Westfälische Wilhelms-Universität Münster

- Institute for Geoinformatics -

Evaluation of Visual Feature Detection Algorithms to Implement Augmented Reality Applications for Indoor Environments on Mobile Devices

Philipp Weiß

A thesis submitted for the degree of
Master of Science

December 27, 2012

First reviewer: Prof. Dr. Christian Kray

Second reviewer: Thomas Bartoschek

Abstract

This master thesis describes the evaluation of different image recognition algorithms. The aim is to find an appropriate technique to provide real time indoor Augmented Reality applications. Therefore, the promising approach of using existing infrastructure in the form of images or shop logos instead of markers is verified. Furthermore, the most appropriate algorithm in detection accuracy and time, SIFT, is tested for its real time abilities. Several techniques of how to improve the not sufficient calculation latency are tested and discussed. With the gained information a prototypical Augmented Reality that is based on image recognition is developed.

Contents

List of figures

List of tables

1. Introduction

Through the rise of mobile technologies and their increasing processing power, there are many new possibilities how the world could be explored. In addition to the calculation speed of mobile phones, the sensors inside this small and powerful innovation can open new ways of discovering the environment.

All the small helpers, like for example the Global Positioning System (GPS), have already become companions in everyday situations. They allow adding a personal note to information. This way, navigation systems and other location-based services become usable without having to carry additional hardware.

Furthermore, the ability to track the user's position in combination with the possibility to access a nearly unlimited amount of information by the Internet makes it possible to provide custom location-based information. One of the most common and used applications is Google Maps [1]. This application provides map information and search functionalities to the user. Thereby, the known location of the user is used to provide location-based results.

One possibility besides the use of maps and GPS is to use the camera for the observations. In this area of application, the Augmented Reality (AR) is an important method [2]. An Augmented Reality has three typical characteristics defined by Azuma [3]. First of all, it combines the real and the digital world to provide additional information. The second property is the ability for real time interaction. The last characteristic is that an AR is always registered in 3D.

The technique of combining the real and the digital world brings a lot of benefits. In this field, the Augmented Reality is one of the main showcases. As people imagine, digital information can be integrated into their real world. Even before the technical options for a realization existed, people thought about the possibility to extend the real world with further information. But now, the science fiction has become reality, achieved by the improvement of technology.

This technique is already used successfully in outdoor environments [4]. For an implementation, most applications use the GPS position and the device sensors to register the orientation and the viewing angle of the camera. This way, digital information is displayed on the screen and the user can explore the environment in a new and intuitive way.

Precisely this is also desirable for the usage in indoor environments. To do so, it would be necessary to replace the GPS localization with other techniques that can be used. The possibility of Wi-Fi and other localization methods have the drawback that they do not provide the needed level of precision.

To avoid this, I thought about using visual detection for the placement of virtual information. Thereby, it would be the best option to use the existing environment for the localization of an object. This also means that no extra infrastructures like visible markers are needed.

All of this sounds like a great possibility for the usage in indoor Augmented Reality applications and like a good alternative for placing markers or installing cost intensive localization systems. There are only some facts that need to be clarified for the usage. There exist several algorithms and techniques of image recognition. In this thesis, I search for the best option to realize a fast and accurate real time Augmented Reality application on current mobile devices. I discuss the problems and the possible solutions for the usage of image based technologies to create an indoor AR application in real time. Therefore, different tests are described in the following chapters.

At first, I perform a test in a simulated environment. The information about the image detection algorithms is verified by an experiment with real world data for the recognition. With this information I identify the most appropriated algorithm for the detection on mobile devices.

In addition, techniques for the improvement of the calculation time are tested and evaluated. To verify the gained knowledge, I prepare a user test that evaluates the different possibilities to decrease the latency in the recognition process.

Finally, I present the prototypical implementation of the preferred settings evaluated in this master thesis.

The main aim is to determine how image recognition could be used to create real time Augmented Reality applications. The final results are necessary to implement a prototype of an AR application.

2. Related Work

There are two main ways how an Augmented Reality can be used and how the area for the augmentation can be spotted.

In the case of the current location of the user being known, it could be used to calculate the relation between points of interest and the user. Feiner et al. [5] presented one of the first wearable location based AR applications using this technique. This procedure is known as location based Augmented Reality. Location based applications are excellent to help the user find his orientation and explore his environment. One of the biggest disadvantages is that there is no all-encompassing solution for the localization in areas where no GPS is available – like inside of buildings.

There are approaches like in the paper of Pandya et al. [6] to localize the user without using GPS, but they need much effort for the preparation and are contingent on high costs. Often the localization is even inaccurate in a way that no precise positions for the digital information can be spotted. The problem in indoor environments is that the space between the user and the digital information is smaller than in outdoor situations. A typical use case for an outdoor AR is to visualize information on top of buildings in the city center, including large ranges between the user and the destination objects. This means that the accuracy of the GPS devices [7] is not that important for outdoor AR applications. The deviation is still so small that the Augmented Reality can provide good results for larger distances. In an indoor environment, in a usual building, a deviation of five meters is a very large distance. Five meters could mean that you are not even in the right room for the visualization. This shows that for indoor location based Augmented Reality applications there is a higher accuracy needed than for the outdoor usage. Even an accuracy of two – three meters in mean like reached by Bahl et al. [8] can be a problem in small rooms.

The other technique is to use the camera to explore the world. Therefore, markers can be used to identify objects and locations where information should be provided. This method is called marker based Augmented Reality and is already used in different projects [9], [10], [11]. This approach has the advantage that the complex and cost intensive indoor localization is not needed for the representation of digital information. The device can detect the locations where information should be displayed by simply identifying the markers. With this technique arises the problem that markers must be placed in all locations where digital information should be presented to the user. In many situations this is not welcome.

Another subtype of the marker based AR is to use image recognition for the detection. Obviously, the biggest advantage is that there is no extra infrastructure needed. Furthermore, it is possible to extend the existing environment with useful digital information and interactions. This technique can be used with every unique image, poster or shop logo. This way, it is possible to use the benefits without the unintentional placement of visual markers like 2D barcodes.

There are already applications that were developed to run Augmented Reality applications with the usage of image recognition [12], [13], [14]. In the paper of Skrypnyket et al., visual based Augmented Reality is implemented. The problem in this field is the platform they use. The implementation on a computer has the ability to provide real time data, while a mobile device has a lack of calculation power to provide this. This work concerns the problem of how current mobile devices can handle detection with common image detection algorithms.

Also, the possibility of providing information by localizing the device in combination with image recognition is a promising approach for an indoor AR. The rough positioning can be done with techniques like WiFi or Bluetooth localization [8] [6]. In the paper of Bernardos, this approach is discussed and the localization is implemented [15]. Further information about the different image recognition techniques and their potential can be helpful to complete their idea of a localization and image recognition combined Augmented Reality.

Another approach to realize indoor Augmented Reality applications on mobile devices is presented by Fakhreddine Ababsa [16]. The idea of the application is to estimate the camera position by identifying visual markers. This idea seems to be a good alternative to identifying every single marker in the room. But this indoor Augmented Reality again has the drawback of unwanted visual markers.

This idea is continued by Mormitsu et al. [17]. The visual method they use for the localization is marker independent. With the help of a feature point detection and the calculation of key graphs, the camera can be localized. The approach of identifying the position with the help of feature point detection is also mentioned in other projects [18], [19], [20]. Here, additional knowledge in feature point detection could also help to optimize existing projects in their accuracy and their calculation speed.

In the master thesis of Henrik Bauer [21], an approach that uses feature point detection to develop an Augmented Reality API is published. His work has the aim to simply allow users to set up their own Augmented Reality application with real time detection and visualization. His work focuses on the usage on a desktop computer. This means that a code transmission for a mobile device may cause difficulties. The major problem is the calculation speed. With detailed knowledge about the possibilities of a current mobile device it could be possible to reuse such an existing code and port it to a mobile platform.

The problem of guarantying a maximum frame rate for the real time detection is mentioned by Franz Lorenz Wendt [22]. He presents an Augmented Reality based on feature point detection techniques. He comes to the conclusion that image detection is a perfect technique to provide well usable Augmented Reality applications. However, he is not able to sufficiently reduce the calculation time to perform a smooth real time usage.

There are various approaches to solving the problem of marker less recognition. Thereby, it is conspicuous that most existing projects use desktop computers for the implementation. For mobile devices, only a handful of approaches exist. The most important barrier in the usage of mobile devices is guarantying real time calculation. Therefore, it is absolutely necessary to know exactly how fast and accurate the existing algorithms are. This knowledge gap is filled here.

3. Theory

For the implementation of the different image recognition algorithms, I use the OpenCV 2 library [23]. This library includes the common algorithms and techniques for computer vision. Furthermore, this continuously maintained tool is used in multiple projects that deal with computer vision. I choose six techniques to recognize images.

The first is called Scaled Invariant Feature Transform (SIFT). It was first published in Lowe [24]. He presents a method that transforms images into a collection of feature vectors. With this, it is possible to detect images independently from their scale, their rotation or their angle. The license rights of this algorithm belong to the University of British Columbia. However, it can be used for non-commercial purposes.

The second alternative is called Speeded Up Robust Features (SURF). It was invented by Bay [25]. The algorithm uses vector information to compare and recognize images. It is also under a non-commercial license. Typical characteristics described in the paper are the robustness and the calculation speed.

Another image recognition algorithm implemented by the OpenCV 2 library is called Maximally Stable External Region (MSER). Michael Donoser and Horst Bischof introduced it in the year 2006 [26]. The algorithm applies temporal information for a fast and stable tracking. Basically, it is used in projects where the complexity of other algorithms leads to high latencies [17].

A feature detection technique introduced in 1994 is called Good Features to Track [27]. The algorithm uses a model of affine transformation to calculate feature points in the image. This model is developed to track features with a minimum of calculation latency and this minimum is the reason why this model is used in many existing projects where an accurate object tracking is needed. One example for its employment is the usage for vehicle tracking [28].

An approach using corner detection to identify images is called FAST. It was introduced by Rosten et al. [29] and Rosten et al. [30]. The algorithm is designed to provide a maximum calculation speed combined with accurate detection results.

The last used feature detection method is the STAR detector. This is a detection technique that is derived from the CenSurE detector [31]. The OpenCV library implements the STAR algorithm. It uses approximations of circle shapes to analyze the feature points in the image. Therefore, rotated squares are used as starting points [32].

During all the tests, I use the techniques and the settings suggested and implemented by the OpenCV library. There are several ways to implement and execute the whole procedure. To avoid calculating all the possible contingencies, I act as similar as possible to the original OpenCV implementation.

4. Methodology

4.1 First Application

In the beginning of my master thesis, I have to explore the usage and the handling of different image recognition algorithms. To do so, I choose to implement an application that allows executing the previously presented image detection algorithms provided by the OpenCV 2 library.

The aim of this implementation is to explore whether the existing algorithms can already handle the image recognition on a current mobile device in real time. Furthermore, the application has the objective to give an overview about the user handling and a subjective knowledge of how effective the different detection methods are.

This test shows if a further investigation of the algorithms is needed to implement a mobile Augmented Reality application. Based on the results of this experiment, I decide how a further analysis of the algorithms can be designed and what kind of objectives must be reached to answer how a usable indoor Augmented Reality application based on image recognition can be realized.

4.2 Test Environment

In this test, the assumptions of the first application is investigated in a more appropriated and previously defined way. The found drawbacks like the high calculation time of different algorithms, as well as the widely scattered accuracy of the different detection methods are analyzed. Therefore, the subjective impressions of the first test are evaluated with reliable data of this standardized experiment.

The standardization of this experiment is necessary to provide reliable information for every algorithm in every situation. This is the reason why I choose to use photo series that are calculated with every algorithm provided by the OpenCV 2 library.

One criterion is that the environment of this test should be as faithful as possible. As test scenario, I choose the environment of a shopping mall. Therefore, the Münster Arkaden are used to collect important data for the test. This includes information about the lightning, the possible distances, and the sizes of the shop logos. With this information, a test environment is created where the different types of algorithms have to prove their reliability.

The usage of a test environment is the best way to evaluate the general understanding of the performance of each detection technique without unforeseeable influences of the environment. This way, all the unintentional objects on real pictures like persons or other shop logos that could lead to wrong detections are excluded.

The aim of this test is to generate knowledge about the behavior of the image detection algorithms on mobile devices. For mobile devices it is essential to know exactly how the performance and the processing power can be used in the most appropriate way. With the information of this test, it is possible to figure out how an Augmented Reality with image detection could be designed. This implies the most applicable algorithm with the best setting for the detection and the calculation speed.

4.3 Real Environment

To prove the results found within the test environment, I choose to execute the same test under real conditions. Therefore, five sets of images with real shop logos are analyzed with the same desktop application. The sets of images again include the four different angles, the variation of distances and the scaling.

Under real conditions also means that the shop logos have different sizes and are bigger than DIN A4 pages. To equalize the conditions for all of the five shop logos, it is important to convert the distances in relation to the size of the logo that should be recognized. This is also important for the comparison with the test environment where all of the images are represented by A4 size paper. By varying the distance, it is possible to ensure the same amount of pixel in the images. This is required in order to be able to compare the results of the test and the real environment.

The test is based on the same technique as the previously tested unreal environment. It aims to analyze the detection behavior of the different detection algorithms under the conditions of varying angles, distances, and scales. Furthermore, the real environment test can help to understand how the algorithms work when unforeseeable influences are included in the image. The data include images of a real environment with participants, different lightning, as well as other uncertainties.

With this approach, the knowledge gained by executing the application in the test environment should be evaluated. This further investigation helps to identify the most appropriated algorithm in accuracy and detection speed.

4.4 Processing Speed

In this chapter, I present an application that investigates different approaches to increase the recognitions per second. For a good usability it is necessary to provide a smooth and stable indoor Augmented Reality application. This is only possible with an acceptable latency and calculation time.

The knowledge gained in the previous tests also includes information about the scaling of images. The linear behavior of the pixel amount and time reduction also allows thinking about other alternatives to reduce the amount of pixel without reducing the image quality. A promising approach is to crop the image. This way, the important parts of the image can be used in good quality and the amount of pixel is still reduced. This kind of acceleration is already practically used by existing computer vision applications. One established application is the ZXing barcode scanner [33]. It applies the crop to focus on the important parts of the image. For the use of a camera application it is a common behavior to focus on the object that should be recognized in the middle of the screen.

Also the usage of multithreading is part of this section. It is possible to increase the frame rate, which allows providing a smoother application without reducing the amount of pixel. Many new devices have the option to use more than one core for a calculation. That is why testing the behavior of a multithreading image recognition application is a necessary part of this investigation.

This part of my master thesis is implemented on a mobile device. The reason for this is the direct association between the calculation latency and the processing power of the device. A desktop application, as used before, would lead to different and much faster results than a mobile device could provide.

The information gained in this test are important for the evaluation if and how a mobile Augmented Reality can be used and how the frame rate can be increased.

4.5 User Test

In the previous chapters, a test environment with a fix amount of recorded images is tested for the detection performance of the different detection techniques. The same test application and procedure is used in a test with images of real environments. With the help of this information it is possible to identify SIFT as the most suitable algorithm for shop logo recognition on mobile devices. Furthermore, techniques are identified for how the frame rate can be increased.

In addition, three methods are presented to improve the acceleration of the detection process. The first one is the implementation of multithreading. This process is independent from the quality of the detection and therefore has no drawback in accuracy.

The second one is the scaling of images. This method gives the possibility to reduce the total amount of pixel, influencing thereby the calculation time, but it has the disadvantage that the quality of the image is reduced. Pictures with small logos or with large distances could lose needed quality.

The last procedure for time reduction is cropping. This method reduces the amount of pixel like the scaling. However, also here one main drawback be evaluated. The crop also means that not the full image is used for the recognition process. The side edges are not included in the recognition. This also means that important information cannot be used for the recognition. One simple example is a huge logo and a user standing right in front of it. It might occur that the needed edges are excluded from the recognition.

In this chapter, the information gained in the previously presented test scenarios are tested under real conditions. Real users have to evaluate the current findings. Therefore, an implementation on a mobile device is provided. This is necessary to give a real impression of how the detection latency and the scaling or cropping can influence the usability of the application.

Furthermore, the test has the aim to obtain information about the user handling for scaling and cropping. Until now, it is not known if the users prefer to scale images or to accept that the image is cropped.

With this test the preferences for the calculation acceleration are explored and the usability of the findings like the selected image detection algorithm and the frames per second increasing processes are proven.

Based on the knowledge of this test, a prototypical application is implemented. The main aim to clarify if and how a mobile indoor Augmented Reality can be implemented on a mobile device is also defined.

5. Implementation

5.1 First Application

In the beginning of the project I implement an own application that allows the user to take a picture that is stored in the application. Later on, the user has the possibility to use the camera preview to search for the taken image. Therefore, the user can select one of the previously presented algorithms. When the user chooses a selection, the previously taken image becomes analyzed and the results are temporarily stored. Afterwards, the selected algorithm also analyzes the currently recorded video frame. The results are two sets of so called feature points that are extracted out of every image. When this is done, the OpenCV functionality is used to compare the two datasets in order to identify feature points that seem to belong to each other. This way, a list of possible matches is gained. The information can be used to match the homography of the findings. In this step, the list of matches is ordered and the matches are evaluated by their location in the image. When there seems to be systematics in the order of detections, the original image is interpreted into the live camera image. Therefore, also rotation and scale are considered. There is no multithreading included. One image is compared at a time. This leads to high delays depending on the calculation speed of the device and the complexity of the algorithm.

The user has the possibility to select one of the previously presented algorithms. This helps to understand the differences and the process of recognition. Furthermore, the application gives the possibility to change the resolution of the camera image. A hypothesis is that reducing the amount of pixel can save calculation time. This can be necessary to speed up the recognition and make it possible to use the application in real time. With this setting, the user can test the difference between detection rate, calculation time and image resolution.

The first implementation also has the aim to give an overview about the relation of device speed and algorithm effort. This way it is possible to estimate and schedule the further work.

5.2 Test Environment

In the test environment all of the different techniques to recognize images have to prove their quality in detection and speed. For this I use a typical camera of a mobile device. The Autofocus and an image resolution of 720p (1280 x 720 pixel) are used. The images that should be recognized are collected in predefined distances and angles. From every position, two images of the shop logo are used for the detection. This way, five datasets with different shop logos are collected. These are then used to test the different recognition techniques.

For the illumination of the test environment, I measure the light intensity at the shop logos in the Münster Arkaden. The result is that almost all logos are in a bright environment. The average of the measurements results in a lux value of nearly 500. This value is also used in the following test. The aim of creating similar illumination conditions is to get test results that are as reliable as possible. Therefore, it is important to emulate real conditions, as far as this is possible.

For the distance, the Münster Arkaden also provide the data that are used in the test environment. The farthest distance a person can have between the logo and the wall on the opposite site of the shopping center when standing in front of the logo is used. With this technique, it turns out that a distance of ~10 meters is reasonable in a shopping mall like the Münster Arkaden. Of course, it can be possible to reach longer distances between a shop logo and a user by varying the angle. This effect is neglected because of the endless distance that could be reached by varying the angle. Also, the usual position for logo detection is in front of a logo and not far away with a high angle.

With this information, it is possible to calculate the distances where the pictures must be collected in the test environment. One important fact is that the distance does not behave linearly to the pixel size of the logo in the taken image. If this is taken into account, it is possible to create reliable data sets with all kind of image sizes. The solution is that the height, as well as the width shrinks by moving further away from the object. This means that the pixel inside of the image are reduced exponentially to the distance. To counter this effect, it is important to use the square root from the area of the shop logos. With this technique, it is possible to calculate the real relation of two shop logos.

For the test I use the maximum range of ten meters. I reduce the distance in steps of 2.5 meters. At the distance of 2.5 meters, the shop logo almost fills the whole image. This is the closest value for the test. The staggering leads to four different distances that are used with every test setting.

In the experiment, the angle is also decisive. The detection highly depends on the resolution quality, but also the viewing angle of the object that should be recognized. The suggestion is that the algorithms can handle shop logos without a spatial transformation easier than shifted or distorted ones. To test this hypothesis and to clarify which technique for image recognition can handle angles in the most accurate way, different angles are used for the image recognition. To avoid creating a too large amount of test data and waste time during the preparation, I choose to vary the angle in steps of 22.5 degrees. This way, there are four different angles for every distance.

To explore the usual bounds of a common shop logo I collect the sizes of all businesses in the shopping mal. At all, I take the average of 36 different shop logos for the following tests. Here, it is important to say that not the length of the sides is important for the image recognition, it is the amount of pixel that represent the logo in the picture. This means that the area is more important than the shape of the object that is supposed to be recognized.

Another important influence for the accuracy and the speed of the image detection is the size of the images in which an object should be detected. For the experiment I use a camera in a mobile device that supports the resolution on 720p, this corresponds to a resolution of 1280 x 720 pixel. I choose this setting because of the fact that most mobile devices support this kind of representation for their video functionalities. To vary the resolution and to observe the associated effects, the images is scaled. Therefore, the image size is reduced to 75, 50, and 25 percent of the original edge size. This is done with all images.

For the first experiment in a test environment I choose to use shop logos printed on a DIN A4 page. These logos are well known and can also be found in common shopping malls. As mentioned before, it is important to adjust the position on the photo in relation to the logo size. In this case, a distance of round about 60 centimeters for DIN A 4 sized logos represents a real world distance of 2.5 meters. The background of the standardized test is a white wall. The aim of this is to avoid incalculable influences from other objects. I choose to use five different logos for this test. The logos are normal brands emblems that can also be found on real shops. The brands are Apple, Google Android, Coca Cola, Heide Park, and dm.

To get an improved accuracy and to reduce side effects, two images of every combination are used. This leads to an amount of 640 images for this test. This high number of images is caused by the many possible variables like distance, angle, scaling, and the five different logos.

All of these are processed with the six detection techniques. To run this test efficiently and in an acceptable time interval, I choose to implement the application on a normal desktop computer. The first implementation, described in the previous chapter, gives the suggestion that such a high number of cost intensive calculations would overtax the processing power and the memory of mobile devices. Furthermore, it is possible to transfer the results in relation to each other on mobile devices. This means that an algorithm that is twice as fast as another behave in the same way on a mobile device. The current market of smartphones also provides such a large range of different devices with different abilities in speed and memory that running this test on a mobile device would give only a limited impression of other mobile phones.

The images of this test have to be named in a way that further investigations can work with. That is the why a clearly identifiable namespace is used. In this, the name of the image that is shown, as well as information about the angle, the distance, and the scale is included.

The pictures recorded in this way are now ready for further preparation and image detection. Therefore, I implement the previously described image detection techniques with the help of the OpenCV 2 library. The application is implemented in C++.

Figure 1 represents the working process of the matching application implemented on a desktop computer. The first thing done by the application is to load the image that should be detected. For this image, the feature points of the first detection technique is detected and extracted. To compare it later on to the images recorded in the test environment, the information is stored globally. When the source image has finished the detection, the application starts to load the images in which it should be recognized. Therefore, the application navigates to the folder where the pictures are placed. For every image, the information stored in the namespace is used to identify the image that is processed. Following, the feature points of this image are detected and extracted, just like in the source image that should be recognized. When this is done, the feature points found in the source image is compared to the set of feature points found in the destination image. Every comparison leads to a set of feature points that could be equal. This set is ordered by accuracy. The OpenCV library gives the settings used for this. Now, the best matches are analyzed for their homography to each other, just like explained in the chapter of the first application implementation. When this is done, the part of “recognition finished” is reached in Figure 1. This also includes the storage of important information. This means that every recognition process stores the data of angle, distance, and image type. Furthermore, the application collects information on calculation time and success.

The next part is the scaling. Therefore, the application checks whether a scaling is needed or not. The original image is scaled three times in steps of 25 percent. After each scaling, the recognition process starts again.

If no further scaling is needed, the next image is loaded and the whole process, including scaling and recognition, starts again. This is done until all images of the test environment are calculated with the current detection technique.

When no more images are available, the whole process loops with another detection method. This way, all of the techniques used in this test are processed. The new recognition also includes that the feature points of the source image are updated by the new recognition technique.

When this is done and all of the detections are finished, the complete recognition stops. Now, the information of all recognitions is exported into a CSV file. This is the easiest way to make the information accessible for further applications.

The working sequence in Figure 1 represents only one single detection process. This whole process reiterates for all of the available source images and their supplied images.

During the preparation of this application I realize that there is much wrong detection. The result that is delivered consists of four points that build a rectangle around the detected object. There are some use cases where the detection supplies bad results. It is possible that the lines are overlapping or that the detected area has the shape of a very flat trapeze. This detection normally can be dismissed as wrong detection. To filter such results, the application has to test the rectangle for its consistency. Therefore, I prepare three tests that have to be executed before the recognition can be defined as true.

At first, the line intersection is tested. When a line is overlapping another the test result is evaluated as false.

The same holds true for the size of the rectangle. If the rectangle is smaller than 10 percent of the image size, the recognition is discarded. This prevents from too small detections. The recognition of an image smaller than 10 percent of the original image is quite unlikely because of the small amount of pixel included in this area. For a resolution higher than 720p, this value must be redefined.

The last review of the recognition inspects the angles of the rectangle. In the detection, the angle can be so small that a real recognition seems to be unrealistic. To avoid such wrong detections, I implement a scan for the angles. If an angle is smaller than the capacity threshold of 35 degrees, the recognition is discarded.

With these three techniques, the amount of false recognitions should be reduced significantly. To prove this hypothesis, I implement a method to syndicate the recognition manually and automatically. The application has the option to enable a manual detection. This means that the user sees the detections visualized in the image where the object should be recognized. In this view it is possible to identify whether the detection is successful or not. The result is a list of recognitions with the information if the manual and the automatical match in the findings. I process all 640 images with this mechanism. The result is very promising. With this methods and their settings it is possible to filter all the wrong detections represented by deformed shapes.

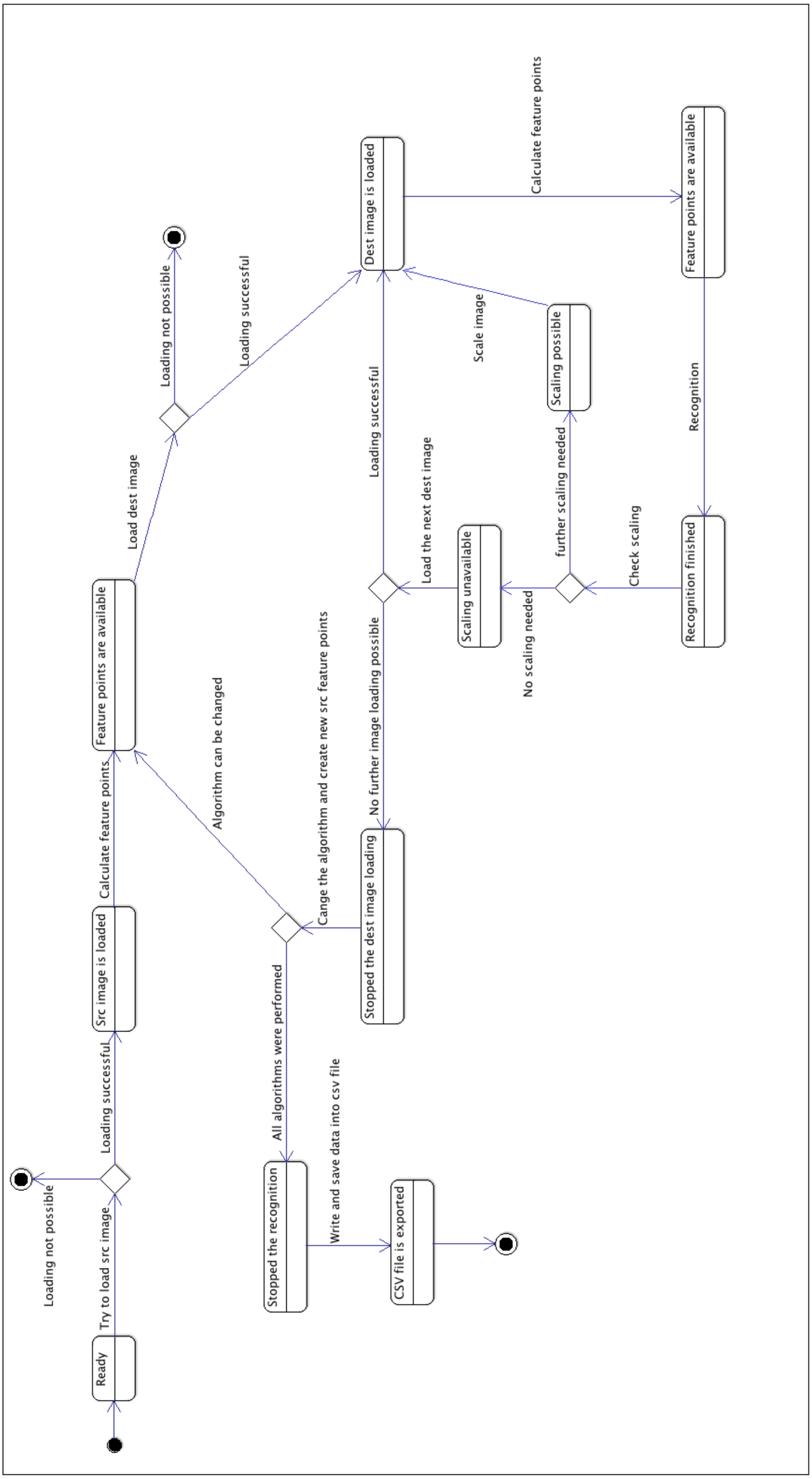

Figure 1 - Activity diagram of the image recognition desktop application's life cycle to test the different algorithms in their scale, distance, and angle behavior. – Src image is the image that should be recognized; Dest image is the camera recording the src image should be recognized in.

The automatic processing of the 640 images with the six different detection techniques takes more than 23 minutes on a MacBook Pro with a 2.4 GHz i5 dual-core processor. This proves the necessity of implementing the application on a desktop computer and not on a mobile device.

To evaluate the CSV files and the included information, I choose to use Java and the apache commons math library [34]. With its help, it is possible to standardize and automate the statistical functionalities. For the parsing of the CSV files I use OpenCSV [35]. All of the values are stored in a custom object. These values are sorted with respect to their scale, resolution, and the distance.

5.3 Real Environment

The implementation and execution of the real environment test includes nearly the same procedure the test environment uses. The main difference is the data set of used images. For the previous experiment, image sets without uncertain influences and with objects standardized in their size and their construction are used.

For this test, I also record five data sets of images. The angles, distances, and scales are managed in the same way as in the test environment. The distance is also adjusted equally. The area of pixel in the recorded image has the same amount as in the pictures of the previous test. This is needed to guarantee the same experiment conditions. Besides, the recorded images display brands in everyday situations. The images include different lightning situations, pedestrians and other influences in a real environment.

The same application is used for the analysis of the images and the recognition process. It is not necessary to implement any new features for this test. Only the folder structure has to be adjusted.

The evaluation of the results is also done in the same way. Therefore, again the commons math library is used to prove assumptions and to generate certain statistics.

For further information about the application construction read the previous chapter explaining how the test environment application is implemented.

5.4 Processing Speed

In this chapter, I implement an iPhone application that is able to log the calculated frames per second. It is used to test different approaches to find out how a more effective Augmented Reality application could be designed.

The detection algorithm SIFT is used to detect predefined shop logos with the camera of the mobile device. The implementation is simultaneous to the one described in the chapter first application. The main difference is that the calculation time and the frames per second for every detection are recorded and stored on the device.

The application includes two different techniques to accelerate the detection speed. The first one is the cropping of the recorded video frame. Therefore, the edges of the images are cropped. This is done with an increasing area for pixel reduction. The measurement starts with no cropping and increases by 12.5 percent of the side length until only 25 percent of the original side length is left.

The same calculation is executed with an image scaling. The reduction of the image resolution is also staggered like the cropping variant. This means that for both methods, the amount of pixel in the image is exactly the same.

Furthermore, the cost intensive feature point detection method and the feature point extraction method are outsourced into own threads. The information about how the multithreading influences the calculation latency and speed is recorded.

5.5 User Test

In this experiment, the users evaluate their favorite calculation time reduction method for the SIFT algorithm. Therefore, I prepare a mobile application that can dynamically change the crop and the scale factor. The user has the possibility to choose between three different settings for the recognition. All of the detection methods have exactly the same amount of pixel. The main difference is that the reduction of pixel is reached with different combinations of scaling or cropping the image. With this setting, the calculation time does not influence the decisions of the participants. The same amount of pixel automatically means the same gain of time. This way, the user can focus on the detection behavior of the different scale and crop settings.

As option for the different settings I choose to use the amount of pixel with a scales side length of 50 percent. This setting represents the level of scale that still provides applicable accuracy for the detection. This is supplemented by a 100 percent cropped option. 25 percent crop on every side leads to the same amount of pixel and therefore the same calculation time. The third option in this test is a combination of both techniques. Therefore, the image scale is reduced by 25 and the crop by 12.5 percent on every edge. These three settings use the SIFT recognition method to detect the shop logos.

The application designed for this test enables the user to change the crop and scale combination easily. The level of cropping is visualized by a half transparent rectangle. The free area in this rectangle represents the information that is used for the recognition process. The same behavior is also used for the ZXing barcode scanner [33].

When a correct recognition is found, the result is presented to the participant by visualizing the approximated rectangle in the screen and highlighting the crop border in green. When there is no recognition, the border changes the color to its default again. This way, the user can evaluate the amount of correct detections.

The camera of the application automatically uses the Autofocus to optimize the results. The user can see the current camera image in maximal resolution on the mobile screen. This way it is possible to give the user a better impression and feeling than by only showing the low scaled or cropped image. Additionally, the real detection rate is logged on the mobile device.

The test is executed with ten participants. All of them are familiar with the usage of mobile devices and the usage of AR applications. Their task is to detect a shop logo from different distances and angles. The angles and distances are structured similarly to the previously described tests. Angles of 22.5 degree and distances of 2.5 meters are combined to an amount of 16 combinations.

The participants have to answer questions about the three types of detection (Questionnaire in the appendix). Therefore, the participants have to grade all of the three detection combinations for each distance and angle combination. Furthermore, the real detection results are stored to compare the subjective user impressions and the real data sets. This information is not visualized. This way, it cannot influence the participant´s decision.

Additionally, the whole scale and crop process should be evaluated with a grade from one to six. Furthermore, they have the possibility to specify their grade. The test duration for each participant is round about 45 minutes.

5.6 Prototype

The previous experiments lead to some techniques and methods that should be preferred for an implementation on mobile devices. To get an impression of how a completed implementation could look like and to test the gained knowledge, I choose to implement a prototypical application with the new knowledge in accuracy and time reduction. As platform for the prototypical application I choose iOS 5. The test device is an iPhone 5. Therefore, the algorithm with the most appropriated accuracy and calculation time combination identified in the test and in the real environment is used. This leads to a robust detection with a maximum level of accuracy. The implementation is similar to the user test application.

Another important factor is the needed time for the recognition process. Only with a solution that provides enough frames per second, a real usage is possible. In the previous chapter I present different techniques for how such a time reduction can be realized.

One possible solution is the usage of more than one thread. This leads to an increase of the frames per second by the factor two. Exactly this method is also used in the prototype. It helps to calculate twice as much images in the same time. This also means that wrong detections does not have such a huge impact, because of the increased amount of calculations. Increasing the frames per second with multithreading does not automatically mean that the calculation effort and thereby the calculation time is reduced. The latency stays the same with the implementation I present. That is the reason why other methods and techniques are also considered.

I implement a dynamically changeable relation of the scale and crop combination. This way, the user can influence the standards for the recognition process. An adaption for this personal needed situation becomes possible without increasing the amount of pixel in the image. This means that the time reduction can also be guaranteed in every situation.

With all these techniques and settings, it is possible to create an application able to provide one frame per second. This still seems to be not enough for a real time Augmented Reality application. The problem to guarantee an accurate detection and a useable calculation time is still valid for current mobile devices.

6. Results

6.1 First Application

This implementation only suggests a subjective impression of how the different detection techniques work.

It seems like the detection highly depends on the resolution of the camera. The first impression is that with a larger amount of pixel, the detection is more accurate and faster. The calculation time is also influenced by the resolution.

Furthermore, I test the detection out of different perspectives to the test object. This gives the impression that the detection also depends on the distance and the angle to the object. It is also interesting that the algorithms seem to react differently on the detection scenarios. The application furthermore creates the impression that there are some algorithms that provide a high frame rate, which means a good calculation speed, as well, and others have a better accuracy in detection. A good example for the accuracy is SIFT and for the speed, FAST. During the detection, I analyze the process of processing. The result is that in this application, the most effort for the calculation is needed to detect the key points in the image. The matching only takes a smaller amount of time.

To test this hypothesis, I run different tests that are explained in the next chapter. There, I use standardized and more objective criteria to investigate this assumption. Furthermore, the app is build to present the idea of my work in the master thesis defense.

6.2 Test Environment

In this section, the results of the test environment are presented. The first result that is visualized in Figure 2 shows the relation of angle and detection success for the different algorithms. All of the previously presented detection methods are also included in this diagram. Additionally, an average line is added to visualize the prominent techniques. The left axis represents the percentage of detection success. The detection results can be ordered into three different groups. In the first one are the FAST and the GOOD detection methods. Figure 2 shows their low success rate. In none of the angles tested they are able to reach a value above five percent.

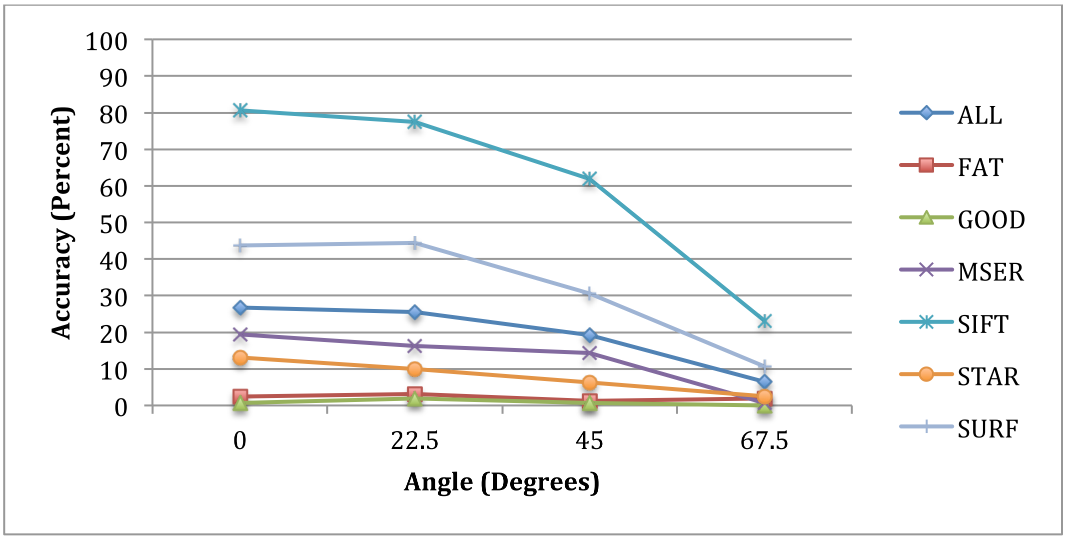

Figure 2 - The relation of different angles and detection accuracy for the tested image recognition algorithms in the test environment

STAR and MSER represent the second group. They provide better results than the algorithms in the first group, but they still perform below average.

Two detection techniques are able to exceed the average of all detection methods. SIFT and SURF provide by far the best results in detection accuracy, with SIFT significantly overpowering SURF. Especially the first three angles show the huge range of their accuracy. The difference for the first angle is nearly 40 percent.

This shows that in the test environment, SIFT can handle the variations of different angles better than any other detection technique. Also the second best algorithm, SURF has huge problems to reach the accuracy of SIFT (Figure 2).

The most conspicuous fact is that the detection of nearly all detection techniques highly decreases the accuracy with the maximum angle. During the first three angles the success rate slowly diminishes in comparison to the last reduction. This means that the last angle worsens massively the results of all detection techniques. That is the reason why I compile one statistics including and one excluding the maximum angle for further tests. This should help to get a closer view on the real detection abilities of each detection technique.

Figure 3 represents the relation of distance and success. The distances of 2.5, 5, 7.5, and 10 meters represent the converted distances in the test environment. With the combination of distance and detection success it is possible to draw conclusions about the accuracy of every detection method for each distance. Most of the visualized lines seem to behave linear. This means that they reduce their accuracy by increasing distance.

The distribution of detection results is assimilable to the results of Figure 2. The order of the detection methods is similar and the three groups can also be found in this statistic. While FAST and GOOD provide the worst detection results and STAR and MSER are again below the average, SURF and SIFT overtop all other results. Thereby, there is still a huge range between the accuracy of SIFT and SURF.

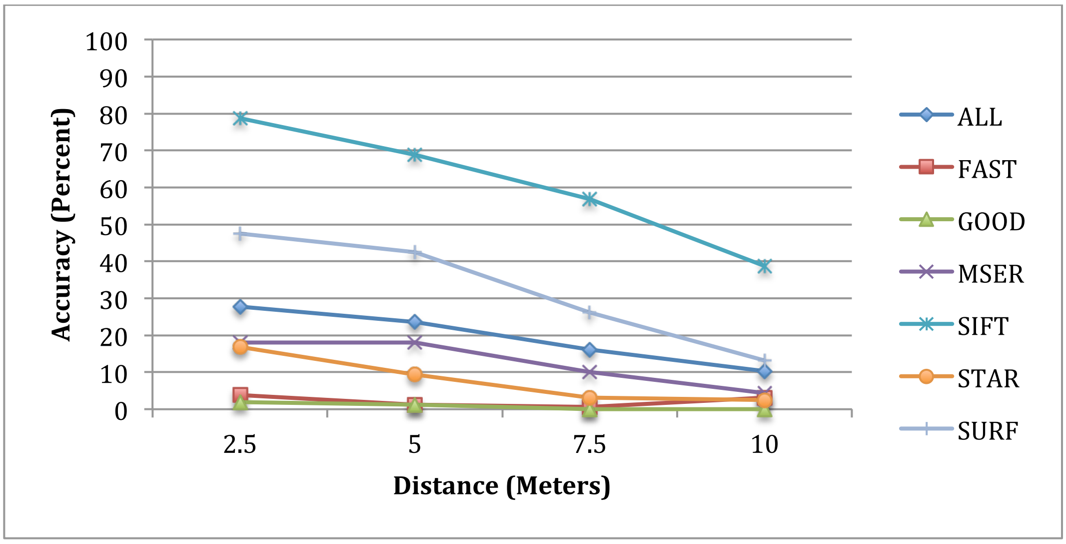

Figure 3 - The relation of different distances and detection accuracy for the tested image recognition algorithms in the test environment

In Figure 2 I assume that the angle of 67.5° delivers worse detection results for all the available detection methods. Therefore, I prepare this statistic without the largest angle (Figure 4). It shows that this assumption is correct. The measured values without the angle of 67° provide better results for all detection techniques. The behavior of the curves seem quiet similar in both diagrams.

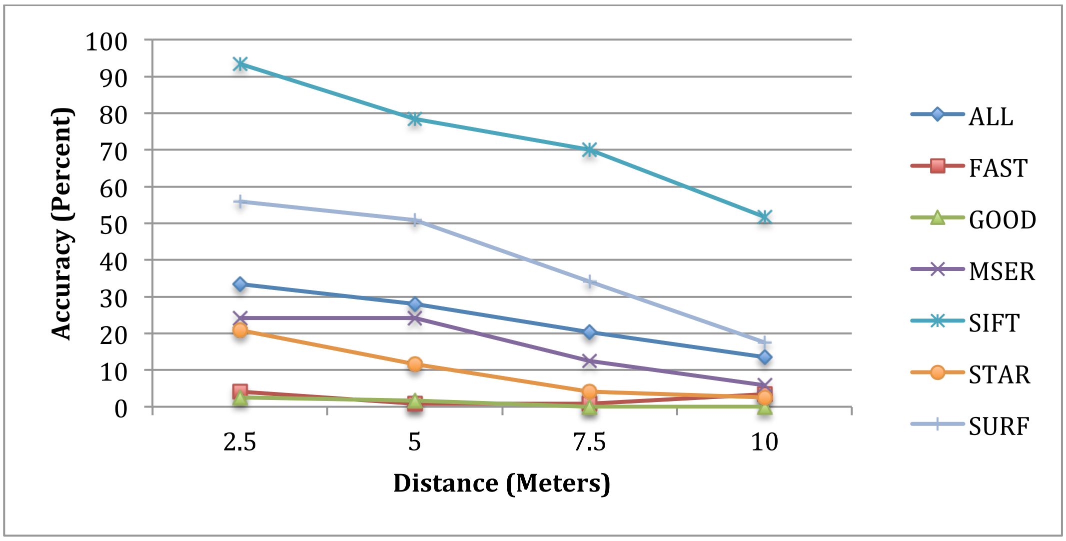

Figure 4 - The relation of different distances and detection accuracy for the tested image recognition algorithms in the test environment without the angle of 67.5 °

The main difference is that their accuracy is improved. SIFT can increase its accuracy by more than 10 percent. Similar progression can be observed in the SURF curve. Here, the values are increased by more than five percent. These two diagrams affirm the cognitions of Figure 2. SIFT clearly dominates in all of the tested conditions. Also, the t-test shows that SIFT is towering SURF with $p < 0.05$ with and without the biggest angle of 67.5°.

The next statistical preparation is the relation of success and scale (Figure 5). This diagram gives nearly the same impression as the previous statistics. There are three groups in the detection. The first one is again FAST and GOOD. Both present yet again a very low detection quality, similar to Figure 2.

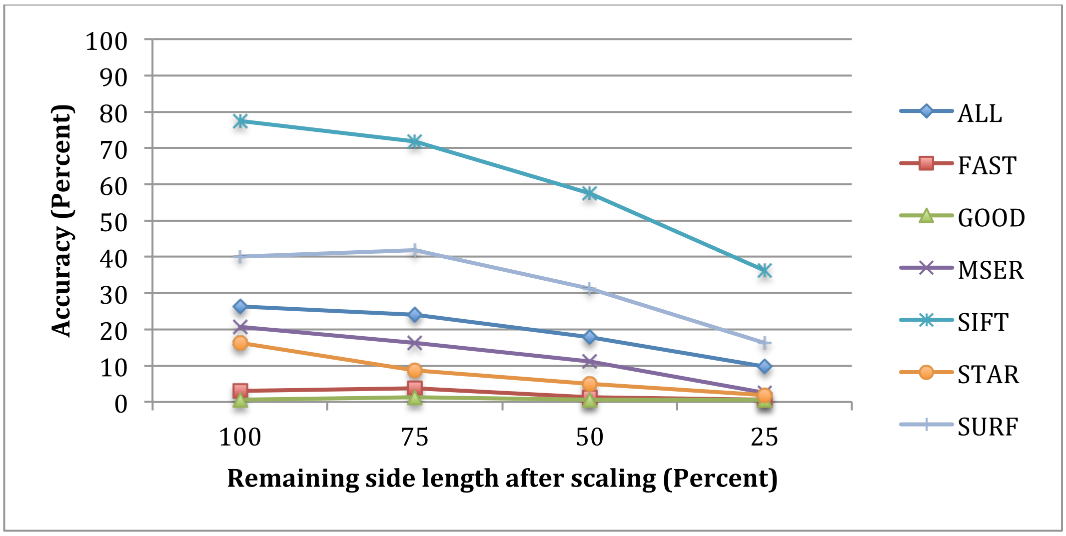

Figure 5 - The relation of scaling and detection accuracy for the tested image recognition algorithms in the test environment

Also, the second group is not able to bear down the average of successful detections. Whereby MSER provides better detection results than STAR for the three first distances. The difference for the distance of ten meters in detection only has a deviation of one percent.

The most interesting detection results can again be found for the SIFT and the SURF detection methods. SIFT again dominates SURF. The correct detections are shrinking the more the scale is reduced. The reduction of the first scaling of SIFT is only from 77 to 71 percent. The scaling of 50 and 75 percent worsens the results dramatically. This can also be observed in Figure 6, where the angle of 67.5° is excluded. The removal of the biggest angle again has the main effect that all of the detection techniques provide better results for all use cases. For the relation of SIFT and SURF I also prepare a t-test. The results for both deviations is that p is smaller than 0.05. This again proves the assumption that SIFT is clearly the best detection method in relation to accuracy.

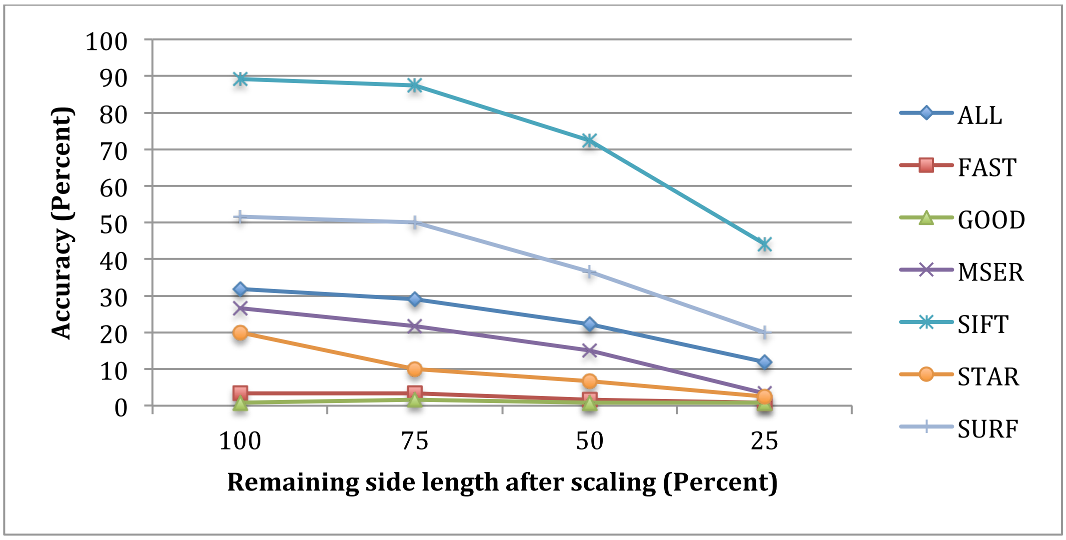

Figure 6 - The relation of scaling and detection accuracy for the tested image recognition algorithms in the test environment without the angle of 67.5 °

The most important aspect of the scale success relation is the time that can be saved by reducing the amount of pixel. Therefore, the average duration is listed in Figure 7.

This statistic represents the duration for each detection process. The SURF algorithm gives the results with the highest calculation time. With a start value of 0.8 seconds, this is nearly twice as much as SIFT with a start value of 438 milliseconds. The gain of time visualized in the curves of SIFT and SURF seem to behave similar. The save of time is the highest for the first scaling. In every further step, the time is reduced by nearly half of the first reduction. But this is not true for all the algorithms.

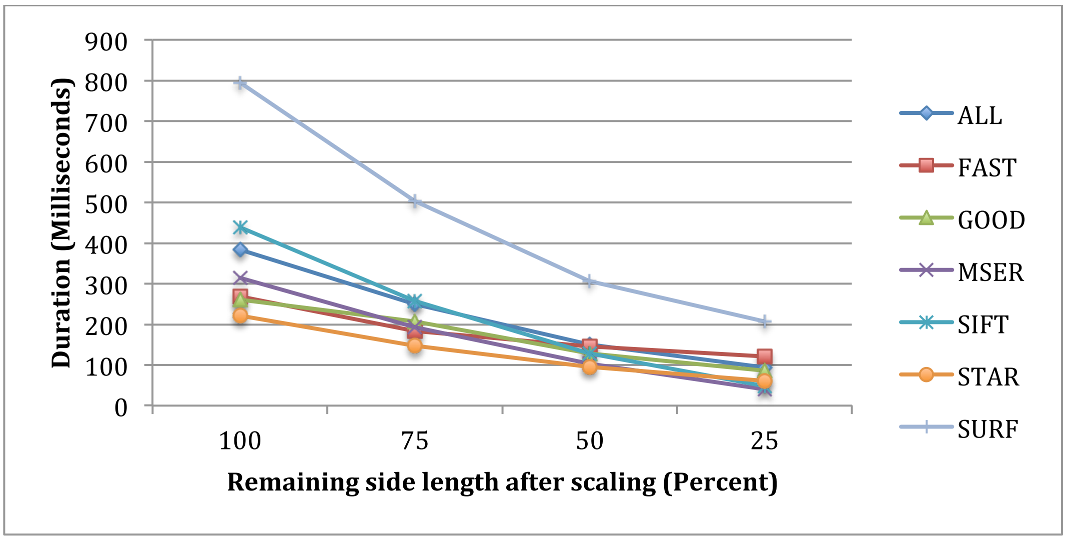

Figure 7 - The relation of scaling and duration for the tested image recognition algorithms in the test environment without the angle of 67.5 °

Table 1 gives a closer impression of the calculation times of the different techniques. One main difference is the start calculation time. There, FAST and GOOD provide the shortest calculation times for large amounts of pixel. In general, most of the detection methods are close by in their calculation time. MSER also delivers exiting results. It starts slower than FAST and GOOD, but is able to catch up on them. For the smallest scaling of 25 percent, it is the fastest detection method in this test. Another surprise is the STAR algorithm. Its starting calculation time is faster than every alternative. Even so, this benefit is lost during the scaling of the image.

Another interesting aspect is the standard deviation. Some detection techniques have a very high variation in their detection time (Table 1). SURF is the algorithm with the biggest variation. This applies to all scales. The most constant process is represented by SIFT. This shows that SIFT not only has the best results for accuracy, it is also very reliable in their needed calculation time.

To prove the impressions of this test, the test is also executed with real shop logos in a real environment.

Table 1 - Results of the relation of detection accuracy and scale for the different image recognition algorithms.

Algorithm	Side length (%)	Success (%)	Calculation time (milliseconds)	Standard deviation (milliseconds)
All	100	31.9	383	219.7
All	75	29	249	138
All	50	22.2	151	92.8
All	25	11.9	94	93.4
FAST	100	3.3	268	122.6
FAST	75	3.3	183	81.6
FAST	50	1.6	146	69.8
FAST	25	0.8	121	74.6
GOOD	100	0.8	260	28
GOOD	75	1.6	207	30.4
GOOD	50	0.8	129	34.8
GOOD	25	0.8	86	30.1
MSER	100	26.6	314	37.2
MSER	75	21.6	193	24.3
MSER	50	15	103	16
MSER	25	3.3	40	11.2
SIFT	100	89.1	438	28
SIFT	75	87.5	258	22.3
SIFT	50	72.5	128	15.8
SIFT	25	44.1	48	10.5
STAR	100	20	221	48.2
STAR	75	10	148	46.9
STAR	50	6.6	95	45.1
STAR	25	2.5	61	39.2
SURF	100	51.6	795	192.1
SURF	75	50	503	139.6
SURF	50	36.6	306	113.2
SURF	25	20	207	156.7

6.3 Real Environment

The observations of the test environment are completed with the impressions of the real environmental results. The findings of this test are presented in this section.

The first graphic represents the dependence of angle and detection accuracy (Figure 8). This diagram must be compared to Figure 2. The most notable fact is that all of the algorithms provide much worse results than in the test environment.

It is interesting that most of the order changed for the algorithms STAR, MSER, FAST, and GOOD. The important detection methods SIFT and SURF do not show a substantial changed behavior in the order. SIFT is still on top of all other detection methods.

Again, the biggest angle delivers very bad results for all the detection methods. This is visualized much more clearly in the real environment than in front of the white test wall. For that reason I also provide statistics without the angle of 67.5°.

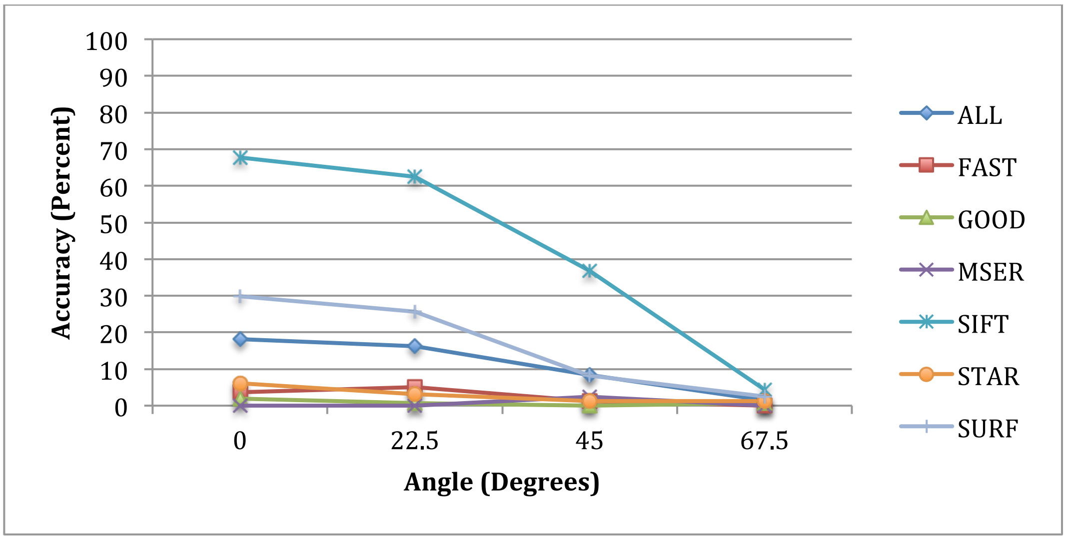

Figure 8 - The relation of different angles and detection accuracy for the tested image recognition algorithms in the real environment

The statistic of the distances and the accuracy provide unexpected results for the first distance (Figure 9). The closest distance has the worst recognition accuracy. Noticeable is that this is only true for SURF and SIFT. The other algorithms work as expected through the knowledge of the previous tests.

Even so, there are some unexpected results for the combination of distance and accuracy. The main assertion of the test is still that SIFT also dominates in detection success. The statistic with the biggest angle also provides the same results.

The main difference is that all algorithms can provide better results when the angle of 67.5° is excluded. I also prepare a t-test that proves the relation between SIFT and SURF. Again it is possible to show that SIFT provides significantly better results than SURF ($p < 0.05$).

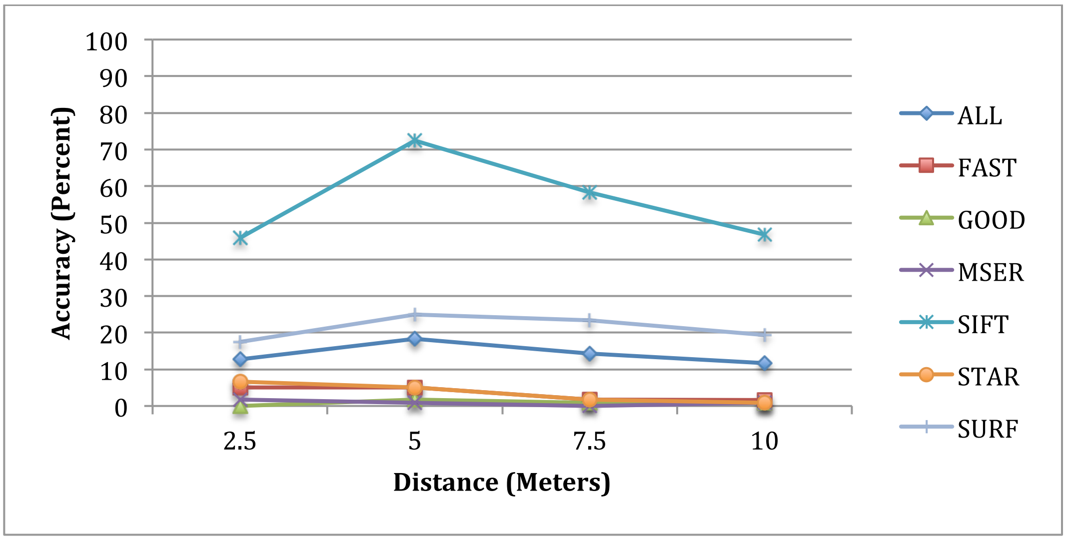

Figure 9 - The relation of different distances and detection accuracy for the tested image recognition algorithms in the real environment without the angle of 67.5°

The last statistic for the real environment is the relation of scale and success (Figure 10). Here, nearly the same situation as in the other diagrams is present. SIFT proves his dominance again with a t-test that affirms its transcendences in accuracy with a value of 95%.

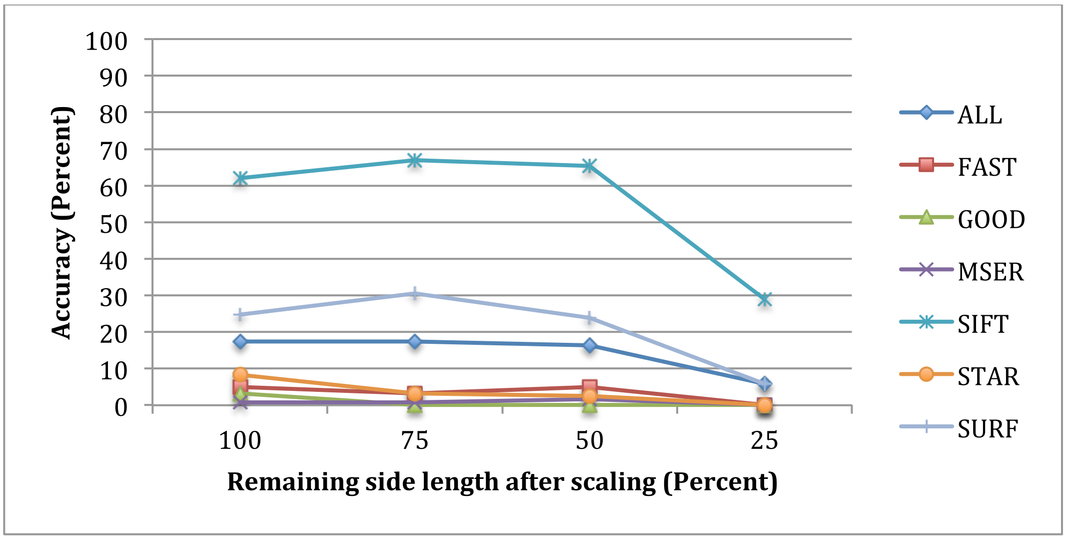

Figure 10 - The relation of scaling and detection accuracy for the tested image recognition algorithms in the real environment without the angle of 67.5 °

The more interesting aspect is the behavior of SIFT's accuracy for the scaled images. The first three scales are represented with nearly the same accuracy. The obverse in Figure 6 instead shows a continuously shrinking that is growing with the level of scale. This aspect is discussed in the chapter progressing speed.

To understand the behavior and the composition of the first three scales in Figure 10, the distance must also be taken into assessment. Therefore, Figure 11 presents the detection accuracy of SIFT with different distances and scales. It is interesting to note that only the maximum-scaled curve is falling out of the series. The other three curve progressions run similarly at the first glace. However, a closer consideration shows that low-scaled images have a higher accuracy for lower distances than high-scaled images. In the further course, the increased distance is negating the effect and the images with a higher quality lead to better results.

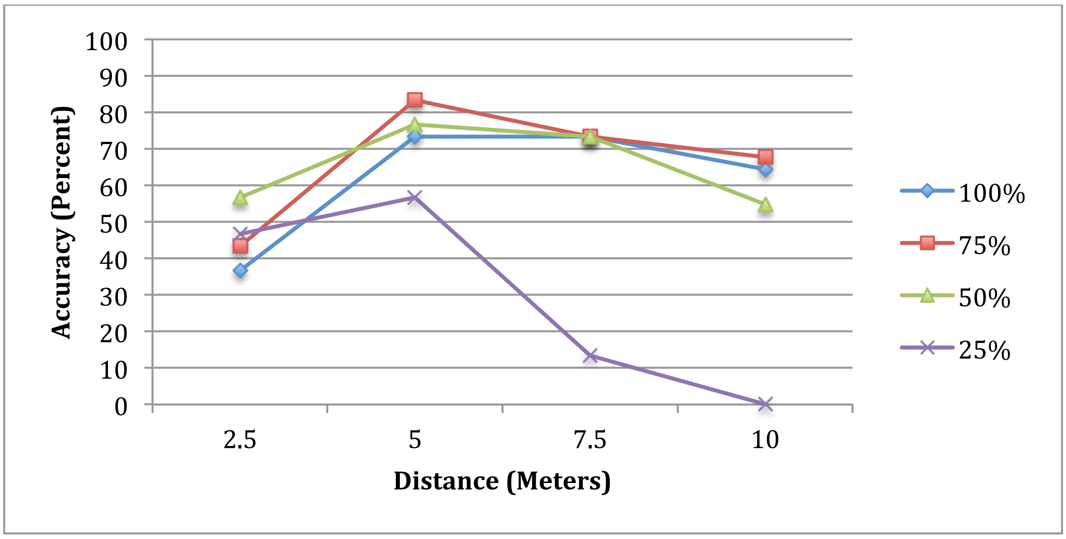

Figure 11 - The relation of scaling and detection accuracy for the SIFT algorithm in the real environment without the angle of 67.5 °

6.4 Processing Speed

The implementation on mobile devices requires a different approach than on a desktop computer. The limited resource of memory and calculation power enforces maximal focus on the efficiency of the application.

Therefore, the scaling of the camera image has proven a possible option for an acceleration of the calculation duration. Figures 6 and 7 show the accuracy and the duration of the detections methods. To easily compare these values I prepare a representation that includes accuracy and the calculation effort in one graphic.

Figure 12 and 13 show this relation. The time and the accuracy are represented in percent. 100 percent time is automatically the maximum calculation time that is achieved by the slowest scale adjustment.

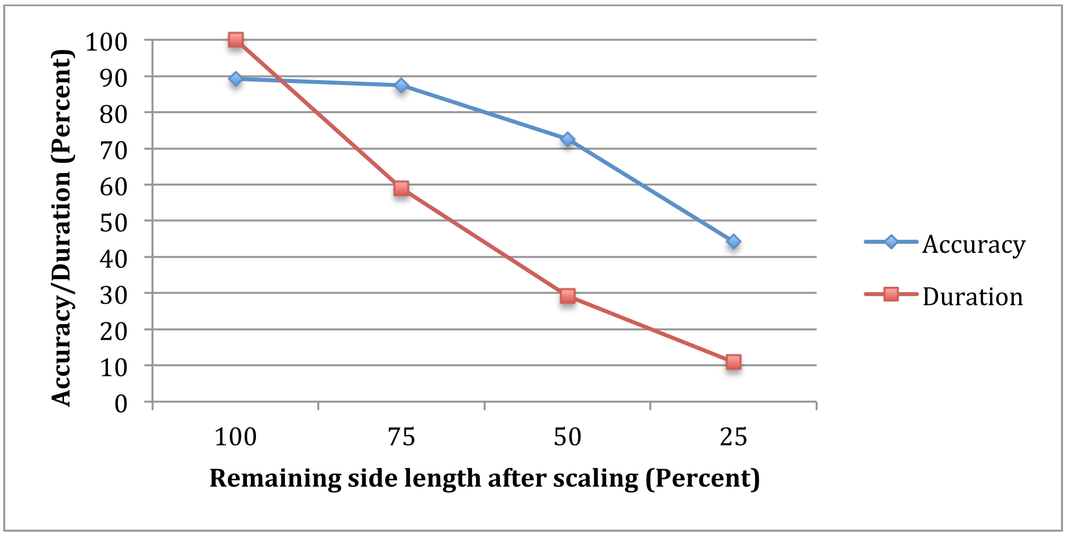

Figure 12 - The relation of accuracy/duration and scaling for the SIFT algorithm in the test environment without the angle of 67.5°.

The two curves show a varying course. While the accuracy drops slowly at the start and rapidly deteriorates in the end, the calculation speed curve reduces first clearly and is flattening for the further scaling. At first, it becomes obvious that the time reduction of Figure 13 behaves equally to the time reduction of Figure 12. This shows that the time reduction of SIFT is environment independent.

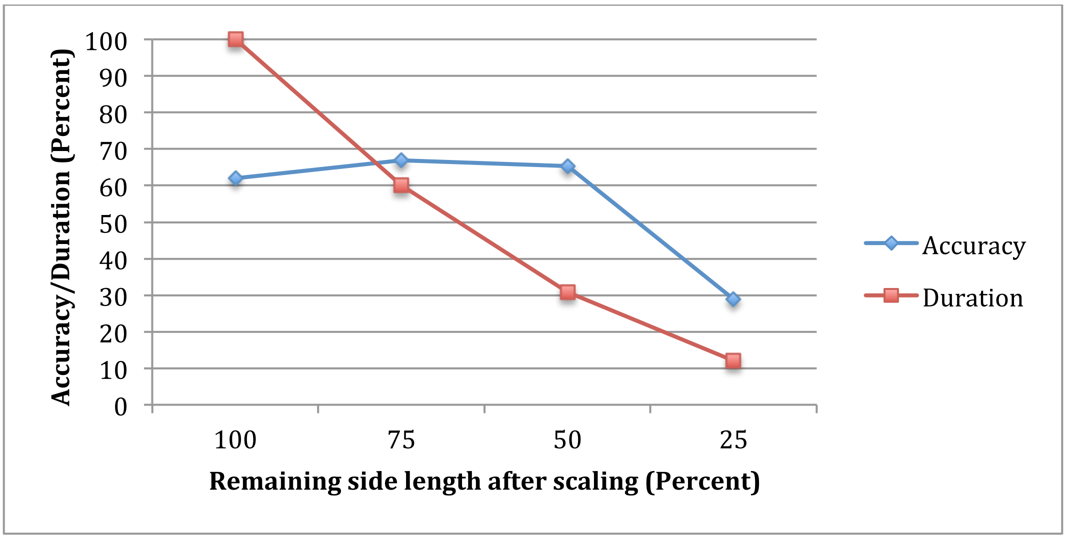

Figure 13 - The relation of accuracy/duration and scaling for the SIFT algorithm in the real environment without the angle of 67.5°.

A remarkable access is that for the first three scales, the level of accuracy seems to be consistent. The last value is again inapplicable for a good detection.

Figure 14 shows the needed calculation time for a complete image recognition with the SIFT algorithm on an iPhone 5. The reduction of pixel is done once by scaling and once by cropping. The x-axis represents the side length of the current image in relation to the original picture. At first glace, the accordance of the two curves is significant. The curves are nearly equal in their time reduction. Only two times the crop can provide a result that is 0.2 seconds faster than the scaling.

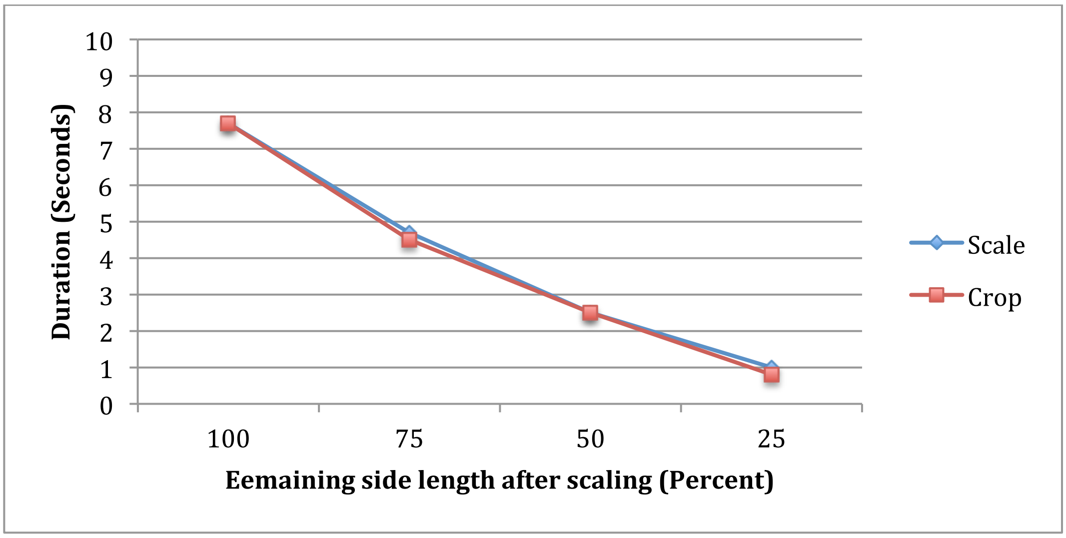

Figure 14 - The relation of calculation time and side length for scaling and cropping the recorded image with the SIFT algorithm without the angle of 67.5°

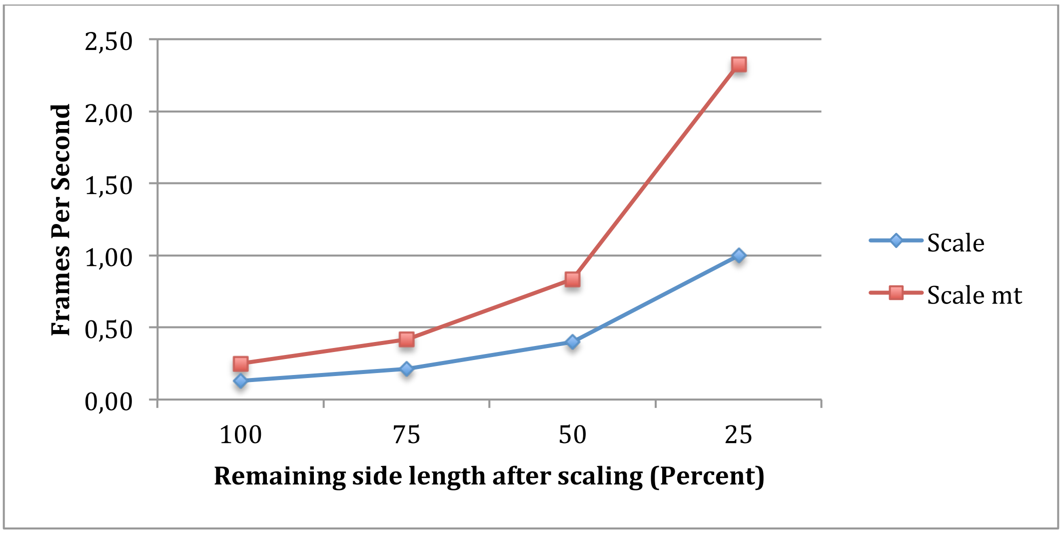

Figure 15 - The relation of frames per second and side length for the SIFT algorithm without the angle of 67.5° with and without multithreading (mt)

An additional concept for an increase of the frame rate is to use more than one processor core in the mobile device. Figure 15 shows how multithreading influences the frames per second for different scales. This figure can be summarized by the awareness that two simultaneously running processes can accelerate the frame rate by nearly twice.

6.5 User Test

The results of this test are presented in Figure 16 and 17. Figure 16 uses grades for the evaluation of the different scale and crop combinations. In this rating, the 5 represents the best possible result and 0 the worst.

For Figure 17, the subjective evaluation of Figure 16 is extended by reliable data of the detection process. This more detailed representation of the user test is listed in percent.

The three different types visualized in the graphics are:

Table 2 - Explanation of the different settings for the user test

Type	Scaling (%)	Cropping (%)
0	50	0
1	25	25
2	0	50

In this process, an interesting structure is revealed. The order of the data in the two figures is the same. It is conspicuous that the types that have the best detection for large distances have a lack of accuracy for close recordings.

In contrast, type 0, which has a huge scaling, can convince for short distances. The disadvantage of this combination of scale and crop is the rapidly shrinking detection rate with growing distances. Figure 17 shows that, at a distance of 7.5 meters, the pure scaling method is useless to recognize shop logos in the camera image of a mobile device. Exactly the same development of type 0 can be observed in the user-evaluated data. In the near field, the combination of a high-scaled and low-cropped image seems to be the best solution – Figure 16 and 17 confirm this assumption.

Type 2, representing the opposite of the previously discussed technique, has its disadvantages in the close areas. The crop of this method is with 50 percent of the side length the largest in this test. This also means that short distances users have the problem that they cannot completely capture the logo in a single image. This can also be observed in Figure 16 and 17. There it is striking that the measured percent of detection and also the subjective user grades are improving their values by increasing the distance. This is true for the second distance. The third and also the last distance of 10 meters reduce the measured level of accuracy.

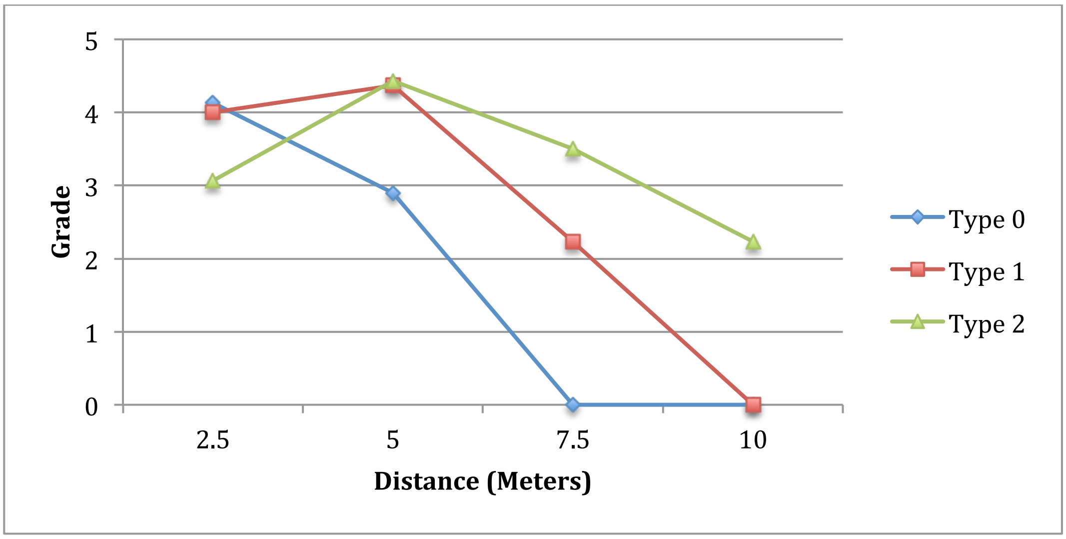

Figure 16 - Overview of the grades for the different detection techniques by the participants. Type 0 represents a scaling of 50% and no crop; Type 1 represents 25% scaling and 25% cropping; Type 2 no scaling and 50% cropping.

By a crop of 50%, the closest distance misses important details of the shop logo. This is not the case when the user has a higher distance to the target image. The recognition with a high crop and without scaling can provide the best results for large distances. This can also be observed in Figure 16 and 17. Type 2 has clearly the best results for large distances.

Type 1 represents a combination of cropping and scaling the recorded image. This way, the same gain in time the other two types reach can be provided. The data shows that a combination of these techniques also implicates the merge of the advantages and drawbacks of the two time reduction methods. This is particularly apparent for the measured accuracy in large and close distances. Type 1 is able to deliver nearly the same quality of detection for distances of 2.5 meters. For distances of 5 meters, the combination has the best result of all the tested scale crop combinations. However, this trend cannot be continued for larger distances. There, the maximum-cropped alternative still provides by far the best detection results. The grades of the users confirm the observations with their impressions of the detection. All in all, the combination of crop and scale seems like a good solution for close and medium distances.

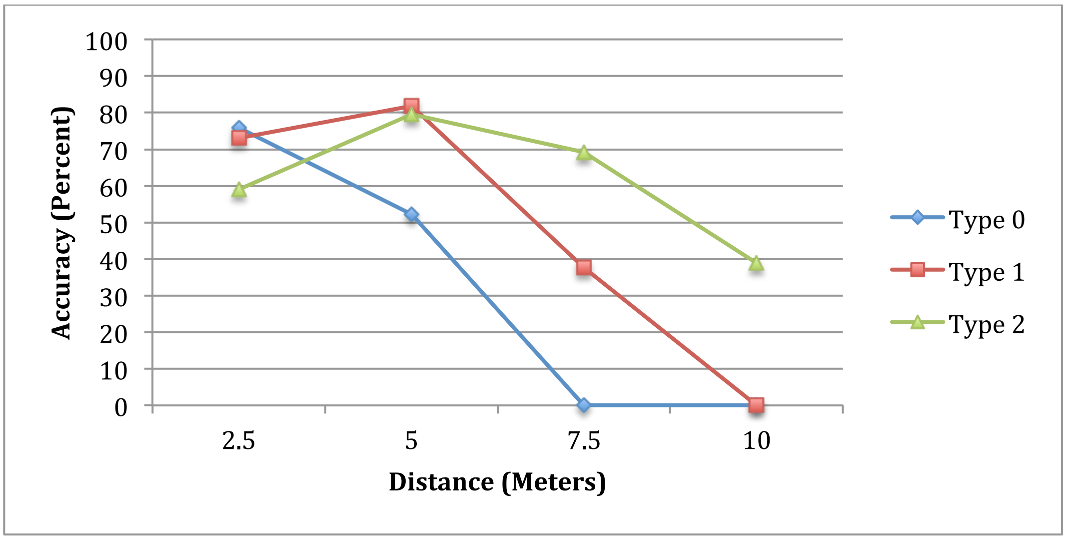

Figure 17 - Percentage overview of the measured accuracy during the user test for the different distances. Type 0 represents 50% scaling and no cropping; Type 1 represents 25% scaling and 25% cropping; Type 2 no scaling and 50% cropping.

Table 3 presents the grades of the users for the different combinations of scaling and cropping the recorded camera image. Most people prefer the solution without scaling the image. Their main argument is the better performance for large distances. This also explains the result of type 0. There, the crop is excluded and the scale is maximized. This leads to a higher detection for close distances, but has a lack of accuracy the higher the distances are. The combination of scale and crop provides very good results for close distances, but also has a problem to stick to the detection accuracy of a pure cropped image (Type 1). This can also be observed in the over all user grade.

Table 3 - Rating for the different settings in the user test by the participants. Grade from 0 to 5. 5 represents the best possible results.

Type	User grade	
	Average	Standard deviation
0	2.4	1.0
1	3.5	0.7
2	3.9	0.8

The results of Table 3 show that the users prefer the combination where all the distances have the chance to produce positive recognition. It also means that the participants have the opinion that recognition from a farther distance point is also important for an accurate recognition procedure.

One main use case expressed by the participants is to receive additional information about shops and their products. The second answer given by the participants is the usage in art exhibitions. There, it could be used to display additional information about the artwork or link to the website of the artist. Another suggestion by the participants is the use for the detection of traffic signs. This can help the driver to get a better interpretation of his surrounding. Missed traffic signs can be auto detected and visualized internally in the vehicle. The complete answers of the user experiment can be found in the appendix

7. Discussion

The experiments in this master thesis have the aim of identifying an image detection algorithm that has the characteristics to provide robust, stable, and fast image recognition.

The first application that is realized shows that there are certain techniques that have the possibility to detect object in pictures. However, it also points out that the existing methods have certain drawbacks. To get a better impression of those, a further experiment where the entire spectrum of OpenCV feature point detection algorithms is tested is executed.

The results of this experiment show how efficient the different detection techniques are. In the test environment, all of the presented statistics allow to say that the SIFT algorithm provides the maximal level of accuracy that is reached by any algorithm. This is true for the different distances, as well as for the angles and the scaling. The second best alternative is the SURF algorithm. But SIFT still dominates all of the results. Even the t-tests have proven that these findings are no coincidence.

The real environment is tested to find if there are differences in the level of accuracy when there are uncertainties and distraction in the image. The results show that there is one main difference. All of the algorithms provide a worse detection rate in the real environment. This is probably caused by factors like other logos, participants, or other distractions in the recorded picture. The images in the test environment only include the shop logo that should be recognized in front of a white wall. These settings allow less wrong interpretations than the setting in the real environment.

Another difference to the test environment is the worse detection accuracy for the closest distance in the real environment. This kind of behavior could have many reasons. The simplest one could be the quality of the images. To prove this assumption, I check the images by the manual detection of the application. During this visual test I cannot find any obvious reason for the wrong detections of the images that are close to the object. The algorithm to evaluate and determine the wrong detections works fine for all images. Another reason for this could be that when the image is recorded closer to the object, the object fills most of the image. This also means that it reinforces the amount of feature points that are found. If some points are now connected in a wrong way, the matching of the homography fails. An image recorded with more distance means that less details are visible for the detection. That also includes wrong detections. This seems to be the most appropriate explanation for this effect.

This explanation is confirmed by the results presented in Figure 11. In this diagram, it is shown that the recognition process has a higher accuracy for short distances when the image resolution is reduced. This effect gives confidence to the previously assumption that too high levels of detail in the images with low distances can have an adverse influence on the recognition process.

The first two experiments also evaluate how the scaling of images can influence the calculation time. Therefore, Figure 7 gives an overview of how the calculation time is related to the scaling of the image. Here, it becomes clear that the algorithm with the best accuracy is not automatically the slowest algorithm. Especially for the low-scaled values, SIFT can provide the best calculation time for a single recognition process. With larger images, the calculation time of SIFT is higher than the calculation time of most other algorithms. The main point why the SIFT recognition still seems like the best option for a mobile device is the combination of time and accuracy. If we compare an algorithm like MSER with an accuracy of 26.6 percent and a time interval of 314 milliseconds, SIFT - with more than 89 percent - still seems to be the better option, even if the time interval of 418 milliseconds is much higher for the image in this scale. The benefit of time is also obsolete if we choose the option of scaling the image. This way, SIFT would still provide better results with less time for a detection.

This speaks against the assumption mentioned by Morimitsu [17]. In this paper, SIFT is mentioned as an accurate but cost intensive algorithm that leads to good results, but has a high latency.

The experiments in this thesis show that SIFT may have a higher calculation time, but this effect can be neglected by the possible time reduction without worsening the results more than using another algorithm would.

The test in the real environment confirms the assumption that SIFT is the most appropriate algorithm in detection accuracy. The time is also no criteria to exclude SIFT as the best detection method. All in all, SIFT is the best combination of absolutely necessary accuracy and the also important calculation time. This is the reason why further implementations focus only on this detection technique.

The processing speed is the main drawback of the image recognition in an AR. The algorithms currently need more processing power than the mobile devices can provide. Therefore, a reduction of the latency between two frames is essential for a smooth application. The three techniques presented in the chapter Progressing Speed are able to reduce the calculation time to a minimum.

However, the possible time reduction is limited. The cropping and the scaling of images use the reduction on image pixel to gain acceleration. This effect does not behave linearly in the reduction of accuracy and time. The first scaling in Figure 12 is a good example for a good gain of time. While the accuracy is reduced by 2.5 percent, the calculation time is pared down to 59 percent. This means a maximum time reduction with nearly no loss in correct detections. For the side length reduction of 50 percent, the loss of successful detections grows. But the accuracy reduction of nearly 16 percent could still be justified with the time reduction of more than 70 percent. For the last scale, it is hard to argue that this reduction is worthwhile. The loss in accuracy exceeds the gain of time by a multiple. The same impression is given by the information collected in the real environment (Figure 13). This shows for the scaling that a reduction can increase the calculations per second, but the pixel reduction has to be practiced with caution. A high loss of accurate detections is an issue that goes along with a high level of time reduction.

To understand the reduction of time, it is necessary to include the amount of pixel into the dataset. By reducing the side length to 75 percent, it is possible to reduce the amount of pixel by nearly 44 percent. This value coincides with the gain of time in Figure 12.

Also, the reduction of 50 percent delivers a value that fits to the gain of time in Figure 12. To dive the side length into the half means that the amount of pixel is reduced by 75 percent. This approximates the values that can be found in the test.

This example can also be executed with the lowest scaling in the test. There, the side length reduction of 75 percent leads to only 6.25 percent of the pixel-starting sum. The needed time in the tests is nearly twice as much.

It is important to note that there are also processes that cannot be reduced in their time consume. They are fix processes in the application that are separated from the image size. When taking into account these processes, the time reduction of all scales converges with the amount of pixel. The fix amount of time for each recognition is round about 6 percent. This is also the reason why the time reduction is shrinking the more the image is scaled.

The other pixel reduction method that is presented is called cropping. Figure 14 proves the assumption that a pixel reduction by cropping is equivalent to a pixel reduction by scaling. The main difference is the influences on the accuracy. While scaling enables the application to focus on a fast recognition on the whole recorded image, the crop only uses the center of the picture. This is also shown in the results of the user test. There, scaling has huge problems to detect the object from a further distance, while the crop provides worse precision on close distances.

For both pixel reduction strategies there are drawbacks for the usage inside of an indoor Augmented Reality. When a crop is performed, the application may not be able to detect an image that is close to the camera because of the discarded edges. Another disadvantage is also that for AR purposes, the large viewing angle is a preferred feature. This allows covering a bigger area and visualizing more objects on the screen. This effect is lost when cropping the edges of the recorded image.

A detection of images that are close to the edge is possible with scaled images. However, the distance is the problem with this setting. The experiments give the knowledge that recognition is only applicable for short distances. This also leads to problems in the detection process.

This can also be observed in the results of the user test. Figure 16 and 17 show that for close distances, a scaling solution leads to a higher detection rate than a cropped image can provide. This effect is reversed with increasing distance. A combination of crop and scale is able to deliver a mixed result. This means that the detection accuracy is in between the two curves of full scaling and cropping, which also agrees with the explanation that crop is good for far and scale for near distances.

It would actually be helpful to use both features without drawbacks. One option is that the combination of crop and scale could be adjusted inside the application. But for this, an intervention of the user would be necessary. This could lead to other problems, like a wrong controlling of the setting. Even with the knowledge of how to adjust it correctly, the extra investment for the user should be avoided. One option to do so is the automated distance measuring with the help of the autofocus procedure [36]. This could optimize the combination of scaling and cropping the image without an interaction of the user.

The last presented technique to improve the calculations per second is to outsource the calculation into different threads. This way, it is possible to double the amount of calculations. Despite this success, there is still a main drawback. The acceleration of the frame rate does not automatically mean that the calculation time of the recognition is also accelerated. In this case, the delay from the image recording to the completed recognition requires the same time with and without multithreading. This means that splitting calculation processes based on each other increases the frame rate, but not the calculation time. This effect is desirable for the user experience, but does not solve the problem of latency. A possible option to reduce the latency is a reconstruction of the OpenCV library. With the including of multithreading in single calculation processes it could be possible to manufacture results faster and without latency.

Another possible variant is to use the Graphic Processor Unit (GPU) for calculations [37]. With this technique it could also be possible to gain an acceleration of calculation.

Figure 14 and 15 suggest that a time reduction is possible but limited. To make use of all the possible calculation time reduction methods it could be helpful to combine multithreading, scaling, and also cropping. This is done in the prototype implementation and leads to a calculation latency of nearly one second.

However, the reached reduction is still not enough to provide a smooth real time AR application. This confirms the finding of Wendt that AR applications based on image recognition are possible, but still time-intensive [22]. Even with different time reduction methods and the newest hardware, one frame per second for each SIFT detection seems to not be enough for real time performance.

One possible chance to use an AR without providing 24 calculations per second is to estimate the position of recognized objects – the current standard for movies [38]. This can be done with the device internal sensors like the gyroscope or the accelerometer. The technique to do this is already implemented in existing Augmented Reality projects. In the droidar project, this method is applied to display a solar system around the user [39]. The objects keep the same position even when the device is turned around. After a successful recognition, this technique could help to show the information in the right place even when the mobile device is turned.

This can help to analyze the shift of the rotation and not the movement of a user. Therefore, the accelerometer provides important information about the human locomotion [40]. It is possible to extract the distance covered by the moved steps [41]. A combination of these two techniques could also provide a smooth application handling without the amount of 24 recognitions per second. But there is the problem that one calculation in one second may still not be enough, even with supporting features. For an appropriate usage of the image recognition methods, the devices still need more processing power. Surely, the point when mobile devices have enough capabilities to provide this functionality is coming. Especially the field of mobile devices shows a fast development of processing power, like the benchmark of Schmerer illustrates [42].

8. Conclusion

In this master thesis I investigate the question of how an indoor Augmented Reality based on image recognition can be realized on mobile devices.

During several experiments I come to the conclusion that there are image recognition techniques that already provide the accuracy to detect objects and images on mobile devices.

In addition, also the calculation time and possible reduction methods are discussed. Therefore, the reduction of pixel by scaling and cropping the image, as well as the usage of more than one central processing unit for the calculation seems like a promising approach. In particular, it is possible to reduce the needed calculation time of the most reliable image recognition algorithm SIFT to a minimum.

In the application prototype, it is possible to reach one calculation per second without a strong deterioration of the detection results. Even though this represents a high reduction of time, it is still not enough for the usage of a real time application.

To reach this aim, further investigation on how to process an Augmented Reality application with a minimum of reliable information have to be concluded.

9. Future Work

The work presented in this master thesis shows that there are certain drawbacks in the field of image recognition based Augmented Reality applications for mobile devices. The main obstacle is the calculation speed. Therefore, it would be interesting to determine whether it is possible to decrease the latency of the calculation.

One possible approach is to rebuild the OpenCV library and implement the detection algorithms and their feature point detection in more than one thread. Especially for the future, this reconstruction can be helpful. The trend of built mobile devices with more than one core is probably increasing in the future. This also means that the impact of a speed extension by multithreading also gains importance.

Furthermore, an implementation on another platform like Android would be interesting. The fact that Apple supports only a couple of devices also implies that the range of supported processor capabilities is limited. For example, many device manufacturers support the Android operation system platform. This means also that there are currently low pricing devices with low processing power. However, there are also devices that outperform the calculation capabilities of all the Apple devices. This is the reason why it would be interesting to analyze the behavior in cost intensive image recognition.

Another possible improvement of this work is the investigation of how the current frame rate can be used to provide usable applications. Therefore, the usage of the internal sensors is a promising approach. The frame rate may be stabilized with equalization. The set of sensors in a mobile device also include the Gyroscope [43] and an Accelerometer [44]. These sensors have the ability to register movements and rotations. When the image recognition could be combined with these sensors, a tracking without looping the recognition process is imaginable. The only precondition needed is an initial recognition. This stabilization of the frame rate is also a chance to provide a smooth usability.

Another interesting question is how image recognition for Augmented Reality applications might be used in an appropriate way. The underlying data sets that must be provided to support recognitions of objects in all environments require a large amount of information. The storage of this information on a device can lead to certain problems. A possibility to solve such problems is the link of location-based services. This could help to save storage and performance on the devices. Furthermore, knowing the approximated position may also be an advantage in the recognition process. When the location can eliminate wrong detections and improve the detection of images that should have a high chance to occur at this position, the recognition process may be more effective. The combination of image recognition and location-based services is an interesting topic for future studies.

A possible improvement for the detection process might be the combination of further computer vision techniques. The evaluation of color detection like Juang et al. [45] did in combination with image recognition is interesting for providing a better detection process. If it is possible to eliminate regions in the image by their color, the detection algorithms could be able to reduce the amount of data they have to deal with.

The same point may be made with regards to the shape of the objects that should be recognized like presented by Ballard [46]. If the shape is known and it is for example a rectangle, it can be an option to search for these forms in the recorded image. A recognition process could directly proceed with the recognition inside of the rectangles.

However, it is also conceivable to use a combination of color detection and shape recognition in order to increase the calculation speed of the image recognition process. This is also an option that might be an interesting topic for further studies.

10. Literature

[1] Google Maps. [Online]. https://maps.google.de/ (Retrieved December 25, 2012)

[2] Paul Milgram, Haruo Takemura, Akira Utsumi, and Fumio Kishino, "Augmented Reality: A class of displays on the reality-virtuality continuum," in *SPIE Conference on Telemanipulator and Telepresence Technologies, Volume 2351*, 1994, pp. 282-292.

[3] Ronald Azuma, "A Survey of Augmented Reality," in *In Presence: Teleoperators and Virtual Environments, Volume 6, Number 4*, 1997, pp. 355-385.

[4] wikitude. [Online]. http://www.wikitude.com/ (Retrieved December 20, 2012)

[5] Steven Feiner, Blair MacIntyre, Tobias Höllerer, and Anthony Webster, "A Touring Machine: Prototyping 3D Mobile Augmented Reality Systems for Exploring the Urban Environment," in *Personal and Ubiquitous Computing*, 1997, pp. 208 - 217.

[6] Dhruv Pandya, Ravi Jain, and Emil Lupu, "lndoor location estimation using multiple wireless technologies," in *14th IEEE International Symposium on Personal, lndoor and Mobile Radio Communication Proceedings, Volume 3*, 2003, pp. 2208 - 2212.

[7] Michael Wing, Aaron Eklund, and Loren Kellogg, "Consumer-Grade Global Positioning System (GPS) Accuracy and Reliability," in *Journal of Forestry, Volume 103, Number 4*, June 2005, pp. 169-173.

[8] Paramvir Bahl and Venkara Padmanabhan, "RADAR: an in-building RF-based user location and tracking system," in *19th Annual Joint Conference of the IEEE Computer and Communications Societies, Volume 2*, 2000, pp. 775 - 784.

[9] Daniel Wagner and Dieter Schmalstieg, "ARToolKit on the PocketPC Platform," in *IEEE International Augmented Reality Toolkit Workshop*, 2003, pp. 14 - 15.

[10] Hirokazu Kato and Mark Billinghurst, "Marker Tracking and HMD Calibration for a Video-based Augmented Reality Conferencing System," in *2nd IEEE and ACM International Workshop on Augmented Reality (IWAR'99)*, 1999, pp. 85 - 94.

[11] Jun Rekimoto, "Matrix: a realtime object identification and registration method for augmented reality," in *3rd Asia Pacific Computer Human Interaction*, 1998, pp. 63 - 68.

[12] Iryna Skrypnyk and David Lowe, "Scene Modelling, Recognition and Tracking with Invariant Image Features," in *3rd IEEE/ACM International Symposium on Mixed and Augmented Reality (ISMAR '04)*, 2004, pp. 110 - 119.

[13] Andrew Comport, Eric Marchand, Muriel Pressigout, and Francois Chaumette, "Real-Time Markerless Tracking for Augmented Reality: The Virtual Visual Servoing Framework," in *IEEE Transactions on Visualization and Computer Graphics*, 2006, pp. 615 - 628.

[14] Youngmin Park, Vincent Lepetit, and Woontack Woo, "Multiple 3D Object tracking for augmented reality," in *7th IEEE/ACM International Symposium on Mixed and Augmented Reality (ISMAR '08)*, 2008, pp. 117 - 120.

[15] Ana Bernardos, Jesús Cano, Josué Iglesias, and José Casar, "MOBILE INDOOR AUGMENTED REALITY Exploring Applications in Hospitality Environments," in *International Conference on Pervasive and Embedded Computing and Communication Systems (PECCS '11)*, 2011, pp. 232 - 236.

[16] Fakhreddine Ababsa and Malik Mallem, "Robust Camera Pose Estimation Using 2D Fiducials Tracking for Real-Time Augmented Reality Systems," in *Virtual Reality continuum and its applications in industry (VRCAI '04)*, 2004, pp. 431 - 435.

[17] Henrique Morimitsu, Marcelo Hashimoto, Rodrigo Pimentel, Robert Cesar-jr., and Roberto Hirata-jr., "Keygraphs for Sign Detection in Indoor Environments by Mobile Phones," in *8th international conference on Graph-based representations in pattern recognition (GbRPR'11)*, 2011, pp. 315 - 324.

[18] A Murillo, J Guerrero, and C Sagüés, "SURF features for efficient robot localization with omnidirectional images," in *IEEE International Conference on Robotics and Automation*, 2007, pp. 3901 - 3907.

[19] Toon Goedemé, Beat Fasel, and Luc Van Gool, "The Visual Virtual Tourist Guide: A Markerless Camera-based LBS System," in *The Integration of Location Based Services in Tourism and Cultural Heritage*, 2007, pp. 21 - 32.

[20] Toon Goedemé et al., "Is structure needed for omnidirectional visual homing?," in *IEEE International Symposium on Computational Intelligence in Robotics and Automation*, 2005, pp. 303 - 308.

[21] Henrik Brauer, "Entwicklung eines Augmented Reality Frameworks auf Basis von Kamera-basierten Trackingverfahren," in *M.S. thesis, Dept. Informatik, Hamburg University of Applied Sciences*, Hamburg, Germany, 2010.

[22] Frank Wendt, "SIFT based Augmented Reality," in *M.S. thesis, Dept. Geomatics, Computer Science and Mathematics, University of Applied Sciences in Stuttgart*, Stuttgart, Germany, 2007.

[23] OpenCV. [Online]. http://opencv.willowgarage.com/wiki/ (Retrieved December 26, 2012)

[24] David Lowe, "Object Recognition from Local Scale-Invariant Features," in *International Conference on Computer Vision*, 1999, pp. 1150 - 1157.

[25] Herbert Bay, Tinne Tuytelaars, and Luc Van Gool, "SURF: Speeded Up Robust Features," in *European Conference on Computer Vision (ECCV '06)*, 2006, pp. 404 - 417.

[26] Michael Donoser and Horst Bischof, "Efficient Maximally Stable Extremal Region (MSER) Tracking," in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition, Volume 1 (CVPR'06)*, 2006, pp. 553 - 560.

[27] Jinabo Shi and Carlo Tomasi, "Good Features to Track," in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR '94)*, 1994, pp. 593 - 600.

[28] Neeraj Kanhere and Stanley Birchfield, "Real-Time Incremental Segmentation and Tracking of Vehicles at Low Camera Angles Using Stable Features," in *IEEE Transactions on Intelligent Transportation Systems, Volume 9, Number 1*, 2008, pp. 148 - 160.

[29] Edward Rosten and Tom Drummong, "Fusing points and lines for high performance tracking," in *IEEE International Conference on Computer Vision (ICCV '05), Volume 2*, 2005, pp. 1508 - 1515.

[30] Edward Rosten and Tom Drummond, "Machine learning for high-speed corner detection," in *European Conference on Computer Vision (ECCV '06)*, 2006, pp. 430 - 443.

[31] Motilal Agrawal, Kurt Konolige, and Morten Rufus Blas, "CenSurE: Center Surround Extremas for Realtime Feature Detection and Matching," in *Lecture Notes in Computer Science, Volume 5305 (ECCV '08)*, 2008, pp. 102 - 115.

[32] (2008) Willow Garage - Keypoint Detectors. [Online]. http://pr.willowgarage.com/wiki/Star_Detector?action=AttachFile&do=get&target=KeypointSlides.pdf (Retrieved December 23, 2012)

[33] ZXing. [Online]. http://code.google.com/p/zxing/ (Retrieved December 24, 2012)

[34] Apache Commons. [Online]. http://commons.apache.org/math/
(Retrieved December 26,2012)

[35] OpenCSV. [Online]. http://opencsv.sourceforge.net/
(Retrieved December 26, 2012)

[36] Renewed March. (2008, March) Nikon - Predictive Focus Tracking System. [Online].
http://www.nikon.com/about/technology/rd/core/software/caf/index.htm
(Retrieved December 25, 2012)

[37] John Owens et al., "GPU Computing," in *Proceedings of the IEEE, Volume 96, Number 5*, 2008, pp. 879 - 899.

[38] Kevin Brownlow. (1980) Silent Films - What Was the Right Speed? [Online]. http://web.archive.org/web/20110708155615/http://www.cinemaweb.com/silentfilm/bookshelf/18_kb_2.htm (Retrieved December 23, 2012)

[39] Simon Heinen. droidar. [Online]. http://code.google.com/p/droidar/
(Retrieved December 22, 2012)

[40] K. Aminian, Ph. Robert, E. Buchser, and B. Ruts, "Physical activity monitoring based on accelerometry: validation and comparison with video observation," in *Medical and Biological Engineering and Computing, Volume 37, Number 3*, 1997, pp. 304 - 308.

[41] Quentin Ladetto, "On foot navigation : continuous step calibration using both complementary recursive prediction and adaptive Kalman filtering," in *13th International Technical Meeting of the Satellite Division of The Institute of Navigation (ION GPS '00)*, 2000, pp. 1735 - 1740.

[42] Kai Schmerer. (2012, October) iPhone 5 im Benchmarktest. [Online].
http://www.zdnet.de/88127121/iphone-5-im-benchmarktest/
(Retrieved December 25, 2012)

[43] Shannon Range and Jennifer Mullins. (2011, February) NASA - Brief History of Gyroscopes. [Online].
http://solarsystem.nasa.gov/scitech/display.cfm?ST_ID=327
(Retrieved December 24, 2012)

[44] Kong Chen and David Bassett, "The technology of accelerometry-based activity monitors: current and future.," in *Medicine & Science in Sports & Exercise*, 2005, pp. 490 - 500.

[45] Chia-Feng Juang, Wen-Kai Sun, and Guo-Cyuan Chen, "Object detection by color histogram-based fuzzy classifier with support vector learning," in *Neurocomputing Volume 72, Number 10–12*, 2009, pp. 2464 - 2476.

[46] D. Ballard, "Generalizing the Hough transform to detect arbitrary shapes," in *Pattern Recognition, Volume 13, Number 2*, 1981, pp. 111 - 122.

Statutory Declaration

I hereby affirm that this Master thesis at hand is my own written work and that I have used no other sources and aids than those indicated.

All passages which are quoted from publications or paraphrased from these sources are indicated as such.

All illustrations and charts in this thesis have been created by my person. Illustrations and charts which are taken from publications or paraphrased from these sources are indicated as such.

This thesis was not submitted in the same or in a substantially similar version to another examination board.

Münster, December 27, 2012

Philipp Weiß

Annex

The following pages include the annex of this master thesis. The elements are the following:

- Questionnaire used for the user experiment
  (1 page)
- Results of the user experiment
  (15 pages)
- DVD (pictures, source code)

Questionnaire used for the user experiment

Nutzerbewertung in Noten 0-5 (5 = sehr gut; 0 = sehr schlecht)

Typ 0	2.5m	5m	7.5m	10m
0°				
22.5°				
45°				
67.5°				

Typ 1	2.5m	5m	7.5m	10m
0°				
22.5°				
45°				
67.5°				

Typ 2	2.5m	5m	7.5m	10m
0°				
22.5°				
45°				
67.5°				

Gesamtnote für die Einstellmöglichkeiten

0: 1: 2:

Warum diese Note:

0:

1:

2:

Wie sinnvoll ist das Skalieren des Bildes? Note:

Wie sinnvoll ist das Abschneiden der Bildränder? Note:

Wozu könnte ein Programm mit Bilderkennung nützlich sein?

Results of the user experiment

Participant 1:

Rating of participant 1 for the settings of type 0 (50% scaling, no cropping) in the user test. Grade from 0 to 5. 5 represents the best possible results.

	2.5m	5.0m	7.5m	10.0m
0°	5	5	0	0
22.5°	4	4	0	0
45°	0	2	0	0
67.5°	0	0	0	0

Rating of participant 1 for the settings of type 1 (25% scaling, 25% cropping) in the user test. Grade from 0 to 5. 5 represents the best possible results.

	2.5m	5.0m	7.5m	10.0m
0°	5	5	5	0
22.5°	4	5	3	0
45°	0	2	0	0
67.5°	0	0	0	0

Rating of participant 1 for the settings of type 2 (no scaling, 50% cropping) in the user test. Grade from 0 to 5. 5 represents the best possible results.

	2.5m	5.0m	7.5m	10.0m
0°	5	5	5	5
22.5°	3	5	4	4
45°	0	2	0	0
67.5°	0	0	0	0

Total grading and comments of the different settings of participant 1. Type 0 represents 50% scaling and no cropping; Type 1 represents 25% scaling and 25% cropping; Type 2 no scaling and 50% cropping. Grade from 0 to 5. 5 represents the best possible results.

Type/Parameter	Grade	Comment
0	4	oft keine Erkennung
1	3	ganz okay
2	2	beste Erkennung in größerer Entfernung
Scaling	5	nicht so sinnvoll, weil Typ 2 mehr erkannt hat
Cropping	1	sehr gut

Measuring of the detection success of participant 1 for the settings of type 0 (50% scaling, no cropping) in the user test.

	2.5m	5.0m	7.5m	10.0m
0°	100	100	0	0
22.5°	81	70	0	0
45°	0	33	0	0
67.5°	0	0	0	0

Measuring of the detection success of participant 1 for the settings of type 1 (25% scaling, 25% cropping) in the user test.

	2.5m	5.0m	7.5m	10.0m
0°	100	100	100	0
22.5°	73	100	75	0
45°	0	44	0	0
67.5°	0	0	0	0

Measuring of the detection success of participant 1 for the settings of type 2 (no scaling, 50% cropping) in the user test.

	2.5m	5.0m	7.5m	10.0m
0°	100	100	100	100
22.5°	43	90	95	82
45°	0	43	0	0
67.5°	0	0	0	0

Participant 2:

Rating of participant 2 for the settings of type 0 (50% scaling, no cropping) in the user test. Grade from 0 to 5. 5 represents the best possible results.

	2.5m	5.0m	7.5m	10.0m
0°	5	5	0	0
22.5°	4	5	0	0
45°	3	2	0	0
67.5°	0	0	0	0

Rating of participant 2 for the settings of type 1 (25% scaling, 25% cropping) in the user test. Grade from 0 to 5. 5 represents the best possible results.

	2.5m	5.0m	7.5m	10.0m
0°	5	5	5	0
22.5°	5	5	5	0
45°	3	4	0	0
67.5°	0	0	0	0

Rating of participant 2 for the settings of type 2 (no scaling, 50% cropping) in the user test. Grade from 0 to 5. 5 represents the best possible results.

	2.5m	5.0m	7.5m	10.0m
0°	5	5	5	5
22.5°	5	5	4	5
45°	0	4	3	0
67.5°	0	0	0	0

Total grading and comments of the different settings of participant 2. Type 0 represents 50% scaling and no cropping; Type 1 represents 25% scaling and 25% cropping; Type 2 no scaling and 50% cropping. Grade from 0 to 5. 5 represents the best possible results.

Type/Parameter	Grade	Comment
0	3	Fast die Hälfte der Versuche sind erfolgslos. Je weiter die Distanz ist, desto schlechter wird die Erkennung
1	4	Bei sehr weiten Entfernungen wird Erkennung schlecht
2	3	wie 0
Scaling	4	bringt was
Cropping	5	funktioniert

Measuring of the detection success of participant 2 for the settings of type 0 (50% scaling, no cropping) in the user test.

	2.5m	5.0m	7.5m	10.0m
0°	80	84	0	0
22.5°	82	100	0	0
45°	20	20	0	0
67.5°	0	0	0	0

Measuring of the detection success of participant 2 for the settings of type 1 (25% scaling, 25% cropping) in the user test.

	2.5m	5.0m	7.5m	10.0m
0°	80	90	74	0
22.5°	80	90	70	0
45°	10	82	0	0
67.5°	0	0	0	0

Measuring of the detection success of participant 2 for the settings of type 2 (no scaling, 50% cropping) in the user test.

	2.5m	5.0m	7.5m	10.0m
0°	86	88	70	62
22.5°	80	80	74	84
45°	0	70	24	0
67.5°	0	0	0	0

Participant 3:

Rating of participant 3 for the settings of type 0 (50% scaling, no cropping) in the user test. Grade from 0 to 5. 5 represents the best possible results.

	2.5m	5.0m	7.5m	10.0m
0°	4	0	0	0
22.5°	5	0	0	0
45°	1	0	0	0
67.5°	0	0	0	0

Rating of participant 3 for the settings of type 1 (25% scaling, 25% cropping) in the user test. Grade from 0 to 5. 5 represents the best possible results.

	2.5m	5.0m	7.5m	10.0m
0°	5	5	0	0
22.5°	5	5	0	0
45°	3	2	0	0
67.5°	0	0	0	0

Rating of participant 3 for the settings of type 2 (no scaling, 50% cropping) in the user test. Grade from 0 to 5. 5 represents the best possible results.

	2.5m	5.0m	7.5m	10.0m
0°	5	5	4	2
22.5°	5	5	5	3
45°	0	5	2	0
67.5°	0	0	0	0

Total grading and comments of the different settings of participant 3. Type 0 represents 50% scaling and no cropping; Type 1 represents 25% scaling and 25% cropping; Type 2 no scaling and 50% cropping. Grade from 0 to 5. 5 represents the best possible results.

Type/Parameter	Grade	Comment
0	1	Ab einer gewissen Entfernung und einem größeren Winkel klappt es nicht mehr
1	3	wie Type 0 nur besser
2	4	am besten
Scaling	3	sorgt für eine deutliche Beschleunigung
Cropping	5	ein guter Filter von uninteressanten Inhalten

Measuring of the detection success of participant 3 for the settings of type 0 (50% scaling, no cropping) in the user test.

	2.5m	5.0m	7.5m	10.0m
0°	100	0	0	0
22.5°	100	0	0	0
45°	6	0	0	0
67.5°	0	0	0	0

Measuring of the detection success of participant 3 for the settings of type 1 (25% scaling, 25% cropping) in the user test.

	2.5m	5.0m	7.5m	10.0m
0°	100	100	0	0
22.5°	100	80	0	0
45°	22	20	0	0
67.5°	0	0	0	0

Measuring of the detection success of participant 3 for the settings of type 2 (no scaling, 50% cropping) in the user test.

	2.5m	5.0m	7.5m	10.0m
0°	100	94	88	32
22.5°	100	84	84	62
45°	0	80	6	0
67.5°	0	0	0	0

Participant 4:

Rating of participant 4 for the settings of type 0 (50% scaling, no cropping) in the user test. Grade from 0 to 5. 5 represents the best possible results.

	2.5m	5.0m	7.5m	10.0m
0°	5	0	0	0
22.5°	5	0	0	0
45°	4	0	0	0
67.5°	0	0	0	0

Rating of participant 4 for the settings of type 1 (25% scaling, 25% cropping) in the user test. Grade from 0 to 5. 5 represents the best possible results.

	2.5m	5.0m	7.5m	10.0m
0°	5	5	0	0
22.5°	5	5	0	0
45°	3	0	0	0
67.5°	0	0	0	0

Rating of participant 4 for the settings of type 2 (no scaling, 50% cropping) in the user test. Grade from 0 to 5. 5 represents the best possible results.

	2.5m	5.0m	7.5m	10.0m
0°	5	5	4	0
22.5°	3	5	5	0
45°	2	0	0	0
67.5°	0	0	0	0

Total grading and comments of the different settings of participant 4. Type 0 represents 50% scaling and no cropping; Type 1 represents 25% scaling and 25% cropping; Type 2 no scaling and 50% cropping. Grade from 0 to 5. 5 represents the best possible results.

Type/Parameter	Grade	Comment
0	2	schlechte Erkennung
1	3	besser
2	4	am besten
Scaling	2	
Cropping	4	

Measuring of the detection success of participant 4 for the settings of type 0 (50% scaling, no cropping) in the user test.

	2.5m	5.0m	7.5m	10.0m
0°	100	0	0	0
22.5°	100	0	0	0
45°	33	0	0	0
67.5°	0	0	0	0

Measuring of the detection success of participant 4 for the settings of type 1 (25% scaling, 25% cropping) in the user test.

	2.5m	5.0m	7.5m	10.0m
0°	100	83	0	0
22.5°	100	85	0	0
45°	66	0	0	0
67.5°	0	0	0	0

Measuring of the detection success of participant 4 for the settings of type 2 (no scaling, 50% cropping) in the user test.

	2.5m	5.0m	7.5m	10.0m
0°	100	82	80	0
22.5°	80	80	75	0
45°	11	43	0	0
67.5°	0	0	0	0

Participant 5:

Rating of participant 5 for the settings of type 0 (50% scaling, no cropping) in the user test. Grade from 0 to 5. 5 represents the best possible results.

	2.5m	5.0m	7.5m	10.0m
0°	5	5	0	0
22.5°	5	5	0	0
45°	4	5	0	0
67.5°	0	0	0	0

Rating of participant 5 for the settings of type 1 (25% scaling, 25% cropping) in the user test. Grade from 0 to 5. 5 represents the best possible results.

	2.5m	5.0m	7.5m	10.0m
0°	5	5	5	0
22.5°	5	5	5	0
45°	3	5	5	0
67.5°	1	0	0	0

Rating of participant 5 for the settings of type 2 (no scaling, 50% cropping) in the user test. Grade from 0 to 5. 5 represents the best possible results.

	2.5m	5.0m	7.5m	10.0m
0°	3	5	5	5
22.5°	4	5	5	5
45°	3	4	5	4
67.5°	1	0	0	0

Total grading and comments of the different settings of participant 5. Type 0 represents 50% scaling and no cropping; Type 1 represents 25% scaling and 25% cropping; Type 2 no scaling and 50% cropping. Grade from 0 to 5. 5 represents the best possible results.

Type/Parameter	Grade	Comment
0	4	sehr gut aus der Nähe, schlecht aus der Ferne
1	5	guter Kompromiss
2	4	schlecht aus der Nähe
Scaling	4	
Cropping	4	

Measuring of the detection success of participant 5 for the settings of type 0 (50% scaling, no cropping) in the user test.

	2.5m	5.0m	7.5m	10.0m
0°	87	100	0	0
22.5°	100	100	0	0
45°	71	94	0	0
67.5°	0	0	0	0

Measuring of the detection success of participant 5 for the settings of type 1 (25% scaling, 25% cropping) in the user test.

	2.5m	5.0m	7.5m	10.0m
0°	72	90	100	0
22.5°	91	83	100	0
45°	47	90	100	0
67.5°	12	0	0	0

Measuring of the detection success of participant 5 for the settings of type 2 (no scaling, 50% cropping) in the user test.

	2.5m	5.0m	7.5m	10.0m
0°	44	95	95	90
22.5°	65	50	94	95
45°	33	42	94	87
67.5°	4	0	0	0

Participant 6:

Rating of participant 6 for the settings of type 0 (50% scaling, no cropping) in the user test. Grade from 0 to 5. 5 represents the best possible results.

	2.5m	5.0m	7.5m	10.0m
0°	5	5	0	0
22.5°	4	5	0	0
45°	4	4	0	0
67.5°	0	1	0	0

Rating of participant 6 for the settings of type 1 (25% scaling, 25% cropping) in the user test. Grade from 0 to 5. 5 represents the best possible results.

	2.5m	5.0m	7.5m	10.0m
0°	5	5	5	0
22.5°	5	5	3	0
45°	3	4	2	0
67.5°	0	0	0	0

Rating of participant 6 for the settings of type 2 (no scaling, 50% cropping) in the user test. Grade from 0 to 5. 5 represents the best possible results.

	2.5m	5.0m	7.5m	10.0m
0°	5	5	5	5
22.5°	5	5	4	4
45°	0	3	2	2
67.5°	0	0	0	0

Total grading and comments of the different settings of participant 6. Type 0 represents 50% scaling and no cropping; Type 1 represents 25% scaling and 25% cropping; Type 2 no scaling and 50% cropping. Grade from 0 to 5. 5 represents the best possible results.

Type/Parameter	Grade	Comment
0	2	oft keine Erkennung
1	3	Mitte
2	4	am besten
Scaling	4	
Cropping	4	

Measuring of the detection success of participant 6 for the settings of type 0 (50% scaling, no cropping) in the user test.

	2.5m	5.0m	7.5m	10.0m
0°	100	100	0	0
22.5°	75	100	0	0
45°	53	63	0	0
67.5°	0	10	0	0

Measuring of the detection success of participant 6 for the settings of type 1 (25% scaling, 25% cropping) in the user test.

	2.5m	5.0m	7.5m	10.0m
0°	100	100	100	0
22.5°	100	100	100	0
45°	29	86	30	0
67.5°	0	0	0	0

Measuring of the detection success of participant 6 for the settings of type 2 (no scaling, 50% cropping) in the user test.

	2.5m	5.0m	7.5m	10.0m
0°	100	100	100	100
22.5°	87	100	100	89
45°	90	36	79	27
67.5°	0	0	0	0

Participant 7:

Rating of participant 7 for the settings of type 0 (50% scaling, no cropping) in the user test. Grade from 0 to 5. 5 represents the best possible results.

	2.5m	5.0m	7.5m	10.0m
0°	5	5	0	0
22.5°	5	4	0	0
45°	0	0	0	0
67.5°	0	0	0	0

Rating of participant 7 for the settings of type 1 (25% scaling, 25% cropping) in the user test. Grade from 0 to 5. 5 represents the best possible results.

	2.5m	5.0m	7.5m	10.0m
0°	5	5	1	0
22.5°	4	5	0	0
45°	2	3	0	0
67.5°	0	0	0	0

Rating of participant 7 for the settings of type 2 (no scaling, 50% cropping) in the user test. Grade from 0 to 5. 5 represents the best possible results.

	2.5m	5.0m	7.5m	10.0m
0°	1	5	3	0
22.5°	3	5	1	0
45°	0	4	0	0
67.5°	0	0	0	0

Total grading and comments of the different settings of participant 7. Type 0 represents 50% scaling and no cropping; Type 1 represents 25% scaling and 25% cropping; Type 2 no scaling and 50% cropping. Grade from 0 to 5. 5 represents the best possible results.

Type/Parameter	Grade	Comment
0	3	nah gut und weiter weg nicht
1	3	Mischung aus 0 und 2
2	2	Man bekommt nicht immer alles ins Bild.
Scaling	4	bei Nähe gut
Cropping	3	in der Entfernung besser

Measuring of the detection success of participant 7 for the settings of type 0 (50% scaling, no cropping) in the user test.

	2.5m	5.0m	7.5m	10.0m
0°	50	100	0	0
22.5°	100	88	0	0
45°	0	0	0	0
67.5°	0	0	0	0

Measuring of the detection success of participant 7 for the settings of type 1 (25% scaling, 25% cropping) in the user test.

	2.5m	5.0m	7.5m	10.0m
0°	88	100	30	0
22.5°	87	100	0	0
45°	15	71	0	0
67.5°	0	0	0	0

Measuring of the detection success of participant 7 for the settings of type 2 (no scaling, 50% cropping) in the user test.

	2.5m	5.0m	7.5m	10.0m
0°	7	100	87	0
22.5°	55	66	23	0
45°	0	72	0	0
67.5°	0	0	0	0

<u>Participant 8:</u>

Rating of participant 8 for the settings of type 0 (50% scaling, no cropping) in the user test. Grade from 0 to 5. 5 represents the best possible results.

	2.5m	5.0m	7.5m	10.0m
0°	5	0	0	0
22.5°	5	4	0	0
45°	5	0	0	0
67.5°	0	0	0	0

Rating of participant 8 for the settings of type 1 (25% scaling, 25% cropping) in the user test. Grade from 0 to 5. 5 represents the best possible results.

	2.5m	5.0m	7.5m	10.0m
0°	5	5	0	0
22.5°	5	5	0	0
45°	5	2	0	0
67.5°	0	0	0	0

Rating of participant 8 for the settings of type 2 (no scaling, 50% cropping) in the user test. Grade from 0 to 5. 5 represents the best possible results.

	2.5m	5.0m	7.5m	10.0m
0°	4	5	5	4
22.5°	5	5	5	4
45°	0	3	0	0
67.5°	0	0	0	0

Total grading and comments of the different settings of participant 8. Type 0 represents 50% scaling and no cropping; Type 1 represents 25% scaling and 25% cropping; Type 2 no scaling and 50% cropping. Grade from 0 to 5. 5 represents the best possible results.

Type/Parameter	Grade	Comment
0	1	aus der Nähe gut; man ist nicht oft so nah an einem Schild, dass es noch gut geht
1	4	Kombination
2	5	weit am besten
Scaling	3	
Cropping	4	

Measuring of the detection success of participant 8 for the settings of type 0 (50% scaling, no cropping) in the user test.

	2.5m	5.0m	7.5m	10.0m
0°	80	0	0	0
22.5°	100	50	0	0
45°	100	0	0	0
67.5°	0	0	0	0

Measuring of the detection success of participant 8 for the settings of type 1 (25% scaling, 25% cropping) in the user test.

	2.5m	5.0m	7.5m	10.0m
0°	80	100	0	0
22.5°	70	100	0	0
45°	80	33	0	0
67.5°	0	0	0	0

Measuring of the detection success of participant 8 for the settings of type 2 (no scaling, 50% cropping) in the user test.

	2.5m	5.0m	7.5m	10.0m
0°	60	100	100	80
22.5°	100	100	100	40
45°	0	40	0	0
67.5°	0	0	0	0

Participant 9:

Rating of participant 9 for the settings of type 0 (50% scaling, no cropping) in the user test. Grade from 0 to 5. 5 represents the best possible results.

	2.5m	5.0m	7.5m	10.0m
0°	5	5	0	0
22.5°	4	0	0	0
45°	3	3	0	0
67.5°	0	0	0	0

Rating of participant 9 for the settings of type 1 (25% scaling, 25% cropping) in the user test. Grade from 0 to 5. 5 represents the best possible results.

	2.5m	5.0m	7.5m	10.0m
0°	5	5	5	0
22.5°	5	5	5	0
45°	0	4	4	0
67.5°	0	0	0	0

Rating of participant 9 for the settings of type 2 (no scaling, 50% cropping) in the user test. Grade from 0 to 5. 5 represents the best possible results.

	2.5m	5.0m	7.5m	10.0m
0°	5	5	5	5
22.5°	5	5	5	2
45°	0	3	3	1
67.5°	0	0	0	0

Total grading and comments of the different settings of participant 9. Type 0 represents 50% scaling and no cropping; Type 1 represents 25% scaling and 25% cropping; Type 2 no scaling and 50% cropping. Grade from 0 to 5. 5 represents the best possible results.

Type/Parameter	Grade	Comment
0	4	
1	4	
2	5	
Scaling	3	
Cropping	4	

Measuring of the detection success of participant 9 for the settings of type 0 (50% scaling, no cropping) in the user test.

	2.5m	5.0m	7.5m	10.0m
0°	100	100	0	0
22.5°	87	0	0	0
45°	72	20	0	0
67.5°	0	0	0	0

Measuring of the detection success of participant 9 for the settings of type 1 (25% scaling, 25% cropping) in the user test.

	2.5m	5.0m	7.5m	10.0m
0°	100	100	100	0
22.5°	100	100	100	0
45°	0	80	50	0
67.5°	0	0	0	0

Measuring of the detection success of participant 9 for the settings of type 2 (no scaling, 50% cropping) in the user test.

	2.5m	5.0m	7.5m	10.0m
0°	100	100	100	88
22.5°	100	100	100	22
45°	0	50	69	16
67.5°	0	0	0	0

Participant 10:

Rating of participant 10 for the settings of type 0 (50% scaling, no cropping) in the user test. Grade from 0 to 5. 5 represents the best possible results.

	2.5m	5.0m	7.5m	10.0m
0°	5	5	0	0
22.5°	5	5	0	0
45°	5	4	0	0
67.5°	0	0	0	0

Rating of participant 10 for the settings of type 1 (25% scaling, 25% cropping) in the user test. Grade from 0 to 5. 5 represents the best possible results.

	2.5m	5.0m	7.5m	10.0m
0°	5	5	4	0
22.5°	5	5	5	0
45°	0	5	0	0
67.5°	0	0	0	0

Rating of participant 10 for the settings of type 2 (no scaling, 50% cropping) in the user test. Grade from 0 to 5. 5 represents the best possible results.

	2.5m	5.0m	7.5m	10.0m
0°	5	5	5	0
22.5°	5	5	5	2
45°	1	5	1	0
67.5°	0	0	0	0

Total grading and comments of the different settings of participant 10. Type 0 represents 50% scaling and no cropping; Type 1 represents 25% scaling and 25% cropping; Type 2 no scaling and 50% cropping. Grade from 0 to 5. 5 represents the best possible results.

Type/Parameter	Grade	Comment
0	2	zu schlechte Erkennung
1	3	mittel
2	4	bessere Erkennung
Scaling	4	
Cropping	4	

Measuring of the detection success of participant 10 for the settings of type 0 (50% scaling, no cropping) in the user test.

	2.5m	5.0m	7.5m	10.0m
0°	100	100	0	0
22.5°	100	100	0	0
45°	100	46	0	0
67.5°	0	0	0	0

Measuring of the detection success of participant 10 for the settings of type 1 (25% scaling, 25% cropping) in the user test.

	2.5m	5.0m	7.5m	10.0m
0°	100	100	0	0
22.5°	100	100	0	0
45°	100	46	0	0
67.5°	0	0	0	0

Measuring of the detection success of participant 10 for the settings of type 2 (no scaling, 50% cropping) in the user test.

	2.5m	5.0m	7.5m	10.0m
0°	100	100	100	0
22.5°	100	100	100	11
45°	30	100	40	0
67.5°	0	0	0	0

All participants:

What for a program with image recognition could be useful?

Participant	Comment
1	Zusatzinformation über Logos und Bilder erhalten
2	Bei Ausstellungsobjekten
3	Augmented Reality
4	Verkehr
5	Augmented Reality, Informationen zu Produkten
6	Zusatzinformation über Logos und Bilder erhalten (Öffnungszeiten/Produkte/Filialen)
7	Sachen identifizieren
8	Museen; um Kunst wieder zu erkennen usw… Infos über Produkte
9	-
10	Ladenschilder