

Institute for Visualization and Interactive Systems
University of Stuttgart
Universitätsstraße 38
70569 Stuttgart
Germany

Bachelor Thesis Nr. 163

Ego perspective video indexing for life logging videos

Mathias Landwehr

Course of Study:	Software Engineering
Examiner:	Prof. Dr. Albrecht Schmidt
Supervisor:	Dipl.–Des. Dipl.-Komm.-Wirt Katrin Wolf M.Sc. Yomna Abdelrahman
Commenced:	10.07.2014
Completed:	09.01.2015
CR-Classification:	H.3.1

Kurzfassung

Diese Arbeit befasst sich mit Lifelogging Videos, die mit auf dem Kopf getragenen Geräten aufgenommen wurden. Das Ziel ist es eine Methode zu entwickeln, um wichtige Teile aus einem Lifelogging Video heraus zu filtern. Das bedeutet, dass wir herausfinden müssen welche Teile eines Videos überhaupt als wichtig erachtet werden. Um die Wichtigkeit einzelner Videoabschnitte festzulegen, müssen wir herausfinden wie das autobiographische Gedächtnis¹ funktioniert, um einen indexing Mechanismus zu erstellen, der auf ähnliche Weise funktioniert. Um die Videos mit verschiedenen Informationen zu indexen müssen zunächst diese Informationen aus dem Video selber gewonnen werden. Da Gesichter ein wichtiger Teil des autobiographischen Gedächtnisses sind, wird image processing benutzt, um Gesichter aus den Videos zu erkennen. Zusätzlich können wir die GPS Daten benutzen um den Ort zu bestimmen. Nachdem die ganzen Informationen gesammelt wurden, werden sie in sogenannten Events gespeichert. Für jedes Event muss definiert werden, welche Personen an welchem Ort zu welcher Zeit auftauchen. Um eine gute Zusammensetzung von Events zu gewährleisten wurde ein Prototyp entwickelt um Lifelogging Videos in kleinere Segmente aufzuteilen, die momentan nur auf Gesichtern, Orten und Zeit beruhen. Dieser Prototyp kann in Zukunft beliebig erweitert und verbessert werden. Dieses Projekt dient als Grundlage für die spätere Entwicklung eines geeigneten Lifelogging Navigationstools.

¹ <http://www.sciencedirect.com/science/article/pii/S0079742108604521> 06.01.2015

Abstract

This thesis deals with life logging videos that are recorded by head worn devices. The goal is to develop a method to filter out parts of life logging videos which are important. This means it is to determine which parts are important. To do this we take a look at how the autobiographical memory¹ works and try to adapt an indexing mechanism which works on similar aspects. To index life logging videos with the expressive metadata successfully we first need to extract information out of the video itself. Since faces are an important part of autobiographical memory recall, image processing which consists of face detection, tracking and recognition is used. This helps to get the people in a scene. Another part is the location data which is accessed by using GPS data. After all the information is gathered we can index those information in so called events. For each event we have to define the people that are present during this event, which place and at what time the event takes place. To do this an indexing algorithm was developed which segments the video into smaller parts by using the faces, location and time. The result is a prototype algorithm which can be further developed to improve the actual segmentation of life logging videos. This project serves as an information collecting and creation application for future life logging video navigation tools.

Table of contents

Kurzfassung	1
Abstract.....	2
Table of figures and tables.....	4
1 Introduction.....	5
2 Related Work	7
3 Concept of video indexing.....	15
4 Implementation	21
4.1 Information Extracting	22
4.1.1 Face Detection, recognition and tracking	23
4.1.2 GPS data extraction.....	33
4.2 Storage of indexing information	34
4.3 Implementation of segmentation.....	37
5 User Study.....	41
5.1 Aim.....	41
5.2 Expected outcome	41
5.3 Method	42
5.4 Participants	42
5.5 Apparatus	42
5.6 Tasks.....	46
5.7 Measurements.....	46
5.8 Procedure.....	46
5.9 Design.....	46
5.10 Results	47
5.10.1 Scenario 1: Dialogue.....	49
5.10.2 Scenario 2: Walking.....	51
5.10.3 Scenario 3: Meeting	53
5.10.4 Scenario 4: Eating	55
5.11 Discussion	57
6 Summary and Future Work.....	63
References.....	66
Declaration.....	69

Table of figures and tables

Figure 1. Gopro http://ecx.images-amazon.com/images/I/41aE7Oejq2L.jpg .06.01.2015...	5
Figure 2. Google Glasses http://www5.pcmag.com/media/images/354883-google-glass.jpg 06.01.2015.....	5
Figure 3. Named Entity View, Christel, M. G. 2008	8
Figure 4. General Segmentation functionality	16
Figure 5. Implementation overview	21
Figure 6. Information extraction	22
Figure 7. Frame processing.....	24
Figure 8. Haar feature detection (picture of lena included in the openCV library)	26
Figure 9. Local binary pattern.....	28
Figure 10. Local binary histograms	28
Figure 11. Tracking algorithm overview	32
Figure 12. Location indexing overview	38
Figure 13. Face indexing overview.....	40
Figure 14. Scenario 1	43
Figure 15. Scenario 2	43
Figure 16. Scenario 3	44
Figure 17. Scenario 4	44
Figure 18. Apparatus overview	45
Figure 19. Face selection screen	45
Figure 20. Scenario1: Reasons used	49
Figure 21. Scenario 1: Importance rating	50
Figure 22. Scenario 2: Reasons used	51
Figure 23. Scenario 2: Importance ranking.....	52
Figure 24. Scenario 3: Reasons used	53
Figure 25. Scenario 3: Importance Ranking	54
Figure 26. Scenario 4: Reasons used	55
Figure 27. Scenario 4: Importance ranking.....	56
Table 1. Overview of indexing methods.....	11
Table 2. Pros and cons of face tracker	30

1 Introduction

Recording life logging videos has become more and more important for people. First of all I like to explain what life logging is in general. Life logging describes the act of recording parts of one's life. This could include some kind of medical measurements from devices that are worn or recording videos of the surroundings. In this thesis we concentrate on the video taking part where people are walking around with cameras and record all their life. Regarding that there are new devices like google glasses and other head worn devices it is becoming much easier for people to record those kind of videos since they do not have to hold the camera but instead they can just wear them on their head.



Figure 1. Gopro



Figure 2. Google Glasses

Those devices will probably become a common thing in the future. Because you can easily record videos with these devices the amount of video material will probably increase too. The problem with this is that when people want to re watch some of their recorded memories it can be really hard to find those parts again. For that purpose it is important to have a browsing tool to easily find the video parts they want to see.

There are already a lot of browsing tools for all kinds of video types. The main problem with life logging videos is that they are different from professional recorded videos like movies or TV shows. The first and probably most important issue is the camera movement and the environment. Professional videos are usually recorded in a special studio where you can change the lighting and camera placement just so it looks the best. Another issue is the segments in which videos are recorded. In professional videos there are always cuts and the view also alternates between different cameras to give a better overview of the scene. In addition the cameras are mostly static. But even when the camera moves it is a smooth movement. Additionally the professional actors concentrate themselves towards the cameras so it makes image processing in terms of face recognition much easier. All of those issues usually do not apply to videos recorded by head worn devices. We cannot control the lighting or the environment. Also the camera moves almost all the time since our heads are rarely standing still. Because of those problems it is not possible to segment a life logging video like a movie or TV show. That said it is important to look at the information we can get in order to segment those type of videos. In this case a segment would be a part of the video which helps the user to remember the situation better. This could be

for example, a dialogue with another person or a group event like going to a party. Humans generally tend to remember specific events, here segments, by linking the information they got on this event together. Those information are mostly based on time, people, place, objects, tasks and emotions. The focus in this thesis lies on the faces. Time and places are also covered but they are very easy to handle and do not require that much effort unlike the faces. Objects, tasks and emotions are not covered here and can be considered in the future. The goal is to develop a meta-concept to segment the videos into smaller parts by the indexing information we can get. As mentioned the main focus will lie on the person based segmentation together with places and time. To identify the faces we can use face detection and recognition just like in professional videos. The problem is that the quality can be much worse so it is harder to actually detect all faces in the video. Another problem is the higher number of false positives. But besides all of those issues we will use the normal face detection and recognition available. Identifying the location is much easier since we can use the GPS data from the recording devices. Detecting activities is a more difficult issue which is not covered in this thesis.

The thesis consists of the following parts. First I will discuss some related work to show what has already been done on this topic. Then I will give an overview of the idea behind the indexing and the information extraction and how this can be implemented. The last parts are the experiment I conducted to see if the segmenting algorithm selects important faces in a way a normal person would do it. At the end I will conclude with a summary over the Bachelor thesis and discuss some future work.

2 Related Work

Indexing videos always goes together with an appropriate browsing tool. Packing Videos in a compact layout and divide them into smaller segments is a common task nowadays. Countless of research papers has been written about this subject. But most of them focus on different video genres, like video surveillance, movies, TV series or news videos. We, however, focus on first person life capture videos. But there are still many helpful approaches that can be useful.

First I will summarize some browsing methods which help to view the indexed information.

The most common layouts are the timeline and the storyboard. Timelines are used the most, like in (Haesen, M. et al., 2013; Christel, M. G., 2008; Nunes, M. et al., 2006). Here they are used to give an overview over a certain period of time in which videos can be viewed chronologically.

Storyboards are also a very easy but also very clear presentation. Those are mainly treated by Haesen, M. et al. (2013) and Jackson, O. et al. (2013). Though the latter one are using them to show loops instead of static key frames. Additionally the individual loops are moving along a timeline, so that the user can watch the whole video without changing his focus from one loop to another.

While timelines are mainly focusing on the chronologically order of events storyboards are more useful to give an overview over the whole video by using indexed information.

Boreczky, J. et al. (2000) and Uchihashi, S. et al. (1999) developed an extension of storyboards which was also used later by Chiu, P. et al. (2005). Here the individual key frames are sized based on their importance. That means that more important frames are displayed larger like in a comic book or a manga. This is useful to set the focus of the user to more important events. Chiu, P. et al. (2005) utilized this technique in their 3 dimensional city in which every façade of a building gives an overview of the video like in a comic book or manga. This approach is also one of the more rare 3D presentations of videos.

Additionally to the already named presentations there exists other ones. These are mainly focusing on the content of the videos. For example one can map the content of the videos to the location in which they were recorded. Techniques to extract the content of a video via speech recognition were conducted by (Haesen, M. et al., 2013; Christel, M. G., 2008; Snoek, C. G. M. and Worring, M., 2005). In Addition to the already mentioned presentations there also exists other layouts like the Map View, the Vibe View and the Named Entity View, which are all treated by Christel, M. G. (2008). The Map View is a geographically map on which the locations, which are named in the video or where the video itself was recorded, are marked. The Vibe View shows different topics, which are mentioned in videos and assign the individual videos to the topics in a 2 dimensional view. Although more than one topic can be covered by a video which simply means that the video will be coordinated between those topics. Last but not least there is the Named Entity View. This one looks like a simple mind map. In this presentation individual topics, persons, and locations are added as different entities and connected if there is a connection between them in a video.

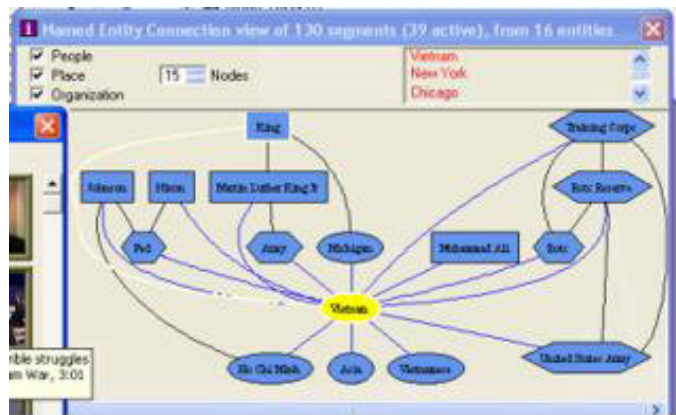


Figure 3. Named Entity View, Christel, M. G. 2008

This presentation is very interesting since in our approach also uses relations between different persons, locations and events. In this paper the user can show the videos in all of the above mentioned presentations but for every presentation a new window is opened. In this thesis we concentrate all possible given information that can be potentially shown in a 3 dimensional view which combines all of the named views above.

The different layouts are good to show whole videos in a big presentation. But what if we want to show only little segments of a video in an easy to understand presentation? For this approach there are basically 2 possible options. The first one is the so called slit scan method. It is basically a sequence of pixel wide cutouts of a sequence of frames. In Timeline: Video Traces for Awareness (Nunes, M. et al., 2006) this technique is used to monitor a specific area using a static camera like a webcam. For instance one can monitor a door to see when a person enters or leaves the room. The second one is the use of transparency and blur effects which are shown in Multi-frame video representation using feature preserving directional blur (Yamauchi, Y., 2007). In this paper the frames which already have been shown and the ones which are yet to be seen are made transparent and blurred out, leaving only the frame which is focused in a clear view. This technique is useful to get an overview over a short segment of a video.

The topic of what, when and how individual videos are segmented into parts is also covered in a lot of papers. The methods to index videos is heavily based on the genre of the videos. An overview of the methods and how to index is shown in Multimodal Video Indexing: A Review of the State-of-the-art (Snoek, C. G. M. and Worring, M., 2005). The main segmentation is mostly based on so called shots, which is simply a coherently shot of a single camera. In a news video for example it is easier to detect shots and using audio recognition techniques than in a home video. To detect shots Wujie Zhang, J. et al. (2004) are using a set of detectors. These are the fade in and out detector, the cut detector and the gradient transition detector. Those techniques are extremely hard to realize in ego perspective recorded videos because they can be several hours long and they usually don't have cuts or transitions. In Addition there can be parts in which there occurs no talking which makes it also difficult to use audio recognition. This is also the reason why face recognition is much more useful in home videos. Despite this fact Ma, W. and Zhang, H. are still looking for shots. But here these aren't based only on cuts and transitions but also on an indirect camera change, like when the camera itself is changing its direction.

In order to index information of the faces in a video, Cast indexing for videos by NCuts and page ranking (Gao, Y. et al., 2007) introduce a new technique to identify main characters and their relationship between other characters in a series, movies or even home videos. Those relationships are also important for this thesis since we want to show the connections between different people and locations.

For face recognition itself there exists a vast variety of algorithms. Gao, Y. et al. (2007) are using a recognition technique based on neural network which is able to detect, recognize and track frontal faces which are rotated in image plane.

A very interesting approach on face recognition was approached by Krishna, S. et. al. (2005) were trying to develop a robust face recognition algorithm for Individuals with visual impairments. They also encountered the problem with the difference of videos which were recorded by a head worn device like different angles of faces an illumination. They tested different algorithms to see which one was the most suitable for face recognition in a real-world environment.

An additional method to index the content of a video is the gathering of location data. In Creating map-based storyboards for browsing tour videos (Pongnumkul, S. et al., 2008) the user has to manually upload a map in which he can mark the separate locations at which he recorded the video. This method doesn't apply to us since in a worst case scenario the user has to upload and edit the same places over and over. Another possibility is described by Xu, Q. et al. (2010). Here the background is separated from the foreground and then analyzed by the use of Distinctive Image Features from Scale-Invariant Keypoints (Lowe, D. G., 2004) and compared to other frames afterwards. This method is also not sufficient enough for us since vacation videos are normally not recorded on the same place which means that there is a vast amount of backgrounds which in worst case looks the same as in another location which would produce false positives. The safest method would without a doubt be the use of GPS data like in HUGVid (Ma, H. et al., 2012). Here the location is gathered from the recording device, like a smartphone. The main problem lies in the absence of GPS data which can occur if the GPS sensor is deactivated or the device cannot be sensed. In addition the GPS data are not 100% accurate most of the time. However this method is still the best way to go.

To present and view personal histories and vacation videos a novel approach was proposed by Al-Hajri, A. et.al. (2014). Here the focus lies on the video sequences which the user watches the most. This is a good idea since the user himself knows best what he wants to see and what not. The problem is that the video material can be several years long which means it is pretty hard to get an entry point to look for specific events which the user wants to watch.

Recording the personal life has also become a greater issue over the years. Life logging is useful to counter the weakness of the human memory. By recording his own life a person is able to look up specific details which he cannot remember any more. For example simple things, like remembering where the car keys are or where a specific photo from the last vacation is, can be looked up very easy in database applications like LifeByBits (Bell, G. et al., 2006). Here a huge amount of information a human is capable of doing can be saved in a database. This counts for documents, messages, phone calls, videos, photos, music files, interaction done on the pc and so on. The reason for saving all this information is because the human memory can recall things easier by remembering the context in which the information the human wants to remember has taken place. For example, the user wants

to look up a specific photo of his last vacation. He doesn't know the name of the photo or where he has saved it, but he does know, what day it was and that he looked at it after a phone call from a specific friend. Now he just has to search all the phone calls with that person on a specific day and can additionally search for a specific time period after this phone call to see his interaction. By doing this he can see which pictures he had opened in that time period and is more likely to find the specific photo he wants to see.

It is clear that the context is very important to remember things. But to be able to get a huge amounts of links between documents, time, interactions and so on it is necessary to record every little detail of a person's life.

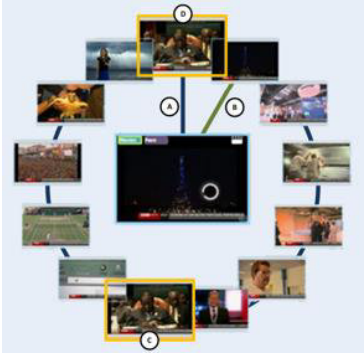
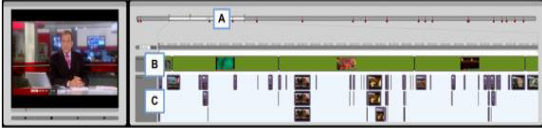

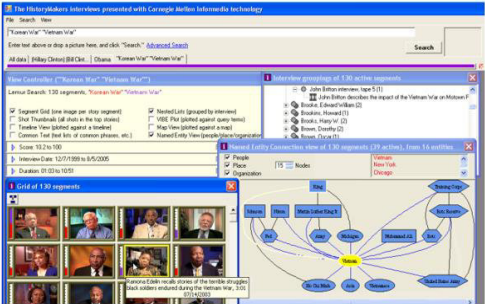
Jones, G. J. F. and Chen, Y. (2010) also provide an interface for searching through all kinds of data stored. They tried to create an easy to use searching tool which can be used by all kind of different people. They also describe a guideline on how to develop a life log application.

In our application we focus only on video material, which is also covered by the two applications named above, but they do not segment their videos in more interesting parts. In LifeByBits there does not even exists a face recognition feature yet.


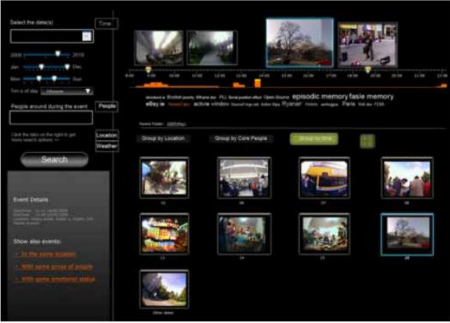
The main problem with head worn devices is the movement of the head itself. Even simple activities like walking already correspond in worse video quality. To counter this problem Kopf, J. et. al. (2014) developed a method to convert those kind of videos into hyper-lapse videos, i.e., time-lapse videos with a smoothly moving camera. They first compute the 3D camera input path and calculate an improved, smoothed path for the camera of the output video. Then they put the output video together by using image based rendering. The result is a smooth video without all the head shaking that happens normally when you are walking or cycling.

Another problem of head worn life logging devices is the huge amount of data that is recorded. Most of the material isn't important at all and is also most likely not to be watched again because nothing interesting happens. This problem was addressed by Aghazadeh, O, et. al. (2011). They used novelty detection to detect interesting parts in videos. In their approach they used deviation from background as a heuristic to detect novelty. Ghosh Joydeep (2012) also approached this problem. But instead of looking at deviation in the background, he was concentrating on important faces and objects with which the camera wearer interacts. This is achieved by using object detection to look for hand positioning and the interaction with different objects.

Table 1. Overview of indexing methods

Interface	Reference	Concept	Indexing method
	<p>Haesen, M. J. et al. 2013</p>	<p>Clock, Timeline</p>	<p>Shot detection, speaker pause, textual analysis</p>
	<p>Boreczky, J. et al. 2000</p>	<p>Storyboard</p>	<p>Shot detection, frame clustering</p>
	<p>Uchihashi, S. et al. 1999</p>	<p>Storyboard</p>	<p>Shot detection, frame clustering</p>
	<p>Christel, M. G. 2008</p>	<p>Storyboard, Timeline, TextLabel (Mapview, Text view, Vibe View, Named Entity View)</p>	<p>manual indexing</p>

	<p>Chiu, P. et al. 2005</p>	<p>Storyboard</p>	<p>-</p>
	<p>Jackson, D. et al. 2013</p>	<p>Storyboard, (video loops)</p>	<p>Time</p>
	<p>Nunes, M. et al. 2006</p>	<p>Timeline</p>	<p>Time</p>
	<p>Al-Hajri, A. et al. 2014</p>	<p>Timeline</p>	<p>Time, user history</p>

	<p>Gemmell, J. et al., 2006</p>	<p>Timeline, storyboard, maps, textlabels (textview), Cluster (time view)</p>	<p>Background analysis</p>
	<p>Chen, Y., and Jones, G. J. F. 2010</p>	<p>Storyboard, Timeline, textlabels (textview)</p>	<p>-</p>

In conclusion it can be said that there has already been done a lot of work on this topic. The only thing that has been barely covered is the indexing of life logging videos with general more video material than normal videos like movies etc. Not many concentrate on life logging in the first place. Additionally there is no practical browsing tool which supports the human memory in a considerable way. Therefore it would be nice to have an indexing tool which supports the human memory and provides all necessary information.

3 Concept of video indexing

The final goal is to create events which contain all the necessary information of what is happening in the current segment of the video. That information then can be used to create a browsing tool in which the user wants to navigate through his recordings. So first of all I have to explain what an event is.

An event is basically a part of the original video. In the end, every second of the video will belong into one event. For an easier understanding of what an event is it can be compared to the scenes in professional videos. Even an unimportant part of the video is considered an event. For example if the person is recording while he is asleep we probably have around 7 hours of video material where nothing happens. Here those 7 hours will define one event. In general we can say that the shorter an event is the more important it will be. Of course the importance of an event is subjective and really depends on the person segmenting the video. The algorithm has to figure out which parts of the video are important and should get their own events.

I have now explained what an event is so now let us take a short look on how the human brain works. This is important because to figure out which parts of a video are important we need to know how a person would remember this situation in the first place. The human brain tends to remember following information:

1. Time
2. Place
3. Persons
4. Events
5. Emotions

Those are the 5 most prominent indicators to remember something. Events conclude special cases like a vacation trip or a party and include further information like objects and tasks.

Of course the combination of those 5 different information also plays a major role when it comes to human memory. It is more likely that a person remembers a person if he knows in which context he met this person in the first place. For example I want remember a person that I met one week ago at the university. If I want to remember that person I subconsciously link this person with the time (i.e. one week ago) and the place (i.e. university) and given that information I can narrow down the actual person.

In this thesis I will only concentrate on time, location and faces. Tasks and Objects are not considered at this point since it would be too much to consider in the scope of this thesis.

To provide the necessary information to connect different aspects of the events like faces, places and time, I created an XML schema too save all the information that are necessary for each event which will be explained in the following section.

Before we get into the actual segmentation I have to say that all design decisions and rules that made and created are solely based on my personal intuition.

The general idea behind the segmentation is shown in the figure below:

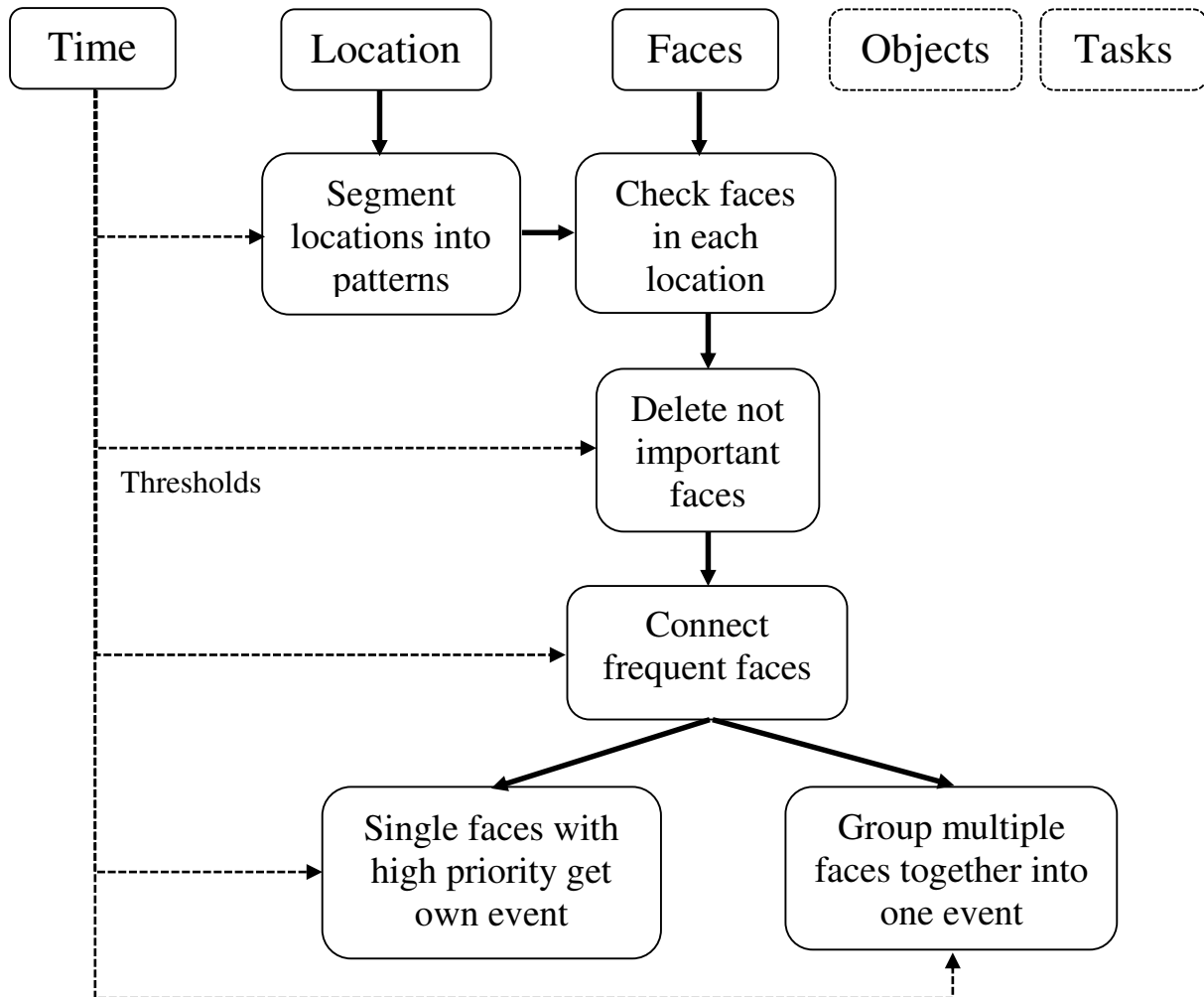


Figure 4. General Segmentation functionality

The first thing that is being indexed are the locations. The goal is to divide the locations into pattern of moving and static locations. The main thought behind this is that for every static location I need to move to get to another static location. Therefore the locations can be segmented into those parts. After that we look at all the faces that appear in every location and further segment the video. The location are hard cuts which means that even if I am walking with a person into another place like a cinema There will still be a hard cut once we are not moving any more. In terms of faces information that does not necessarily make sense since the face that was walking with me was there the whole time. The problem is that we have to define some hard cuts somewhere otherwise we would have events that are just too long in the first place. I also think that it would probably make more sense to start a new event once you reach a certain location. Because once you reach the cinema a new event which is only based around the cinema is created rather than the whole trip to

the cinema. This seems like the most logical solution to make hard cuts and applies to most scenarios. That is why I have decided to do it this way. Once the locations are indexed we can take a look at the faces that appear in each location. Here we have multiple scenarios:

1. No faces
2. One face
3. Multiple faces

For no faces we do not have to do anything. The person recording is probably walking alone which means that there are no people in the focus. Of course it is possible that the person recording is just looking at some beautiful scenery. But since this mainly falls into the location category we do not have to take this into consideration since it was already done by the location indexing.

For one face the segmentation is also pretty easy. Since only one person is in the focus we can look at the conversations that are hold with this person. An important conversation in this case would be a long segment where only this one person is on screen. Then this conversation will get its own event. Of course that can happen multiple times when I walk around with someone important. But that way we can also make sure that not the whole evening is one event but instead have smaller events with this important person which is probably better for remembering specific conversations rather than having one big event where the user has so skip through to find what he wants.

When we have multiple faces the situation changes. We can still look for important conversations with one person by using the rules from before. But now the whole group event is probably more important than smaller ones during that big one. A good example is a party situation. Here we would have multiple different people appearing all the time. So instead of creating one event for each person it will be better to group them together into one big event instead. Since we are talking about multiple years of video material and I want to look back at events that happened one year ago I probably do not want to know what happened during this one minute on that specific day. I would like to see the situation as a whole. When we look at the party scenario I most likely want to see when this party was. Therefore it is better to have one event for the whole party instead of 20 small ones.

The last part of the segmentation are the thresholds which controls the algorithm in the first place. This is also where the time component comes into place. We have to define the rules mainly based on time. Here is a list of all the thresholds that are currently used. The more the algorithm gets developed the more thresholds need to be defined so this is basically an open end list that gets more and more detail the further the algorithm is implemented:

1. Minimum event length: It is useful to define a threshold for the minimum length that an event should have. This way we can make sure that not every little detail will be considered an event. Otherwise we would have such a huge amount that it will not be possible to navigate through the huge amount of data we have. Of course there are special cases in which the minimum event length rule can be

broken, for example when a dialogue occurs. Even a 30 second dialogue can be important. A good value for the minimum event length is very dependent of the situation and the video length. If we have a video of 10 min it will be a different value than for a video with 1 year length. Since we do not have that much video material at this point a value of 2 min should be sufficient.

2. One face screen time: This is the threshold for a dialogue situation. It defines how long a single person needs to be on screen until this person will get its own event. Since even short dialogues can be important I set the value to 30 seconds.
3. Face frequency: Defines the time the person can be lost from the face tracker until it will be considered absent. The idea behind this is that while talking to someone this person does not necessarily always look into the camera. Therefore it is possible that the face is lost from time to time. This means that the face is not detected and therefore not considered to be there at this moment. But the chance is high that the person is actually there but could not be detected by the face detection. To counter this problem the threshold is introduced to define the time a face can be absent before it will be considered lost. If the different appearances of the face happen to occur inside those time intervals the person will be considered on screen all the time.
4. Minimum occurrence time: If a person only appears once and only has a very small screen time which leads to the conclusion that this person was just a passenger that was passing by but still was recognized. Those people are not important and are ignored if their appearance time is below this threshold. A good value is about 5 seconds.
5. Face overlapping time: This threshold is important when we have multiple people on screen, meaning we have a group event. To make sure that a group event is considered a group event and not just a collection of single face events this threshold is introduced to check how long the group event shall be. In a group event we have faces appearing all the time which means we have to check if the time between the face appearances small enough to group them together into the previous event or if a new event should be created. A good value for this would be 30 seconds.
6. Location length: Defines how long the person should stand in one location until this location will be considered an actual location. Since the locations define hard cuts for the event we have to make sure that those cuts have a considerable length. Looking at a simple example can help to understand this situation. Imagine if you would be walking on your way home from work. On your way you have to stop at a traffic light. Now we would have a big moving location divided by many small static locations. Now to prevent the algorithm to create new static locations every time the person stops walking this threshold is introduced to make sure that a location needs to have a specific length. Otherwise we would have countless locations and therefore countless of events. The value for this threshold could be identical to the minimum event length (2 min).

7. Movement speed: One aspect of the segmentation algorithm is to calculate the actual movement speed. It does not affect the event creation that much since we could still have some faces while walking or driving in a car. However it is an information that we can easily access. I have already mentioned above how the movement speed is calculated. Now we need some thresholds to determine which type of movement we have. Are we walking or driving in a car or even flying on a plane? Therefore we need thresholds to set borders which define the maximum walking speed, or driving speed. I set those thresholds to values that seemed reasonable for me. Movement speed will be between 1 and 12 km/h, cycling speed 12 – 30 km/h and everything above is considered driving. Of course those thresholds can be easily affected by things like traffic jams and so on. For those situations the algorithm would expect walking speed besides actually driving with a car. But once I drive slow enough it is also easier to detect some faces so the situation can be considered the same as actually just walking.

4 Implementation

The implementation of the project consists of 2 major parts. The first part is the information extraction part which extracts multiple information from the video files. The second part is the actual segmentation algorithm which divides the original video file into smaller events.

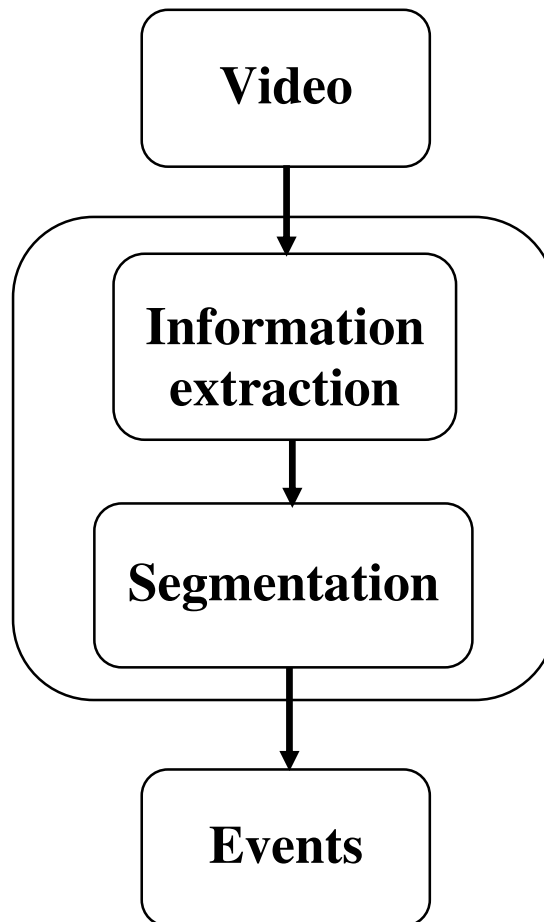


Figure 5. Implementation overview

4.1 Information Extracting

The information extraction will mostly be done by using image processing functions. But first, let us have a closer look at all the information we need to extract from the videos. As I have mentioned before the information we need are:

1. Time
2. Location
3. Faces

The general workflow of how the information extraction works is shown in the figure below:

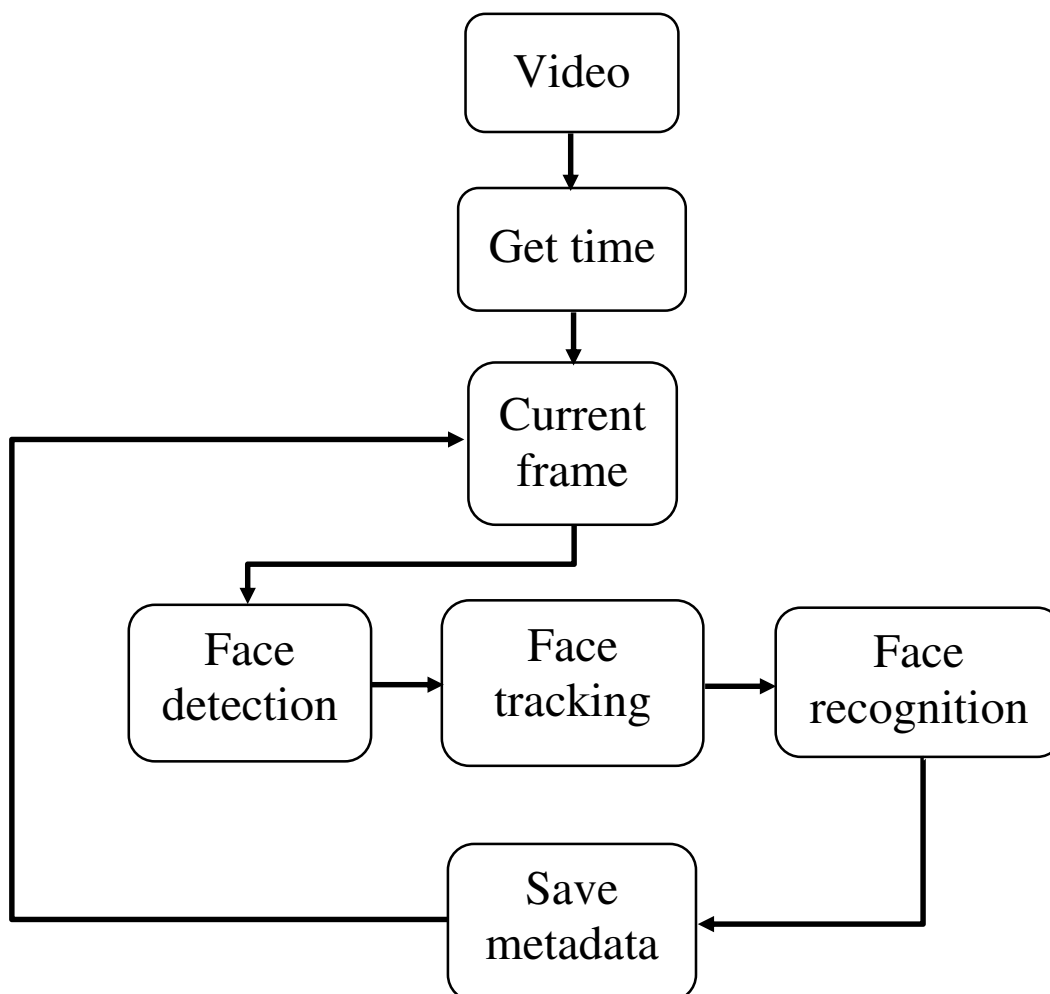


Figure 6. Information extraction

The first and easiest information to get is without a doubt the time. Here we have to look at multiple time values. The first one is the time when the video was recorded. This information is important since the person who has been recording the video only remembers the time when he actually recorded the video. So for that reason I assume that the creation date of the video file is the same as when the recording button of the capturing device was pressed which is also the case for most recording devices. The second value is the current time of the video. This time can easily be calculated if we take the current frame number and multiply it with the framerate. That way we can access the seconds the video is running and

therefor are able to calculate the real time at which the current frame was taken by simply adding this value to the creation date itself.

To access the creation date of the video file I used a java API called Non-blocking I/O (NIO) which provides functions to access metadata of files. Java NIO itself is part of the JDK. For the location and the faces the situation will get a little bit more complicated. The exact information of how to access those information will be discussed in the upcoming subsections.

4.1.1 Face Detection, recognition and tracking

As I mentioned above another important aspect of defining events are the persons that are appearing in it. The easiest way to determine the people in a video is to perform face recognition. But only face recognition is not enough. The first part is to actually locate faces on the current frame. This is called face detection. After that I implemented a face tracker which, as the name already says, is able to keep track of faces. The last part is the actual face recognition. In the following I will explain how I implemented those 3 parts and what library I used.

There exist a lot of image processing libraries that can be used to perform face detection and recognition. One of the best open source libraries is the open Computer Vision library (openCV²). OpenCV is basically a standard library for almost everything that has to do with image processing. It is free to use and is also available in multiple languages like C++, Python, C# or Java. Another advantage of openCV is that it is constantly maintained and improved which means that the algorithms for certain functions will get better over time. The only downside is that there are probably better algorithms out there to perform face recognition and such but most of them are not free and therefore not recommended for an open Source project like this bachelor thesis. Since the algorithm will be written in Java I decided to use the Java wrapper for openCV, namely JavaCV³.

The whole image processing part is shown in figure 7. Each frame of the video will be processed so that we can get the maximum amount of information. To make the face detection more reliable I implemented a skin detector which enlarges the regions of potential skin textures. After that the actual face detection comes into place. Once a face is found it will be tracked. So the next step would be to look if the face is already tracked by one of the trackers. That is done by simply comparing the region of the rectangles around the faces and look for overlapping areas. If there is no tracker yet a new one is created. If a tracker is found than instead of creating a new one the current tracker is updated. At the end all current face trackers are updated by giving them the current frame to calculate the current position of the tracked face.

² <http://opencv.org/> 06.01.2015

³ <https://github.com/bytedeco> 06.01.2015

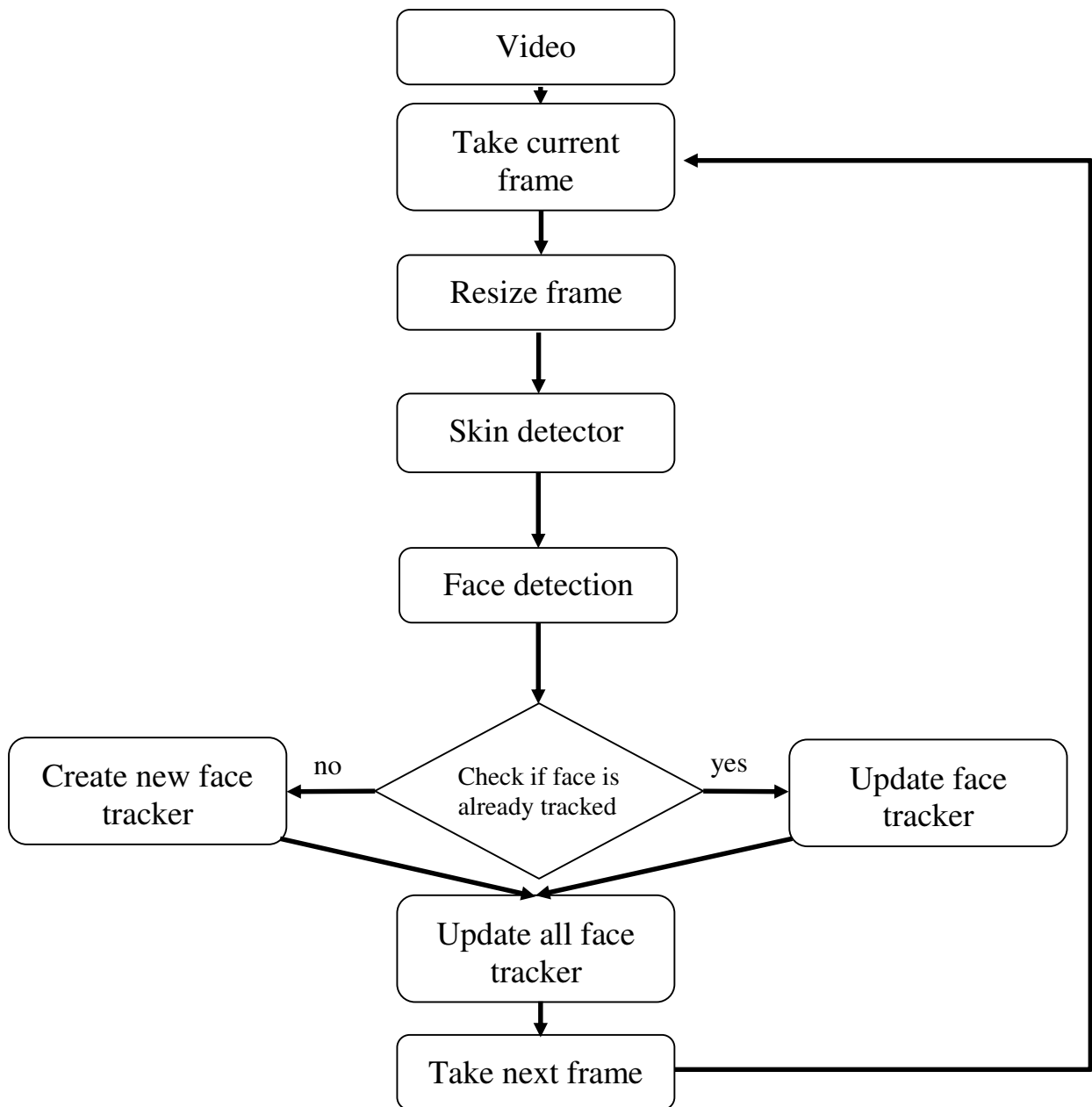


Figure 7. Frame processing

In the following I will briefly describe how the algorithms for face detection, tracking and recognition work and also explain why I used those algorithms and some problems that occurred using them.

For the face detection there aren't that many algorithms to choose from since openCV only provides one. This is the one which was developed by Viola and Jones⁴. I will now shortly explain the theory behind this algorithm. I will not go into the details here since the exact way on how this algorithm works can be read in the corresponding papers.

⁴ http://docs.opencv.org/trunk/doc/py_tutorials/py_objdetect/py_face_detection/py_face_detection.html
06.01.2015

First of all the Viola Jones algorithm is not limited to faces. It can be used to detect all kinds of different patterns and objects. Faces are just a sub category of that. First let us take a look at the advantages that Viola Jones brings with itself:

1. Robust. It has a very high detection rate and normally a very low false positive rate
2. Processing time. The algorithm can work in real time but only if the video file itself has some specific properties. The most important property is the resolution. For simple webcam applications it is possible to achieve a real time face detection.

Here I have to talk a little bit about the videos that we are dealing with. The fast processing time for the face detection is an aspect that normally applies to simple face detection application for webcams and so on. There the resolution is pretty low to begin with. The problem with life logging is that we have a very high resolution. The gopros that were used in this project had a resolution of 1920x1080. For the sake of faster processing time I reduced the frames to a solid 800x600 resolution beforehand. The results were still pretty good despite the low resolution. But still the face detection takes the most time of all the processing operations. Another issue with the videos we had is that there is a huge variations of different faces that can appear. In a normal webcam applications the faces usually always have the same size and look in the direction of the camera. In our videos the head size and orientation can vary quite a bit depending on the situation. Taking all those problems into regard I can say that the face detection does not run in real time but it is still decent enough.

I talked about the advantages and the actual situation we have to deal with. Now I will explain the main steps the algorithm takes to detect some faces. The detection mainly has 4 stages:

1. Haar feature selection
2. Creating integral images
3. Adaboost training algorithm
4. Cascaded Classifiers

The first step depends on the so called Haar features. Those are described by black and white rectangles which can be applied to the image to see if the pattern on the images matches the color values in the rectangles. An example of the rectangles and how they are applied are shown in figure 8.



Figure 8. Haar feature detection (picture of lena included in the openCV library)

The haar features are based on the properties that all human faces share. An example of the haar features are:

- The eyes region is darker than the upper-cheeks
- The nose region is brighter than the eye regions to the left and right

After those patterns are applied the sum of the color in the dark rectangle is calculated and subtract from the sum of the color values in the white rectangle. That way we get a value which indicates if the region can be described by the applied pattern. This is done to multiple image regions. If the values for all of those regions are good enough the region that is currently looked at is most likely a face.

That is the basic way of how the algorithm works. Of course the theory and the reality tends to differ a bit. Therefore I will talk about the problems that I had using face detection. First of all I have to say that of all the image processing I perform the face detection had the best results despite the moody situations that can occur by recording videos. The illumination is one aspect that can heavily affect the face detection since it only works on color values. But for the videos that I had there was not a lot of change in illumination to begin with and even if there was the face detection still worked pretty well. Another issue is the different position and sizes that a face can have. For the size it is pretty easy to filter out those who are too small since they are too far away and probably not that important if they are not near the person who is recording. The rotation and position of the faces tends to be a bigger problem. Normally people do not stare at the person they are talking to all the time. Sometimes the person that is being recorded is talking with another person in front of the camera. Therefore those persons are more likely to look at each other instead of the camera wearer. That can mean that we don't have any frontal faces that can be detected.

To further improve the face detection I implemented a skin detector which functions as a preprocessing mechanism. The face detection itself will not get better by this but at least we can decrease the numbers of false positives by a lot since we only have to search for faces in the regions that the skin detector gave us. The skin detection is also based on the color values, the saturation and the hue value. That means that the detector can find regions where he thinks some faces are because they have the same values which are defined as skin. This would lead to the assumption that we can simply use the skin detection instead of the normal face detection algorithm. The problem is that not every single bit of skin is detected by this detector. A face could fulfill the value requirements to be considered skin but sometimes if the illumination is not good enough the eyes can turn out really dark which then will be not considered as skin even though they are technically skin color. If we would just take those regions and run the face detection afterwards the detection could become even worse. Therefore I also implemented an enlargement algorithm which basically just finds all the pixels which are considered skin and enlarges the region around that. Once we have the enlarged regions a mask is created which is then applied to the original video frame. That way we can make sure that when a face is detected by the skin detector, the whole face region will be available when the face detection runs over the image.

As I mentioned the main objective is to minimize the area for the face detection. Therefore it is not that problematic if some regions which do not contain any skin are detected. If the region is still considered as a possible face then that does not automatically mean that the face detection will find one. The bigger issue is when some faces are not detected as skin. If that happens the skin detector completely locks out this regions even if there was a face. I personally did not see this happen even once in the videos that I had besides some people in the background where the faces were extremely small (around 20x20 pixels). Those faces are not actually important and can be ignored in the first place. Despite that the possibility of an important face to be not considered the right skin color still exists but since I did not have that many problems with it I did not focused too much on this.

That covers the part of the face detection. Now let us take a closer look at the recognition.

For the face recognition, openCV provides 3 possible recognition algorithms. Those are the:

1. Eigenface recognizer
2. Fisherface recognizer
3. Local binary histogram pattern recognizer (LBHP)

Eigenface and LBHP are dependent on the appearance of the faces, meaning they work with mainly with the color values. Therefore I would have expected that the fisherface recognizer would be the best choice. However for the video that I have tested the LBHP recognizer actually had the best results. Additionally the LBHP recognizer is much easier to handle since it has more parameters that can be changed. The big advantages of the LBHP recognizer are that it does not require grayscale images unlike the other two and the training images do not have to be the same size either. Also whenever a new face is saved

in the database you do not have to train all the images again, instead it is possible to simply update the training model and add the new image. That alone makes it easier to use the LBHP one over the other two. In the following I will explain the rough functionality of the LBHP algorithm. A more detailed explanation is provided in this paper⁵. The algorithm looks at the neighborhood of each pixel on the face image and calculates an eigenvalue at this position.

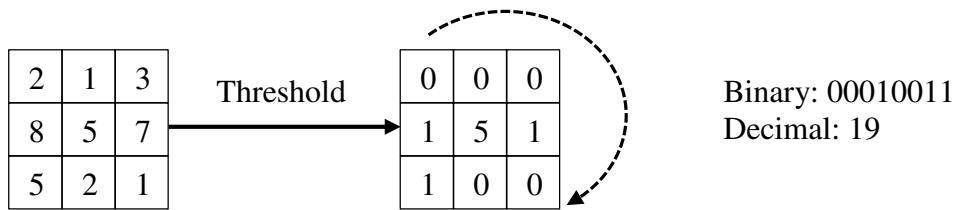


Figure 9. Local binary pattern

The algorithm looks at all surrounding color values in the neighborhood and assigns new values i' which are defined as:

$$i'(x) = \begin{cases} 1 & \text{if } x \geq i(x) \\ 0 & \text{otherwise} \end{cases}$$

After that an 8 bit number f is created by accessing all those values in a clockwise manner as shown above. That way we get a new color value at every pixel position in the image, resulting in a new image. The recognition then is performed by dividing the image into a grid and calculating the histogram for each cell.

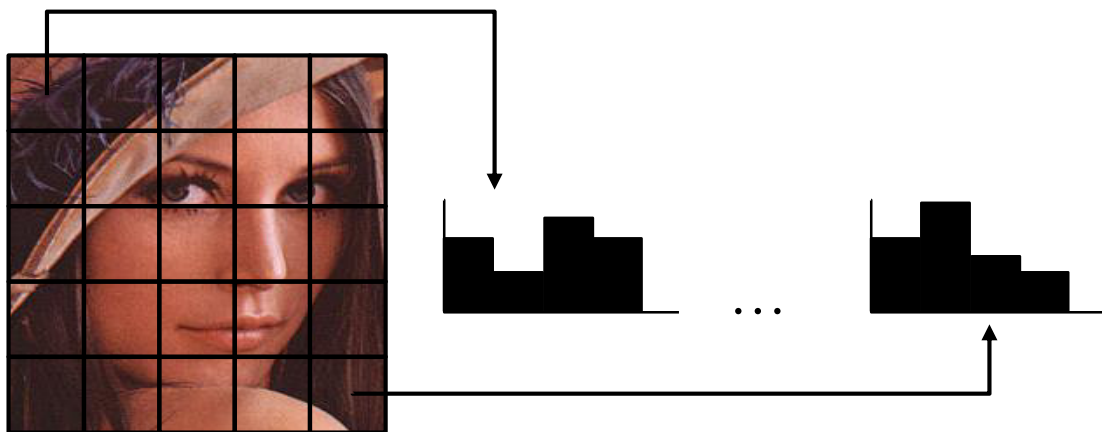


Figure 10. Local binary histograms

⁵ http://www.researchgate.net/profile/Chi-Ho_Chan/publication/225116203_Multi-scale_Local_Binary_Pattern_Histograms_for_Face_Recognition/links/00b495225c98186d23000000.pdf
06.01.2015

Those histograms are then used to calculate the feature vector which can be compared to other images.

It becomes clear that the algorithm performs better if only the actual face region is saved. This basically applies to the other two algorithms as well. The more unnecessary background is in the image the more values we get which most likely do not match with the pictures in the database. This is a huge problem since the region containing the face given by the face detection can vary quite a bit. Despite that issue I decided to just take the region that was given by the face detection.

Another problem is the actual recognition success rate. The success rate decreases the more people are already saved in the database. This is pretty much self explanatory since it becomes more likely for a new face to be recognized as an already existing face. That is basically the fault of the threshold which defines how big the difference has to be until a face will be considered as an unknown face. If this threshold is too high then multiple people will be recognized as the same person, if it is too low the chances are that the same person is recognized as an unknown person and therefore gets assigned a new id. The easiest way to go is to take the standard value provided by openCV which, for the LBHP recognizer, is 120. The succession rate can be improved if there are more pictures of one person in the database with different angles, illuminations and such. Therefore I set the maximum number of images per person to 50. The downside is that the recognition will take longer since the algorithm has to look at more pictures but processing time is not as important as correct results.

One issue that I have to mention is that we are trying to detect important persons in a video. But since the camera is running most of the time the chances are high that non important people will also be captured on screen. That being sad we need some kind of method to distinguish faces that are important and those that are not important. A simple face tracking algorithm helps us out in this regard. With this algorithm it is possible to keep track of detected faces and only perform face recognition after a specific number of frames. That way we exclude all faces that appear only for a small amount of frames. But the face tracker also brings some problems with it as shown in the table below.

Table 2. Pros and cons of face tracker

Standard face detection and recognition	Own implementation with face tracking algorithm
+ better performance + less false positive (in absolute numbers) + can operate without regarding of past memories	+ stricter face detection + able to keep track of persons + non important people are not recognized
- cannot keep track of people (no memory) - recognizes people that are just passing by who generally don't need to be recognized	- higher false positive number (false positives are also tracked) - more room for potential errors

In order to keep track of a face on screen we need some kind of motion estimation in pictures. OpenCV helps us out here once again by providing algorithms to detect feature points and to track them by using optical flow calculation. The face tracking algorithm was inspired by the `pi_face_tracker`⁶. This algorithm also uses openCV to track a face. The project is open source which means we can use it and change it. Unfortunately the code is written in python. That means I had to translate it to Java and perform some additional changes in order to get it to work.

The feature points in the face region are determined by using a SIFT feature detector. Once those feature points are found the optical flow of those points is calculated. The functionality of the algorithm is shown in figure 11.

First we need to detect a face as explained in the face detection section. Once we detect a face we find some feature points on it. To achieve this we use the scale invariant feature transform detector (SIFT) provided by openCV. After calculating the feature points on the face we take the next frame from the video. Now the motion of the feature points from the previous frame to the current one is calculated by the openCV implementation of the Lucas Kanade algorithm⁷.

This is how the basic face tracking works. But in order to get better results we need a lot of utility functions. One for example is to enlarge the region in which the algorithm searches for feature points. The amount of feature points is checked multiple times during one iteration of the image processing part. The first time we can enlarge the image region and look for feature points again. If there are still not enough feature points after that the tracker can

⁶ http://wiki.ros.org/pi_face_tracker 06.01.2015

⁷ http://docs.opencv.org/master/doc/py_tutorials/py_video/py_lucas_kanade/py_lucas_kanade.html
06.01.2015

be deleted. At this point I have to mention that if the tracker is just calculating the current position of the feature points the points will be lost after some time because the optical flow calculation automatically disregards all points which have a too high error value. That means that the feature points also need to be updated from time to time. This can be done by using the face detection. Once a face is detected which is already tracked we automatically have the perfect image region of the face which then can be used again to reset all the feature points. That way we can make sure that the feature points stay in the face region and do not get lost over the time. Of course if the face detection does not detect the face again after a specific amount of time the tracker potentially loses all the feature points and it can be deleted. The most important part of the tracker is the recognition. With the tracker we can perform the recognition every few frames instead of every single one which saves some processing time. One important issue that has to be considered here is that the recognition needs the actual face region to successfully detect some faces. The tracker itself also provides a region around the feature points which marks the tracked region but this one does not necessarily contain the whole face region. In fact it almost never contains this region except when the tracker was just updated. Therefore a recognition flag is introduced which is set to true after every few frames in order to initiate the recognition. The next time we gain access to the actual image region is when the face detection updates the tracker. If the recognition flag is set to true at this point the recognition is also performed. That way we can make sure that for every recognition we have a perfect face region to compare.

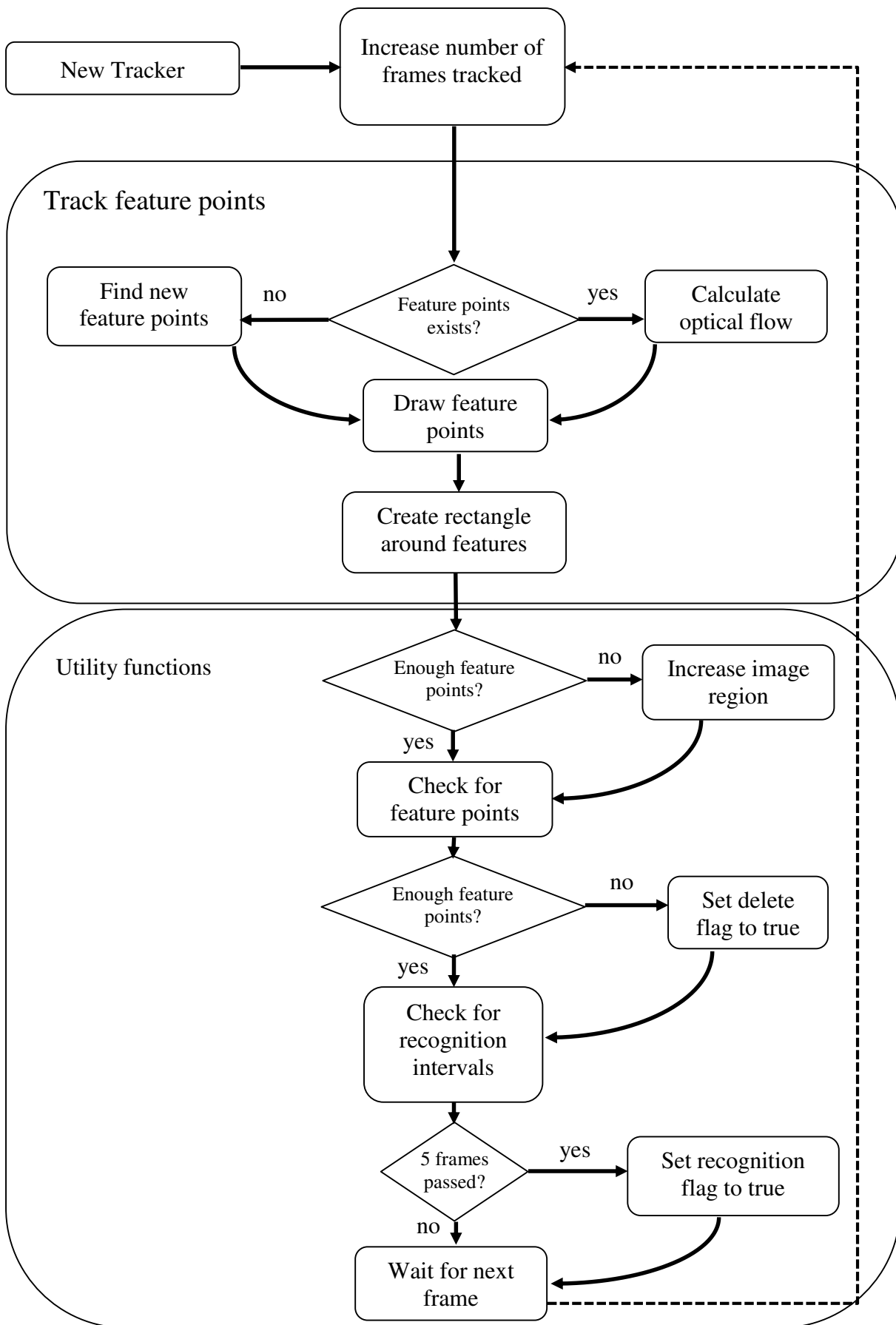


Figure 11. Tracking algorithm overview

Another issue that I have to address is how the recognition can be improved. I have mentioned before that the recognition can be improved by saving more images of a person in the database. But to make sure that all those faces do not look the same cause they were taken every single frame, for a video with 30 frames per second that would mean like 2 seconds of screen time, I created a threshold which defines of how many frames the faces has to be tracked before a new picture can be saved. That way we can make sure that not every picture of the person looks the same and we therefore get more variety in the angles, illumination and position of the faces. The downside of this is, that if the difference between the first few images is too big, then the chance that the person that was tracked is recognized as a different or a new person is increased which ends up with a database were multiple people get different ids. To make sure that this does not happen the threshold when a new picture can be taken has to be small enough that the person is still recognized as the right one but also big enough that we get actual changes of the face in the picture. In conclusion it is probably better to set the threshold too small than too high, because we can still achieve a bigger variety of faces by simply increasing the amount of images a person can have. In this context I set the threshold to a 1/6 of the framerate. For a framerate of 30 frames per second that would mean that every 5 frames a picture of the tracked face can be saved in the database. Of course this also gets problematic when we have a scenario where a person is just sitting still a directly looking at the camera. Then we would get the same picture over and over again. Therefore I also implemented a simple comparison which takes the value of likelihood from the recognition and only saves new images if the value exceeds a specific threshold. The threshold for defining one person is currently 120. While the value is below 60 no new images are taken. Besides that there is still much work to do in order to achieve perfect perfect results in face recognition but that is beyond the scope of this thesis.

4.1.2 GPS data extraction

Till now we discussed the part of time and face extraction of the videos. The last aspect that is left to extract is the Location. To define the Location of a certain object GPS is used. The question is only how to get the GPS data in the first place. Normally this information can be extracted from the recording device. Sometimes it is even saved in the files that were recorded. Unfortunately the devices we used in the experiment did not have a GPS sensor. This was a problem I solved by creating the data manually. Since the segmentation algorithm is only a concept for the future it does not really matter much if the data is fake or not. The data just need to be realistic. In the future we will probably have better technology to begin with. Furthermore we can assume that the GPS data is available at all times. We do not have to take scenarios into account where the GPS data suddenly gets lost for example if a video is recorded inside a building where no GPS is available for some reason.

Unfortunately for the manual creation of GPS data there is no better way than watching the video and see when and how fast the person recording moves and manually entering the Location for every time interval. For easier GPS data creation I created a simple user

interface where it is possible to enter the GPS location for a specific second in a video. For easier data insertion I only created GPS data in intervals of 5 seconds.

4.2 Storage of indexing information

To store the information we have to look at the different types of information we want to save. The first type is the result of the information extraction algorithm. What we need to save here are the information of the faces and the GPS locations we added manually. For every video two csv files are created to store those information, one for the faces and one for the GPS data.

The csv file for the faces contains 4 columns with following information:

1. Id: The id of the face.
2. Length: The number of frames the face is present.
3. Frame: The number of the frame when the face appears.
4. Occurrence: The time of the face appearance in real time in milliseconds.

The csv file for the locations contains 4 columns with following information:

1. Longitude: The longitude of the GPS data.
2. Latitude: The latitude of the GPS data.
3. Frame: The number of the frame for which the GPS data was saved.
4. Occurrence: The time for the GPS data in the video in milliseconds.

The second type of information we have to save are the results of the Segmentation algorithm. To store the information we decided to use the xml format since this is probably the easiest one to create and also very practical for saving and loading information.

The exact xml schema is shown below. It serves as a structure to make sure every event contains the same type of information. Each event has its own unique id which serves as an identifier. Timestamps which contains the frame number and real time are used to mark the start and end of the event. For the location a set of 2 GPS data are used to identify the place. If the two GPS data are different ones that means we have a moving location. A list of face instances is used to list all the faces of all people who appear in this event. To identify the faces the id and length of the appearance time is saved. The faces also have a priority ranking which is solely based on appearance time at this point. The idea behind this is that faces that appear often are probably more important than other ones. This can and will probably change in the future when there is more information for the different persons available. There is also a placeholder for the name of the person which currently just holds the id but can serve as a better identification in the future. A placeholder for future tasks and objects also exists but are not used at the moment. The remaining elements contain general information about the events:

1. An id to identify the event
2. The path to the video file containing the event
3. The movement type (driving, cycling, walking, none)
4. The identification of the event (party, walk, work, etc...)

```

<xs:schema targetNamespace="LifeLoggingVideoIndexing/events" element
  FormDefault="qualified"
  xmlns:xs="http://www.w3.org/2001/XMLSchema"
  xmlns:tns="LifeLoggingVideoIn
    dexing/events">

  <xs:element name="Event" type="tns:event"></xs:element>

  <xs:complexType name="event">
    <xs:sequence>
      <xs:element name="start" type="tns:TimeStamp"></xs:element>
      <xs:element name="end" type="tns:TimeStamp"></xs:element>
      <xs:element name="faces" type="tns:Face"
        maxOccurs="unbounded"
        minOccurs="0"></xs:element>
      <xs:element name="locations" type="tns:Location" max
        Occurs="unbounded" minOccurs="1"></xs:element>
      <xs:element name="objects" type="tns:Object"
        maxOccurs="unbounded" minOccurs="0"></xs:element>
      <xs:element name="tasks" type="tns:Task"
        maxOccurs="unbounded"
        minOccurs="0"></xs:element>
    </xs:sequence>
    <xs:attribute name="id" type="xs:int" use="required">
    </xs:attribute>
    <xs:attribute name="type" type="xs:string" use="optional">
    </xs:attribute>
    <xs:attribute name="movement" type="tns:Movement"
      use="required"></xs:attribute>
    <xs:attribute name="video" type="xs:string" use="required">
    </xs:attribute>
  </xs:complexType>

  <xs:complexType name="TimeStamp">
    <xs:sequence>
      <xs:element name="frameNumber" type="xs:Long"></xs:element>
      <xs:element name="realTime" type="xs:dateTime">
      </xs:element>
    </xs:sequence>
  </xs:complexType>

  <xs:complexType name="Face">
    <xs:sequence>
      <xs:element name="occurrence" type="tns:TimeStamp"
        maxOccurs="unbounded" minOccurs="1"></xs:element>
    </xs:sequence>
    <xs:attribute name="id" type="xs:int" use="required">
    </xs:attribute>
    <xs:attribute name="name" type="xs:string" use="optional"
      default="placeholder"></xs:attribute>
    <xs:attribute name="priority" type="xs:int" use="required">
    </xs:attribute>
  </xs:complexType>

  <xs:complexType name="Location">
    <xs:sequence>
      <xs:element name="occurrence" type="tns:TimeStamp">
      </xs:element>
    </xs:sequence>
  </xs:complexType>

```

```

    <xs:attribute name="id" type="xs:int" use="required">
    </xs:attribute>
    <xs:attribute name="priority" type="xs:int" use="required">
    </xs:attribute>
    <xs:attribute name="Longitude" type="xs:double"
        use="required"></xs:attribute>
    <xs:attribute name="Latitude" type="xs:double"
        use="required"></xs:attribute>
</xs:complexType>

<xs:complexType name="Object"></xs:complexType>

<xs:complexType name="Task"></xs:complexType>

<xs:simpleType name="Movement">
    <xs:restriction base="xs:string">
        <xs:enumeration value="Walking"></xs:enumeration>
        <xs:enumeration value="Cycling"></xs:enumeration>
        <xs:enumeration value="Driving"></xs:enumeration>
        <xs:enumeration value="None"></xs:enumeration>
    </xs:restriction>
</xs:simpleType>

</xs:schema>

```

4.3 Implementation of segmentation

First of all I have to mention the technical side of the segmentation. For the indexing I used pure java code without any major external libraries. I only used some third party libraries as little helper functions like xstream⁸, to create xml files and opencsv⁹ to read and write csv files. This could have been done without external algorithms but they make the work extremely easy and also save a lot of code. In the sections above I have explained the general idea behind the segmentation. In the following I will show in detail how every single step is done.

First of all the whole location information has to be read from the csv file. After that the pattern of moving and static is calculated. To do this we have to iterate over all the GPS data. The main idea is to find static locations. Everything in between them have to be moving locations. To get the static location we have to look at two consecutive locations. If they are the same that means we have a static location. Now we have to create a tolerance space around the location to make sure that not every step the recording person does will be considered a location change. Therefore a circle of 10 meter radius is created around the static location. Those 10 meters are based on personal intuition and can vary quite a bit according to the situation. In the normal context of having a static location for like staying at home or at the office 10 meter seems to be a good value. While the recording person moves around in this tolerance circle the GPS change will not be considered as a location change. Once the person moves out of that circle the algorithm considers this as a moving location until two following GPS data are the same and the procedure repeats again. Every location has 2 GPS data. 1 for the beginning and one for the ending. If those two data are the same the location is static otherwise it is a moving location. Additionally we can calculate the movement of the moving location. To do this we just take the distance between the start and end locations and divide by the time it took to get from the start to the endpoint. That way we can also define the movement type. The specific type is calculated by looking in which movement speed category the current speed belongs to. Walking speed can be from 1 up to 12 km/h. between 13 and 30 km/h it will be considered cycling and everything above is considered driving. After we have all the locations the algorithm iterates over all the locations again and checks if the location is of a considerable length. To determine the necessary length for a location the “location length” threshold is used. The functionality of the location segmentation algorithm is shown in figure 12.

⁸ <http://xstream.codehaus.org/> 06.01.2015

⁹ <http://opencsv.sourceforge.net/> 06.01.2015

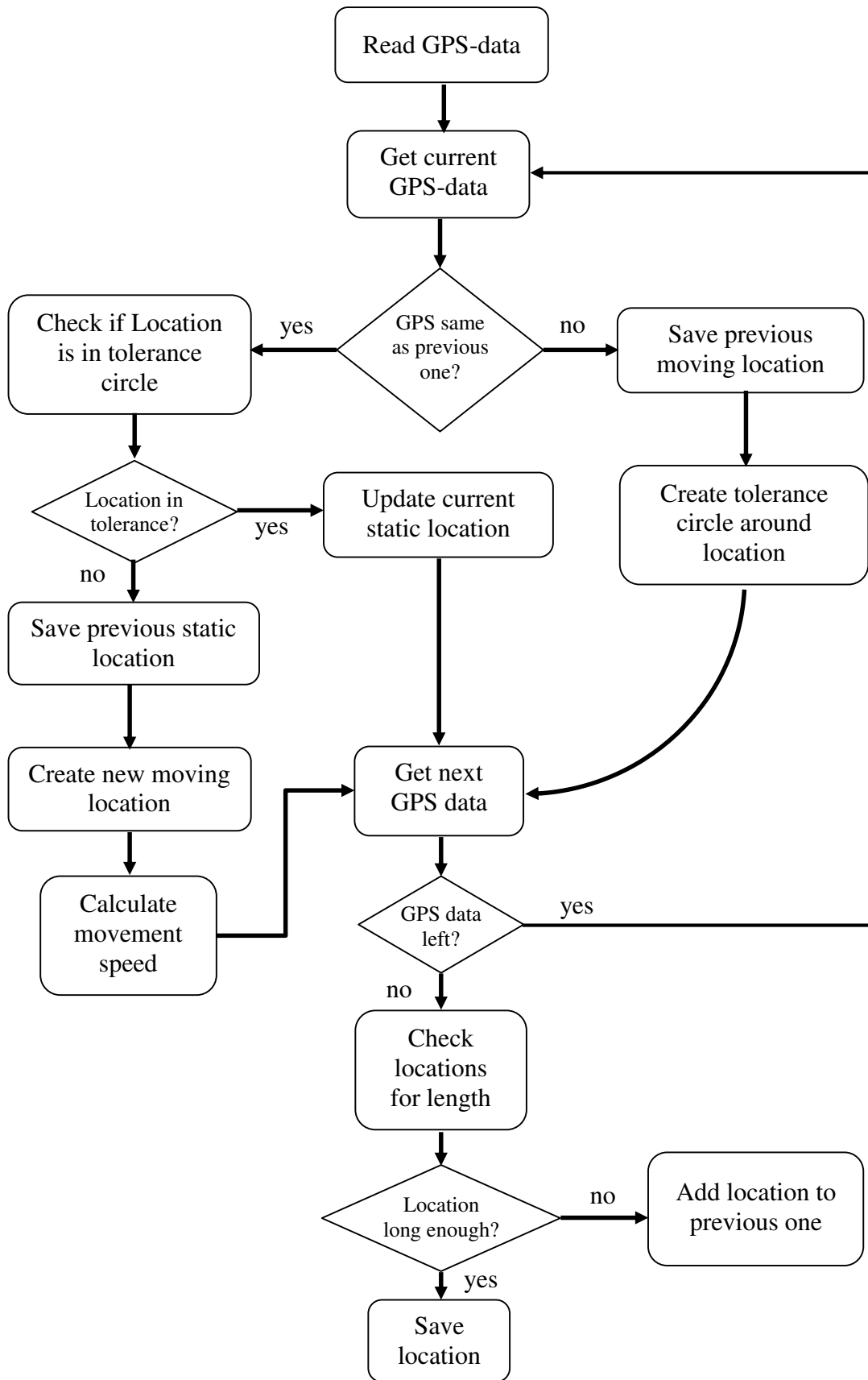


Figure 12. Location indexing overview

The method to segment faces is based on the same idea as for the locations. First the algorithm iterates through all the faces and looks how many times a face appeared. After that the appearance time is taken into consideration. If the face only appeared once and the overall appearance time is below the “minimum occurrence” threshold, the face will be not included in the segmentation since it is not important. The second step connects all the faces together. If one face appears multiple times and the time between those appearances is below the “face frequency” threshold those face instances are grouped together into one instance that combines the appearance time of all the face appearances. That way we can not only reduce the number of faces and increase the performance of the algorithm it also makes the overall segmentation a lot easier. After that step the algorithm iterates over all faces again and checks for the current situation of each face. Is it a single face, or are there multiple faces? This can be determined by looking at intersections of the face appearances. If the faces intersect with each other than we have a group event and if not a single person is on screen.

In the first case the “one face screen time” thresholds comes into place to give the face its own event if it is on screen long enough. If that is the case then every face up to the current one will be grouped in the current event. After that a new event for the single face is created and the next face will be focused.

For the second case the “face overlapping time” threshold comes into place. Here multiple faces are grouped together into one big event. To make sure they are grouped together we look at the intersection of the face appearances again. If the following appearances overlap in a certain degree defined by the threshold the faces are taken into the current event. If the gap between two following faces is too big then another scenario comes into place. If this gap is bigger than the “minimum event length” then the current event up to this point is save and a new event is created for this part without any face appearances. If the gap is too big to be considered a group event but still small enough to get its own event the time where nothing happens is simply added to the current event and a new event starts for the next face appearance. The functionality of this algorithm is also shown in figure 13.

After the events are created an xml parser creates an xml file for each event. Furthermore a video cutting algorithm, that is based on openCV, cuts the corresponding video into the parts which are defined by the events. The result is a collections of all the events in xml and video form. The reason behind this is that the visualization project which tries to create the 3D browsing tool for the video files needs the videos for each event. More information about this topic is written in the future work section.

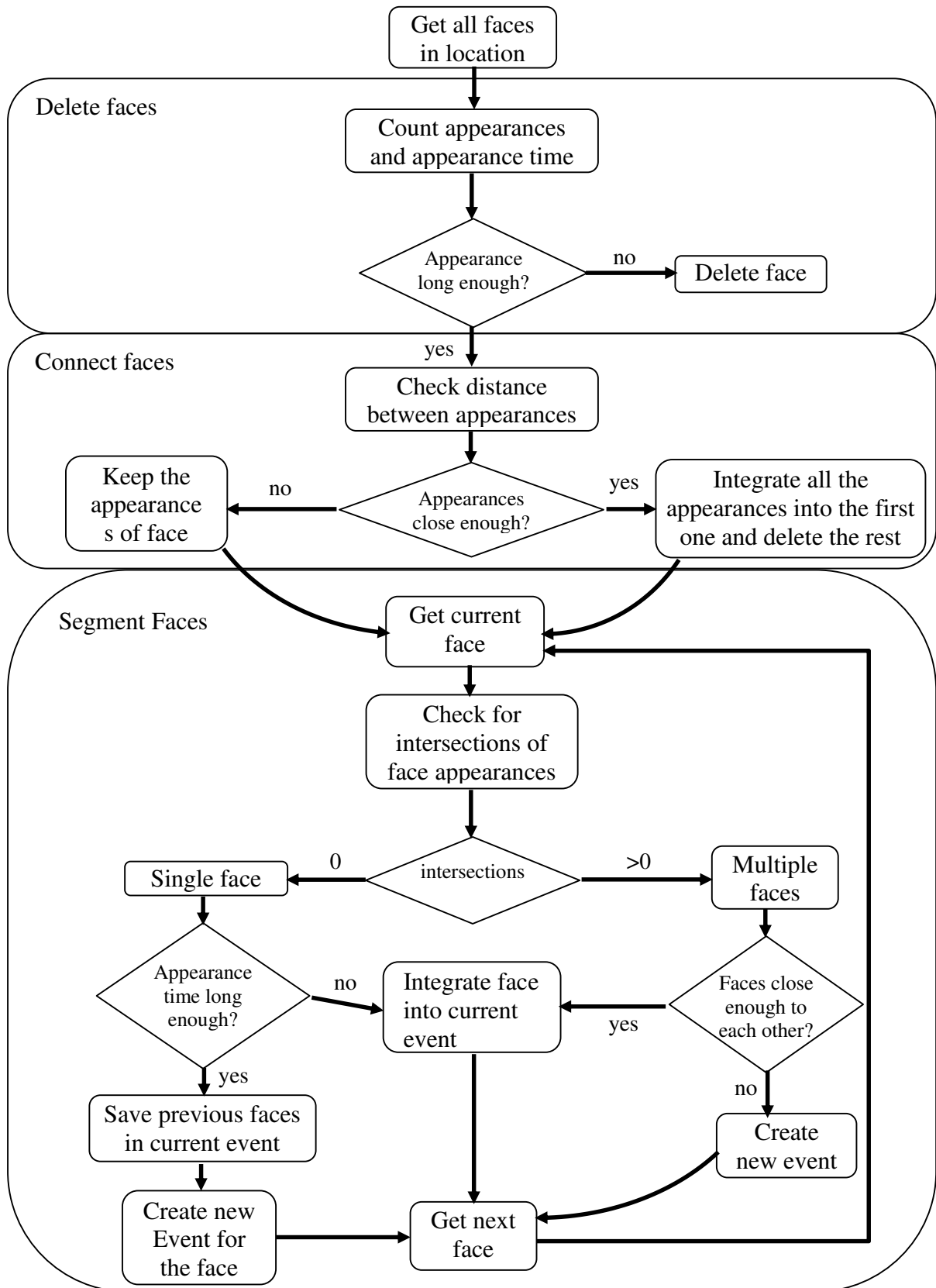


Figure 13. Face indexing overview

5 User Study

5.1 Aim

The aim of this user study is to figure out the importance of different faces in a video. Since we need to add the meta-information about the different people that appear in the video we need to identify which people have some relevance. Therefore we want the participants to select and rank faces that seemed important. In the end we want some indicators which show when a face is considered important. Then we can compare those results with the functionality of the indexing algorithm.

The things we want to check are following:

1. Which faces are selected
2. Why was the person selected?
 - a. Active reasons (interactions with the person)
 - b. Passive reasons (screen time, size, etc...)
3. How important was the person ranked?

The goal is to find out if the reason for selecting faces of the participants are similar to the ones we used to rank the importance between faces. Also if the results strafe too far away from our algorithm we will have to see what we can do to fix those issues. At this point it is important to say that the algorithm can never be as good as labeling done manually by people who actually watched the video. Every human has different tastes and preferences so it is hard to create an algorithm which satisfies all different preferences. Then there is also the issue of what the algorithm is capable of doing. If the participants all rank the importance based on interaction like talking and hand gestures it will also be difficult to adjust the algorithm since facial expression and tasks are not to be considered at this point. If it is possible to integrate the results into the algorithm than it can be changed, but that of course is only the case if the participants follow a common trend in ranking the faces. If every one of them uses different rankings than it will be impossible to integrate them as well. In this case we will have to find a way that will get as close as possible to every ones preferences.

5.2 Expected outcome

Since we didn't tell the participants the restrictions that the algorithm currently has they have more possibilities on how and why they want to rank the faces. There is the problem of having every person acting in a different way but I personally don't think that will be the case. The human brain tends to remember faces better if they leave a strong impression on them. In that case it will be more likely that the active interactions of the people in the videos are probably more important than the passive ones. Active interactions would be the change of facial expressions, talking or making hand gestures. Passive are parameters like size of the head and screen time. I believe that the majority will rank the people by those active interactions, meaning that people that are currently talking will be ranked higher than the people listening. Of course the lead of the active person can change during the video, so the choice of marking the same person with multiple rankings based on the current interaction is still an issue that has to be considered. Do the participants treat the conversation

as a whole or would they separate it into smaller parts, for example, when the person who is currently talking changes? I believe the participants will select the people only once since the videos are very short to begin with and also the persons in the video do not drastically change their engagement in the video. When I look up a meeting that I had years ago I would probably label the persons in that event on their interactions during the whole meeting instead of each single interaction. It depends on how much the participants put themselves into the situation of looking up the video years later. So to summarize the expected outcome I assume following selection reasons and importance ranking:

1. Persons who are interacting with the person recording or with other people in front of the camera are more likely selected than people who are just sitting in front of the camera and do nothing. However the specific parts of interacting that are a reason for considering it an interacting in the first place can change a bit from person to person.
2. People who are interacting will also have a higher importance ranking than passive people

5.3 Method

As I stated previously we want to find out how the participants rank the importance of faces in a video and also why they would be ranked that way. The goal is to find clear indicators which describe how important a person is in a video. To get good results we mainly have to look at the two different rating aspects. The first aspect is why people are selected and the second one is how important that person will be based on those reasons. By analyzing the results which will hopefully be almost the same for each participants it will be possible to compare those results with the way the algorithm works now. As a little reminder: The algorithm currently looks for frontal faces and ranks those people based on screen time. If the participants use different means to rank the importance of a face we can look at those results and try to change the way the algorithm works so it can give similar results to those that we got in the user study.

5.4 Participants

For the user study we had 16 participants with an age range between 23 and 76. The average age was 42 and the standard deviation 19.6. Overall we had 9 male and 7 female participants.

5.5 Apparatus

For the video that the participants are going to see we prepared 4 scenarios that we recorded beforehand. Each of those videos is about 1.5 min long and the final video will be a combination of all 4 of them. To achieve some better result we divided the participants into 4 groups. Each group will get to see a different combination of those videos. That way we can make sure if the memory of the previous scenarios somehow affect the decisions the participants make. The final video itself is about 6 min long and was recorded with the gopro mentioned in the introduction. The video has a normal framerate of 30 frames per second. Also to help the participants to concentrate on only the visual part I cut out the audio from the videos. For recording the 4 scenarios we had a total of 4 people were one of them was wearing the recording device.



Figure 14. Scenario 1

The first scenario is a simple dialogue in one static location between one person and the person wearing the camera.

Here we only have one person on the screen so it will be pretty clear which person the participants are going to select and which ranking the person gets. That way we can fully concentrate on the reason why this person was ranked as important and figure out how the algorithm should behave if there is only one person on the screen.

The second scenario was a walk through a building.

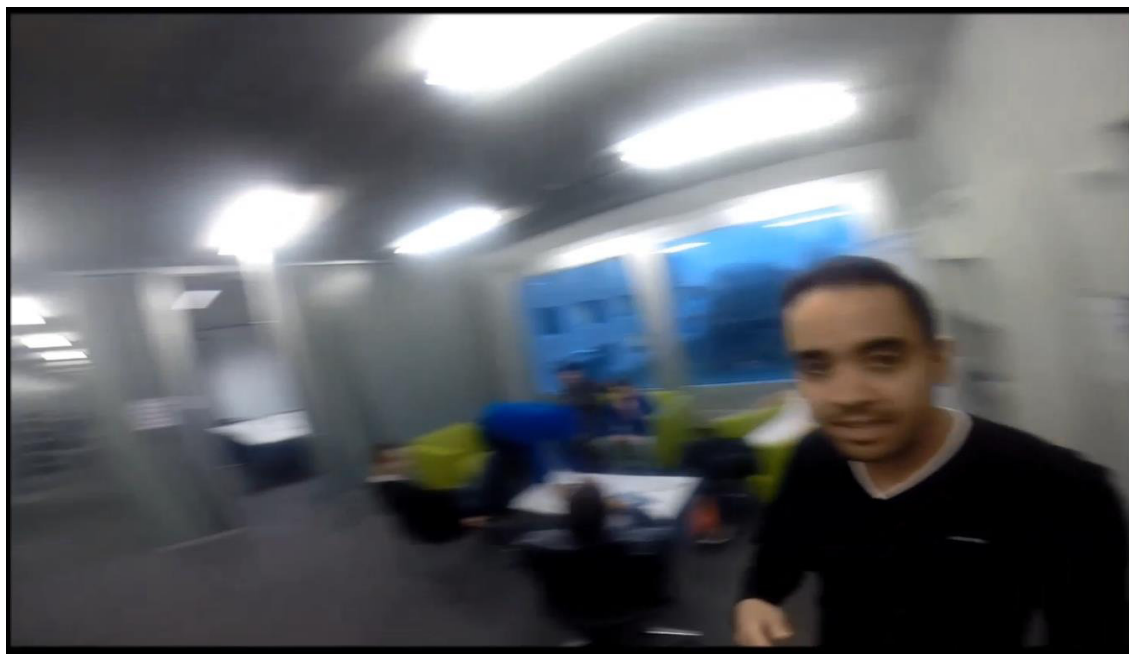


Figure 15. Scenario 2

Here the camera was moving almost all the time. This scenario will give us a better insight of how movement affects the face selection. There were also some people in the background but those were only seen for a very short amount of time. So it will be interesting to see if those persons are actually selected or not.

The third scenario was a group meeting where we had multiple people talking.



Figure 16. Scenario 3

This is basically the same video as the first one only with two additional people. This scenario provides us information of how the participants rank different people on screen.

The last scenario was a meeting scenario in a place with a lot of people in the background. We had 3 people on front of the camera. Two of them were talking (the 2 on the left) and the third person was just eating the whole time and didn't engage in the conversation at all. There also is a person in white on the left who appeared from time to time and additionally there were a few people in the background.



Figure 17. Scenario 4

Here we have a huge variety of different situations. We can see when a person is considered important and also how do participants deal with people that are on screen only for a few seconds and also how to treat multiple people in the background.

For the apparatus I created a little web applications which allows the participants to select and rank faces in a video.

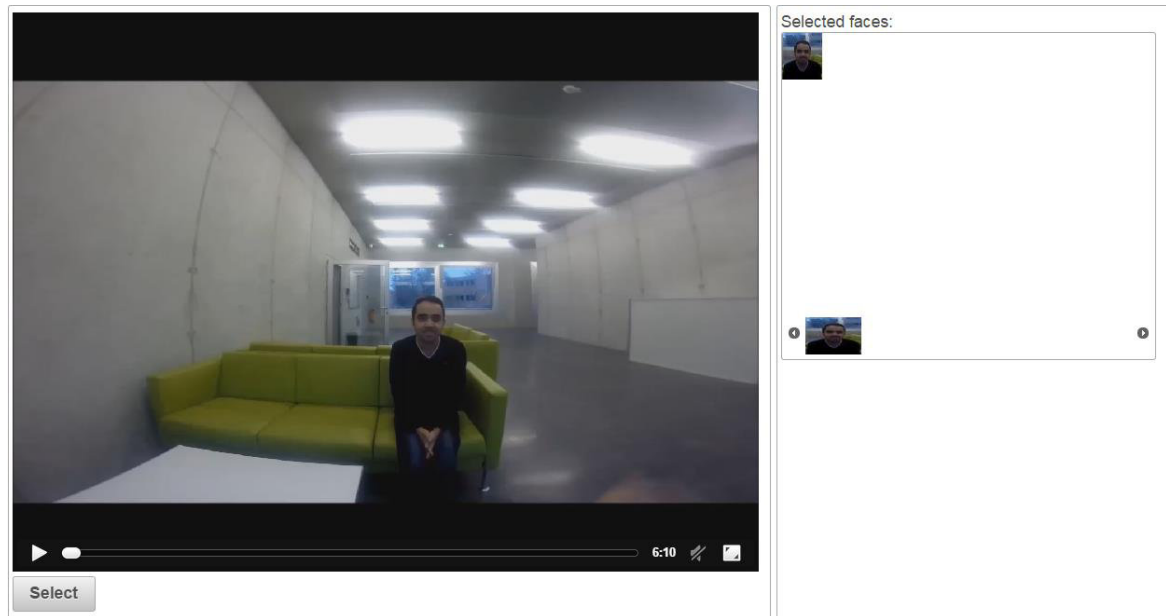


Figure 18. Apparatus overview

The web application provides a simple video player in which the participants can view the given video. On the right there is also a galleria of all the faces that were selected up to now. On the bottom of the video player there is a button which opens following dialog:

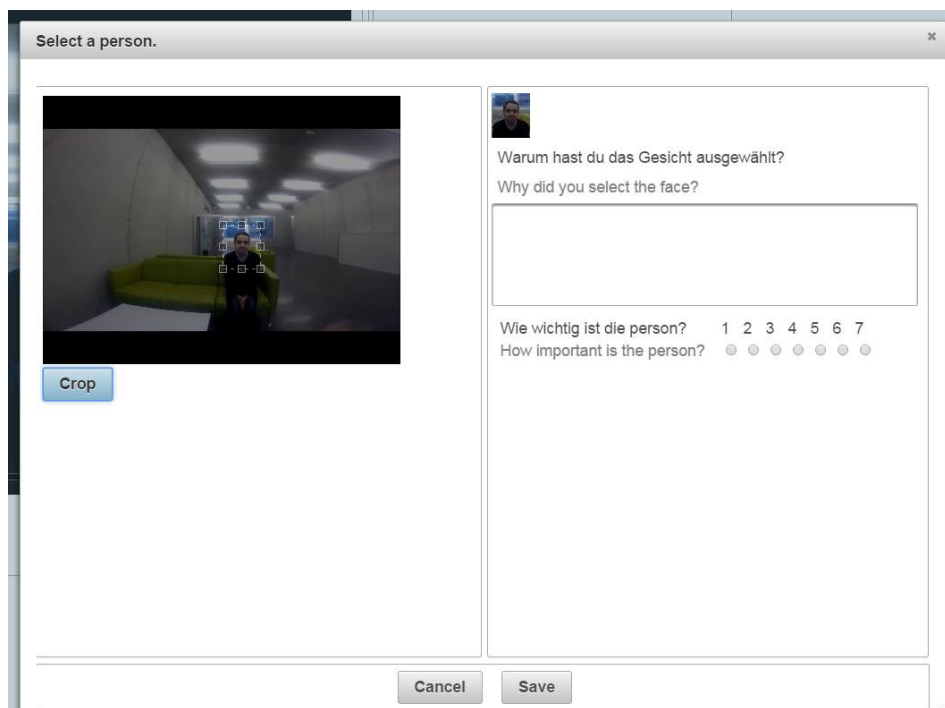


Figure 19. Face selection screen

On the right the user can see the current frame of the video. Here the user can draw a rectangle around the face of the person he wants to select. Below the picture is a button which crops the selected part of the image out and shows it on the top of the left side. Below this picture is a text area in which the user can write down their reason for selecting the face and also a group of radio buttons for the importance ranking.

5.6 Tasks

We asked the participants to perform their given task which is selecting people from a video which seem important. The participants were asked to imagine the fact that they have been life logging their life for several years. After some years they want to watch some of those recorded videos. So now they have to look at those videos and define which persons in the scenes are important and which were not. Therefore they are given the apparatus described above where they can see the video, select the faces, give a reason for the selection and rank them on a scale from 1 to 7.

5.7 Measurements

The things that we can measure are:

1. The selected faces
2. The reasons for selecting the faces
3. Importance ranking of the faces

The selected faces are saved in JPG format. Those faces help to identify the important people in a scene. The reasons are in text format and can contain multiple reasons in order to figure out why some people are more important than other people. Another reason for giving an explanation is to figure out if the selected faces have anything in common. The last measurement is the importance ranking. The importance can be selected on a scale from 1 to 7 where 7 is the most important and 1 being the least.

5.8 Procedure

We invited different volunteers to participate in the user study. After that the user study and the task described above was explained to them. Some constraints for the study were:

1. The video has no audio, so the participants don't get distracted by that.
2. The participants has to put himself into the situation of a life logging person.
3. The participants has to perform the task for each of the 4 scenarios in the video

The Participants first had to watch the current scenario and then select the faces in this segment by the rules described above.

5.9 Design

Each of the participants had to solve the given task. The independent variables in this study were the 4 different video arrangements and the dependent variables were:

1. The selected faces
2. Reasons for the selections
3. Face importance ranking

The goal was to figure out what faces were selected to be important, what indicates the importance of a person and how important the faces are ranked. This is important to determine if the decisions that were made in this thesis are somewhat correct and can also be applied to a larger scale. If the results differ from our own decisions then we have to look at how the algorithm can be improved in order to match all the necessary decisions that a human would perform. I have to mention that the selection done by human hand is, of course, always better than that of a machine since the human itself knows best what is important for himself. We also have to take into account the different preferences of human individuals. So the goal would be to come as close as possible to the decisions a human being would make. Therefore we cannot take every single person into account. But at least we can try to satisfy the majority of the people.

5.10 Results

The task was to select important faces and give reasons for the selection and a ranking of the faces. Since the reasons can vary quite a bit I had to break them down to the actual meaning behind it which then can be used to actually implement them. For example 6 out of 16 participants said that the person who appeared in the walking scenario (scenario 2) was showing the way. This information does not help to potentially improve the algorithm. Therefore I had to break it down to 2 information instead. What indicates that the person is showing the way? It basically means that the person appeared often in the video and was using hand gestures which lead to the impression of him showing the way. Hand gestures usually indicate some kind of interaction between the person using the hand gesture and the person wearing the camera. Lip movement and speech further help to identify the conversation but those are just alternative options since they are not that easy to implement. Hand gestures are the easiest one to identify since it would be possible to search for hands or skin color and then track the movement or similar means to identify hand gestures. That being said I created some main categories in which all given reasons in every scenario can be divided to. Those categories are the following:

1. Screen time (includes reasons like: “person is on screen all the time”; used 68 times by 10 participants)
2. Appearance frequency (includes reasons like: “Person is showing the way”, “person is the only one which can be seen multiple times in the scene” or “Person is walking with the cameraman”; used 15 times by 15 participants)
3. Holding a conversation (including reasons like: “person is talking” or “person is holding a conversation”; used 104 times by 16 participants)
4. Hand gestures as a sign of indication of interaction between the person and the camera man or another person in the screen (includes reasons like: “Person is showing the way” or “person uses a lot of gestures”; used 30 times by 11 participants)
5. Only person on screen (used 10 times by 10 participants)

6. Person is known in real life (meaning a person that appeared in the video is known to the participants; used 14 times by 6 participants)
7. Unknown person (opposite of 6.; used 7 times by 3 participants)
8. Person sits in center of screen (meaning the person sits in the center of visual focus of the person wearing the camera; used 5 times by 3 participants)
9. Person is wearing a colorful t-shirt (used 4 times by 2 participants)
10. Person is eating (used 8 times by 8 participants)
11. Eye contact to camera (used 13 times by 4 participants)
12. Facial expression (includes reasons like: "Person is grinning a lot"; used twice)
13. Person is near important person (means that a person is sitting next to another person who is extremely important in the scene, for example if there is a conversation and a person is sitting next to the people who are talking but is not directly part in the conversation; used once by 1 participants)
14. Person is not part of the conversation (used 8 times by 8 participants)

Those are the main categories which cover all the reasons that were given.

I have to address how the categories were actually build. The progress of creating a category was basically looking through all the reasons and then creating a category for each reason. Once there were similar reasons I packed them together into one category if the actual meaning behind those reasons were the same. So even if a reason was used only once there is still a category for it, unless the reason is packed together with other similar reasons. Category 13 for example was based on one single reason that only one participant was given. Category 12 was also only used twice. That being said, I included every reason that was given even though it was only used once. Of course there are way more reasons than the ones that were given. We only had those 4 scenarios so it is highly possible that there are other more important criteria for other situations. Therefore the mean value of how many times a face was selected is 1 and the standard deviation is 0.

An interesting thing is that the participants had the option to select the faces multiple times if the importance would change. But nobody actually made use of this option. Maybe they were just too lazy. If the importance changes of a person that usually resulted in a comparison in the reasons, for example "Person on the left was talking more at the beginning than in the end".

Category 7 (unknown person) was very specific since almost none of the participants actually knew the people appearing in the video. This reason was only given when there was also a person that they personally knew in the scene. Most of the time this reason was not used even though it applied to almost everyone. I have to address this since the factor of knowing a person or not mostly decided if the known person was ranked with a higher rating than the one they did not know. This case only happened in the meeting and eating scenario so it was not important for the other two. I will talk in detail about this issue at the end of this section.

I have described the 4 different scenarios in the sections above. To get some useful results I will present and discuss the results for each scenario separately. Therefore it is necessary to name the different persons that play a main role in the scenes. Those persons were also

the only ones that were selected. For the presentation of the results of each scenario I display the following information:

1. How many participants selected the person in the scene
2. What reasons were given and how many times was this reason used to justify the selection and the importance ranking
3. The importance ranking of the persons

To display the exact results I will treat each scenario on its own and in every scenario I will further divide it into the different person that were selected in the scene.

5.10.1 Scenario 1: Dialogue

In this scenario there was only one person on the screen.

How many participants selected the person in the scene?

The person was selected by every single participant.

What reasons were given for the selection?

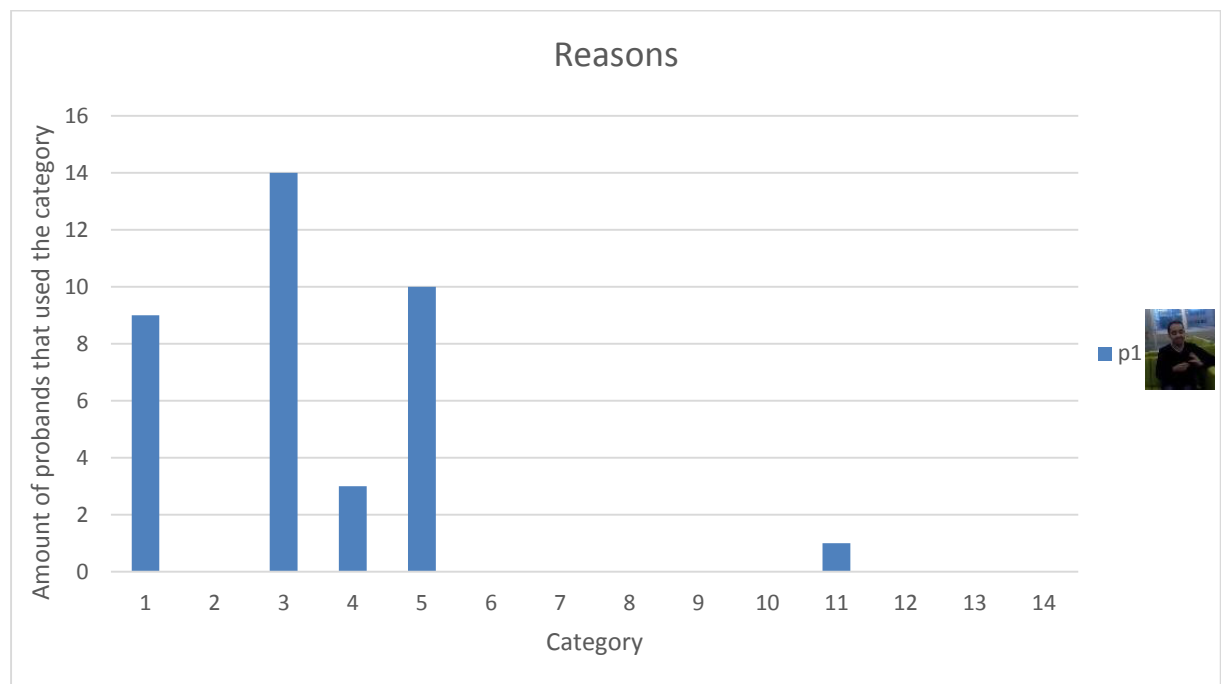


Figure 20. Scenario1: Reasons used

As we can see here that the most prominent reasons for the importance of the face was mainly concentrated on the screen time of the person. Additionally the person was the only on the screen which of course resulted in a high value for category 5. The most important part was the conversation category. Almost every participant said that the person was important because he is talking. Some additional reasons were based on hand gestures and eye contact which also seemed important. In the context of this scene the hand gestures and eye

contact could also be considered together with the conversation part. The conversation is mainly based on a person talking but eye contact and hand gestures is also important to make a conversation interesting.

How important was the person ranked?

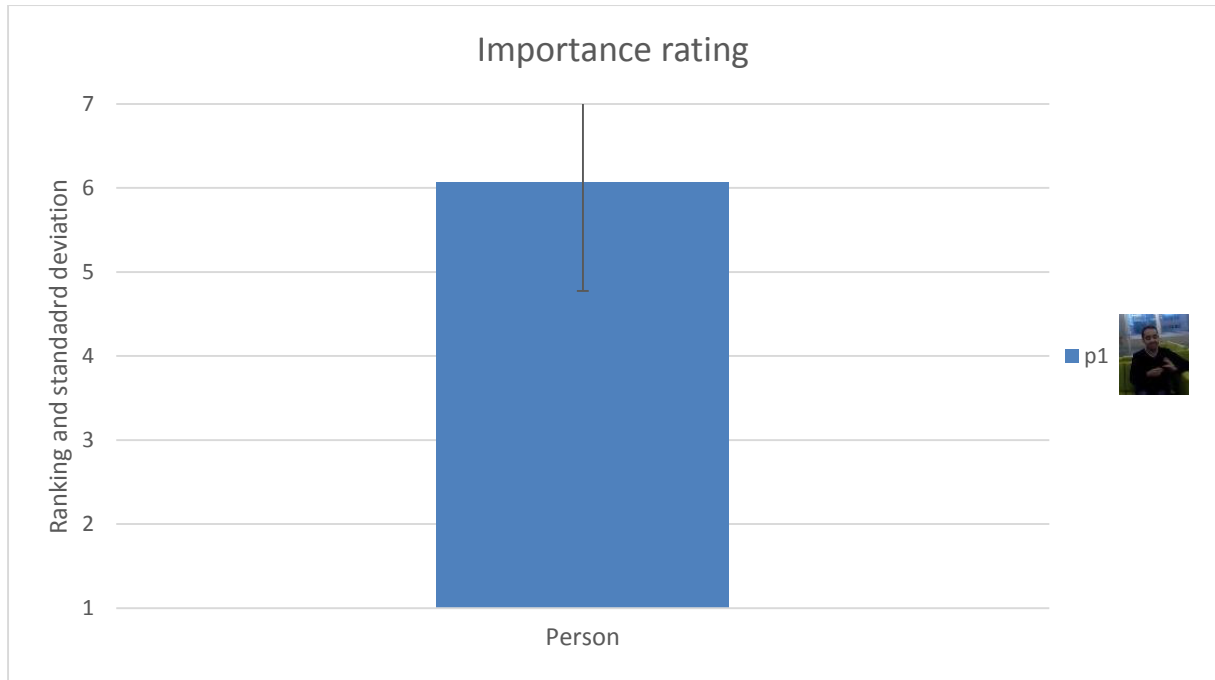


Figure 21. Scenario 1: Importance rating

The mean value of the ranking was 6.06 with a standard deviation of 1.3. So the overall ranking was pretty high but there were still a few rankings in the average region.

5.10.2 Scenario 2: Walking

In this scenario there was one person walking with the person recording. While walking through the building there were multiple people in the background.

How many participants selected the person in the scene?

The person was selected by 15 participants.

What reasons were given for the selection?

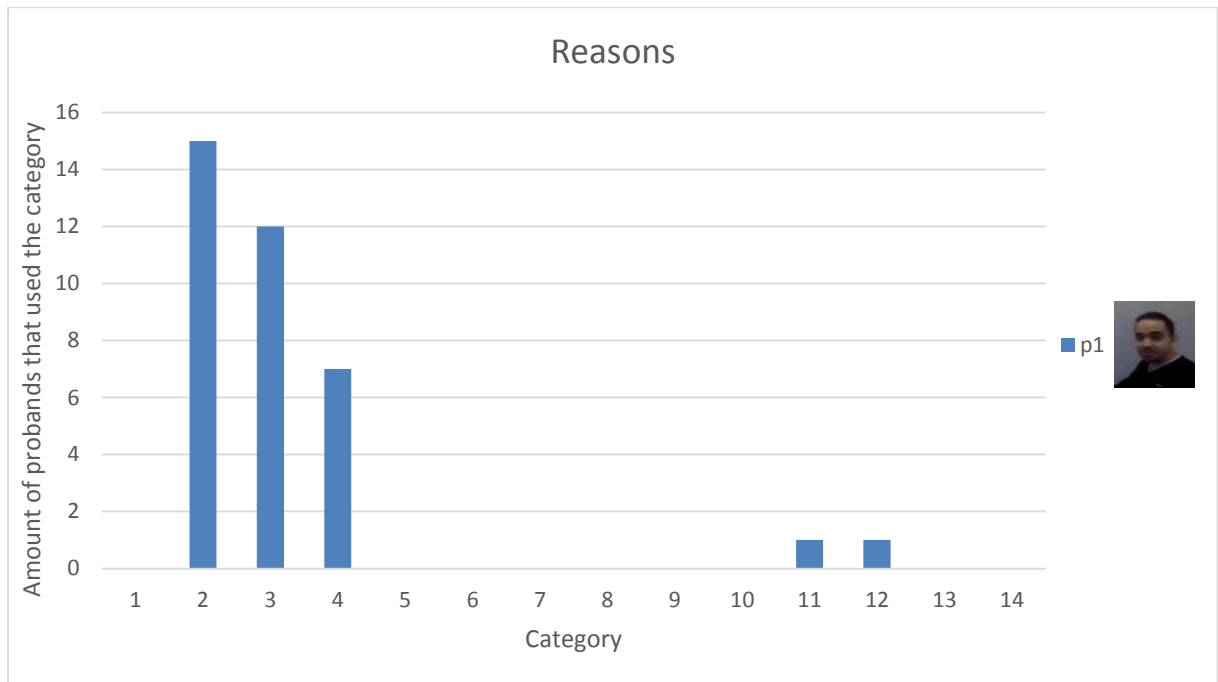


Figure 22. Scenario 2: Reasons used

Here the reasons for the importance of a person while walking also seems to be quite clear and almost the same as in scenario 1. Instead of screen time here the amount of appearances matters more. If you take those categories a little bit more lose than category 1 and 2 can be basically considered the same. The more a person appears the more screen time he has. But since there is still a difference between being still there and appearing all the time I decided to make two categories for this. The conversation part also seemed important. Additionally many of the participants stated that it looked like as the person on screen was showing the way. As I mentioned at the definition of the categories I treated this statement as the appearance frequency as well as hand gestures. That is why the hand gestures are also important in this scene. Another reason for the importance was also the eye contact that the person has with the camera and also that the person was grinning sometimes which resulted in the facial expression category.

How important was the person ranked?

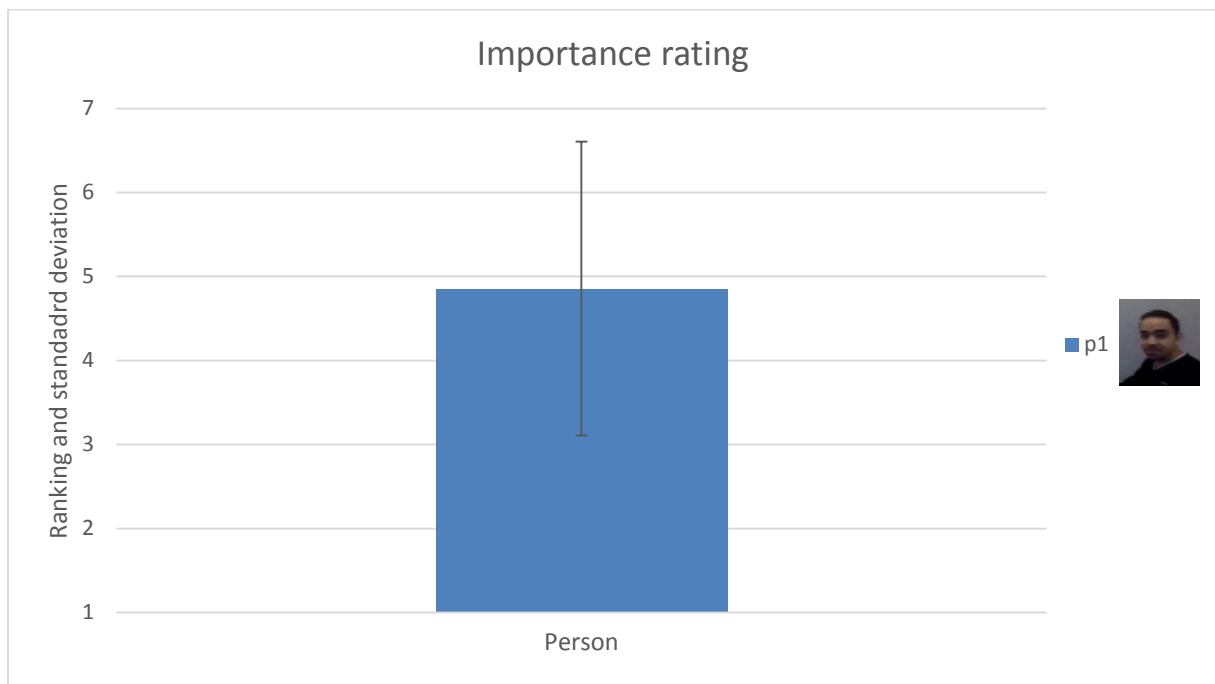


Figure 23. Scenario 2: Importance ranking

In this scene the mean ranking value was 4.85 with a standard deviation of 1.74. This shows that overall the person was ranked above average. However there are still some lower rankings. The ranking here is very subjective for the persons as it seemed that even though the person was walking with the camera for some participants the walk itself was more important than the few seconds the person was on screen.

5.10.3 Scenario 3: Meeting

In this scenario there were 3 people. The numeration of the people is based on their position on the sofa. The person on the left is person 1, the person in the middle person 2 and the person on the right person 3.

How many participants selected the persons in the scene?

All three persons were selected by every participant.

What reasons were given for the selection?

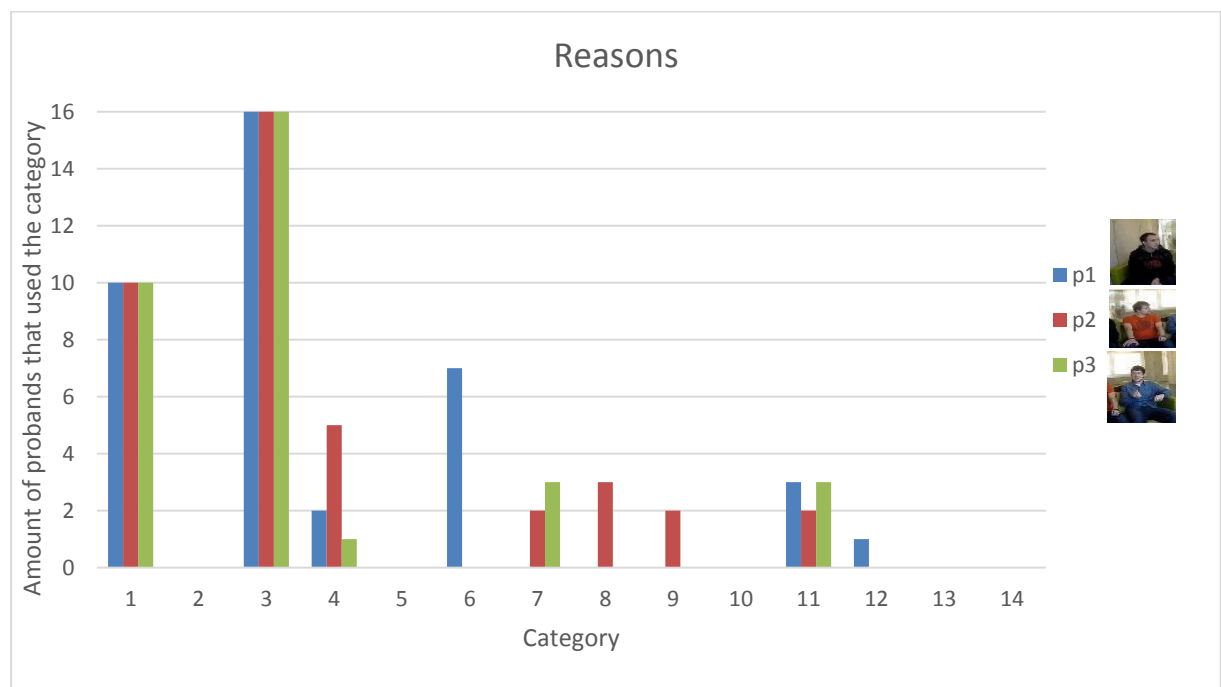


Figure 24. Scenario 3: Reasons used

Here we have a colorful mix of all kinds of different reasons. Most of them are based on the conversation part. This includes the conversation itself, hand gestures and eye contact. The screen time also plays an important part again. This time there was also the issue of knowing a person in the video. All of the participants that used this reason knew person 1 but not the other two. Additionally this time person 2 also seemed to get more attraction because he was sitting in the center and was wearing a bright t-shirt (“Person is wearing a bright orange T-shirt”, Participant X).

How important are the persons ranked?

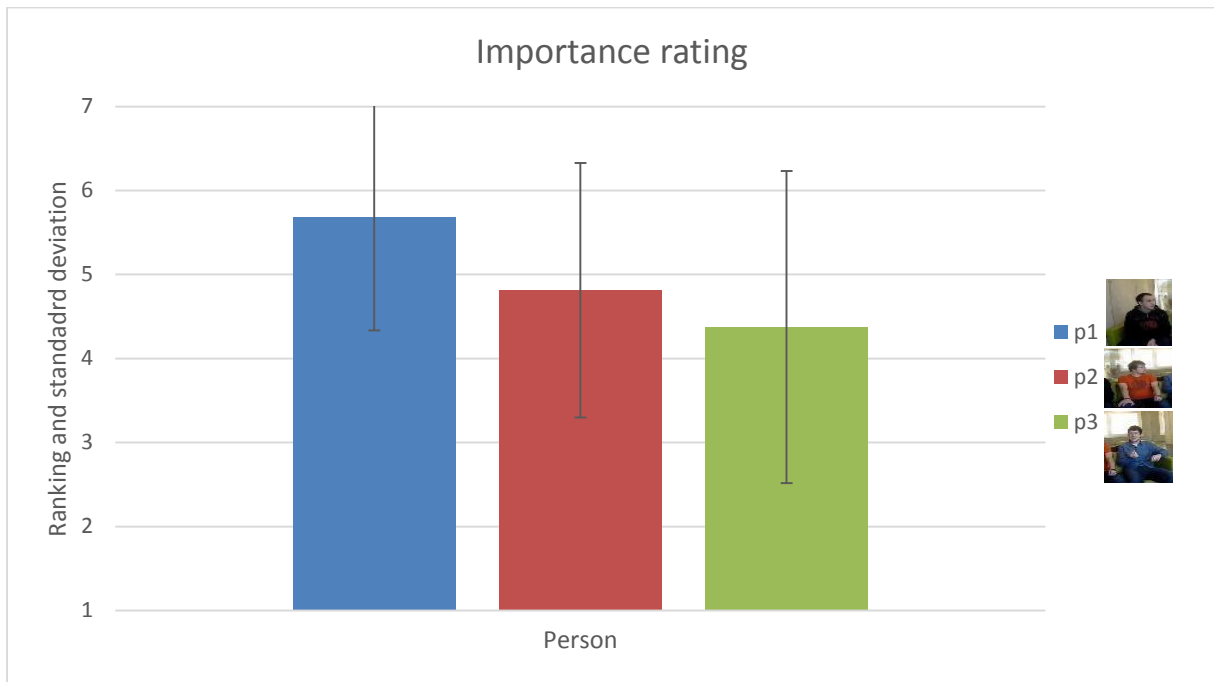


Figure 25. Scenario 3: Importance Ranking

In this scenario the importance ranking changes a lot. We can see that person 1 was ranked the highest (mean value of 5.68) before person 2 (mean value of 4.81) and person 3 (mean value of 4.37) was being ranked the lowest. We can also see that the variation is quite high. The standard deviation for person 1 was 1.35, for person 2 1.51 and for person 3 1.85. The conversation part played an important role for defining which person is more important. But we also see that the fact of knowing a person in real life also helps in resulting in a higher rating. Person 2 was getting slightly more attention due to his clothing and his positioning in the center. But the overall difference between person 2 and 3 is not that high. The ranking is mostly based around the amount of conversation time each person has.

5.10.4 Scenario 4: Eating

In this scenario there were 3 main people in the focus and multiple people in the background and at the side of the camera view. The numeration of the people is based on their position at the table. The person on the left is person 1, the person in the middle person 2 and the person on the right person 3.

How many participants selected the persons in the scene?

Person 1 and 2 were selected by all 16 participants.

Person 3 was selected by 11 participants

What reasons were given for the selection?

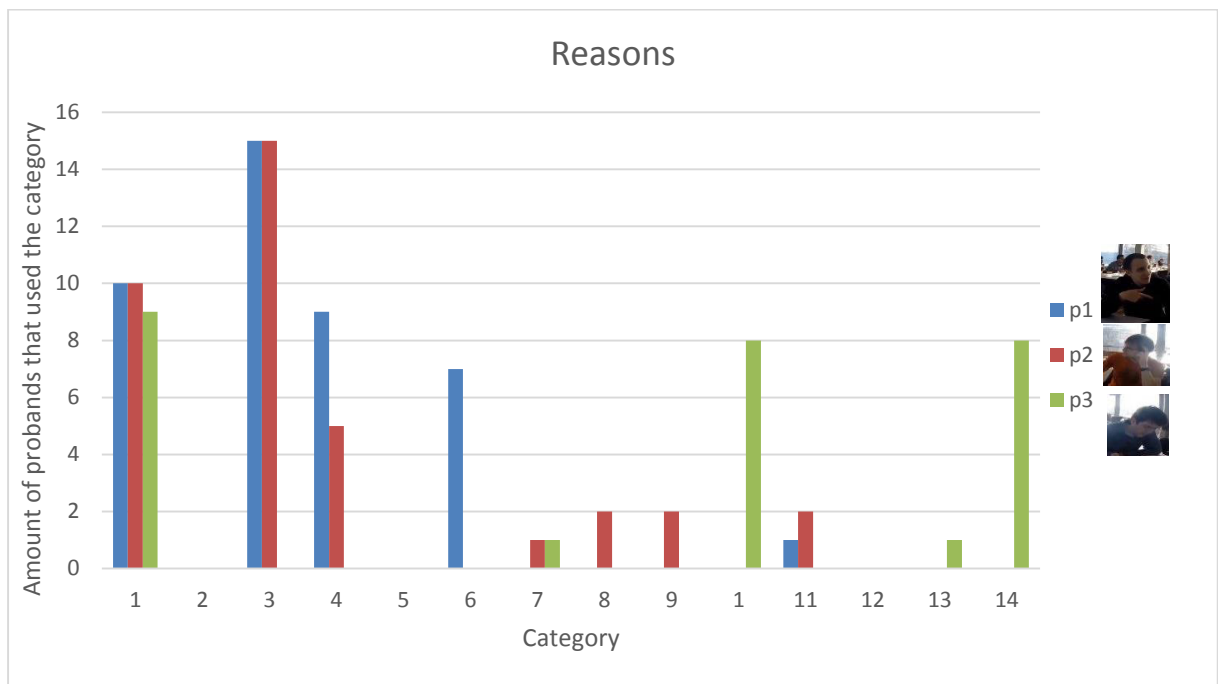


Figure 26. Scenario 4: Reasons used

The reasons for person 1 and 2 are mostly the same. Again the screen time and conversation part is the most important part of the scene. Just like in scenario 3 the personal known component plays a major role here too (category 6). Additionally the sitting order of the three people is the same as in scenario 3 which again results in the mentioning of the colourful outfit of person 2 and also that he sits in the middle. The more interesting part is person 3. This person did not engage in the conversation of person 1 and 2 and was solely concentrating on eating his meal. Therefore he was still one the screen all the time but did not take an active role in the scene. That results in the reasons of him not engaging in the conversation.

How important are the persons ranked?

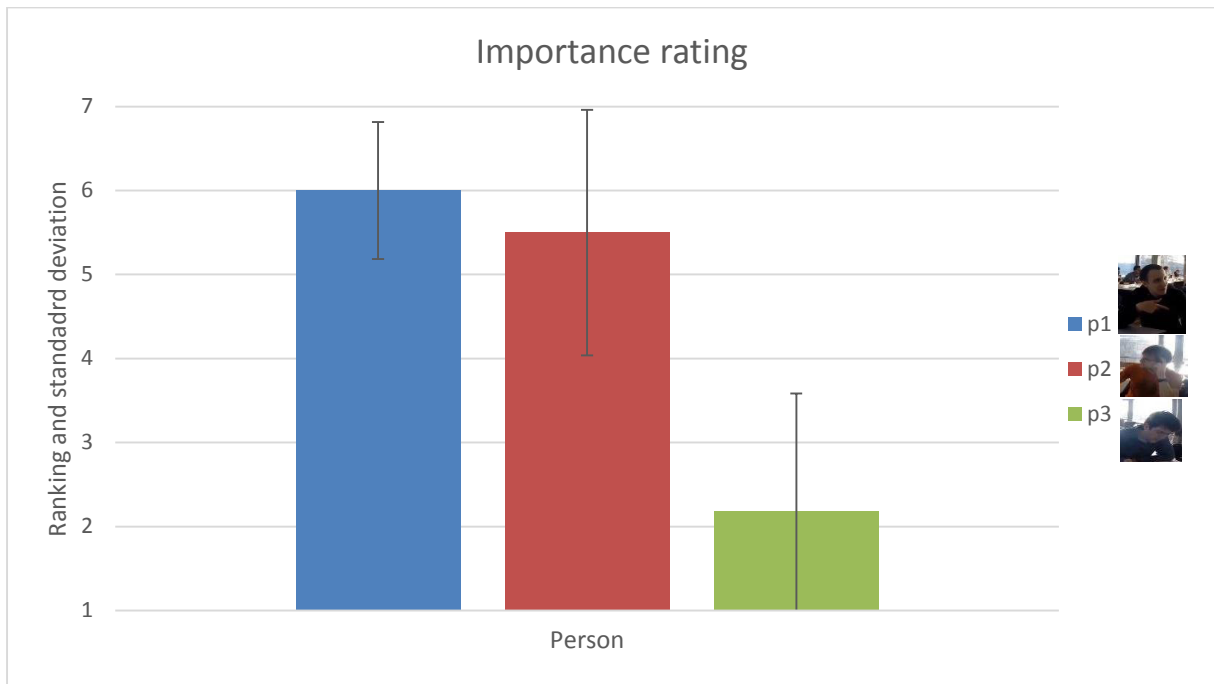


Figure 27. Scenario 4: Importance ranking

Here we have a clear picture again. Person 1 and 2 are kind of ranked the same, person 1 being a little bit higher (mean value of 6) ranked than person 2 (mean value of 5.5). For person 3 the ranking was extremely low overall with a mean value of 2.18. The standard deviation is also higher for person 2 and 3, with a value of 0.81 for person 1, a value of 1.46 for person 2 and a value of 1.4 for person 3. For person 1 we can see again that category 6 is probably the strongest indicator of why he is higher ranked as person 2. The also seems to play a part in the higher variance of the ranking of person 2. The ranking for person 3 also varies quite a bit since there was one participant who ranked him with 6 which heavily affected the variance since the person was also only selected by 11 people.

5.11 Discussion

The results of the user study are shown in the section above. Now we have to interpret and discuss those results in order to come to a conclusion. In the following I will discuss and interpret the results of each scenario.

Scenario 1: Dialogue

For the dialogue scenario the results are pretty clear. There was only one person on the screen which was also selected by every participant. But that was already expected. What is more important are the actual reason for selecting and also the importance ranking. The reasons for selecting the face are mostly the same for all the participants. Of course there were some minor differences between them but the major part was all based on the screen time, the conversation part and the fact that the person was the only one on the screen. Hand gestures were also important for some people here. I personally did not look at the hands that much since for me it seemed like the person was using normal hand gestures that everyone would use in a conversation but this of course shows that the person is engaging actively in the conversation so it can be considered an important aspect of identifying a conversation. The importance ranking also seems to be quite stable. There was not anything else in the scene despite the one person. Therefore the ranking was mostly prominent in the latter third of the scale. I believe that the ranking is also influenced by personal preference. There were some rankings below 5. I would imagine that those rankings are probably based on the boring scene. It was not an interesting video because the participants only had to look at the visual part and did not have access to the audio. Surprisingly those low rankings had the same reasons as the higher ones. This probably means that the ranking is also driven by the mood the user gets by watching the video. The conversation was boring to watch and therefore the ranking was probably lower.

Scenario 2: Walking

For the walking scenario the results are also straightforward. I personally do not understand why one person did not select the person which was walking with the cameraman. But that does not change the rest of the results that much. Maybe the walk was more important for him than the actual faces appearing in the scene. The reasons are also mainly concentrated on the conversation and the multiple appearances part. Since the scene was filmed while walking the person recording does not have the option to always look at the person walking with him. The only time when the person is seen is when the recorder is looking at him. This also suggests that they are talking while walking because otherwise there would be no real reason to look at the face. During the short time the face can be seen it is also clear that the person is talking. Another interesting part is that some people said that it seemed like the person was showing the way. This was not actually the case since I (who was recording) actually said where we should go. But the fact that the person was making hand gestures and sometimes walking in front of me let it seem like he was showing the way. That was an important reason for some participants. For the actual results it does not change that much but it was interesting to see. It also becomes clear that hand gestures are a very important feature in order to remember some situations better. 2 additional reasons were based on the person grinning some times and the eye contact which of course can also be an

important indicator for remembering. The importance ranking for this scene are mostly pretty clear. The main ranking was in the latter third which means that the face was actually important. The fact that some participants ranked the person lower is probably due to the fact that we were walking all the time. As soon as you start walking the focus goes from the faces to the surroundings. The conversations are mostly based on audio at this point. That being said I can understand the reason for ranking the face lower in this scene. The only interesting part is, that despite the low rankings the reason for selecting the faces were the same as the one for the higher rankings. That probably means that the person was still noticeable by the same features but the personal impression is different for each person.

Scenario 3: Meeting

Here again the choice for selecting faces was pretty clear. There were 3 people sitting in front of the camera and talking. The reasons for the selection and ranking are mostly the same for all three. Again the screen time and conversation part played the major role here. For each person there was also additional reasons given. For the first one some people actually knew the person personally, which therefore resulted in the mentioning of that fact. The second person was wearing a t-shirt with an eye-catching color which made him more noticeable. The fact that he was sitting in the middle also helped him to get more attention. This reason was only given by one or two participants so I would not give it that much attention. The ranking was not affected by those little facts anyway so they can be ignored for the moment. The issue of knowing a person however plays a major role, especially when you want to remember faces. Since the 3 persons in the scene are all talking during this conversation they are mostly the same in terms of visual observation. This means that small differences are more noticeable between them. As I mentioned the fact of already knowing a person is quite important. That is also the reason why person 1 was ranked overall higher than the other two. The reasons that some of the participants knew person 1 is that this person was actually me and I asked some of my friends and family members to do this study which resulted in some of them mentioning that person 1 was more important. Another important indicator to determine the importance was the time each person was talking in the conversation. Since the video was very short and it was not already clear who was talking at each time the ranking was varying quite a bit. Some participants said that person 3 was talking the most, others said that one was talking more than the other 3 and some said that person 2 was just listening the whole time. The main indicator for knowing when each person was talking was to look at the hand gestures. Because the person who is talking is moving his hands more than a person listening. I believe that the huge variance in the ranking and also in the reasons is due to the lack of audio material. The lighting was not the best and therefore it was sometimes hard to tell which person was talking when he was not using hand gestures at that time. But still, it became clear that the person who is talking the most is more important than a person who is just sitting there and listening. The second issue that became prominent is the importance of knowing a person beforehand. With this information we can try to further improve the automatic labeling of the faces.

Scenario 4: Eating

The last scenario was the most interesting one in my opinion. We had 3 people in the focus, where two of them were talking all the time and the third one was just eating. Additionally we had multiple people in the background and even a person to the left who was on screen sometimes when the camera was looking to the left. The results were kind of what I had expected them to be. Again the main focus lies on the conversation of person 1 and 2. Therefore the reasons for selecting them were also concentrated on this aspect. The reasons for selecting them are the same as in scenario 3 so I will not go into the details here again. The more interesting thing was to see how the participants handle person 3 who was playing the unimportant one. Surprisingly for almost half of the participants this person seemed to be the one that was demanding most of the attention. The participants said that he was so concentrated on eating all the time, completely uninterested in the conversation. But the ranking for him was still pretty low. Here I have to mention the issue of the difference between noticeable and important. Every participant who was focused on person 3 still ranked him very low despite the one person who ranked him with importance 6. That means he was the most interesting person to look at but also had the least relevance to the scene. For some participants he was not even worth selecting in the first place since the main part of the scene was the conversation between person 1 and 2. So in conclusion we can say that taking part in a conversation is more important than watching a person eating all the time. But the fact that person 3 was still selected is mainly due to the fact that he is sitting in the focus of the field of view and is also of considerable size in the scene.

One important thing that I have to address is the how the participants usually came up with their reasons for selecting a person. We can see in the results that the screen time of a person plays an important part in the selection. However only around 10 out of 16 wrote down this reason even though it applied to every person on screen despite for scenario 2. I believe that some of those reasons still applied to the situation but were subconsciously suppressed by the participants because they took it for granted. Another example is the conversation reason. Almost all the participants gave this as a reason why a person is important. But when we look at scenario 1 only one participant said that the eye contact was important. Eye contact is an important part for every conversation because it basically tells us which person the person in front of us is talking to. It can make a huge difference between actually engaging in a conversation and only listening to a conversation of two other people. Another issue I have already mentioned above is the aspect of already being familiar with a person that is seen on screen. It did not make any difference to change the order of the scenarios. The reasons for selecting faces were still the same, but already knowing a person beforehand makes a huge difference. As I have mentioned almost all the people in the video are unknown to the participants. But the reason for a ranking due to the fact that I am unfamiliar to this person still affected the importance ranking. Also one of the participants said that if she would watch this video 5 years later she would only remember that the person she knew was in this scene. The other people would be probably already forgotten years later. Of course it also makes a big difference if I have personally been in this situation that I am watching. The participants did not know the situation so they would have probably acted different if they had personally experienced it. All of those issues show that it is

extremely difficult to determine important faces in a video. It is not an easy task and will require much more work than what is covered in this thesis.

In conclusion I will summarize the most important information that I got by conducting this user study

1. A face needs to be on screen to actually be important. This is a necessary condition but not a sufficient one, see scenario 4. Here the person was still on screen but not important. But still we was in the focus of the camera unlike people in the background which were not important at all. As I said this condition is necessary for people being selected in the first place but is not sufficient for a high ranking. It needs to be in connection with other reasons, especially conversations.
2. Identifying people who are talking and figuring out which person is part of a conversation is important to determine an importance ranking. This means that having a conversation is more important than a person just being on screen and standing still. Therefore it is important to figure out how a conversation can be detected. Usually one big part of remembering a conversation is what was said and with who you were talking to so this is one if not the most important part to figure out important people.
3. Being familiar with a face place a major role. This is important because people tend to remember conversations better if it was with a person they actually knew for a long time. I also asked some participants what they would remember if they were recording the videos. One said that she would not remember anything besides person 1 being in the scene since she does not know the other persons. Another person said that she still remembers what her daughter said 10 years ago. So I asked her if she would still remember the same thing if it was being said by a random stranger and she said the she would probably not remember it if that were the case. So it seems that knowing a person in real life is an important part for life logging videos. When I record my whole life than I will have mostly the people on camera that are more important to me to begin with and I want to know more about their lives than some random person. That being said it is necessary to figure out which people are known in real life and also close to me as a person.

No we have to see what we can actually do to improve the current algorithm. The screen time part (category 1) is already taken care of so what is left are the conversation aspect (category 2) and the familiar face issue (category 3).

How can we determine if a person is holding a conversation?

The most naive approach that comes to mind is to use image processing to identify the person who is talking. I believe this would be a very hard thing to do and would require a face tracker to actually work. Luckily I already have implemented one. The problem is still to figure out the mouth movement when a person is talking. Since the mouth consist only of a few pixels it will be really hard to do this. I believe it would be easier by analyzing the audio material. This would mean to assign each person his identification voice. By doing

this it would not only be possible to determine the person talking it would also improve the face recognition since we can link the voice to the person it belongs to. This will also deal with the problem to determine if a person is still around even if the face detection does not detect the face. At this point we do not concentrate on audio since it is also a complicated matter and also a very delicate topic considering the whole privacy issue.

Another problem that occurred is when the recording person is watching a conversation instead of engaging into one himself. If that is the case then we do not only have frontal faces which can be detected. Most of the time we would have profile faces. OpenCV also provides Haarcascade classifier for the profile face detection. This detector is not as reliable as the frontal face detection. The results are worse and the false positive detection is higher. But since we also have a skin detector those false positives can be heavily reduced which makes the profile face detection a valuable extension. The major downside is that the processing times goes up again. Another big problem is to successfully link the frontal faces with the corresponding profile faces. The normal algorithm would just see a frontal or profile face and saves it with a specific id. The problem is that the profile face and the frontal face of the same person would result in 2 separate ids. When a face is already recognized and changes from frontal to profile view the id is still the same. But then we get additional problems. What if a face has some frontal entries but no profile ones? If the face appears at frontal first that will not be a problem. But if the face appears with a profile view the algorithm does not know this person and will create a new one. Of course the chance of this happening is not that high but it is still a possibility. Another problem is that ones the face is lost the tracker is deleted which means that during the change between frontal and profile view the tracker will not be present anymore since all the feature points from the frontal face might be lost. This problem can be solved by extending the longevity of the trackers a little bit so instead of deleting them once the face is lost the tracker could be kept alive for a few frames. If it is not updated in a specific time interval than we have to create a new one. The obvious question here would be if the tracker should be deleted in the first place or wait for a longer time period. The problem with this is that the longer the pause between the face appearances is the higher the chance that the face belongs to a completely different person. If I look at a face then I usually concentrate the face to the center of my field of view. If I look at someone else than that person will be in the center again. Therefore we cannot keep the tracker alive for too long since the id will not be correct anymore.

In theory the profile detector should perform very well and in practice as a standalone it gives good results. The problem comes from integrating the profile face detector in the normal frontal face detection algorithm. The idea of tracking a face to make sure that a specific frontal face belongs to the corresponding profile face does not work that well. The main problem is the way the tracking algorithm works. It seems that just tracking the optical flow of feature points is not sufficient for this task since many feature points get lost during the movement or the calculation itself. Therefore we need a more robust face tracker in the future.

The last problem was the issue of identifying already known faces. This is not an easy task. The simplest approach would be to just see if the person already has an entry in the face database. But that method is nor very reliable since every person, who appears only once,

can have an entry. A better way would be to provide a database with all the meta-information. Every face appearance would have to be saved. The problem then would be that we would have a large amount of data very fast. But this will be necessary in the long run anyway since we cannot just take the information from the current scene. It will also be important to link this information with previous experiences. Another issue would be to actually look up the current recognized face in the database to see if the person appears more frequently or not. This would need to be done every single time a face is recognized which would also result in a lower performance than it already is. However it is still in scope of what is possible and it also does not require that much work to implement despite setting up an efficient database. One problem for the life logging process over the years is also the age of the people. A person ages over time and therefore the face also changes. That has to be considered when using face recognition. That means that the face recognition should focus on features that do not change over the years. The geometric aspect is one thing. The distance between the eyes and the nose and such usually does not change over the years. Using those features to recognize people will be probably better than just looking at some color values of the face in the current state.

6 Summary and Future Work

In conclusion I have to say that the whole idea of indexing life logging videos is not a simple task. It is extremely complex and very hard to achieve results that satisfy all the preferences of different people. Additionally we have to take every imaginable situation into account to make sure that the indexing makes sense in each of those situations. All in all this project is a first step into a much more complex and bigger one. The basic idea to use the information of time, places, faces, tasks and objects is definitely a promising one, but to detect the connection between all of those will be a complicated task in the future. That being said the algorithm provides reasonable results but they probably do not help people remembering the situations since there is so much information that is not included at the moment. The videos that were used to test the algorithm also only include some very specific situations. Due to the lack of huge amount of video material it is hard to tell how the algorithm will perform to segment those huge video files. Another issue I have to address is the performance. The video processing takes quite an amount of time. This of course also depends on the video. If there are more faces then the processing time can be almost twice as slow as it already is. Another issue here is the use of javaCV which can easily result in some memory leaks. That is also one of the reasons why the algorithm cannot be used for huge video material right now. Of course it is possible that the implementation was not done with performance in mind but the algorithms for image processing that were used also take quite some time. This whole project is only a prototype so we have to look into the future at this point. The algorithms for face detection and recognition for example are constantly improved and maintained. Additionally there are new ideas and algorithms developed all the time which means that the results and the performance will probably get better over time.

A huge part for conducting a bachelor thesis is to learn some new stuff. I will summarize the things that I learned during this time of development. The most interesting part of this bachelor thesis in my opinion was the image processing part. I have learned a lot about different algorithms like Viola Jones and the recognition algorithms and also some basic ideas of image processing and what can be done with images in the first place. The idea of finding textures in images and analyzing them is very fascinating. Apart from that I also learned a little bit about videos and different devices and what can be recorded with them.

I have mentioned in the introduction that this whole project mainly consists of two parts. The first is to actually get the information from the video files the other one is the display of those information. The goal is to provide a browsing tool for life logging videos. The first part was covered in this bachelor thesis. The second part is currently developed as well. The main idea is to create a new browsing tool which provides the user with a 3D view to navigate through all the events, faces and locations. The events and faces are displayed as blocks which furthermore are connected with other blocks if they have some connections, for example the same faces appear in them or they take place at the same location.

In summary I can say that the algorithm is working fine but I also have to address the problems that I had. Most of the problems of the specific aspects can be read in the corresponding sections above. I will now briefly summarize the main problems that I had. Most of the problems appeared on the image processing side. Another big issue was the idea to actually figure out the exact way the segmentation algorithm should work. For the image processing part the most difficult was to get the face tracking to work. I have mentioned before that I adapted the main idea for the face tracking algorithm from the pi_robot⁶ project. But since the algorithm was written in python I had to translate all the code into equivalent JavaCV code. Unfortunately there is now documentation for JavaCV which means that I had to figure out the idea of how the algorithm works and then build it new with JavaCV. Additionally the original algorithm used some different way to detect feature points. But I was not satisfied with those results and decided to use a SIFT feature detector instead. Another issue was the technical part since JavaCV is just a wrapper for openCV. The actual image processing happens in C++ the standard language for openCV. That means that JavaCV just creates pointers to C++ objects. That means that most of the errors came from this connection instead of simple Java code. So it was hard to figure out where the actual exception occurred and why if something went wrong. But despite that most of the problems could be solved by using common sense. OpenCV is actually pretty easy to understand and there are tons of examples and tutorials about the specific actions that can be performed.

The second big issue was to actually figure out the rules to define events in the first place. How can we decide if a face is actual important in the context of a scene and when not? That is also the reason why I conducted the user study. How can faces and locations be combined to create Events? How can we define events in the first place? Is it more important to go too much into details and create multiple events instead of one bigger events? All of those questions cannot be answered easily. For that reason I had to make a decision which made the most sense to me. The actual details of the decisions I made can be read in the sections above.

In conclusion I would also suggest some methods to improve the algorithm in the future:

1. Improve the information extracting, especially the recognition and tracking. The indexing algorithm can only work efficiently if the data it gets is actually correct. Face recognition in an everyday situation is still not an easy task to perform. But I am pretty sure that there will be more advanced techniques in the future. Also the tracking right now is only based on feature points and optical flow. This works to keep track of a more static face but as soon as there is too much movement or the face rotates too much the optical flow calculation will not provide correct results anymore. I believe there are already some better tracking algorithms out there that are worth testing. The question is if they perform as good on life logging videos as they do with their static webcams with which they are mostly tested.
2. Introduce object and task detection. Task detection is important since this was also one of the main reasons for identifying and interesting conversation. The actions we perform by using our hands also help to actually remember conversations. If a

person is talking while standing still the whole time seems more boring than someone who is using his hands.

3. Improve the segmentation. The segmentation is limited to some specific kinds of situations right now. Of course there are countless of situations that can be considered but I am sure at least some of them can be identified if the indexing algorithm goes into more detail. Of course for that to work the image processing part has to be improved first. The more information we have the more we can do with it.

References

- Al-Hajri, Abir; Miller, Gregor; Fong, Matthew, and Fels, Sidney S., 2014. Visualization of personal history for video navigation. In Proceedings of the 32nd annual ACM conference on Human factors in computing systems (CHI '14). ACM, New York, NY, USA, 1187-1196.
- Boreczky, John; Girgensohn, Andreas; Golovchinsky, Gene, and Uchihashi, Shingo. 2000. An interactive comic book presentation for exploring video. In Proceedings of the SIGCHI conference on Human Factors in Computing Systems (CHI '00). ACM, New York, NY, USA, 185-192.
- Chiu, Patrick; Girgensohn, Andreas; Lertsithichai, Surapong; Polak, Wold, and Shipman, Frank. 2005. MediaMetro: browsing multimedia document collections with a 3D city metaphor. In Proceedings of the 13th annual ACM international conference on Multimedia (MULTIMEDIA '05). ACM, New York, NY, USA, 213-214.
- Christel, Michael G., 2008. Supporting video library exploratory search: when storyboards are not enough. In Proceedings of the 2008 international conference on Content-based image and video retrieval (CIVR '08). ACM, New York, NY, USA, 447-456.
- Gao, Yong; Wang, Tao; Li, Jianguo; Du, YangZhou; Hu, Wei; Zhang, Yimin, and Ai, HaiZhou. 2007. Cast indexing for videos by NCuts and page ranking. In Proceedings of the 6th ACM international conference on Image and video retrieval (CIVR '07). ACM, New York, NY, USA, 441-447.
- Gemmell, Jim; Bell, Gordon, and Lueder, Roger. 2006. MyLifeBits: a personal database for everything. *Commun. ACM* 49, 1 (January 2006), 88-95.
- Haesen, Mieke; Meskens, Jan; Luyten, Kris; Coninx, Karin; Becker, Jan Hendrik; Tuytelaars, Tinne; Poulisse, Gert-Jan; The Pham, Phi; and Moens, Marie-Francine., 2013. Finding a needle in a haystack: an interactive video archive explorer for professional video searchers. *Multimedia Tools Appl.* 63, 2 (March 2013), 331-356.
- Jackson, Dan; Nicholson, James; Stoeckigt, Gerrit; Wrobel, Rebecca; Thieme, Anja, and Olivier, Patrick. 2013. Panopticon: a parallel video overview system. In Proceedings of the 26th annual ACM symposium on User interface software and technology (UIST '13). ACM, New York, NY, USA, 123-130.
- Joydeep Ghosh. 2012. Discovering important people and objects for egocentric video summarization. In Proceedings of the 2012 IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (CVPR '12). IEEE Computer Society, Washington, DC, USA, 1346-1353.
- Kopf, Johannes; Cohen, Michael F. and Szeliski, Richard. 2014. First-person hyper-lapse videos. *ACM Trans. Graph.* 33, 4, Article 78 (July 2014), 10 pages.

Krishna, Sreekar; Little, Greg; Black John and, Panchanathan, Sethuraman. 2005. A wearable face recognition system for individuals with visual impairments. In Proceedings of the 7th international ACM SIGACCESS conference on Computers and accessibility (Assets '05). ACM, New York, NY, USA, 106-113.

Li, Yuan; Ai, Haizhou; Huang, Chang, and Lao, Shihong. 2006. Robust head tracking with particles based on multiple cues fusion. In Proceedings of the 2006 international conference on Computer Vision in Human-Computer Interaction (ECCV'06), Thomas S. Huang, Nicu Sebe, Michael S. Lew, Vladimir Pavlović, and Mathias Kölsch (Eds.). Springer-Verlag, Berlin, Heidelberg, 29-39.

Lowe, David G., 2004. Distinctive Image Features from Scale-Invariant Keypoints. *Int. J. Comput. Vision* 60, 2 (November 2004), 91-110.

Ma, He; Zimmermann, Roger, and Kim, Seon Ho. 2012. HUGVid: handling, indexing and querying of uncertain geo-tagged videos. In Proceedings of the 20th International Conference on Advances in Geographic Information Systems (SIGSPATIAL '12). ACM, New York, NY, USA, 319-328.

Ma, Wei-Ying, and Zhang, HongJiang; An Indexing and Browsing System for Home Video, Hewlett-Packard Laboratories, 1501 Page Mill Road, Palo Alto, CA 94304-1126

Nunes, M.; Greenberg, S.; Carpendale, S., and Gutwin, C. 2006. Timeline: Video Traces for Awareness. In *Video Proceedings of CSCW 2006*.

Pongnumkul, Suporn; Wang, Jue, and Cohen, Michael. 2008. Creating map-based storyboards for browsing tour videos. In Proceedings of the 21st annual ACM symposium on User interface software and technology (UIST '08). ACM, New York, NY, USA, 13-22.

Rowley, H. A.; Baluja, S., and Kanade, T. 1998. Rotation Invariant Neural Network-Based Face Detection. In Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR '98). IEEE Computer Society, Washington, DC, USA, 963-.

Snoek, Cees G. M., and Worring, Marcel. 2005. Multimodal Video Indexing: A Review of the State-of-the-art. *Multimedia Tools Appl.* 25, 1 (January 2005), 5-35.

Uchihashi, Shingo; Foote, Jonathan; Girgensohn, Andreas, and Boreczky, John. 1999. Video Manga: generating semantically meaningful video summaries. In Proceedings of the seventh ACM international conference on Multimedia (Part 1) (MULTIMEDIA '99). ACM, New York, NY, USA, 383-392.

Wujie Zheng, Jinhui; Chen, Le; Ding, Dayong; Wang, Dong; Zijan Tong, Dong; Wang, Huiyi; Wu, Jun; Li, Jianmin; Lin, Fuzong, and Zhang, Bo, Tsinghua University at TREC-CVID 2004: Shot Boundary Detection and High-level Feature Extraction

Xu, Qianqian; Wu, Zhipeng; Li, Guorong; Qin, Lei; Jiang, Shuqiang, and Huang, Qingming. 2010. Memory matrix: a novel user experience for home video. In Proceedings of

the international conference on Multimedia (MM '10). ACM, New York, NY, USA, 927-930.

Yamauchi, Yasunobu. 2007. Multi-frame video representation using feature preserving directional blur. In ACM SIGGRAPH 2007 posters (SIGGRAPH '07). ACM, New York, NY, USA, Article 66.

Yi Chen and Gareth J. F. Jones. 2010. Augmenting human memory using personal lifelogs. In Proceedings of the 1st Augmented Human International Conference (AH '10). ACM, New York, NY, USA, Article 24.

Zhang, Li; Ai, Haizhou, and Lao, Shihong. 2006. Robust face alignment based on hierarchical classifier network. In Proceedings of the 2006 international conference on Computer Vision in Human-Computer Interaction (ECCV'06), Thomas S. Huang, Nicu Sebe, Michael S. Lew, Vladimir Pavlović, and Mathias Kölsch (Eds.). Springer-Verlag, Berlin, Heidelberg, 1-11.

Declaration

I hereby declare that the work presented in this thesis is entirely my own and I did not use any other sources and references than the listed ones. I have marked all direct or indirect statements from other sources contained therein as quotations. Neither this work nor significant parts of it were part of another examination procedure. I have not published this work in whole or in part before. The electronic copy is consistent with all submitted copies.

Place, date, signature