FAKULTÄT FÜR !NFORMATIK
Faculty of Informatics

Event-Driven 3D Vision for Human Activity Analysis in Context of Dance and Fitness Training of Elderly People

DIPLOMARBEIT

zur Erlangung des akademischen Grades

Diplom-Ingenieur

im Rahmen des Studiums

Medieninformatik

eingereicht von

Thomas Hahn
Matrikelnummer 0502377

an der
Fakultät für Informatik der Technischen Universität Wien

Betreuung: Priv.-Doz. Mag. Dr. Hannes Kaufmann
Mitwirkung: Dr. Bernhard Kohn, AIT Austrian Institute of Technology

AUSTRIAN INSTITUTE OF TECHNOLOGY

Wien, 29. August 2011

(Unterschrift Verfasserin) (Unterschrift Betreuung)

FAKULTÄT FÜR !NFORMATIK
Faculty of Informatics

Event-Driven 3D Vision for Human Activity Analysis in Context of Dance and Fitness Training of Elderly People

MASTER'S THESIS

submitted in partial fulfillment of the requirements for the degree of

Diplom-Ingenieur

in

Media Informatics

by

Thomas Hahn
Registration Number 0502377

to the Faculty of Informatics
at the Vienna University of Technology

Advisor: Priv.-Doz. Mag. Dr. Hannes Kaufmann
Assistance: Dr. Bernhard Kohn, AIT Austrian Institute of Technology

Vienna, 29. August 2011 _____ _____
(Signature of Author) (Signature of Advisor)

Erklärung zur Verfassung der Arbeit

Thomas Hahn
Feldgasse 2, 3382 Mauer bei Melk

Hiermit erkläre ich, dass ich diese Arbeit selbständig verfasst habe, dass ich die verwendeten Quellen und Hilfsmittel vollständig angegeben habe und dass ich die Stellen der Arbeit - einschließlich Tabellen, Karten und Abbildungen -, die anderen Werken oder dem Internet im Wortlaut oder dem Sinn nach entnommen sind, auf jeden Fall unter Angabe der Quelle als Entlehnung kenntlich gemacht habe.

(Ort, Datum) (Unterschrift Verfasserin)

Acknowledgement

This thesis was developed within the Master studies Media Informatics at the University of Technology Vienna - at the Institute of Software Technology & Interactive Systems within the Interactive Media Systems Group (E188/2) in cooperation with the AIT, Austrian Institute of Technology. At this point I want to take the chance to thank everybody who took part in the development of this thesis.

First of all I want to thank my advisor Hannes Kaufmann for his assistance, guidance and scientific support whenever I needed something during my thesis. Furthermore I want to thank the AIT Austrian Institute of Technology and especially Bernhard Kohn for giving me the opportunity to take part in this EU project Silvergame [SJSB09] and for his support. Special thanks go also to the Reha Zentrum Lübben and their fitness advisor who provided the choreography for the fitness dance in this thesis.

As this thesis determines the end of my studies I want to use the opportunity to thank people who were a great support during my study period. In the first place special thanks are due to my parents who not only supported me financially but also mentally throughout this whole time, which I am very grateful for. Sincere thanks go also to my wife who even gave me the impulse to start this field of study and who was always a motivation to finish it. In addition I would also like to express my gratitude to all my colleagues who accompanied me through my time in Vienna as well as abroad.

Don Ferguson, who works as an english trainee, also deserves special thanks for proof reading my thesis.

This work is supported by the project SilverGame ”aal-2009-2-113” running under the Ambient Assisted Living Joint Programme (AAL JP). The project was co-funded in Austria by the European Commission (EC) and the Federal Ministry of Transport, Innovation and Technology (BMVIT) under Grant number 823510.

Abstract

Over the last years many implementations concerning the recognition of human motion have been developed. In doing so different systems for human motion detection reaching from recognition of simple gestures to more dynamic complex motions have been invented. The application area of these systems is thereby wide spread from input for Human Computer Interaction to human motion analysis in the field of rehabilitation exercises or sports. Systems that are designed for elderly people are becoming more important, especially in the physical training application area. This is because the population is tending to live to an older age and there will be more and more elderly people in the near future.

In this thesis a system for recognition of human motion in the area of dance and fitness training for elderly people is introduced. This module within the EU project Silvergame [SJSB09] is thereby intended to help elderly people to keep their level of health as well as to gain a higher fitness level so that they can stay healthy to an older age. With the system the users can then be encouraged to move more by performing the dance which they see on their home TV screen. In doing so such a dance consists of different human activities which the system recognizes. Furthermore, it also provides some sort of feedback via the given output device. As the input device, a novel event-driven 3D vision sensor, developed at the AIT Austrian Institute of Technology is used in this approach. What is special in this case is that only data is transferred if an intensity change in the field of view is detected. Therefore, less data then with ordinary video systems is generated. Another difference worth mentioning is that this information is communicated not frame-based but pixel wise. Keeping this constraint in mind and based on the information transferred from this sensor, elementary features that are used as input for classification are obtained. Through a detailed research of the literature about the up-to-date classification methods, the most promising technique and features for the motion detection system were chosen. This thesis thereby shows the performance of the designed application and points out the opportunity for further employments. Though it was significant how the chosen classification method can be used for the obtained features from the received data. Additionally first performance measurements were done. For this first implementation MATLAB was chosen as the main platform and further applications shall be based on this gained knowledge.

For experimentation with the implemented algorithm a database including 580 samples with 8 different activities from 15 individuals, using the 3D sensor, was recorded. To obtain representative experimentation results a cross validation was applied and different settings were used to compare the results. Additionally, test sessions were done on different data sets and for the best results the training and evaluation time was recorded to point out the possibility of real-time usage. The best results thereby reached an average correct recognition rate of around 96%.

Kurzfassung

Im Bereich der menschlichen Gesten- beziehungsweise Bewegungserkennung wurden in den letzten Jahren einige wertvolle Ansätze entwickelt, wobei die Komplexität dieser Systeme von der Erkennung von einfachen Gesten bis zu komplexeren dynamischen Bewegungen, im Bereich der Computerinteraktion oder Rehabilitationsübungen reichen kann. Dabei werden vor allem Systeme die zum physischen Training für ältere Personen dienen immer interessanter, da die Bevölkerung immer älter wird und daher es in nicht allzu ferner Zukunft viel mehr ältere Personen, als heutzutage, geben wird.

Diese Diplomarbeit beschäftigt sich nun mit der Entwicklung von Algorithmen zur Erkennung von Bewegungen von Tanz- und Fitnessübungen für ältere Personen. Dabei soll das Modul im EU Projekt Silvergame [SJSB09] vor allem dafür sorgen dass diese Personen ihre Fitness beziehungsweise Beweglichkeit auch im höheren Alter halten oder sogar verbessern. Um dies erreichen zu können, werden die Nutzer dazu animiert, vor ihrem TV Gerät gewisse Figuren und ganze Tänze nach zu tanzen, und Feedback über die Ausführung gegeben werden. Als Eingabegerät für das System soll dabei ein neuartiger event basierter 3D Sensor, welcher am AIT, Austrian Institute of Technology, entwickelt wurde, dienen. Ein wesentlich Unterschied zu anderen videobasierten Systemen ist, dass der Sensor nur Änderungen anhand von der Helligkeit in dem Sichtfeld aufzeichnet und so weniger Daten übertragen werden. Jedes Pixel ist autark und überträgt über ein Bussystem die Daten asynchron. Aus den Daten werden dann grundlegende Features für die zur Erkennung verwendete Klassifizierungsmethode extrahiert. Mittels einer Literaturrecherche werden die heute verwendeten Methoden zur Bewegungserkennung analysiert und die vielversprechendste Methode ausgewählt. Diese Methode soll danach, in Matlab als Entwicklungsumgebung implementiert werden, und mit Hilfe der Daten und der daraus berechneten Features evaluiert werden. Dabei sollen bei der Evaluierung neben der Berechnung der Erkennungsrate auch erste Laufzeiten analysiert werden.

Für eine Evaluierung wurde ein Beispieltanz, welcher sich in 8 unterschiedliche Aktivitäten gliedert, ausgewählt. Bei Testaufnahmen wurden 580 Beispiele der 8 Aktivitäten von 15 unterschiedlichen Personen mit dem 3D Sensor aufgezeichnet und in einer Datenbank gespeichert. Mittels verschiedener Parameter wurde danach eine Kreuzvalidierung mit den implementierten Algorithmen durchgeführt. Dabei erreichten die besten Ergebnisse eine durchschnittliche Erkennungsrate von ungefähr 96%.

Contents

List of Illustrations

List of Tables

CHAPTER 1

Introduction

1.1 Motivation

It is obvious that especially for elderly people it is important that they make some regular excises to keep themselves healthy and vital. Especially since today people are living longer. Sooner or later there will be more and more elderly people. Because of this it is very important to focus on implementations which can provide this target group with facilities so they can stay healthy. Some elderly people do not have the opportunity to go to private trainers. Sometimes this is because they are badly located, so that they can't meet regularly for fitness lessons with others of their age. To stay fit some also want to do a more exhausting training then just going for a walk. These people motivate themselves and train alone or they can take advantage of state-of-the-art techniques. Therefore, the EU project Silvergame [SJSB09] is designed to offer a modern approach for a game-based fitness/dance training application. By using this system, people have more fun and can get a less expensive home workout.

With this platform, which is easily installed in any home, even more people can be connected with one another. This platform provides users with the opportunity to train/dance on their own. Most importantly this system gives a certain feedback about the dances which they are performing, thereby providing additional information. For the detection of the certain activities and providing some sort of feedback, a motion recognition algorithm must be developed. In the area of motion detection it is a challenging task to detect the motion without any markers on the users. The aim of this thesis is to develop algorithms for a vision based sensor in which no markers are used. There were several popular approaches published in past years and some new sensors have been developed to manage a good detection without markers. One of these new sensors is the 3D ATIS sensor developed from the AIT, Austrian Institute of Technology. In the field of motion detection no algorithms for the usage of this data have been developed until now. The motivation now is to develop dance and fitness classification algorithms especially for the sensor's data so that they can be used in the system. Special focus is paid to the selection of the most promising classification method and to test how it can be employed in a sufficient way. As the algorithms are intended to operate on a DSP integrated in the system first measurements

of computational costs are also performed. The results established from this motion recognition system can also be used in fields of motion recognition other than dance and fitness training.

1.2 Problem statement

For motion detection in general there are many different sensors and classification methods that are used today. The main challenge is to detect certain activities without the usage of additional markers mounted on the designated subject. Typically such sensors for this application area are vision based where detection is done with visual data on a frame base. So called time of flight cameras have also been used. Microsoft recently introduced Kinect and brought a new way of marker less detection to the market. In this thesis the motion detection is based on a novel event-driven 3D vision sensor, developed from the AIT, Austrian Institute of Technology [Pos11]. One important part of this work is to select a classification technique and to point out how it can be applied to the data. In doing so the field of application is the detection of dance and fitness training of elderly people.

In general this novel sensor is based on the context of human perception and transmits data only when a change of intensity at a pixel occurs. In doing so it immediately transfers information for each changed pixel, a so-called Address-Event (AE), in contrast to frame based sensors. For the development of motion detection algorithms it is now important to keep in mind that only these 'activated' pixels are transferred. The requirement is thereby to obtain applicable features based on this pixel information. Furthermore, a basic design of this recognition system and algorithms for live detection shall be composed in MATLAB, so that it is adaptable to other programming languages.

As the main algorithm has to be integrated on an embedded system, special focus must be paid to saving computational power. Therefore, a good detection system with a high recognition rate and less computational expenses shall be found. Besides the performance of the classification method, the selection of inexpensive features is also an important fact for the overall performance. Though in order to guarantee good appliance in the platform, attention must be also paid to achieving a high recognition rate with these features.

1.3 Aim of the work

A principal point of this work is to obtain first results for good detection algorithms for motion or rather fitness and dance training analysis. The aim of this recognition system is then the future integration into the module of the EU project Silvergame [SJSB09]. Therefore, full integration into the multimedia-application management-platform is assumed. Besides a good recognition rate of the activities, computational inexpensive algorithms shall be used for the implementation. In doing so literature research is intended to point out the methods that are used with respect to the two mentioned points in the field of human motion detection. Within the implementation the recognition system is also intended to give some feedback about the performed activity. First results should show how the recognition system could be used to give such feedback.

For the motion recognition system a good range of features shall be acquired. In doing so good detection results for the trained activities shall be reached. Thereby a special focus

is paid to the performance and to take advantage of the used AE Data to save computational power. With this first implementation a basic framework in MATLAB that can be adapted to any other programming language shall be supplied. It is also intended to show first results about the correctness and the performance of the system. Additionally, it shall demonstrate the practicality with the sensor and how it can be used in the case of dance and fitness recognition. Furthermore, it shall provide a basic approach for a future extension with several more activities and an architecture, which shall be implemented, on an embedded system.

1.4 Methodological approach

After a literature research about the different methods which have been used for motion recognition systems so far, the most promising method shall be used for a first implementation. Many different methods are thereby used for several motions or gestures. The right technique for this approach shall be found to classify motion sequences as correctly as possible.

In this approach, dance and fitness training consists of sequences of different shorter motion activities which are combined into one dance. Therefore, a set of samples containing such activities is recorded and is used for test sessions with the chosen classification technique. From these recorded samples a database with the extracted features is generated which can be used for training and testing the classification method. Features from related motion recognition systems in the literature are thereby used for experimentation. With this database a representative testing method, amongst others a cross validation, shall be used to evaluate the recognition rate. Different scenarios to compare the different features and settings are thereby generated. For the implementation of the classification method, MATLAB is used as the primary programming environment as it allows quick experimentation and already provides many of the methods mentioned in the literature. This implementation therefore has to be ported to another programming language (C++). In addition to the best classification results, the performance times were recorded and showed how the system is suitable for practical use.

1.5 Structure of the work

In the course of time many new methods and sensors in the field of human gesture and motion detection have been developed. In Chapter 2 the main components which are used in state-of-the-art motion recognition systems are pointed out. Focus is thereby paid to different hardware systems, such as vision-based or time of flight cameras. Furthermore, the most common motion classification techniques which are used in the literature are explained. The decision for the chosen sensor and technique is also discussed.

In Chapter 3 a general overview about the system and which classification methods are used can be found. Furthermore, how the fitness training is split into sequences and which features are extracted for the use as the input for the classification method is illustrated.

In addition to the previous sections, Chapter 4 is concerned with a general description of the implemented software. Both a workflow for the software and an explanation of the used implementation in MATLAB are given. Some parts are the acquisition of the data, further processing and feature extraction. After analysing the algorithms with the recorded test data, Chapter 5 presents the experimental results and a comparison to related work as well as open issues are discussed. Chapter 6 then states a summary and the work for the future.

CHAPTER 2

State of the art

2.1 Hardware

In the field of motion and gesture recognition many different hardware systems have been developed so far. Most of them are constrained by the fact that the person who needs to be tracked has to wear markers. With these markers it is then possible to reach very accurate results for human motion recognition. Now the challenging part is to detect such motions without the usage of markers. In the literature, many approaches without the usage of markers can be found but only a few of them reach satisfying recognition rates. For this thesis it is of high importance that the system works without the usage of markers. Therefore state-of-the-art sensors which are used in motion recognition systems are introduced in this section.

2.1.1 Video based

The first approaches within motion detection have been developed with the use of general information from video cameras. These cameras have already been in use for decades and so they were already used in the beginnings of motion detection. In some areas they still are used especially where only video information is available, during live broadcasts for example. Another advantage is the low cost, even for cameras with high resolution. On the other hand, due to constraints on lighting, video systems only deliver moderate results. In general, the concept of video based systems is widely known and many approaches have been developed based on such systems.

Today most of the low priced video cameras have a high-definition resolution of 1920x1080 pixels with a frame rate up to 60Hz. So in this price range good equipment is provided. However, high-speed cameras are still very expensive. Although recording fast movements or sport analysis is still expensive, in the last few years another trend of in the market has been the development of 3D cameras for home usage. Panasonic was one of the first companies that produced a 3D camcorder. More information and an article about this device can be found on the homepage of

CNET[1]. With this camera depth information can be obtained even for private, non-commercial use. The disadvantage is that good 3D systems are still high priced compared to normal video systems. Additionally, the usage of 3D information in the application is not always desirable.

Video based systems	
Price	10< € (July 2011)
Setup	very easy & universal
Dimension	very small (4.4x15mm <)
Rate	very high frame rate (24 to ~1000fps) possible
Resolution	high pixel rate (1920x1080 pixels) available
Working range	~0.3 < m
3D Information	YES (complex computation)

Table 2.1: Properties of video based systems

2.1.2 Time-of-flight

Another technique which is used in the field of motion detection, is the so-called time-of-flight (TOF) camera. With these cameras the absolute distance can be calculated by measuring "...the absolute time that a light pulse needs to travel from a target to the detector" [LS01]. With this knowledge a distance image of the 3D scene can be obtained directly from the camera. That means for practical use that an active light source and a receiver are mounted on the system which is illustrated in Figure 2.1. These two devices have to be located very closely to avoid shadowing effects. A more detailed explanation about the main principle of time-of-flight cameras can be found in the approach of Lange and Seitz [LS01]. As many different devices have been developed until now, two of the most common TOF cameras are described in detail. These cameras are of commercial use and have been developed for many application areas but are today more and more used for motion detection.

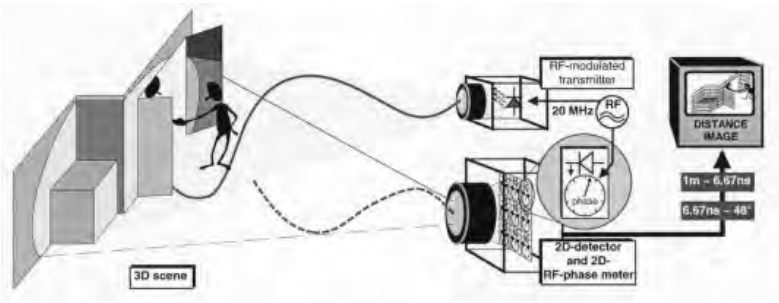


Figure 2.1: Principle of a non scanning 3D TOF camera [LS01]

[1] http://news.cnet.com/8301-30685_3-20015334-264.html

PMD[vision]© CamCube

The PMD[vision]© CamCube includes in its latest version 3.0 a new image sensor and was developed from PMD Technologies. With a resolution of 200x200 pixels the device offers simultaneous capture of grey scale and distance information. Due to the new chip used in this sensor even frame rates (3D) from 40fps @ 200x200 pixels are reached. With the integrated SBI (suppression of background illumination) technology the product can be used indoors and outdoors, which gives it more flexibility is a great advantage. It also has to be noted that with an optic of 12.1mm the sensor works from a range of 0.3 to 7m, by which a notable angle of view can be managed. For extension of the field of view even more sensors can be connected by using different frequency channels. [PMD11]

Another key fact is that the sensor has a very compact design which only measures around 60x67x60mm. Even with the whole sensor including the two illumination units, which are mounted on the side, the design remains very compact. The basic configuration including the camera unit as well as the illumination unit (one at each side of the camera) is shown in Figure 2.2. Additionally, an example of a colour coded 3D image and a 3D & grey scale image taken with the CamCube is illustrated in Figure 2.2.

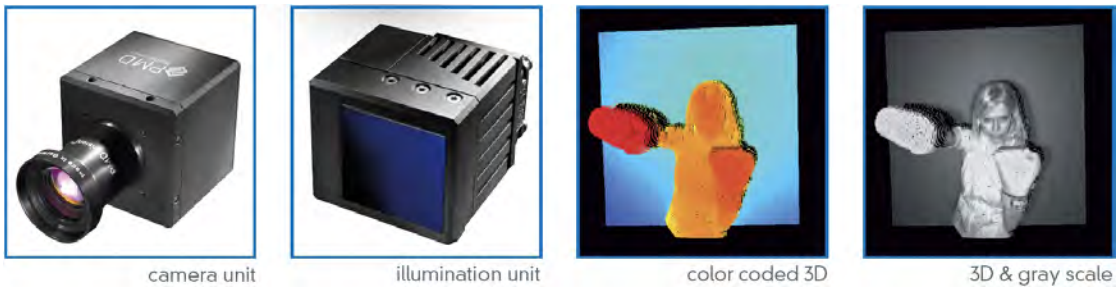


Figure 2.2: PMD[vision]CamCube units and example images [PMD11]

The primary application areas [PMD11] the device was designed for are:

- Games and consumer electronics
- Media and advertising
- Mobile robotics
- Factory automation
- Security & surveillance
- Automotive industry
- Medical technology and life sciences

With all of these features the CamCube becomes competitive in many different areas. Much more additional information like demo video or other older sensor versions can be found on the home-page of the PMD Technologies [2] itself.

PMD[vision]© CamCube	
Price	~6490 € (July 2011)
Setup	easy & multi sensor usage possible
Dimension	compact (60x67x60mm)
Rate	high frame rate (40fps) possible
Resolution	average pixel rate (~200x200 pixels)
Working range	0.3 to 7m
3D Information	YES (directly from the sensor)

Table 2.2: Properties of the PMD[vision]© CamCube

SwissRanger^{TM} SR4000

The second device worth mentioning is the SwissRanger^{TM} SR4000, invented from by MESA Imaging it provides high-resolution 3D image data in real time. It also works with the time of flight principle and the general structure looks similar to the product mentioned before. This sensor reaches a resolution of 176x144 pixels and a maximum frame rate of 50fps. To handle unpropitious background light conditions it uses a method for background light suppression like the CamCube in the previous illustration. Additionally, two measurement ranges, up to 5m and up to 10m, are provided to allow more flexibility and more measurement accuracy. It is also independent of object colour and reflectivity. For use with different applications two different fields of views are also available. So it is possible to use a wide field of view camera with 69° (h) x 56° (v) instead of a standard field of view camera with 43° (h) x 34° (v) for indoor usage. [Ima11]

Figure 2.3: SwissRanger^{TM} SR4000 [Ima11]

[2] http://www.pmdtec.com/

As a matter of fact, by using different illumination frequencies, the SwissRanger^{TM} SR4000 can also be used simultaneously with up to 3 cameras. As the LEDs and the camera are packed into a single case, one of the differences to the CamCube is the even more compact design. The dimension of the whole sensor is only 65x65x68mm. The sensor was mainly designed for indoor use such as in [Ima11]

- Logistics
- Surveillance & Security
- Machine Vision & Robotics
- Medical & Biometrics.

and therefore has a more specific area of application. More additional information can be found on the homepage of the MESA Imaging [3] itself.

SwissRanger^{TM} SR4000	
Price	~6000 € (July 2011)
Setup	easy & multi sensor usage possible
Dimension	compact (65x65x68mm)
Rate	high frame rate (50fps)
Resolution	average pixel rate (~176x144 pixels)
Working range	5 to 10m
3D Information	YES (directly from the sensor)

Table 2.3: Properties of the SwissRanger^{TM} SR4000

2.1.3 Kinect

A sensor which is specially designed for the gaming area is called Kinect. It was developed by Microsoft to be used with the Xbox 360 as a control input device for games. An important fact is that it works without the usage of markers. Since there are not many sensors, which can handle gesture recognition without markers, it was also adopted for other fields. Many projects intended to use the sensor on other platforms other than the Xbox and therefore several approaches have been developed so far.

In Figure 2.4 the general design of Kinect is shown. As illustrated in the image, the bar consists of two sensors, which are responsible for the 3D depth image calculation, a RGB camera and a so-called 3D audio microphone. On the left the IR light source is fixed which projects the scene by invisible IR light as a point matrix. On the right a standard CMOS Sensor handles the IR information and obtains data for the calculation of the scene depth image. In the centre the normal RGB camera can be found which generates additional information such as face detection. In Figure 2.5 the general structure of the platform is explained.

[3] `http://www.mesa-imaging.ch`

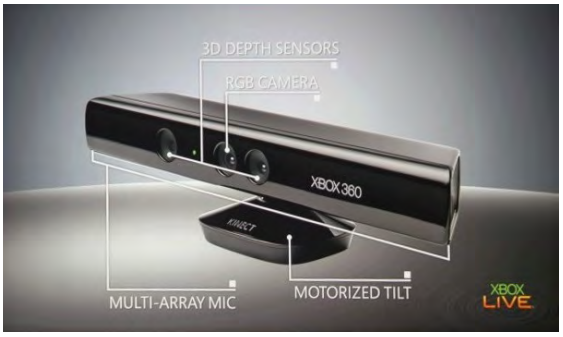


Figure 2.4: The Kinect sensor bar. Source: `http://www.hardware-infos.com/img/startseite/kinecttechnologiese3.jpg`

Electronic Theatre [The10] states that the depth sensor works with a resolution of 320x240 and the RGB camera provides a resolution of 640x480. It has been stated in the literature that both sensors are working at 30 frames per second and the depth sensor is supposed to work in the range of 1.2m to 3.5m. iFixit [iFi10] mentioned that the accuracy of the sensor can measure within 1 cm at two meters but on the other hand they stated that in real the world "...you don't get anywhere near that accuracy..." [iFi10]. Worth mentioning is also that it works with up to 2 active players and can track up to 6 people with their provided software so far.

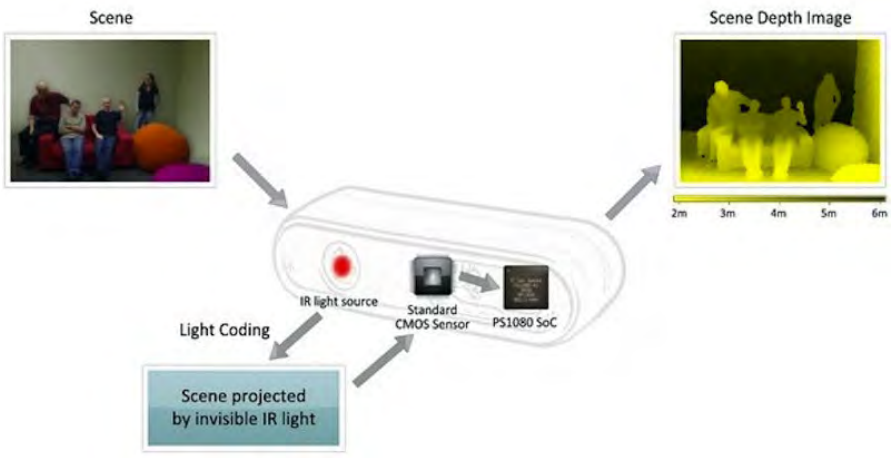


Figure 2.5: Illustration of calculation of a scene depth image with the Kinect. Source: `http://www.primesense.com/images/siteCont/Content_70.7.jpg`

Basically Microsoft [Mic11] states that the sensor bar is used for detection of the whole body and to recognize certain movements i.e. navigation throw menus. Additionally, face recognition for detecting different players is done and with the audio microphone speech recognition is provided. It has to be noted that Kinect is only available for the consumer market and no versions for industrial usage are planed up to now.

Kinect	
Price	~100 € (July 2011)
Setup	easy
Dimension	compact (~300x63x38mm)
Rate	high frame rate (30fps)
Resolution	average pixel rate (~320x240 pixels)
Working range	1.2 to 3.5m
3D Information	YES (calculated from CMOS sensor information)

Table 2.4: Properties of the Kinect sensor bar

2.1.4 ATIS - biomimetic, frame-free vision sensor

A sensor that works on a frame-free vision basis was developed at the AIT, Austrian Institute of Technology. The newest generation of the sensor is the so-called ATIS (Asynchronous, Time-based, Image Sensor) which is a visual sensor that combines multiple bio-inspired approaches. In doing so the technique is different from other conventional image sensors as it works not on a frame basis but on single pixel information [Pos11].

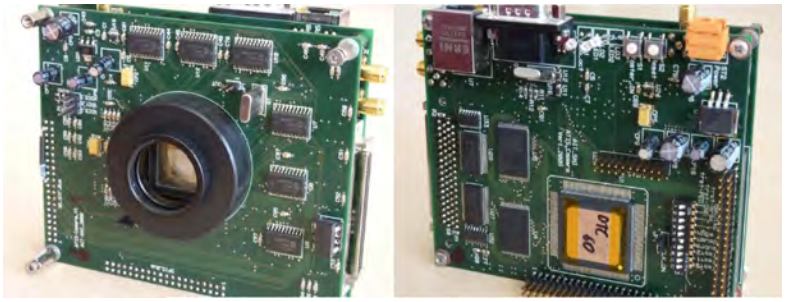


Figure 2.6: ATIS camera system [PMW+10]

With the invention of the new ATIS sensor a higher resolution is supported and it now works with a QVGA pixel array size of 304x240. The "...sensor is based on an array of fully autonomous pixels that combine a change detector and a conditional exposure measurement device" [PMW+10]. That means that a pixel is activated immediately after a change of brightness in the field-of-view takes place. In doing so less data compared to normal image sensors is transferred and it reaches a temporal resolution from 50kfps(@ 1000lux).

In addition, the sensor works in different illumination conditions to support a wide dynamic range so that it can be used inside as well as outside. As it is pointed out in Figure 2.7, even at a lighting of 10 lux, which corresponds to street lighting, a temporal resolution of 500fps can be reached. Moreover, the sensor can be used from 2 lux up to more than 100 klux [LPD08]. In Figure 2.6 the general ATIS camera system design is shown.

TABLE I. Summary table of camera specifications.

Sensor technology	0.18µm CMOS
Sensor size	$9.9 \times 8.2mm^2$
Optical format	2/3“
Pixel array	QVGA (304 × 240)
Pixel size	30µm × 30µm
SNR (typ.)	>56dB (9.3bit) @ >10Lx
Temporal resolution EM	500 fps equ. (@ 10Lx), 50kfps equ. (@ 1000Lx)
Temporal resolution CD	100kfps equ. (@ > 100Lx)
DR (static)	143dB
DR (30fps equivalent)	125dB
FPN	<0.25% @ 10Lx (with TCDS)
Readout formats	- raw timed address-events (TAE): array address, time-stamp, CD/EM bit, polarity/threshold bit - grayscale events: array address, gray-level

Figure 2.7: ATIS sensor summary [PMW+10]

In addition this optical sensor can also be used for event-driven 3D stereo vision and therefore can be used in many application fields [oT11] such as

- Fast moving object detection and classification
- High-speed measurement tasks in industrial automation
- Robotics and autonomous moving systems
- Micromanipulation and robot-assisted surgery
- Low-data rate video for mobile systems
- HDR imaging and video for scientific tasks

On the homepage of the AIT[4] some demo videos about event-driven scene generation can be found where the wide dynamic range and the high temporal resolution is demonstrated in practice. An additional fact worth mentioning is that the system has an integrated DSP for direct implementation of algorithms without the use of any additional hardware.

[4] http://www.ait.ac.at/research-services/research-services-safety-security/new-sensor-technologies/chip-design-for-intelligent-optical-sensor-chips/atis-biomimetic-frame-free-vision-sensor/?L=1

ATIS	
Price	~1500 € (July 2011)
Setup	easy
Dimension	compact (~300x150x38mm)
Rate	ultra high frame rate (500fps to 50kfps) possible
Resolution	average pixel rate (~304x240 pixels)
Working range	0.5 to 6 m
3D Information	YES (directly from the sensor)

Table 2.5: Properties of ATIS stereo system

2.2 Hardware discussion

Overall four different hardware types and techniques which are used today for human action recognition were described in the previous section. The oldest method, the video based devices, are restricted to certain illumination conditions. Also obtaining stereo information involves a lot of computational power. On the other hand, it is still the least expensive and most widely used technique and in some areas it still enjoys great popularity.

A more expensive but more accurate method is then the time-of-flight sensors whereby two well-established examples were introduced. As the detection is very accurate the TOF cameras are also used for human action recognition. Especially advantageous is that no calculation for the depth must be done as the camera "...directly delivers the depth values for each pixel" [Ima11]. On the other hand, the disadvantage is the relatively high price of such cameras, which is why they do not enjoy much popularity in the consumer market.

A rather new device is the Kinect from Microsoft. The advantage which it provides is an accurate detection for many different motions at a low price. As the setup is also easy it can be employed in many applications. The disadvantage is that the detection also needs a lot computational power with additional hardware and the working range is limited to maximum three meters. This is mainly because the sensor is designed for gaming on the XBOX 360 in homes where not so much space is needed. Another fact is that up to now Kinect is only licensed for consumer usage with the XBOX 360.

The last sensor is the so-called ATIS sensor from the AIT, Austrian Institute of Technology. This event-driven system needs less computational power then other sensors as the main calculation is already done from the sensor itself. The main advantages are the low data rates, the high scan rate and the wide dynamic range. Another point to mention is that the sensor works at long distances as well as for close shots. Compared to the high priced TOF cameras, the ATIS is in a medium range. On the other hand, a DSP is already integrated so that motion recognition algorithms can be implemented in the system without the use of additional hardware.

In choosing the most promising sensor a good balance between computational power and good accuracy are two of the main points. ATIS satisfies achieves both requirements. Therefore, this sensor is used for the activity recognition implementation. Especially because motion detection is realized in this project, movements are easily detected from the sensor. Therefore, in

comparison to other sensors, with ATIS no further calculations need to be done. Furthermore, no additional hardware for the implementation of the algorithms is needed as it can be implemented directly on the integrated DSP unit. Also no algorithms for this event-based sensor have been invented so far. Therefore the usage within the field of motion recognition systems shall be analysed.

	Pros	**Cons**
Video based	cheap&widely-used easy setup high resolution possible low price (~50 €)	light constrains no depth information not very accurate for fast movement (high data rate)
Time of flight	independent from color&reflectivity high frame rate easy setup	affected by background light drift with temperature high price (~6000 €)
Kinect	very accurate low price (~100 €) easy setup	small working range only consumer market license for XBOX 360 so far
ATIS	high scan rate possible less data rate wide dynamic range integrated DSP	data only at intensity change medium price (~1500 €)

Table 2.6: Pros and Cons of the different hardware system suitable for motion recognition systems

2.3 Classification techniques

Aside from the decision about which sensor to use, a main point is choosing a method for the classification of the different activities. Since motion and gesture recognition had its beginning in the 1990's [YOI92, MT91], many approaches for different movements have been developed so far. Also another important fact is the evaluation of the features used as an input for the classification technique. This section deals with the different methods and their uses in related approaches. Furthermore, some different features that are used for motion recognition are explained and these methods and features are compared.

2.3.1 Methods

Since the technical background has made it possible to reach better detection results, over the past decades motion and activity recognition have become more and more important. Therefore, many approaches with different methods have been developed.

In general it can be said that not every method is suitable for every type of motion. Some methods are typically for static gesture (pose) recognition and cannot solve dynamic gesture

recognition problems in the same way as other methods. A dynamic gesture is assumed to be a gesture with pre stroke, stroke and post stroke phases compared to a static gesture in which a certain pose or configuration is expected. [MA07]

Neural Networks

Neural Networks (NN) were first used for gesture recognition in the early 90's [MT91] and were later adapted to other appliances. Basically NNs became more important in the area of gesture recognition or in static gesture recognition tasks [MA07]. The main appliance for detecting human motion with Neural Networks can be seen in the recognition of sign language such as in [MT91]. Some other fields for NNs are sales forecasts, industrial process control, customer research and so on [SS].

In general Artificial Neural Networks (ANNs) are based on the functional principle of the human brain, a biological nervous system. Neurons are processing information to solve certain problems. Through a learning process, ANNs are trained for specific applications such as pattern classification. A simple neuron, with multiple inputs and just one output can be seen in Figure 2.8. A special firing rule is then used to handle input patterns and to decide if the neuron is activated or not. [SS]

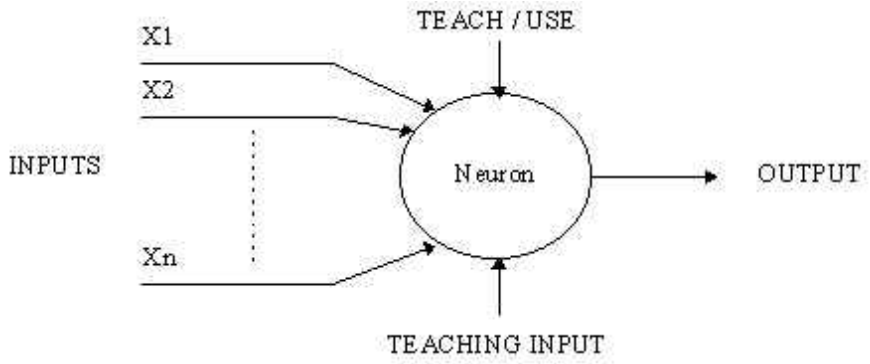


Figure 2.8: A simple neuron. Source: `http://www.doc.ic.ac.uk/~nd/surprise_96/journal/vol4/cs11/report.simple_neuron.jpg`

As for this application dynamic gestures in particular fitness activities shall be detected, Neural Networks are not the first choice and are not explained in detail here. A report by Christos Stergiou and Dimitrios Siganos [SS] and a detailed description of Neural Networks can be found online on the homepage about Neural Networks [5].

[5] `http://www.doc.ic.ac.uk/~nd/surprise_96/journal/vol4/cs11/report.html`

Support Vector Machines

Aside from Neural Networks, Support Vector Machines were also recently adopted for use in human motion recognition and were first introduced by Cortes and Vapnik [CV95] in the field of machine learning techniques. [SLC04, MPB07] As Burges [Bur98] states, SVMs are used in the case of pattern recognition for handwritten digit recognition, object recognition, face detection in images and so forth.

The basic idea behind Support Vector Machines is to find an optimal hyper plane which separates two classes with the largest margin. Thereby a differentiation between what is linear separable data and what is not linear separable data has to be done. In Figure 2.9 the optimal hyper plane and the optimal margin for the linear separable case are shown. In doing so the grey marked boxes represent "...the margin of the largest separation between the two classes." [CV95]. A more detailed explanation about SVMs in general, and their applications can be found in the approach of Cortes and Vapnik [CV95] or on the Homepage about Support Vector Machines [6].

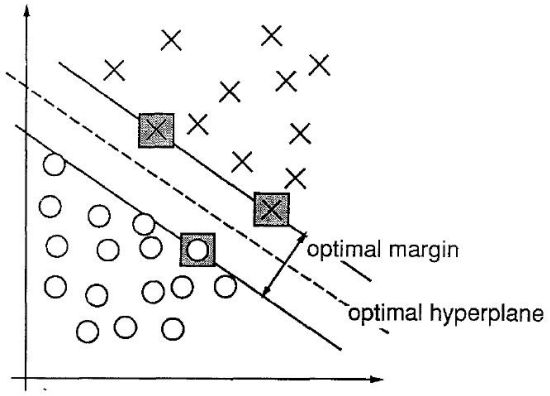


Figure 2.9: A linear separable example [CV95]

In the above mentioned approaches by Schüldt et al. [SLC04] and by Meng et al. [MPB07], two methods for using Support Vector Machines in the field of human motion recognition were introduced. Both methods used the KTH Database which was generated by Schüldt et al. [SLC04] with six human actions, i.e. Walking, Jogging, Running, Boxing, Hand waving and Hand clapping. Thereby different features for the SVM were used.

[6] http://www.support-vector-machines.org/

Hidden Markov Models

Hidden Markov Models have their origin in speech recognition [Rab89] but are also now used more and more for motion recognition. One of the first approaches for human motion recognition was introduced by Yamato et al. [YOI92] and many more have since been presented [AMPdlB07, CRG08, WYSL08, LN06].

As a general definition HMMs are a version of a finite state machine with the following formal parameters: [Sho10]

- Hidden states $Q = q_i, i = 1, ..., N$
- Transition probabilities $A = a_{ij} = P(q_j a_t t + 1 | q_i att)$, where P(a | b) is the conditional probability of a given $b, t = 1, ..., T$ is time, and qi in Q. A is the probability that the next state is q_j given that the current state is q_i.
- Observations (symbols) $O = o_k, k = 1, ..., M$
- Emission probabilities $B = bi_k = b_i(o_k) = P(o_k|q_i)$, where o_k in O. B is the probability that the output is o_k given that the current state is q_i
- Initial state probabilities $\Pi = p_i = P(q_i att = 1)$

To denote a Hidden Markov Model a so-called triple of $\lambda = (A, B, \Pi)$ as a compact notation is used as the states Q and the outputs O are typically self-evident [Sho10]. In Figure 2.10 an example of a HMM is shown with the Hidden states and the output each state produces at the current state.
Three main problems of HMMs are pointed out in the literature and have to be solved in most applications: [DD96]

1. How to compute $P(O|\lambda)$ with the given model $\lambda = (A, B, \Pi)$?
2. How to find the most likely sequence of states for a given output sequence for the model $\lambda = (A, B, \Pi)$ so that $P(O, I|\lambda$ is maximized?
3. How to adapt the parameter of the model $\lambda = (A, B, \Pi)$ that $P(O|\lambda$ is maximized?

To obtain a solution for these problems three algorithms can be applied. To get the probability of occurrence of the observation sequence stated in problem 1 the Forward and Backward algorithms are used. To solve the problem of finding the optimal state sequence the Viterbi algorithm is applied. To find the best matching state transition and output probabilities the Baum-Welch algorithm is used. Detailed information about the employment can be found in [Sho10] and in [DD96].

In the approaches mentioned in the beginning different types of Hidden Markov Models have been applied in the field of motion detection. For example Yamato et al. [YOI92] used HMMs with discrete outputs whereby in contrast Mendoza and de la Blanca [AMPdlB07], Chakraborty and González [CRG08], and Wang et al. [WYSL08] used Gaussian Hidden Markov Models (GHMM) for their classification systems. Thereby discrete values or Gaussian density functions can be used as outputs to provide two different types of Hidden Markov Models.

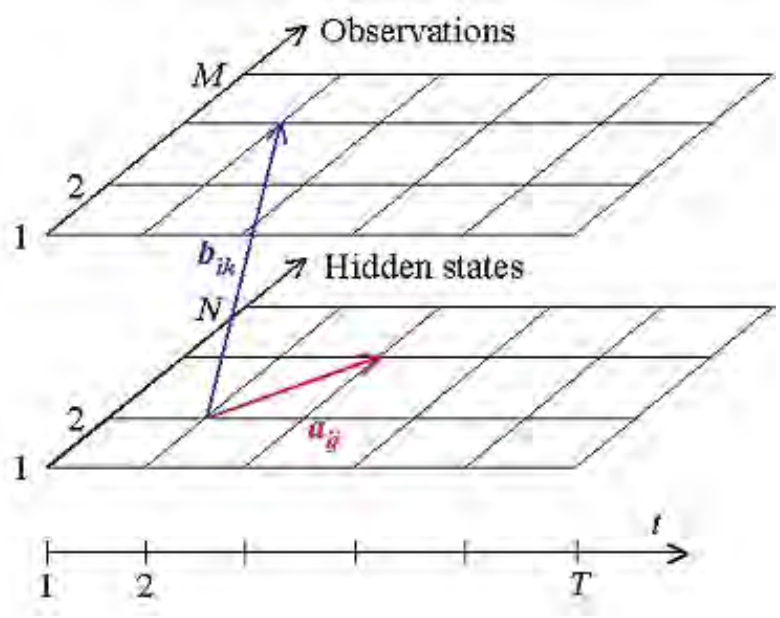


Figure 2.10: HMM (Hidden Markov Model). Source: `http://www.shokhirev.com/nikolai/abc/alg/hmm/images/hmm75.gif`

2.3.2 Features

The selection of the best features for the classification is a very important task. So it is the case that using redundant features causes unnecessary computation power and selection of 'bad' features may lead to false classifications. The task now is to select the right features which describe the content as well as possible. In this paragraph some features used for human motion detection in related approaches which are especially of interest for this approach are described. The collection contains features, which make use of 3D information and some that are using image-processing techniques. It may also be noted that most of the features in the methods are used as input for Hidden Markov Models but can be also be related to other classification methods.

Relative pixelcount

Yamato et al. [YOI92] have used a so-called relative pixel count as the input for a HMM classification in their approach. They have thereby generated binary images from conventional video information and selected a region of interest (ROI) around the human body. This region is then divided into meshes. For each of the meshes the relative count of black pixels is calculated. The feature vector is built with equation 2.1 by the "...ratio of black pixels in each mesh..." [YOI92].

$$f(i,j) \; = \; \frac{number\ of\ black\ pixels(i,j)}{(M_m \; x \; N_m)} \tag{2.1}$$

Where M_m, N_m are the counts (with size m, in this approach m is 8) and i, j are the indexes along the image dimension. The complete mesh feature generation is also illustrated in Figure 2.11. With this equation they generate a feature vector sequence from the time-sequential images. A symbol sequence which serves as the input for the HMMs is obtained by compressing the feature vector with a vector quantization. By taking only this feature they have already reached an average recognition rate of 96% for the detection of 5 different tennis strokes.

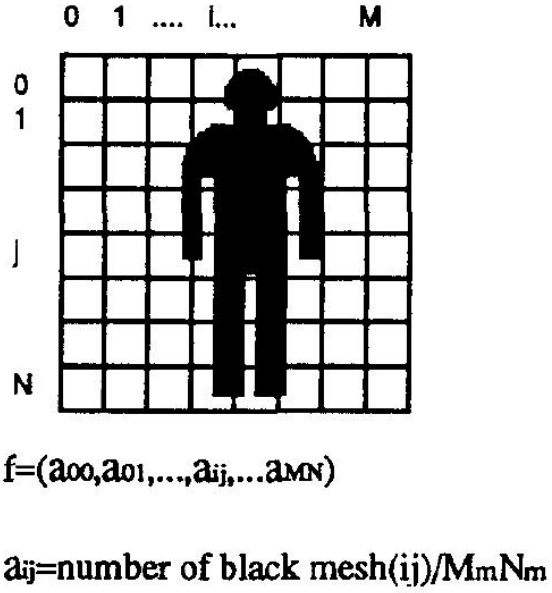


Figure 2.11: Mesh Feature [YOI92]

Shape-context

Another feature which works on a subdivided human body region was introduced by Mendoza and Pérez de la Bianca [AMPdlB07]. They are dividing the calculated human contour region into 8 tiles (2 horizontal, 4 vertical), and calculating the shape context [BMM06] for every tile. Basically they are pointing out that the usage of shape contexts "...is robust to occlusions or bad background segmentation, cluttered environments and shadows, and exploits the contours' good qualities...' [AMPdlB07]. An example of how the division and the contours of the individuals is realized is shown in Figure 2.12.

As the shape context basically calculates the distance and angle between N chosen distance/angle bins in a tile they obtained in their approach a "256 D Vector (8 tiles x 4 distance bins x 8 angle bins)" [AMPdlB07]. Furthermore, to use this feature vector as input for HMMs they are reducing the size of the vector by employing a discrete cosines transformation (DCT). In doing so they are shrinking the dimension to 64 coefficients. Experiments applying the feature vector with a HMM classification indicated that an average recognition rate of 93.11% can be reached.

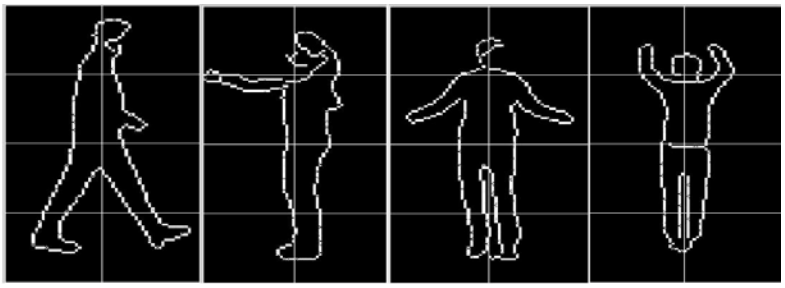


Figure 2.12: Eight independent split region [AMPdlB07]

Gradient

Analogous to the before mentioned feature calculation method, is the calculation of gradients. Chakraborty and González [CRG08] therefore used in their approach a 6 bin angle histogram vector on blocks(8 x 8). Their set of features is defined as

$$\mu_j \;=\; \frac{1}{6}\sum_{i=1}^{6} g_i \tag{2.2}$$

whereby the gradient histogram is represented by g_i of the jth bin. By taking the mean they are getting the general orientation of a body part and additionally they are using a Gaussian Mixture Model (GMM) to get key poses which are used as input for learning Hidden Markov Models.

Depth information

One special feature, which is available with 3D system, is the usage of the depth information. It can be used for calculation of the relative distance of certain body parts to the sensor. Therefore for example Wang, et al. [WYSL08] used pre-processed depth images to calculate three depth based features. Among other things they derived the following features for their system: [WYSL08]

- Body centroid (2-D):

$$x_b^s \;=\; \frac{\sum_{n=1}^{N} x_n^s 1\{x_n^d \in Depth\ level\ of\ body\}}{\sum_{n=1}^{N} 1\{x_n^d \in Depth\ level\ of\ body\}} \tag{2.3}$$

- Hand displacement (2-D):

$$x_h^s \;=\; x_b^s - \frac{\sum_{n=1}^{N} x_n^s 1\{x_n^d \in Depth\ level\ of\ hand\}}{\sum_{n=1}^{N} 1\{x_n^d \in Depth\ level\ of\ hand\}} \tag{2.4}$$

- Relative depth level of hand (1-D):

$$d_h^r \;=\; x_b^d - x_h^d \tag{2.5}$$

x_b^d...depth level (grayscale value) of the body
x_h^d...depth level (grayscale value) of the hand

In this example they used these features for detection of 9 different gestures for gaming applications, i.e. different boxing actions. In their experiment the overall recognition accuracy was around 92%.

2.4 Classification techniques discussion

2.4.1 Overview

In the previous section different classification techniques including several methods and features were introduced. In Figure 2.7 an overview of some examples where these techniques were used is illustrated. It can be seen that approaches using Hidden Markov Models compared to Support Vector Machines and Neural Networks reached predominantly better results. In addition to the recognition rate, the used databases, features, activities/gestures, pros and cons are also listed.

Methods	Paper	Database		Activities/Gestures	Features	Correctness	Pros	Cons
NNs (Neural Networks)	Murakami and Taguchi [MT91]	Posture database	42	finger alphabet of 42 japanese characters	16 data items(bending, angles, positional data)	~80%	good for pose recognition	not suitable for dynamic recognition
SVM (Support Vector Machine)	Meng et al. [MPB07]	KTH* database	6	Walking, Jogging, Running, Boxing, Hand waving, Hand clapping	Motion Geometric Distribution (MGD), Histogram of Motion History Image	~80%	Computational inexpensive - for usage on embedded systems	Not very accurate
	Schüldt et al. [SLC04]	KTH* database (Split with respect to subjects: 8/training, 8/validation, 9/testing)	6	Walking, Jogging, Running, Boxing, Hand waving, Hand clapping	Local features kernel as motion descriptor	~72%	Robust to variations in the scale, the frequency and the velocity of the pattern	Not very accurate with similar motions (jogging-running-walking)
HMM (Hidden Markov Model)	Mendoza and Pérez de la Blanca [AMPdlB07]	KTH* database	10	Walking left, Walking right, Jogging left, Jogging right, Running left, Running right, Boxing left, Boxing right, Hand clapping, Hand waving	Contour Features based on Shape Context	~96%	Low level features usage for real-time appliance	contour evaluation necessary
	Chakraborty et al. [CRG08]	KTH* database + Hermes indoor sequences	6	Walking, Jogging, Running, Boxing, Hand waving, Hand clapping	HOG (Histogram of oriented gradients) features of different body parts from 8x8 blocks	~80%	Detection of body parts is robust	not very accurate, detection of body parts necessary
	Yamato et al. [YOI92]	300 sequences, 3 individuals	6	Backhand stroke, Forehand volley, Forehand stroke, Smash, Serve	rated pixelcount as mesh feature of 8x8 blocks	~96%	High accuracy, easy feature evaluation	only pixelcount used, tested on low resolution images
	Wang et al. [WYSL08]	336 sequences, 8 individuals	9	Defense, Dodge, Dash, Hook, Uppercut	Body centroid (2D), Hand displacement (2D), relative depth level of hand (1D)	~92%	High accuracy, usage of depth information	Downsampling due to complexity necessary

*...2391 sequences, 25 individuals

Table 2.7: Comparison of related approaches

2.4.2 Methods

Three main methods which are used in human gesture or rather motion recognition were pointed out in the Methods section.

Neural Networks were successfully applied in many areas and have found their application in pattern recognition in the field of static (pose) detection. As sign language involves static and dynamic parts in [MT91] an approach with NNs is presented. In this approach a recognition rate of 77% for independent users was reached and they are therefore suitable for motion recognition systems.

Approaches with Support Vector Machines in the field of motion detection, as mentioned above, showed that they are not as accurate as other methods. Literature research pointed out that the use of different features also did not accomplish results better than around 85% mentioned in [MPB07].

Hidden Markov Models compared to the described methods in the section of motion recognition reached more efficient results around 96%. On the other hand, the disadvantage may be that they are getting computational expensive for a great data volume or if wrong settings are used.

For this approach the main goal is the recognition of dynamic activities. As HMMs delivered the best recognition rates in the literature, they are used in this thesis for the classification in context of fitness and dance training.

2.4.3 Features

Some examples of good features for motion and gesture recognition are described in the previous section. Most of them are based on basic image processing tasks which have been use since the beginning of motion detection. With the increased use of stereo systems, features for depth information are also becoming more important.

The most trivial approach is the relative pixel count introduced by Yamato et al. [YOI92] which delivered accurate information about the image content. In contrast a more complex way is the calculation of the shape context, which involves more computational power. Especially as the testing results showed that both features delivered good performances a less sophisticated method may be preferred.

In addition to the general mono features, the information from depth images is a common way of detecting movements. Especially motions concerning a change of the distance to the sensor can be detected as these movements are not well recognized by normal cameras. The experiments therefore revealed that for the boxing game example the detection works very well [WYSL08]. On the other hand, for other applications taking just the depth information may lead to bad results.

In general, a combination of stereo features with mono features could be a promising solution for reaching even higher recognition rates then listed above. Of course it has to be taken care that no redundant features are used and so evaluation of the single features is also an important task.

CHAPTER 3

Methodology

3.1 Multimedia platform system design

The general system design for the multimedia management platform in the EU project Silvergame [SJSB09] consists of several hardware and software components. The hardware components are a standard TV, a web cam, a microphone, a set-top-box, a tablet as remote control and the event-driven 3D vision sensor (ATIS). In Figure 3.1 the general overview of the multimedia platform including the used hardware components is shown.

The sensor is connected to the multimedia-application management platform with the TV that is used as an output for the whole system. The platform thereby offers a basic communication system, which supports chat and video conferencing. Additionally, it can be extended through modules whereby for the first system three modules have been developed. One module offers a virtual driving simulator, which provides the possibility to train for a real live driving situation. A second module contains a virtual song club where elderly people can come together and sing together. The third one is the dance and fitness training module for which the human activity recognition algorithms shall be implemented. This recognition of human motion is done with the novel event-driven 3D vision sensor. A virtual fitness advisor performs a dance which the users should try to follow. With this module also some feedback about the performed dance shall be provided.

Figure 3.1: Overview of the multimedia platform system design of the EU project Silvergame [SJSB09]

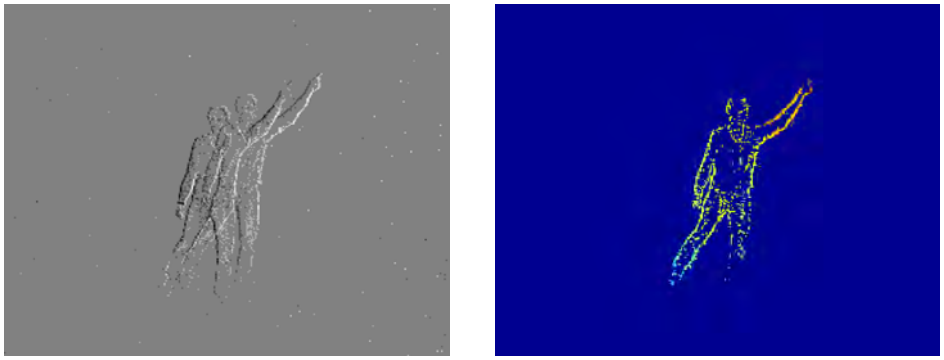
3.2 Event-driven 3D Vision Sensor (ATIS)

As described briefly in the previous paragraph the dance and fitness application for recognition of human actions is based on an event-driven 3D vision sensor. Therefore, the latest version named ATIS will be used for the system. The sensor is placed on the top of the TV or close to it so that the exercises can be done directly in front of it. In doing so the sensor offers a wide field of view and offers a large motion area.

As mentioned before, the ATIS sensor is intended to detect a change in intensity and to transmit this change for each pixel. It has to be noted that the sensor doesn't work on a frame based acquisition but provides so-called address event (AE) data which is described in the next part. With these characteristics in mind it is obvious that when a person is moving in front of the sensor this movement is extracted automatically. This brings the major advantage that the moving person is separated from the background and so the demanding part of extracting the person is already done by the camera. It also must be noted that the systems works in a wide dynamic range, so that it can be used even in bad light conditions. This and the fact of a high time resolution makes it predestined for indoor & outdoor motion recognition. Examples of the data acquired during one of the fitness activities, both an overlay left-right image and the depth image, are shown in Figure 3.2.

In addition to the 'normal' AE information, the sensor with its stereo configuration provides depth information about the activated pixels. Calculations via depth analysis can be included in the classification process with this supplemental information. This particularly helps to de-

tect movements along the z-axis or rather a change in distance to the camera, which cannot be detected easily with normal video information.



(a) overlay left-right image (b) stereo image

Figure 3.2: ATIS examples (displayed frame-based) recorded during execution of one activity

3.2.1 AE data format

The 3D vision sensor streams its data as so-called address event data, abbreviated AE data. With this data several pieces of information about an activated pixel, within the field of view, is transmitted. For this version and to use the 3D information of the sensor the 32bit pTAE format 1 with ATIS stereo extension for disparity information coding is used. The pTAE stands for polarity Time Address Event because data addressing pixels and a time stamp are transferred.

More precisely a data stream contains three different 32bit data words:

- address event (AE) data
- time stamp (TS) data
- wrap around (WA) data.

In a sequence it is specified that it starts with a TS followed by an AE word. A WA can only occur before a TS or another WA. For example a correct sequence looks like this:

[TS AE TS AE TS AE ... TS AE WA TS AE TS AE ...]

In addition to the option of transferring stereo data, mono data can also be processed with this format. In Figure 3.3 the structure of the elements is defined.

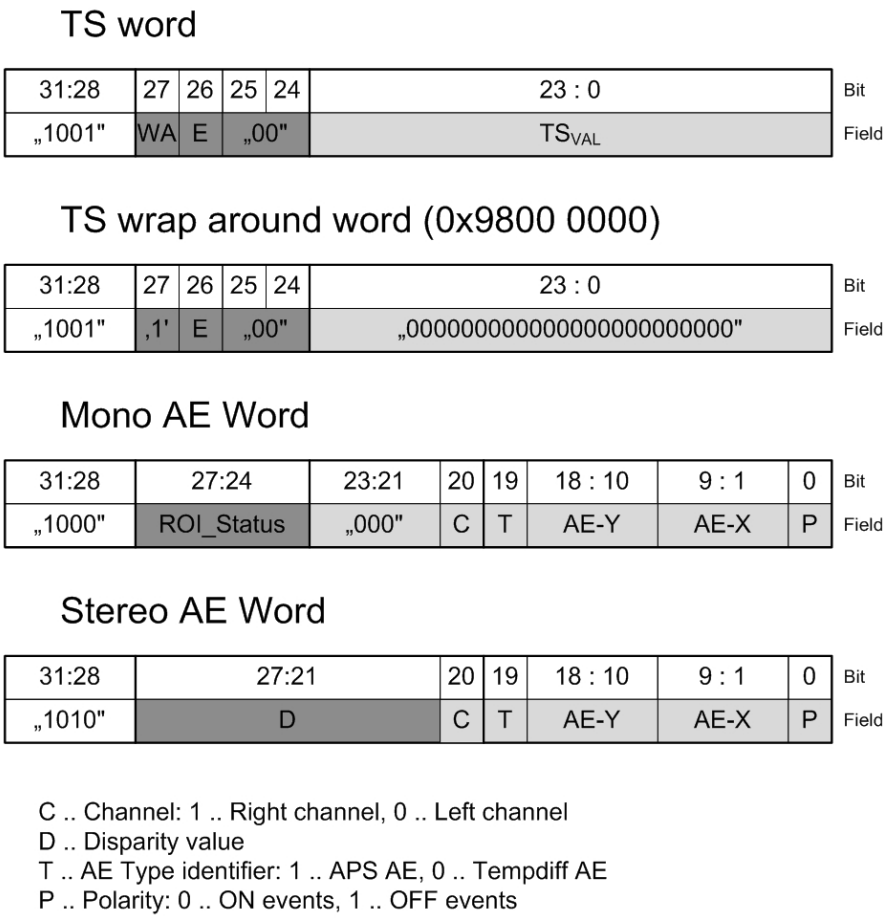


Figure 3.3: pTAE structure

On the basis of the defined words from Figure 3.3 and the sequence stream, data for the motion recognition is acquired. Among other things this data format provides full information about the pixel coordinates and the time when the pixel is activated. In doing so the pixels within a defined period of time can be combined and displayed as an ordinary AE image frame or as a depth image as shown in the last subsection in Figure 3.2.

3.3 Dance and fitness training

For the recognition of dance and fitness training exercises, it is very important to define which figures shall be detected and how. In the human motion recognition various approaches for different kinds of human motion have been introduced so far. There is thereby a difference between the complexity of the different gestures and which methods as well as which features are used for the design of the classification system. So it is obvious that the detection of gestures for human computer interaction is not the same as detecting certain human motion during sport exercises. The general structures including the steps that are necessary before the classification part have been introduced in the previous section. Therefore, the Reha-Zentrum Lübben[1] made a choreography suitable for elderly people which combines eight different activities accompanied by the song Mama Mia (music band ABBA). An overview about the different fitness activities which can be combined to whole dances and expanded in the future and which features are relevant as an input for our classification method will now be introduced.

3.3.1 Fitness activities

Before the activity recognition is introduced, it is important to know which activities shall be detected and what they look like. In general, for the application of dance and fitness training, it can be noted that one dance is a combination of N different figures or activities. More precisely, each activity is defined by a certain movement. Activities which are then aligned with each other are so-called dances. For example, one activity can involve a movement of arms and feet and the next one only the arms. It is now obvious that with this wide range of combinations many different activities and even more dances can be defined. For this work certain activities have been defined and chosen for the recognition task, as the recognition of motions of single body parts would involve too much complexity. A chart of the different parts which can be combined into one activity are shown in Figure 3.4. With these different motions one dance was defined for tests of the recognition system.

[1] http://www.rehazentrum.com/

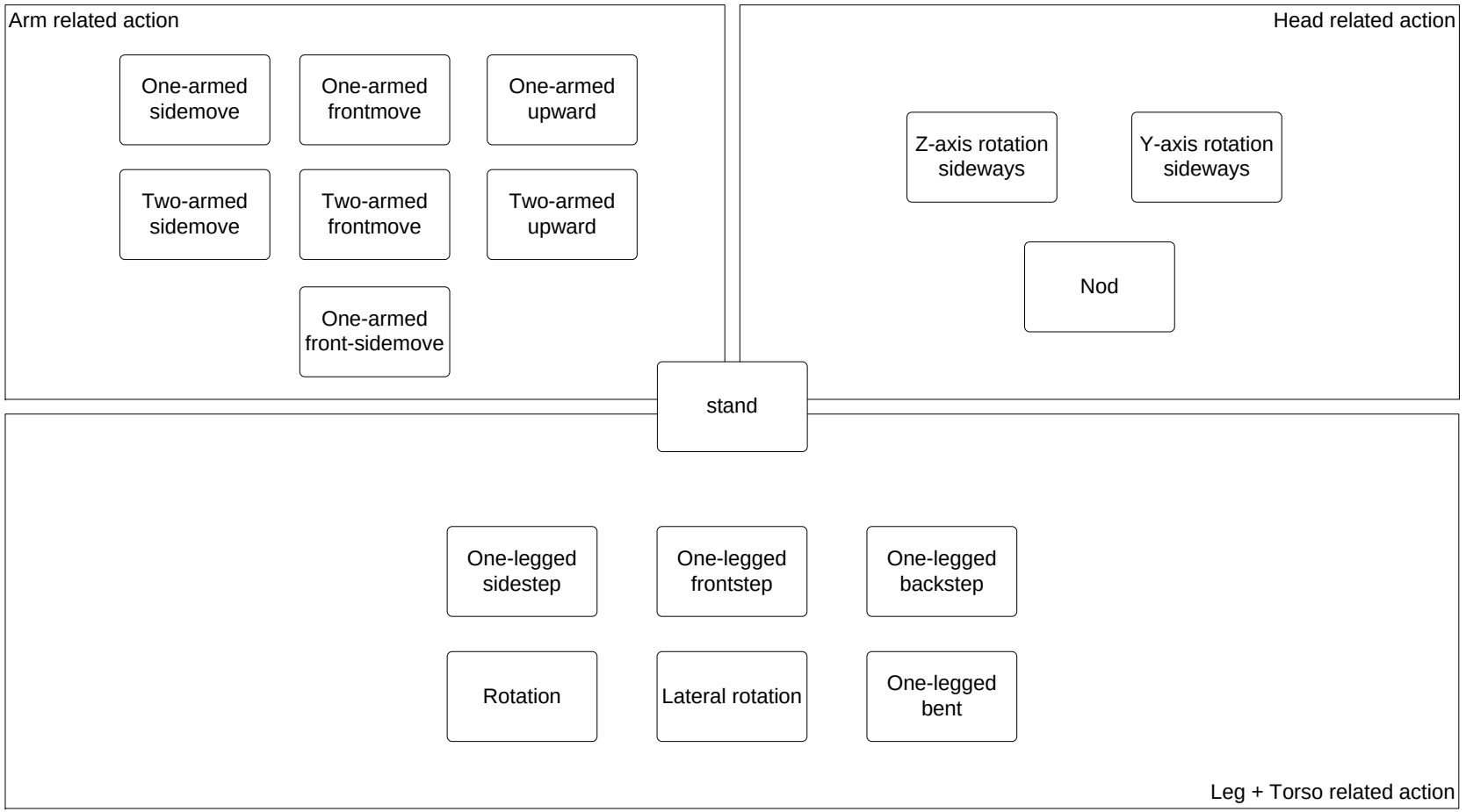


Figure 3.4: Different motions which can be combined to an activity

For the first tests 8 different activities were defined and combined into a dance. The filmstrips in Figure 3.5 roughly show some examples of motions in these activities performed from a fitness advisor of the Reha-Zentrum Lübben:

(a) Activity 1: ArmsFrontStrechtedAndCrossed

Figure 3.5: Examples of different Activities (1)

(b) Activity 2: ArmsPointingWith180DegreeLeftRightSideRotation

(c) Activity 3: ArmsPointingWith360DegreeAxisLeftRotation

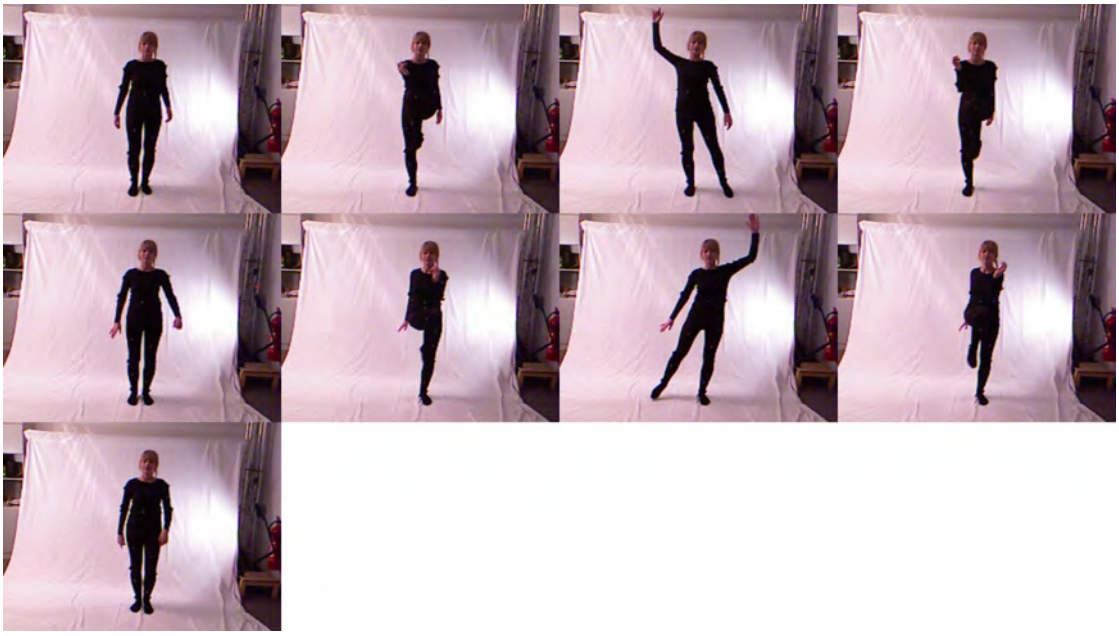
(d) Activity 4: ArmsPointingWith360DegreeAxisRightRotation

Figure 3.5: Examples of different Activities (2)

(e) Activity 5: ArmsWavingTopDown

(f) Activity 6: BentDownWithArmsBackCrossed

Figure 3.5: Examples of different Activities (3)



(g) Activity 7: ElbowToKnee

(h) Activity 8: LegsFrontStretchedWithShoulderRolling

Figure 3.5: Examples of different Activities (4)

3.4 Motion recognition system workflow

For the design of the motion recognition algorithm it is necessary to define a general structure of how the recognition is done. Several steps must be done before a feature vector that can be used as an input for the classification method is obtained. In the previous chapter research of the literature showed that the most promising method to achieve good recognition rates, for motion recognition, are Hidden Markov Models. Therefore, HMMs have been chosen as the classification method in this thesis.

The first step in this process is the acquisition of data from the already described event-driven 3D vision sensor ATIS. As the sensor itself delivers a relatively high resolution of 304x240 pixels a down sampling to a lower pixel rate may be necessary for a real time recognition system. For the first implementation and tests of the system such a step is not included so far but can be easily added for further tests. As the sensor delivers contour-based data including depth information no additional pre-processing steps must be applied.

After these steps the actual recognition can be done. As an option a calculation of a bounding box around the region of interest, the human body, may be applied. In doing so the area that is used for the feature calculation can be decreased. For this implementation the bounding box calculation was left out and as the next step the feature extraction is done.

For performance reasons it is important to choose features which are not too computationally expensive. Common features such as relative pixel count, gradient, and shape context are good examples to use. More precisely for this system the relative pixel count and the depth information are used.

For reducing the size of the feature vector a quantization step is applied, whereby the vector is compressed with the help of a discrete cosines transformation (DCT), which is also used in the approach of Mendoza and Pérez de la Bianca [AMPdlB07]. After doing this the vector is taken as an input for the activity recognition. This is done with Hidden Markov Models in particular HMMs with Gaussian outputs.

An overview of the general structure of this activity recognition analysis can be found in Figure 3.6. The process of the data acquisition, feature extraction and activity recognition with HMMs is described in detail in the following sections.

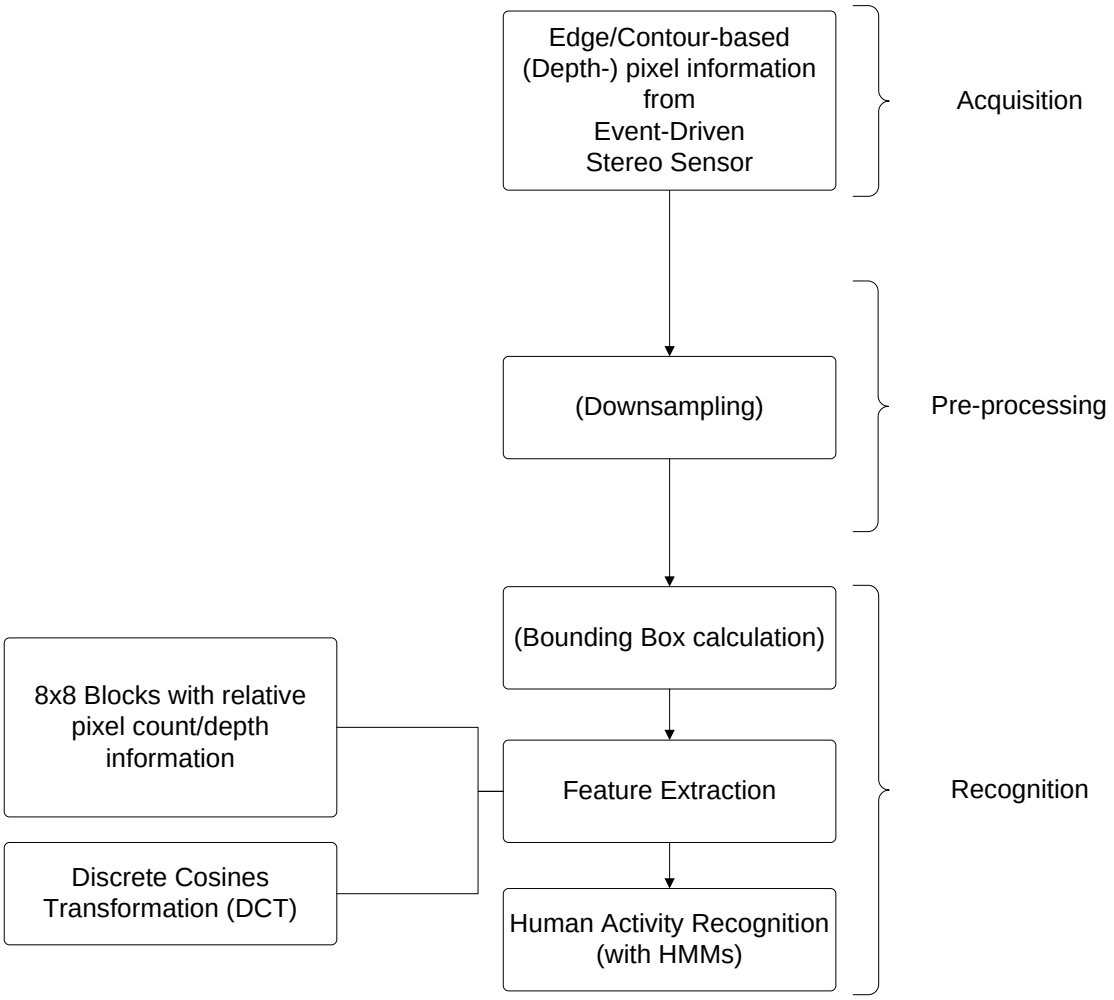


Figure 3.6: A general structure and workflow of the motion recognition system

3.5 Features

Based on the literature, the event-driven sensor, the defined activities and keeping in mind that the feature calculation shall be computationally inexpensive, two promising types of features namely relative pixel count and relative disparity are used in this thesis. This calculation is based on so-called mesh features, such as in the approach of Yamato et al. [YOI92], which is illustrated in Figure 3.7 for the 3D data used in this work. The feature vector is then calculated with the Equation 3.1 by using one method of the two different features listed below. Therefore feature types were chosen with the HMMs in mind. They can also be used for other classification methods.

$$featureVector = (f(0,0), f(0,1), ..., f(i,j), ...f(M,N)) \tag{3.1}$$

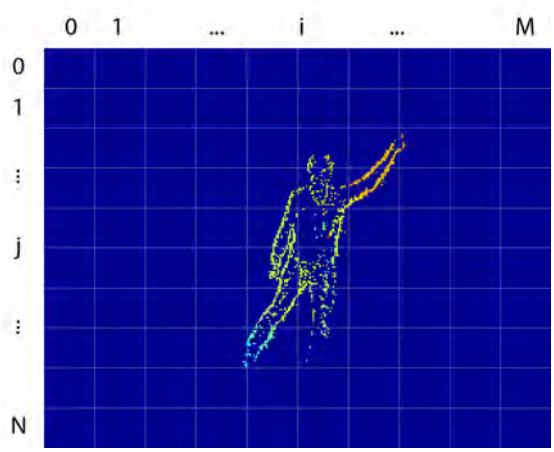


Figure 3.7: Features extracted from the blocks using the equations 3.2 and 3.3

3.5.1 Relative pixel count

With the use of the relative pixel count as a feature vector as introduced by Yamato et al. [YOI92], a representative feature can be extracted. Therefore this method is also used here. Originally this feature was used for motion detecting of certain tennis strokes during tennis matches. For the activities, mentioned in Section 3.3, the relative pixel count is extracted from 8x8 blocks like in [YOI92]. It has to be noted that with this sensor the calculation can be done on the fly as it delivers the activated pixels. By only calculating the sum of the pixels in the blocks again computational power can be saved. The calculation of the features for each block is illustrated in the equation 3.2 and the feature vector is generated with the equation 3.1.

$$f(i,j) = \frac{number\ of\ activated\ pixels(i,j)}{(M_m\ x\ N_m)} \tag{3.2}$$

3.5.2 Relative disparity/distance

As the sensor also enables the use of depth information and in terms of these activities the usage is helpful, a feature from this information is also calculated. A technique similar to the relative pixel count is used for the calculation of the relative disparity. In doing so the distance is also calculated from 8x8 blocks along the x-and y-dimensions from the frames of each activity. The main difference is that within these blocks the maximum disparity value is calculated and divided through the maximum disparity value of the whole frame. This task is mainly done to guarantee that individuals at a distance are treated the same as very close individuals. According to the equation 3.3 the features for each block are calculated and again the feature vector with equation 3.1 is generated.

$$f(i,j) \; = \; \frac{max\ disparity\ of\ activated\ pixels(i,j)}{max\ disparity(M_m\ x\ N_m)} \tag{3.3}$$

To use these two feature classes as an input for the HMM classifier each class is used individually or combined. For mono data only the relative pixel count is used. A combination of these two feature types, as in Equation 3.4 is done for stereo data. Additionally, as the high resolution yields a high feature vector size, a discrete cosines transformation (DCT), as for example in [AMPdlB07], to reduce the vector size is applied. In doing so the first 16 coefficients, which represent the extracted features from common video information, are used when only the pixel count feature class is taken for the feature vector. In almost the same manner the calculation for both feature classes is done. For the combination of the pixel count and the disparity, the first 8 coefficients from the pixel count and the first 8 coefficients from the disparity information are used. For a better usage of the data with the toolbox in MATLAB these vectors are then also standardized.

$$\begin{aligned} combinedFeatureVector \; = (&f_{pc}(0,0), f_{pc}(0,1), ..., f_{pc}(i,j), ..., f_{pc}(M,N), \\ &f_{dy}(0,0), f_{dy}(0,1), ..., f_{dy}(i,j), ..., f_{dy}(M,N)) \end{aligned} \tag{3.4}$$

$f_{pc}(i,j) = feature\ vector\ for\ relative\ pixel\ count$
$f_{dy}(i,j) = feature\ vector\ for\ relative\ disparity$
$i = 0, 1, ..., M$
$j = 0, 1, ..., N$
$M, N = number\ of\ blocks\ (block\ size\ =\ 8x8\ pixels)\ along\ axis$

With these steps the final feature vector sequences for the different activities are prepared. In the next section the general learning and recognition process for the HMMs is described.

3.6 Classification with HMMs

In this thesis HMMs are used for motion recognition and in Section 2.3 the different classification methods including Hidden Markov Models were described in general. As already mentioned these HMMs are a good method for dynamic motion recognition, and that is why they are used in this work as classification technique.

Basically the triple $\lambda = (A, B, \Pi)$, including the transition probabilities A, emission probabilities B, and the initial state probabilities Π are the main parameters which must be defined for a usage of HMMs. These parameters are defined through a randomized initialization process, stating the number of states Q and observations O must be done before. Besides these two parameters some more details need to be known for a more detailed description of HMMs.

Overall discrete observations can be distinguished from continuous observations and the transitions properties can also vary between the different types of HMMs. So it is possible to define the transitions by the order of the Hidden Markov Model. The order thereby defines the memory of the model. That means that the system with an order N only depends on N states and not more. Another special case is a so-called left-right model, which can be seen as a model of order 1.

Different types of Hidden Markov Models exit and in this approach left-right HMMs are used. An illustration of such a 4-state model and the form of the state transition matrix is shown in Figure 3.8. Additionally, the outputs of the states can be represented with discrete or continuous values. In this thesis a Gaussian mixture distribution is used to represent the outputs and to handle complex behaviour. In doing so in addition to the parameters mentioned above, the number of Gaussian density function M must be specified. A Gaussian output is defined by taking just one Gaussian function. However mixed Gaussian outputs are then described as a sum of n functions. The way in which the learning and recognition process is realized is pointed out as follows.

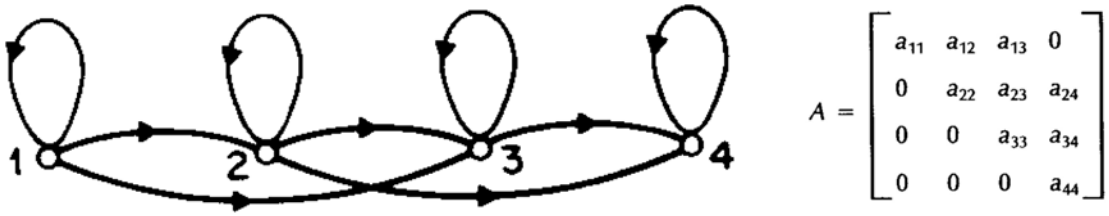


Figure 3.8: A 4-state left-right model including the corresponding state transition matrix [Rab89]

3.6.1 Learning

As mentioned above the initialized parameters must be adapted to the data for better recognition results through a learning phase. In this process the parameters $\lambda = (A, B, \Pi)$ are adopted so that $P(O|\lambda)$ is maximized and that the model fits the data.

For this problem the Baum-Welch algorithm also referred to as Forward-Backward algorithm is applied in this thesis. In doing so the algorithm uses a generalized expectation-maximization (EM) algorithm to iteratively determine the likelihood of a defined set of data [Dud04]. As the recognition algorithms are designed to detect different activities, one HMM for each activity is initialized and therefore for each of these HMMs a learning phase is carried out. The models with the adopted, trained parameters are then used to compare a sequence with the models and to classify the activity sequences. The detailed recognition process is described in the following sections. How the Forward-Backward algorithm works is illustrated in the pseudo code in Algorithm 3.1.

begin initialize a_{ij}, b_{jk}, training sequence V^T, convergence criterion $theta, z \leftarrow 0$
  do $z \leftarrow z + 1$
    compute $\hat{a}(z)$ from $a(z-1)$ and $b(z-1)$ by Eq. 3.5
    compute $\hat{b}(z)$ from $a(z-1)$ and $b(z-1)$ by Eq. 3.6
    $a_{ij}(z) \leftarrow \hat{a}_{ij}(z)$;
    $b_{jk}(z) \leftarrow \hat{b}_{jk}(z)$;
  until $\underset{i,j,k}{max}[aij(z) - a_{ij}(z-1), b_{jk}(z) - b_{jk}(z-1)] < 0$;
  return $a_{ij} \leftarrow a_{ij}(z); b_{jk} \leftarrow b_{jk}(z)$
end

Algorithm 3.1: Pseudo code of the Baum-Welch alogrithm [Dud04]

$$\hat{a}_{ij} = \frac{\sum_{t=1}^{T} \gamma_{ij}(t)}{\sum_{t=1}^{T} \sum_{k} \gamma_{ik}(t)} \tag{3.5}$$

$$\hat{b}_{jk} = \frac{\sum_{t=1, v(t)=v_k}^{T} \sum_{l} \gamma_{jl}(t)}{\sum_{t=1}^{T} \sum_{l} \gamma_{jl}(t)} \tag{3.6}$$

- $a_{ij}...transition\ probability$
- $b_{jk}...emission\ probability$
- $\hat{a}_{ij}...transition\ probability\ ratio\ from\ \omega_i(t-1) to\ \omega_j(t)$
- $\hat{b}_{jk}...emission\ probability\ ratio\ from\ \omega_j(t-1) to\ \omega_k(t)$
- $\gamma...transition\ probability$

3.6.2 Recognition

For the recognition of an activity the generated and trained Hidden Markov Models are used. For this task two algorithms can be applied. In general this calculation can be seen as an evaluation or decoding task by deriving log likelihoods. Therefore these two algorithms are often referred to Evaluation and Decoding algorithms.

Evaluation

One opportunity is to use the Forward- or Backward algorithm to get the probability that the HMM generated the actual activity. Such a likelihood is generated for all the HMMs from the learning phase. With a maximum likelihood estimation, which is done by taking the HMM with the highest likelihood, the activity is estimated. In doing so the HMM that is most likely to generate a certain activity is recognized. The pseudo code of the HMM Forward algorithm is illustrated in Algorithm 3.2 and the time-reversed version thereof, the HMM Backward algorithm is shown in Algorithm 3.3.

1
2 **initialize** $t \leftarrow 0, a_{ij}, b_{jk}$, visible sequence $V^T, \alpha_j(0)$
3 **for** $t \leftarrow t + 1$
4 $\alpha_j(t) \leftarrow b_{jk}\upsilon(t) \sum_{i=1}^{c} \alpha_i(t-1)a_{ij}$;
5 **until** t=T
6 **return** $P(V^T) \leftarrow \alpha(T)$ for the final state
7 **end**
8

Algorithm 3.2: HMM Forward algorithm [Dud04]

initialize $\beta_j(T), t \leftarrow T, a_{ij}, b_{jk}$, visible sequence V^T
    for $t \leftarrow t-1$
        $\beta_i(t) \leftarrow \sum_{j=1}^{c} \beta_j(t+1) a_{ij} b_{jk} v(t+1);$
    until t=1
return $P(V^T) \leftarrow \beta_i(0)$ for the known initial state
end

Algorithm 3.3: HMM Backward algorithm [Dud04]

Decoding

The other method is to use the method of finding the optimal state sequence or also called Viterbi path. This is done with the Viterbi algorithm. With the computation of the so-called Viterbi path the different models can be compared by taking the path which is generated for a given sequence. Again the path with the highest likelihood is chosen as the expected activity. Again the pseudo code of decoding the HMM can be found in Algorithm 3.4.

begin initialize Path $\leftarrow \{\}, t \leftarrow 0$
        for $t \leftarrow t+1$
            $j \leftarrow j+1;$
            for $j \leftarrow j+1$
                $\alpha_j(t) \leftarrow b_{jk} v(t) \sum_{i=1}^{c} \alpha_i(t-1) a_{ij};$
            until $j = c$
            $j' \leftarrow = arg\ \underset{j}{max} \alpha_j(t);$
            Append $\omega_{j'}\ to$ Path
        until $t = T$
    return Path
end

Algorithm 3.4: HMM decoding algorithm [Dud04]

In this thesis the Forward algorithm is chosen for deriving the maximum log likelihood, as it was also used in the approaches [WYSL08, CRG08].

CHAPTER 4

Implementation

4.1 Overview

For the implementation of the activity recognition system, MATLAB was chosen as programming platform. The findings of this implementation shall then be used for porting the algorithm in C++ programming language and for an integration in the fitness and dance module in the EU project Silvergame [SJSB09].

In general this system contains among other things the tasks for

- **Acquisition** of the data transferred from the event-driven
- **Test database** generation from the acquired data
- **Feature extraction** from the generated test database
- **Training HMMs** for each activity in the test database
- **Classification with trained HMMs** with the usage of a cross validation

In Figure 4.1 a general overview of the workflow for the implemented methods in MATLAB is shown. For the actual classification with the HMMs multiple steps have been implemented and have to be executed. To establish the classification with Hidden Markov Models the toolbox from Murphy [Mur05] was used and adopted for this system. A detailed description about the applied evaluations can be found in the following sections.

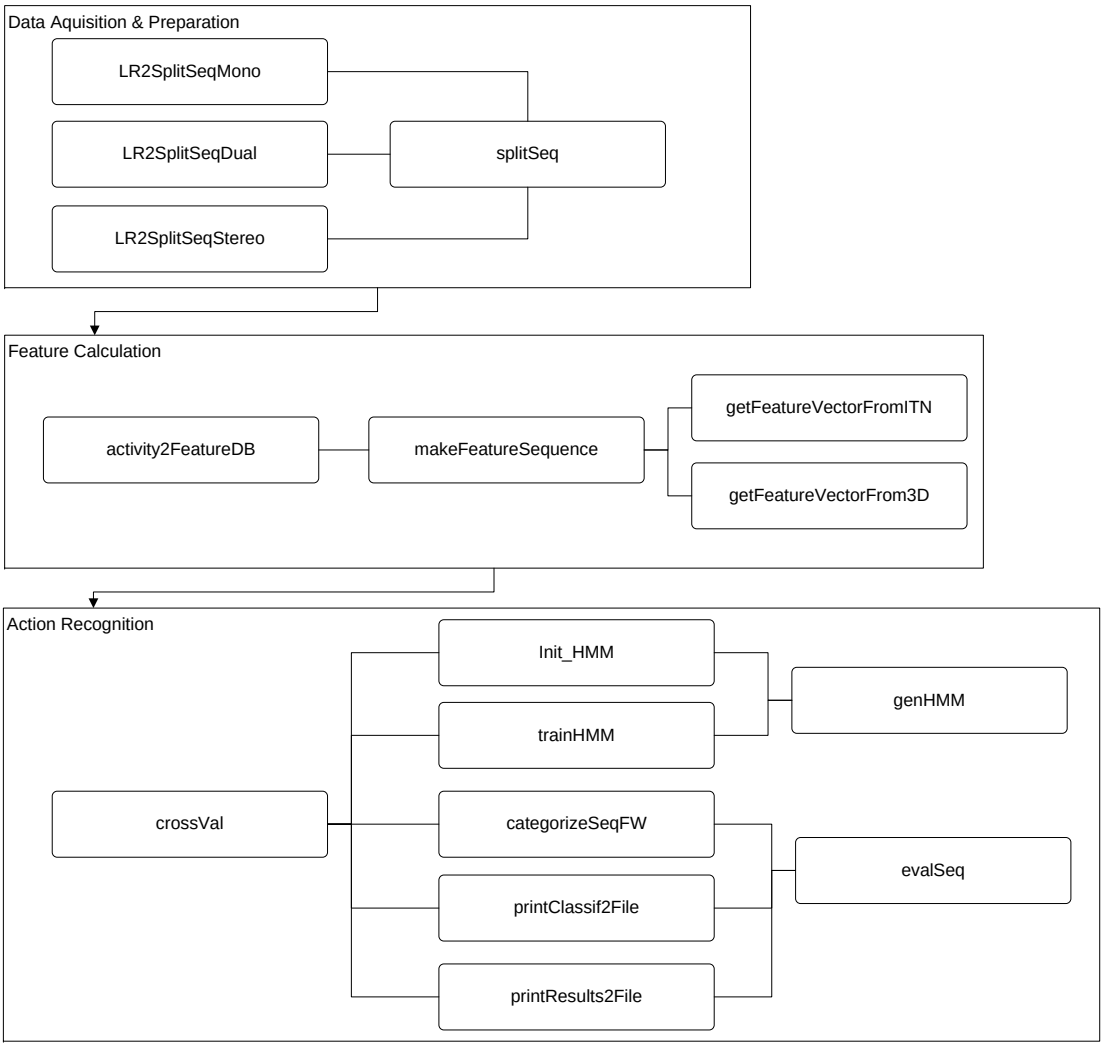


Figure 4.1: Components used for the recognition system

4.2 Data acquisition and preparation

For the data acquisition and preparation several steps are performed. First the AE data stream from the sensor is saved in .bin files. As a stereo system was used an individual file for the left and the right sensor is saved. For each activity a session with several iterations is recorded. Each iteration of an activity needs to be split into a separate file.

To handle the splitting of the data files with the several iterations the `splitSequences` package was created to generate a sample database. Furthermore a very important step is to prove that by using a stereoscopic sensor the recognition rate increases. Therefore three different sets of sequence files have been created:

- `Mono data` (only data from one sensor is used)
- `Left-right overlay mono data` (data from left and right sensor are overlaid, slight shift of the person is seen but no stereo depth information is used)
- `Stereo data` (stereo data is used, disparity information from the slight shift is calculated)

The splitting of the streams into the single activities is done with the help of the function `splitSeq(A,B,begin,ende,fall)` whereby `A` is the data stream, `B` is the array with the absolute timestamps, `begin` is the start time and `ende` is the end time of the activity to be extracted. The option `fall` sets which data (mono, left-right or stereo) shall be extracted.

For automation purpose following scripts are provided:

- `LR2SplitSeqMono`
- `LR2SplitSeqdual`
- `LR2SplitSeqStereo`

The scripts have to be configured regarding the path of the input files and the path where the divided activities shall be saved. This is done with a config file where these parameters are set. For the stereo calculation the application `stereolib_atis.exe` computes the 3D sequences with the camera specific parameters, which are specified in the stereo config file.

After this process of preparation, the features can be extracted and a test and training database with the features can be generated.

4.3 Feature extraction and DB generation

For the extraction of the features and generation of respective databases the responsible package `featureCalculation` is used. The script `activity2FeatureDB.m` handles this calculation and stores the feature vectors in databases. Several databases from the samples were thereby created as some of the samples have been recorded with different settings and cannot be used for the database to train HMMs. Some of the samples are also ”bad” examples and have

to be stored in a separated test database which can be used for further evaluations. This script needs a config file with several parameters for correct appliance. In this file among other settings the type of data (mono, left-right, stereo), the input/output folders and the activity files used for training or test database are specified.

Based on these settings, a feature sequence is extracted and the feature database for every activity is generated. Within the function `makeFeatureSequence.m` the functions `getFeatureVectorFromITN(aes,varargin)` and `getFeatureVectorFrom3D(aes,varargin)` are used.

In both functions the relative pixel count and for the 3D case additionally the relative disparity, see Section 3.3, is derived from NxM pixel blocks. In addition, a discrete cosines transformation (DCT) is applied to reduce the size of the feature vector sequence. The size of the pixel blocks and the number of coefficients can be specified within the two functions. In Algorithm 4.1 the function for the calculation of the combined feature vector is illustrated as pseudo code. The calculation of the feature vector for a single feature is similar, apart from the fact that only `countVector` or `dispVector` are used for the feature vector.

In order to have a better input for the training of the HMMs within this HMM toolbox the feature vector sequences were scaled. This is basically done because during the training process it can occur that the log-likelihood becomes positive. Murphy [Mur05] states in the toolbox manual that the function `KPMstats\standardize` can be used to guarantee that the input vector has small and matchable magnitudes. By applying this function the feature sequence is added to the database matrix in the `makeFeatureSequence` function. After executing the package activity, databases including the features are stored and can be used for the recognition with HMMs.

```
1
2  initialize frameAECount ← {current frame with count of AE's},
3             frameAEDisp ← {current frame with disparity values of AE's},
4             count = 8
5
6  for x ← 1 step count to xsize
7      for y ← 1 step count to ysize
8          block ← frameAECount(x : x + (count − 1), y : y + (count − 1);
9          countVector(index) ← sum(block)/(no. of meshes);
10         block ← frameAEDisp(x : x + (count − 1), y : y + (count − 1);
11         dispVector(index) ← max(block)/max(frameAEDisp);
12     end
13 end
14 countVector ← DCT(countVector, 8);
15 dispVector ← DCT(dispVector, 8);
16 featureVector ← standardize(countVector, dispVector);
17 return featureVector
18
```

Algorithm 4.1: Pseudo code for the combined feature vector

4.4 Activity Recognition

With the training database the recognition with Hidden Markov Models can be realized. Therefore the package `Activity Recognition` provides several scripts and function for different steps containing the training and classification of the HMMs and a cross validation. How this is done in detail is described as follows.

4.4.1 Training HMMs

A step before the training of HMMs is to initialize the parameters that are needed. Basically the script `init_HMM(data, Q, M, cov, left_right)` is used to get the initial parameters which are needed as an input for the training function. In doing so only the database, the number of states `Q`, the number of Gaussian density function `M`, the covariance type `cov` and the option for the dimension of the HMM `left_right` must be specified. The covariance type can be switched from 'full', 'spherical' and 'diag' and the option 1 for `left_right` leads to the usage of a so-called left-right Hidden Markov Model.

The actual training of the HMMs is done with the function `trainMHMM(data, prior0, transmat0, mu0, Sigma0, mixmat0, numb_it, cov, prior_adj)`.. The training is thereby done with the initialized parameters according to the number of training iterations `numb_it` and the prior adjustment value `prior_adj`.

As a Hidden Markov Model must be calculated for every activity database the process of the initialization and training is automated within the script genHMM. For each activity a model is generated and saved in a struct. Thereby n=1...N defines the amount of HMMs, HMM_1...HMM_n, generated.

4.4.2 Classification with trained HMMs

After training and saving the generated HMMs the next step in the system is the classification of sequence with these models. Basically in this system the two different methods for this evaluation are implemented in the functions

- `categorizeSeqFW(data,HMM)` using the Forward algorithm
- `categorizeSeqVP(data,HMM)` using the Viterbi algorithm.

A sequence of data is thereby compared with the generated HMM, using one of the two common algorithms, and the likelihood is returned. To classify a given sequence with HMMs the functions only have to be called with the structure of HMMs and the data to be tested.

For automation the script `sevalSeq` with an appropriate config file can be used to evaluate a sequence of data with a specified HMM structure. Within this script the functions are called and the analysis is recorded with helper functions, which are described in detail in the next section.

4.4.3 Cross Validation

Additionally in this package a leave-one(-person)-out cross validation was implemented for testing the classification system. With this implementation a data set with N individuals can be selected and an evaluation of the classification for N-1 people is done. Not only one sample is thereby left out but also if there exist more samples of the same person all the samples are not used for training. The left out samples will be used as a test set for the classification. In doing so a representative evaluation of the classification technique is provided, whereby the classification is done with the Forward algorithm in the function `categorizeSeqFW(data,HMM)`. For detailed analysis the two helper functions

- `printClassif2File(fn, loglikV, index, person, activity)`
- `printResults2File(fn, evalMat, person, all)`

are used in the `crossval` script. The first function prints the evaluation matrices for each classification cycle and a full classification matrix including the whole cycles, to a file. The second function saves the log likelihoods of each tested sample to a file. In doing so further analysis on the tested samples can be done and detailed information about the classification steps can be gained. In the Algorithm 4.2 the pseudo code with an overview of the implemented leave-one(-person)-out cross validation can be found.

```
1
2  initialize Q ← {no. of states}, M ← {no. of Gaussian mixtures},
3             cov ← {'full','spherical','diag'}, lr ← {1, 0},
4             P ← 15,      % No. of individuals
5             A ← 8        % No. of activities
6  for j ← 1 to P
7      for i ← 1 to A
8          data ← GetDataWithoutPerson(i, j);
9          parameters ← initHMM{data, Q, M, cov, lr};
10         HMM ← trainHMM{parameters};
11     end
12     for i ← 1 to 8
13         data ← GetDataWithPerson(i, j);
14         classId ← classify{data, HMM};
15         printClassif2File(...);
16     end
17     printResults2File(...);
18 end
19
```

Algorithm 4.2: Pseudo code for leave-one(-person)-out cross validation for each person

CHAPTER 5

Experimental results

5.1 Setup and Recording

For the evaluation of the implemented recognition system several activities with different individuals have been recorded. In Figure 5.1 the setup of the recording session is shown.

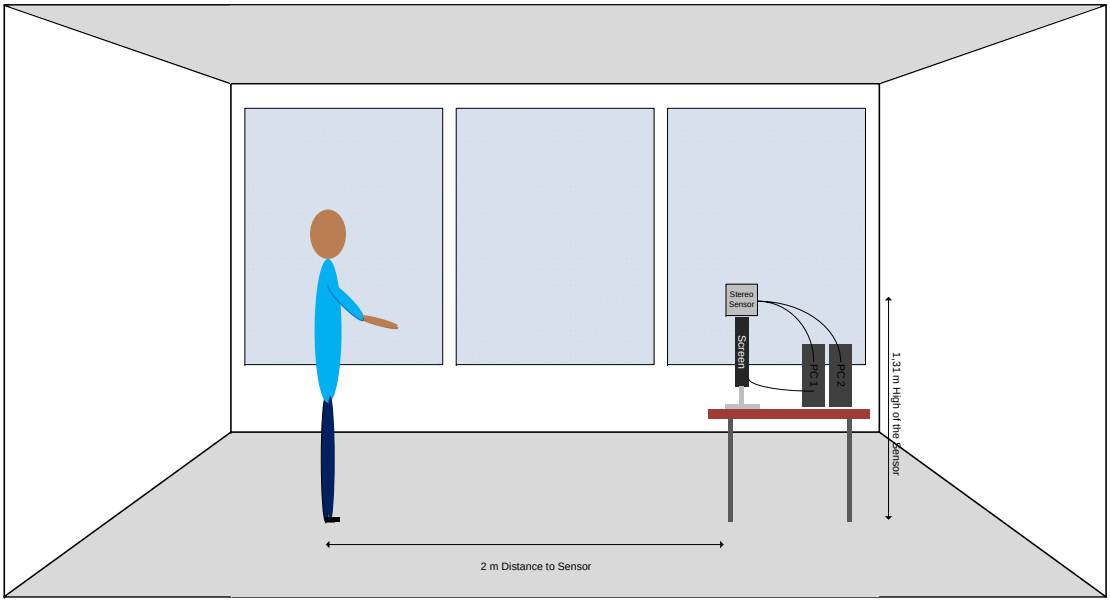


Figure 5.1: Setup of the test session related to normal housing conditions

Attention was paid to the design being comparable to normal housing conditions. Therefore, the sensor was placed above a monitor, like the TV in a private home. On this output device

the fitness activities, performed by an advisor, are displayed and the people tried to follow these activities from an initial distance to the sensor of around 2 meters. The sensor was connected to two personal computers for recording the different activities. As the sensor is in a prototype state, the two sensors were connected to a separate computer.

For this test session the Reha-Zentrum Lübben[1] made a choreography suitable for elderly people. The choreography combines eight different activities, see Section 3.3, accompanied by the song Mama Mia (music band ABBA).

To make it easier for the elderly to follow the advisor, each activity was displayed several times: the first 3 cycles are intended to serve for training, thus were followed by 2 normal cycles and 2 faster cycles. All in all 15 individuals carried out 8 different activities, with several iterations including short pauses between the activities. In doing so 120 sequences were recorded. These sequences had to be processed in the manner explained in section 4.2.

After applying this process step, a full database containing 856 records was obtained. Table 5.1 shows that not all the records can be used for training the HMMs and are therefore stored in different databases. In doing so the difference in speed of the recorded samples can be used to train two different levels of difficulty or it can be used to recognize activities performed with two different speeds.

	Training Database		Test Database		**Total**
	'normal' (1)	'fast' (2)	'normal' (1)	'fast' (2)	
Activity 1	77	23	14	3	117
Activity 2	81	26	13	0	120
Activity 3	72	24	18	2	116
Activity 4	80	24	13	2	119
Activity 5	29	0	1	0	30
Activity 6	90	26	1	0	117
Activity 7	83	23	8	3	117
Activity 8	68	23	26	3	120
Total	580	169	94	13	856

Activity 1...ArmsFrontStretchedAndCrossed
Activity 2...ArmsPointingWith180DegreeLeftRightSideRotation
Activity 3...ArmsPointingWith360DegreeAxisLeftRotation
Activity 4...ArmsPointingWith360DegreeAxisRightRotation
Activity 5...ArmsWavingTopDown
Activity 6...BentDownWithArmsBackCrossed
Activity 7...ElbowToKnee
Activity 8...LegsFrontStretchedWithShoulderRolling

Table 5.1: Database with all recorded samples

[1] `http://www.rehazentrum.com/`

The generated database was used to establish the relative pixelcount/disparity as feature vector for the input of the HMMs. For every activity a Hidden Markov Model is generated with randomized initial values and is trained with the feature data. The recognition process is performed by taking the maximum likelihood estimation algorithm from a sample compared with the models. The model with the maximum likelihood is thereby the estimated correct activity.

As mentioned in section 4.2 three different test set scenarios were generated to prove that a stereoscopic system leads to better recognition rates. In doing so the usage of the additional disparity information compared to normal pixelcount information is analysed. For the scenarios the database 1 from the training set with 580 samples is used for a cross validation. A leave-one(-person)-out cross validation is thereby performed containing the following two basic steps:

1. Training the HMMs without samples of person N

2. Recognition with the samples of person N.

5.2 Results

In this section the results for the three different scenarios are presented. The scenarios are divided into the different data from the sensor thus the first scenario is based on mono information. The second scenario is then related to overlay information and scenario 3 works additionally with disparity information gained from stereo data. All tests were done on Intel(R) Core^{TM}2 Duo CPU E8200 @ 2.66 GHz with 3.23 GB random access memory and the operation software was MATLAB R2010b. As shown in section 4.4 the training of the HMMs depends on the values chosen for the states `Q` and the numbers of Gaussian mixtures `M`. Thus the three scenarios where done with a number of states from 8 to 14 and number of Gaussian mixtures from 4 to 8, increasing both by 2. The number of training iterations `numb_it` was set to 5 in all scenarios. In addition to the results stated in this section, full evaluation matrices of the best results of all three scenarios can be found in the appendix. Each convolution matrix shows the values for each cross validation cycle with the test samples of the left out person.

5.2.1 Results for mono data

For the first scenario the normal pixelcount values are calculated from the mono information as input for classification with the Hidden Markov Models. In Table 5.2 the different parameters mentioned above and the recognition rates are shown. The best setting is used for the final classification configuration. It can be seen that the best result is located around 90% with a setting of 12 States (Q) and 8 Gaussian mixtures (M). In doing so so-called confusion matrices stating the whole cross validation were produced for the different parameters. In Table 5.3 the confusion matrix for the best parameters (Q=12, M=8) is illustrated. For the best test scenario the average duration of the training of the 8 HMMs (with around 521 samples) was 238 sec and the average duration for classification of one sample was 0.24 sec.

Gaussian Mixtures	States			
	8	10	12	14
4	87.70	89.28	88.90	88.36
6	89.40	88.86	89.82	90.53
8	88.62	88.99	90.66	89.97

Table 5.2: Accuracy for mono data

Activities	A1	A2	A3	A4	A5	A6	A7	A8	Recognition rate in %
A1	64	8	0	0	0	0	0	5	83.12
A2	0	76	1	0	0	0	0	4	93.83
A3	0	1	65	0	0	0	0	6	90.28
A4	0	0	0	75	0	0	1	4	93.75
A5	0	0	0	0	27	0	1	1	93.10
A6	1	1	0	0	0	85	0	3	94.44
A7	0	0	0	0	0	0	82	1	98.80
A8	0	1	6	0	7	0	1	53	77.94

Table 5.3: Full evaluation matrix for mono data with Q=12, M=8

5.2.2 Results for overlay data

In this scenario the normal pixelcount values from the overlay data are used as input for classification with the Hidden Markov Models. Table 5.4 states again the used parameters and the recognition rates whereby the best setting is used for the final classification configuration. It is obvious that 10 states and 6 Gaussian mixtures delivered the best result. Again so-called confusion matrices for the whole cross validation were produced for the different parameters. In Table 5.5 the confusion matrix for the best parameters (Q=10, M=6) is illustrated. The average duration of the training of the 8 HMMs (with around 521 samples) was 192 sec and the average duration for classification of one sample was 0.16 sec.

Gaussian Mixtures	States			
	8	10	12	14
4	97.11	96.46	96.29	96.36
6	96.99	97.33	96.85	96.97
8	90.97	97.32	96.70	96.83

Table 5.4: Accuracy for overlay data

Activities	A1	A2	A3	A4	A5	A6	A7	A8	Recognition rate in %
A1	77	0	0	0	0	0	0	0	100.00
A2	0	80	0	0	0	0	0	1	98.77
A3	0	2	66	0	0	0	0	4	91.67
A4	0	0	0	78	0	0	0	2	97.50
A5	0	0	0	0	29	0	0	0	100.00
A6	0	1	1	0	0	86	2	0	95.56
A7	0	0	0	0	1	0	79	3	95.18
A8	0	0	0	0	0	0	0	68	100.00

Table 5.5: Full evaluation matrix for overlay data with Q=10, M=6

5.2.3 Results for stereo data

For the third scenario in addition to the pixelcount a relative depth or rather disparity information from stereo data is used as input for classification with the Hidden Markov Models. Again for the different parameters table containing the recognition rates was generated to show the overall classification rate. The results for this validation can be seen in Table 5.6. It can be seen that with the settings Q=10, M=6 the best results are achieved. A full confusion matrix with the detailed recognition rates are shown in Table 5.7. For the best test scenario the average duration of the training of the 8 HMMs (with around 521 samples) was 190 sec and the average duration for classification of 1 sample was 0.15 sec.

Gaussian Mixtures	States			
	8	10	12	14
4	94.26	94.70	95.61	94.44
6	94.85	95.65	95.63	94.65
8	94.69	95.60	95.44	94.58

Table 5.6: Accuracy for stereo data

Activities	A1	A2	A3	A4	A5	A6	A7	A8	Recognition rate in %
A1	77	0	0	0	0	0	0	0	100.00
A2	0	76	0	0	0	0	1	4	93.83
A3	0	2	66	0	0	0	0	4	91.67
A4	0	1	0	73	0	0	2	4	91.25
A5	0	1	0	0	28	0	0	0	96.55
A6	0	1	0	0	0	86	0	3	95.56
A7	0	0	0	0	1	0	80	2	96.39
A8	0	0	0	0	0	0	0	68	100.00

Table 5.7: Full evaluation matrix for stereo data with Q=10, M=6

5.2.4 Discussion of results

From the executed scenarios it can be seen that all three results are higher than 88%. The most stable scenarios are the ones based on the data of both sensors. Both the overlay data as well as the stereo data deliver good steady features for the classification system. They delivered classification rates of around 94-97%. From the matrices it can be seen that the overlay data reaches slightly better results, which may be caused by the higher amount of activated pixels. On the other hand more stability is enabled with the additional use of the disparity information from the stereo data. This is because the constraint of the distance of the person to the sensor is eliminated. Therefore the usage of the depth information will be the preferred data for feature extraction and implementations.

In addition to the recognition rates, the results also delivered time performance benchmarks usable for an embedded system implementation. From the time stated in the sections above it can be seen that stereo and overlay data results are the two with the fastest classification time. This results from the number of states and mixtures that are used, as the complexity increases with the number of theses parameters. Therefore it can be seen that with the advantage of the slightly better performance the stereo data delivers another criteria for real-time usage with a classification rate almost as high as the one with the overlay data.

5.3 Comparison to related work

As shown in the previous section, with the used feature vectors and the HMM classification algorithm, good results are reached. In the best case the overall correct classification rate is around 97% and a performance as good as or better than other motion detection approaches is reached. Additionally, an integration into an embedded system can be reached. In general no approaches with exactly the same application have been implemented so far as the database and the activities have been generated especially for this system. To compare the recognition results with other approaches the KTH database with different motions such as walking, running, boxing can be used. In Table 5.8 a comparison to related approaches using the KTH database or similar actions is visualized. It has to be noted that not only experiments with HMMs but also other techniques are used for comparison. It can be seen that this approach with even more complex movements compared to the KTH database reaches an equivalent or slightly better result than the other approaches. Although these are not the same databases for a comparison it can be seen that for human motion detection the results around 96% are already state-of-the-art and promising for future implementation on an embedded system. It may be also noted that recognition was based on a first implementation and so the rate may be enhanced by further feature analysis.

Method	DB	Author/Approach	Average recognition rate in %
SVM	KTH	Schüldt et al. [SLC04]	71.60
	KTH	Meng et al. [MPB07]	80.30
HMM	Dance	This approach (overlay)	97.00
	Dance	This approach (stereo)	96.00
	Game	Wang et al. [WYSL08]	92.00
	KTH	Yamato et al. [YOI92]	96.00
	KTH	Mendoza and Pérez de la Blanca [AMPdlB07]	95.59
	KTH	Chakraborty and González [CRG08]	79.50

Table 5.8: Comparison to related approaches in the field of human motion detection with different methods

5.4 Open Issues

Many components for the recognition system have been implemented so far and have shown good results and good performance. Still, for further and practical usage some open issues may be implemented. Basically the recognition and the extraction of the features were based on the whole frame in this approach. In the recording session the individuals started from almost the same position. Up to now, how the starting point of the person influences the recognition rate has not been investigated. However, an influence can certainly be expected. As information is only generated when the person is moving in front of the sensor, non-moving parts are not visible and so an algorithm for a good selection of a bounding box around the human body may be implemented. With this implementation such inconstancies in position of the person may be removed. Of course, a system can be integrated to tell the person in front of the sensor to move to the correct position, but the bounding box may lead to a more pleasant approach for the participants.

A general open issue is a live implementation of this approach so that the detection can be tested directly during the exercises. As the workflow of the activities during a dance is known the log likelihood can be used to give a sort of feedback about the executed activity. First tests showed that the log likelihood of well-fitting activities stays in a certain range of likelihood. Therefore, this knowledge can help further analysis. In doing so, whether a good range can be found for giving useful feedback to the users should be evaluated.

CHAPTER 6

Conclusion and future work

Many different approaches concerning the recognition of human motion have been developed. Most of these implementations covered detection of motions like walking, running, waving and so on, or motions during sports as in the example of tennis strokes. In this approach a basic framework in MATLAB for human dance and fitness training, using a novel event-driven 3D vision sensor, is presented. In order to gain first test results, the recognition system was designed with the help of Hidden Markov Models, which were trained to each of the different activities. The test results delivered good performance within an execution of a leave-one(-person)-out cross validation. This demonstrates the practicability of further developments. Research of the literature showed that the usage of trivial features results not only in a high classification rate but also in a promising performance. By keeping this in mind, elementary features were used to save computational power and to assure usage on a system with lower power. In doing so experimentations showed that real-time usage is possible as the classification time is fast enough.

For future implementations the use of a bounding box around the human body will guarantee more robust features calculation. To save even more computational power a down sampling of the relatively high resolution may be used for further applications. Based on this approach live detection can be implemented and with a configuration of log likelihood ranges a sort of feedback about the activity can be given. All in all with the implemented system using Hidden Markov Models, the opportunity for state-of-the-art classification results and more detailed motion analysis is provided. Therefore, it showed that it and can be used within the EU project Silvergame [SJSB09].

Appendix

Evaluation matrices for mono data (Q=12, M=8, numb_it=5)

Evaluation matrix (Testing with Person 1)

Activities	A1	A2	A3	A4	A5	A6	A7	A8	Recognition rate in %
A1	0	8	0	0	0	0	0	5	0.00
A2	0	13	0	0	0	0	0	0	100.00
A3	0	0	12	0	0	0	0	1	92.31
A4	0	0	0	13	0	0	0	0	100.00
A5	0	0	0	0	2	0	0	0	100.00
A6	0	0	0	0	0	13	0	0	100.00
A7	0	0	0	0	0	0	13	0	100.00
A8	0	0	0	0	0	0	0	13	100.00

Evaluation matrix (Testing with Person 2)

Activities	A1	A2	A3	A4	A5	A6	A7	A8	Recognition rate in %
A1	13	0	0	0	0	0	0	0	100.00
A2	0	13	0	0	0	0	0	0	100.00
A3	0	0	13	0	0	0	0	0	100.00
A4	0	0	0	13	0	0	0	0	100.00
A5	0	0	0	0	2	0	0	0	100.00
A6	0	0	0	0	0	13	0	0	100.00
A7	0	0	0	0	0	0	13	0	100.00
A8	0	0	4	0	7	0	0	1	83.33

Evaluation matrix (Testing with Person 3)

Activities	A1	A2	A3	A4	A5	A6	A7	A8	Recognition rate in %
A1	3	0	0	0	0	0	0	0	100.00
A2	0	4	0	0	0	0	0	0	100.00
A3	0	0	3	0	0	0	0	0	100.00
A4	0	0	0	4	0	0	0	0	100.00
A5	0	0	0	0	2	0	0	0	100.00
A6	0	0	0	0	0	5	0	0	100.00
A7	0	0	0	0	0	0	4	0	100.00
A8	0	0	0	0	0	0	0	3	100.00

Evaluation matrix (Testing with Person 4)

Activities	A1	A2	A3	A4	A5	A6	A7	A8	Recognition rate in %
A1	3	0	0	0	0	0	0	0	100.00
A2	0	3	0	0	0	0	0	0	100.00
A3	0	0	3	0	0	0	0	0	100.00
A4	0	0	0	5	0	0	0	0	100.00
A5	0	0	0	0	2	0	0	0	100.00
A6	0	0	0	0	0	5	0	0	100.00
A7	0	0	0	0	0	0	4	0	100.00
A8	0	0	0	0	0	0	0	0	NaN

Evaluation matrix (Testing with Person 5)

Activities	A1	A2	A3	A4	A5	A6	A7	A8	Recognition rate in %
A1	0	0	0	0	0	0	0	0	NaN
A2	0	0	1	0	0	0	0	3	0.00
A3	0	0	0	0	0	0	0	1	0.00
A4	0	0	0	0	0	0	0	0	NaN
A5	0	0	0	0	1	0	0	1	50.00
A6	1	1	0	0	0	0	0	3	0.00
A7	0	0	0	0	0	0	1	1	50.00
A8	0	1	2	0	0	0	0	1	25.00

Evaluation matrix (Testing with Person 6)

Activities	A1	A2	A3	A4	A5	A6	A7	A8	Recognition rate in %
A1	2	0	0	0	0	0	0	0	100.00
A2	0	4	0	0	0	0	0	0	100.00
A3	0	0	2	0	0	0	0	0	100.00
A4	0	0	0	5	0	0	0	0	100.00
A5	0	0	0	0	2	0	0	0	100.00
A6	0	0	0	0	0	5	0	0	100.00
A7	0	0	0	0	0	0	4	0	100.00
A8	0	0	0	0	0	0	0	4	100.00

Evaluation matrix (Testing with Person 7)

Activities	A1	A2	A3	A4	A5	A6	A7	A8	Recognition rate in %
A1	5	0	0	0	0	0	0	0	100.00
A2	0	5	0	0	0	0	0	0	100.00
A3	0	0	5	0	0	0	0	0	100.00
A4	0	0	0	5	0	0	0	0	100.00
A5	0	0	0	0	2	0	0	0	100.00
A6	0	0	0	0	0	5	0	0	100.00
A7	0	0	0	0	0	0	5	0	100.00
A8	0	0	0	0	0	0	0	5	100.00

Evaluation matrix (Testing with Person 8)

Activities	A1	A2	A3	A4	A5	A6	A7	A8	Recognition rate in %
A1	5	0	0	0	0	0	0	0	100.00
A2	0	4	0	0	0	0	0	0	100.00
A3	0	0	4	0	0	0	0	0	100.00
A4	0	0	0	5	0	0	0	0	100.00
A5	0	0	0	0	2	0	0	0	100.00
A6	0	0	0	0	0	4	0	0	100.00
A7	0	0	0	0	0	0	4	0	100.00
A8	0	0	0	0	0	0	0	4	100.00

Evaluation matrix (Testing with Person 9)

Activities	A1	A2	A3	A4	A5	A6	A7	A8	Recognition rate in %
A1	5	0	0	0	0	0	0	0	100.00
A2	0	4	0	0	0	0	0	0	100.00
A3	0	0	4	0	0	0	0	0	100.00
A4	0	0	0	4	0	0	0	0	100.00
A5	0	0	0	0	1	0	1	0	50.00
A6	0	0	0	0	0	5	0	0	100.00
A7	0	0	0	0	0	0	4	0	100.00
A8	0	0	0	0	0	0	1	2	66.67

Evaluation matrix (Testing with Person 10)

Activities	A1	A2	A3	A4	A5	A6	A7	A8	Recognition rate in %
A1	5	0	0	0	0	0	0	0	100.00
A2	0	4	0	0	0	0	0	0	100.00
A3	0	1	4	0	0	0	0	0	80.00
A4	0	0	0	5	0	0	0	0	100.00
A5	0	0	0	0	2	0	0	0	100.00
A6	0	0	0	0	0	5	0	0	100.00
A7	0	0	0	0	0	0	5	0	100.00
A8	0	0	0	0	0	0	0	1	100.00

Evaluation matrix (Testing with Person 11)

Activities	A1	A2	A3	A4	A5	A6	A7	A8	Recognition rate in %
A1	5	0	0	0	0	0	0	0	100.00
A2	0	4	0	0	0	0	0	0	100.00
A3	0	0	1	0	0	0	0	3	25.00
A4	0	0	0	1	0	0	0	4	20.00
A5	0	0	0	0	2	0	0	0	100.00
A6	0	0	0	0	0	5	0	0	100.00
A7	0	0	0	0	0	0	5	0	100.00
A8	0	0	0	0	0	0	0	5	100.00

Evaluation matrix (Testing with Person 12)

Activities	A1	A2	A3	A4	A5	A6	A7	A8	Recognition rate in %
A1	4	0	0	0	0	0	0	0	100.00
A2	0	5	0	0	0	0	0	0	100.00
A3	0	0	4	0	0	0	0	0	100.00
A4	0	0	0	5	0	0	0	0	100.00
A5	0	0	0	0	2	0	0	0	100.00
A6	0	0	0	0	0	5	0	0	100.00
A7	0	0	0	0	0	0	5	0	100.00
A8	0	0	0	0	0	0	0	5	100.00

Evaluation matrix (Testing with Person 13)

Activities	A1	A2	A3	A4	A5	A6	A7	A8	Recognition rate in %
A1	5	0	0	0	0	0	0	0	100.00
A2	0	5	0	0	0	0	0	0	100.00
A3	0	0	2	0	0	0	0	0	100.00
A4	0	0	0	1	0	0	1	0	50.00
A5	0	0	0	0	2	0	0	0	100.00
A6	0	0	0	0	0	5	0	0	100.00
A7	0	0	0	0	0	0	5	0	100.00
A8	0	0	0	0	0	0	0	5	100.00

Evaluation matrix (Testing with Person 14)

Activities	A1	A2	A3	A4	A5	A6	A7	A8	Recognition rate in %
A1	4	0	0	0	0	0	0	0	100.00
A2	0	3	0	0	0	0	0	1	75.00
A3	0	0	4	0	0	0	0	0	100.00
A4	0	0	0	4	0	0	0	0	100.00
A5	0	0	0	0	1	0	0	0	100.00
A6	0	0	0	0	0	5	0	0	100.00
A7	0	0	0	0	0	0	5	0	100.00
A8	0	0	0	0	0	0	0	0	NaN

Evaluation matrix (Testing with Person 15)

Activities	A1	A2	A3	A4	A5	A6	A7	A8	Recognition rate in %
A1	5	0	0	0	0	0	0	0	100.00
A2	0	5	0	0	0	0	0	0	100.00
A3	0	0	4	0	0	0	0	1	80.00
A4	0	0	0	5	0	0	0	0	100.00
A5	0	0	0	0	2	0	0	0	100.00
A6	0	0	0	0	0	5	0	0	100.00
A7	0	0	0	0	0	0	5	0	100.00
A8	0	0	0	0	0	0	0	4	100.00

Full evaluation matrix (Testing with all Individuals)

Activities	A1	A2	A3	A4	A5	A6	A7	A8	Recognition rate in %
A1	64	8	0	0	0	0	0	5	83.12
A2	0	76	1	0	0	0	0	4	93.83
A3	0	1	65	0	0	0	0	6	90.28
A4	0	0	0	75	0	0	1	4	93.75
A5	0	0	0	0	27	0	1	1	93.10
A6	1	1	0	0	0	85	0	3	94.44
A7	0	0	0	0	0	0	82	1	98.80
A8	0	1	6	0	7	0	1	53	77.94

Evaluation matrices for overlay data (Q=10, M=6, numb_it=5)

Evaluation matrix (Testing with Person 1)

Activities	A1	A2	A3	A4	A5	A6	A7	A8	Recognition rate in %
A1	13	0	0	0	0	0	0	0	100.00
A2	0	13	0	0	0	0	0	0	100.00
A3	0	0	13	0	0	0	0	0	100.00
A4	0	0	0	13	0	0	0	0	100.00
A5	0	0	0	0	2	0	0	0	100.00
A6	0	0	0	0	0	13	0	0	100.00
A7	0	0	0	0	0	0	12	1	92.31
A8	0	0	0	0	0	0	0	13	100.00

Evaluation matrix (Testing with Person 2)

Activities	A1	A2	A3	A4	A5	A6	A7	A8	Recognition rate in %
A1	13	0	0	0	0	0	0	0	100.00
A2	0	13	0	0	0	0	0	0	100.00
A3	0	1	12	0	0	0	0	0	92.31
A4	0	0	0	13	0	0	0	0	100.00
A5	0	0	0	0	2	0	0	0	100.00
A6	0	0	0	0	0	13	0	0	100.00
A7	0	0	0	0	1	0	12	0	92.31
A8	0	0	0	0	0	0	0	12	100.00

Evaluation matrix (Testing with Person 3)

Activities	A1	A2	A3	A4	A5	A6	A7	A8	Recognition rate in %
A1	3	0	0	0	0	0	0	0	100.00
A2	0	4	0	0	0	0	0	0	100.00
A3	0	0	3	0	0	0	0	0	100.00
A4	0	0	0	4	0	0	0	0	100.00
A5	0	0	0	0	2	0	0	0	100.00
A6	0	0	0	0	0	5	0	0	100.00
A7	0	0	0	0	0	0	3	1	75.00
A8	0	0	0	0	0	0	0	3	100.00

Evaluation matrix (Testing with Person 4)

Activities	A1	A2	A3	A4	A5	A6	A7	A8	Recognition rate in %
A1	3	0	0	0	0	0	0	0	100.00
A2	0	3	0	0	0	0	0	0	100.00
A3	0	0	3	0	0	0	0	0	100.00
A4	0	0	0	5	0	0	0	0	100.00
A5	0	0	0	0	2	0	0	0	100.00
A6	0	0	0	0	0	5	0	0	100.00
A7	0	0	0	0	0	0	4	0	100.00
A8	0	0	0	0	0	0	0	0	NaN

Evaluation matrix (Testing with Person 5)

Activities	A1	A2	A3	A4	A5	A6	A7	A8	Recognition rate in %
A1	0	0	0	0	0	0	0	0	NaN
A2	0	4	0	0	0	0	0	0	100.00
A3	0	0	0	0	0	0	0	1	0.00
A4	0	0	0	0	0	0	0	0	NaN
A5	0	0	0	0	2	0	0	0	100.00
A6	0	1	1	0	0	1	2	0	20.00
A7	0	0	0	0	0	0	1	1	50.00
A8	0	0	0	0	0	0	0	4	100.00

Evaluation matrix (Testing with Person 6)

Activities	A1	A2	A3	A4	A5	A6	A7	A8	Recognition rate in %
A1	2	0	0	0	0	0	0	0	100.00
A2	0	4	0	0	0	0	0	0	100.00
A3	0	0	2	0	0	0	0	0	100.00
A4	0	0	0	5	0	0	0	0	100.00
A5	0	0	0	0	2	0	0	0	100.00
A6	0	0	0	0	0	5	0	0	100.00
A7	0	0	0	0	0	0	4	0	100.00
A8	0	0	0	0	0	0	0	4	100.00

Evaluation matrix (Testing with Person 7)

Activities	A1	A2	A3	A4	A5	A6	A7	A8	Recognition rate in %
A1	5	0	0	0	0	0	0	0	100.00
A2	0	5	0	0	0	0	0	0	100.00
A3	0	0	5	0	0	0	0	0	100.00
A4	0	0	0	5	0	0	0	0	100.00
A5	0	0	0	0	2	0	0	0	100.00
A6	0	0	0	0	0	5	0	0	100.00
A7	0	0	0	0	0	0	5	0	100.00
A8	0	0	0	0	0	0	0	5	100.00

Evaluation matrix (Testing with Person 8)

Activities	A1	A2	A3	A4	A5	A6	A7	A8	Recognition rate in %
A1	5	0	0	0	0	0	0	0	100.00
A2	0	4	0	0	0	0	0	0	100.00
A3	0	0	4	0	0	0	0	0	100.00
A4	0	0	0	5	0	0	0	0	100.00
A5	0	0	0	0	2	0	0	0	100.00
A6	0	0	0	0	0	4	0	0	100.00
A7	0	0	0	0	0	0	4	0	100.00
A8	0	0	0	0	0	0	0	4	100.00

Evaluation matrix (Testing with Person 9)

Activities	A1	A2	A3	A4	A5	A6	A7	A8	Recognition rate in %
A1	5	0	0	0	0	0	0	0	100.00
A2	0	4	0	0	0	0	0	0	100.00
A3	0	0	4	0	0	0	0	0	100.00
A4	0	0	0	4	0	0	0	0	100.00
A5	0	0	0	0	2	0	0	0	100.00
A6	0	0	0	0	0	5	0	0	100.00
A7	0	0	0	0	0	0	4	0	100.00
A8	0	0	0	0	0	0	0	3	100.00

Evaluation matrix (Testing with Person 10)

Activities	A1	A2	A3	A4	A5	A6	A7	A8	Recognition rate in %
A1	5	0	0	0	0	0	0	0	100.00
A2	0	4	0	0	0	0	0	0	100.00
A3	0	1	4	0	0	0	0	0	80.00
A4	0	0	0	5	0	0	0	0	100.00
A5	0	0	0	0	2	0	0	0	100.00
A6	0	0	0	0	0	5	0	0	100.00
A7	0	0	0	0	0	0	5	0	100.00
A8	0	0	0	0	0	0	0	1	100.00

Evaluation matrix (Testing with Person 11)

Activities	A1	A2	A3	A4	A5	A6	A7	A8	Recognition rate in %
A1	5	0	0	0	0	0	0	0	100.00
A2	0	4	0	0	0	0	0	0	100.00
A3	0	0	4	0	0	0	0	0	100.00
A4	0	0	0	4	0	0	0	1	80.00
A5	0	0	0	0	2	0	0	0	100.00
A6	0	0	0	0	0	5	0	0	100.00
A7	0	0	0	0	0	0	5	0	100.00
A8	0	0	0	0	0	0	0	5	100.00

Evaluation matrix (Testing with Person 12)

Activities	A1	A2	A3	A4	A5	A6	A7	A8	Recognition rate in %
A1	4	0	0	0	0	0	0	0	100.00
A2	0	5	0	0	0	0	0	0	100.00
A3	0	0	4	0	0	0	0	0	100.00
A4	0	0	0	5	0	0	0	0	100.00
A5	0	0	0	0	2	0	0	0	100.00
A6	0	0	0	0	0	5	0	0	100.00
A7	0	0	0	0	0	0	5	0	100.00
A8	0	0	0	0	0	0	0	5	100.00

Evaluation matrix (Testing with Person 13)

Activities	A1	A2	A3	A4	A5	A6	A7	A8	Recognition rate in %
A1	5	0	0	0	0	0	0	0	100.00
A2	0	5	0	0	0	0	0	0	100.00
A3	0	0	2	0	0	0	0	0	100.00
A4	0	0	0	1	0	0	0	1	50.00
A5	0	0	0	0	2	0	0	0	100.00
A6	0	0	0	0	0	5	0	0	100.00
A7	0	0	0	0	0	0	5	0	100.00
A8	0	0	0	0	0	0	0	5	100.00

Evaluation matrix (Testing with Person 14)

Activities	A1	A2	A3	A4	A5	A6	A7	A8	Recognition rate in %
A1	4	0	0	0	0	0	0	0	100.00
A2	0	3	0	0	0	0	0	1	75.00
A3	0	0	4	0	0	0	0	0	100.00
A4	0	0	0	4	0	0	0	0	100.00
A5	0	0	0	0	1	0	0	0	100.00
A6	0	0	0	0	0	5	0	0	100.00
A7	0	0	0	0	0	0	5	0	100.00
A8	0	0	0	0	0	0	0	0	NaN

Evaluation matrix (Testing with Person 15)

Activities	A1	A2	A3	A4	A5	A6	A7	A8	Recognition rate in %
A1	5	0	0	0	0	0	0	0	100.00
A2	0	5	0	0	0	0	0	0	100.00
A3	0	0	2	0	0	0	0	3	40.00
A4	0	0	0	5	0	0	0	0	100.00
A5	0	0	0	0	2	0	0	0	100.00
A6	0	0	0	0	0	5	0	0	100.00
A7	0	0	0	0	0	0	5	0	100.00
A8	0	0	0	0	0	0	0	4	100.00

Full evaluation matrix (Testing with all Individuals)

Activities	A1	A2	A3	A4	A5	A6	A7	A8	Recognition rate in %
A1	77	0	0	0	0	0	0	0	100.00
A2	0	80	0	0	0	0	0	1	98.77
A3	0	2	66	0	0	0	0	4	91.67
A4	0	0	0	78	0	0	0	2	97.50
A5	0	0	0	0	29	0	0	0	100.00
A6	0	1	1	0	0	86	2	0	95.56
A7	0	0	0	0	1	0	79	3	95.18
A8	0	0	0	0	0	0	0	68	100.00

Evaluation matrices for stereo data (Q=10, M=6, numb_it=5)

Evaluation matrix (Testing with Person 1)

Activities	A1	A2	A3	A4	A5	A6	A7	A8	Recognition rate in %
A1	13	0	0	0	0	0	0	0	100.00
A2	0	13	0	0	0	0	0	0	100.00
A3	0	0	12	0	0	0	0	1	92.31
A4	0	1	0	9	0	0	0	3	69.23
A5	0	1	0	0	1	0	0	0	50.00
A6	0	0	0	0	0	13	0	0	100.00
A7	0	0	0	0	0	0	13	0	100.00
A8	0	0	0	0	0	0	0	13	100.00

Evaluation matrix (Testing with Person 2)

Activities	A1	A2	A3	A4	A5	A6	A7	A8	Recognition rate in %
A1	13	0	0	0	0	0	0	0	100.00
A2	0	13	0	0	0	0	0	0	100.00
A3	0	0	13	0	0	0	0	0	100.00
A4	0	0	0	13	0	0	0	0	100.00
A5	0	0	0	0	2	0	0	0	100.00
A6	0	0	0	0	0	13	0	0	100.00
A7	0	0	0	0	1	0	12	0	92.31
A8	0	0	0	0	0	0	0	12	100.00

Evaluation matrix (Testing with Person 3)

Activities	A1	A2	A3	A4	A5	A6	A7	A8	Recognition rate in %
A1	3	0	0	0	0	0	0	0	100.00
A2	0	4	0	0	0	0	0	0	100.00
A3	0	0	3	0	0	0	0	0	100.00
A4	0	0	0	4	0	0	0	0	100.00
A5	0	0	0	0	2	0	0	0	100.00
A6	0	0	0	0	0	5	0	0	100.00
A7	0	0	0	0	0	0	3	1	75.00
A8	0	0	0	0	0	0	0	3	100.00

Evaluation matrix (Testing with Person 4)

Activities	A1	A2	A3	A4	A5	A6	A7	A8	Recognition rate in %
A1	3	0	0	0	0	0	0	0	100.00
A2	0	3	0	0	0	0	0	0	100.00
A3	0	0	3	0	0	0	0	0	100.00
A4	0	0	0	4	0	0	1	0	80.00
A5	0	0	0	0	2	0	0	0	100.00
A6	0	0	0	0	0	5	0	0	100.00
A7	0	0	0	0	0	0	4	0	100.00
A8	0	0	0	0	0	0	0	0	NaN

Evaluation matrix (Testing with Person 5)

Activities	A1	A2	A3	A4	A5	A6	A7	A8	Recognition rate in %
A1	0	0	0	0	0	0	0	0	NaN
A2	0	0	0	0	0	0	0	4	0.00
A3	0	0	0	0	0	0	0	1	0.00
A4	0	0	0	0	0	0	0	0	NaN
A5	0	0	0	0	2	0	0	0	100.00
A6	0	1	0	0	0	1	0	3	20.00
A7	0	0	0	0	0	0	1	1	50.00
A8	0	0	0	0	0	0	0	4	100.00

Evaluation matrix (Testing with Person 6)

Activities	A1	A2	A3	A4	A5	A6	A7	A8	Recognition rate in %
A1	2	0	0	0	0	0	0	0	100.00
A2	0	4	0	0	0	0	0	0	100.00
A3	0	0	2	0	0	0	0	0	100.00
A4	0	0	0	5	0	0	0	0	100.00
A5	0	0	0	0	2	0	0	0	100.00
A6	0	0	0	0	0	5	0	0	100.00
A7	0	0	0	0	0	0	4	0	100.00
A8	0	0	0	0	0	0	0	4	100.00

Evaluation matrix (Testing with Person 7)

Activities	A1	A2	A3	A4	A5	A6	A7	A8	Recognition rate in %
A1	5	0	0	0	0	0	0	0	100.00
A2	0	5	0	0	0	0	0	0	100.00
A3	0	0	5	0	0	0	0	0	100.00
A4	0	0	0	5	0	0	0	0	100.00
A5	0	0	0	0	2	0	0	0	100.00
A6	0	0	0	0	0	5	0	0	100.00
A7	0	0	0	0	0	0	5	0	100.00
A8	0	0	0	0	0	0	0	5	100.00

Evaluation matrix (Testing with Person 8)

Activities	A1	A2	A3	A4	A5	A6	A7	A8	Recognition rate in %
A1	5	0	0	0	0	0	0	0	100.00
A2	0	4	0	0	0	0	0	0	100.00
A3	0	0	4	0	0	0	0	0	100.00
A4	0	0	0	5	0	0	0	0	100.00
A5	0	0	0	0	2	0	0	0	100.00
A6	0	0	0	0	0	4	0	0	100.00
A7	0	0	0	0	0	0	4	0	100.00
A8	0	0	0	0	0	0	0	4	100.00

Evaluation matrix (Testing with Person 9)

Activities	A1	A2	A3	A4	A5	A6	A7	A8	Recognition rate in %
A1	5	0	0	0	0	0	0	0	100.00
A2	0	4	0	0	0	0	0	0	100.00
A3	0	0	4	0	0	0	0	0	100.00
A4	0	0	0	4	0	0	0	0	100.00
A5	0	0	0	0	2	0	0	0	100.00
A6	0	0	0	0	0	5	0	0	100.00
A7	0	0	0	0	0	0	4	0	100.00
A8	0	0	0	0	0	0	0	3	100.00

Evaluation matrix (Testing with Person 10)

Activities	A1	A2	A3	A4	A5	A6	A7	A8	Recognition rate in %
A1	5	0	0	0	0	0	0	0	100.00
A2	0	4	0	0	0	0	0	0	100.00
A3	0	0	5	0	0	0	0	0	100.00
A4	0	0	0	5	0	0	0	0	100.00
A5	0	0	0	0	2	0	0	0	100.00
A6	0	0	0	0	0	5	0	0	100.00
A7	0	0	0	0	0	0	5	0	100.00
A8	0	0	0	0	0	0	0	1	100.00

Evaluation matrix (Testing with Person 11)

Activities	A1	A2	A3	A4	A5	A6	A7	A8	Recognition rate in %
A1	5	0	0	0	0	0	0	0	100.00
A2	0	4	0	0	0	0	0	0	100.00
A3	0	0	3	0	0	0	0	1	75.00
A4	0	0	0	5	0	0	0	0	100.00
A5	0	0	0	0	2	0	0	0	100.00
A6	0	0	0	0	0	5	0	0	100.00
A7	0	0	0	0	0	0	5	0	100.00
A8	0	0	0	0	0	0	0	5	100.00

Evaluation matrix (Testing with Person 12)

Activities	A1	A2	A3	A4	A5	A6	A7	A8	Recognition rate in %
A1	4	0	0	0	0	0	0	0	100.00
A2	0	5	0	0	0	0	0	0	100.00
A3	0	0	4	0	0	0	0	0	100.00
A4	0	0	0	5	0	0	0	0	100.00
A5	0	0	0	0	2	0	0	0	100.00
A6	0	0	0	0	0	5	0	0	100.00
A7	0	0	0	0	0	0	5	0	100.00
A8	0	0	0	0	0	0	0	5	100.00

Evaluation matrix (Testing with Person 13)

Activities	A1	A2	A3	A4	A5	A6	A7	A8	Recognition rate in %
A1	5	0	0	0	0	0	0	0	100.00
A2	0	5	0	0	0	0	0	0	100.00
A3	0	0	2	0	0	0	0	0	100.00
A4	0	0	0	1	0	0	0	1	50.00
A5	0	0	0	0	2	0	0	0	100.00
A6	0	0	0	0	0	5	0	0	100.00
A7	0	0	0	0	0	0	5	0	100.00
A8	0	0	0	0	0	0	0	5	100.00

Evaluation matrix (Testing with Person 14)

Activities	A1	A2	A3	A4	A5	A6	A7	A8	Recognition rate in %
A1	4	0	0	0	0	0	0	0	100.00
A2	0	3	0	0	0	0	1	0	75.00
A3	0	0	4	0	0	0	0	0	100.00
A4	0	0	0	4	0	0	0	0	100.00
A5	0	0	0	0	1	0	0	0	100.00
A6	0	0	0	0	0	5	0	0	100.00
A7	0	0	0	0	0	0	5	0	100.00
A8	0	0	0	0	0	0	0	0	NaN

Evaluation matrix (Testing with Person 15)

Activities	A1	A2	A3	A4	A5	A6	A7	A8	Recognition rate in %
A1	5	0	0	0	0	0	0	0	100.00
A2	0	5	0	0	0	0	0	0	100.00
A3	0	2	2	0	0	0	0	1	40.00
A4	0	0	0	4	0	0	1	0	80.00
A5	0	0	0	0	2	0	0	0	100.00
A6	0	0	0	0	0	5	0	0	100.00
A7	0	0	0	0	0	0	5	0	100.00
A8	0	0	0	0	0	0	0	4	100.00

Full evaluation matrix (Testing with all Individuals)

Activities	A1	A2	A3	A4	A5	A6	A7	A8	Recognition rate in %
A1	77	0	0	0	0	0	0	0	100.00
A2	0	76	0	0	0	0	1	4	93.83
A3	0	2	66	0	0	0	0	4	91.67
A4	0	1	0	73	0	0	2	4	91.25
A5	0	1	0	0	28	0	0	0	96.55
A6	0	1	0	0	0	86	0	3	95.56
A7	0	0	0	0	1	0	80	2	96.39
A8	0	0	0	0	0	0	0	68	100.00

Bibliography

[AMPdlB07] Maria Ángeles Mendoza and Nicolás Pérez de la Blanca. Hmm-based action recognition using contour histograms. In Joan Martí, José Benedí, Ana Mendonça, and Joan Serrat, editors, *Pattern Recognition and Image Analysis*, volume 4477 of *Lecture Notes in Computer Science*, pages 394–401. Springer Berlin / Heidelberg, 2007. 10.1007/978-3-540-72847-4_51.

[BMM06] Serge Belongie, Greg Mori, and Jitendra Malik. Matching with shape contexts. In Nicola Bellomo, Hamid Krim, and Anthony Yezzi, editors, *Statistics and Analysis of Shapes*, Modeling and Simulation in Science, Engineering and Technology, pages 81–105. Birkhäuser Boston, 2006. 10.1007/0-8176-4481-4_4.

[Bur98] Christopher J.C. Burges. A tutorial on support vector machines for pattern recognition. *Data Mining and Knowledge Discovery*, 2:121–167, 1998. accessed 14 June 2011.

[CRG08] Bhaskar Chakraborty, Ognjen Rudovic, and Jordi Gonzalez. View-invariant human-body detection with extension to human action recognition using component-wise hmm of body parts. In *Proc. 8th IEEE Int. Conf. Automatic Face & Gesture Recognition FG '08*, pages 1–6, 2008.

[CV95] Corinna Cortes and Vladimir Vapnik. Support-vector networks. *Machine Learning*, 20:273–297, 1995. 10.1007/BF00994018.

[DD96] Rakesh Dugad and U.B. Desai. A tutorial on hidden markov models. Technical Report SPANN-96.1, Signal Processing and Artificial Neural Networks Laboratory Department of Electrical Engineering Indian Institute of Technology - Bombay, Powai, Mumbai 400 076, India, May 1996.

[Dud04] Richard O. Duda. *Pattern Classification 2nd Edition with Computer Manual 2nd Edition Set*. John Wiley & Sons, 2004.

[iFi10] iFixit. Microsoft kinect teardown. http://www.ifixit.com/Teardown/Microsoft-Kinect-Teardown/4066/1, 4 November 2010. accessed 08 June 2011.

[Ima11] MESA Imaging. Swissranger^{TM} sr4000 overview. http://www.mesa-imaging.ch/prodview4k.php, 2011. accessed 07 June 2011.

[LN06] Fengjun Lv and Ramakant Nevatia. Recognition and segmentation of 3-d human action using hmm and multi-class adaboost. In Ale Leonardis, Horst Bischof, and Axel Pinz, editors, *Computer Vision ECCV 2006*, volume 3954 of *Lecture Notes in Computer Science*, pages 359–372. Springer Berlin / Heidelberg, 2006.

[LPD08] Patrick Lichtsteiner, Christoph Posch Posch, and Tobi Delbruck. A 128 x 128 120 db 15 μ s latency asynchronous temporal contrast vision sensor. *Solid-State Circuits, IEEE Journal of*, 43(2):566 –576, feb. 2008.

[LS01] Robert Lange and Peter Seitz. Solid-state time-of-flight range camera. *Quantum Electronics, IEEE Journal of*, 37(3):390 –397, mar 2001.

[MA07] Sushmita Mitra and Tinku Acharya. Gesture recognition: A survey. *Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on*, 37(3):311 –324, may 2007.

[Mic11] Microsoft. Willkommen bei kinect für xbox 360. http://www.xbox.com/de-DE/Kinect/, 2011. accessed 08 June 2011.

[MPB07] Hongying Meng, Nick Pears, and Chris Bailey. A human action recognition system for embedded computer vision application. In *Computer Vision and Pattern Recognition, 2007. CVPR '07. IEEE Conference on*, pages 1 –6, june 2007.

[MT91] Kouichi Murakami and Hitomi Taguchi. Gesture recognition using recurrent neural networks. In *Proceedings of the SIGCHI conference on Human factors in computing systems: Reaching through technology*, CHI '91, pages 237–242, New York, NY, USA, 1991. ACM.

[Mur05] Kevin Murphy. Hidden markov model (hmm) toolbox for matlab. http://www.mathworks.com/help/toolbox/stats/f8368.html, 8 June 2005. accessed 11 May 2011.

[oT11] AIT Austrian Institute of Technology. Atis - biomimetic, frame-free vision sensor. http://www.ait.ac.at/research-services/research-services-safety-security/new-sensor-technologies/chip-design-for-intelligent-optical-sensor-chips/atis-biomimetic-frame-free-vision-sensor/?L=1, 2011. accessed 08 June 2011.

[PMD11] PMDTechnologies. Pmd[vision]© camcube 3.0. `http://www.pmdtec.com/fileadmin/pmdtec/downloads/documentation/datenblatt_camcube3.pdf`, 2011. accessed 31 May 2011.

[PMW+10] Christoph Posch, Daniel Matolin, Rainer Wohlgenannt, Michael Hofstätter, Peter Schön, Martin Litzenberger, Daniel Bauer, and Heinrich Garn. Biomimetic frame-free hdr camera with event-driven pwm image/video sensor and full-custom address-event processor. In *Biomedical Circuits and Systems Conference (BioCAS), 2010 IEEE*, pages 254–257, November 2010. accessed 08 June 2011.

[Pos11] Christoph Posch. Next generation bio-inspired vision. *ERCIM NEWS*, 84:24–25, January 2011. accessed 31 May 2011.

[Rab89] Lawrence R. Rabiner. A tutorial on hidden markov models and selected applications in speech recognition. *IEEE_J_PROC*, 77(2):257–286, 1989.

[Sho10] Nikolai Shokhirev. Hidden markov models. http://www.shokhirev.com/nikolai/abc/alg/hmm/hmm.html, 15 February 2010. accessed 15 June 2011.

[SJSB09] Beate Seewald, Michael John, Joachim Senger, and Ahmed Nabil Belbachir. Silvergame - a project aimed at social integration and multimedia interaction for the elderly. 2009.

[SLC04] Christian Schüldt, Ivan Laptev, and Barbara Caputo. Recognizing human actions: a local svm approach. In *Pattern Recognition, 2004. ICPR 2004. Proceedings of the 17th International Conference on*, volume 3, pages 32 – 36 Vol.3, aug. 2004.

[SS] Christos Stergiou and Dimitrios Siganos. Neural networks. `http://www.doc.ic.ac.uk/~nd/surprise_96/journal/vol4/cs11/report.html`. accessed 14 June 2011.

[The10] Electronic Theatre. Kinect technical specifications & xbox 360 requirements revealed. http://electronictheatre.co.uk/index.php/xbox360/xbxo360-news/5162-kinect-technical-specifications-a-xbox-360-requirements-revealed, 29 June 2010. accessed 08 June 2011.

[WYSL08] Yong Wang, Tianli Yu, L. Shi, and Zhu Li. Using human body gestures as inputs for gaming via depth analysis. In *Multimedia and Expo, 2008 IEEE International Conference on*, pages 993 –996, 23 2008-april 26 2008.

[YOI92] Junji Yamato, Jun Ohya, and Kenichiro Ishii. Recognizing human action in time-sequential images using hidden markov model. In *Proc. CVPR '92. IEEE Computer Society Conf Computer Vision and Pattern Recognition*, pages 379–385, 1992.