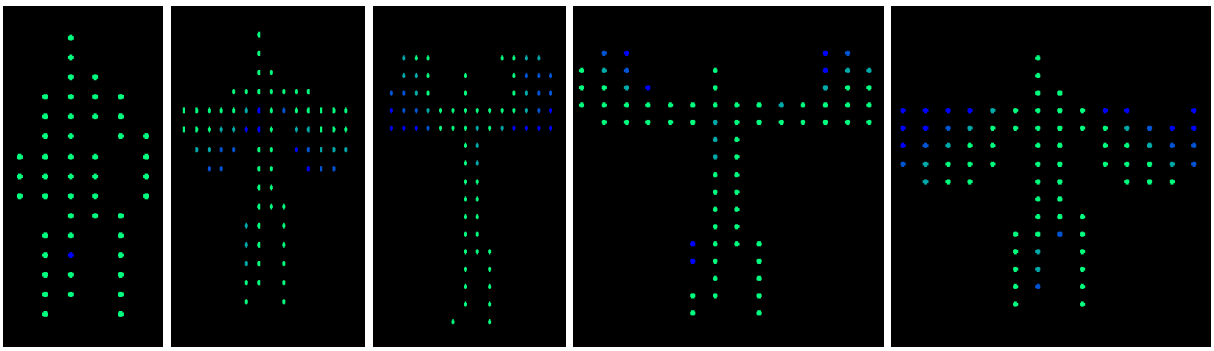

Figure 30 A two hand wave action, captured in a series of 'short' MH-STP representations.

In order to successfully classify the neutral poses from the non-neutral poses, several parameters need to be thought of. Some of these parameters however do not only have an effect on the performance of the SVM classifier but also have a direct effect on the classification of the action itself, in later stages of the action recognition framework. The following parameters need to be tuned:

- How many frames should be used to build the short MH-STP representations?
- What grid resolution should be used? A higher resolution increases detail but also computation time.
- To which dimension should the original representation be reduced to using OCL?

For these parameters, it is chosen to initially use the method and dimensionality that obtained the best classification results in the offline recognition case. The amount of frames per short MH-STP is set to 5, as an average action sequence is around 30 frames of length; a higher amount of frames would lead to a very low amount of short MH-STP representations while a lower amount would remove most of the useful temporal information. Other parameters of importance for the SVM are:

- Box constraint
- Sigma

There is no good way to know on forehand to what these parameters should be set in an optimal way. Therefore, an optimization is performed using a grid search over a set of possibilities to find out which parameters would lead to optimal results.

4.3.2 Using an action graph as action classifier

In the offline, preliminary test case, the cosine distance classifier was implemented. In the online case the action graph is implanted as classifier. The main benefit of this classifier is that it requires fewer training samples to operate, allows easy addition of another action class without retraining earlier trained samples, it is robust to intra-action variances and is, if needed, even able to recognize actions before the action itself has ended. This allows for a very small latency, which is ideal in an online situation. After all, we would like to be able to respond to an emergency call of an elderly in distress as soon as possible.

The working principle of an action graph classifier is intuitively easy to understand in a graphical way. The following steps to train the action graph models are followed.

1. A set of action samples used for training, is fed through the algorithm which generates the 'short' MH-STP representations. The total set of representations is used to determine a mapping towards a lower dimensionality. Then, all the neutral samples are removed, leaving only the non-neutral set of features, which could be thought of as individual 'poses' during an action. These can be seen in Figure 31, upper left image.
2. In Figure 31 mapped features originating from two different classes are depicted with blue squares and black stars, but in the clustering phase that follows the labels are not used. Using an unsupervised clustering method such as a k-nearest neighbor (KNN), followed by an expectation maximization (EM) algorithm, a pre-specified amount of key poses (or salient poses) are to found in the cloud of short MH-STP representations. These salient poses are shown with red diamonds in Figure 31. Note again that the clustering algorithm does not use any class information, as it is unsupervised clustering! Also note that the salient poses are not assigned to any class. On top of that it is very hard to predict what amount of clusters or salient poses is needed, as this highly depends on the observed data.
3. Using these salient poses, action graph models can now share the same global information, although the models also capture information in an action graph per action class. The individual actions are encoded as the probability of going from one salient pose to another salient pose, for every salient pose in the feature space. A visual example: sitting down or standing up might share some salient poses although the order of the consecutive salient poses differs.

In the lower part of Figure 31, two different actions are encoded: one in the lower left and one in the lower right. Both actions happen to be cyclic, and in one direction only. It can be seen that there is no 'path' to B in case of the 'star' class. Now, if a certain sequence of short MH-STP is observed from a certain recorded action (far right of Figure 31), what class should it be classified to? The obvious, simple answer is the 'square' class as the other class has no transitional probability for this path.

In most cases however, the answer is not that easy to find. For instance, the data points with the short MH-STP representations could be much less easy to separate. The question 'how likely it is that a certain data point belongs to a certain salient pose' becomes much more important in that case. What if the odds are 50-50 that a point belongs to A or B? Is it then still easy to say that an observed sequence ABDC must belong to the 'square' class in the lower left? Or should we also take into the count the probability that the MH-STP point now classified as a B salient pose in fact should belong to A (or C) and that the model for the star class is therefore also a potential candidate?

To cope with these calculations, the action graph uses a Viterbi algorithm to determine the most likely path through all the salient poses in the feature space. It does so by calculating the probability that a certain model (the action graph) could have created an observed sequence of an action by following the optimal path through the different action graphs. Then, by comparing these probabilities, while keeping an eye on prior probabilities, a classification can be made possible.

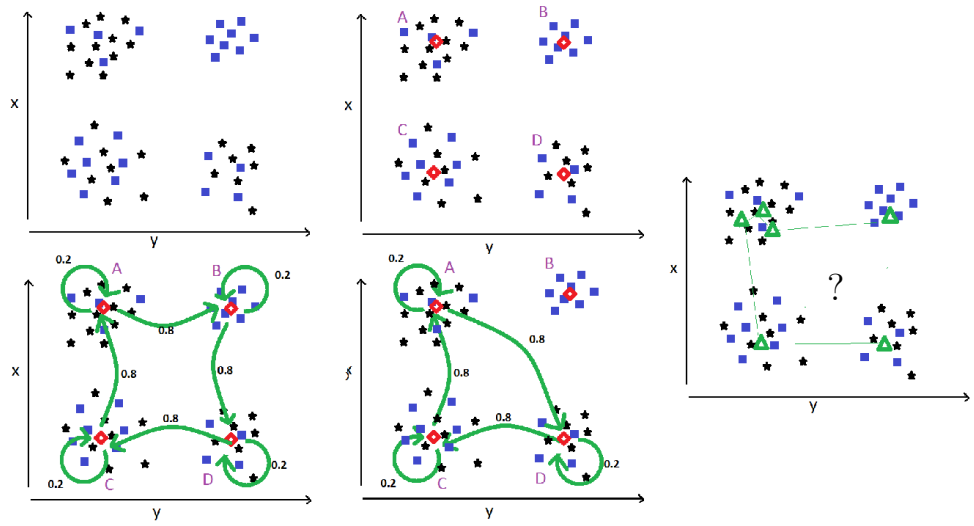

Figure 31 Graphical representation of an action graph. Each blue square and black star represents a mapped short MH-STP feature belonging to class 1 and class 2 respectively. Salient poses of the action graph are shown in red. Class 1 transitions in a square fashion, class 2 in a triangular fashion. If a new action sequence is observed (green triangle dots at far right) the observed path through its features is indicating that the observed sequence belongs to class A. Calculations using transition matrices of both classes will confirm this.

Looking back now at the graphical, visual representation as sketched in Figure 31, it should be noted that this specific situation is hardly ever applicable as the feature space is usually in a much higher dimensionality. Depending on the dimensionality the MH-STP are reduced to this can easily lead to a dimensionality of 100 or more. Also, the individual features are by far not as neatly separated as in the example, let alone that it is known how many clusters there should be. Imagine you would look at the samples of 10 different actions performed by 10 different subjects, all of them cut in 5-frame pieces.

This will result a large variety in features that have very unpredictable values and variances, especially after dimensionality reduction. Not only because movements between different actions differ, but also because the actions might be performed in different ways, or at a different speed by the subjects. Too few clusters / salient poses and information might be missed, too many clusters and overfitting starts to become an issue. At this moment in time there is no reliable way to calculate the optimal amount of clusters; therefore most researchers, including the original author of the action graph method [64] as well as Vieira in [27], will run the algorithm for a number of different options and take the amount that leads to optimal results. Note that this is not an ideal method, as it might easily lead to biased classifiers.

4.3.3 Different ways of using the action graph

Given a trained action graph classifier, the general way of decoding a new observation and classify it to the most likely action is done in three consecutive steps:

1. Find the most likely path through a set of trained salient poses.
2. Compute the likelihood of each encoded action model to produce the determined path.
3. Classify the action as the encoded action model having the maximum likelihood, if the maximum likelihood is above a certain set threshold. Otherwise, classify the observation as an unknown action.

In a formula, this task is expressed as:

$$\psi^* = arg \max_{\psi \in \Psi, S \in \Omega} p(\psi) p(s_1, s_2, \dots, s_n | \psi) \prod_{t=1}^{n} p(x_t | s_t) \quad [64]$$

Where

$\Psi, \psi \; and \; \psi^*$	=	The total set of classes, a specific class and a classified class respectively.
x_t	=	Observed feature at a certain step in the action sequence.
s_t	=	Salient pose at a certain step in the action sequence.
$p(\psi)$	=	Prior probability of a certain class (a).
$p(s_1, s_2, \dots, s_n \| \psi)$	=	Likelihood of a sequence of salient poses given a certain action class (b).
$\prod_{t=1}^{n} p(x_t \| s_t)$	=	Total probability that observations x_t belong to be salient poses s_t (c).

Formulas (a) and (b) are called 'specific action knowledge' and formula (c) 'shared knowledge' as the probabilities of a certain observation to belong to a certain salient pose are shared by all the action models. Looking more closely at formula (b), the sequence of salient poses is obtained by the first of the three consecutive steps as mention in the beginning of this paragraph. There are three main ways of doing so:

1. **Action Specific Viterbi decoding**
   Search for an optimal path per encoded action specifically: this finds the optimal path through the salient poses of every individual action so should lead to superior results. However, it comes at the cost of a much higher computational and memory demand once the amount of actions gets high.
2. **Global Viterbi decoding**
   Search for an optimal path in the total set of salient poses. A good tradeoff of accuracy and cost, as the expensive calculation needs to be performed only once instead of once per action, while at the same time taking the transitional probabilities of individual classes into account.
3. **Maximum likelihood decoding**
   Don't search for an optimal path, only for the most likely salient pose per observation of the sequence individually. The likelihood of the total path itself does not matter. Very memory friendly and fast.

The method used in this thesis is the global Viterbi decoding method, as it seems to be a good tradeoff between the required processing power and memory versus the accuracy of the system. On top of that, when looking at the field of application for this system, humans (elderly) can perform a very broad range of actions. It would at this point very likely require too much processing power to use action specific Viterbi decoding. Also, as the maximum likelihood decoding cannot benefit from the sequential information of the actions it is less likely to lead to good results when the amount of actions is higher. It is also not able to reliably differentiate actions that are similar in nature, but reversed such as pick up / put down and sit down / stand up.

4.3.4 Classifying actions as unknown?

Briefly mentioned in the third step of the action graph method in the previous paragraph, is the notion that there is the existence of classifying an observation as an 'unknown' action. This could happen in case not only the returned action graph maximum likelihood that an observed sequence belongs to a certain action is of importance, but also that this likelihood should be above a certain threshold. What is troublesome in this case is that the issue of setting the right threshold is ignored in any of the articles discussing the action graph as classifier. What if all action models return a very low probability? Or one action graph model returns a value very close to another action graph model? The confidence that a certain sequence which is classified to one action really belongs to that particular action is not discussed at all. The most reasonable answer why this is not covered yet is the fact that there is no way to know (yet) which threshold should be set, in order to determine which outcome is sufficiently confident and which is not. Above all, setting a threshold to such a complex classifier seems an arbitrary thing to do since it is impossible to know when an observed sequence of features is a new, untrained action, or the same trained action performed just a bit differently. The risk of classifying actions as 'unknown' that would be otherwise classified perfectly fine is high when the threshold is set too high, as well as the risk of misclassifying actions is high when the threshold is set too low.

On top of that: the action graph model outcomes are not in the range '0% - 100% probable'. Instead, looking at the formula in the previous paragraph, the produced probabilities greatly depend on the sequence length, since longer sequences most likely lead to much lower probabilities produced by all different encoded action models. In the opinion of the author of this thesis, it would be a much better solution to check the relative probabilities of the different classes. If the maximum probability is e.g. three times the probability of the 2nd highest probability, the evidence is stronger than the case where the relative difference in probability is only very small. This latter situation is an additional problem in this classification task.

One other assumption why this topic has not been dealt with is that the threshold might be greatly dependent on the application's environment. In a typical research set with X different actions the minimum threshold could be left out of the equation, as it is not possible to encounter actions not trained for initially, unless one deliberately trains the models using only a subset of X and tests with the full set. In case there is a large chance of encountering actions not trained for initially the threshold might be useful, but still the question remains what value to put it on. In the research of this thesis, the issue of setting a threshold is ignored since we have a limited dataset. The observed sequence is therefore always classified to the action model with the highest probability of generating that particular observation sequence. Also, a relative comparison with other results is left out of the scope to not further complicate matters and focus on the performance of the MH-STP in an online setting.

This does not mean that an action is now impossible to be classified as 'unknown'. This is because there is one exception: in case the maximum probability is not uniquely given by one specific class but instead by multiple, the action is classified as 'unknown'. If multiple models give the same probability there is no way of knowing which action to classify the sequence to.

This can occur if a certain step in the observed MH-STP sequence is assigned to the transition from one salient pose to another salient pose by the global Viterbi algorithm, but the salient pose transition itself is not trained in any of the action graph transition models. Therefore, $p(s_1, s_2, \dots, s_n|\psi) = 0 \; for \; all \; \psi$ and all encoded action models will produce 0 as probability. This is especially likely to happen in case a too low amount of training samples is used, since less of the variance can be accounted for in the training phase of the action graph classifier. One way to solve this would be the use of action specific Viterbi decoding but as for this thesis it was decided that global Viterbi is used to save additional needed processing power.

One option to solve this issue is to replace all the values in all action specific transition models that are 0 after the initial training phase with a very low transitional probability and then perform a secondary EM pass to update the action specific transition and prior matrices while keeping the salient poses unchanged and equal to the global set. This secondary EM pass will however have implications on the required processing power especially if the amount of actions increases. This method is therefore not preferred but perhaps inevitable to be used.

A secondary option is to exploit a property of the MH-STP not discussed so far. The short MH-STP features can be super positioned to reconstruct the MH-STP representation for the total action sequence. If good results are shown in the offline scenario it might be worth investigating if the offline classifier can be used as a secondary classification stage in case an observed action sequence cannot be classified using the action graph model. This is the second novelty described this thesis.

4.3.5 Evaluation method

Similar to the offline case, evaluation is performed using three different tests:

1. 33% training samples, cross instance test.
2. 66% training samples, cross instance test.
3. 50% training samples, cross subject test.

The accuracy is tested for different amount of salient poses and for different dimensionalities to reduce the short MH-STP's to, to study the effects of these parameters. All of the tests are averaged over five runs. Results belonging to the same test, but with different amount of poses and dimensions are determined with the exact same training/test set so a good comparison is possible.

In case an action is classified as 'unknown' it is registered as a misclassification, since the system should be able to classify all actions correctly: missing a performed action by an elderly person could lead to dangerous situations that are not alerted for.

4.4 Combining offline and online methods by reconstructing MH-STP's

In Section 4.3 it was explained that the action graph classifier may classify observed action sequences as 'unknown', either due to a certain set threshold (if applicable) or due to the fact that a transition is observed that has not been encountered in the training phase in case no secondary EM pass is performed. This is undesired behavior, since 'unknown', in case of our elderly monitoring system, is just as bad as 'false negative' (failing to detect dangerous action). On the other hand, 'unknown' is better than 'false positive' (false detection of dangerous action) but as this has much less impact than in the 'false negative' case, the 'unknown' action should be avoided as much as possible.

Assuming the thresholds are not used nor needed in this research, which solves the first cause, the second cause for 'unknown' classifications can (partially) be solved for by using a secondary per-class EM pass to re-estimate the prior and transitional probabilities in order to reduce the amount of transitions with probability zero. This is done by keeping the initial, globally estimated variables of the GMM distribution that describes the salient poses fixed: only the transition matrices and prior probability matrices are updated from their initial, globally estimated state. Note however that before this EM pass, all the zeros of the transition and prior probability matrix have to be replaced by a low probability and then re-normalized. This is needed since the EM method cannot change the zeros into nonzero values.

However, although the accuracy will most likely improve, this will also increase the required processing power in the training phase significantly. Furthermore, it does not offer a solution in case two action class models produce very similar probabilities: the highest probability is still taken as the classification result even if the difference with the second best result is marginal.

To prevent the excessive, additional processing costs of another EM pass per action, while at the same time being able to deal with the 'unknown' actions and the issue of very similar online classification probabilities, a very useful property of the short-MH-STP can be used. This idea is built upon the very simple thought to give 'unknown' observation sequences a second chance, and that this, should lead to better results even if the second stage classifier performs only moderately.

As an example, consider the following 2 situations. Both situations use a first-stage action graph classifier that is resulting in a 25% 'unknown' rate. However, of the not-unknowns, a very good 93.33% is classified correctly, which leads to 70% of the total observations classified correctly and 5% of the total observations incorrectly. In case no second classification stage was implemented, the misclassified plus unknown observations would both count towards the error rate, in total 25% + 5% = 30%. In case second stage classifier would be used as a second stage classifier for the unknown samples and it had a 75% recognition rate on average, the error rate would drop to 11.25%. Even when the second stage would be only correct in 50% of the occasions the error rate would drop to 17.5% from 30%.

1. 70% Classified OK, 5% Classified NOK, 25% Unknown
    - Total score: 70% good and 30% bad results
2. 70% Classified OK, 5% Classified NOK, 25% Unknown > Unknown 2nd Phase : 75% OK, 25%NOK
    - Total score: 88.75% good and 11,25% bad results

To achieve this, the second stage classifier should be chosen. In this work the earlier described offline classification method is used for this secondary classification task, since it obtained good results and it relies on the same MH-STP representation. In this case however, the MH-STP of the full sequence is needed, which can be reconstructed easily by super positioning the pre-OCL 'short' MH-STP representations of the sequence that were used in the in the first classification step. The system is evaluated using the same test scheme as described in paragraph 4.3.5.

The main benefits of the system are as follows:

- The MH-STP allows a relatively easy implementation of the secondary stage.
- The additional processing costs are minimal in the training phase and much less then needed for the secondary EM pass solution.
- Observed sequences that would be classified to an action with only very small confidence are better handled by this system. As setting an absolute threshold is not an easy task a relative threshold. The relative threshold was set to 100; if one action graph produces a probability within a factor 100 of another action graph a reclassification is justified by the second stage classifier. This amount seems rather high but as the returned probabilities are logarithmic in scale it is clearly not the case.
- Unknown actions are now accounted for, the recognition rate depends on the performance of the 2nd stage classifier.

The main drawbacks of the system are as follows:

- Additional processing costs are needed in the testing phase, although only when an action cannot be classified in the first stage.
- The grid size in the online setting limits the grid size that can be used for the reconstructed MH-STP in the offline setting.
- Results might not be as good as the secondary-pass EM action graph system.
- A relative threshold has to be chosen to decide which observed sequences should be classified in the second stage and which can better be classified in a secondary stage. This is needed in case two action graph models produce similar results. Choosing the percentage is not supported scientifically and might need additional research.

The process is as follows:

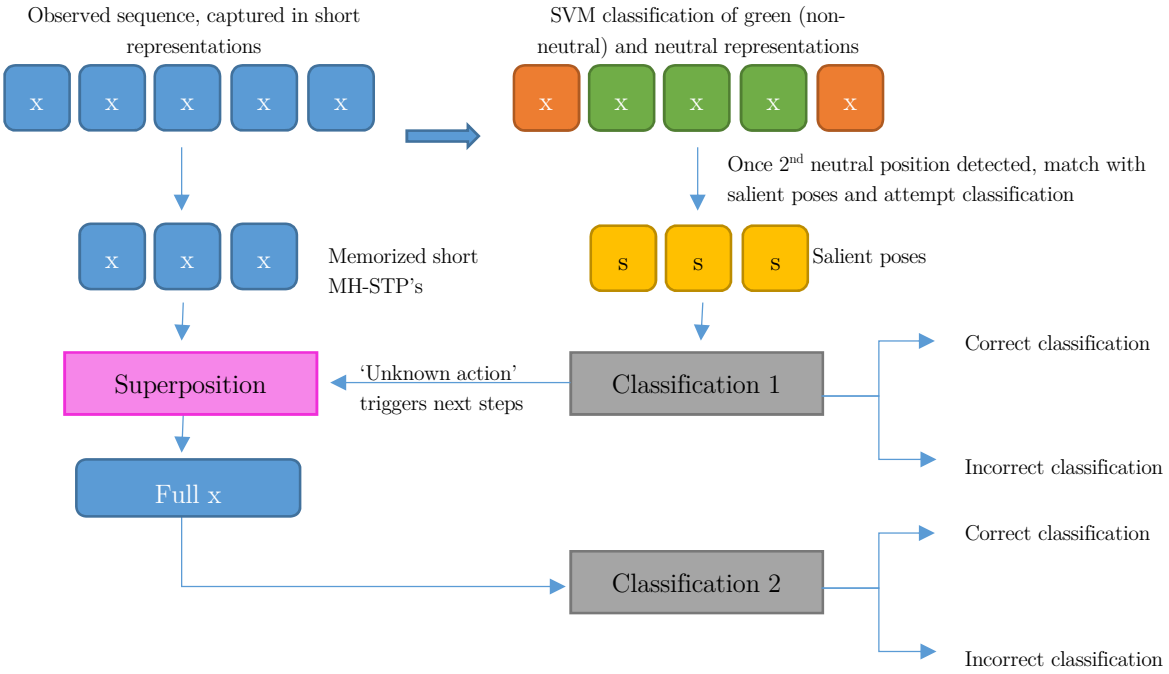

Figure 32 Schematic overview of a 2 stage online action recognition method.

The super positioning process works as shown in the schematic below:

1. Start with the first non-normalized short MH-STP representation.
2. Subtract the amount of frames used to create the short MH-STP representations.
3. Next, overwrite all MH-STP cells where the second short MH-STP is not zero.
4. Repeat 2 and 3 for all remaining short MH-STP representations.

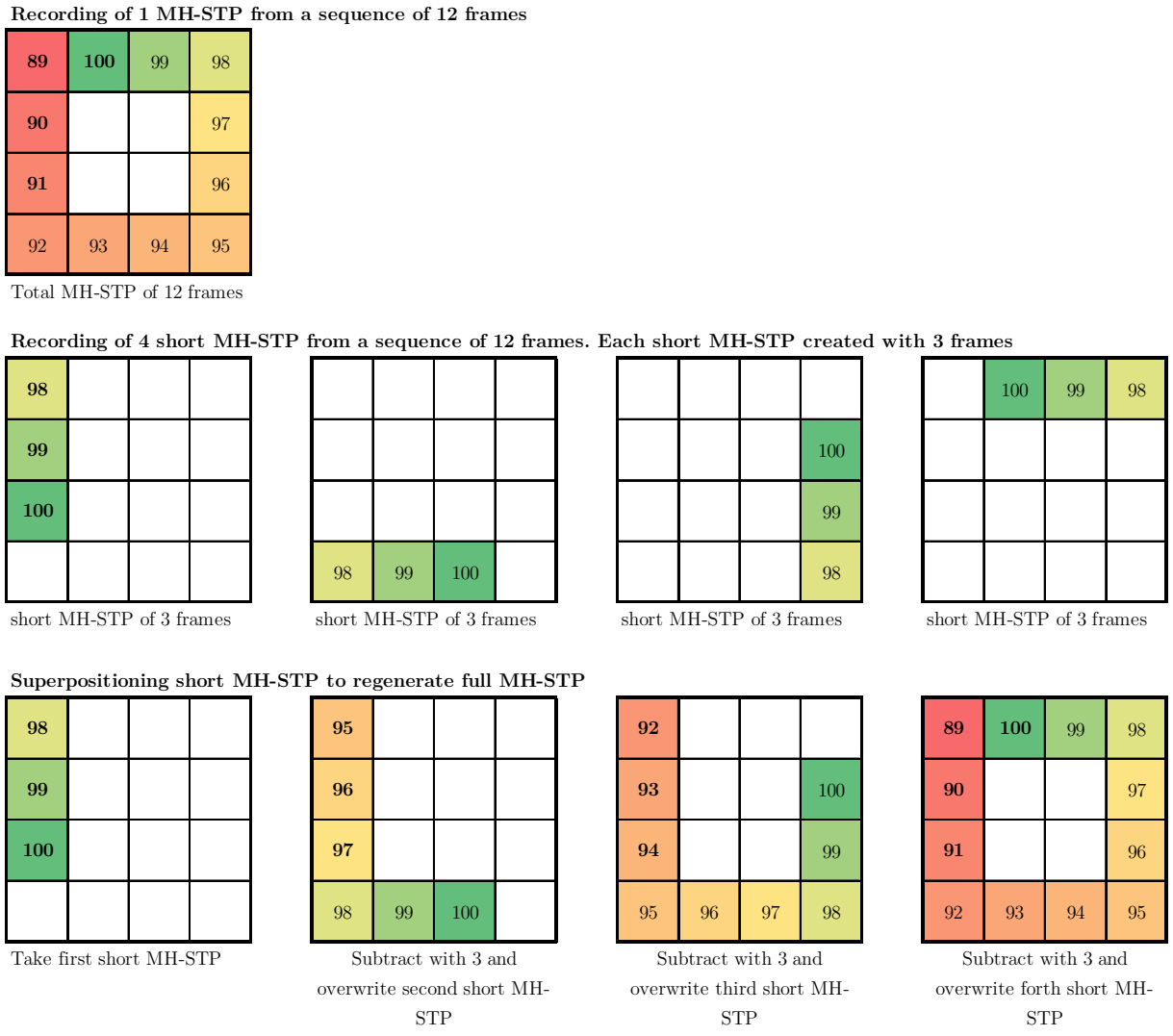

Figure 33 Example of super positioning short MH-STP features to recreate an MH-STP representation of the full action.

4.5 Summary of the MH-STP exploration and evaluation methods

The novel MH-STP is a representation that takes the best three aspects of three existing techniques: a spatial-temporal pattern generated using interpolated motion-history data of an intermediate skeleton representation. The MH-STP is generated by observing the joint locations of the tracked intermediate skeleton representation over time for a certain action. Every frame the space time cells within the MH-STP that are 'touched' by the limbs of the skeleton are updated with new values.

The benefits of the three separate methods are:

- The spatial-temporal pattern preserves spatial-temporal information of the sequence.
- 3D MHI gives additional temporal information to the representation.
- Skeleton representation reduces many sources of variance: appearance, location, rotation.

The drawbacks of the separate methods are:

- STP might be negatively influenced by intra-action variance due to its binning nature.
- 3D MHI might induce intra-action variance.
- The obtained skeleton data quality depends on external skeleton tracking algorithm. At this moment in time this tracker is still limited to certain situations which may not make the method suitable for all environments. As these algorithms continuously improve, the recognition rates of the novel solution are expected to improve as well.

To determine if the novel MH-STP representation can lead to promising results, first a small scale evaluation is performed in a manner similar to the state of the art STOP method. This is done by performing a recognition test as is used in the STOP method, with similar initial parameters and an identical dataset. The main difference however is the representation: the novel MH-STP representation is used instead of the STOP representation. Offline classification is done with a cosine distance classifier, which is a similarity measure between data points that is well suited in case PCA or OCL are used as dimensionality reduction technique.

The online recognition case is more complex. The same method to build MH-STP representations is used, but this time during an action every N frames a ‘short’ MH-STP representation is generated. Note that this short MH-STP representation does no longer contain more than 1 time segment as was the case in the offline situation: time information is encoded in the representation by the MH-STP only. Dividing the N frames into multiple time segments is therefore omitted. One action consists of many frames, one single action therefore leads to a sequence of generated short MH-STP representations. Each of these representations is then classified by a neutral pose classifier after which the sequence of non-neutral representations can be classified to actions using an action graph classifier.

This type of classifier is beneficial as it requires low amounts of samples, is robust to variance and easily expandable for new actions. During training an action graph model is constructed per action, which is in fact a set of salient poses together with a matrix that gives the probabilities that one pose is followed by another in a certain action. Using these models, classification occurs to the one that most likely can create a certain observed sequence of short MH-STP representations. This is determined with help of a method called global Viterbi decoding, which searches for an optimal path in the total set of salient poses.

Unfortunately some observations cannot be classified well. Probabilities of different classes might be very close to each other. It may even be that all of the action graph models return zero as probability since the action specific models might not be trained for a certain transition between salient poses that is observed.

To cope with the unknown actions, there are two possible approaches: a 'standard' secondary action specific EM pass that only updates action specific prior and transitional probabilities can be used *or* a novel two stage classification framework can be used. The second classification stage of this framework exploits the fact that the short MH-STP can be super positioned allowing classification as in the offline situation to take place.

As in the offline case, a preliminary test is used to estimate optimal parameters such as the amount of salient poses, the dimensionality to reduce to and MH-STP grid size. With these parameters the proposed methods are then validated in three different tests in three different modes. It should be noted that an 'unknown' action is considered to be a recognition error.

Action recognition modes

1. Global Viterbi using EM clusters, with global prior and transitional probability matrices.
2. Global Viterbi using EM clusters, with action specific EM passed prior and transitional probability matrices.
3. Global Viterbi using EM clusters, with global prior and transitional probability matrices, and a second stage classifier to cope with the unknown classifiers.

Validations

1. 33% training samples, cross instance test.
2. 66% training samples, cross instance test.
3. 50% training samples, cross subject test.

Chapter 5

Experimental results in an offline setting

5.1 Datasets used in this research

This section is discussing the datasets used to test the proposed methods and it also discusses the testing scheme that is used for the evaluation of the algorithms. Initially, a dataset of Microsoft is used: the MSR action 3D dataset. This dataset was used in the STOP article as well as in a number of other researches on action recognition. Tests performed on this dataset have as main goal to determine if the new representation has the potential it is thought to have.

The second part of the evaluation involves testing the novel representation on a second dataset, as testing the method on only one dataset does not really tell much about the performance of the developed method: it only tells what the performance of the developed method is on that specific dataset. This second test therefore functions as a manner to confirm that positive recognition rates are not biased on one test set. The second dataset also needs to have action samples from multiple angles for testing the performance under a rotation variance, since the MSR Action 3D dataset does not offer this data. To achieve this, a novel dataset has been created, as there were no pre segmented public datasets available for this purpose that contained both accurate skeleton and depth data from multiple orientations, which is needed to compare the different methods. Apart from the initial tests to check the viability of the representation, all results are obtained from tests using the novel data set. The datasets are discussed in 5.1.1 and 5.1.2.

5.1.1 The MSR action 3D dataset

The MSR action-3D dataset is a dataset consisting of depth and skeleton sequences that is captured by a depth camera. The dataset consists of twenty different actions, performed by 10 different subjects. Each subject performs each action two or three times, leading to a total amount of 567 depth map sequences. Note that in the MSR action 3D dataset the subjects were all facing the camera. Unfortunately, after visual inspection it turned out that the available skeleton data is of rather poor quality compared to the current standards. The samples contain many skeleton representations that are affected greatly by noise, some even to the point that no skeleton was visually distinguishable anymore. Also, for some data sequences skeleton data was not available at all. This is likely caused by the skeleton tracker, as it apparently failed to track the skeleton in the observed scene accurately enough. Also, some of the actions might have been less suitable for front-facing skeleton tracking. Examples of skeletons in increasing degree of incorrect tracking can be seen in Figure 34.

For the method in this thesis the skeleton data is of vital importance. Therefore some samples from the dataset were not usable as the data was too incorrect. This led to a total of ‘only’ 406 skeleton sequences that were considered to be of a quality that could meet the expected standards of commonly available skeleton tracking methods. Note that noisy records were preserved as much as possible, but as soon as recorded skeletons were getting really inaccurate (by subjective visual inspection) the data sequence was discarded.

Note that one should normally be very careful when choosing to discard samples from the dataset or not. In this part of the research however it was chosen to minimize the influence of the skeleton tracker performance on the total recognition result. Therefore, as the main interest of this thesis lies in increasing the performance contribution of the used representation, it was decided that discarding the samples was a good measure. The choice of discarding samples does however mean that less training data is available, which can potentially result in lower recognition rates.

Most researchers that use the MSR action 3D dataset to validate and compare their methods, have split the set of 20 actions in three different subsets, each having eight different actions (Table 1). Subset 1 and 2 are collections of actions that have similar movements and the third subset is a collection of complex actions. To validate the novel representation from this work and compare it to the other works, the same testing logic is used. However, due to the discarding of skeleton data, the action 'pickup & throw' and action 'bend' are no longer available. The method used to estimate the skeletons joints using the observed scene seemed to be completely unable to track the skeleton as soon as a bending motion was observed: often no skeleton data was available at all or the data looked like the right image on Figure 34. This was the case for all subjects and nearly all samples within these actions: only the pickup & throw action had some useful samples remaining but there were too few samples left to be used for training and testing.

Subset 1	Subset 2	Subset 3
Horizontal arm wave	High arm wave	High throw
Hammer	Hand catch	Forward kick
Forward punch	Draw X	Side kick
High throw	Draw tick	Jogging
Hand clap	Draw circle	Tennis swing
Bend *	Two hand wave	Tennis serve
Tennis serve	Forward kick	Golf swing
Pickup & throw *	Side boxing	Pickup & Throw *

Table 1 Action collection in different subsets for the MSR action 3D dataset.

This decreases the total amount of actions in Subset 1 to 6, and subset 3 to 7. This will likely cause the preliminary results of these subsets obtained with the algorithm from this work to increase, as less classes are available to classify to. However, it seemed worse to use really bad data (such as the mixed pile of joints in the right image on Figure 34), as classifying this data correctly does not really tell you anything; it just tells that the classifier is able to recognize extremely noisy piles of joints and not if it correctly classified the action 'bending' or 'pick up and throw'.

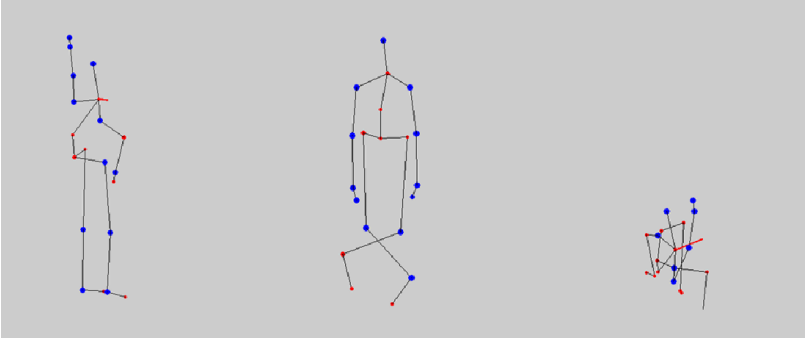

Figure 34 Example of noisy (Left), mildly incorrect (middle), and really incorrect skeleton data.

As a side note: some remarks on the contents of the MSR 3D action dataset and the use of the action subsets such as stated in Table 1 should be made. Although it does not add anything to the discussion on the performance of our own novel representation, it is debatable if the combined set of actions in subset 1 and 2 are really a combined group of similar actions: when observing all the different samples one by one, the actions in subset 1 and 2 do not seem to have that similar movements at all, and neither is the choice of 'complex' actions very logical. Finally, for the Microsoft Research 3D Action dataset a beta version of the Kinect is used instead of the finalized Kinect product, as was confirmed by Zicheng Liu by email. Although it is not certain, this could have been one potential cause of the many inaccuracies in the dataset.

5.1.2 A novel 3D action dataset

This novel dataset is recorded using Matlab in combination with Microsoft's Kinect and consists of a set of 12 different actions which are performed by 9 different subjects. Every subject performs each action 3 times and faced in 3 different orientations relative to the depth sensor: -45 degrees, 0 degrees, and +45 degrees. Care is taken that a diverse set of subjects are used for the dataset: it is a good mix of different lengths, posture, gender, and age. This leads to a total dataset of 108 samples per person, or 972 in total. The sampled data is stored in individual Matlab '.mat' files per sample, and each sample contains:

- Scaled down (to save data size) 240*320 uint8 grey value image.
- Scaled down (to save data size) 240*320 uint8 depth map.
- 'Flat' Joint series (20x2) for projection of joints onto images.
- 3D Joint coordinates (20x3) in meters relative to the sensor.
- Scaled down 240*320 binary segmentation map (note: due to bad scaling of the sensor drivers, this map turns out not to binary.).

The actions in the dataset are the following:

1. Stand still - This is, in fact, an action that occurs a lot and can be considered 'neutral'.
2. Sit down - Sit on a chair from a standing position.
3. Sit still - Second neutral action of the dataset: a person sitting still on a chair.
4. Tie laces - An action such as tie laces / pickup / pull on shoes often leads to incidents.
5. Drink - Drinking is often done in a sitting position and could be an important action to track while monitoring the health of elderly people.
6. Stand up - Stand from sitting position: often leads to incidents as people can easily fall.
7. Bend / pickup - Bend / pick something from floor: one of the major causes of fall incidents.
8. Side-reach - Reach towards the side to pick up an object.
9. Wave right - Could be used as signaling by an elderly in future systems.
10. Double wave - Could be used as signaling by an elderly in future systems.
11. Point up - Could be used as signaling by an elderly in future systems.
12. Point down - Could be used as signaling by an elderly in future systems.

The action pairs sit down/stand up, point up/point down and wave right / double wave are chosen because of their similarity in movement. The static actions stand still and sit still are chosen as they are not often used in datasets although they are in fact occurring very often, plus the two actions can be used to train a neutral pose classifier for the online action recognition framework. Considering the choice of actions, care is taken that some actions look familiar and that the actions are not too far from what you can expect to see in the daily activities of elderly and some signaling actions. Of course, this still does not mean that a good performance of this dataset also indicates that a good action recognition performance can be achieved when used on data obtained from real elderly people.

The setup of the recorded scene was as follows: a Kinect was placed in a freed up area in the storage department of DAZA Opticare, on a 1m elevation and was carefully aligned with the floor. Markings were made on the floor at a distance of ~3m and a chair was placed, together with a few stacked boxes. The chair was used for the actions that involved 'sitting', the stacked boxes were used for the action 'side reach'. The markers on the floor ensured that the setup was identical under each of the three different rotations. In Figure 35, an example is given of a subject performing the action double wave under three different orientations. Note that the chair in the lower right image of the example is accidently segmented by the Kinect's algorithm as being part of the person, but that the skeleton representation is still extracted. Also note that it is a known issue for the first Kinect version that accurate leg positions are not always available. Often times the legs 'shake' a bit. Perhaps these issues were solved in the new Kinect that is released with the Xbox One but at the time or recording no such device was available.

Note that the action data is recorded in such a way that it is already segmented in time reasonably well: the actions start and end within a few frames of the start and end of the recoded action sample. The scaling down of the depth map and RGB data is done to reduce the size of the dataset significantly. The 3D point cloud can be obtained by converting the depth map using the publicly available Matlab function 'depthToCloud' by Liefeng Bo and Kevin Lai.

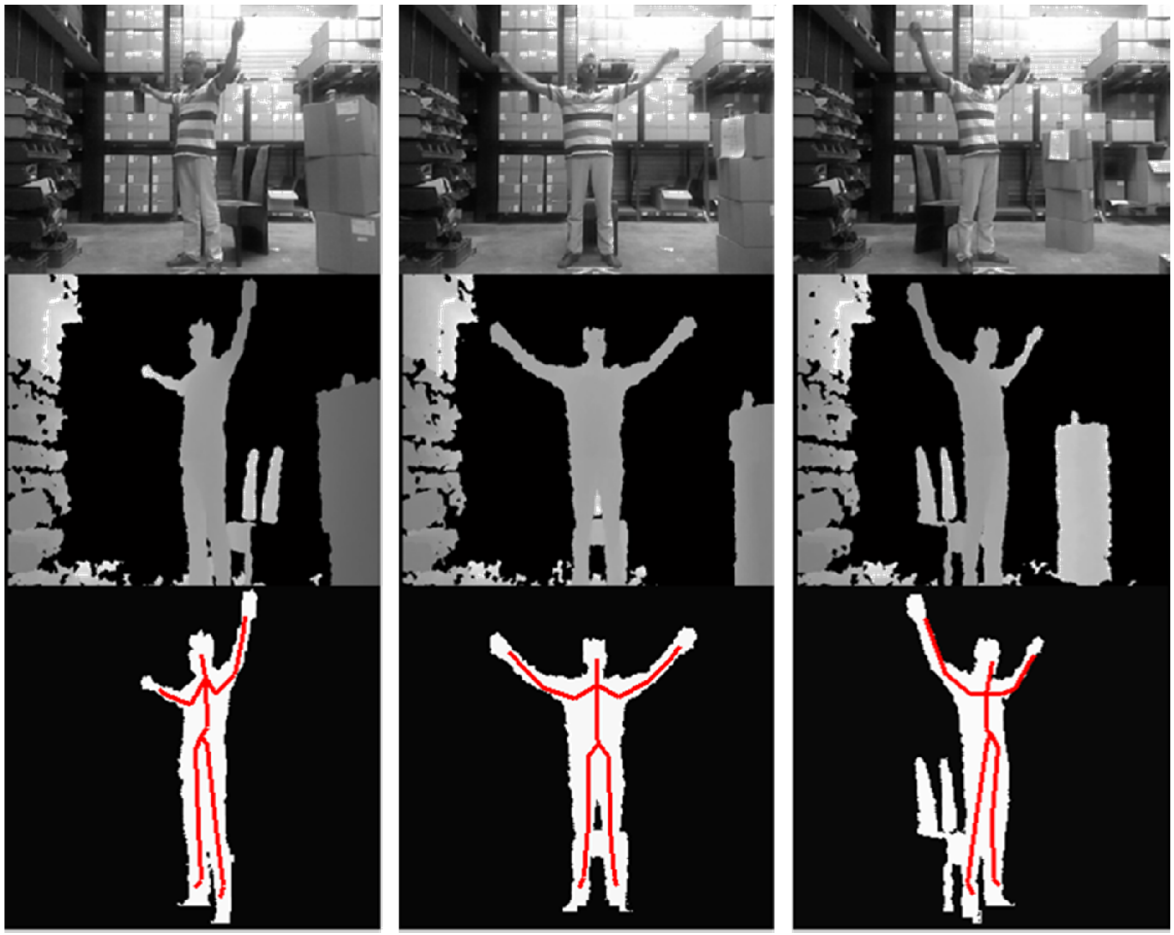

Figure 35 Frames of the action '2 hand wave' recorded from different orientations. (Frame 23/16/29)
Top row: gray value data. Middle row: depth data. Bottom row: segmentation map with skeleton overlay.

5.2 An offline exploration of the novel MH-STP representation

This section explores the different aspects of the MH-STP It also shows preliminary results and small improvements to the representation, followed by the effects of these improvements.

5.2.1 Preliminary results of the MH-STP validation test

Initial test setup

Using the methodology as described in paragraph 5.1.1, three tests are performed on the three action subsets that are stated in Table 1. The tests are designed as follows:

1. In this test $1/3^{rd}$ of the samples of each action is used for training, $2/3^{rd}$ is used for testing.
2. In this test $2/3^{rd}$ of the samples of each action is used for training, $1/3^{rd}$ is used for testing.
3. In this test half of the *subjects* is used for training and the other half for testing.

The third of these tests is considered to be the most challenging test: cross subject testing is considered in general to be a harder problem because of the variance in the way actions are performed and the variance of appearance of the different subjects.

The initial parameter setup of the representation are set up in such a way that it allows comparison of results with other articles. The MH-STP 3D space resolution is set up to have 10 bins in all three directions and the time is split up in 3 time segments. This leads to a total of 3000 space time cells. As one cell is 20 cm wide in this case, a joint interpolation distance of 2.5 cm is considered to be sufficient: this way, the interpolated limbs between the skeleton joints have sufficient chance to 'touch' a space time cell it crosses. A higher resolution would most likely only increase computational times while hardly increasing recognition rates.

When it comes to the actual build-up of the MH-STP feature, the MHI delay is set to 100: the larger the delay, the longer movements remain visible. The value 100 is higher than the frame count of the longest action sequence, so it is guaranteed that even the first observed joint locations remain visible in the representation. Once the representation is built up, compensation for different action sequence lengths and speed of performance can be achieved by normalizing the MH-STP values from the domain [minvalue .. maxvalue] to the domain [0..1].

The last parameter chosen is the dimensionality to be reduced to as the original dimensionality of 3.000 is much too high: a lot of the dimensions would only contribute to noise or contain redundant information. Although the original STOP article claims to have used a reduced dimensionality of 300 it remains unclear how this was achieved as the maximum dimensionality to reduce to using OCL is the amount of available training samples used in the training phase. This maximum dimensionality to reduce to is thus equal to the amount of training samples in test 1 on subset 1, which is $1/10^{th}$ of the total data set or about 40.

Different subsets, different results

This set of parameters leads to preliminary results as shown in Table 2. In this table, the reported best results of the research 'Action recognition based on a bag of 3d points' by Li et al [25] and the reported best results of the STOP method are added for comparison. The results column indicated with Skt. is a result obtained by the authors of the STOP method as comparison for their own method. In this test they are using fast Fourier transform (FFT) coefficients on the curves of the joint positions over time, which at that time apparently gave the best recognition results when using skeleton data.

The results show that our novel MH-STP method, without optimizing for any parameter, is showing similar performance to the method as proposed by Li et al and it shows much better results than the other method that used skeleton representations. Unfortunately, performance is still 4 to 10 percent lower than the best results achieved by the STOP method. Nonetheless, this first result is a promising outcome of the preliminary test as our method uses far less training samples than other works and uses non-optimized parameters.

The results table does also show some outcomes that require some further discussion. The performance increase between test 1 and test 2 is most likely an effect of the increased amount of training samples. This performance increase can also be observed in subset 3 although this subset obtains much higher results over all tests: the set of actions in subset 3 has more intra class variance and it is likely that it is therefore easier to classify. However, when looking at the differences in the result of the different subsets this large difference is not seen in the methods of Li et al. and STOP.

One possible explanation for the differences between tests is that the MH-STP in its current form might not be very suitable to allow the classifier to deal with variances of an action introduced by different persons. This is especially the case when a low amount of persons is used in the training set and a person that has not been used in training performs the observed actions. As appearance, rotation and size variance are already excluded as much as possible, this leads to the idea that the main cause of this drop of performance in test 3 is caused by inter action variances. Subset 3 is less affected because it has more intra class variance.

	Test 1				Test 2				Test 3			
	Li et al	Skt.	STOP	MH-STP	Li et al	Skt.	STOP	MH-STP	Li et al	Skt.	STOP	MH-STP
Set 1	89.50	68.00	98.23	89.64	93.30	72.97	99.12	95.15	72.9	40.28	84.7	68.45
Set 2	89.00	73.86	94.82	88.52	92.90	70.67	96.95	90.49	71.90	50.00	81.30	64.66
Set 3	96.30	78.67	97.35	95.26	96.30	83.78	98.67	96.90	79.20	73.91	88.40	86.55
Mean	91.36	73.51	96.80	91.14	94.20	75.81	98.25	94.18	74.70	54.73	84.80	73.22

Table 2 Preliminary results using a 10x10x10x3 grid, OCL dim 40, MHI delay 100.

After manually inspecting the dataset this thought was confirmed: some people tend to perform the actions much faster than other persons, which can lead to different MH-STP representations. This is happening because the frame rate of the depth sensor is only 15 frames per second which causes a semi arbitrary split of the action to 15 different action snapshots and the points in these frames are used to fill the STP. Note that this 'discretization' of the performed action even leads to large differences in the MH-STP of actions performed by the same subject and this is very likely the cause of the large variation in performance.

5.2.2 Improving the MH-STP representation by 3D skeleton interpolation

To make the MH-STP representation less dependent on the speed the action is performed at and to reduce the dependency of the discretizing nature of the frame rate, an inter-frame interpolation operation was added to the MH-STP creating algorithm. This operation interpolates the motion of the intermediate skeleton representation between two frames.

The operation works as follows: at the first frame an interpolation step is performed between the 3D joint locations of a certain limb. This is also done on the second frame. If two sets of interpolated points belonging to the same limb are known, points are interpolated between the limbs. An example frontal MH-STP view can be seen on Figure 36. Although the speed of performance and frame rate discretization still can lead to differences in the STP, the majority of variance should now be accounted for. The results obtained with the exact same approach as the first test but this time with a different MH-STP method are shown in Table 3.

The modification turned out to have a beneficial effect and especially the recognition rate of the cross subject dataset is increased with an average increase of 3.4%. This time, note that although the large difference in performance between subsets is still present, higher recognition rates are achieved than the STOP method for subset 3, test 3, and on average higher results are achieved than the method of Li et al. unfortunately, the modification did not lead to a lower difference between the different subsets. As in subsets 1 and 2 the actions are more similar to each other than in subset 3, it is still thought that this is a potential cause of this difference.

	Test 1				Test 2				Test 3			
	Li et al	Skt.	STOP	Our	Li et al	Skt.	STOP	Our	Li et al	Skt.	STOP	Our
Set 1	89.50	68.00	98.23	90.95 (+1.31)	93.30	72.97	99.12	94.70 (-0.45)	72.9	40.28	84.7	72.66 (+4.21)
Set 2	89.00	73.86	94.82	90.15 (+1.62)	92.90	70.67	96.95	91.26 (+0.77)	71.90	50.00	81.30	66.73 (+2.08)
Set 3	96.30	78.67	97.35	96.42 (+1.16)	96.30	83.78	98.67	98.27 (+1.38)	79.20	73.91	88.40	90.47 (+3.91)
Mean	91.36	73.51	96.80	92.51 (+1.37)	94.20	75.81	98.25	94.74 (+0.56)	74.70	54.73	84.80	76.62 (+3.40)

Table 3 Second set of preliminary results using a 10x10x10x3 grid, OCL dim 40, MHI delay 100. This time, our method used an interpolated intermediate skeleton representation. The difference indicator tells how much better the recognition performed compared to the non-interpolated counterpart.

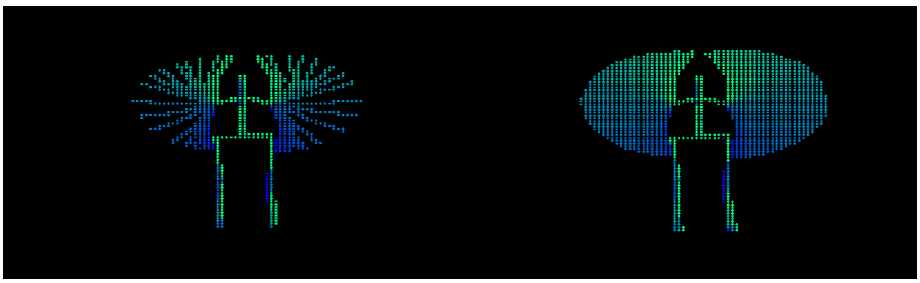

Figure 36 Left: a 30x30x30 STP built from the intermediate skeleton representation. Right: STP built from the same skeleton representation, but this time the intermediate skeleton representation movements are interpolated.

5.2.3 Preliminary results on the novel dataset

Using the improved MH-STP on the novel data set, the set of results as given in Table 4 is acquired. Results look very promising, as all tests have reported achievable results of over 90% recognition rate. Although the high recognition rate might partially be explained by the increased number of samples in this dataset, there is also an increased number of actions to classify between (12 instead of 7) which makes correct classification harder. The same parameters as with the MSR Action 3D dataset are used for the test: a 10x10x10x3 grid for the STP, leading to 3000 space time cells, which is filled by an interpolated skeleton representation using an MHI delay of 100. The resulting MH-STP is leading to a high dimensional vector whose dimensionality is reduced using OCL dim 40, and classification occurs using an offline classifier.

This time the results of the 'improved' method that uses the additional intermediate skeleton representation interpolation are not much better than the original MH-STP method. However, there is also a much lower difference between the results of test2 and test3, compared to the difference of test 2 and test 3 on the MSR 3D Action data. This suggests that the actions are performed more similar by the different persons in the novel dataset so that the inter subject differences in performance are lower. After visually inspecting this turns out to be the case: there exists less fluctuation in the recordings of the actions and there is less noise. Perhaps this is caused by the fact that subjects were shown first what kind of actions were expected from them.

	Test 1		Test 2		Test 3	
	MH-STP	Interp. MH-STP	MH-STP	Interp. MH-STP	MH-STP	Interp. MH-STP
Total dataset	94.49	94.52	95.40	95.67	92.21	92.58

Table 4 Preliminary results using a 10x10x10x3 grid, OCL dim 40, MHI delay 100. Comparison of the MH-STP method the 'improved' MH-STP that uses interpolated skeleton movement (2).

Although the test indicates a very good result, the question remains whether the MH-STP method leads to better results than the STOP method if that method were to be tested on the novel dataset. Therefore the STOP implementation as described in [27] is recreated in this work as well to try and make a sensible comparison. The parameters of the STOP representation are set equal to the most likely parameters used to publish the results of STOP in the original work. They are as follows: a 10x10x10x3 grid for the STP, dimension reduction using OCL to dimensionality 300 and a saturation parameter of 128.

Note that for this test, the point clouds are location normalized in order to center the STP around the skeleton in an appropriate manner. This is done by centering the STP on the lower back joint of the skeleton. How this is done in the test setup of STOP is unknown as location normalization was not discussed in that article, so this location normalization might already lead to better results than obtained in the original STOP set of results. As a secondary test, the point cloud is also rotation normalized before recreating the STOP representation. This is done with the help of the intermediate skeleton representation which was originally not used. It is expected that this would increase the recognition rates of the original STOP method a lot. Note that this method is not at all used nor discussed in the STOP article and this is built in solely to see how the MH-STP results would hold up against such an improved STOP method.

The results from these tests are shown in Table 5, together with the results drawn from the MH-STP representation test. The first outcome that can be observed is that our method has a higher recognition rate in test 1, and a lower recognition rate in test 2 when compared with the original STOP method. This could be caused by the fact that the original stop method seemingly needs more training samples to overcome the rotation in the samples, as this rotation variance is not present in the MH-STP representation.

A result that was not expected was the outcome that the STOP method that was rotation and location normalized (by the author of this thesis) does perform slightly better in test 1 and test 2 than our novel MH-STP method. This is a surprise, as it was considered that the appearance invariance of the MH-STP gave it a benefit over the STOP method. However, it should be noted that the MH-STP representation is not parameter optimized at this moment in contrast to the parameters used in the STOP method.

Finally, one more result can be observed: for the challenging test 3, the cross subject test, the proposed method in this thesis performs much better than the original STOP method and even 5% better than the rotated STOP method; even prior to parameter optimization. This result gives a good indication that the proposed method indeed starts to perform better in comparison to the STOP method once more inter subject variance is present: as the systems are not trained with all subjects in this test, the STOP method did not learn to deal with these variances in appearance and hence results start to drop, whereas the MH-STP method is much less influenced by this variance. More specifically, as during all different tests the exact same datasets are used for training and testing, it can now also be assumed that the rotation variance contributes to at least 8% recognition rate drop in this case.

	Test 1			Test 2			Test 3		
	STOP	Rotated STOP	Interp. MH-STP	STOP	Rotated STOP	Interp. MH-STP	STOP	Rotated STOP	Interp. MH-STP
Total dataset	91.84	96.13	94.52	96.06	98.16	95.67	79.43	87.11	92.58

Table 5 Preliminary STOP results using a 10x10x10x3 grid, OCL dim 40, MHI delay 100.
A comparison of the STOP method to the 'improved' MH-STP of this work that uses interpolated skeleton movement (2). Also, a version of the STOP method that has been improved (made invariant to rotation/location) by the author of this thesis is shown for comparison.

5.2.4 The influence of the MH-STP settings on its performance

As the first preliminary tests are performed and an indication that the novel method might indeed outperform other existing methods is found, deeper investigation into the representation is needed to determine which parameters might lead to an improvement of the system. Hopefully, this also gives some insight into the question what the strong and weak parts of the MH-STP representation are.

5.2.4.1 The use of normalizations

One of the key points of the MH-STP method is the possibility to make the representation invariant to location, rotation, size of the person and other appearance related variances. In this paragraph the first three of these will be discussed individually. The forth one, appearance variation, influences how well the variations can be dealt with, as this variance largely determines how well the skeleton algorithm is able to estimate a correct skeleton. Although it is very likely that there are points of improvement for this algorithm, this aspect is not discussed in this research.

Realizing invariance to location

Location invariance is very useful since the person that is being observed is not always performing actions at the same location: instead, he or she is moving through the observed scene. As the STP requires that it's 'space' is centered on the region of interest, creating an invariance to location basically means that the STP stays centered on the persons' location, independent of the movements of that person. For example, one could take the average 3D point cloud position or use a bounding box around the person of interest. In case of the MH-STP this can be realized more easily by making use of the estimated intermediate skeleton representation. By centering the STP on the lower back joint, almost half of the skeleton is below and half of the skeleton is above the center of the STP. Also, this joint stays on its place for most of the actions, which makes it a good position to center the STP.

Realizing invariance to rotation

One of the benefits of using the skeleton as intermediate representation for the MH-STP is that the skeleton representation immediately reveals to what direction the skeleton is facing. Using his information, it is not very hard to compensate for rotations around the axis of the skeleton by making use of 3D rotation matrices. In this research rotations around the longitudinal axis are of importance, to allow normalization of skeletons facing any right or left direction relative to the camera. Secondly, a rotation around the transverse axis, as is performed by the skeletons torso when a person is bending, is not of importance as this is encoded as useful information in the STP. Thirdly, rotations around the frontal axis can only be caused by incorrect placement of the depth camera, which is considered not to be the case in this research.

One crucial element of choosing this rotation normalization is the determination in which direction the skeleton is facing. The heading is usually determined in related works by:

1. A vector perpendicular to two vectors: joints neck – left hip and neck - right hip.
2. A vector perpendicular to two vectors: lower back – left shoulder and lower back – right shoulder.
3. A vector perpendicular to a plane, best-fitted to both the L/R hips and L/R shoulders.

Upon visually inspecting the data samples in both the MSR Action 3D dataset and the novel dataset, it turns out that the joint estimation of the shoulders is more stable than the joint estimation of the hips. This observation is also visible in the MH-STP results as shown in Table 6. Therefore, to achieve rotation invariance, the first of the described methods is used, as this leads to the best results. Note however that the resulting difference is not very high and that it does not seem to matter that much, which means that depending on the actions observed it could well be that another method is more suitable. Nonetheless, method one is chosen to be used in this research.

	Test 3: cross subject validation		
	Direction method 1	Direction method 2	Direction method 3
Average result	92.31 %	91.32%	91.40%

Table 6 Direction normalization method results using a 10x10x10x3 grid, OCL dim 40, MHI delay 100, average results of 100 repetitive tests.

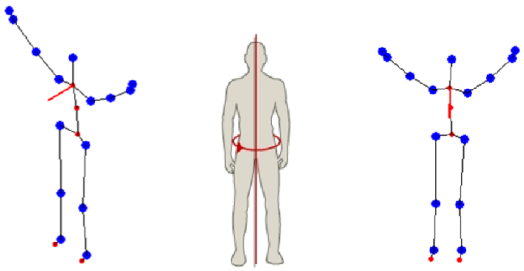

Figure 37 Left: non-camera facing skeleton. After a rotation around the longitudinal axis, the skeleton now faces the camera. Only longitudinal rotations are corrected for the MH-STP.

Realizing invariance to size

The skeleton representation not only allows making the MH-STP representation invariant to rotation and location, it also allows for size independency. This is achieved by averaging the limb lengths over all data samples in both the MSR Action 3D dataset and the novel action dataset, to obtain a set of average limb lengths, which is shown in Table 7. Using this data, the limbs of any skeleton representation, obtained from a large or a small person, can be resized to the limb size of average Joe. Note that this does not cause the rotation angles of the skeleton joints to change!

Additionally, an unexpected benefit of the size normalization was discovered. Apart from normalizing limb sizes that differ from person to person, it turned out the normalization also reduced limb size variances of the skeleton estimation itself. For instance, the 'hand' and 'foot' limbs are often very noisy. Also, some actions lead to skeleton poses that are hard to estimate. For instance, when someone was pointing in the direction of the depth sensor, the sensor had difficulties to estimate the size of the limbs. Using the size normalization, this variation was now also accounted for.

Limb	Avg.Size in m	Limb	Avg.Size in m	Limb	Avg.Size in m	Limb	Avg.Size in m
Neck	0.20	L-hand	0.09	Low-back	0.08	L-lower leg	0.36
L-Shoulder	0.21	R-upper arm	0.26	L-Hip	0.11	R-lower leg	0.36
R-Shoulder	0.21	R-lower arm	0.25	R-Hip	0.11	L-foot	0.09
L-upper arm	0.27	R-Hand	0.09	L-upper leg	0.49	R-foot	0.09
L-lower arm	0.25	High-back	0.36	R-upper leg	0.49		

Table 7 Average limb lengths of skeleton data as present in both MSA3D and novel dataset.

5.2.4.2 Varying the dimensionality: effect on recognition rate

So far the MH-STP representation is only created using a 10 x 10 x 10 grid and 3 time segments, leading to a dimensionality of 3.000 space time cells, which is being reduced to 40 dimensions using PCA. This initial dimensionality of the representation and the dimensionality that is being reduced to however do not necessarily lead to the best recognition rate. This section discusses the relationship of the STP dimensionality and reduced feature dimensionality to the recognition performance and computational demand.

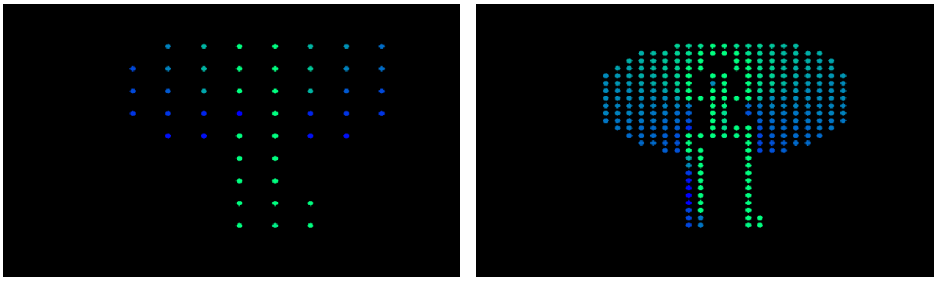

Figure 38 Left: an MH-STP using a grid of 10x10x10 Right: an MH-STP using a grid of 30x30x30.

Effects of changing the dimensionality to reduce to

As indicated in Section 2.9, a high amount of features is not a guarantee for high recognition rates. To find which features are most informative and useful for the classification process, the dimensionality reduction method orthogonal class learning or OCL is used, which is a method closely related to PCA. Unfortunately there is no general optimal amount of dimensions to reduce to. This depends on the amount of training samples available, the way the STP is filled and the dimensionality of the STP. An example can be seen in Figure 38 Left: an MH-STP using a grid of 10x10x10 Right: an MH-STP using a grid of 30x30x30.

If the dimensionality to reduce to is low, the required computing power is lower but chances increase that important descriptors of feature relations that vary a lot between different actions are not taken into account. Especially in cases where a high intra class variance exists this will lead to a drop of performance as in this case these descriptors are not as strong, and therefore more of them are needed to achieve a higher recognition rate. If the dimensionality to reduce to is too high, redundant information and noise are potentially taken into account too much. Also, this increases the computing power that is needed.

The mentioned effects on the recognition rate are visible in the left graph of Figure 39. A dimensionality that is too low leads to a very low performance, but as the dimensionality increases the performance increases rapidly, up to the point where there is not much performance increase anymore or even a performance decrease. The latter is especially the case for the 10x10x10 grid on the cross subject test: when ~10% of its original dimensionality is used (300 out of 10x10x10x3 = 3000), noticeable hinder of noise or the use of redundant features is visible. The performance of the grids with sides of 20 and 30 does not show this effect, as the new dimensionality uses only 1% and 0.3% of the original dimensionality respectively.

The same observation can be seen on the right of the two graphs, this time when sufficient training material is present and there is less hinder of the variance compared to the cross subject test. The MH-STP results are compared with the STP that is filled using the (rotation normalized) STOP method. As can be seen, both the STP's with grid sizes 10 and 20 that use the STOP method need a higher dimensionality to perform reasonably well, compared to their MH-STP counterparts. This is expected as the MH-STP is built up using the intermediate skeleton representation that offers stronger relations between space time cells than is the case when the STP is filled using noisy point clouds.

However, when higher dimensionalities are allowed, the rotation normalized STOP method can sometimes outperform the MH-STP method on higher dimensional grid sizes. One possible explanation could be that the point clouds recorded of an action cross many more space time cells and perhaps help to average the general motion of the action with the saturation scheme over the space time cells, rather than trying to describe the motion very precisely using the intermediate skeleton representation. This could prevent the added inter class variance that will be discussed in the next two paragraphs.

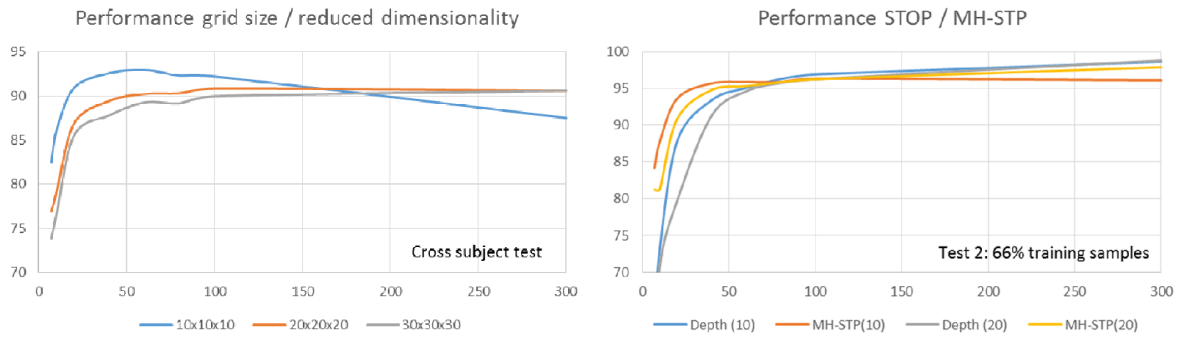

Figure 39 showing the performance in two different tests for different sizes of dimensionality reduced feature vectors.

Effects of changing amount of time segments

The 4D MH-STP so far had 3 separate segments in its time dimension. The main benefit of having different time segments in the representation is the additional information this provides: instead of only being able to capture a certain movement of the action, it is now also known if this movement was in the beginning, middle or end of the action. An example of a STOP representation with three time segments is given in Figure 3: a person performs the action 'kick' and as can be seen from the sequence, most of the movement occurs in the middle part of the action.

There are some drawbacks too when using time segments. For starters, as the total MH-STP representation is 4 dimensional, each added time segment increases the dimensionality by the grid dimension to the third power. In a 10x10x10 grid this adds 1.000 dimensions to the total MH-STP, up to a dimensionality increase of 27.000 for a 30x30x30 grid. This dimensionality increase requires additional computing power.

A secondary drawback of using multiple time segments relates to the fact that not every action is performed in the same way and that time segmentation of the actions is not perfect. This can lead to one subject performs a 'kick' action that is primarily placed in the middle segment of the MH-STP, while another subject performs the actual kick action only slightly later in the action sequence, causing the actual kick to be placed in the last or in the last two segments. This might cause the introduction of a variance in the representation which will most likely lead to deteriorating results in case more time segments are used.

The effects on the recognition rate can be observed in the results as shown in Figure 40. In the left graph, the results improve a lot when time 3 instead of 1 time segment is used, followed by a decline in performance. In the right graph, where a higher dimensionality is used for the reduced feature vector, the results slightly increase when another 2 time segments are added, after which the results start to decline.

The performance increase is likely due to the increased informativeness of using three time segments instead of one. The observation that further addition of time segments does not improve the results a lot is most likely caused by the introduced inter class variance and possibly to the amount of features used relative to the original dimensionality. The computational demand however keeps increasing with every added time segment, as the dimensionality increases. This is shown in the final paragraph of this section.

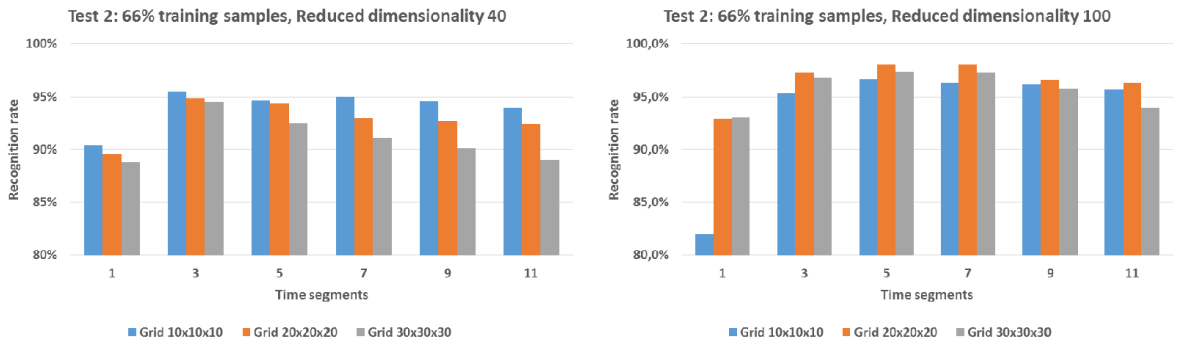

Figure 40 Comparison of performance between different amounts of time segments.

Effects of changing the grid size

Just as was the case for the amount of time segments, the resolution of the MH-STP grid also influences recognition results a lot. It is a delicate balance between loosing information when the amount of dimensions is too low and a computational demand that increases rapidly as the dimensionality increases. For instance, using a 40 x 40 x 40 x 3 grid will lead to a 192.000 dimensional feature vector which puts a lot of pressure on the available processing power. However, although a 5x5x5 grid will only lead to 125 dimensions, but fails to capture movements of individual body parts due to the fact that one cell corresponds to an area of approximately 40x40x40 cm.

One other effect of using a high grid resolution is similar to the described drawback of increasing the amount of time segments: it may lead to the introduction of variance caused by the exact way an action is performed: in an STP with large bins similar movements lead to the same space time cells being crossed by the skeleton representation while in an STP with very small bins it is more likely that each movement leads to different space time cells being crossed during the motion of the action. This effect is more noticeable when lower amounts of training samples are available as the system cannot be trained to cope with these variances.

This effect can be observed in the test results as shown in Figure 41: an OCL feature mapping towards a very low dimensionality is less able to benefit from the intra class variances due to the introduced additional inter class variance if a higher grid resolution is used to create the STP. Only when the feature dimensionality reduction is leading to a sufficiently high dimensionality (e.g. 300), the feature is again gaining informativeness as it is now more able to cope with the inter class variances. This effect can also be observed by looking at reduced dimensionality 40 and comparing the difference between 33% and 66% training results: there is a larger difference between the results of the 10x10x10 grid and 30x30x30 grid in the case of 33% training data. Looking at test 3, these effects are expected to be worse, which is indeed the case. It can be observed, as up to the point of using a reduced dimensionality of 100, that the 10x10x10 grid dimension leads to a better performance, as the larger bins of the STP most likely allows for a better handling of the inter class variance that is less accounted for in the cross subject test.

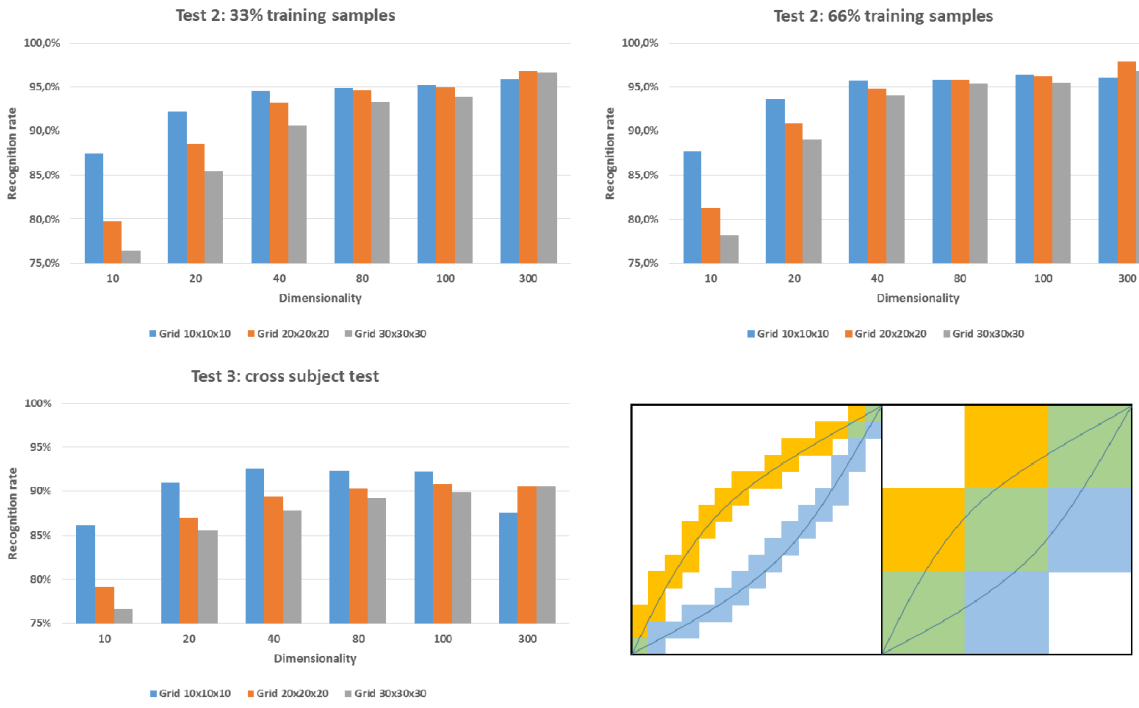

Figure 41 The effects of grid resolution, showing the effects of the dimensionality reduced to and grid size chosen in three different training situations. Lower right: the idea behind the added inter class variance .

5.2.4.3 Varying the dimensionality: effect on computational demand

As indicated in the previous sections that discuss the effects on recognition rate, changing the parameters also have their effect on the computational demand of the used method. The computational demand on generating features, training and testing are reviewed separately, and a comparison is made for different methods, dimensionality to reduce to, STP grid dimension and the amount of time segments used.

Computational demand of generating the STP

The STP generation algorithm the novel MH-STP method uses is not independent of the used grid dimensionality and time segments, as between the joints of the intermediate skeleton representation more points need to be interpolated when the grid dimensions increase. This is to ensure that at least all the space time cells that the limbs cross during their movement are activated in the STP. For the STOP method this is not the case: the amount of 3D depth points used is exclusively dependent on the input resolution of the depth sensor and the region of interest.

The STOP method in general requires much more points to be processed than is needed for the MH-STP representation, but it has no further needed algorithms such as interpolation, rotation and size normalization. Instead, the depth points can be mapped directly to the STP. In terms of required memory however, the MH-STP requires much less space as it only needs to store 20 3D skeleton joint positions per frame instead of a large collection of raw 3D depth points, typically around 6.000 – 9.000 points.

The differences in processing time and used memory of the used methods and the used grid dimensionality is shown in Table 8. When large grid dimensions are used, the STOP method is 4-5 times faster than the MH-STP method, but for a lower grid dimensionality there is hardly any difference. However, it is questionable how bad the higher 0.009 seconds per frame (And: not on a very fast computer) of the MH-STP method for the larger grid size really is: as long as the rest of the algorithm is not too time consuming, it still allows the processing of data at very high frame rates.

A larger difference can be seen in the required memory to store and use the recorded data and the memory required to store a single STP. The STOP method uses ~500-600 times as much memory due to the much larger amounts of points that need to be stored per frame and a single STP with a resolution of 30x30x30 requires 27 times more memory than an STP with sides of 10x10x10, which is of course an expected result. This means that to store the entire novel dataset in memory and use it as training and testing data, 1.76GB of memory is needed in case of a 30x30x30x3 STP, which is in strong contrast to the 69MB of memory needed for a 10x10x10x3.

	Data size in dataset per frame	Data size in memory per frame	Data size in memory of one STP	Processing time in seconds per frame
MH-STP 10x10x10x3	480 B	480 B	23 kB	~ 0.002
STOP 10x10x10x3	300 kB	~200 kB[1]	23 kB	~ 0.002
MH-STP 30x30x30x3	480 B	480 B	633 kB	~ 0.009
STOP 30x30x30x3	300 kB	~200 kB[1]	633 kB	~ 0.002

Table 8 Memory and processing time comparison. [1]Average values for STOP method are used.

Computational demand of calculating a mapping to reduce the STP dimensionality

In the graph in Figure 43 results are shown of the computational demand to compute the mapping from the set of training samples towards the reduced dimensionality. The dimensionality to reduce to and the grid dimensionality of the STP are both compared to offer a clear comparison. Note that these timings are obtained by code in Matlab; it is therefore very likely that further optimization is possible by either writing the code in e.g. c++ or code optimization in Matlab itself. However, the resulting graph does give insight in the relative processing time between the results obtained by using different parameters.

One resulting outcome that is immediately visible, but which is also quite trivial, is that the larger the dimensionality of the STP is, the more computing power is needed. On average, calculating the mapping of a dataset with 600 samples towards a lower dimensionality by using a 30x30x30x3 STP requires 11,5 second, which is 4 times more than using a 20x20x20x3 representation and 20-25 times more processing time than is the case when a 10x10x10x3 dimensional STP is used. While this does not lead to much problems in an offline setting, it might be problematic in case the method should work in an online setting, as this might require a frequent retraining of the dataset.

Another result is that in general the higher the dimensionality is to reduce to, the higher the required processing time. This is due to the fact that the training samples need to be mapped to the new reduced dimensionality too, and it is more computational to do so for 81.000 dimensional vectors than 3.000 dimensional vectors. However, in the training phase this contribution to the total computational time needed is limited.

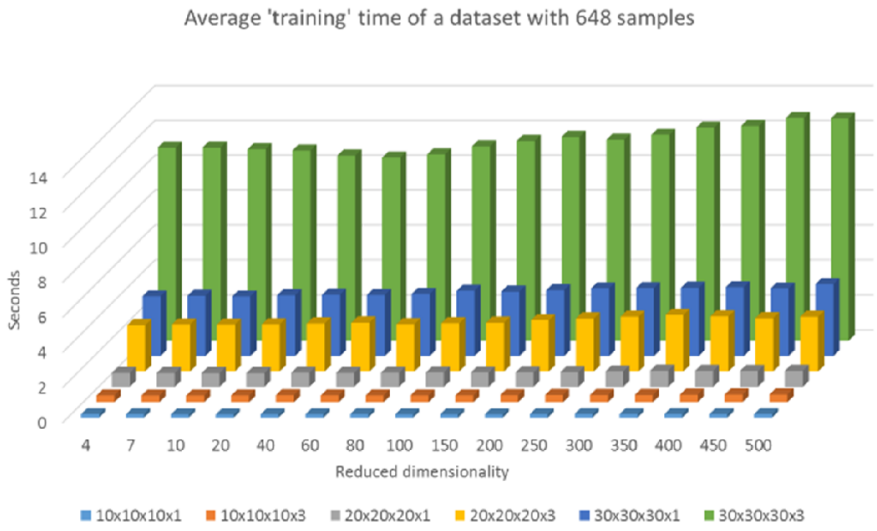

Figure 42 Comparison of computational demand for the 'training' phase of offline classification.

Computational demand classification

In contrast to the results in Figure 42, the computational demand of classifying 1000 STP representations using the offline classification scheme as discussed shows a much greater dependence to the size of the reduced feature, as can be seen in the top graph of Figure 43. The computational dependence on the initial STP dimensionality is clearly visible in this phase as well. This is caused by the fact that the classification of an STP consists of two parts: mapping the STP towards a lower dimensionality and determining to which class the average distance of the training samples to the tested sample is lowest. The first part is related to the grid dimensionality and reduced dimensionality and the second part is related to the reduced dimensionality and the amount of available training samples.

Looking at the resulting computational time in detail, the contribution of both these two parts to the total computational time is given in the lower 2 graphs of Figure 43. Clearly, the dependence on the size of the reduced dimensionality is visible in both graphs. The dependence of the computational time on the grid size of the STP is as discussed only seen in the part where the STP is mapped to a vector of reduced dimensions. Although not directly visible from the graphs, as the axis are not linear, the amount of original space time cells and the reduced dimensionality have a linear relation with the amount of computations needed. One sample mapping of an STP with high dimensionality M towards low feature dimensionality N takes $\mathsf{N(2M-1)+M}$ operations, where the distance calculation towards all training samples S takes $\mathsf{S(6N+1)}$ operations. As in general $M \gg N$ and $M \gg S$, this too shows that most of the processing power is used to map the STP to the reduced dimension.

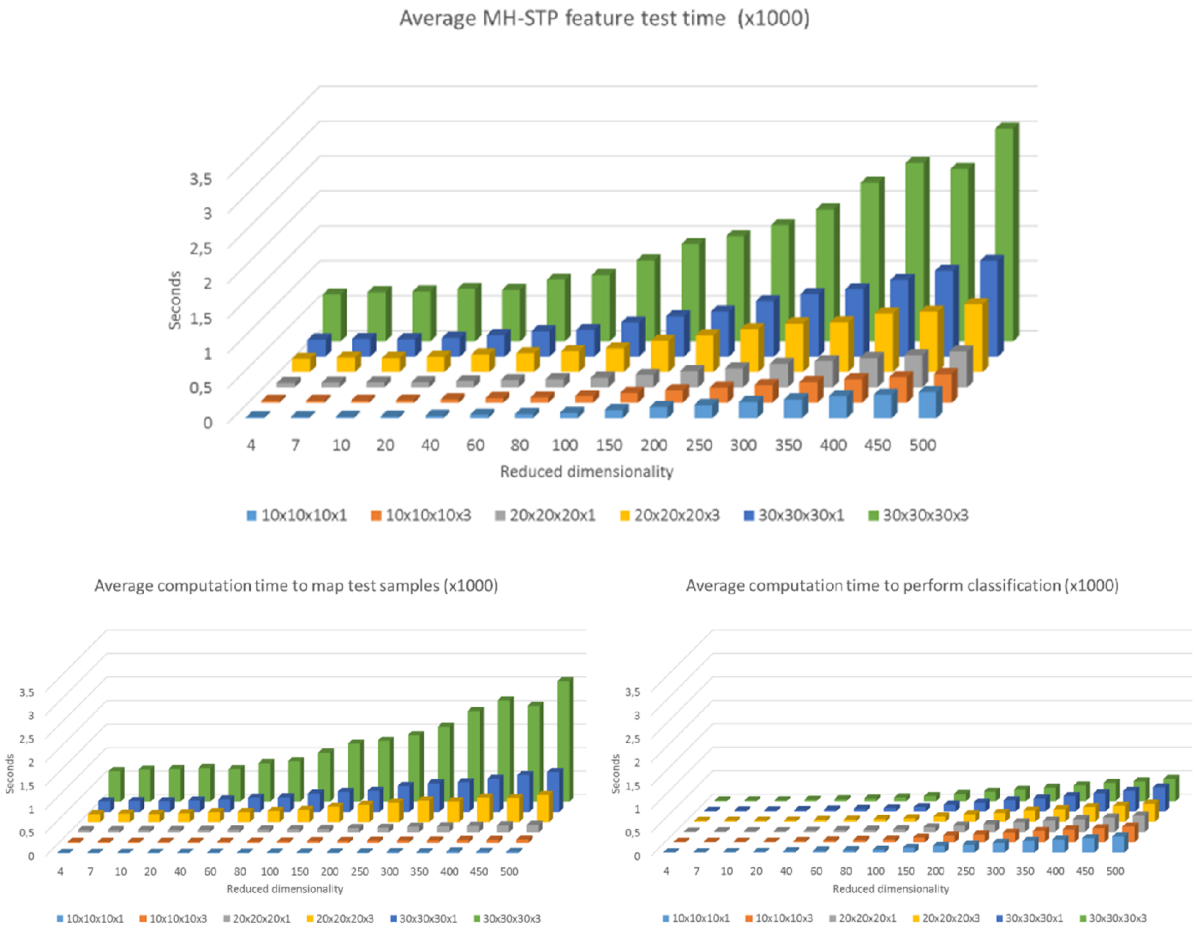

Figure 43 Comparing the computational demand of testing an MH-STP, with effects shown on reduced dimensionality and grid size. Clearly visible is the large effect of the grid size on the performance.

5.2.5 Speed and accuracy tradeoff of the MH-STP

Although section 5.2 is about the offline performance of the MH-STP, in an online situation up to 5 MH-STP representations need to be generated per second. These all need to be mapped towards a lower dimensionality and finally the sequence of representations needs to be classified. A higher grid size or higher reduced dimensionality may increase the results, but may also make the process too slow for a successful application. This paragraph discusses the tradeoff between speed and accuracy of the MH-STP.

The results from previous sections show that at least the use of a 30x30x30x3 STP is not advised. The reason for this is that an STP with this dimensionality requires too much processing power and memory, as the mapping towards the lower dimensionality requires is an expensive operation. Also, results are not much better and sometimes even worse than the situations where a 10x10x10x3 or 20x20x20x3 dimensionality is used.

To get a better overview of the tradeoff between accuracy and required processing times, Figure 44 shows a normalized table. To obtain these values, the corresponding speed and accuracy are independently normalized and weighed equally. The higher the score, the greener the color in the figure and the better the result. The result is as expected: for lower dimensionalities too much information is lost, for higher dimensionality the processing time increases plus for a lower grid size the classifier over-fits. Also, having three time segments is worth the extra computational demand even though the computational need is higher and it suffices to use a smaller 10x10x10 grid, as there is not much difference with the 20x20x20 case in accuracy but there is in required processing time.

OCL dim →	7	10	20	40	60	80	100	150	200	250	300	350	400	450	500
10x10x10x1	63%	69%	82%	87%	88%	88%	87%	86%	83%	78%	46%	58%	59%	60%	60%
10x10x10x3	77%	83%	92%	95%	95%	95%	95%	95%	95%	93%	92%	91%	90%	87%	81%
20x20x20x1	53%	60%	80%	86%	88%	89%	89%	89%	88%	87%	86%	85%	85%	84%	82%
20x20x20x3	69%	69%	84%	90%	91%	91%	91%	90%	90%	89%	88%	88%	86%	86%	84%

Figure 44 Table showing normalized scores that include both accuracy and computational time. This is tested for different dimensionalities and four different grid sizes. A reduced dimensionality of 40 in a 10x10x10x3 has the highest rating: best accuracy in combination with a low computational cost.

5.3 Lessons learned about the MH-STP in an offline setting

In this Chapter the properties of the MH-STP in an offline setting, supported with results, was discussed extensively. First of all it is important to notice that results, obtained using any method, do not mean much if no clear information is given on the data that was used as train and test data. In the offline action recognition evaluation using the MH-STP, two different data sets were used:

- An existing dataset from Microsoft, MS Action 3D that has been used in quite a few other works
- A novel dataset constructed especially for this thesis

The main reason for the use of two different data sets, is that it helps to prevent obtaining biased results. Does the method only work well on one dataset or does it also perform well on other datasets? There are many differences in the datasets. For starters, the novel dataset does contain more samples. Furthermore the MSA3D dataset has more different actions but these were always tested in subsets of 8. In the novel dataset, no subsets are made: all 12 actions are used at once. Also, it contains recordings under three different angles for all actions and subjects, which is not available in the MSA3D dataset. The Novel dataset was created with a different skeleton tracker, leading to less noisy results. In the MSA3D dataset, some data was manually discarded, as the skeletons tracked were simply not available or completely incorrect.

In the initial test using MSA3D data, where the MHI delay was set to 100, STP grid of 10 x 10 x 10 x 3 space time cells and an OCL mapped dimensionality of 40. Promising results were obtained, especially since there was no real initial knowledge of the behavior of the MH-STP representation was known. In test 1 (33% training), test 2 (66% training) and the challenging cross subject test only slightly lower than state of the art results were achieved. Only subset 1 and 2 in the cross subject had much lower recognition rates, most likely due to the fact that the low amount training samples could not deal with the lower inter-action variance.

Results improved even more (up to 3%) after a small improvement to the motion-history property of the representation. By interpolating between frames, intra-action variance in the features was reduced. Evaluating the method, without making alterations to the parameters, lead to excellent results on the novel dataset. For comparison, the STOP method was recreated and tested on the novel dataset twofold: one time with normal data, and one time with its point cloud data made invariant to location and rotation, using an intermediate skeleton. This adjustment to the original STOP method was tested as this was expected to be one of the biggest possible improvements. The results of the MH-STP representation were excellent. Although in test 1 the MH-STP performed slightly better than STOP and drew similar results in test 2, the cross subject recognition rate of the MH-STP was more than 12% higher. Results of the skeleton-'corrected' STOP method also showed an obvious improvement to the original the STOP method.

Different aspects of the MH-STP were explored. For starters, the influence of the method to make the datasets invariant to rotation. Although similar results were drawn it did show why a certain choice of method was made. Making the datasets invariant to size is also discussed: this reduces a large source of noise and it removes the differences between tall and small people. This method turned out to be very useful. As parameter of the actual representation, the effect of the STP grid dimensions on recognition rates was tested and discussed thoroughly. It was shown that a too low dimensionality lead to the removal of too much information but that a dimension that is too high would lead to much higher computational and memory demands. Also, it is thought that an increase of the dimensionality would lead to lower results as it would induce intra-action variance, which could only be coped with when more training data would be available.

Results with varying grid dimensions and reduced dimensionality furthermore showed that the MH-STP performed better with lower dimensionalities than traditional methods. Most likely due to the 'binning' nature of the STP a lot of intra-action variance was dealt with. This was especially shown in the situation where a low amount of samples was available with a lot of intra-action variance as in the cross subject test: the lower dimensionality and low grid size outperformed higher dimensional solutions. There is however one exception: the offline action recognition turned out to suffer from the removal of the 4th dimension from the MH-STP. If only one time segment was used instead of three, results were on average 10% lower. However if more than 3 time segments were used and less training data was available results started to deteriorate as well. The cause of this is most likely that the 'segmenting in time' is performed at an arbitrary moment. The impact of this phenomena becomes larger when more time segments are added, as is shown in the results.

The favor for these lower dimensions is a beneficial result, as it was also shown that higher dimensional grids and the use of higher dimensional feature vectors had very large influence on computational times. This was especially true in the representation generation and mapping phase. The actual classification did not suffer much from the higher dimensionality. This chapter on the offline recognition of actions provided important insight into the performance of the MH-STP. Now let's take a look at the performance of the online classifier in the next chapter, as this is a more important result since elderly people cannot be offered extra safety in their homes without the use of an online system.

Chapter 6

Experimental results in an online setting

Using the results and experience from the preliminary offline action recognition test discussed in Chapter 5, a better view on the usefulness and different properties of the novel MH-STP representation is now available. This lead to the belief that the MH-STP representation should also be useful for the online action recognition case. This Chapter shows the different results obtained from the various aspects of the online action recognition framework. The Chapter is split in five main sections: first the dataset that is used is revisited, followed by the results of the neutral pose classifier using the MH-STP. Next, results are shown of the single-stage online classifier, followed by an adjusted version of the classifier that employs a secondary action specific EM pass at the cost of extra computational demand. This is done in order to minimize the risk of 'unknown' actions that have to be regarded as incorrect results. Finally, the results obtained by the new two-stage classifier implementation of the MH-STP is discussed: is a classification over two stages, only for the otherwise unknown actions, a good replacement of the more computationally demanding implementation of the secondary EM pass to solve the problem of the unknown actions? Finally, at the end of this Chapter a summary of the results is given.

6.1 Dataset used for the online setting

In paragraph 5.1.2 the novel dataset that was recorded for this thesis was discussed. In short, it consists of a set of 12 different actions which are performed by 9 different subjects. Every subject performs each action 3 times and faced in 3 different orientations relative to the depth sensor: -45 degrees, 0 degrees, and +45 degrees. This leads to a total dataset of 108 samples per person, or 972 action sequences in total.

For the online action recognition part of this work, the same data is used but the action sequences are split into segments of 5 frames on forehand, leading to a set of 10686 segments all consisting of 5 frames only. Segments with a lower amount of frames lose their temporal information and with a higher amount of frames there would be too few segments in an average action. The total set of segments is stored in a Matlab data file, together with contextual information of the sample they are obtained from: the feature mode used to generate the short MH-STP, grid size, action type, subject, orientation, instance, and placement in the action sequence.

The reason for this pre-segmentation is twofold: first of all we need to mimic the sequential availability of data as is the case in a real time situation, secondly this pre-segmentation saves processing power when performing the actual tests and it allows for manual labeling of the segments: which action segments contain a neutral pose and which are part of an actual action or 'non-neutral'? Note that the manual labeling of all 10686 action segments is required since in our action recognition framework SVM has to be trained as a neutral pose classifier, in order to determine the start and stop of an observed action sequence. This is discussed in Section 4.3.

Of the data segments, 5494 segments are marked as neutral pose and 5192 as non-neutral pose. While all of the segments in action 1 (stand still) and action 3 (sit still) are marked as neutral pose, the other neutral poses are occurring, obviously, at the start and end of an action. In the online action recognition experiments, action 1 and 3 are only used to provide additional neutral poses to the SVM classifier, they are not used in the training or testing phase of the action graph classifier. This leaves the online action recognition test with 10 different actions.

6.2 Results of the neutral pose classifier

Determining which of the short MH-STP features as visualized in Figure 30 is not very hard if done manually, but as with most things in vision based technologies doing it automatically is not as simple. However, manual labeling of over 10k+ samples is very time consuming, not to mention the manual labeling of the segments in case a frame count of less than 5 per segment was chosen. The following two sections discuss the performance of the SVM with the MH-STP as representation in relation to its parameters and both show the results from automatic and manual labeling.

6.2.1 Results obtained using automatic labeling for various parameters

To prevent this amount of work, if possible, a first attempt in training the SVM classifier was done without manual labeling. Instead, all segments of action 1 (stand still) and 3 (sit still) are considered as neutral poses, as well as the first and last N segments of each action sequence using an automatic labeling process. Training was performed using $1/3^{rd}$ of all segments and tested using the other $2/3^{rd}$ of the segments. The results were averaged over 10 process runs in which the training and testing sets were randomly drawn.

However: how can we verify the performance of the SVM classifier without having ground truth, manually labeled data available? The first idea is by visual inspection, as our MH-STP is easily interpretable. However, this would be as much work as manually labeling the data in the first place. A second option is not to visually verify individual segments, but the action sequences instead. After all, a neutral action such as standing still is expected not to have any of its segments classified as non-neutral and a non-neutral action is expected to start with a neutral segment, then have some non-neutral segments and finally end with a neutral segment again.

This would still require the manual inspection of all 900+ action sequences, note down the performance (good or bad) and average the results, every single time a test is performed. It would be better to have an average result for each individual action, but unfortunately an action can be performed very fast or very slowly, resulting into some sequences with only 5 segments, while others have 10. This problem is solved by using a linear time normalization (similar to [65]) over the segment results and then averaging over all sequences of that action.

This allows us to visually compare the effects of different parameters on the performance of the neutral pose classifier, such as the effect of grid size, the effect of the amount of frames N at the start and end of the sequences to be labeled as neutral (Figure 45, Figure 46) and PCA dimensionality to reduce to (Figure 47). In each of the figures the action sequences start at the top of the figure and end at the bottom of the figure. The situation that is thought of as 'most likely to be expected' is given in the most left result, the other views obtained by varying the parameters can be compared with this result. Note that the neutral/non-neutral percentages are displayed because these were the actual average percentages acquired but it is much more informative to visually inspect the red shape that gradually shifts between neutral and non-neutral over the course of the action.

The easiest visual inspection is looking at the results of the stand still action under different circumstances, since all the segments of the action should be marked as neutral. In the case of N = 3, misclassification occurs frequently in both grid sizes. This is most likely due to the automatic labeling of segments: the beginning and or ending segments of some non-neutral actions were marked as neutral while in reality they were not. This could occur because of the fact that some subjects started their action later than others while they were asked to do so, causing neutral segments to be labeled as non-neutral segments. The fact that less misclassification occurs when N=4 supports this thought. Another effect can also be noted: when the grid is 20x20x20 more misclassifications occur. This could be caused by the fact that the MH-STP is less able to capture variances when its grid size increases.

Stand still action, different N and grid size

	Expected ideal result		*N = 3, Grid = 10x10x10*		*N = 4, Grid = 10x10x10*		*N = 3, Grid = 20x20x20*		*N = 4, Grid = 20x20x20*	
	%neut	%nonneut	%neut	%nonneut	%neut	%nonneut	%neut	%nonneut	%neut	%nonneut
start	100%	0%	96%	4%	100%	0%	89%	11%	96%	4%
.	100%	0%	89%	11%	100%	0%	89%	11%	89%	11%
.	100%	0%	100%	0%	100%	0%	95%	5%	95%	5%
.	100%	0%	88%	13%	100%	0%	100%	0%	100%	0%
.	100%	0%	100%	0%	100%	0%	85%	15%	100%	0%
.	100%	0%	100%	0%	100%	0%	100%	0%	100%	0%
.	100%	0%	100%	0%	100%	0%	100%	0%	100%	0%
.	100%	0%	100%	0%	100%	0%	100%	0%	100%	0%
.	100%	0%	95%	5%	100%	0%	100%	0%	100%	0%
Stop	100%	0%	100%	0%	100%	0%	100%	0%	100%	0%

Figure 45 Comparing the effects of grid size and N value on the SVM performance for a neutral action

The mislabeling effect that is most likely causing misclassifications is also present the other way around. When an action sequence is short, the amount of segments is low: an N value of 3 will already assign 6 of its segments a neutral label. It is very likely this is the cause why so many segments around the middle of the sequence are classified as neutral. Especially when the variance is less accounted for in the 20x20x20 case, Figure 46 shows that on average $8/10^{th}$ of the action is classified as neutral in $1/3^{rd}$ of the time! Only the middle segment is classified as non-neutral in all occasions, while it is expected that at least the middle 2-4 should be near 100% non-neutral. Of course, the red curve is still clearly visible but the results are much more distributed over the neutral/non-neutral bars then was expected.

Two hand wave, different N and grid size

	Expected ideal result		*N = 3, Grid = 10x10x10*		*N = 4, Grid = 10x10x10*		*N = 3, Grid = 20x20x20*		*N = 4, Grid = 20x20x20*	
	%neut	%nonneut	%neut	%nonneut	%neut	%nonneut	%neut	%nonneut	%neut	%nonneut
start	100%	0%	67%	33%	85%	15%	70%	30%	89%	11%
.	30%	70%	41%	59%	50%	50%	36%	64%	55%	45%
.	10%	90%	29%	71%	29%	71%	21%	79%	36%	64%
.	0%	100%	13%	88%	19%	81%	19%	81%	31%	69%
.	0%	100%	0%	100%	0%	100%	0%	100%	0%	100%
.	0%	100%	9%	91%	14%	86%	5%	95%	14%	86%
.	0%	100%	10%	90%	38%	62%	5%	95%	38%	62%
.	10%	90%	56%	44%	56%	44%	44%	56%	56%	44%
.	30%	70%	78%	22%	100%	0%	78%	22%	96%	4%
Stop	100%	0%	89%	11%	96%	4%	89%	11%	96%	4%

Figure 46 Comparing the effects of grid size and N value on the SVM performance for a non-neutral action.

The third comparison in Figure 47 shows the effect of the dimensionality to reduce to with PCA and shows the effects on the same non-neutral action as in Figure 46. The effect is very strongly visible. A lower dimensionality has more difficulties classifying non-neutral segments correctly, a higher dimensionality is no longer able to classify any neutral segment correctly anymore. The ideal dimension to reduce to seems to be 20 in this particular example. Note that this dimensionality is unfortunately not the same as the ideal dimensionality used for the classification of actions in the offline case. Also, it might also not be the best dimensionality for action recognition in the online case. However, multiple separate mappings might be performed if the recognition rate would be affected too much, although this would come at the cost of additional computational cost, unfortunately.

After having reviewed the outcomes for all the actions, unfortunately, automatic labeling of the segments did not look like a good solution, mainly due to the somewhat arbitrarily chosen value of N=3 or 4. Although the automatic labeling method the advantage that no manual labor was needed, the accuracy of the entire action recognition framework would most likely deteriorate too much by the performance of the SVM. Another option would be automatic labeling with the help of movement cues to determine the action start and stop position in the sequence, but because of issues with noise this idea did not seem plausible.

Two hand wave, effects of different PCA

	Expected ideal result		*Reduced dim = 5*		*Reduced dim = 20*		*Reduced dim = 40*		*Reduced dim = 80*	
	%neut	%nonneut	%neut	%nonneut	%neut	%nonneut	%neut	%nonneut	%neut	%nonneut
start	100%	0%	93%	7%	85%	15%	30%	70%	0%	100%
.	30%	70%	55%	45%	50%	50%	18%	82%	0%	100%
.	10%	90%	50%	50%	29%	71%	14%	86%	0%	100%
.	0%	100%	31%	69%	19%	81%	13%	88%	0%	100%
.	0%	100%	25%	75%	0%	100%	0%	100%	0%	100%
.	0%	100%	18%	82%	14%	86%	0%	100%	0%	100%
.	0%	100%	48%	52%	38%	62%	5%	95%	0%	100%
.	10%	90%	78%	22%	56%	44%	22%	78%	0%	100%
.	30%	70%	100%	0%	100%	0%	30%	70%	0%	100%
Stop	100%	0%	96%	4%	96%	4%	44%	56%	0%	100%

Figure 47 Comparing the effects of dimensionality on the SVM performance for a non-neutral action

6.2.2 Results obtained using manual labeling for various parameters

The findings from the previous paragraph justified the tedious job of manually labeling the segments. Therefore a small program was written to allow quick insight in the situation and do the manual labeling in the most efficient way. This led to the results as visible in Figure 48, where it can be seen that there are on average only minor differences with the ground truth; in fact the accuracy was 98.1%. Finally, a comparison with earlier expectations is shown: there are differences with the ground truth, but the manual labeling seems to be justified.

	Ground truth action 1		*Test results action 1*		*Ground truth action 10*		*Test results action 10*		*Expected action 10*	
	%neut	%nonneut	%neut	%nonneut	%neut	%nonneut	%neut	%nonneut	%neut	%nonneut
start	100%	0%	99%	1%	91%	9%	93%	7%	100%	0%
.	100%	0%	100%	0%	40%	60%	41%	59%	30%	70%
.	100%	0%	100%	0%	16%	84%	16%	84%	10%	90%
.	100%	0%	100%	0%	5%	95%	6%	94%	0%	100%
.	100%	0%	100%	0%	2%	98%	2%	98%	0%	100%
.	100%	0%	100%	0%	1%	99%	1%	99%	0%	100%
.	100%	0%	100%	0%	23%	77%	20%	80%	0%	100%
.	100%	0%	100%	0%	75%	25%	73%	28%	10%	90%
.	100%	0%	100%	0%	89%	11%	91%	9%	30%	70%
Stop	100%	0%	100%	0%	99%	1%	99%	1%	100%	0%

Figure 48 Comparing ground truth and test results for the neutral action 1: 'stand still', and a comparison with the ground truth for action 10: two hand wave, with its test data and the results that were expected earlier. Dimensionality was set to 20, grid to 10x10x10.

Although the average accuracy was over 98%, it is still a good idea to check in more detail which segments were misclassified, to be able to give an estimation of the impact on the overall classification. Upon reviewing multiple action sequences of subjects, it is very likely correct to assume that the misclassifications made by the SVM will have a minimal effect. The reason for this is that the misclassifications all take place at the boundaries of the actual action, meaning that most of the action is covered in the non-neutral action sequence. Because of the nature of the online action recognition method used, the action graph, it should not be a problem at all that one of the segments at the start, end or both of an action sequence are missing. On top of that it is questionable how wrong exactly the misclassification was, as the ground truth for ambiguous sequences is subjectively determined.

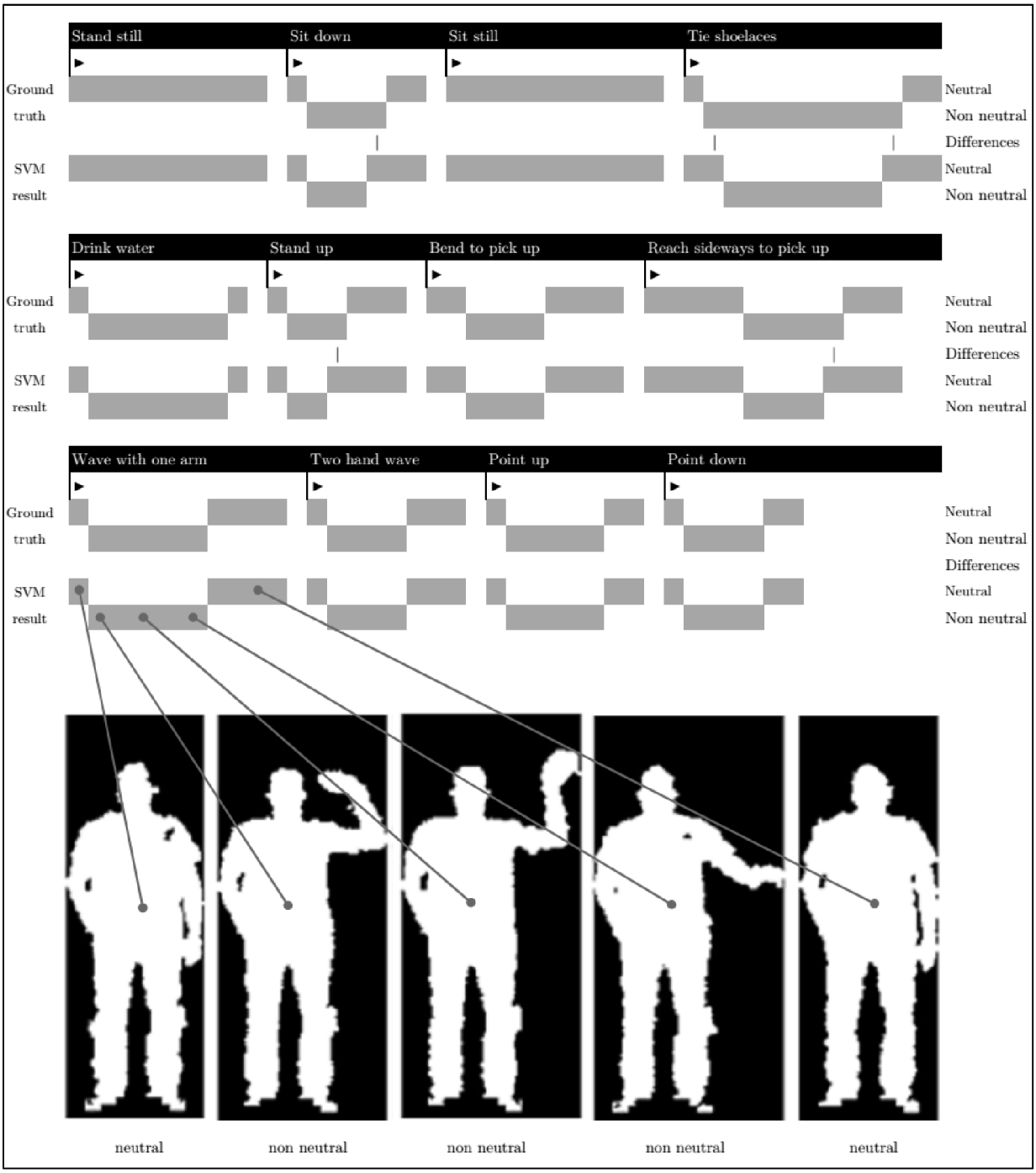

Figure 49 Neutral pose classification result of one entire series of actions performed by test person 5. Note: these are NOT the PCA-MHSTP features but instead only a silhouette of the person performing the action is shown for visualization purposes.

6.3 Results of online action recognition framework

With the segments now marked as neutral and non-neutral by the neutral pose SVM classifier, the set of segments that is most informative to the action can now be fed to the third phase of the action classification framework, which does the actual classification of the observed action sequence using an action graph. In this paragraph the results of the action graph classification method are given and the effects of the different parameters on the obtained results are shown. The effects of the following parameters on the recognition performance were studied: size of the reduced dimensionality, amount of salient poses to estimate and grid size. Note that it is tried to obtain the results for a wide range of variables, but that the computational power and memory demand formed a hard limit to obtain some results. The performance of the following classification modes is tested:

1. Global Viterbi using EM clusters, with global prior and transitional probability matrices.
2. Global Viterbi using EM clusters, with action specific EM passed prior and transitional probability matrices.
3. Global Viterbi using EM clusters, with global prior and transitional probability matrices, and a second stage classifier to cope with the unknown classifiers.

Similar to the offline case, the evaluation of the action is performed using three different tests

1. 33% training samples, cross instance test.
2. 66% training samples, cross instance test.
3. 50% training samples, cross subject test.

Test 1 and 2 are performed in three folds, to allow all instances to become part of training and the test data. Every fold is performed 10 times to compensate for the small variations in performance that the (most likely non-optimal) clustering algorithm introduces as clustering features to salient poses is a very important part of the action recognition framework. In test 1, the training set is created from all data corresponding to one of the three instances per sample, and the test set to the other two instances. In test 2, the training set is created from all data corresponding to two of the three instances. For test 3, five different sets of 5 subjects are chosen randomly. The other subjects are used as test set. This is performed in 5 folds and every fold is ran 10 times to limit influences of clustering algorithm as indicated for test 1 and 2.

6.3.1 Preliminary performance and dependence of variables tested

As an explorative test as performed in the offline recognition case, a subject cross validation test is performed as this is the most challenging task. All data samples of the 10 non-neutral classes were used for this test. The reduced dimensionality is varied from 5…40 and the number of salient poses is varied from 5…75 poses. The grid dimensionality of the MH-STP is set to 10x10x10. The results are shown in Figure 50. The training data contained 8 out of the 9 subjects and the test data consisted of the single subject left. Results were averaged over three folds in case of cross instance validation and all different subjects in the cross subject test.

Looking at the dimensionality, a familiar connection can be observed as in the offline result overviews: a lower dimensionality deteriorates the result (missing information), while a dimensionality that is too high also makes results worse (noise / redundancy). The amount of salient poses (or states) that form the nodes in the action graph also shows a strong relation to the recognition result. Too few nodes in the action graph means there is not enough possibility to capture the variance in the sequential nature of all different actions. Using too many nodes in the action graph leads to much less misclassifications as now the sequences can be captured better. However, the amount of observations that cannot be classified (unknown) increases with the amount of different nodes in the action graph, as more nodes requires more training data and increases risk on bias.

From the figure, it seems that the combination of reducing the dimensionality of the MH-STP to 15 and using 45 nodes as salient poses in the action graph leads to the best performance in this particular case: a successful recognition rate of 83%. The case where 65 salient nodes is used performs equally well, but might be a bit less suitable since the higher amount of nodes requires more computing power.

Finally, in the preliminary test it was also determined that the use of a 20x20x20 grid instead of the 10x10x10 grid did not improve results much at the expense of an 8 times higher processing load, making this an unsuitable alternative. Therefore the 10x10x10 grid is preferred for further research.

Misclassifications

states \ dims	5	10	15	20	25	30	35	40
5	65%	69%	53%	62%	56%	57%	58%	46%
10	57%	43%	36%	36%	36%	36%	30%	36%
15	41%	40%	35%	24%	23%	30%	34%	25%
20	33%	24%	18%	26%	25%	30%	21%	27%
25	22%	29%	18%	21%	18%	28%	17%	21%
30	27%	18%	15%	17%	16%	19%	23%	16%
35	24%	17%	13%	13%	18%	14%	17%	13%
40	21%	19%	15%	13%	15%	16%	13%	12%
45	18%	22%	11%	15%	12%	15%	16%	10%
50	20%	13%	11%	8%	13%	14%	15%	13%
55	18%	15%	12%	12%	9%	10%		
60	18%	13%	9%	9%	11%	11%		
65	21%	12%	4%	10%	6%	7%		
70	15%	12%	13%	7%	8%	6%		
75	19%	13%	5%	5%	11%	7%		

% of unknown actions

states \ dims	5	10	15	20	25	30	35	40
5	0%	0%	0%	0%	0%	0%	0%	0%
10	0%	1%	2%	1%	1%	1%	1%	1%
15	2%	2%	3%	3%	3%	4%	4%	3%
20	2%	5%	4%	4%	6%	4%	7%	2%
25	7%	5%	3%	4%	7%	7%	6%	6%
30	5%	7%	4%	4%	6%	9%	8%	10%
35	6%	6%	10%	6%	8%	9%	13%	14%
40	7%	6%	9%	10%	6%	12%	16%	13%
45	10%	12%	6%	10%	10%	19%	16%	20%
50	13%	10%	13%	17%	12%	17%	17%	18%
55	10%	12%	14%	11%	11%	15%		
60	14%	12%	11%	13%	14%	21%		
65	7%	14%	13%	20%	18%	21%		
70	15%	19%	16%	23%	15%	17%		
75	17%	15%	19%	17%	21%	19%		

Total classification issues

states \ dims	5	10	15	20	25	30	35	40
5	65%	69%	53%	62%	56%	57%	58%	46%
10	57%	44%	38%	37%	38%	38%	31%	37%
15	43%	42%	38%	27%	25%	34%	39%	29%
20	35%	29%	22%	30%	31%	34%	28%	29%
25	29%	34%	21%	25%	26%	35%	23%	27%
30	32%	24%	20%	22%	23%	28%	31%	26%
35	29%	23%	22%	19%	25%	23%	30%	27%
40	28%	25%	24%	23%	21%	29%	29%	24%
45	28%	34%	17%	24%	22%	34%	32%	30%
50	33%	23%	24%	25%	24%	31%	32%	31%
55	27%	27%	26%	23%	20%	24%		
60	31%	25%	20%	22%	25%	31%		
65	28%	26%	17%	31%	24%	28%		
70	30%	31%	29%	30%	23%	24%		
75	35%	28%	24%	22%	32%	27%		

Figure 50 Preliminary classification results using a cross subject validation with 1 subject as test and 8 as training data. Averaged over all 9 subjects.

6.3.2 Mode 1: Action recognition using global salient poses & global transfer matrices

In this mode of the action recognition framework, the salient poses (clusters within the feature space) are globally determined. Transfer and prior matrices are determined using this global dataset directly, without searching for optimal transfer and prior matrices in an action specific manner. This has as benefit that it requires relatively low amounts of processing power compared to the per-class optimized counterpart.

From the preliminary tests, it seems that the minimum dimension to reduce to should be 15. More than 25 increases the risk of 'unknown' actions, plus it directly affects the computational demand of the entire action recognition framework. About the amount of salient nodes can be said that the minimum amount should be no less than 35, to prevent a higher rate of misclassifications. The maximum amount should be 65. More nodes increases the risk of observed sequences to be classified as 'unknown' too much and directly increases the required computational power. Using the estimated best parameters (10x10x10 grid for the short MH-STP, dimensionality 15, 45 salient poses) from the preliminary test, the following results are obtained:

	Misclassification	Classified as Unknown	Total classification issues
Cross instance / 33% training	9%	23%	32%
Cross instance / 66% training	8%	11%	19%
Cross subject / 50% training	10%	18%	28%

Table 9 Average obtained results (10x10x10 grid for the short MH-STP, dimensionality 15, 45 salient poses).

	Misclassification	Classified as Unknown	Total classification issues
Cross instance / 33% training	6%	18%	24%
Cross instance / 66% training	4%	6%	10%
Cross subject / 50% training	7%	12%	19%

Table 10 Best obtained results (10x10x10 grid for the short MH-STP, dimensionality 15, 45 salient poses).

Looking at the results in Table 9 and Table 10 on page 96 the misclassification rate is low, even when only 33% training data is used a recognition rate of 91% is achieved on average (max 94%). However, the action recognition framework regularly fails to classify observed sequences leading to a high rate of observations classified as unknown. In the maximum case even 30% of the sequences were classified as unknown. As discussed earlier, this is likely related to the amount of available training samples, as difference between the 33% training data (average 23% unknown) and 66% training data (average 11% unknown) tests shows. In the cross subject test the amount of sequences classified is similar to the 33% cross instance test: although the cross subject test has more training data the introduced subject variance seems to have an effect on a good positioning of the salient poses.

One question that arises is which actions are most often classified as unknown actions. Table 11 shows the relative percentages of observed actions classified as unknown per type of test. The framework is remarkably often not able to classify actions 4 (tie shoelaces), 2 (sit down) and 6 (stand up) to a certain class. On the other hand, action 10 is almost always classifiable. The most likely explanation for the behavior on action 'tie shoelaces' is that this action contains the most skeleton noise of all the classes. By bending over from the chair to tie the shoelaces the skeleton representation often misaligns with the person. This might lead to representations and thus features being generated for salient poses that are mostly based on noise.

A possible explanation for the behavior of the classification framework on the pair of actions 'sit down' and 'stand up' is that the biggest part of the action is classified as neutral since both sitting and standing are neutral positions. This might lead to very short non-neutral sequences and only limited path possibilities directly after global trained. If any other salient pose is used in a test sequence, transfer probabilities will go to 0 easily.

Lastly, the very good performance of action 10 (two hand wave) is likely caused by the fact that this is a very clear movement that is perfectly captured by the skeleton and easily distinguishable from the other actions visually. On top of that, many non-neutral short MH-STP exist in sequences from this class as most actors performed the action rather slowly. Note that the action recognition framework also detected a two-hand wave that was performed three times in one recorded sequence: there was no problem to recognize the action as the action graph allows the detection of these cyclic movements.

Action	1	2	3	4	5	6	7	8	9	10	11	12
33% ci	n.a.	15%	n.a.	19%	4%	12%	10%	8%	10%	3%	12%	8%
66% ci	n.a.	18%	n.a.	20%	4%	15%	8%	6%	11%	1%	11%	6%
50% cs	n.a.	16%	n.a.	16%	9%	11%	9%	6%	11%	2%	10%	9%

Table 11 Distribution of observed sequences that cannot be classified and therefore result in an 'unknown' classification. Total per cross instance (ci) and cross subject (cs) test sums up to 100%.

Another useful view on the obtained results is shown in the confusion matrices in Figure 51. In all tests, action 11 (point up) is often mistaken for action 12 (point down). Confusion between pointing up and down can be caused by the fact that the action of pointing up fully overlaps the action of pointing down; only the end position differs. This introduces the risk that the salient poses as determined by the clustering algorithm are positioned in such a way that the different end position MH-STP representations result in the same salient pose. This might lead to misclassification.

Also, the point upward is often mistaken with action 9 (right arm wave) and action 5 (drink while sitting) just as 6 (stand up) is sometimes mistaken for 4 (tie shoelaces). One possible explanation is that the MH-STP grid size is of too low dimension, causing the MH-STP representations of point upwards and wave (with right arm) to look too much like one another. The same reasoning might explain the other misclassifications. Increasing the grid dimension to e.g. 15x15x15 or 20x20x20 might solve this issue but will demand both additional training samples and an increased computational demand.

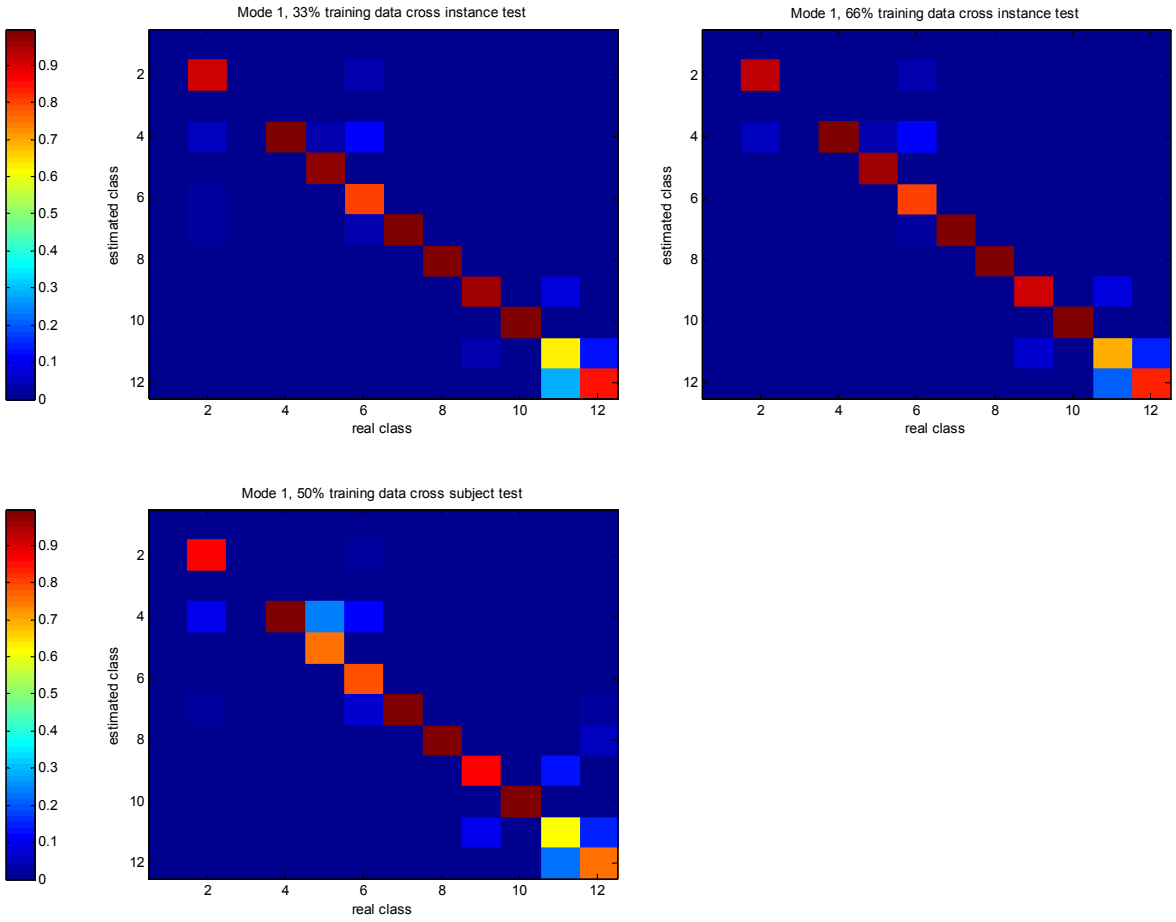

Figure 51 Confusion matrices for all three tests in classification mode 1. Darker red means more actions are classified correctly.

6.3.3 Mode 2: Action recognition with global salient poses & action specific transfer matrices

The percentage of observations that cannot be classified (and thus remain unknown) in the results in Table 9 and Table 10 are caused by the unknown salient pose transitions. To prevent the occurrence of these salient pose transitions that are not trained for in any of the action graph models ($p(s_1, s_2, \dots, s_n|\psi) = 0 \; for \; all \; \psi$) a second action recognition mode of the action recognition framework is evaluated.

All the transitional probabilities in all action specific transition matrices that are 0 after the initial training phase are replaced with a very low transitional probability to allow them to be updated. Then a secondary optimization will update the action specific transition and prior matrices while keeping the clustering of salient poses unchanged and equal to the global set. As this secondary optimization run needs to be performed on each individual class it might be less suitable in case the amount of distinguishable actions grows as it does require quite some processing power.

The results in Table 12 and Table 13 of the three different test shows a drastic overall improvement. The large portion of the action sequences that could not be classified in mode 1 is now no longer there. For instance, the cross subject classification improved from 39% total classification issues to 16%, while keeping the misclassifications equal. In the cross instance tests however, a small deterioration in misclassifications is visible unfortunately; which is likely caused by the possibility that action sequences that were previously not classifiable might be somewhat more difficult to classify correctly.

	Misclassification	Classified as Unknown	Total classification issues
Cross instance / 33% training	14% (+5%)	0% (-23%)	14% (-18%)
Cross instance / 66% training	12% (+4%)	0% (-11%)	12% (-7%)
Cross subject / 50% training	16% (+6%)	0% (-18%)	16% (-12%)

Table 12 Average obtained results (10x10x10 grid for the short MH-STP, dimensionality 15, 45 salient poses) and the performance delta to results obtained in testing mode 1.

	Misclassification	Classified as Unknown	Total classification issues
Cross instance / 33% training	9% (+3%)	0% (-18%)	9% (-15%)
Cross instance / 66% training	5% (+1%)	0% (-6%)	5% (-5%)
Cross subject / 50% training	11% (+4%)	0% (-12%)	11% (-8%)

Table 13 Maximum obtained results (10x10x10 grid for the short MH-STP, dimensionality 15, 45 salient poses) and the performance delta to results obtained in testing mode 1.

In this mode, the confusion matrices in Figure 52 shows similar results to the confusion matrices obtained from the tests performed in mode 1. This is particularly true for both cross instance tests. No major differences can be found; only a small amount of sequences from action 5 (drink while sitting) is now also classified as action 2 (sit down).

The cross subject confusion matrix shows stronger differences to the one obtained in mode 1. Most interesting is the increased misclassification of action 2 (sit down). The reason for this might be that this action had the most sequences classified to unknown in mode 1. This might also confirm the thought that actions previously not classifiable in mode 1 might be harder to classify correctly.

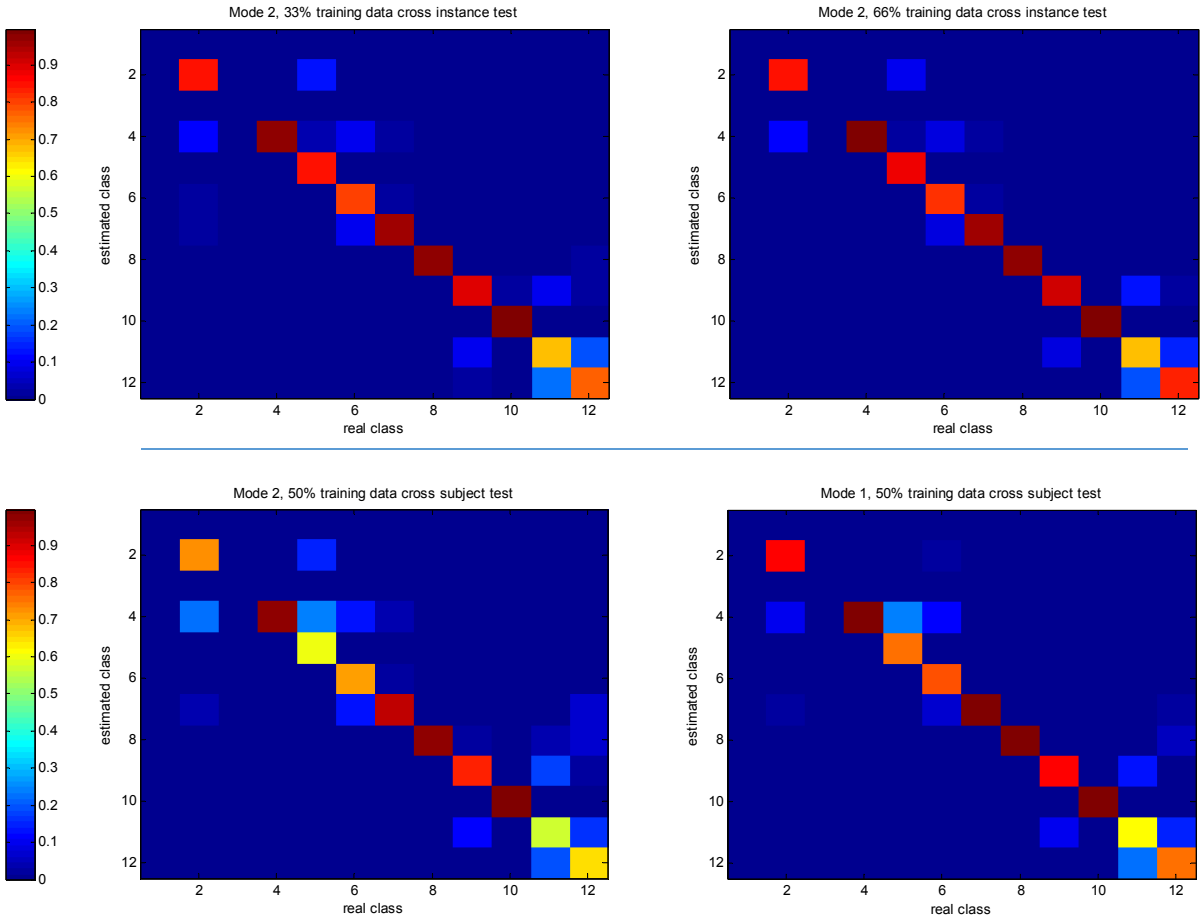

Figure 52 Confusion matrices for all three tests in classification mode 2. Darker red means more actions are classified correctly. The lower part shows a comparison of cross subject obtained confusion matrixes obtained mode 1 and 2 for clarity.

6.3.4 Mode 3a: Two-step action recognition with super positioned short MH-STP

The third mode of the action recognition framework is using a two-stage classification method that exploits the nature of the MH-STP representation by super positioning the short MH-STP features into a feature that spans the entire non-neutral part of the action. The unique point is that it keeps the action graph models working on a global dataset which is therefore easily expandable and (re)trainable with minimal effort while at the same time guaranteeing high recognition rates. As second stage classifier the straightforward cosine distance classifier is used as discussed in Chapter 4, with its results published in Chapter 5.

The results as shown in

Table 15 should be interpreted as shown in Figure 53; B is the portion of all tested samples A that are misclassified directly at stage 1. D is the misclassified portion of all C tested samples using stage 2 that could not be classified in stage 1. The total set of misclassified samples is B+D. Note that in the result tables, percentages are used and these cannot simply be added! The total misclassification percentage is (B + D) / A.

A first glance at the results learns that the proposed novel two stage classifier easily outperforms mode 1 and even outperforms mode 2, which is the 'standard' method as proposed in the work of Li et al. [25] and used in the STOP article [27]. Misclassification in stage 1 and the amount of actions not classifiable by the system are roughly equal to the results obtained in mode 1, but the misclassification percentage in stage 2 seems to be rather high in comparison with results obtained in the offline setting. This is most likely caused by the fact that the full MH-STP used in this second stage classification method is only 10x10x10x1 instead of the preferred 20x20x20x3 situation as in the optimal offline case, as the online classifier works in 10x10x10 dimensions. Furthermore the full MH-STP in the online second stage case is only containing the non-neutral information and this might influence the performance of the second stage classifier a bit as well.

The confusion matrices show that the same misclassifications are made, most particularly between the actions 11 (point down) and 12 (point up) but certainly less misclassifications are occurring in comparison to mode 2. Confusion matrices for the 33% and 66% are very similar. In the cross subject test no performance increase in comparison with the per-class optimized classification mode 2 is noticeable. However, computational demand is much less! Also, when comparing the confusion matrix of mode 3 with the one obtained with mode 2, it can be seen that less mistakes are made within actions 2..6.

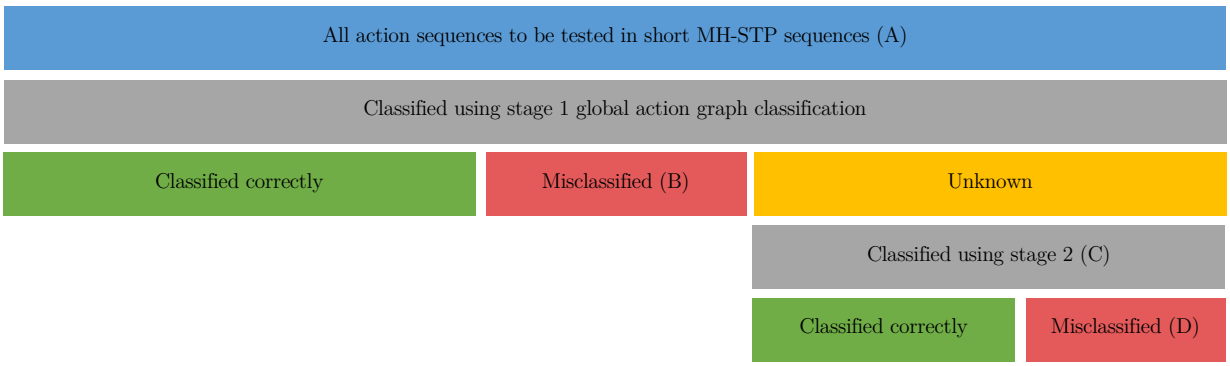

Figure 53 Schematic overview that shows how to interpret the results of the two stage classifier.

	Misclassification in stage 1	Classified as Unknown	Misclassification in stage 2	Total classification issues + deltas
Cross instance / 33% training	9%	22%	15%	10% (-22% / -4%)
Cross instance / 66% training	9%	11%	11%	9% (-10% / -3%)
Cross subject / 50% training	10%	18%	20%	12% (-16% / -4%)

Table 14 Average / Maximum obtained results (10x10x10 grid for the short MH-STP, dimensionality 15, 45 salient poses, 2nd stage 10x10x10 dimensionality 40). Results are given per test and individual parts of the test. The total classification issues (mode1 / mode2) are compared with mode 1 and mode 2. A decrease is shown in green and means it is an improvement over the other modes.

	Misclassification in stage 1	Classified as Unknown	Misclassification in stage 2	Total classification issues+ deltas
Cross instance / 33% training	6%	18%	11%	8% (-16% / -1%)
Cross instance / 66% training	6%	8%	0%	6% (-4% / 0%)
Cross subject / 50% training	7%	12%	16%	8% (-11% / -3%)

Table 15 Average / Maximum obtained results (10x10x10 grid for the short MH-STP, dimensionality 15, 45 salient poses, 2nd stage 10x10x10 dimensionality 40). Results are given per test and individual parts of the test. The total classification issues (mode1 / mode2) are compared with mode 1 and mode 2. A decrease is shown in green and means it is an improvement over the other modes.

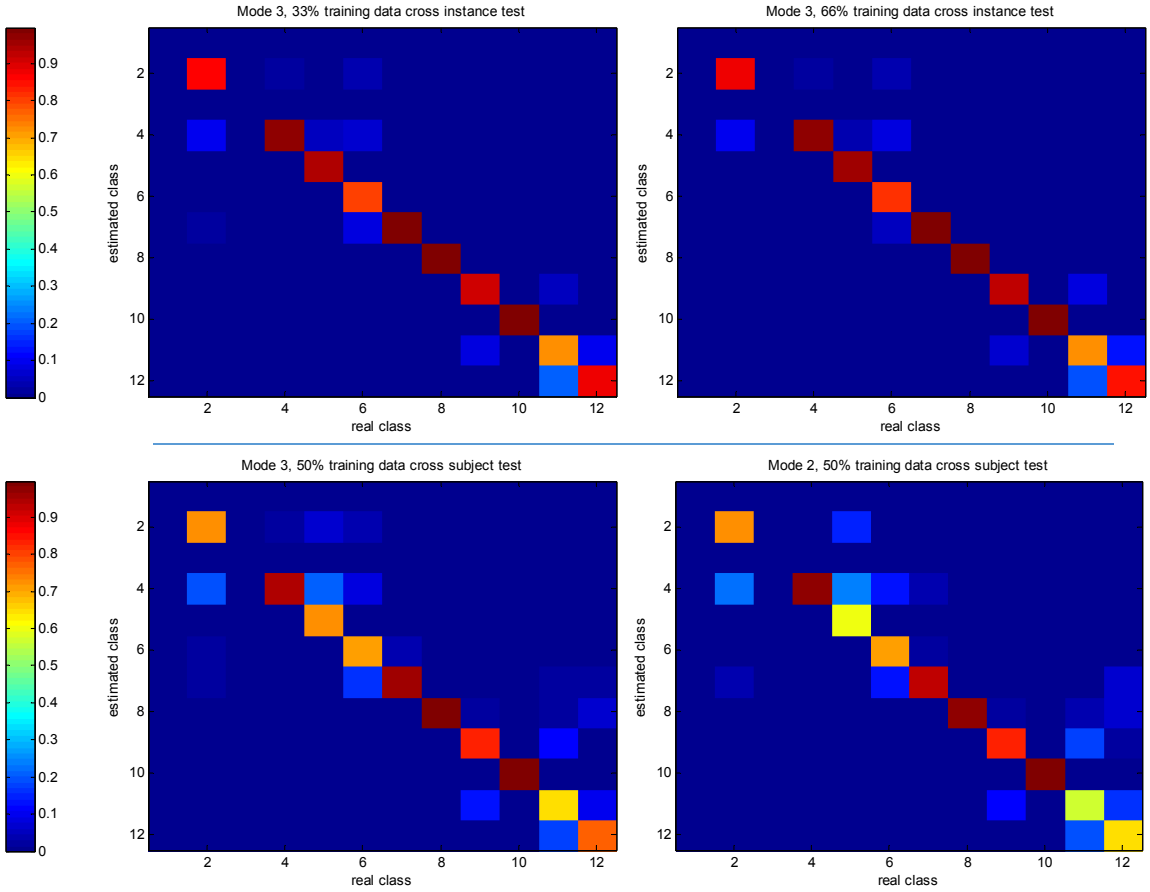

Figure 54 Confusion matrices for all three tests in classification mode 2. Darker red means more actions are classified correctly. The lower part shows a comparison of cross subject obtained confusion matrixes obtained in mode 2 and 3 for clarity.

6.3.5 Mode 3b: Two-step action recognition with super positioned short MH-STP and confidence check

Apart from the possibility to classify action sequences with stage 2 of the action recognition framework that could otherwise not be classified, the two stage classification method offers the possibility of reclassification. Reclassification is something to consider in case two or more action graph models return probabilities that lie very close to one another. In this report, the relative threshold was set to a factor 100. The results as shown in Table 17 should be interpreted as shown in Figure 55: misclassification in stage 1 is indicated with B, classified as unknown are all yellow fields and misclassification in stage 2 (D) is taken as the sum of all red (misclassified) fields at the lower bar. The total classification error is the confident part of the misclassifications plus D.

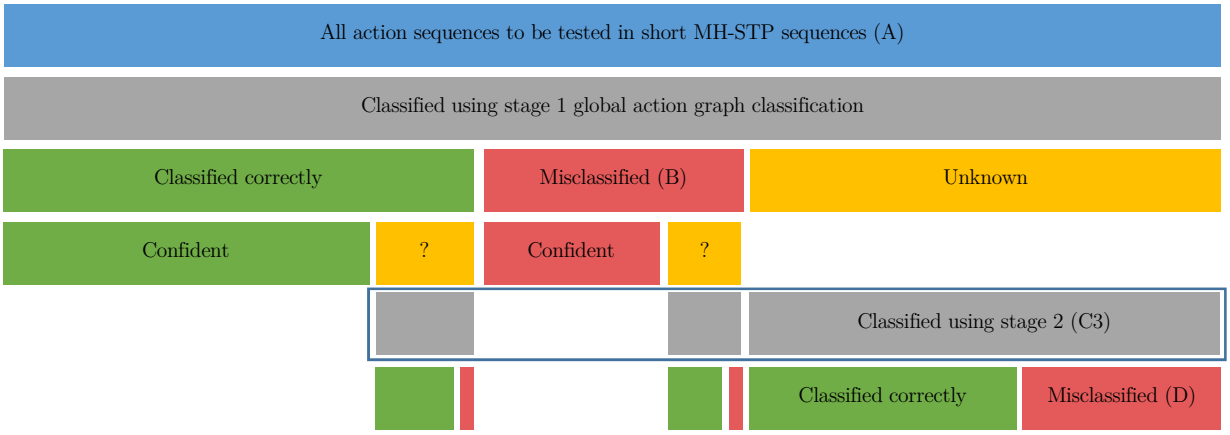

Figure 55 Schematic overview that shows how to interpret the results of the two stage classifier that employs a reclassification possibility.

The results shown in Table 17 show a further improvement in comparison to both mode 2 and mode 3 without reclassification. The confusion matrices show much better results and the effect of the confidence check and reclassification is clearly visible, especially for the actions 11 (point down) and 12 (point up). Drilling down into the reclassification the effect of the confidence check becomes evident. As shown in Table 16 the majority of samples that were reclassified (regardless of their initial classification) are improved: initial classification was wrong, secondary stage classification was ok. Note that some samples that were initially classified well are also reclassified to an incorrect action. This cannot be avoided as it is not known on forehand obviously what actions are classified correctly.

	Amount of test runs	Reclassifications in all tests runs	Result improved	Result deteriorated	Equal result
Cross instance / 33% training	30	4% (716)	41% (294)	4% (30)	55% (392)
Cross instance / 66% training	30	7% (587)	40% (234)	5% (30)	55% (323)
Cross subject / 50% training	50	5% (959)	36% (344)	7% (69)	57% (546)

Table 16 The effect of reclassification (Using a 10x10x10 grid for the short MH-STP, dimensionality 15, 45 salient poses, 2nd stage 10x10x10 dimensionality 40, factor 100 as relative threshold).

	Misclassification in stage 1	Unknown / Low confidence	Misclassification in stage 2	Total classification issues + deltas
Cross instance / 33% training	9%	22%	14%	9% (-5% / -1%)
Cross instance / 66% training	8%	10%	10%	6% (-6% / -3%)
Cross subject / 50% training	10%	18%	18%	10% (-6% / -2%)

	Misclassification in stage 1	Unknown / Low confidence	Misclassification in stage 2	Total classification issues+ deltas
Cross instance / 33% training	5%	19%	8%	6% (-5% / -1%)
Cross instance / 66% training	5%	6%	3%	3% (-2% / -3%)
Cross subject / 50% training	7%	12%	13%	9% (-2% / +1%)

Table 17 Average / Maximum obtained results (10x10x10 grid for the short MH-STP, dimensionality 15, 45 salient poses, 2nd stage 10x10x10 dimensionality 40). Results are given per test and individual parts of the test. The total classification issues (mode2 / mode3) are compared with mode 2 and mode 3 without reclassification. A decrease is shown in green and means it is an improvement.

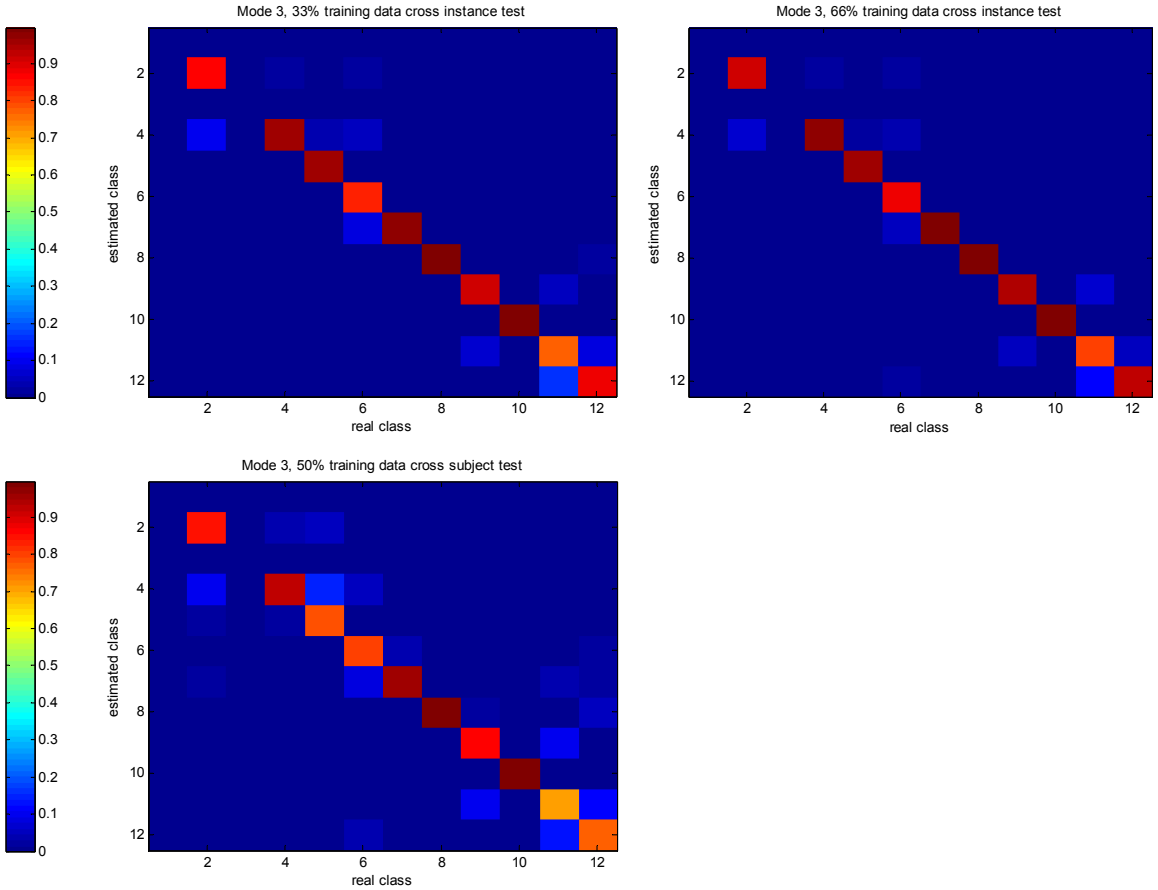

Figure 56 Confusion matrices for all three tests in classification mode 3 with confidence check. Darker red means more actions are classified correctly.

6.4 Comparison of results to the STOP method

The previous Section showed excellent results; especially when using the novel MH-STP representation in the proposed two stage action recognition setup. But how does this compare to the state of the art STOP[27] method if this method is applied on the novel dataset? Using the optimal saturation parameter for this method as shown in Figure 23 as well as a 10x10x10 grid classification is performed. One side note has to be taken into account: although the neutrality (neutral/non neutral) of the short MH-STP can be classified near 100% correctly, it is not tested if this is also the case for STOP features. The following STOP tests however are performed using ground truth neutrality labels so real recognition results of the STOP method on the novel data set in an online setting could be less optimistic. As dimensionality to reduce to a dimensionality of 30 is chosen.

The STOP method is tested using the action recognition framework mode 1 (Action recognition using global salient poses & global transfer matrices) and mode 2 (Action recognition with global salient poses & action specific transfer matrices). The third mode using the two-step classification method cannot be tested unfortunately as the STOP method cannot benefit from the superposition property that the MH-STP has, unless the two step method as used in this report would be adapted. The performance comparison is shown in Table 18.

	STOP Mode 1	STOP Mode 2	MH-STP Mode 1	MH-STP Mode 2	MH-STP Mode 3 confidence checked
Cross instance / 33% training	64%	69%	68%	86%	91%
Cross instance / 66% training	73%	71%	81%	88%	94%
Cross subject / 50% training	44%	57%	72%	84%	90%

Table 18 Recognition rates per recognition method and test method. Higher % means better result.

These results show very clearly that the novel MH-STP representation, especially in combination with the two stage classification plus confidence check performs much better than the STOP method that drew state of the art results on the MSA3D dataset. One possible explanation could be that the reduced dimensionality to 30 could be of an issue for the STOP method, but the results shown in Figure 57 (limited test) shows hardly any possible improvement by choosing a higher dimensionality.

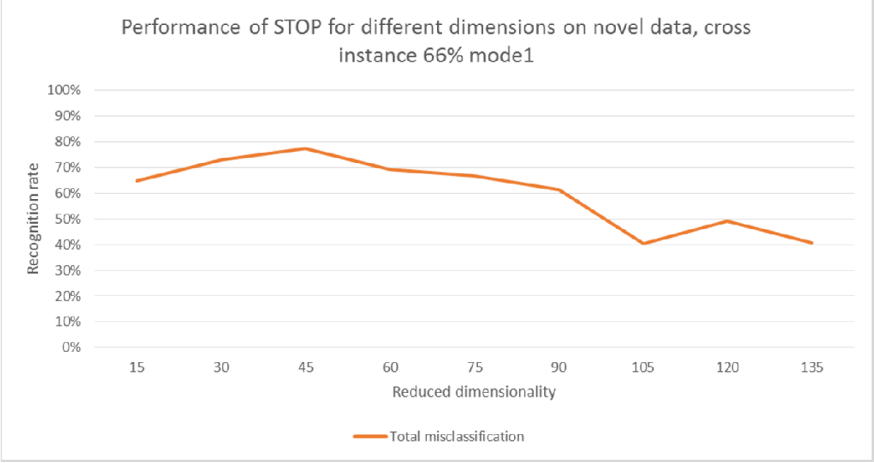

Figure 57 Online STOP recognition rate on the novel dataset in relation to the dimensionality.

One other explanation could be that the STOP method relies more heavily on the amount of samples that is available for training; a higher amount of available training samples might lead to better recognition rates. Looking at the STOP article itself, it can be seen that no results are given for the three different test schemes as used in the offline case (33%/66% cross instance, 50% cross subject). The only result that was reported for the online case is a recognition rate of 98.41%, obtained in a 10-fold cross subject test: every test the samples of one subject are tested while the system is trained using all other subjects. This methodology is mimicked and applied to the novel dataset (but 9 fold, since only 9 subjects are available in the novel dataset) for comparison and results are shown in Table 19. The same as in the previous results table can be observed here: STOP performs considerably less than the MH-STP in the online setting.

	STOP	MH-STP Mode 3 confidence checked
Cross subject, 10 fold	72%	96%

Table 19 Results of a 10 fold cross subject test.

For completeness, a few confusion matrices belonging to the STOP results are shown. It can be seen that the STOP method has large difficulties with some actions. Apart from the actions 11 (point down) and 12 (point up) that also cause the MH-STP a bit of trouble, action 9 (right arm wave) is also a noticeable cause of misclassification, just as action 5 (drink) is very often misclassified while this rarely happens in the MH-STP method as proposed in this thesis.

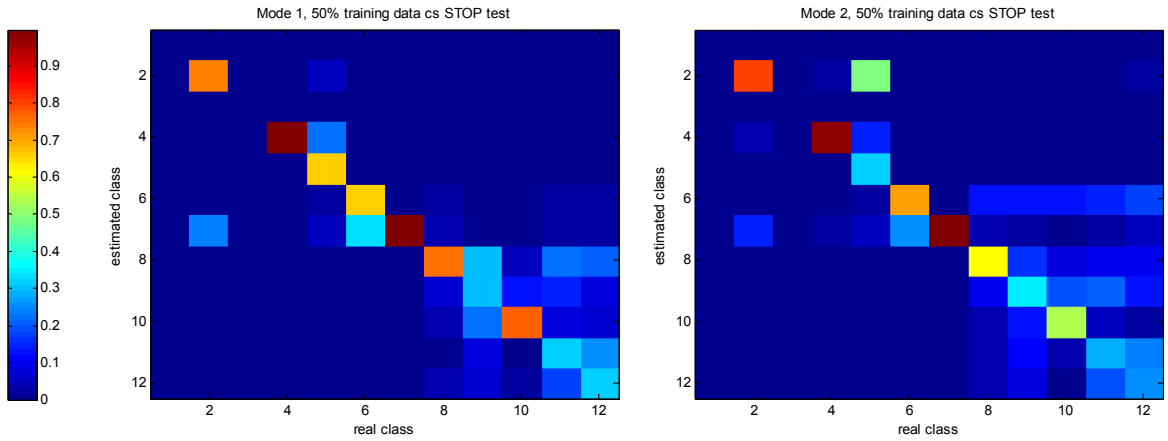

Figure 58 Confusion matrices of the obtained STOP results on the novel dataset. Darker red means more actions are classified correctly.

6.5 Lessons learned about the MH-STP in an online setting

In this chapter the performance of the MH-STP in an online setting was evaluated and compared to the STOP method that reported state of the art recognition rates. In the online case, the recorded actions are not used as a whole to generate representations. Short MH-STP representations are generated instead, each constructed using 5 consecutive frames of one action. One action of 40 frames will therefore lead to the generation of 8 short MH-STP representations. Fewer frames would likely reduce informativeness, more frames would lead to too few representations generated per action. Features are then extracted after which they are classified using a neutral pose classifier. This is a very important step, as the transition from neutral to non-neutral and vice versa marks the start and end of an action and therefore determines greatly the accuracy of the overall online action recognition setup. The sequence of neutral and non-neutral features belonging to one action can then be classified using an action graph.

In the online case, the novel data set was used which is discussed in the previous Chapter extensively. However, although the action classes are known in the dataset, the neutrality labels of the short action pieces was not known. This work describes the difference in performance for two different methods that can be used to obtain these labels. One method is using an automated process and the other is using manually labeled samples. This evaluation required a *manual* labeling task of over 10.000 samples. However, given the importance of the neutral pose classifier it is considered to be worth the effort.

The SVM neutral pose classifier with the short MH-STP as representation of choice showed results as expected in both the automatic as manually labeled case. With the automatic labeling method that marked N frames at the start and end of an action sequence as neutral, reasonable results were obtained. The effect of the value N on the results was determined and this indeed showed that if N was taken small, more samples were marked as non-neutral and vice versa. The cause of this is most likely the mislabeling effect of the automatic labeling process. When varying the grid size from 10x10x10 to 20x20x20 a similar observation can be seen as in earlier tests: results deteriorate. Again, the most likely reason for this is that the increased grid size, although it increases the informativeness, also induces intra action variance.

Luckily far more samples are available, so this effect is less noticeable in the neutrality labeling results then in the offline action recognition results. The biggest influence in the performance of the SVM neutral pose classifier turned out to be the dimensionality reduction. If the dimensionality is reduced the SVM tends to classify too many samples as neutral and when the dimensionality is increased past a certain point the SVM is no longer able to find any neutral actions. Unfortunately, the ideal dimensionality to reduce to turns out to be not identical to the dimensionality as determined before. This however should not lead to any problems, although it might increase required computational times. Finally, manually labeling the neutral poses showed the true potential of the SVM classification using the short MH-STP representation: an excellent accuracy of 98.1% was achieved and the samples that were misclassified were all constructed from pieces of action that contained a transition from neutral to non-neutral or the other way around.

With a functional neutral pose SVM classifier and a fully labeled dataset available, the overall performance of the online action recognition framework was evaluated and the parameters of influence were studied as was done in the offline action recognition case: the grid dimensionality, the dimensionality to reduce to and the amount of salient nodes in the action graph. As it is impossible to predict how many salient poses would be a good value and it was also not known which dimensionality to reduce to would lead to good results, a grid search is performed. For 5 to 75 salient poses and 5 to 40 feature dimensions the results were determined using a 10x10x10 grid size. Best results in a 10 fold test were obtained with 45 salient poses and a MH-STP dimensionality of 15. In only 11% of the cases a misclassification occurred. The grid size was also varied to 20x20x20 as a test. However recognition rates increased only slightly and computational demand increased so much that this was considered to be non-feasible.

Evaluating the results in an online setting on the novel data set was again performed with three tests: 33% and 66% training cross instance and 50% cross subject. Using a 10x10x10 MH-STP grid, dimensionality reduction to 15 dimensions and 45 salient poses. At first, action recognition was performed using global salient poses & global transfer matrices, conveniently called 'mode 1'. Although computational demand in this mode is minimal and only 10% misclassifications were observed, another 18 % of the samples no classification turned out to be possible leading to an overall performance of only 72%. This was caused by transitional probabilities that were zero: a certain path through the global set of salient poses was observed in the testing phase but no action model has been trained for it.

To solve this issue, a secondary EM pass on the conditional and transitional probabilities of all the action graphs of the individual classes is performed (mode 2), while keeping the global set of salient poses fixed. While this leads to an overall performance of 84 %, computational power increased a lot. A secondary, novel solution was evaluated as well which is the third action recognition mode tested in this work; When comparing the additional EM method to the two step classification method s proposed in this work, the latter shows an overall better performance. Although this third online action recognition mode is more memory demanding, computational demand is much lower while at the same time achieving a better results. In the 50% cross subject test a recognition rate of 88% was obtained.

Further improvement turned out to be possible by adding a confidence check in the third mode of operation: not only the samples that cannot be classified in the first stage of classification but also samples that were classified with a low confidence are (re)classified in the second stage. Using this confidence check, of the reclassified samples 40% resulted in a correct classification while being classified incorrectly in the first stage and 7% resulted in a misclassification while being classified correctly in the first stage. This resulted in a total recognition rate of 90% in the 50% cross subject test.

Chapter 7

Conclusions

In the past few years, a very large progression in the field of action recognition has been made. Most influential was the transition from 2D to 3D after the introduction of affordable 3D depth sensors such as Microsoft's Kinect. Because of that, one especially useful representation could be used more reliably: the skeleton representation. From a representation, features can be extracted allowing actions to be classified. Very well performing classification methods currently available are 'action points', 'bag of points', 'Eigen joints' and STOP which is an acronym for spatial-temporal occupancy pattern. The results achieved using STOP on Microsoft's MSA3D dataset are still the state of the art.

STOP inspired the creation of the novel representation as presented in this work: a motion history spatial-temporal pattern or MH-STP. It is a combination of the best aspects of three existing methods: motion history to include additional time information of the movements, a spatial-temporal pattern to preserve space and time information in the representation, and an intermediate skeleton representation to reduce sources of variance such as rotation, size and appearance. Looking back at the research question:

> *Can we create an action recognition system that is capable of recognizing in-home daily actions of persons, independent of their orientation, posture and appearance, without requiring very large sets of training data?*

The exploration of the motion-history spatial-temporal in this work provided a positive answer

> *The MH-STP is able to recognize multiple actions of different subjects, at different locations with a 96% recognition rate. It is invariant to rotation, posture and appearance and it performs well in an online action recognition framework without using very large data sets of training data*

Evaluation was performed using three different test sets: 33% and 66% training cross instance test and a challenging 50% training cross subject test. The MH-STP performance was first evaluated in an initial test to determine the viability of the principle. The dataset used was the well-known Microsoft Research action 3D dataset and initial tests were performed using initially-guessed parameters in an offline setting.

This led to a non-optimized result that was already much better performing (+20%) than one of the more recent methods as proposed by Li et al: Action recognition based on a bag of 3D points. In comparison with the STOP method the *non-optimized* results of the MH-STP method in an offline setting was 84.4% on average which was 5% lower than the reported state of the art. Maximum recognition rates with the novel method were achieved in test 2, with an impressive 94% average accuracy.

This initial test proved sufficient ground to continue the exploration of the MH-STP, allowing to determine optimal offline parameters with respect to both accuracy as computational demand using a novel dataset. Optimal results were obtained when the motion history data was interpolated between successive frames, a grid dimension of 10x10x10 with 3 time segments was used with a dimensionality reduction to 40 dimensions. An average accuracy of 94% is achieved on the novel test set in the challenging 50% cross subject test, exceeding the STOP method by 5% in the offline setting.

In the online situation, special care was taken to research the performance of a neutral pose support vector machine (SVM) when MH-STP was used. After manually labeling 10.000 samples, the SVM classifier was able to correctly classify 98.1% of the samples to neutral or non-neutral. That said, the 1,9% of the misclassifications were even doubtful to be misclassified as these were ambiguous samples even when manually labeling them.

The overall recognition rate of the online classifier was unfortunately a bit lower than in the offline case initially. This was expected, as it is a more challenging task and intra-action induced variance could have taken its toll here. During an initial explorative test, a recognition rate of $> 70\%$ was achieved. However, a large number of samples could not be classified. This was caused by the fact that the online action recognition mode was only globally optimized. The risk of this is that individual action graphs are not trained to deal with all salient pose transitions, which can lead to problems in the during testing phase. To make sure all samples can be classified two methods, two different approaches are tested.

The first approach is the introduction of a secondary EM pass that is performed on the conditional and transitional probabilities of all the individual action graphs corresponding to the different actions. This lead to an improved average recognition rate of 84% but at a high computational price in the training phase. The second approach was the introduction of a classification framework that implemented a novel two-stage classification method. This allowed classification with a secondary classification step in case the first failed or was with low confidence. This resulted in a total recognition rate of 90% in the challenging 50% cross subject test.

To compare the novel MH-STP and action recognition framework with the state of the art, the method as described in the STOP[27] was also tested on the novel data. The results of this STOP method were considerably lower (up to 33% lower) in the cross subject test. Even when a 9 fold cross subject test was performed the STOP method performed much worse (72% recognition rate) than the novel method as proposed in this work that acquired a 96% recognition rate. No good explanation can be found to explain the large gap of performance between the STOP and the novel MH-STP method, especially since the novel MH-STP method draws similar or only slightly lower results in the offline case on the MSA3D dataset. One possibility is that the STOP method is biased to the MSA3D dataset but this is of course to be proven

Concluding, the MH-STP representation can draw very high results and copes very well with many sorts of variance such as rotation, location, size and appearance that can be encountered in the setting of an elderly home. This holds in both offline and online setting due to the combined properties of the skeleton representation, the binning nature of the STP and added time information of the motion history. However, intra-action variance can be introduced easily in case wrong parameters are used. For example, the same binning nature of the STP can cause two similarly performed actions to lead to completely different representations if the grid dimensionality is too high. It is therefore advised to use more training data.

Given the above, the MH-STP can be seen as a valuable addition to existing methods. Hopefully, elderly in the near future can benefit from solutions like the ones proposed in this work, either directly or indirectly, allowing them to live longer in safer environments.

Chapter 8

Future work

In this work the possibilities of the MH-STP were shown and discussed elaborately. Hopefully, this research can be a contribution to the continuous path of research and the development of a household robot or any system that may prove of value to allow elderly to live in a safer environment.

Unfortunately, this work had to be limited to certain boundary conditions. For instance, as in most researches the system that acquires data is limited to one viewpoint only as this simplifies matters a lot. In this work it was limited to three different angles. Reality however might be very different from the situations discussed. For instance, practical situations could require a system to be using observations from any angle and position. The system might even be moving while safeguarding the people it should protect. This is a challenging task as skeleton trackers need to become much more robust, but it is very interesting to implement.

To prevent the introduction of intra-action variance, one possibility would be to not only update MH-STP cells directly that contain the interpolated points of the limbs, but also update cells that are in the vicinity of these interpolated points. This would allow the representations of different samples within one action to 'overlap' more and thereby likely reduce the chance to introduce intra action variance.

Another point of attention is that datasets are often limited to only a few actions. In this work 12 actions were used, but in real life situations there are of course many, many more. This has two challenges. The first challenge is that computational time will increase with every action added to the system as the Viterbi algorithm and probability estimation need to be applied to more and more actions. This might prove to be a problem. Perhaps a system could be thought of to preselect the most likely few paths globally, after which only applicable action graphs are tested.

The second challenge of the limited data sets is that the system can at this point only classify between actions. If offered an observation sequence from a totally new action it will still try to classify the action as a known action rather than determining that the action is new. In [109] a method was shown to detect an unknown action as 'novel'. It showed very good results and it would be well worth investigating if the MH-STP in combination with the used classifier would lead to further improved results in this method.

Finally, in order to create an eventual end product that can help elderly people live longer and safer in their own homes it would be a very good idea to start developing a representative dataset of the target audience. Although their might be privacy concerns it is in the opinion of the author of this work a crucial factor towards the development of a successful system as currently all evaluations are performed on non-representative datasets.

References

[1] N. Yorkin, "L.A. 2013, A day in the life," *Los Angeles Times*, Los Angeles, pp. 6–23, 03-Apr-1988.

[2] M. Hägele, "Market Study on European Service Robotics," *Eur. Robot. Coord. Action euRobotics*, pp. 1–32, 2012.

[3] B. Gates, "A robot in every home.," *Sci. Am.*, vol. 296, no. 1, pp. 58–65, Jan. 2007.

[4] MarketsAndMarkets, *Service Robotics Market (Personal & Professional) – Global Forecast & Assessment by Applications & Geography (2012 – 2017)*. Dallas, Texas, USA: MarketsAndMarkets, 2012.

[5] Newswire, "The Domestic Service Robots Market 2020 Size and Trends Report Published," *newswire*. [Online]. Available: http://www.newswire.com/news/the-domestic-service-robots-market-2020-size-and-trends-report.

[6] R. Poppe, "A survey on vision-based human action recognition," *Image Vis. Comput.*, vol. 28, no. 6, pp. 976–990, Jun. 2010.

[7] P. Turaga, "Machine recognition of human activities: A survey," *Circuits Syst. …*, vol. 18, no. 11, pp. 1473–1488, 2008.

[8] J. K. Aggarwal and M. S. Ryoo, "Human activity analysis," *ACM Comput. Surv.*, vol. 43, no. 3, pp. 1–43, Apr. 2011.

[9] G. J. Burghouts and K. Schutte, "Spatio-temporal layout of human actions for improved bag-of-words action detection," *Pattern Recognit. Lett.*, vol. 34, no. 15, pp. 1861–1869, Nov. 2013.

[10] M. a. Naiel, M. M. Abdelwahab, M. Elsaban, and W. B. Mikhael, "Simultaneous Human detection and action recognition employing 2DPCA-HOG," *2011 IEEE 54th Int. Midwest Symp. Circuits Syst.*, no. 1, pp. 1–4, Aug. 2011.

[11] M. Mubashir, L. Shao, and L. Seed, "A survey on fall detection: Principles and approaches," *Neurocomputing*, vol. 100, pp. 144–152, Jan. 2013.

[12] C. Rougier, E. Auvinet, and J. Rousseau, "Fall detection from depth map video sequences," *Towar. Useful Serv. …*, pp. 121–128, 2011.

[13] Z. Zhang and W. Liu, "A viewpoint-independent statistical method for fall detection," *Pattern Recognit. (ICPR …*, 2012.

[14] V. Bloom, D. Makris, and V. Argyriou, "G3D: A gaming action dataset and real time action recognition evaluation framework," *2012 IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. Work.*, pp. 7–12, Jun. 2012.

[15] L. Shao, J. Han, D. Xu, and J. Shotton, "Computer Vision for RGB-D Sensors: Kinect and Its Applications," vol. 43, no. 5, pp. 1314–1317, 2013.

[16] S. Nowozin and J. Shotton, "Action Points: A Representation for Low-latency Online Human Action Recognition," pp. 1–18, 2012.

[17] Z. Lan and F. De Torre, "Joint Segmentation and Classification of Human Actions in Video," *Science (80-.).*, pp. 3265–3272, 2011.

[18] J. Assfalg, M. Bertini, C. Colombo, and A. Bimbo, "Semantic annotation of sports videos," *Multimedia, IEEE*, pp. 52–60, 2002.

[19] J. Garssen and C. Van Duin, "Grijze druk zal verdubbelen," pp. 14–19, 2007.

[20] C. Van Duin, "Bevolkingsprognose 2008–2050: naar 17,5 miljoen inwoners," pp. 15–22, 2009.

[21] Veiliheid.nl, "Meer ouderen maken fatale val," *NOS.nl*, 2013. [Online]. Available: http://nos.nl/artikel/482768-meer-ouderen-maken-fatale-val.html.

[22] N. V. voor K. Geriatrie, *Richtlijn preventie van valincidenten bij ouderen*, vol. 31, no. 3. 2006.

[23] Annemieke Hoogland, "Valincidenten ouderen deels te voorkomen," *19-09-2011*. [Online]. Available: http://www.gezondheidsnet.nl/medisch/nieuws/6214/valincidenten-ouderen-deels-te-voorkomen.

[24] J. Van Der Leeuw, "Functiewijzer Domotica voor dementiezorg," *Vilans*, 2013. [Online]. Available: www.vilans.nl.

[25] W. Li, Z. Zhang, and Z. Liu, "Action recognition based on a bag of 3d points," *… Vis. Pattern Recognit. …*, pp. 9–14, 2010.

[26] X. Yang and Y. Tian, "Effective 3D action recognition using EigenJoints," *J. Vis. Commun. Image Represent.*, pp. 1–10, Mar. 2013.

[27] A. Vieira, E. Nascimento, and G. Oliveira, "Stop: Space-time occupancy patterns for 3d action recognition from depth map sequences," *… Pattern Recognition, …*, pp. 1–8, 2012.

[28] T. Zhang, S. Liu, C. Xu, and H. Lu, "Boosted multi-class semi-supervised learning for human action recognition," *Pattern Recognit.*, vol. 44, no. 10–11, pp. 2334–2342, Oct. 2011.

[29] E. Ohn-Bar and M. Trivedi, "Joint angles similiarities and HOG2 for action recognition," 2013.

[30] A. Levin, R. Fergus, F. Durand, and W. Freeman, "Image and depth from a conventional camera with a coded aperture," *ACM Trans. ...*, vol. 26, no. 3, p. 70, Jul. 2007.

[31] J. Han, L. Shao, D. Xu, and J. Shotton, "Enhanced Computer Vision with Microsoft Kinect Sensor: A Review," *IEEE Trans. Cybern.*, vol. 43, no. 5, pp. 1318–34, Oct. 2013.

[32] S. Zhang, "Recent progresses on real-time 3D shape measurement using digital fringe projection techniques," *Opt. Lasers Eng.*, vol. 48, no. 2, pp. 149–158, Feb. 2010.

[33] J. Zhu, L. Wang, R. Yang, and J. Davis, "Fusion of time-of-flight depth and stereo for high accuracy depth maps," *Comput. Vis. Pattern ...*, pp. 1–8, Jun. 2008.

[34] J. W. Davis and A. F. Bobick, "The Representation and Recognition of Action Using Temporal Templates," no. 402.

[35] J. A. K. Suykens and J. Vandewalle, "Least Squares Support Vector Machine Classifiers," pp. 1–10, 1999.

[36] U. of Chicago, "Websters Revised Unabridged Dictionary," 2013. .

[37] I. Schopenhauer, A. and Edman, *The Philosophy of Schopenhauer*. 1947.

[38] S. J. Blakemore and J. Decety, "From the perception of action to the understanding of intention.," *Nat. Rev. Neurosci.*, vol. 2, no. 8, pp. 561–7, Aug. 2001.

[39] J. Decety and J. Grèzes, "Neural mechanisms subserving the perception of human actions.," *Trends Cogn. Sci.*, vol. 3, no. 5, pp. 172–178, 1999.

[40] R. Blake and M. Shiffrar, "Perception of human motion.," *Annu. Rev. Psychol.*, vol. 58, no. July 2006, pp. 47–73, Jan. 2007.

[41] G. V. de W. Spelke, Elizabeth S., "Perceiving and reasoning about objects: Insights from infants," *Spat. Represent. Probl. Philos. Psychol.*, pp. 132–161, 1993.

[42] M. Brand, "Understanding manipulation in video," *Proc. Second Int. Conf. Autom. Face Gesture Recognit.*, pp. 94–99, 1996.

[43] A. (MIT M. L. Bobick, "NSF/DARPA Workshop Perception of Action," 1997. [Online]. Available: http://vismod.media.mit.edu/conferences/nsf-action-97/workshop.html.

[44] M. Piccardi, "Background subtraction techniques: a review," *2004 IEEE Int. Conf. Syst. Man Cybern. (IEEE Cat. No.04CH37583)*, pp. 3099–3104, 2004.

[45] P. Power and J. Schoonees, "Understanding background mixture models for foreground segmentation," *Proc. image Vis. Comput. New Zeal.*, no. November, 2002.

[46] Y. Benezeth, P. M. Jodoin, B. Emile, H. Laurent, and C. Rosenberger, "Review and evaluation of commonly-implemented background subtraction algorithms," in *2008 19th International Conference on Pattern Recognition*, 2008, pp. 1–4.

[47] Y. Sheikh, O. Javed, and T. Kanade, "Background subtraction for freely moving cameras," *Comput. Vision, 2009 IEEE*, no. Iccv, 2009.

[48] A. Elqursh and A. Elgammal, "Online moving camera background subtraction," *Comput. Vision–ECCV 2012*, 2012.

[49] E. Hayman and J. Eklundh, "Statistical background subtraction for a mobile observer," *Proc. Ninth IEEE Int. Conf. Comput. Vis.*, no. Iccv, pp. 67–74 vol.1, 2003.

[50] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," *CVPR 2005*, 2005.

[51] S. Avidan, "Fast Human Detection Using a Cascade of Histograms of Oriented Gradients," *2006 IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. - Vol. 2*, vol. 2, pp. 1491–1498, 2006.

[52] P. Viola and M. Jones, "Robust Real-time Object Detection," 2001.

[53] J. Shotton, T. Sharp, and A. Kipman, "Real-time human pose recognition in parts from single depth images," *Microsoft Res.*, vol. 2, no. 3, pp. 1297–1304, 2013.

[54] J. J. Gibson, "The Perception of the Visual World," 1950.

[55] B. Horn and B. Schunck, "Determining optical flow," *Artif. Intell.*, 1981.

[56] J. Barron, D. Fleet, and S. Beauchemin, "Performance of optical flow techniques," *Int. J. Comput. ...*, 1994.

[57] S. S. Beauchemin and J. L. Barron, "The Computation of Optical Flow," vol. 27, no. 3, 1995.

[58] W.-T. Li, H.-S. Chang, K.-C. Lien, H.-T. Chang, and Y.-C. F. Wang, "Exploring visual and motion saliency for automatic video object extraction.," *IEEE Trans. Image Process.*, vol. 22, no. 7, pp. 2600–10, Jul. 2013.

[59] G. Johansson, "Visual perception of biological motion and a model for its analysis," *Percept. Psychophys.*, vol. 14, pp. 201–211, 1973.

[60] E. Yu and J. K. Aggarwal, "Human action recognition with extremities as semantic posture representation," *2009 IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. Work.*, pp. 1–8, Jun. 2009.

[61] H. Fujiyoshi and A. Lipton, "Real-time human motion analysis by image skeletonization," *… Comput. Vision, 1998. WACV'98. …*, 1998.

[62] S. Obdrzálek, G. Kurillo, F. Ofli, R. Bajcsy, E. Seto, H. Jimison, and M. Pavel, "Accuracy and robustness of Kinect pose estimation in the context of coaching of elderly population.," *Conf. Proc. IEEE Eng. Med. Biol. Soc.*, vol. 2012, pp. 1188–93, Jan. 2012.

[63] R. Poppe, "Discriminative vision-based recovery and recognition of human motion," 2009.

[64] W. Li, Z. Zhang, Z. Liu, and S. Member, "Expandable Data-Driven Graphical Modeling of Human Actions Based on Salient Postures," vol. 18, no. 11, pp. 1499–1510, 2008.

[65] S. Sempena, N. U. Maulidevi, and P. R. Aryan, "Human action recognition using Dynamic Time Warping," *Proc. 2011 Int. Conf. Electr. Eng. Informatics*, vol. 4814, pp. 1–5, 2011.

[66] P. Dollár and V. Rabaud, "Behavior recognition via sparse spatio-temporal features," *Vis. Surveill. …*, 2005.

[67] S. Savarese, J. Winn, and a. Criminisi, "Discriminative Object Class Models of Appearance and Shape by Correlatons," *2006 IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. - Vol. 2*, vol. 2, pp. 2033–2040.

[68] J. Niebles and L. Fei-Fei, "A hierarchical model of shape and appearance for human action classification," *… , 2007. CVPR'07. IEEE Conf.*, 2007.

[69] T. T. Thanh, F. Chen, K. Kotani, and H.-B. Le, "Extraction of Discriminative Patterns from Skeleton Sequences for Human Action Recognition," in *2012 IEEE RIVF International Conference on Computing Communication Technologies Research Innovation and Vision for the Future*, 2012, pp. 1–6.

[70] M. Barnachon, S. Bouakaz, B. Boufama, and E. Guillou, "Ongoing human action recognition with motion capture," *Pattern Recognit.*, pp. 1–10, Jun. 2013.

[71] M. Ye, R. Yang, and M. Pollefeys, "Accurate 3D pose estimation from a single depth image," *2011 Int. Conf. Comput. Vis.*, pp. 731–738, Nov. 2011.

[72] D. Weinland, R. Ronfard, and E. Boyer, "A survey of vision-based methods for action representation, segmentation and recognition," *Comput. Vis. Image …*, 2011.

[73] V. Krüger, D. Kragic, A. Ude, and C. Geib, "The meaning of action: a review on action recognition and mapping," *Adv. Robot.*, pp. 1–36, 2007.

[74] G. Guo and A. Lai, "A survey on still image based human action recognition," *Pattern Recognit.*, vol. 47, no. 10, pp. 3343–3361, 2014.

[75] A. Laurentini, "The visual hull concept for silhouette-based image understanding," *Pattern Anal. Mach. Intell. IEEE …*, 1994.

[76] L. Gorelick, M. Blank, E. Shechtman, M. Irani, and R. Basri, "Actions as space-time shapes.," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 29, no. 12, pp. 2247–53, Dec. 2007.

[77] D. Weinland, R. Ronfard, and E. Boyer, "Free viewpoint action recognition using motion history volumes," *Comput. Vis. Image …*, vol. 3, no. October, pp. 249–257, 2006.

[78] I. Laptev and T. Lindeberg, "Space-time interest points," *Proc. Ninth IEEE Int. Conf. Comput. Vis.*, pp. 432–439 vol.1, 2003.

[79] G. Willems, T. Tuytelaars, and L. Van Gool, "An Efficient Dense and Scale-Invariant Spatio-Temporal," pp. 650–663, 2008.

[80] K. Hatun and P. Duygulu, "Pose sentences: A new representation for action recognition using sequence of pose words," *2008 19th Int. Conf. Pattern Recognit.*, pp. 1–4, 2008.

[81] B. Tahayna and M. Belkhatir, "Human action detection and classification using optimal bag-of-words representation," *… Technol. its …*, pp. 75–80, 2010.

[82] D. M. Gavrila and L. S. Davis, "Towards 3-D model-based tracking and recognition of human movement : a multi-view approach," pp. 3–8, 1995.

[83] A. Yao, J. Gall, G. Fanelli, and L. Van Gool, "Does Human Action Recognition Benefit from Pose Estimation?," *Procedings Br. Mach. Vis. Conf. 2011*, pp. 67.1–67.11, 2011.

[84] a. Agarwal and B. Triggs, "3D human pose from silhouettes by relevance vector regression," *Proc. 2004 IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognition, 2004. CVPR 2004.*, vol. 2, pp. 882–888, 2004.

[85] A. Micilotta, E. Ong, and R. Bowden, "Real-time upper body detection and 3D pose estimation in monoscopic images," *Comput. Vision–ECCV 2006*, 2006.

[86] M. Müller, T. Röder, and M. Clausen, "Efficient content-based retrieval of motion capture data," *ACM Trans. Graph.*, vol. 1, no. 212, pp. 677–685, 2005.

[87] J. Davis, D. Nehab, R. Ramamoorthi, and S. Rusinkiewicz, "Spacetime stereo: a unifying framework for depth from triangulation.," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 27, no. 2, pp. 296–302, Feb. 2005.

[88] U. Dhond and J. Aggarwal, "Structure from stereo-a review," *Syst. Man Cybern. ...*, 1989.

[89] K. Creath and J. Wyant, "Moiré and fringe projection techniques," *Opt. Shop Test.*, 1992.

[90] B. Bartczak and R. Koch, "Dense depth maps from low resolution time-of-flight depth and high resolution color views," *Adv. Vis. Comput.*, 2009.

[91] R. B. Rusu and S. Cousins, "3D is here: Point Cloud Library (PCL)," *2011 IEEE Int. Conf. Robot. Autom.*, pp. 1–4, May 2011.

[92] M. Budiu and J. Shotton, "Parallelizing the training of the Kinect body parts labeling algorithm," *Big Learn. Algorithms, ...*, pp. 1–6, 2011.

[93] J. Taylor and J. Shotton, "The Vitruvian manifold: Inferring dense correspondences for one-shot human pose estimation," *Comput. Vis. ...*, 2012.

[94] R. Girshick, J. Shotton, P. Kohli, A. Criminisi, and A. Fitzgibbon, "Efficient regression of general-activity human poses from depth images," *2011 Int. Conf. Comput. Vis.*, pp. 415–422, Nov. 2011.

[95] L. Xia, C. Chen, and J. Aggarwal, "View invariant human action recognition using histograms of 3D joints," *... Vis. Pattern Recognit. ...*, pp. 20–27, 2012.

[96] I. Theodorakopoulos, K. Dimitris, G. Economou, and S. Fotopoulos, "Pose-based human action recognition via sparse representation in dissimilarity spacelias," *J. Vis. Commun. Image Represent.*, pp. 1–12, Apr. 2013.

[97] S. Lin, C. Shie, and S. Chen, "Human Action Recognition Using Action Trait Code," *Pattern Recognit. (...*, no. Icpr, pp. 3456–3459, 2012.

[98] A. Jalal, S. Lee, J. Kim, and T. Kim, "Human activity recognition via the features of labeled depth body parts," *Impact Anal. Solut. Chronic ...*, pp. 246–249, 2012.

[99] S. Baysal, M. C. Kurt, and P. Duygulu, "Recognizing Human Actions Using Key Poses," *2010 20th Int. Conf. Pattern Recognit.*, no. d, pp. 1727–1730, Aug. 2010.

[100] F. Lv and R. Nevatia, "Single view human action recognition using key pose matching and viterbi path searching," *Comput. Vis. Pattern Recognition, ...*, pp. 1–8, Jun. 2007.

[101] X. Cao, B. Ning, P. Yan, and X. Li, "Putting poses on manifold for action recognition," *2011 IEEE Int. Work. Mach. Learn. Signal Process.*, pp. 1–6, 2011.

[102] D. Weinland, E. Boyer, and R. Ronfard, "Action Recognition from Arbitrary Views using 3D Exemplars," *2007 IEEE 11th Int. Conf. Comput. Vis.*, pp. 1–7, 2007.

[103] A. Chaaraoui, "Evolutionary Joint Selection to Improve Human Action Recognition with RGB-D devices," *Expert Syst. with ...*, 2013.

[104] W. R. Schwartz, A. Kembhavi, D. Harwood, and L. S. Davis, "Human detection using partial least squares analysis," *Comput. Vis. 2009 IEEE 12th Int. Conf.*, no. Iccv, pp. 24–31, 2009.

[105] C. J. C. Burges, "Dimension Reduction: A Guided Tour," *Found. Trends® Mach. Learn.*, vol. 2, no. 4, pp. 275–364, 2009.

[106] L. I. Smith, "A tutorial on Principal Components Analysis Introduction," 2002.

[107] G. L. Oliveira, E. R. Nascimento, A. W. Vieira, and M. F. M. Campos, "Sparse Spatial Coding: A novel approach for efficient and accurate object recognition," *2012 IEEE Int. Conf. Robot. Autom.*, pp. 2592–2598, May 2012.

[108] J. a Stevens, P. S. Corso, E. a Finkelstein, and T. R. Miller, "The costs of fatal and non-fatal falls among older adults.," *Inj. Prev.*, vol. 12, no. 5, pp. 290–5, Oct. 2006.

[109] T. Moerland, "Knowing What You Don ' t Know," 2015.

List of figures

List of Tables