**Русенски университет “Ангел Кънчев”**

Факултет: Природни науки и образование

Катедра: Информатика и информационни технологии

**ДОПУСКАМ ДО ЗАЩИТА**

Ръководител на специали-

зиращата катедра: ……………...

(доц. д-р Румен Русев)

**ОБЯСНИТЕЛНА ЗАПИСКА**

**НА ДИПЛОМЕН ПРОЕКТ**

**Тема**: **Motion capture data processing and analysis**

**(Обработване и анализ на данни от следене на движение)**

Дипломант: ………………… Ръководител: …………………

(Р. Русева) (доц. д-р Румен Русев)

Русе, 2018 г.

**Русенски университет “Ангел Кънчев”**

Катедра **“Информатика и информационни технологии”**

Регистрирал УТВЪРЖДАВАМ

Факултет: ПНО Р-л на специализи-

Инспектор: …………….. ращата катедра: …………

(M. Чолакова) (доц. д-р Румен Русев)

....../......, 2018 г.

**Задание за дипломен проект**

На студент: Радина Росенова Русева Фак.№ 146529

Специалност: Компютърни науки

ОКС: **бакалавър**

**1. Тема:** Motion capture data processing and analysis (Обработване и анализ на данни от следене на движение)

**2. Изходни данни:** Python

**3. Съдържание на обяснителната записка:**

* + Увод
  + Обзор и анализ на проблема. Обзор на съществуващите решения. Изводи. Цел и задачи.
  + Архитектурен модел на системата
  + Програмна реализация
  + Основни резултати. Изводи и препоръки.
  + Използвана литература

Срок за предаване на дипломния проект: септември 2018г.

Зададено на: февруари 2018 г.

Дипломант: ………………… Ръководител: …………………

(Р. Русева) (доц. д-р Румен Русев)

Table of Contents

[List of Abbreviations - 0 -](#_Toc524678919)

[1 Introduction - 1 -](#_Toc524678920)

[2 Theory. Analysis. Aim. - 2 -](#_Toc524678921)

[2.1 Motion Capture - 2 -](#_Toc524678922)

[1.1.1 History of motion capture - 2 -](#_Toc524678923)

[1.1.2 Optical-based motion capture system VICON - 4 -](#_Toc524678924)

[2.2 Sign language analysis - 6 -](#_Toc524678925)

[1.1.3 Hamburg Notation System - 7 -](#_Toc524678926)

[2.3 Optical-based motion capture for sign language synthesis - 13 -](#_Toc524678927)

[2.4 Segmentation of SL - 13 -](#_Toc524678928)

[1.1.4 Method for segmentation - 14 -](#_Toc524678929)

[2.5 Related work - 15 -](#_Toc524678930)

[1.1.5 ViSiCAST system - 15 -](#_Toc524678931)

[1.1.6 eSIGN editor - 17 -](#_Toc524678932)

[1.1.7 SignCom - 17 -](#_Toc524678933)

[1.1.8 Sign3D - 18 -](#_Toc524678934)

[1.1.9 Korean sign language - 18 -](#_Toc524678935)

[1.1.10 Data-driven Finger Motion Synthesis for Gesturing Characters - 19 -](#_Toc524678936)

[1.1.11 Mapping sound to human gesture - 19 -](#_Toc524678937)

[1.1.12 Segmenting Motion Capture Data Using a Qualitative Analysis - 20 -](#_Toc524678938)

[2.6 Conclusion - 21 -](#_Toc524678939)

[3 System Architecture - 22 -](#_Toc524678940)

[3.1 Python - 22 -](#_Toc524678941)

[3.2 Sublime Text code editor - 23 -](#_Toc524678942)

[3.3 C3D file format - 23 -](#_Toc524678943)

[3.4 Activity flow - 24 -](#_Toc524678944)

[1.1.13 Input data module - 25 -](#_Toc524678945)

[1.1.14 Segmentation module - 26 -](#_Toc524678946)

[1.1.15 Sign analysis module - 27 -](#_Toc524678947)

[3.5 Input data - 27 -](#_Toc524678948)

[3.6 Hand marker set - 28 -](#_Toc524678949)

[3.7 Segmentation - 29 -](#_Toc524678950)

[3.8 Co-ordinate system - 31 -](#_Toc524678951)

[3.9 Noise and methods for filtering - 32 -](#_Toc524678952)

[3.10 Sign properties - 35 -](#_Toc524678953)

[1.1.16 Hand Location - 36 -](#_Toc524678954)

[1.1.17 Hand Orientation - 36 -](#_Toc524678955)

[1.1.18 Handshape - 37 -](#_Toc524678956)

[3.11 SVM - 37 -](#_Toc524678957)

[3.12 Testing - 37 -](#_Toc524678958)

[4 System implementation - 39 -](#_Toc524678959)

[4.1 Input data module - 39 -](#_Toc524678960)

[1.1.19 File read - 39 -](#_Toc524678961)

[1.1.20 Co-ordinate system - 40 -](#_Toc524678962)

[1.1.21 More active hand - 40 -](#_Toc524678963)

[4.2 Hand marker set - 41 -](#_Toc524678964)

[4.3 Segmentation module - 42 -](#_Toc524678965)

[1.1.22 Hand velocity and acceleration - 42 -](#_Toc524678966)

[1.1.23 Zero-crossing - 43 -](#_Toc524678967)

[1.1.24 Labeling - 44 -](#_Toc524678968)

[1.1.25 Rest pose analysis - 46 -](#_Toc524678969)

[4.4 Sign analysis module - 46 -](#_Toc524678970)

[1.1.26 Real sign borders - 46 -](#_Toc524678971)

[1.1.27 Dominant hand - 47 -](#_Toc524678972)

[1.1.28 One-handed or two-handed sign - 47 -](#_Toc524678973)

[1.1.29 Hand location - 48 -](#_Toc524678974)

[4.5 Output - 50 -](#_Toc524678975)

[4.6 Functional testing - 53 -](#_Toc524678976)

[4.7 Method evaluation - 56 -](#_Toc524678977)

[5 Conclusion - 59 -](#_Toc524678978)

[6 Bibliography - 60 -](#_Toc524678979)

Table of Figures

[Figure 1: "The horse of motion" by Eadwerd Muybridge - 2 -](#_Toc524677833)

[Figure 2: General marker set for human body - 5 -](#_Toc524677834)

[Figure 3: General structure of HamNoSys notation - 8 -](#_Toc524677835)

[Figure 4: Handshapes - 9 -](file:///D:\Radi\MyProjects\MOCAP\Motion%20capture%20data%20processing%20and%20analysis.docx#_Toc524677836)

[Figure 5: Wrist bending - 9 -](file:///D:\Radi\MyProjects\MOCAP\Motion%20capture%20data%20processing%20and%20analysis.docx#_Toc524677837)

[Figure 6: Extended finger orientation (top) and Palm orientation (bottom) - 10 -](file:///D:\Radi\MyProjects\MOCAP\Motion%20capture%20data%20processing%20and%20analysis.docx#_Toc524677838)

[Figure 7: Body parts - 11 -](#_Toc524677839)

[Figure 8: Six zones of signing space and their symbols - 11 -](#_Toc524677840)

[Figure 9: General activity diagram - 25 -](file:///D:\Radi\MyProjects\MOCAP\Motion%20capture%20data%20processing%20and%20analysis.docx#_Toc524677841)

[Figure 10: Activity diagram for Input data module - 25 -](#_Toc524677842)

[Figure 11: Activity diagram for Segmentation module - 26 -](#_Toc524677843)

[Figure 12:Activity diagram for Sign analysis module - 27 -](#_Toc524677844)

[Figure 13: Ilustrates character in Rest-pose: on the left is the reconstructed image from MoCap data; on the right - real signer during recording - 28 -](file:///D:\Radi\MyProjects\MOCAP\Motion%20capture%20data%20processing%20and%20analysis.docx#_Toc524677845)

[Figure 14: Hand markers position on right hand - 28 -](file:///D:\Radi\MyProjects\MOCAP\Motion%20capture%20data%20processing%20and%20analysis.docx#_Toc524677846)

[Figure 15: Graph illustrating Hand displacement along the 3 axes with respect of time - 29 -](#_Toc524677847)

[Figure 16: Graph illustrating the Hand velocity with respect of time; The red, green, blue lines show the graphs for the 3 axes; the magenta line shows the normalized velocity - 30 -](#_Toc524677848)

[Figure 17: Graph illustrating Hand acceleration - 30 -](#_Toc524677849)

[Figure 18:Graph of Hand velocity (blue line) with marked points of interest and initial threshold (red line) - 31 -](#_Toc524677850)

[Figure 19: Division of frequency domain - 32 -](file:///D:\Radi\MyProjects\MOCAP\Motion%20capture%20data%20processing%20and%20analysis.docx#_Toc524677851)

[Figure 20: Graph of hand acceleration (blue line), resulting graph (magenta line) after applied Butterworth filter with Cutoff frequency = 2 - 33 -](#_Toc524677852)

[Figure 21: Graph of hand acceleration (blue line), resulting graph (magenta line) after applied Butterworth filter with Cutoff frequency = 4 - 34 -](#_Toc524677853)

[Figure 22: Graph of hand acceleration (blue line), resulting graph (magenta line) after applied Butterworth filter with Cutoff frequency = 10 - 34 -](#_Toc524677854)

[Figure 23: Graph of hand acceleration (blue line), resulting graph (magenta line) after applied Butterworth filter with Cutoff frequency = 12 - 35 -](#_Toc524677855)

[Figure 24: Example of hand location regions - 36 -](file:///D:\Radi\MyProjects\MOCAP\Motion%20capture%20data%20processing%20and%20analysis.docx#_Toc524677856)

[Figure 25: Points defining the hand and arm planes - 36 -](file:///D:\Radi\MyProjects\MOCAP\Motion%20capture%20data%20processing%20and%20analysis.docx#_Toc524677857)

[Figure 26: System output from raw segmentation before and after refining the threshold value. - 50 -](file:///D:\Radi\MyProjects\MOCAP\Motion%20capture%20data%20processing%20and%20analysis.docx#_Toc524677858)

[Figure 27: Part of system output for sign description. - 51 -](#_Toc524677859)

[Figure 28:System output plot of Hand velocity and acceleration for second found sign - 52 -](#_Toc524677860)

[Figure 29: System output plot of right (in red) and left (in green) hand location - 52 -](#_Toc524677861)

[Table 1: Statistical evaluation of performed test for validating the automatic segmentation algorithm - 57 -](#_Toc524677862)

# List of Abbreviations

3D Three-dimensional

MoCap Motion Capture

fps frames per second

SL Sign language

HamNoSys Hamburg Notation System

RP Rest Pose

# Introduction

Motion capturing is a modern, fast developing data acquiring method capable to record movement in 3D. The detailed data that is retrieved from such recording is very useful not only for movie and game industry, but also in other fields as military, medicine and for validation and control of computer vision and robotics. The benefits of this method attract scientist to utilize it for linguistic analysis. Sign language is the primary alternative to a spoken language. Unfortunately, there are people for whom this is the only alternative that gives them the ability to communicate and share thoughts directly. Sign language uses manual movements and body language to communicate thoughts with others. The basic component of a sign language includes hand gestures, movements, orientation of fingers and hands, hand shapes and facial expressions to communicate certain feelings. Every region in the world has a unique spoken language and similarly, every region has a unique sign language. Thus, sign language varies from culture to culture and from region to region. People with speech and/or hearing impairment find it difficult to communicate with other individuals via sign language due to the inability of most of the people to understand sign language.

The purpose of this paper is to show the methods of studying and understanding the properties of signs from motion point of view and developing a tool for processing sign language data base. It is focused on hand movement and gestures analysis. It is cultural and regional independent, because it uses kinematic and statistical methods for processing the data.

Such database is set of dictionary (lexical items separated by default stance //not sure if it should be here or later in the main part) files with motion capture data. The tool will perform raw segmentation (in the first step) and fine segmentation (using acquired parameters from data) of dictionary items to extract the meaningful information for each sign which will be used for further analysis. It will show the challenges to determine the exact beginning and ending of the sign. The significant problem with the nature of the data – the containing of noise and methods for isolating that noise. Each sign will be processed for extracting its properties. Such as: if the sign is one handed or two, which is the dominant hand, hand location and orientation, finger orientation.

Later with the help of computer learning methods such as SVM the extracted information will be used to cluster signs for further processing.

# Theory. Analysis. Aim.

## Motion Capture

### History of motion capture

In general, the term motion capture (MoCap) is understood as the process of recording the movement of objects, people or animals. It is not specifically related to any device or approach. Today’s motion capture systems are product of many years of tinkering and innovation.

At the beginning MoCap analysis originate as gait analysis and animal locomotion. A pioneer in this field is the photographer Eadwerd Muybridge. In his work he used multiple cameras, triggered by strings to take pictures of moving bodies and animals. He manages to capture what human eye could not distinguish as a separate movement. His work “The horse in motion” was the first work recognized as a motion capture analysis. With a series of photographs of a galloping horse (see *Figure 1*) he proves that horses do have all four hooves of the ground during their running stride [[5](file:///C:\Users\rusev\AppData\Roaming\Microsoft\Word\Muybridge's#_E._Muybridge,_)].

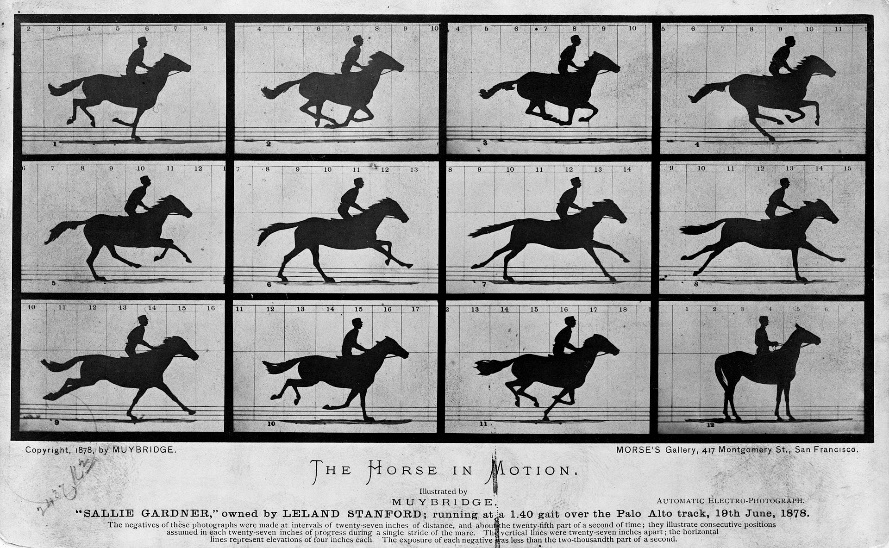


Figure : "The horse of motion" by Eadwerd Muybridge

Since then the technology has developed to the form of devices designed for direct 3D recording. The biggest success and from where it gains its popularity is the usage in the movie industry. Jar Jar Binks is the first fully digital created character with the help of MoCap. It was performed by Ahmed Best in Georges Lucas’ Star Wars Saga in 1990.

Motion capturing techniques started as biomechanics research tool, gained its popularity with its usage in entertainment industry and developed and expanded into education, training, sports, and robotics.

There are different approaches for motion tracking. The most advanced technologies are capable of tracking motion with high precision at very high sample rates. Optical passive system is considered as one of the most accurate and flexible technique and thus very common in the industry. On the other end of the scale are low-cost sensor technologies that most people use daily in their mobile phones, laptops, game controllers and so on. Every system has its advantages and disadvantages depending on its purpose.

The technologies presented in this section include acoustic, mechanical, magnetic, inertial and optical tracking.

**Acoustic tracking** systems calculate position upon the wavelength of an acoustic signal and the speed of sound. This approach is vulnerable due to the fact that sound speed varies with air pressure and temperature.

**Mechanical tracking** systems are usually based on some mechanical construction which measures angles or lengths between mechanical parts. These systems may be worn on the body, for example implementing sensor in exoskeleton or a glove. Mechanical tracking has been popular in music-related work, particularly for the purpose of developing new musical interfaces.

**Magnetic tracking** systems use the magnetic field around a sensor. Passive magnetometers can measure the direction and strength of the surrounding magnetic field. More advanced magnetic systems use an active electromagnetic source and a sensor with multiple coils. The position and orientation of each sensor can be calculated as a function of the strength of the induced signal in each sensor coil. Magnetic trackers are able to operate at high sampling rates (more than 200 Hz) with high theoretical accuracy. However, the systems are sensitive to disturbances from ferromagnetic objects in the tracking area.

**Inertial tracking** systems include those based on accelerometers and gyroscopes. These sensors are based on the physical principle of inertia. Inertial tracking systems have certain strong advantages. Firstly, they are completely self-contained, meaning that they do not rely on external sources such as acoustic ultrasound sensors or cameras which require line-of-sight. Secondly, the sensors rely on physical laws that are not affected by external factors such as ferromagnetic objects or light conditions. Thirdly, the sensors are very small and lightweight, meaning that they are very useful in portable devices; and finally, the systems have low latencies and can be sampled at very high sampling rates.

Various types of video camera are used in **optical motion tracking**. In principle, any digital video camera can be used — in fact, one of the most affordable sensors for conducting motion tracking is a simple web camera. Cameras used in optical motion tracking are either regular video cameras, infrared (IR) video cameras, or depth cameras.

Ordinary video cameras sense light in the visible part of the electromagnetic spectrum. Each pixel in the camera image contains a value corresponding to the amount of light sensed in that particular part of the image.

Infrared cameras sense light in the infrared part of the electromagnetic spectrum. Some infrared cameras can capture heat radiation, e.g., from humans, but the most common use of infrared cameras in tracking technologies is in a slightly higher frequency range. This is achieved by using some active infrared light source, and either capturing the light from this source directly or as reflections on the tracked objects. Typical implementations consist of a group of infrared light-emitting diodes (LEDs) positioned near the infrared camera and capturing the reflection of this light as it is reflected from small spherical markers.

Depth cameras provide a layer of depth information in addition to the regular two-dimensional image. These cameras use some technology in addition to the regular video camera. When not provided by the camera itself depth information can be gained through the use of stereo cameras. This involves two cameras mounted next to each other, providing two similar images. Depth information is found as a correlation function of sideways shifting of the images.

After obtaining the video data various processing is applied to the video stream. The video processing that is performed in optical tracking systems is primarily dependent on two factors: (1) whether or not the tracking is based on markers and (2) the camera configuration. But in any case the first processing step is to remove unwanted information from the video, i.e. separate the foreground from the background.

### Optical-based motion capture system VICON

Motion capture technology was used to generate data files for both continuous and isolated utterances. The data used for this project was recorded at the University of West Bohemia with MX series device from VICON [[6](file:///C:\Users\rusev\AppData\Roaming\Microsoft\Word\Go_Further_with#_)]. This is an overview of the basic principle of the technology on which the system is based. This system is based on optical- passive technology and was chosen for the sign language project because it suits the best for the purposes of the project.

The technology provides accurate data at fast sampling rates, and the same system can be used to capture simultaneously the motion of a wide range of structures, including objects, animals, human bodies, fingers and faces. By using passive (reflective) markers, all processing is done externally, and the captured subject does not need to wear electrical equipment or wires. Which can be considered as advantage in capturing finger motion, because the presence of wires can impede the naturalness of movements. The system is comprised of eight specialized infra-red cameras along with computers and software for image analysis and processing. The cameras detect small markers placed on strategic locations on the captured subjects. Markers are treated as points without volume, and as such only their positon (not orientation) can be tracked. On figure 2 can be seen the general marker set and marker’s placement for human body. For this project passive markers are used, they are coated with retro-reflective material, and this requires the cameras to emit the light, which is reflected back and detected. The external source of light is part of the camera’s body.



Figure : General marker set for human body

The detected information is then processed in dedicated hardware using software provided by the manufacturer to triangulate the 3D locations of the markers. The exact process and algorithms are know-how of devices’ manufacturer, but it is known that it is based on stereoscopic vision.

After reconstructing the 3D points into a point cloud, the system needs to determine which point is which and label each point with a marker id. This process is commonly referred to as marker labelling. The marker data is then used to estimate the kinematic motion of a model of a human skeleton. The estimation in form of bone lengths and joint angles can be used for further processing or animation. This process is called solving. The mentioned software is able to provide semi-automatic calibration of human subject for general movement.

Although this technology comes with highly accurate results and it is very flexible there are some disadvantages which may result in poor data quality or extensive costs in manual post-processing. As every vision-based technology it needs clear line of sight and occlusions may cause serious challenges. Markers placed on fingers are especially problematic and often suffer from self-occlusions when the fingers are bent or the hands are facing towards the body or with palm-up [[7](#_N._Wheatland,_Y.)]. Further challenges arise in situations when several markers come in close contact (such as clapping hands).

To summarize, optical motion capture provides highly accurate data, it is capable of recording one or more components at the same time with high sampling rate, but may require a large amount of manual post-processing.

## Sign language analysis

Sign language is used by millions of people around the world. It is used to facilitate communications with people with speech or hearing impairments. There are two common misconceptions: 1) Sign language is one and same everywhere; and 2) Sign language is dependent on spoken language [[13](#_Perlmutter,_David_M.,)]. But the truth is that there are different sing languages as there are different spoken languages. Linguists has studied and proved that sign languages exhibit the fundamental properties that exist in all languages and there are similarities between both forms, but there are some basic differences. The linguistic mechanism in both is different and therefore it causes difficulties for people with such impairments, especially for those who are born this way, to use even the written form of a spoken language [[2](#_Sandler,_Wendy;_&)]. In spoken language units are organized sequentially (it is not possible to say two different words at the same time), but in sign language the meaning of one unit may be carried by the shape of the hands and their position or/and by the position and movement of head and mimics. These two components can be carried simultaneously.

All these circumstances not only put a communication barrier between people using sign language and majority of hearing community, but also restrict them from most sources of information. Another difference between both is that sign languages does not have its own, unified notation system. This led scientist to develop writing system to represent signs. The pioneer in sign language analysis was W. Stokoe. As it is explained in M. Kato paper [[3](file:///C:\Users\rusev\AppData\Roaming\Microsoft\Word\A#_Mihoko_Kato,_)] Stokoe proves that each sign in American sign language has tree elements that distinguish it from all other signs:

1. Hand Configuration (the distinctive configuration of the hand or hands making a sign),
2. Place of Articulation (the place where a sign is made),
3. Movement (the action of the hand or hands).

Stokoe decided to call the active hand the “designator” or “dez”; the place, the “tabula” or “tab”; and the action, the “signation” or “sig.” A sign is produced by a combination of these three aspects.

Nowadays most of the notion systems for sign languages are based on his study and notion system.

### Hamburg Notation System

The Hamburg Notation System (HamNoSys) [[1](file:///C:\Users\rusev\AppData\Roaming\Microsoft\Word\Hamnosys-representing#_T._Hanke,_)] is a work of scientists from University of Hamburg. It is an alphabetic system that decompose signs to phonetic level and describes their sub lexical parameters such as location, configuration and movement. It is based on Stokoe’s notation system [[2](#_Stokoe,_William_C.)]. It is designed to be usable in variety of context with the following goals in mind:

* International use - HamNoSys transcriptions to be possible for virtually all sign languages in the world.
* Iconicity – because of the large number of parameters variations, newly created glyphs should be designed to be easy to memorize or even deduct the meaning
* Economy – notation of signs should make use of principles as symmetry conditions, thus resulting in shorter notions.
* Integration with standard computer tools – It should be usable for computer –supported transcription, standard text processing and database applications.
* Formal syntax – The notation language should have well-defined syntax.
* Extensibility – As SLs are developing and differ from each other, HamNoSys should allow both for a general evolution and specializations. [[1](file:///C:\Users\rusev\AppData\Roaming\Microsoft\Word\Hamnosys-representing#_T._Hanke,_)]

Because of all described goals of HamNoSys it seems that it is good choice for scientific point of view and preferred from scientific community working in the domain of language analysis and synthesis, although it is not very accepted from deaf community.

This notation system was chosen for sign language project because it suits the best for its purposes. I used it as guidelines in my work for describing the properties of signs from motion point of view.

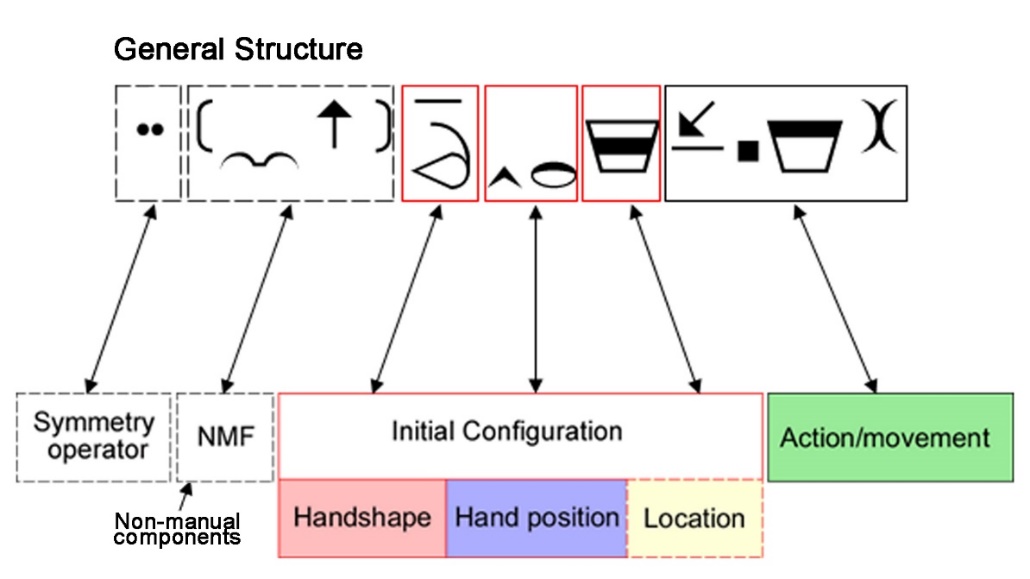


Figure : General structure of HamNoSys notation

In general, a sign notation consists of description of non-manual features, handshape, hand orientation and location, plus the actions changing this posture. If the sign is two handed at the beginning of notation is added an operator to show how the description of dominant hand is copied to the non-dominant hand. The signs are realized in signing space and terms expressing directions are determined from the signer’s perspective. *Figure 3* shows example of sign notation and its general structure with its mandatory (the ones with solid border) and optional components (boxes with dashed border).

#### Handshapes

Handshape is expressed by symbols for basic forms – **Fist, Flat hand, Separated fingers, Thumb combination** and bending (see *Figure 4*)

The configuration of the thumb alters the structural arrangement of the entire hand and thereby define a new group of handshapes, where shapes are derived from other basic shapes.

In addition to the general description of the sign, information for different fingers and finger parts (in respect to the fingers involved), may be included.

Figure : Handshapes

#### Hand orientation

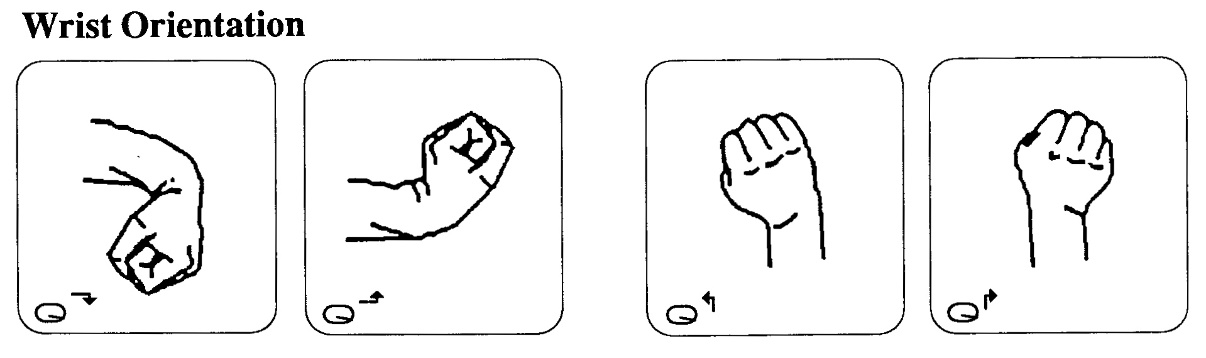
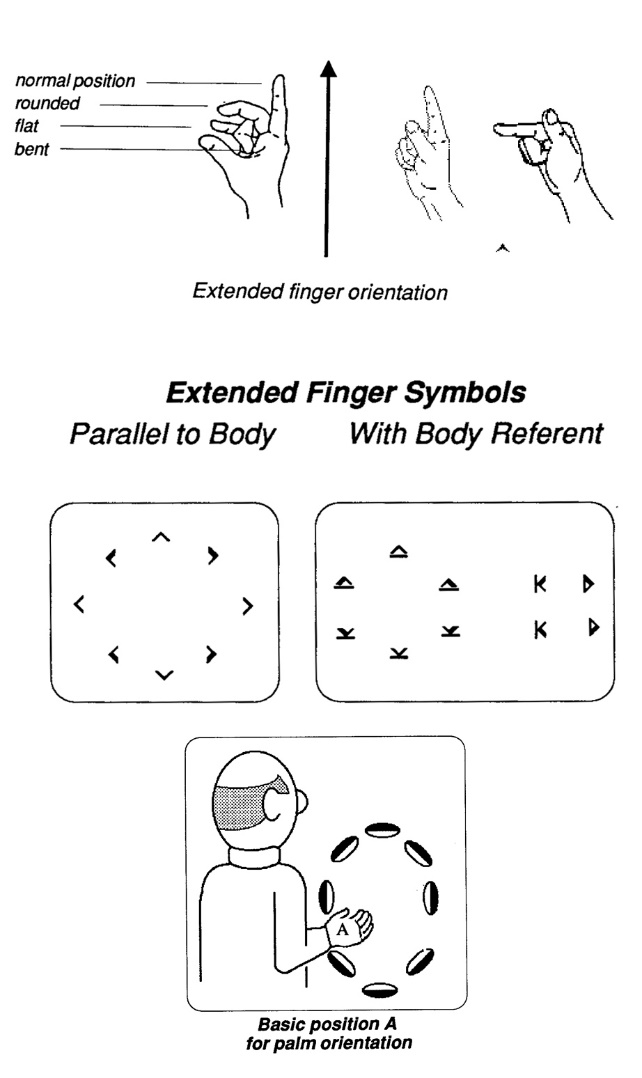
The orientation of hand is described by **Wrist orientation, Extended finger and Palm orientation**. Where wrist orientation is characterized by bending of the wrist toward the pulse or back of the arm, toward the thumb or little finger (see *Figure 5*). It is omitted when the bending is natural consequence of palm orientation and movement.

Figure : Wrist bending

The vector originating at wrist, running along the back of the hand and pointing to the direction pointed by the fully extended finger, shows the orientation of the Extended finger. The vector is shown on *Figure 6*. This orientation may be difficult to be defined when all fingers are bend in some way. As it can be seen on *Figure 6*, the two hand shapes on the right are noted with same symbol. Fingers point in different direction, but the base of the finger (knuckle) points in the same. If finger orientation is not parallel to the body plane, then the notation must include reference to the body (*Figure 6*).

Palm orientation is always noted after the Wrist and Extended finger orientations are. To define it, first Basic Position of the palm must be determined, then the orientation is determined by the orientation of the palm around the shaft of the hand (*Figure 6, bottom*). There two Basic Positions. When Extended Finger orientation is away from the body is Basic Position A, otherwise is Basic Position B.

Figure : Extended finger orientation (top) and Palm orientation (bottom)

#### Hand location

Hand location is noted only if it is very specific on the body or in the signing space. If not noted it is understood that sign takes place in the natural space, in front of the upper part of the body. There are different sets of location signs. One is expressing locations on the body (*Figure 7*), another one describes relative position in space.

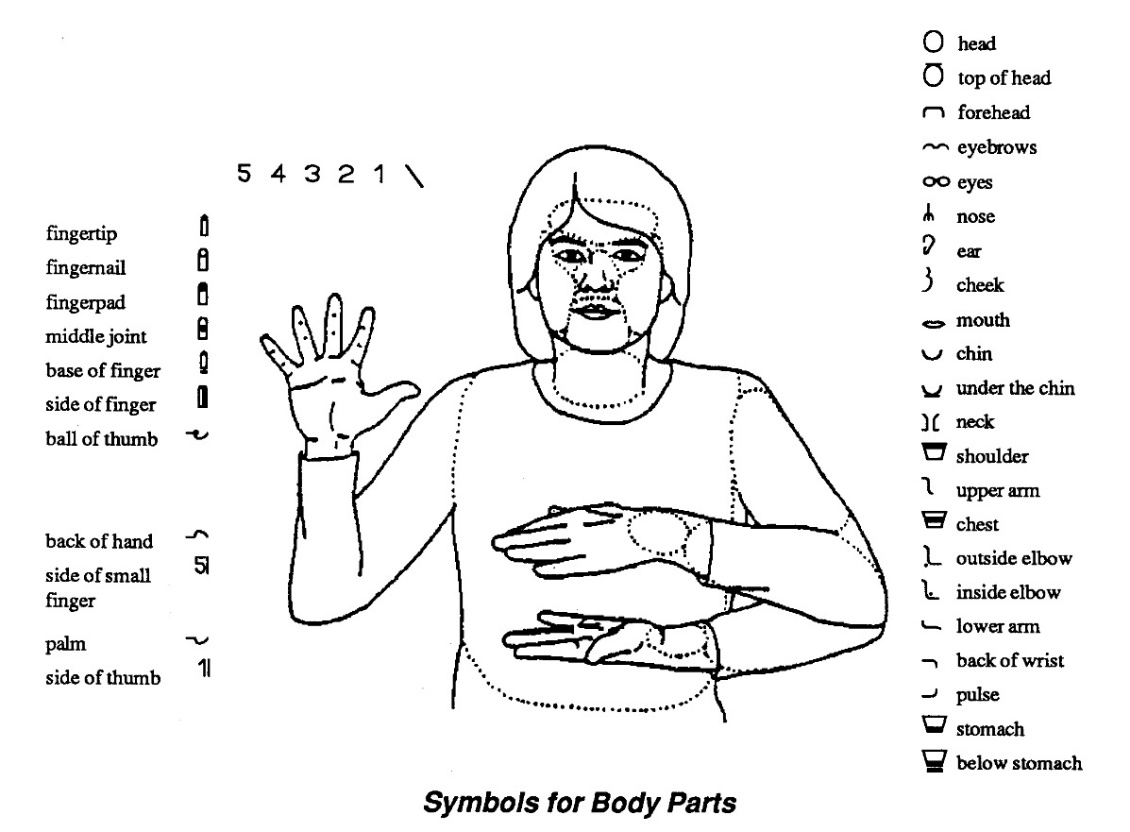


Figure : Body parts

*Figure 8* shows the division of the signing space into six horizontal zones, each zone is noted with different symbol. In two-handed signs location information concerning hands and arms, refers to locations on non-dominant hand and characterize the relation between the two hands. Again all determinations of left and right are made from signer’s point of view.

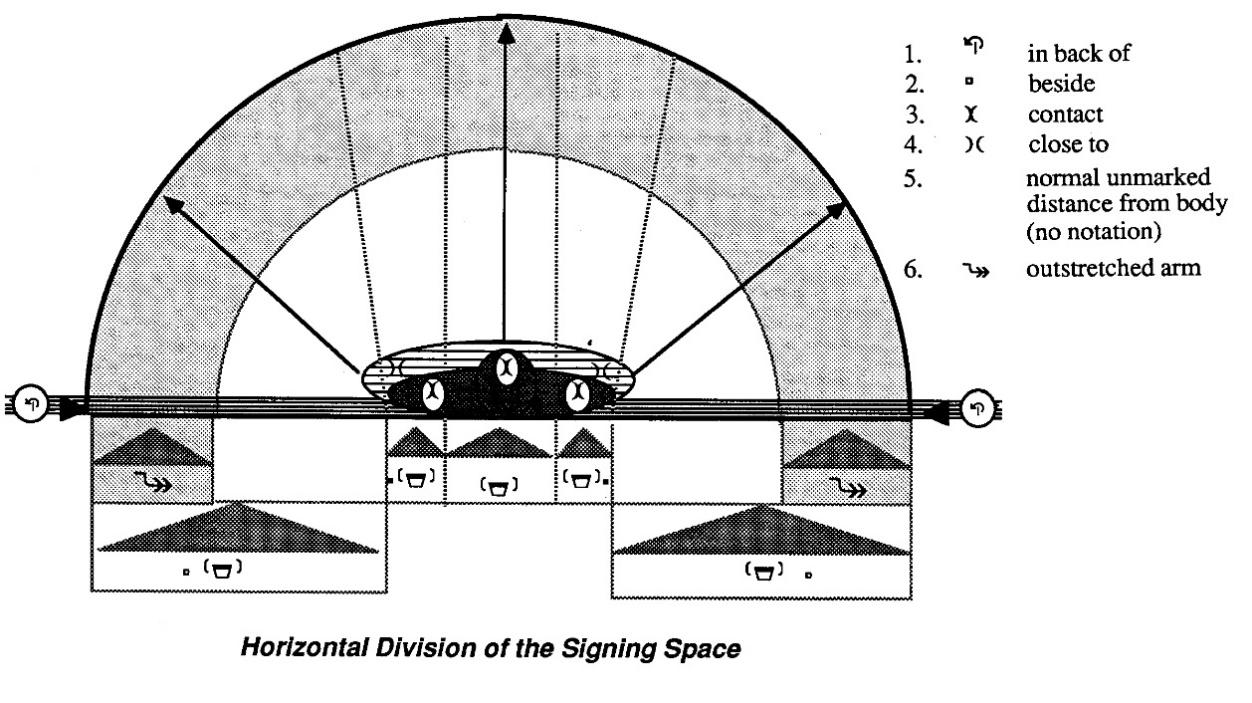


Figure : Six zones of signing space and their symbols

#### Movement

There are few aspects of the movement that have to be considered when describing movement in signs: **Movement type, Manner of movement, Repetitions, Order of movement symbols**. The distinguished types of movement are: straight, curved, wavy, zigzag, circular and spiral.

The manner of movement aims do describe size, speed and intensity of the movement. There are three degrees of size: large, normal and small and only large and small size are explicitly notated. The speed and intensity are described as: fast movement, slow movement, tense, hold or rest (usually on onset or offset of the movement), abrupt halt at the end of the movement.

Two types of repetitions are distinguished. Ones which lead back to the beginning of the movement and those whose initial position continuously change. The number of repetitions is indicated by the number of repeating symbols.

The order of notation of these aspects is strongly defined as follows:

1. Circling movement or direct movement
2. Type of movement:
3. Manner of movement
4. Repetition

#### Two handed signs

Two handed signs are divided into symmetrical and nonsymmetrical signs. In the symmetrical ones both hands show the same handshape and therefore only the dominant hand is notated. But the coordination of both hands should be noted either as mirrored or parallel. Also should be noted if the movement in both hands occurs simultaneously or alternately. In case of nonsymmetrical signs, the movement occurs only with the dominant hand and the handshapes are often different, therefore notation for both hands is needed. Always the notation of the dominant hand should precede the non-dominant.

#### Conclusion

Sign language and spoken language developed separately and this led to different linguistic structures, restricting people with haring and speaking impairments from vast information and free communication. Linguist have been working on developing good sign representation as it is important from many aspects. Although it is not very accepted by deaf community, because of its complexity, the Hamburg Notation System seems to be a good choice for representing signs in computer, as it was developed with the idea to be available for international use and easy for computer implementation.

## Optical-based motion capture for sign language synthesis

There have been several research projects for developing assistive technology for Deaf population. They are based on different techniques, such as key frame techniques and procedural synthesis. As it is explained in [[13](#_L._Naert,_C.)] they allow fine control over the movement of the signing avatar, but are not very well accepted by people, because of lack of human-like movements. Scientists as McDonald et al [[4](#_J._McDonald,_R.)] work on resolving this problem by analyzing noise in motion capture data and adding the human-specific noise in the key-frame driven motion. Other approach is data-driven synthesis. When used with MoCap data from real signer the realism in the motion is preserved and therefore improves the animation.

On the other hand, this approach needs a rich MoCap dataset for sign language, which faces other challenges. As mentioned before SL is a complex composition of simultaneous movements of different parts of human body. The difficulty is to record and synchronize all these components, because every utterance can be done with time and special variance even if performed by one signer. At university of West Bohemia, where I had the opportunity to work, deal with this problem by using the state-of-the-art device for motion capturing [[12](#_P._Jedli_cka,_Z.)]. The system they use is combination of eight cameras and set of 109 retro-reflexive markers with different sizes for different body parts. As this is a new approach, the data set is not that rich yet.

Data sets for SL may differ in their content. Some contain isolated utterances, other whole sentences or phrases with continuous utterances. For the purpose of my work I have used the dataset with isolated utterance.

## Segmentation of SL

The benefits of data-driven methods for SL synthesis were discussed in the previous section, but in order to have good and valid results the data must be reliable. Before proceeding to any kind of analysis the data need to be structured in forms that suits the final purposes.

Segmentation is the process of breaking a continuous sequence of movement data into smaller and meaningful components. The process consists of determining the exact beginning and ending moment of this meaningful component. Identifying of segments is challenging task due to multidimensional nature of SL.

There is a study over French sign language [[13](#_L._Naert,_C.)] which is based on manually segmented and annotated data. Segmentation is performed by expert annotators (deaf signers). As it is explained in the paper manual segmentation and annotation is laborious and time-consuming process. Identifying the exact start and end frame by human is subject to variability, even if made by experts. The focus of this study is *transition* movement in the sign. *Transition* is the motion between end of one sign and beginning of next sign. This movement have no linguistic meaning, but it is important for achieving human-like animation.

There are several studies researching the issue of segmentation, but as far as I am aware, there still have not been developed a fully automatic segmentation of continuous utterance with good precision. The approach for identifying segment boundaries by detecting changes in kinematic features for general motion and in the case of isolated utterance seems promising.

### Method for segmentation

In my work the segmentation is designed to be made over sets of isolated utterance and is done in two steps. In these data sets signs are separated with one exact pose – rest-pose (RP). The identification of segments is based on kinematic analysis of hands movement.

At first step start and end tag define these frames where hands leave and enter, respectively, rest-pose. Knowing this frames helps me to analyze hands behavior during rest pose and thus refine threshold variables for finer segmentation on next step.

At second step the frames that I am interested in are those where the meaningful part of sign begins and ends. Meaningful part is the one that have linguistic meaning and is described by HamNoSys notation. This motion subsequence will be further analyzed.

The main idea behind kinematic analysis is that significant changes in trajectory of a hand can be indicated by changes in velocity and acceleration. Indeed, human body is not a simple mechanism and hand movement (or in our case change in the trajectory of the marker) is related to the kinesthetic muscle response, resulting in changes in velocity and acceleration. The correlation between this points of interest and changes in kinematic features was studied at the university of California for a project for mapping music to a dancer’s motion [[16](#_F._Bevilacqua,_J.)] and also in a research of a French sign language [[13](#_L._Naert,_C.)], where they use these dependencies to refine manual segmentation.

## Related work

Developing signing avatar will improve the quality of life of Deaf community, but it faces a lot of challenges in fields such as linguistic and human machine interaction including: speech and text recognition in order to create a sensible conversation, motion analysis, sign classification and synthesis, creating a realistic animation and other. This research area is relatively young, but two influential European projects, ViSiCAST and eSIGN, had developed technology for signing avatars based on HamNoSys and advancing it to SiGML (XML based language for describing sign properties) and data-driven methods. There are also other projects developing such systems but to the best of my knowledge this systems relay on manual segmented and annotated data, which is time consuming and laborious process. There are also research concerning automatic motion segmentation, but most of them are focused over the general movements and are animation orientated. This work aims to develop a tool that will help with the automatic processing of the 3D motion data and thus ease the process of developing the corpora needed for such systems. In this section such systems and researches are reviewed in more details.

### ViSiCAST system

The goal of ViSiCAST is to improve the quality of life of Europe's deaf citizens by widening their access to services and facilities. The project develops virtual signing technology in order to provide information access and services to Deaf people. It has developed from two previous projects Simon-the-Signer (*see section 2.5.3*) and TESSA signing avatar (*see section 2.5*.2). It aims to automate translation from text subtitles into sign language. This system divides into a front-end, which applies natural language processing techniques and a back-end – the animation system. The interface between these two subsystems is a phonetic-level definition of the required signing sequence expressed in SiGML. At the start of the earlier ViSiCAST project, motion capture was used exclusively to generate signed content. This involves the use of MoCap techniques to record the motion of a real human signer. The recorded data is manually segmented, resulting in a set of motion data files constituting a lexicon of signs (as in the TESSA system). These can then be replayed through the signing avatar on demand. But the use of motion capture technology in this way are relatively high which motivated later development of the system by implementing methods for synthetically generation of sign animation from an input script in SiGML notation. To provide the data needed to animate the virtual human in the receiver, the gesture movements of the human sign language interpreter are recorded in the form of motion capture. Data is captured using individual sensors for the hands, body and face. Data-gloves, which have sensors to record finger and thumb positions, are used to record hand shapes. Magnetic sensors also record the wrist, upper arm, head and upper torso positions in three-dimensional space relative to a magnetic field source. A video face tracker, consisting of a helmet-mounted camera with infrared filters, surrounded by infrared light emitting diodes, records facial expression. Reflectors are positioned at regions of interest such as the mouth and eyebrows. [[17](#_http://www.visicast.cmp.uea.ac.uk/P), [28](#_J._Glauert,_R.)]

#### TESSA Avatar

Tessa is an experimental system that aims to aid transactions between a deaf person and a clerk in a Post office. The system translates the clerk's speech into British Sign Language (BSL) and displays the signs using a specially-developed avatar. Developing a system for use only in a post office, enables the creators to sidestep or simplify the difficulties of speech recognition, automatic translation of text into suitable representation of sign language and displaying this representation by an avatar, by defining a limited set of phrases that can be recognized and displayed in sign-language. Many of the transactions at post office are predictable and hence much of the associated language can be pre-defined. To generate true and realistic signing from virtual human they use different motion capturing techniques for the hands, body and face.

1. Cybergloves with 18 resistive elements for each hand are used to record finger and thumb positions relative to the hand itself.

2. Polhemus magnetic sensors record the wrist, upper arm, head and upper torso positions in three-dimensional space relative to a magnetic field source.

3. Facial movements are captured using a helmet mounted camera with infra-red filters and surrounded by infra-red light emitting diodes to illuminate Scotchlight reflectors stuck onto the face. Typically, 18 reflectors are placed in regions of interest such as the mouth and eyebrows.

The separate streams are integrated, using interpolation, into single raw motion data stream that can drive the virtual human. Each sign is represented by a data file recording motion parameters for the body, arms, hands, and face of a real human signer. For a given text, the corresponding sequence of such motion data files are used to animate the skeleton of the avatar.

The drawback of this system is the substantial amount of work involved in recording the large number of signs required for a complete lexicon [[28](#_J._Glauert,_R.)].

#### Simon the signer

Simon the Signer takes words from a television subtitle stream and renders a sequence of signs in Sign Supported English (SSE) to appear as an optional commentary on screen. The architecture of this system is similar to the TESSA signing avatar, but instead of signing avatar driven of recorded motion by MoCap techniques, a set of video recorded signs is used. The drawback of this system is again the need of predefined set of video recorded signs. Also concatenation of signing is more fluent and controlled for avatar then for video signing, as small changes in positioning, etc. of the avatar can be easily made, while in the case of regular video a new recording have to be made [[28](#_J._Glauert,_R.)].

### eSIGN editor

The eSIGN is an international project funded again by the European Union and it is build over ViSiCAST animation and language technology. It aims to provide important Government information in sign language, using avatar technology. The eSIGN Editor provides an easy-to-use way to script avatar performances. The system is based on databases (or 'lexicons') of signs. This means that when an individual sign has been created for one section of signing, it can be used again in other sections. As these databases of signs grow, it becomes easier and quicker to put signing onto websites. Virtual signing works by sending commands from a website to animation software installed on a user's PC. The information that is sent from the website to the Avatar is more compact than the information that must be sent when downloading video clips. This means that high quality signing can be provided over low bandwidth Internet connections, something that is not possible when using video. In addition, the Avatar itself is a real 3D system, so not only is the quality of the image higher than in the video, but the character can be rotated by the user to provide the optimum view. Another key advantage is the ability to produce signed output by blending together sequences of signs to make new phrases on demand. This has the potential to integrate into web content management systems, again increasing the viability of including signing on websites. The concept of synthetic animation used in eSIGN is to create scripted descriptions for individual signs and store them in a database. While populating this database may take some time signed phrases can be made quite quickly by selecting the required signs from the database and assembling them in the correct order. Contextual issues may mean that some modification of the signs used is required, but generally phrases can be created quite quickly [[20](#_eSIGN_project,_http://www.visicast.), [21](#_U._Ehrhardt,_B.)].

### SignCom

The project seeks to build interaction between users and virtual agents communicating in French Sign Language, and thus engaging in real-time dialog. This is achieved by the human user signing towards a camera by which the system recognizes his/her signs, and by the virtual agent providing culturally- and linguistically-acceptable responses in behavior and sign, respectively.

The system is composed of two large building blocks: one, operating off-line, is a dually-indexed database containing both motion capture and semantic data, and the other, operating on-line, comprises the automatic recognition of signs, the animation of a virtual signer, and a go-between module that produces meaningful and appropriate dialog. For creating the needed corpora, they use optical based motion capture system, with the following setup: 12 motion capture cameras, 43 facial markers, 43 body markers, and 12 hand markers. They also annotated the corpus, identifying each sign type found in the MoCap data with a unique gloss so that each token of a single type can be easily compared. Other annotations follow a multi-tier template which includes a phonetic description of the signs, and their grammatical class. The annotation is manually performed by experts [[18](#_S._Gibet,_N.)].

Later as described in [[13](#_L._Naert,_C.)] they work on developing an as automatic and precise as possible method for segmenting and annotating continuous stream of motion. They propose a way of segmenting sign language utterances by analyzing some kinematic properties of sign language motions and, more specifically, by detecting local minima in the norm of the velocity for each hand. The manual annotations done by deaf experts are then refined by selecting the nearest corresponding minima for each manual segment, each hand is considered separately.

### Sign3D

The Sign3D project aims at creating a range of innovative tools to allow the recording and the processing of motion captured French Sign Language (LSF) content. In this project sign language sentences are directly generated by concatenating motion captured items. For creating the corpus, they use motion capture techniques. For the beginning of the project they capture only twenty sentences. The annotation of the motion is done by deaf signer with the help of reference video and Elan software. Sentences are segmented into signs, labelled by a string conveying its meaning. Other meta-data can also be added to the segments about handshape, face expression, body posture. The corpus annotation allows a mapping between the meanings of the signs (and distinctive features) and their realization presented as motion captured data [[23](#_S._Gibet,_F.), [24](#_Sign3D,_http://www.mocaplab.com/pro)].

### Korean sign language

In a study for the Korean sign language scientist have developed a method for segmenting a continuous utterance and sign recognition based on state automata and Hidden Markov Model.

By analyzing hand motion kinematics, such speed and change of speed, they determine 5 motion phases during gesture (stop, preparation, stroke, moving, end; where only the stroke phase is obligatory for the meaningful gesture). Based on phase patterns they develop 6 rules. Then, the rules are adopted to discriminate meaningless gestures and segment the continuous sign language sentence into several isolated words in the framework of state automata.

To recognize isolated words, they had defined 11 hand motion classes depending on motion features, and adopted Hidden Markov Model (HMM), which is effective to model spatio-temporal information. [[22](#_J.-B._Kim,_K.-H.)]

### Data-driven Finger Motion Synthesis for Gesturing Characters

In their research project for data-driven finger-motion synthesis Sophie Jörg et al. develop a method to automatically add plausible finger motions to body motions. Their algorithm uses a previously recorded database of body and finger motions to generate finger movements that match an input body motion. To effectively search for similar body motions, the input motion must be segmented into meaningful fragments. According to their research a gesture consists of several phases: preparation, where the arm moves away from its rest position, pre-stroke hold, stroke, post-stroke hold, and retraction, where the arm moves back into the rest position. Gestures can be combined without returning to a rest pose. Only the stroke phase is obligatory, all other phases are optional. They suggest a method for segmenting the motion by choosing the moment when the speed of the wrist joint crosses a threshold close to zero, thus separating motion phases (high speed) and hold phases (low speed). They restrict segment length to be no less than 0.33 seconds and no more than 2 seconds. If segments are too long, an additional split is added at a suitable local minimum of the speed. These restrictions reduce spurious segmentation due to noise and avoids overly long segments. [[19](#_S._Jörg,_J.)]

### Mapping sound to human gesture

In a research on gestural control of digital music, performed at the University of California Irvine, group of artists and scientists suggest an approach for gesture segmentation and recognition. For the motion segmentation they use Laplacian edge detection on acceleration curves. For segment recognition they use Principal Component analysis to analyze classes of short motions (training set).

The research is performed over various movements, recorded using 3D optical MoCap. The system consists of 8 Vicon cameras, a set of 30 retroreflective markers, set over a dancer’s body. The recording is made at 120 Hz.

In order the system to have better control over the music, they first break the long sequence of motion into shorter gestural segments. This way each segment can be used to generate single event to control musical process. Their method focuses on data from the motion of a single marker. At first step they try to find transition points, where the marker begins to move. By using the raw 3D data in form of 3-dimensional vector of time, they calculate the first derivatives with respect to time, obtaining the velocities along the 3 axes. They found that there is strong correlation between transition points they are looking for and an increases in the acceleration of the marker. To eliminate the noise, they apply low pass filter to both, the position information as well as the subsequent function. To parse the transition points, they use standard edge detection technique – finding local minima and maxima by looking at increase and decreases in first derivative with respect to time. The leading edge of each peak is found by looking for the first significant increase in the function subsequent to a significant decrease.

After the transition points are found and the long motion is parsed into gestural segments they extract number of parameters to characterize each segment. A list of such properties includes items such as: the time taken to traverse the segment, the total distance traversed over the segment, the average speed while traveling along the segment, and the "curviness" of the path.

Their results show that the system can designate a transition point 3-5 frames after it has appeared (at 120 Hz recording this equals to 100-150 ms). The benefit of this method is that no specific patterns or gestures are searched for, which makes the system available to be used without the need of learning specific gestures and there is no need of prior knowledge of gestures. [[16](#_F._Bevilacqua,_J.), [25](#_Gestural_Control_of), [26](#_Bevilacqua_F.,_J.)]

### Segmenting Motion Capture Data Using a Qualitative Analysis

Scientists from the university of Pennsylvania present a study over segmenting general motion capture data by using qualitative analysis. They propose a method to automatically produce semantic segmentations by examining the qualitative properties that are intrinsic to all motions, using Laban Movement Analysis (LMA). This method searches for semantic motion sequence boundaries by using the confidence of a LMA classifier that is calculated by measuring the agreement of multiple LMA classifiers, each trained to be sensitive to different motion sequence boundaries.

Laban Movement Analysis (LMA) is a system for interpreting and describing human motion that focuses on the relationship between an individual’s internal state and its effect on motion. LMA characterizes human motion as a combination of four components: Body, Shape, Space, and Effort. Body, Shape, and Space describe the articulation of the body, or what motion is performed. Effort describes the quality of a motion, or how it is performed, in terms of four different factors: Space, Time, Weight, and Flow. LMA is useful for producing a semantic segmentation of motion capture data because LMA descriptions are semantically meaningful and present in all motions.

LMA classifiers that use simple kinematic features and neural networks are used to determine the LMA Effort Elements of motion sequences. The confidence of the LMA classifier output is computed by measuring the agreement of many LMA classifiers, each of which is trained to be sensitive to different input sequence boundaries. The segmentation is computed by searching for sequences with high LMA classifier output confidence.

The performance of the LMA segmentation method is measured in two different ways. First, the method’s ability to identify segments that are semantic is measured by counting the number of segments that occur in the automatic LMA segmentation and manual segmentations. Second, the method’s ability to select segment boundaries that are similar to a manual segmentation is measured by evaluating the performance of a simple motion classifier that is trained with manually segmented motion data. The results of the LMA method were also compared to kinematic method based on hand linear velocity. The result shows that LMA segmentation is the most similar to manual segmentations at the segment-level as well as at the frame level. Also the velocity method shows high accuracy at frame level. [[27](#_Bouchard,_D.,_&)]

## Conclusion

Sing language is important for certain part of our community. It is complex and by its nature very different of the spoken language that most of us are familiar with. There have been different researches and projects for developing signing avatars with increasing quality over time. New approaches include data-driven methods, using devices for recording 3D movements in order to save human-like motion. But the issue of automatic processing and computer understanding are still challenging tasks. SL can be reviewed as continuous stream of motion, with no consideration of linguistic meaning of the signs. Therefore, it can be decomposed to phonetic level and described using HamNoSys notation. In order to have valid result proper segmentation of the data is needed, only the ‘meaningful’ motion has to be extracted. Defining start and end moment of the sign is variable task, even performed by experts. But these moments might be detected by analyzing hand kinematics of the motion as velocity and acceleration

# System Architecture

After analyzing the theory behind motion capture technology, sign language and its properties and different approaches for sign language synthesis is possible to proceed with actual describing of the system’s architecture. The system consists of two general modules- segmentation module and module for a sign analysis. The tool will perform two step segmentation over input data. The input data is a long stream of 3D recorded signs performed by professional signer. After that the motion data will be analyzed to extract motion features of each sign.

## Python

Python is a high-level, interpreted and general-purpose dynamic programming language that focuses on code readability. The syntax in Python helps the programmers to do coding in fewer steps as compared to other programming languages. It is very flexible language supporting multiple programming paradigms, including object-oriented, imperative, functional and procedural. It has a large and comprehensive standard library that has automatic memory management and dynamic features. The programmers of big companies use Python as it has created a mark for itself in the software development with characteristic features like -Interactive, Interpreted, Modular, Dynamic, Object-oriented, Portable, High level, Extensible in C++ & C

One of its biggest advantages is its Extensive Support Libraries that include wide range of areas such as string operations, web service tools, operating system interfaces, protocols, data-analysis and more. These libraries make Python more useful for specific purposes. Additionally, Python has become a go-to language for data analysis. With data-focused libraries like NumPy, and matplotlib, making it powerful tool for processing, manipulating, and visualizing data [[14](#_Python_documentation_-)].

* **NumPy**

The most fundamental package, around which the scientific computation stack is built, is NumPy (stands for Numerical Python). It provides an abundance of useful features for operations on n-arrays and matrices in Python. The library provides vectorization of mathematical operations on the NumPy array type, which ameliorates performance and accordingly speeds up the execution.

* **SciPy**

SciPy is a collection of mathematical algorithms and convenience functions built on the NumPy extension of Python. Its different submodules correspond to different applications, such as interpolation, integration, optimization, image processing, statistics, signal processing, special functions, etc.

* **Matplotlib**

Another SciPy Stack core package and another Python Library that is tailored for the generation of simple and powerful visualizations with ease is Matplotlib.

However, the library is pretty low-level, but with a bit of effort, you can make just about any visualizations: Line plots, Scatter plots, Bar charts and Histograms, Pie charts, Stem plots, Contour plots, Quiver plots, Spectrograms.

There are also facilities for creating labels, grids, legends, and many other formatting entities with Matplotlib.

* **C3D**

This is a small library for reading and writing C3D binary files. C3D files are a standard format for recording 3-dimensional time sequence data, especially data recorded by a 3D motion tracking apparatus.

## Sublime Text code editor

Sublime Text 3 is a lightweight, cross-platform code editor known for its speed, ease of use, and strong community support. Its real power comes from the ability to enhance its functionality using Package Control and creating custom settings. There are many plug-ins accessible that make Sublime extremely smooth and pleasant IDE for Python development.

**Anaconda** is an extremely powerful Python package for Sublime. It offers:

* Python code auto completion
* Python lining (highlights both syntax errors and PEP8 violations)
* Python documentation

## C3D file format

C3D is a biomechanics and motion capture file format. It stores raw 3D co-ordinates and analog sample data, together with information that describes the stored data. The C3D format treats information as if it belongs to one of two classes: **Physical Measurements, Parameter Information**.

Physical Measurements - The C3D specification expects physical measurements to be one of two types, either positional information (3D co-ordinates) or numeric data (analog information).

Parameter Information - contain information about the data such as measurement units and data point labels, database information such as the subjects name, diagnosis and other items that may be specific.[[15](file:///C:\Users\rusev\AppData\Roaming\Microsoft\Word\The_C3D_file#_)]

For this project C3D file format is used for the input data. It contains the raw output data of the MoCap system.

## Activity flow

The general architecture of the system consists of the following smaller modules:

* module for reading input data,
* module for segmentation,
* module for sign analysis.

The general control flow of the system is illustrated in UML activity diagrams. *Figure 9* depicts the general overview of the workflow of the system.

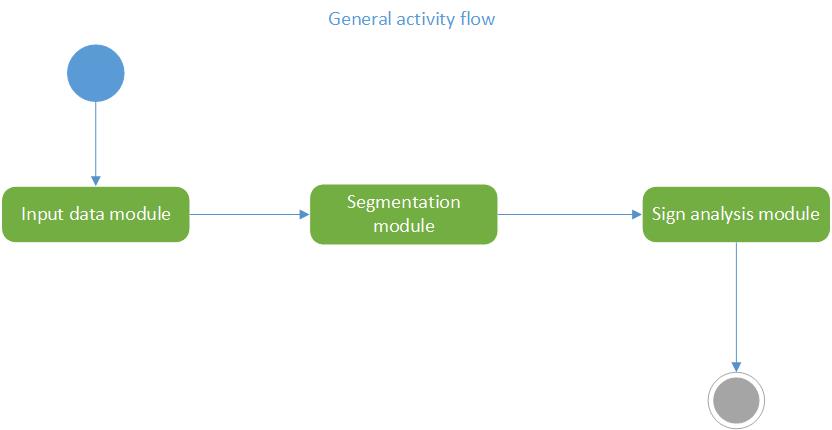
At the beginning the module for reading and initial data transforming is executed. At this stage all the information from the recording session, which will be necessary for further processing, is extracted from the input file. After the data is transformed in suitable form it is used from the segmentation module, where the raw segmentation is performed. The result from this stage is list of pairs of the numbers of the start and end frame of the signs. This pairs mark the frames where the dominant hand leaves rest pose and enters it. This result is used for the next stage, where each element of the partitioned sequence is analyzed for sign feature extraction.

Figure : General activity diagram

### Input data module

The more detailed activity flow in this module is shown in *Figure 10*.

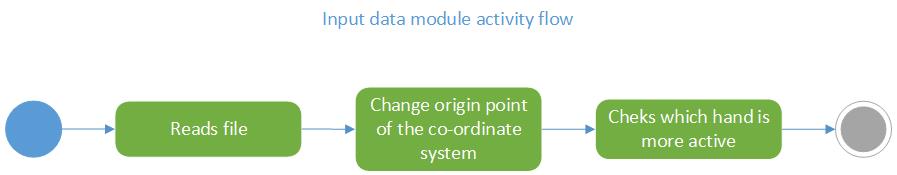


Figure : Activity diagram for Input data module

The input data is in form of .c3d file consisting the parameter and analog data from the recording session. This module will implement methods for reading the file, extracting the 3D information about markers positions, list of the labels of the used markers., as well as useful information about the frame rate of the recording, and the total length (in frames) of the recording. Next step in this activity is to recalculate the 3D co-ordinates for each frame so the origin of the relative co-ordinate system to be in the middle of the body, thus making the data independent from external factors (this will be further explained in section 3.8). At next step is determining the more active hand. The further analysis will be based on the motion of the dominant hand.

### Segmentation module

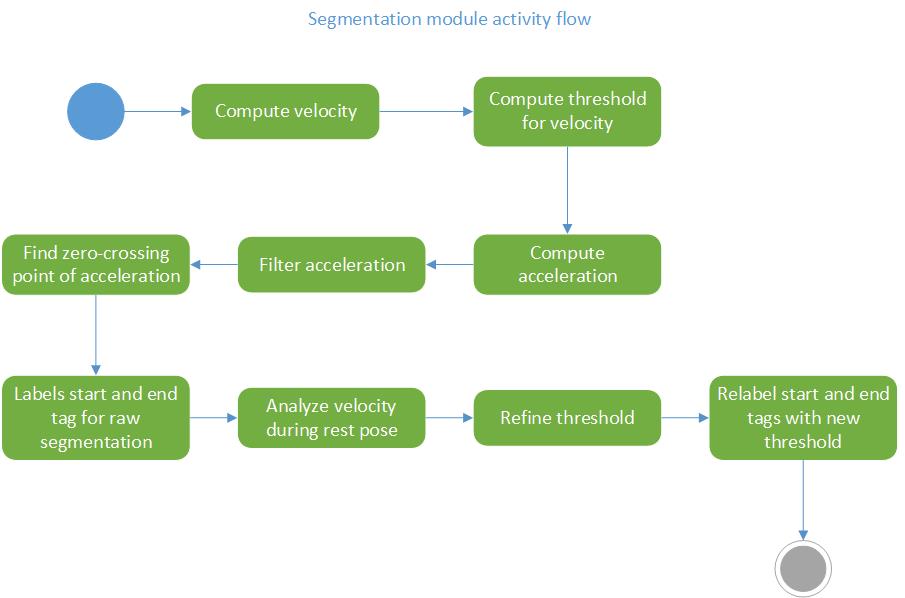


Figure : Activity diagram for Segmentation module

This module will use as input the obtained data from the previous module. The data is considered as a 3 dimensional vector function of time. As it can be seen on the Activity flow diagram for this module (*Figure 11*), first step will be computing the hand velocity. This will be then normalized toward three axes, in order to remove bias toward particular directions. Based on the result hand acceleration will be computed and a threshold parameter for the velocity will be chosen. A crucial step in the processing of the data at this stage is the use of low pass filter to eliminate jitter in the acceleration function. It is important because of the next step – finding the zero-crossing points. After the points of interest are obtained a labeling process will be performed. The result after this step should be pairs of frame numbers, denoting where the hand leaves rest pose and enters it. This result will be then used to analyze hand kinematics during the rest pose and this way set better threshold parameter. Then the process of labeling is repeated with the refined threshold.

### Sign analysis module

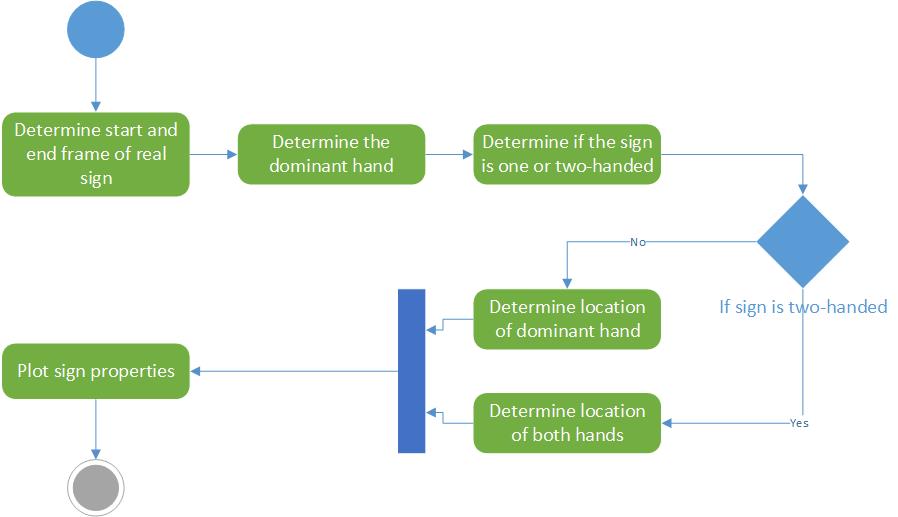


Figure :Activity diagram for Sign analysis module

*Figure 12* shows the flow activity during analysis of the segmented data. First is determined the exact start and end frame of the real sign (the explanation what exactly is the real sign is given in section 2.4). Then the sign is determined as one or two handed. This will determine how the further analysis will be performed. As it was explained in section 2.2 in the case of one handed signs and two handed symmetrical signs only the dominant hand is reviewed, rather than in the case of two handed non symmetrical signs where both hands should be reviewed.

## Input data

The input data is a so called dictionary file, it consists of sequence of lexical items, followed by a rest-pose (see *Figure 13*). A lexical item may refer to a single sign or composition of two or more signs.

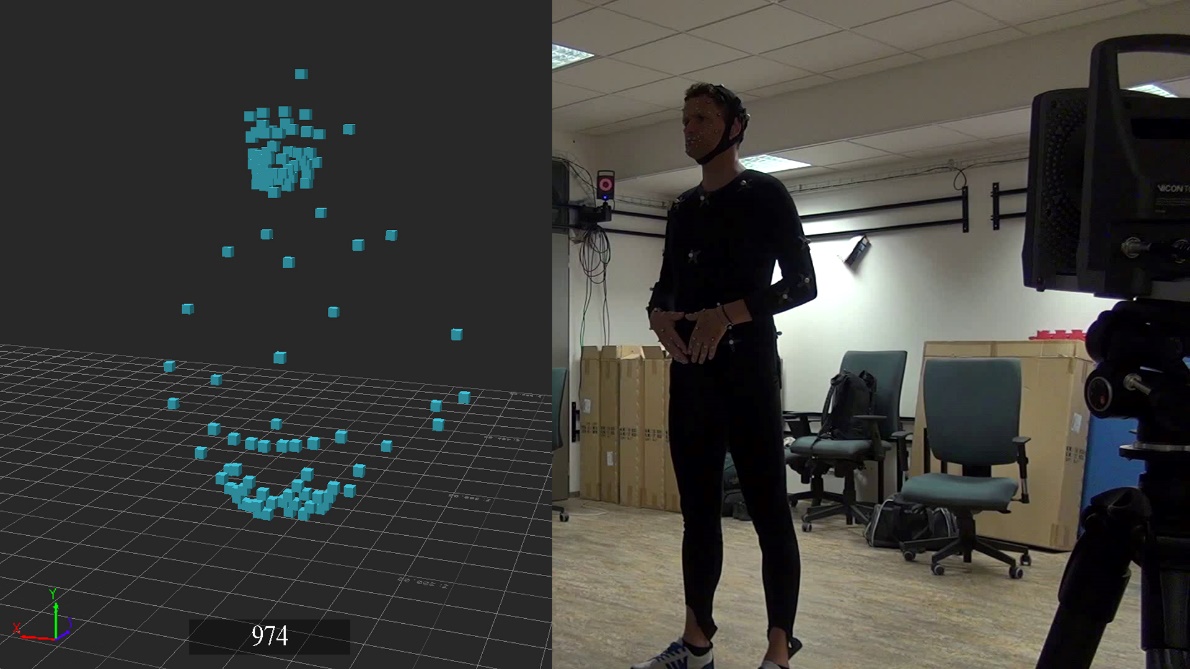
The input data is direct output of a MoCap system in .c3d file format. Basically it is a “cloud” of points with labels and co-ordinates for each frame. The input file should represent a recording session. A common practice when recording motion with 3D MoCap tools is each session to start and end in a fixed posture. Usually this is so-called T-pose. It helps the recording software to automatically recognize the human skeleton and ease the process of solving. The information that needs to be extracted from this file is the 3D information about markers position in each frame, the set of used labels and the frame rate at which the recording was performed.

Figure : Ilustrates character in Rest-pose: on the left is the reconstructed image from MoCap data; on the right - real signer during recording

## Hand marker set

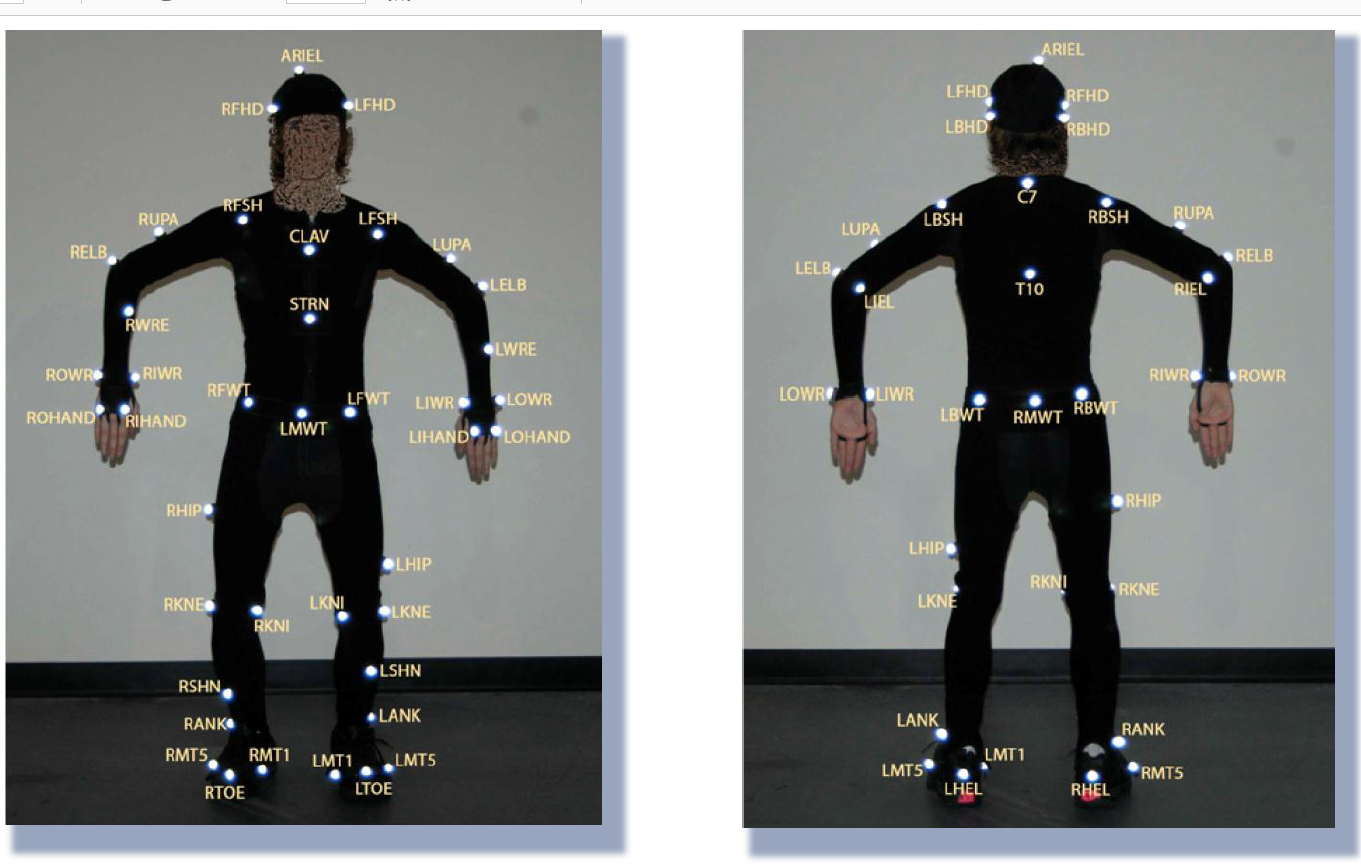
The marker set consist of 109 markers, placed over the torso, hands, fingers, head and face. The focus in this paper is in hands’ motion, not the movement of each marker. And trajectories of the four markers on top of hand can be interpreted as hand trajectory, so-called solving process. Hand trajectory is calculated for each frame as follows:

Figure : Hand markers position on right hand

## Segmentation

Segmentation is performed in the meaning of finding start and end frames of all lexical items in a dictionary file. A common approach for motion segmentation uses kinematics features of the movement. The features that are reviewed here are Hand displacement, Hand velocity and Hand acceleration.

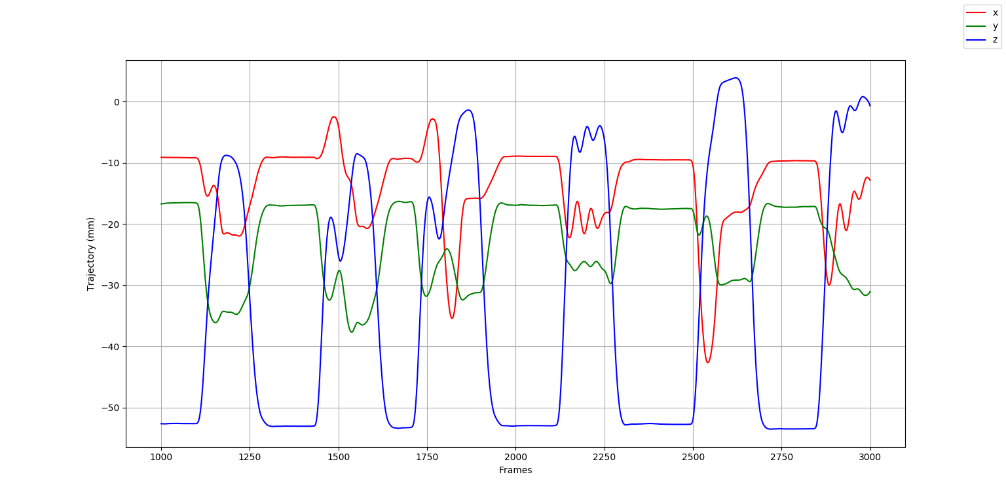
* **Displacement -** defines the change in position that occurs over given period of time. It is a vector. *Figure 15*, shows an example interpretation of the data as simple function of time. *Figure 15* shows the graph of this function for each axis.

Figure : Graph illustrating Hand displacement along the 3 axes with respect of time

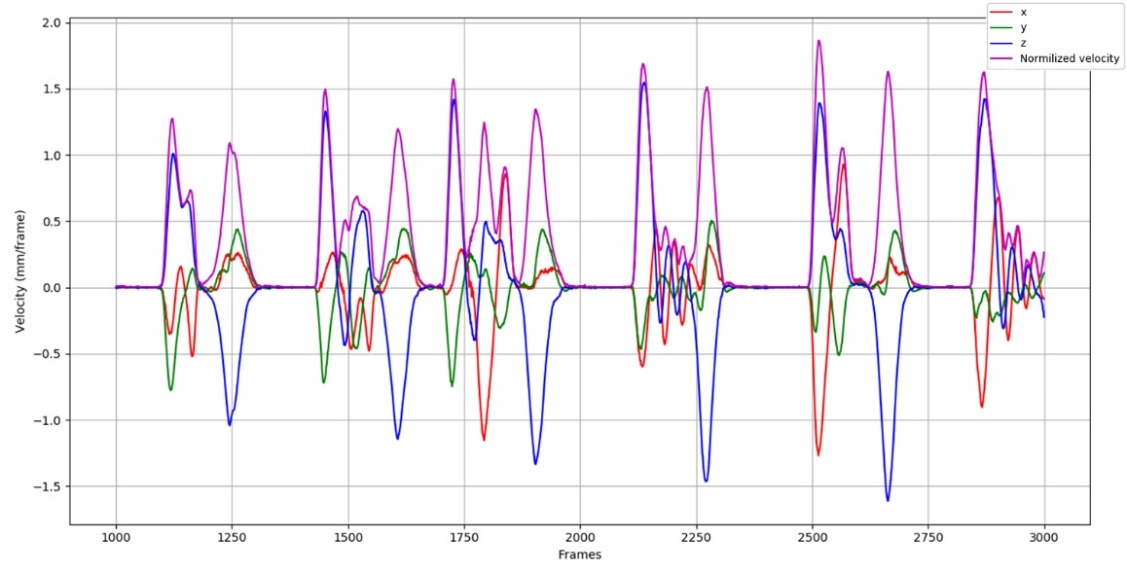
* **Velocity –** it is the change in displacement over time. It is calculated along the three axes.

Figure : Graph illustrating the Hand velocity with respect of time; The red, green, blue lines show the graphs for the 3 axes; the magenta line shows the normalized velocity

For the purpose of segmentation, I am more interested in the general movement of the hand rather than the movement along each axis, therefore a normalization is made by The Frobenius norm equation over each axis for each frame. *Figure 16* shows a graph of hand velocity along the tree axes and the normalized velocity over a period of time.

* **Acceleration -** describes how the velocity changes over time.

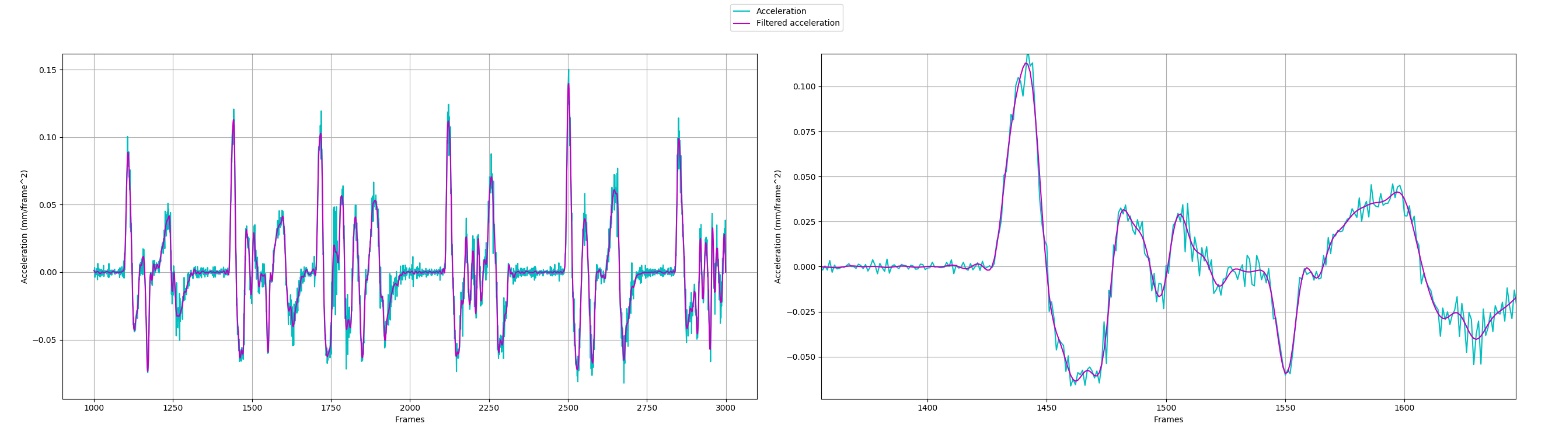


Figure : Graph illustrating Hand acceleration

There is another aspect that has to be considered here – noise. There are many reasons for noisy data and different source of noise. Although the data presented in the time graph in *Figure 15* seems quite smooth, calculating the acceleration greatly increases even the slight variations in it, as it can be seen on *Figure 17* (the blue line). Butterworth filter is applied and the result can be observed on *Figure 17* (purple line). The issue of data noise and filtering is described in more details in section 3.9.

The idea behind kinematic approach is finding the zero-crossings points in acceleration to identify significant changes in velocity. It is possible to find points of interest based on velocity extremums. Through experiments I found it more useful to base them on zero-crossing in acceleration.

The task of finding this points can be narrowed down to finding the points where the function changes its sign, meaning at which frame there are extremums in hand velocity function.

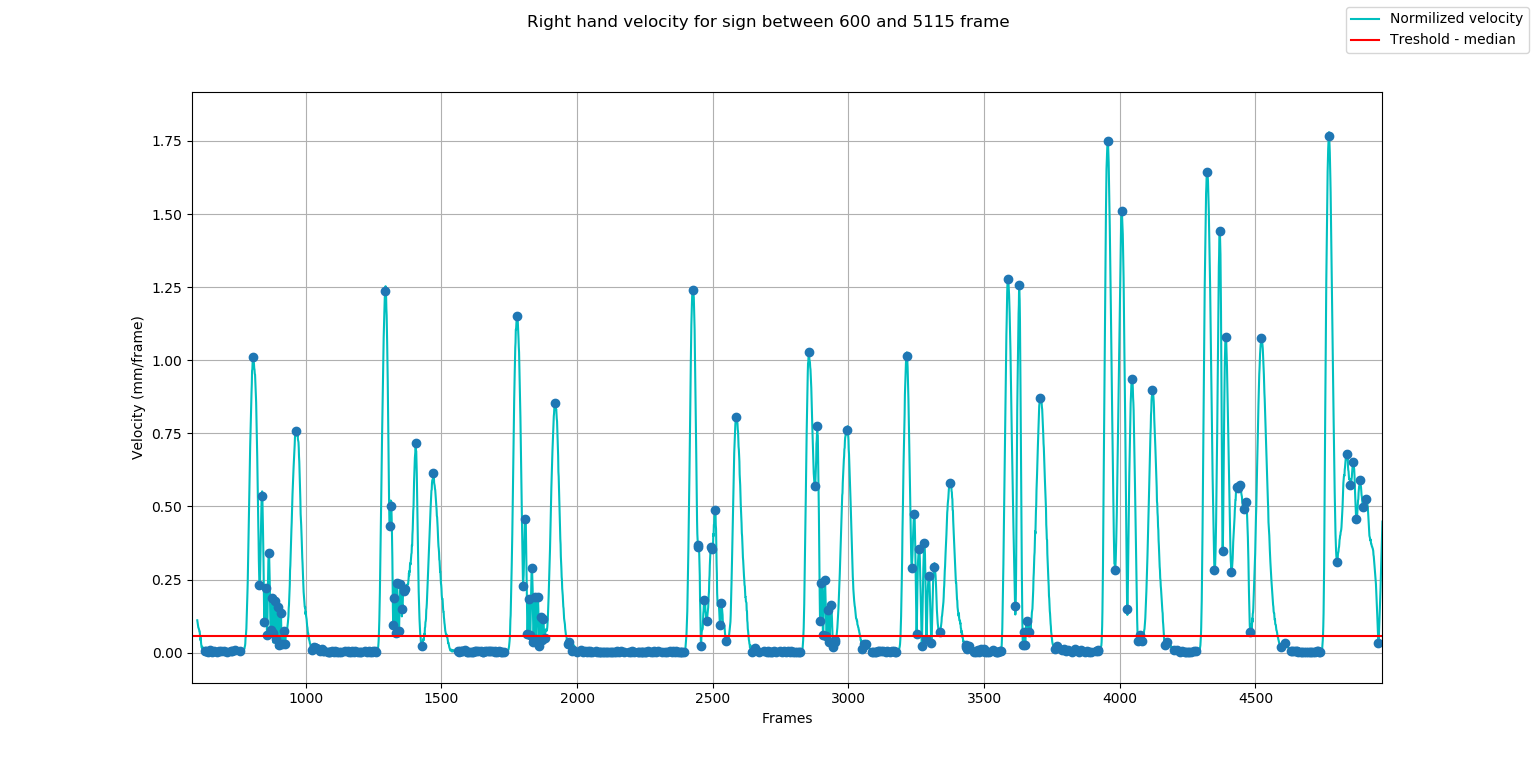
Next step is to decide which points are really releated with the beginning and end of the lexical item. First they are filtered by threshold function. The threshold value is chosen as median value of velocities. It gives a safe value, not to low to miss some point, neither too hight to have too many points. *Figure 18* shows an example of hand velocity during movement in a whole dictionary file, marked points of interest and the chosen initial theshold.

Figure :Graph of Hand velocity (blue line) with marked points of interest and initial threshold (red line)

At this graph the correlation between hand kinematics and points of interested is easeally obesrvable. The next step I s the process of labeling. It aims to destinguish this points at which the hand leaves rest pose and enters rest pose. Final result at this phase is pairs of frame’s number pointing to beginning and ending of each lexical item. Knowing the frames in which hands might be in RP, means that their beaviour during a well defined and almost static position, can be analyzed. This information then will be used to refine the threshold.

## Co-ordinate system

Before recording anything with motion capture system, the systems needs to be calibrated. At this process a co-ordinate system is set. Usually the origin of the absolute co-ordinate system is on the floor in the middle of the captured volume (as in the case of recordings for SL database). This world origin may differ, but it has to be set before the recording.

Before any processing of the data by this tool, a relative co-ordinate system is introduced for several reasons.

SL project is based on data-driven method, so it needs a rich and reliable set of data. It should consist performances of different signers. Meaning that further analysis should not depend of person’s physique, like height, weight, arm’s length, etc., or his position in the captured volume

In order the data to become invariant original co-ordinates are recalculated relatively to the skeleton centric. For the reasons mentioned above and the fact that it is a standard for the movie industry the middle of pelvis is chosen as a skeleton’s center.

## Noise and methods for filtering

Although motion capture systems are improving over the time and tends to provide precise and accurate data, there is still the issue with noisy data. It is one of the most significant challenges that researches have been studied. In general, under the term “noise“ is understood the unwanted modifications of the motion data. There might be several sources of noise.

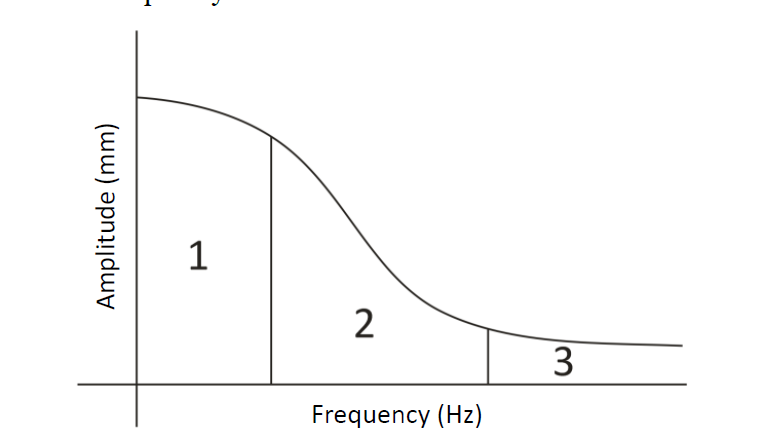
The nature of the noise and its linguistic relevance is reviewed by McDonald et al in their study for a prosodic analysis of a motion [[4](#_J._McDonald,_R.)].They analyze trajectories of different markers and use Fourier transformation to get signal’s spectrum in frequency domain. The frequency spectrum is divided into three sections which have different impact over sign analysis. According to their study based on medicals researches oscillations with frequencies higher than 10-12Hz (section marked with number 3 in *Figure 19*) cannot be produced by human muscles thus these frequencies can be considered as noise caused by recording technology or other external factors and can be safely eliminated with low-pass filter. Other frequencies may be produced by human motion, but not all of them have meaning for sign language. The cutoff frequency between section “1” and “2” may vary according to the type of motion. Human body may produce different types of motion. For example, fingers, because of their lower mass and smaller range of movements are capable of higher frequencies than hips. The study also presents a table of body parts and their cutoff frequencies. Another important finding in this research is that although frequencies in the section marked with “2” have no linguistic meaning, they are produced by human motion and therefore its important from realistic point of view. Further analysis of these frequencies can contribute to achieving more human-like animation.

Figure : Division of frequency domain

A crucial step in the process of segmentation is filtering the signal. For this purpose, a Python implementation of Butterworth filter was used. It is a type of signal processing filter, based on Fourier transform. Through experimentation and based on results from McDonald’s research, the cutoff frequency parameter of 12Hz was chosen.

The following figures shows acceleration in time domain and the resulting graph after applying the filter with different cutoff frequencies. *Figure 20* illustrates the result after applying the filter with cutoff frequency parameter equal to 2.

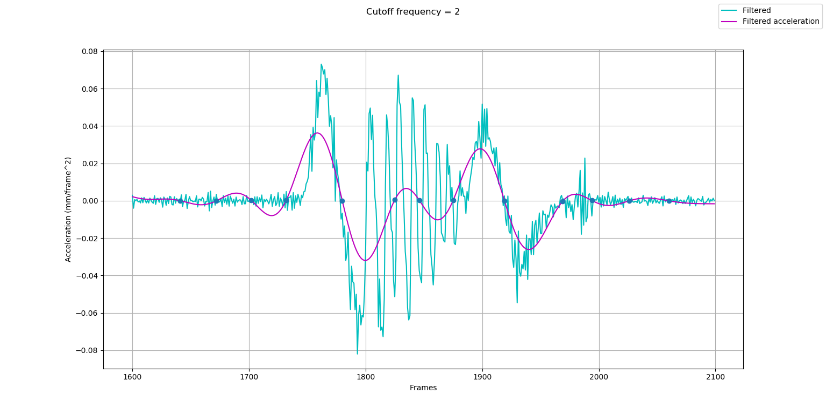


Figure : Graph of hand acceleration (blue line), resulting graph (magenta line) after applied Butterworth filter with Cutoff frequency = 2

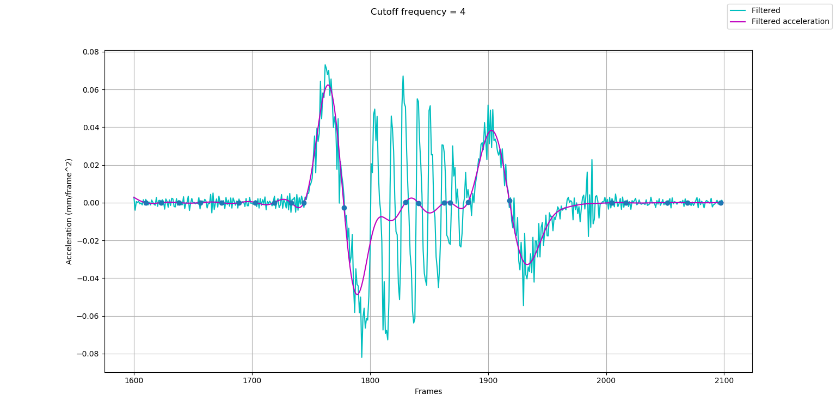
*Figure 21* shows the result where the cutoff frequency parameter is equal to 4.

Figure : Graph of hand acceleration (blue line), resulting graph (magenta line) after applied Butterworth filter with Cutoff frequency = 4

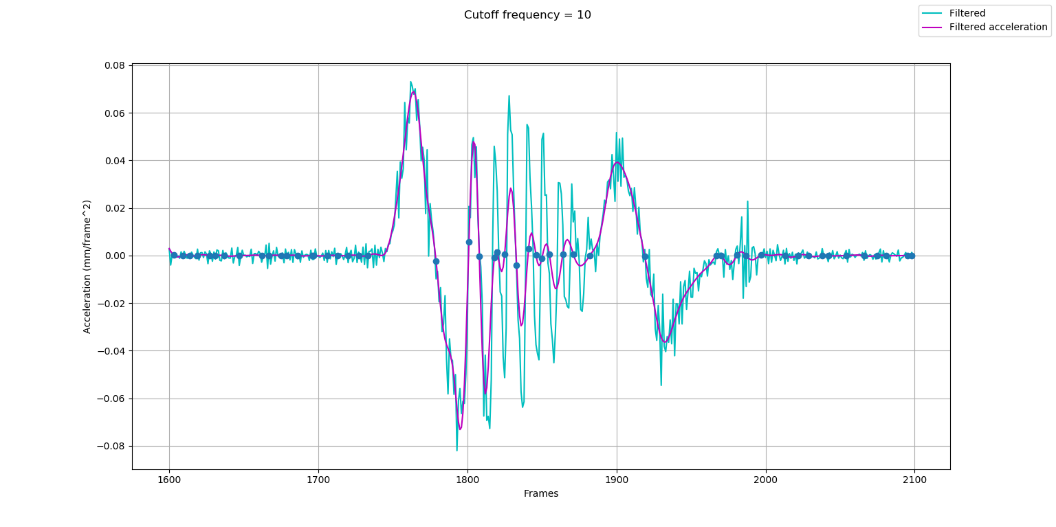


Figure : Graph of hand acceleration (blue line), resulting graph (magenta line) after applied Butterworth filter with Cutoff frequency = 10



Figure : Graph of hand acceleration (blue line), resulting graph (magenta line) after applied Butterworth filter with Cutoff frequency = 12

As discussed earlier according to McDonald’s research wrist limit frequency is 2Hz and fingers limit is never more than 4Hz. Compared to the results after filtering with cutoff frequency equal to 10Hz or 12Hz, I found the first two cases unsatisfying for the raw segmentation. Cutting the frequencies higher than this in section “1” cause too much loss of data and therefore may result in false segmentation

## Sign properties

Each lexical item is investigated in the meaning of HamNoSys, which was discussed earlier in section 2.2.1. The aim of this module is to extract properties of a sign, which describe it from motion point of view. The idea is these properties later to be used for sign classification.

The dominant hand can be easily determined by computing the total length of trajectory for both hands and compare the two measurements. The idea is that no matter if the sign is performed by one hand or both, the dominant hand’s length of the trajectory will be longer. Total distance of each hand can also be used to determine if the sign is one or two-handed, by simply comparing both measurements to threshold. The threshold is dynamically computed over the general movement.

### Hand Location

Hand location is determined for each frame relatively to the signer’s body plane. The space in front of the body is divided in regions according to HamNoSys (*Figure 24* shows an example of body plane sectioning). The exact boundaries of each region are defined dynamically for each frame, by the position of body markers. During the sign, hands can have different handshapes, so in order to be more accurate, each region has margin, which is also calculated for each frame. The method also keeps statistical information about the number of changes of the hand location and time spent in each region.

Figure : Example of hand location regions

### Hand Orientation

The task of defining Wrist and Palm orientation can be narrowed down to minor tasks in linear algebra. The idea is to represent body parts as 3-dimensional vectors.

According to HamNoSys wrist orientation is notated if the wrist is bend toward the pulse or back of the hand, toward the thumb or the little finger. The task of defining the wrist bending can be solved by measuring the angle between the plane of the hand and the plane of the arm.

As we know from linear algebra, a plane can be described by three non-collinear points. For example, body plane can be defined by LFSH, RFSH (markers on left and right shoulders) and one of the front waist markers (e.g. RFWT or LFWT). Therefore, two wrist markers and two hand markers are enough to define the hand plane in the space. And wrist markers (RIWR, ROWR) and arm marker (RWRE) are enough to define hand plane. This is useful when identifying Palm orientation.

Figure : Points defining the hand and arm planes

Important vector is the vector defined by the middle point of segment defined of the two wrist markers (e.g. RIWR and ROWR) and the middle point of the segment defined by the two hand markers (e.g. RIHAND and ROHAND). It is important, because this vector also describes the Extended Finger orientation. As it was explained earlier (see section 2.2.1.2) Palm orientation is two-step process. First has to be determined in which of the two Basic Position is the hand. This can be achieved by determining the Extended Finger orientation in the space according to the body plane.

### Handshape

Different hand shapes were discussed in section 2.2.1.1. Handshape aims to describe the complex configuration of the hand and all fingers.

At this point Handshape and Hand orientation modules are not included in the system output. They still need refinements. The description of hand orientation is tightly related to the specific handshape.

Furthermore, during the sign handshape and orientation may change, thus leading to the question at which exact moment the handshape and orientation should be investigated.

## SVM

Support vector machines (SVMs) are powerful and flexible class of supervised algorithms for classification. Supervised means that it needs data set which has been labeled in order to train a model. The model then can be used to cluster signs for further processing. This module is still not implemented in the tool, but it is included in the future work.

## Testing

The aim of this system is to provide reliable tool for automatic processing of the data retrieved from marker based optical motion capture system. This data will be specifically used for developing a signing avatar. Two types of evaluating of the system are planned. One to evaluate the trueness of the designed algorithms and one that proves the correct workflow of the system. The first one is described in section *4.6 Method evaluati*on. It is based on statistical comparison with the results from manual processing of the data. To test the correct workflow of the system, the following functional test are planned:

**Test Case 1: Performing raw segmentation**

This test aims to verify that a long stream of motion capture is segmented into smaller parts.

The input data is the data from recording session, recalculated with center of the origin between the waist markers.

The result should be a set of pair numbers, where the first number denotes a start frame, second number denotes – end frame.

**Test Case 2: One or two handed sign**

The aim of this test is to verify that for given sign it is correctly determined as one or two handed.

The input data is start and end frame of the sign and the motion data.

The output should be a message if the sign is two handed or one handed.

**Test Case 3: Hands location**

This test aims to verify if hands’ location during given sign is correctly determined.

The input data is start and end frame as well as the motion data.

The output should be 2 sequences of numbers for each hand, indicating the region in which the hand is located.

# System implementation

## Input data module

### File read

At this module is implemented a method for reading the input file. This method has one input parameter – file name. The method makes use of reader handler implemented in c3d python library. Besides extracting the number of all used markers in the file and the total number of frames, it reads and stores the 3-dimensional data in form of multidimensional array where first dimension is the number of the frame, second – marker index, third - the position along the three axes (x, y, z). It also creates a list with the labels of all used markers. At the end of this label list one artificial element is added to store the information for the origin of the Co-ordinate system according to the world origin.



### Co-ordinate system

The implementation of this method gives the ability to recalculate given *data* with new origin (*new\_origin*) of the co-ordinate system. This center can be defined by one point or by set of points, in this case the center is set in the middle between this points. The parameter *mlist* defines the set of all marker labels that are used in the set of *data.* There is one Boolean parameter, which is optional. It is used as a flag if the change of the Co-ordinate system is from relative to absolute or otherwise. This gives the functionality of going back to absolute co-ordinates.



The result of this method is a numpy array with the same dimensions as the input array, with recalculated co-ordinates.

### More active hand

This method measures the length of the trajectory for both hands by adding the absolute difference in the position for two consecutive frames. This distance is measured for both hands and arms for each direction (x, y, z). Then the resulting sums are averaged based on the total length of the given time period. Then they are compared to determine which hand is more active



## Hand marker set

*hand\_marker* is a helping method used from all modules. It basically interpreters the four markers placed on top of the hand as one hand marker. The calculation is made by the formula given at section 3.6, over the whole given set of data. The input parameter *h* denotes if the calculation should be made for right or left hand. The result is new artificial marker in the middleof the hand for each frame.



In order to ease the extraction of information needed for particular marker a method was created, where by given label name (*mname*) and list of marker (*mlist*) labels it returns the index of the marker.



## Segmentation module

### Hand velocity and acceleration

This method takes as parameters the period of time (expressed in the mean of frames), the original data and the hand (left or right) of interest as well as the set of markers’ labels. It calculates the first derivative with respect to time of the artificial hand marker and thus obtaining the velocity along the three axes. Then the built-in function of numpy library for normalization is used to reduce the function into a single scalar function of time.



Hand acceleration is obtained by computing the first derivative of the velocity. The only input parameter is the normalized velocity retrieved from the previous method.



### Zero-crossing

Points of interest are find by analyzing the scalar function of hand acceleration over time. This method searches for these frames where the graph of the function crosses the zero, by comparing the value of the acceleration of each two consequence frames. When change in the sign is noted, a comparison of the absolute value is made in order to be determined the better frame. A numpy array of these frames is returned.



### Labeling

The zero-crossing points determined from the previous method are indicating moments where there are extremums in the function of the velocity. The sequence of this points is further analyzed in order determine which of them indicate start and an end of a sign. This process is called of labeling. In this set of methods, a two steps of labeling are implemented. First frames at which velocity value is unther the threshold are marked with 0, others with 1. Then this labels are examin as vector of zeros and ones. The idea is that during the sign there should be a long sequence of ones. And if the start of a sign is at certan frame it will be precceeded by at least one zero and followed by at least one 1 (ex. 0011), then to denote this frame as start another label is added -1. Respectively if it is end, it would be fowolled by zero (100), it will be labeled with 0, every other case of sequences is labeled with -1. Of course there might be some false detections, because of the bad threshold value or halts and abrupt changes. The aim at this step is to determaine frames where at least one of the hands leaves rest-pose or where enters it. So a check of hands positoin is performed in order to precise the labeling and reduce the errors.

Then the label array is reviewed and the frames marked with 1 and zero from the second labeling are extracted. The final result is pairs of frame’s number pointing to beginning and ending of each lexical item, or moments where RP ends and next one begins.



### Rest pose analysis

Knowing frames in which hands might be in RP, means that their behaviour during a well defined and almost static position, can be analyzed and the threshold for labeling can be refined. At this component the hand velocity is calculated for those windows of time that are noted by the labeling proces as rest pose (from i-th end to i-th +1 start). The calculation is made for the more active hand. The new threshold is chosen as the maximum value of the hand velocity.



## Sign analysis module

### Real sign borders

At the Segmentation module the long sequence was separated by the frames where the hand leaves rest pose and then enters is. The real motion that has meaning for sign description is locked in between these frames. Another level of segmentation is performed for this shorter sequence of motion. The process of labeling is again a two steps process. The first step is same as before. At the second step this time are marked each local minimum - 1 and maximum - 0. A sign real start is defined by the first local minima. A sign real end is defined by the last local minimum before the last local maximum. The algorithm is designed in a way to always return a pair of frames. If no suitable frames are found it will return the frames from the raw segmentation.



### Dominant hand

The logic behind defining the dominant hand for particular sign is same as for the whole recording session.

### One-handed or two-handed sign

Defining if the sign is performed with one or two hands is based on the total measure of hand displacement for each hand and arm. Then these measurements are compared to a threshold. The threshold value is calculated dynamically for each segment.



### Hand location

The horizontal and vertical space in front of the signer is divided into 15 regions, numbered from 1 to 15. The region borders are defined by body markers position as described in section 3.10.1



Because during the signing hand may be in different hand shapes, to the region borders a margin is added. These margins are calculated for each frame.





Hand position is then compared to the regions borders and a single number from 1 to 15 is returned. This process is performed for each frame for the whole segment.



This method also calculates and returns the total number of time when the hand moves from one region to another.

## Output

The output up to this point is a human-readable representation of the extracted features. First shown result is the printed pairs of frame’s number and the total number of found signs. *Figure 26* shows example of the system output for raw segmentation from a test with a test file. The output includes the results before and after refining the threshold value.

At next steps each segment is analyzed individually. *Figure 27* example description of the first two identified signs. The output also includes a plot of the graphic of hand velocity and acceleration in this frame of time with marked points of the sign’s Real start and Real end. (*Figure 28*). Another plot displays the location of the right and left hand during the time defined from real start and real end (*Figure 29*)

Figure : System output from raw segmentation before and after refining the threshold value.



Figure : Part of system output for sign description.

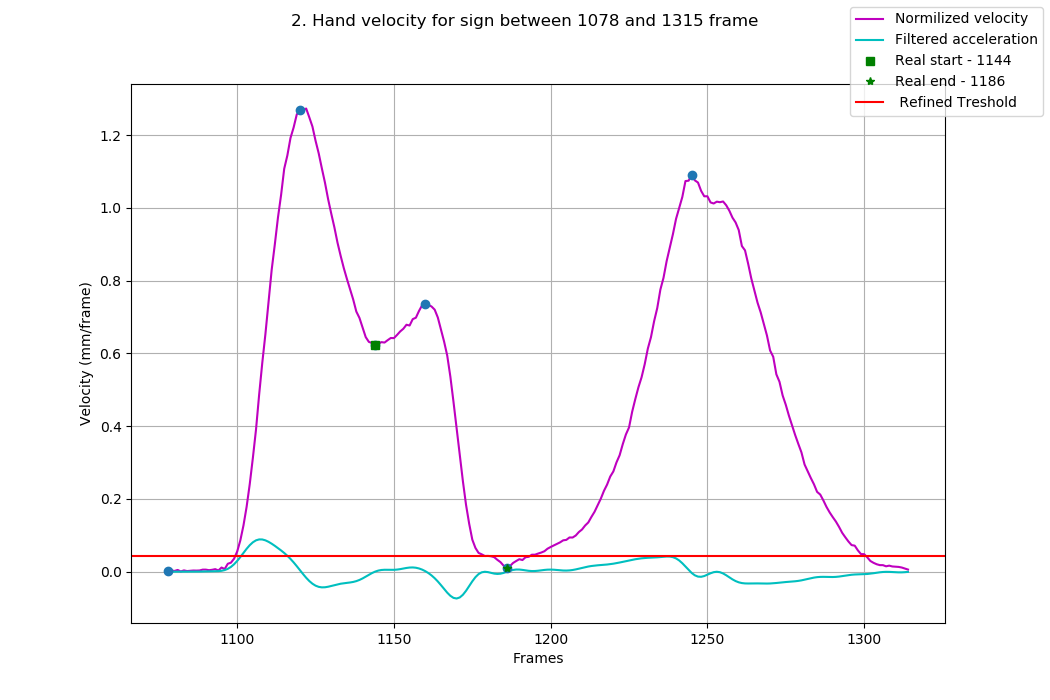


Figure :System output plot of Hand velocity and acceleration for second found sign

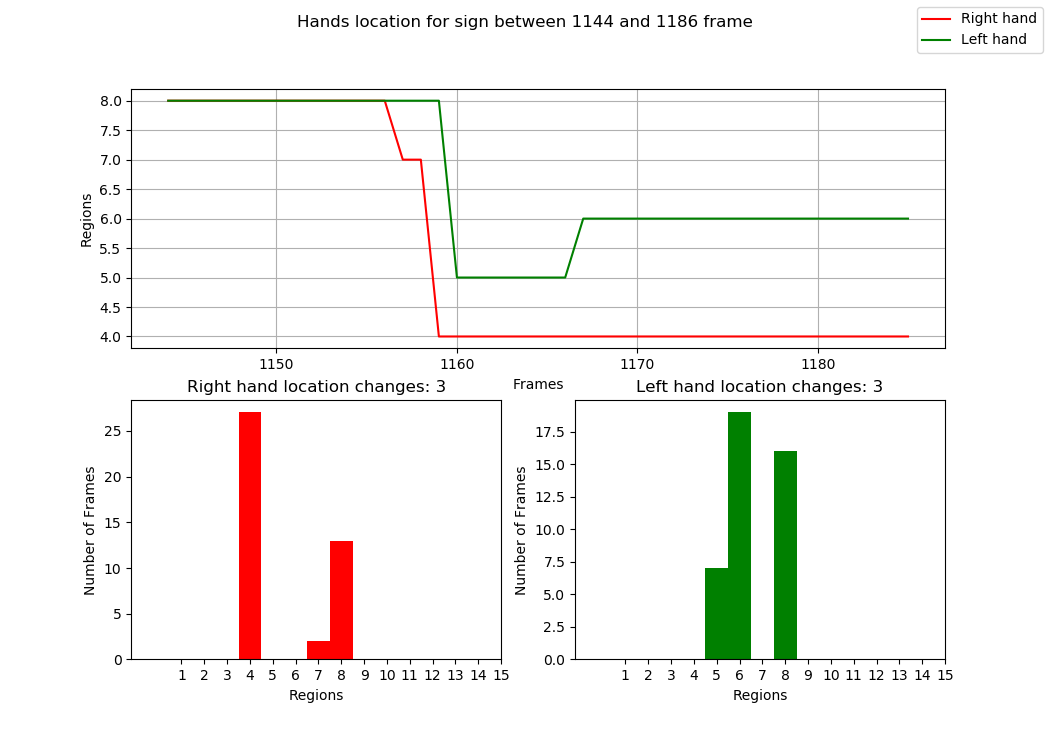


Figure : System output plot of right (in red) and left (in green) hand location

## Functional testing

In this section are described the last results after performing tests based on the planned test cases.

|  |  |
| --- | --- |
| Test case ID | TC1 |
| Test case name | Raw segmentation |
| Test case summary | Verifies that long stream of motion is segmented into smaller parts |
| Test data | 1. Input file projevy\_pocasi\_01\_ob\_rh\_lh\_b\_g\_face\_gaps.c3d  2. initialized variables:  start\_frame = 600  end\_frame = 7445 |
| Expected result | Final result should be a set of 19 pairs of numbers: |
|  | sign #1: 744 - 964  sign #2: 1102 - 1305  sign #3: 1433 - 1656  sign #4: 1707 - 1967  sign #5: 2117 - 2311  sign #6: 2493 - 2713  sign #7: 2846 - 3086  sign #8: 3501 - 3727  sign #9: 3796 - 3984  sign #10: 4097 - 4331  sign #11: 4439 - 4675  sign #12: 4820 - 5005  sign #13: 5154 - 5361  sign #14: 5442 - 5651  sign #15: 5826 - 6039  sign #16: 6203 - 6401  sign #17: 6483 - 6694  sign #18: 6857 - 7079  sign #19: 7146 - 7401 |
| Actual result | sign #1: 721 - 967  sign #2: 1078 - 1315  sign #3: 1427 - 1667  sign #4: 1702 - 1990  sign #5: 2109 - 2311  sign #6: 2484 - 2743  sign #7: 2831 - 3094  sign #8: 3498 - 3732  sign #9: 3773 - 3988  sign #10: 4090 - 4340  sign #11: 4435 - 4691  sign #12: 4799 - 4999  sign #13: 5137 - 5358  sign #14: 5422 - 5654  sign #15: 5812 - 6048  sign #16: 6196 - 6407  sign #17: 6473 - 6708  sign #18: 6840 - 7081  sign #19: 7131 - 7422 |

|  |  |
| --- | --- |
| Test case ID | TC2 |
| Test case name | One or two handed sign |
| Test case summary | Verifies if the sign is correctly determined as one or two handed. |
| Test data | 1. Input file projevy\_pocasi\_01\_ob\_rh\_lh\_b\_g\_face\_gaps.c3d  2. initialized variables:  starting\_frame = 782  end\_frame = 887 |
| Expected result | Two handed sign |
| Actual result | Two handed sign |

|  |  |
| --- | --- |
| Test case ID | TC3 |
| Test case name | Hands location |
| Test case summary | Verifies if the hands location is correctly determined. |
| Test data | 1. Input file projevy\_pocasi\_01\_ob\_rh\_lh\_b\_g\_face\_gaps.c3d  2. initialized variables:  starting\_frame = 1144  end\_frame = 1186 |
| Expected result | The regions of the right hand to be: 7, 4  Right hand location changes: 2  The regions of the left hand to be: 8, 6,  Left hand location changes: 2 |
| Actual result | The regions of the right hand to be: 8, 7, 4  Right hand location changes: 3  The regions of the left hand to be: 8, 5, 6,  Left hand location changes: 3 |

## Method evaluation

For this project testing was performed in order to prove validity and the precision of the methods used for analyzing input data. For this purpose, four dictionary files were used. Dictionary files contain recorded data on a weather thematic. Each file consists of number of lexical items.

Before proceeding to actual test, manual segmentation of the test files was performed. Results of the automatic segmentation are compared to those of the manual.

In the activity flow of the system two types of segmentation are performed. The first one is so-called “raw-segmentation” – determines two frames. *Start frame* is the moment when hand leaves the rest-pose (RP). *End frame* is the one when hand enters RP. Second type of segmentation defines exact *Start* and *End* frame when the real sign begins/ends.

Results from raw segmentation:

Start frame from automatic segmentation is evaluated as successful if it is between end frame of previous sign and start frame of the current sign, defined by the manual segmentation. Respectively, end frame determined of automatic segmentation should be between end frame of the current sign and start frame of the next sign, determined by the manual segmentation. The results show 100% success of defining start frame and 85.8% for defining the end of the sign.

When defining the start frame manually, it is annotated the frame where hand starts the actual motion from a still position, it is the moment where there is visible change in the trajectory. But human body is a complex mechanism and the hand needs preparation before starting to move. The system detects the start around 13 frames earlier, which shows that it is even more sensitive in this aspect than the human eye.

The results also show that detecting the end of the sign is harder than the start. This is due to the fact that the motion is performed by a real person, there is never actual stop. It is just a steady dropdown in the speed, that is why even two different people may not agree about the exact end. It also explains why the maximum error is 51 frames, while the maximum error when defining the start is 26 frames. Also the average error when determining the start is 13 frames while for end is 6 frames. The recording was performed at 120 frames per second, which means that the average error for raw segmentation is less than 0.10 seconds.

Table : Statistical evaluation of performed test for validating the automatic segmentation algorithm

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | success | average error per file (frames) | | | | average error | biggest error | |
|  |  | file 01 | file 02 | file 03 | file 04 |  | frames | seconds |
| raw segmentation | | | | | | | | |
| start tag | 100.00% | 12 | 11 | 13 | 16 | 13 | 26 | 0.216666667 |
| end tag | 85.81% | 11 | 12 | 9 | 19 | 13 | 51 | 0.425 |
| real sign boundaries | | | | | | | | |
| start tag | 76.00% | 6 | 5 | 5 | 5 | 5 | 36 | 0.3 |
| end tag | 58.60% | 8 | 6 | 7 | 7 | 7 | 52 | 0.433333333 |

In the study of the French sign language [[13](#_L._Naert,_C.)] they use the kinematic based segmentation to refine a manually segmented data. In their study they consider the 15 frames period (for a capture frequency of 100Hz, which corresponds to 0.15s of motion) difference between manual and automatic segmentation as relatively small and insignificant. Based on their research I found that difference of 13 frames is quite satisfying, but it has to be taken into account the fact that this results are over relatively small amount of data and further testing is needful.

Real sign boundaries:

The evaluation of the algorithm for identifying the real sign boundaries is made over the difference between manual and automatic segmentation. If the difference is smaller than a threshold value, then the result from automatic segmentation is considered as successful. The average error when identifying the start tag is 5 frames and 7 frames for end tag. Based on the average error the threshold of 6 frames is chosen. *Table 1* shows that the success in identifying sign’s real start is 92.08% and for the end – 87.04%. Which again proves that identifying the end of the motion is more challenging task than identifying the beginning.

# Conclusion

Sign language is the natural language for a people with speech and hearing impairments and their first mean of communication. Sign language vary from culture to culture and from region to region. Although it is the main alternative of spoken language for deaf community, both languages are very different from many points of view. The difference in linguistic aspect makes even the written form of spoken language difficult to understand, especially for prelingual deaf people. These barriers in communication and access to sources of information motivates many researches for SL synthesis. The aim of this works is developing tools for providing human –like movements while signing, which can be accepted from deaf community. Due to this fact and because SL utterance can be seen as continuous stream of motion, recent research based on data-driven synthesis and data from motion capture has been developed. Animations based on data from real signer are more realistic. A weak spot of this approach is that post-processing of MoCap data and skeleton reconstruction are laborious and time-consuming processes, as well as manual segmentation and annotation of SL utterance.

My work aims to develop a tool that can help in processing the data for SL database. I reviewed Hamburg notation system as a way to understand SL from phonetic point of view. My work is focused on hand movement and its kinematic features such as position and its derivatives (velocity, acceleration) in order to decompose sign and extract properties that can describe the sign and help for sign classification.

First step is implementing an algorithm for segmentation of SL dictionary file. The aim is identifying starting and ending moment of a lexical item. The algorithm is based on analysis of hand trajectory and velocity. Unlike speech in which transitions between words is silence, movement never stops. However, there is phase where hand prepares for execution of the sign, and this phase may be indicated by drops in velocity. Values are processed with low-pass filter to prevent the algorithm from detecting incidental minima due to noise in data.

After proper segmenting of the data and extracting only the motion that will have semantic meaning, is possible to proceed with further analysis. At next steps the meaningful parts of the sign are studied. The algorithm aims to extract sign features such as Hand location, Hand orientation and dominant hand.

For my future work I plan to extended the number of sign features extracted from the motion analysis and to utilize computer learning methods to cluster signs for further processing.

# Bibliography

#### T. Hanke, “Hamnosys-representing sign language data in language resources and language processing contexts," in LREC, vol. 4, pp. 1-6, 2004.

#### Stokoe, William C. (1960, 1978), “Sign language structure: An outline of the visual communication systems of the American deaf," Studies in linguistics, Occasional papers, No. 8, Dept. of Anthropology and Linguistics, University at Buffalo. 2d ed., Silver Spring: Md: Linstok Press.

#### Mihoko Kato, “A Study of Notation and Sign Writing Systems for the Deaf ", Intercultural Communication Studies XVII: 4 2008, Toyohashi University of Technology

#### J. McDonald, R. Wolfe, R. B. Wilbur, R. Moncrief, E. Malaia, S. Fujimoto, S. Baowidan, and J. Stec, “A new tool to facilitate prosodic analysis of motion capture data and a data- driven technique for the improvement of avatar motion," 2016.

#### E. Muybridge, “Muybridge's Complete Human and Animal Locomotion”. Collections of ne art in Dover books, Dover Publications, 1979

#### “Go Further with Vicon MX T-Series” Vicon Motion System Limited, rev. 1.3 August, 2010

#### N. Wheatland, Y. Wang, H. Song, M. Neff, V. Zordan, and S. Jörg, “State of the art in hand and finger modeling and animation” Computer Graphics Forum, vol. 34, no. 2, pp. 735–760, 2015

#### M. Dilsizian, Z. Tang, D. Metaxas, M. Huenerfauth, and C. Neidle, “The importance of 3D motion trajectories for computer-based sign recognition" 7th Workshop on the Representation and Processing of Sign Languages: Corpus Mining, The 10th International Conference on Language Resources and Evaluation (LREC 2016), 2016.

#### T. Hanke, S. Matthes, A. Regen, and S. Worseck, “Where does a sign start and end? segmentation of continuous signing" 5th Workshop on the Representation and Processing of Sign Languages: Interactions between Corpus and Lexicon, Language Resources and Evaluation (LREC 2012), 05 2012.

#### Perlmutter, David M., "What is Sign Language?" LSA, 12 April 2014. Retrieved 4 November 2013.

#### Sandler, Wendy; & Lillo-Martin, Diane., “Sign Language and Linguistic Universals”, 2006. Cambridge: Cambridge University Press.

#### P. Jedlicka, Z. Krnoul, and M. Zelezny, “Methods for recognizing interesting events within sign language motion capture data” in Universal Access in Human-Computer Interaction. Designing Novel Interactions (M. Antona and C. Stephanidis, eds.), (Cham), pp. 55-75, Springer International Publishing, 2017.

#### L. Naert, C. Larboulette, and S. Gibet, “Coarticulation analysis for sign language synthesis” in 7th Workshop on the Representation and Processing of Sign Languages, LREC, 2016.

#### Python documentation, https://docs.python.org/3/library/index.html

#### “The C3D file format”, MotionLab Systems, Inc., 2002

#### F. Bevilacqua, J. Ridenour, D. J. Cuccia, “3D motion capture data: motion analysis and mapping to music”

#### <ViSiCAST project, http://www.visicast.cmp.uea.ac.uk/Public.html>

#### S. Gibet, N. Courty, K. Duarte, and T. L. Naour, “The SignCom system for data-driven animation of interactive virtual signers: methodology and evaluation,” ACM Transactions on Interactive Intelligent Systems (TiiS), vol. 1, no. 1, p. 6, 2011.

#### S. Jörg, J. Hodgins, and A. Safonova, “Data-driven finger motion synthesis for gesturing characters,” ACM Transactions on Graphics, vol. 31, no. 6, p. 1, 2012

#### eSIGN project, http://www.visicast.cmp.uea.ac.uk/eSIGN/Summary/Summary.htm

#### U. Ehrhardt, B. Davies, N. Thomas, M. Sheard, J. Glauert, R. Elliott, J. Tryggvason, T. Hanke, C. Schmaling, M. Wells, I. Zwitserlood, “Animating Sign Language: The eSIGN Approach,” 2004

#### J.-B. Kim, K.-H. Park, W.-C. Bang, and Z.Z. Bien. Continuous Korean sign language recognition using gesture segmentation and Hidden Markov Model. Proceedings of the 2002 IEEE International Conference on Fuzzy Systems, 2001

#### S. Gibet, F. Lefebvre-Albaret, L. Hamon, R. Brun, and A. Turki. “Interactive editing in French sign language dedicated to virtual signers: requirements and challenges.” Universal Access in the Information Society, 15(4):525–539, 2016.

#### Sign3D, <http://www.mocaplab.com/projects/sign-3d/>

#### Gestural Control of Music, http://music.arts.uci.edu/dobrian/motioncapture/examples.htm

#### Bevilacqua F., J. Ridenour, and D. Cuccia, “Mapping Music to Gesture: A study using 3D motion capture data”, Proceedings of the Workshop/Symposium on Sensing and Input for Media-centric Systems, Santa Barbara CA, 2002.

#### Bouchard, D., & Badler, N. I. (2015). “Segmenting Motion Capture Data Using a Qualitative Analysis”. Proceedings of the 8th ACM SIGGRAPH Conference on Motion in Games, 23-30

#### J. Glauert, R. Kennaway, R. Elliott, “Virtual Human Signing as Expressive Animation”