Occupancy Analyzer

Global Software Development Project

Christoffer Wilken Pagaard
Jacob Romme Rasmussen
Theresa Brandt von Fackh
Tomas Miseikis

IT University of Copenhagen

15. December 2013

Abstract

Write abstract text...

Contents

1 Introduction

1.1 Context

Smart use of energy resources is an ongoing topic these days. The reduction of expenses is mostly the biggest driving factor for companies. But also the debate around climate change brings new legislation to reduce the waste of energy resources, whose production is damaging to the environment and future generations. The IT University of Copenhagen (ITU) has an interest in producing an occupancy model for commercial buildings, like the ITU building, to detect where energy resources are needed and where they can be saved. Energy resources are needed for e.g. lighting and heat-regulating systems, which are relevant for occupants in a commercial building. With the detected occupancy data the ITU can predict occupancy and develop concepts for a smart use of energy resources in commercial buildings.

The Strathmore University in Kenya has also an interest in building up an occupancy model, but mainly for surveillance reasons. Surveillance can be used for several purposes like traffic monitoring, public safety and facilities surveillance. An IT-based surveillance system can automatically analyze the scene without the use of human resources. By analyzing the scene the detection of occupancy is a major part. Moreover a real-time prediction model on top of the occupancy data can be used to prevent criminal activity by triggering alarms or other surveillance systems.

Currently there is no existing infrastructure to build up an occupancy model in the Strathmore University or the ITU building. Both universities want a solution for an occupancy analyzer based on Raspberry Pi computers due to the minimal consumption of computational and monetary resources. Furthermore, Starthmore University requests for an Android application, which represents a live-feed of the occupants in a monitored room.

A group of students from both universities have to collaborate to come up with a solution for an occupancy analyzer, which can satisfy the needs of both university interests. Ideally a product should be developed, which can be adapted to fit the needs of one or the other university. Furthermore, a collaboration project is mandatory for the student group from ITU, in which they have to face the challenges of global collaboration, navigate compromises and come up with a solution.

This report contains the product result, design of the product, details of the project work and the learning outcomes, which were achieved in the project with the globally distributed team from the perspective of the ITU students. The project team consists of international students located in Nairobi, Kenya (East African Time) and Copenhagen, Denmark (Central European Time).

1.2 Problem

The content of this project is to build an occupancy analyzer, which detects people in a room or corridor and predicts their movement. The occupancy analyzer has to be based on Raspberry Pi computers, which come with computational restrictions, and web cameras. A solution for the right architecture and programming languages has to be found, which can deal with those limited resources.

Due to the usage of web cameras, a visual detection of people has to be made. Analyzing images by detecting people - which are moving objects and not part of the room - is one major challenge to face. Visual conditions of the image can change as a result of daylight. For example, dynamic lighting and moving shadows should not influence the detection of people. To best capture occupants and the requirement to represent the occupants on an Android application, the position of web cameras becomes an important factor. The differentiation between multiple people, as well as between people and the setting of the room, is important for the quality of the detection. Only if reliable data about occupancy exists, can the data be used to construct a reliable prediction model usable for future concepts and projects.

The question of how the collected data can be used, has to be considered. Building prediction models, which relies on historical data and real-time data, is another requirement, which the project dealt with. Decisions on what kind of prediction for a meaningful application - like the one mentioned in paragraph 2 in Section 1.1 - and how the data will be stored and processed have to be made.

Besides the design and implementation of an occupancy analyzer, another task is the collaboration of students from two different universities. Cultural differences, difference in time, spatial distance and locally related influences have to be overcome. Different perspectives have to be combined and compromises have to be made.

1.3 Related Work

Background subtraction is an extremely important concept in surveillance applications, since this is the first step in detecting people and tracking their movement in monitored area. Y. Benezeth, P.-M. Jodoin, B. Emile, H. Laurent & C. Rosenberger [1], in their work *Compara-*

tive Study of Background Subtraction Algorithms, mention a basic motion detection approach by using a simple background subtraction technique. In this technique, the background image is taken when there is a total absence of motion in the monitored area. This background image is then subtracted from every newly taken image of the area in order to detect motion. This is possible since after subtraction the resulting image will either be almost totally black - meaning no motion occurred - or the image will have some resulting bright contours of detected differences.

We intend to apply this simple technique in the beginning of the project for the initial experiments and studies of object detection and tracking. We expect that these experiments and studies will provide us with some knowledge on how basic object detection works and how we can apply it for our specific problem.

Aditi Jog & Shirish Halbe [2], in their paper *Multiple Object Tracking using CAMshift Algorithm in Open CV*, present a better way of building object detection and tracking application than simply using the basic background subtraction approach. For their implementation they build upon an existing algorithm called CAMshift [1], which is capable of tracking multiple moving objects in video streams. In this algorithm, an approximate background image of the monitored area is gradually built up by smoothing or blurring out the motion areas. This is done by using several recent images of the monitored area, and performing arithmetic averaging on them. One can then simply use this built up background image for detecting changes in the monitored area.

This approach applies to the first part of our problem, namely detecting and tracking people in the monitored area. We can build upon this approach as the implementation of it is quite simple yet flexible and adaptive. Moreover, it eliminates the need to worry about fluctuations in lighting and other unpredictable changes in the background that may corrupt the interpretation of the monitored area.

Gellert & Vintan [3], in their work *Person Movement Prediction Using Hidden Markov Models*, presents the use of hidden Markov models to anticipate the next movement of a given person. It is used inside an office building where it is possible to predict the next room given any room based on the history of rooms visited by a person moving within the building. The configuration of the model is optimised by evaluating movement sequences of real persons within an office building.

The concept of calculating the probability, and thereby making a prediction, of the next room given any room can be related to our scenario where we predict the next occupied section of the image given an occupant in any section. This information is used in the process of predicting where an occupant will exit the room.

Ashbrook & Starner [4] demonstrated how locations of significance can be incorporated into a predictive model in the paper *Using GPS to Learn Significant Locations and Predict*

[1] Continuously Adaptive Mean Shift.

Movement Across Multiple Users. They cluster location data from a GPS over an extended period of time and infer meaningful locations. These locations are then incorporated into Markov models where it is used to predict a person's future location given any current location. The idea of using heavily visited areas, that is *locations of significance*, to perform predictions is transferred to our scenario, where each area of the image holds a value reflecting the amount of activity within the area. This value plays an important role in the calculations of probabilities and thereby the final predictions.

1.4 Approach

Given the problems listed in section 1.2 and the related work listed in section 1.3, we propose a system consisting of three webcams, three Raspberry Pis, a single server and an android application. Each webcam is placed in a seperate room recording occupant activity. The webcam is connected to a Raspberry Pi performing analysis on the images received from the cameras. The analysis is made to detect occupants within the image and the information is sent and stored on the server. The android application can request information about the locations of each occupant and predictions about a single occupant. A prediction is essentially a guess about where an occupant will exit a room based on historical data. At any point during the presence of an occupant a prediction can be requested.
Due to the limited computation capabilities of the Raspberry Pis we have to perform benchmarks to find the appropriate programming language and framework for image analysis. For occupant detection, we gradually build up an approximate background image of the monitored area by smoothing and blurring out the areas with activity. The background image is compared to a given image where the difference displays where exactly activity has taken place.
We choose to follow a client-server approach where all communication from and to the devices and database goes through the server. The prediction logic is stored on the server as well and will be based on the approach laid out by Gellert & Vintan[3]. Instead of transitions between office rooms we split the image into sections and calculate probabilities of transitioning to other image sections. Our prediction approach is derived from the hidden Markov Models where we use historical data the past amount of activity in each image section in the calculations.
The design and implementation of the android application is assigned to the Kenyan team with whom we will come to mutual agreements about the data sent from the server and what prediction information can be requested. These decisions will be agreed upon during weekly meetings and regular email correspondences.

1.5 Report Structure

Section 2 analyses the problems presented in section 1.2 and presents and discusses various solutions. Based on the discussions, section 3 presents the final system design. Any implementation specific details about the system that are deemed not immediately obvious will be covered in section 4. Section 5 evaluates the results from tests of the final system. The whole project process including management, issues and collaboration with the Kenyans is covered in section 6. Finally, the system and project process will be concluded in section 8.

2 | Analysis

2.1 Image Analysis

There are numerous factors that must be considered when designing a surveillance system.

Position of the camera. Camera's type, position and angle may differ a lot, depending on the requirements of the surveillance application. In our scenario it was determined that the most optimal position of the camera would be the ceiling, making the camera point down to the monitored area. This would provide us with an increased field of view, and would make detection of multiple people walking side by side easier. However, due to resource limitations and difficulties we would face trying to position the camera in such way, decision was made to simply place the camera on a higher floor and point it down to the monitored area in approximately 50-60° angle.

Object Extraction. As we have already touched a bit in Related Work section, background subtraction plays an important role in detecting people and tracking their movement. There are multiple ways in dealing with background subtraction, but we have mainly covered two - simple background subtraction - where we use a static background image to detect difference in frames - and running average approach - where we gradually create an approximate background image to extract moving objects from the frame. The former approach can work rather poorly and inaccurately in dynamic environments, whereas the latter one deals with the same challenges in a better way, thus have been chosen for our design. This is further discussed in greater detail in Section 3.3.1.

Object Detection. To build up a complete and reliable surveillance system, one must be able to identify whether the detected object is a person, an animal or simply tree leaves moving because of wind. To deal with this, one can possibly apply face detection methods or object's shape recognition techniques. Since these are rather complicating areas in image processing, and our test scenarios mainly dealt with environments where the background is more stable, we did not consider object identification and simply interpreted all changes in the monitored area as human movement. Our object detection approach is explained in more detail in Section 3.3.2.

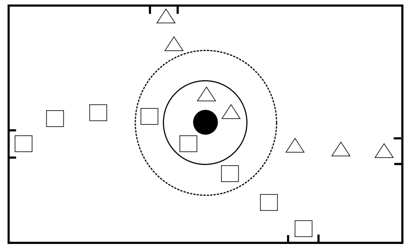


Figure 2.1: An illustration of the *k-nearest neighbors algorithm.*

Object Differentiation. Another important part of surveillance is being able to differentiation between multiple people crossing the monitored area. To deal with this challenge, we looked into two approaches, namely histogram approach - where we calculate individual histogram for every person and try to match it next time we detect people in the frame - and last coordinate approach - where we, instead of calculating person's histogram, calculate person's last coordinates in the monitored area. The main problem these both approaches share is that they are not very precise, however, the latter approach was easier to implement, thus was chosen for our design. Further comparison and discussion on object differentiation is provided in Section 3.3.3.

2.2 Prediction

In our scenario where occupants enter and exit the room constantly, typically within seconds or minutes of each other, it is somewhat difficult to find the immediate use of predicting an occupant's future actions. However, our partner team from Kenya expressed a need for such a system because they have had issues with thefts. Incorporating such a feature to their surveillance systems could warn a guard whenever suspicious activity might occur. A suspicious activity could involve a person moving too close to a given exit - possibly leading to a restricted area. In order to calculate a somewhat reliable prediction, we need to capture and store data about the behavior of previous occupants. When the data set grows, the prediction gradually becomes smarter and generally more precise. In order to satisfy the prediction requirement, various existing solutions have been considered, namely the *k-nearest neighbors algorithm* and an existing prediction model, the hidden Markov model. The two solutions will be covered briefly. The former assigns a given object to a group depending on the k nearest objects. To apply this to our scenario we imagined previously observed occupant positions being separated into groups dependent upon the chosen exit. Given any position we would analyze the k neighboring previous positions of distinct occupants and predict the exit to be that of the majority. Figure 2.1 depicts an example where the center dot is the current position and the squares and triangles are previous occupant positions who chose separate exits. The circle with the solid line resembles a situation where $k=3$ which indicates that the occupant at the center is most likely to take the same exit as the

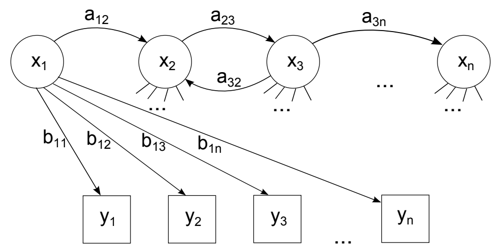


Figure 2.2: An illustration of the hidden Markov model.

occupants previously positioned at the triangle positions. The circle with the dashed line resembles a situation where k=5. This yields a different output where the occupant is most likely to choose the same exit as the occupants previously residing at the positions of the squares. The approach has proven to be inefficient on larger data sets [5], which can be averted by performing clever data reduction. This, however, deemed the approach out of the scope of this project.

Figure 2.2 depicts the different parameters a hidden Markov model works with. Each x is a state and each y is a possible observation, where a is a state transition probability and b is an output probability. When translating this to our scenario we need to define what exactly a state and an observation is and how to calculate the different probabilities. Since the ultimate goal is to predict what exit an occupant is going to take, a possible observation could be that an occupant exits a room at a given location. A state could depict a room area where a transition to another room area would have a state transition probability a. Given an occupant resides in any room area, b is the probability that the person will exit at a given location. Typically, the process is as follows; random probability values are assigned to a and b (maybe a ref to jahmm or something else). Gradually, these values are updated to reflect actual observed actions of occupants. Over time, the probabilities are modified and represent how the average occupant is moving given a current location, resulting in more accurate predictions - this process is commonly referred to as training. Even with the inclusion of a training procedure, the predictions will be incredibly unreliable in the beginning until a large enough data set has been collected. (DISCUSS, this might be wrong) Furthermore, imagine a scenario where an even number of occupants move from one exit to another in both directions, evenly split. At any location along the path the probabilities to each exit remain equal, since equally many previous occupants have taken each exit given the current location. In an attempt to partially avoid these issues, or at least produce some more reliable results with lesser data sets, we wanted to establish rule sets and integrate those into the calculations. Thus, we chose to design our own prediction model, heavily inspired by the hidden Markov model, since the original interpretation of states and probabilities as depicted in figure 2.2 is maintained. The custom model is explained in detail in Section 3.5.2.

Figure 2.2 depicts the different parameters a hidden Markov model works with. Each x is a state and each y is a possible observation, where a is a state transition probability and b is an output probability. When translating this to our scenario we need to define what

exactly a state and an observation is and how to calculate the different probabilities. Since the ultimate goal is to predict what exit an occupant is going to take, a possible observation could be that an occupant exits a room at a given location. A state could depict a room area where a transition to another room area would have a state transition probability a. Given an occupant resides in any room area, b is the probability that the person will exit at a given location. Typically, the process is as follows; random probability values are assigned to a and b (maybe a ref to jahmm or something else). Gradually, these values are updated to reflect actual observed actions of occupants. Over time, the probabilities are modified and represent how the average occupant is moving given a current location, resulting in more accurate predictions - this process is commonly referred to as training. Even with the inclusion of a training procedure, the predictions will be incredibly unreliable in the beginning until a large enough dataset has been collected. (DISCUSS, this might be wrong) Furthermore, imagine a scenario where an even number of occupants move from one exit to another in both directions, evenly split. At any location along the path the probabilities to each exit remain equal, since equally many previous occupants have taken each exit given the current location. In an attempt to partially avoid these issues, or at least produce some more reliable results with lesser data sets, we wanted to establish rule sets and integrate those into the calculations. Thus, we chose to design our own prediction model, heavily inspired by the hidden Markov model, since the original interpretation of states and probabilities as depicted in figure 2.2 is maintained. The custom model is explained in detail in section 3.5.2.

2.3 Communication Model

As there is several devices which have to communicate between each other, we have considered two different solutions:

1. Client-Server, and
2. Peer-To-Peer (P2P)

To decide which solution suited this project the most, each possible solution has been considered from the following perspectives.

Storage. When considering storage we consider how easy it will be to store and retrieve data.

Reliability. When considering reliability we consider how reliable the communication will be between the cameras and the android devices.

Scalability. When considering scalability we consider how the system will perform with different amount of devices connected.

Cost. When considering cost we will consider how much it cost mostly in development time due to complexity and to a lesser degree prices of such implementation.

2.3.1 Client-Server

The Client-Server model consists of two specialized applications: A client and a server. The server application will provide some shared resources. The client can initiate contact to a server and use the provided resources.

The considered advantages and disadvantages of the Client-Server model are as follows:

Storage. Storing data at a central point reduces data flow on the network and availability to old data for all devices. It also makes it easy to backup data.

Reliability. Performs well if there is a heavy load on the network. A server is build to handle a lot of connections and each peer will only have to manage its own connection to the server. If a server is heavily burdened an additional server can be set up and share the load. Also it is only the server that has to be available for all devices which reduces potential problems with different network configurations. A drawback of using a server is that it introduces single point of failure risking that if the server fails the whole system will be down.

Scalability. Performs well with both few and many connections. If the server is overloaded an additional server can be set up increasing the capacity. This model does not overload the network with redundant data as the devices only have to send the data once.

Cost. As server and client is more specialized it is easier to develop and maintain each separate application. Also with this split the server can be updated without third party users have to update.

2.3.2 P2P

The P2P model is a decentralized and distributed network model. It consists of several devices called "peers". Each peer can acts as both server and client which share or store some resources.

The considered advantages and disadvantages of the P2P model are as follows:

Storage. Distributed data storage is possible, but it is very complex and overall space required will be very high due to redundancy.

Reliability. P2P is very reliable as there is no single point of failure. If a peer drops out the data can be provided from another peer. If a Raspberry PI disconnects its web-camera will of course not be available, but the system will continue working. A potential problem is different firewall configurations which might result in the peer not being available for other peers at all.

Scalability. P2P does not scale very good in a system like this as the web-camera peers will either have to send to a lot of peers or the peers will get the data delayed. If a peer requests some old data it can be distributed from another peer reducing traffic to the raspberry Pis.

Cost. Only one application will be developed and maintained, however, this single application will be very complex.

2.3.3 Comparison

Storage. Central storage has a lot of advantages compared to distributed. It consumes less overall space and it creates less network traffic. As most data in this system is used for live tracking a distributed data model will lead to a huge amount of redundant data which will only be used once. For these reasons central storage is preferable.

Reliability. Both solutions can be considered reliable, however a server rarely brakes down and with a lot of firewalls all over the place the chance that a peer will not be available for other peers will probably be higher than a server going down.

Scalability. In a system like this a Client-Server approach will scale a lot better than the P2P since it will require a lot of clients to force a second server and the P2P model will result in either overload of the Raspberry PIs or delayed data deliveries.

Cost. P2P will be incredible complex to develop compared to the Client-Server. Also as the system will be developed by two teams located at different places, a split will make it easier to split up task, without both teams having to update every time the other one finds a bug.

All in all the Client-Server model is much more suited for this type of system than the P2P.

3 | Design

3.1 System Overview

Figure 3.1 displays the different components in the system and how they relate and communicate with each other.

As we can see, the system primarily consists of 4 components:

1. Web cameras,
2. Raspberry Pi computers,
3. The server,
4. and the Android application.

All of these components are discussed further on in the respective chapters.

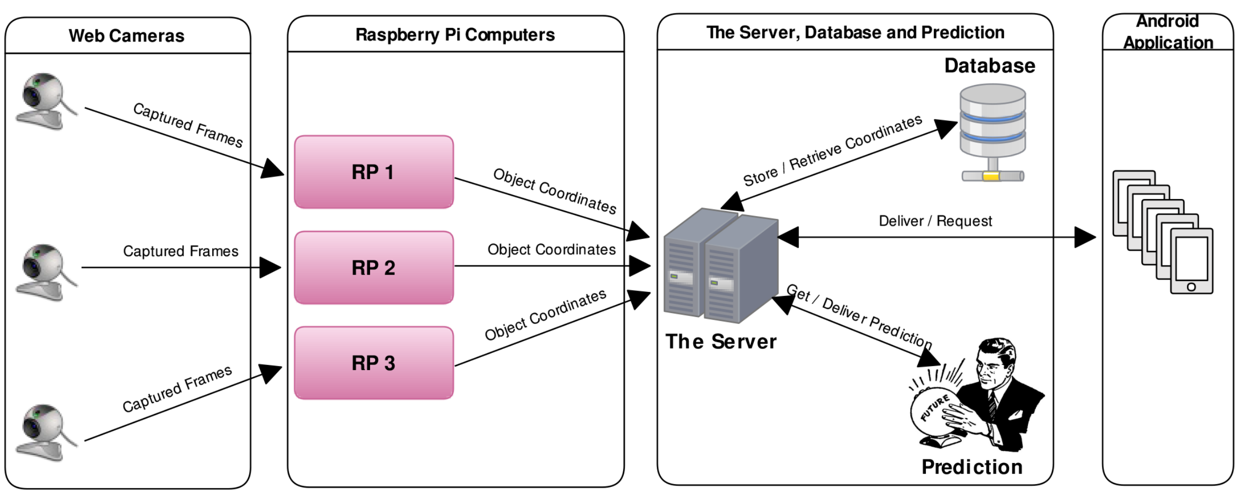


Figure 3.1: Occupancy Analyzer system overview

3.2 Web Cameras

The purpose of the web cameras is simply to surveillance the area they have been placed in and forward the captured frames to Raspberry Pi computers for further processing, as shown in Figure 3.1. The cameras can be placed in a room, corridor, atrium or any other similar place in or outside of the building, where the services offered are required. Cameras can be placed either directly above the observed area, as shown in Figure 3.2 or in the corner of it, as illustrated in Figure 3.3. Naturally, a camera placed above the observed area would give better results, since this increases its field of view and this makes it easier to correctly detect and distinguish between multiple people walking side by side. Furthermore, for the best results one must also take many different factors into account, such as the distance between the camera and monitored area, environmental conditions of the area the camera is placed in, and lighting conditions.

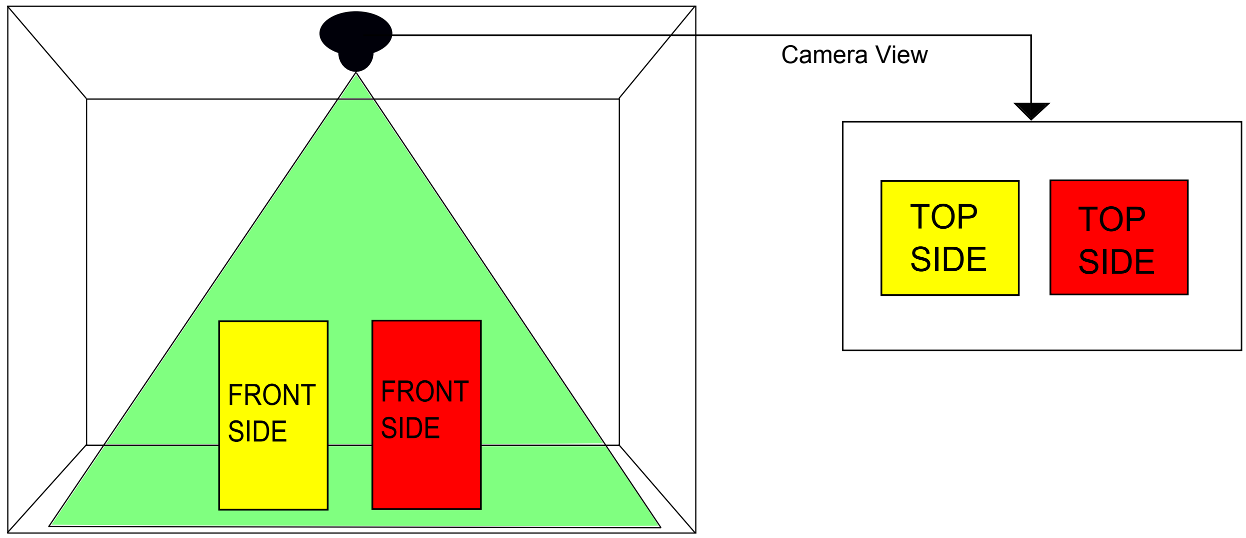


Figure 3.2: Camera top placement

3.3 Raspberry Pi Computers

The computers each have at least one web camera attached and are responsible for processing the frames captured by the cameras. The main goal of processing these frames is to detect people in the monitored area and determine their position in that area. There are multiple challenges in object detection, as well as various concepts and techniques that can be used to overcome them. We discuss these in the following sections.

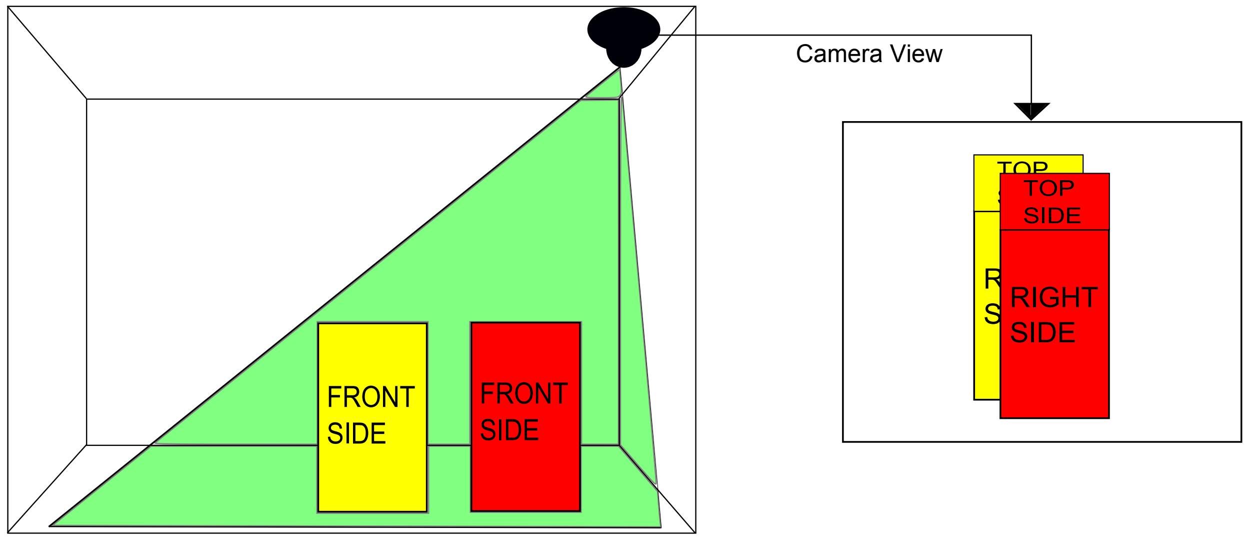


Figure 3.3: Camera corner placement

3.3.1 Object Extraction

To detect and extract objects, or in our case, people and their movement, we need to apply several motion detection techniques on the frames we are receiving from the web camera. First of all, to detect changes in some monitored area, we naturally need to have at minimum two images, which we must compare to see what changes occurred. We will try to look into two different approaches in doing this, a simple one, and one that is a bit more complicated and sophisticated, but much more adaptive and flexible.

3.3.1.1 Simple Background Subtraction Approach

A simple approach - called Background Subtraction - would be to have a static background image of observed area that was taken prior to the analysis and did not have any people in it. Then, one can simply detect changes and movement in the area by subtracting the static background image from every newly taken image of the monitored area [6]. The difference between the two images would then allow us to see if any changes happened, since after subtraction the resulting image would either be almost totally black (Figure 3.4) - meaning no one walked passed the observed area - or the image would have some resulting bright contours of detected differences (Figure 3.5).

After doing some research and experimenting with background subtraction technique, one will quickly discover that there are multiple weaknesses to it.

- First of all, if the initial background image is always static and never changes, this technique will fail in environments where lighting is dynamic. This is perfectly illus-

Figure 3.4: Background Subtraction with no background changes

Figure 3.5: Background Subtraction with background change

trated in Figure 3.6. We can see that the lighting is much darker in the second image, possibly because the light was turned off in the monitored room, thus after subtracting our static background image from this image, the resulting image is simply a lighter version of the two images, and not the intended black image. The reason why this happens is because the original brighter background image has a much higher intensity, thus the average value of it's pixels are higher than the pixel values of the newly taken darker image. Naturally, this is a big problem, because now even if a person moves through the monitored area (Figure 3.7), he or she will not be as easily extractable as in Figure 3.5.

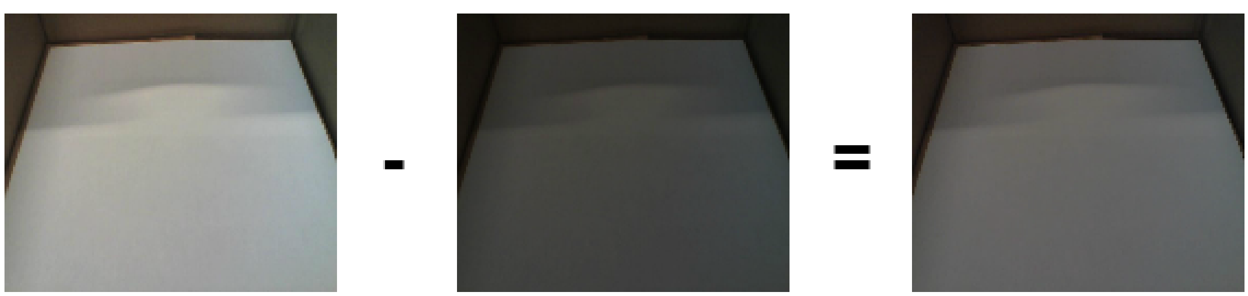


Figure 3.6: Background Subtraction with background lighting change

- Another problem with background subtraction approach surfaces when a camera is placed inside an area which has objects that constantly change their original position

Figure 3.7: Background Subtraction with background lighting change and lego figure appearing

(chairs, tables, appliances, etc.) by being moved, even small changes in object's location will spoil the resulting image after subtraction. As we can see in Figure 3.8, the object is displayed twice in the resulting image, even though we were not even interested in it, making it much harder to detect actual people moving in the area. From now on, the resulting image after subtraction will always be corrupt unless the object is placed back in its original position.

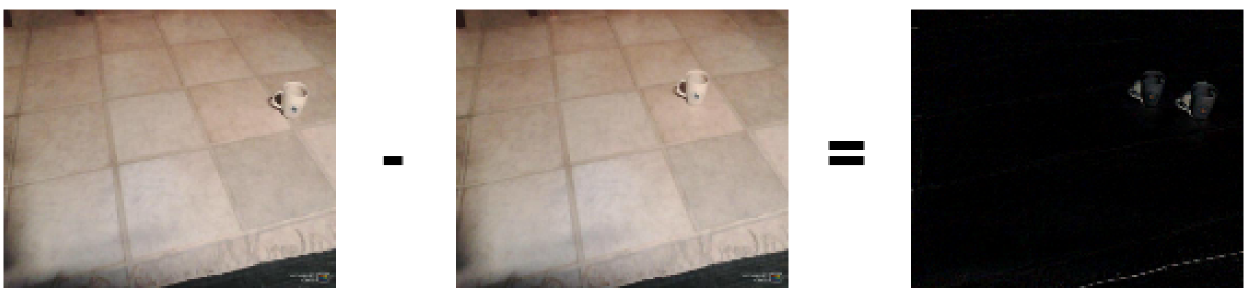
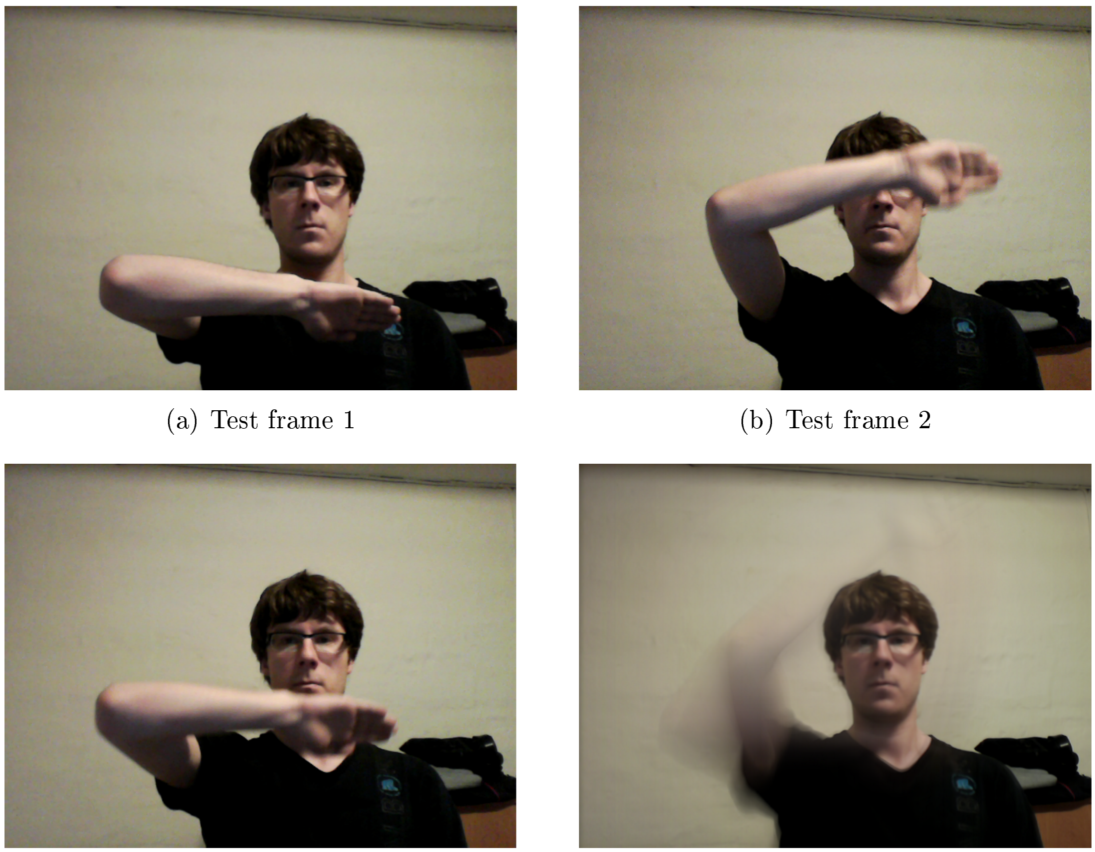


Figure 3.8: Background Subtraction with object changing it's location

In conclusion, we can see that simple background subtraction approach can work well in static environments, however it falls short in dynamic spaces. Naturally, these mentioned drawbacks of background subtraction approach need to be handled for object detection to work well, which creates additional challenges when implementing the system.

3.3.1.2 Running Average Approach

A much better approach for movement detection is using a running average method. In this technique we do not need to rely on a static background image of the monitored area taken prior to analysis. Instead, we try to find a new "approximate" background image by interpreting any changes in the background as noise and blurring them out. This is accomplished by taking a training sequence of multiple previously captured frames and performing arithmetic averaging on that sequence [7]. This exact approach is illustrated in Figure 3.9. As we can see, hand motion moving up and down gets blurred out when applying running average method, thus producing an approximate background image (d) that can be used as a basis for subtraction of the test frames. This method of object extraction works very well and has a lot of flexibility. It can adapt to environmental changes in the monitored area, thus eliminating most of the weaknesses that the background subtraction approach has. For these reasons, the running average approach was chosen for our design.



(a) Test frame 1

(b) Test frame 2

(c) Test frame 3

(d) Running Average frame

Figure 3.9: Example of Running Average technique

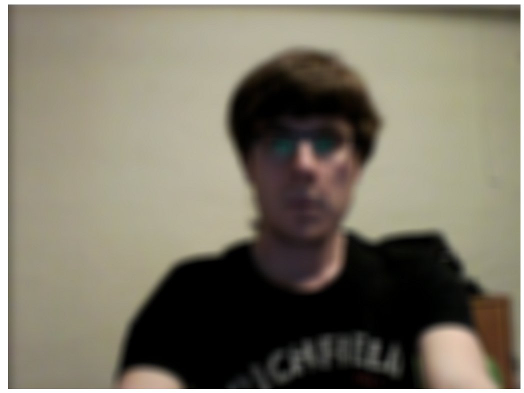
3.3.2 Object Detection

Now that we can extract objects using running average technique, we need to be able to actually find them in the resulting image we get after we perform subtraction. For this, we need to apply several key techniques in image processing.

1. ***Noise Removal.*** After we capture the initial frame of the monitored area, it will often

contain noise and small details that we are not interested in. To deal with this, we must first apply blur or smoothing filter. In blurring technique we calculate weighted averages of areas of pixels in a source image by passing through it [8], which helps to reduce image noise and detail, as shown in Figure 3.10.

(a) Original frame



(b) Frame after blur is applied

Figure 3.10: Blur example

2. ***Background Subtraction.*** After we remove the initial noise, we can perform subtraction using running average approach (described in Section 3.3.1.2). After subtraction we will either get a nearly totally black image, meaning no motion occurred, or an image where some colors stand out, meaning some motion has occurred.

3. ***Channel Reduction.*** In either case, for further processing of the taken frame, we can choose to convert it to a grayscale image. The reason for this is that the original RGB image we get has three channels, while grayscale image has only one [9], thus it is easier to work with. This procedure is illustrated in Figures 3.11(a), 3.11(b) and 3.11(c).

4. ***Thresholding.*** For further processing of the image, we apply threshold technique, which converts the image to black and white and removes some more unwanted details and noise [10]. After threshold is applied we get the image shown in Figure 3.11(d).

5. ***Simplification.*** Moreover we want to expand the interesting parts of the image and contract smaller pieces, which can be considered noise and managed to slip through, even after we performed thresholding. To do so, we use two fundamental operations in morphological image processing, that is, dilation and erosion. Dilation allows us to expand the shapes contained in the image [11], whereas erosion simply shrinks shapes [12], so that bright regions surrounded by dark regions shrink in size, and dark regions surrounded by bright regions grow in size. When we apply dilation and erosion we get the image shown in Figure 3.11(e).

6. ***Bounding Box.*** Now, to project the detected area onto the original frame, we simply use bounding box technique, which gives us the coordinates of the rectangular border

that fully covers the extracted white silhouette [13] that we got in 3.11(e). Then we use these coordinates to draw a simple rectangle around the extracted object, as well as mark it's middle position by a red circle, as illustrated in Figure 3.11(f).

In conclusion, by applying the steps discussed in this section, we can fairly accurately detect people and their movement in the monitored area.

3.3.3 Object Differentiation

There will naturally be cases when multiple people will walk through the monitored area and will be captured by the cameras, therefore we must have a way to differentiate between them. This task becomes rather difficult if people are very close to each other, since they will simply be interpreted as one person. However, as long as people are far enough from each other, the task becomes significantly easier. There are multiple ways of differentiating between objects.

One of them is to simply look at the objects histogram, which gives a graphical representation of the intensity distribution of pixels [14]. Since people are usually dressed in different color clothes, we can simply calculate a histogram for every detected person and remember it. Now, every time we receive a new frame and detect a person in it, we go through our previously saved histograms and check whether any of them are the same as our newly detect persons histogram. If there is such histogram, we interpret the person we detected in our new frame as the same person we detected in the last, otherwise, we conclude that we have not detected this person before, thus save his histogram for future reference. The biggest weakness of this approach is that a person's clothes might have different colors from the front and back. Therefore, his histogram calculated while he is facing the camera might be rather different than the histogram of when his back was towards the camera. For this reason, if the person decides to turn around midway, he might be interpreted as a new person - never seen before by the camera - when in fact his frontal or back histogram was already saved.

Another approach of differentiating between multiple people, and in fact the approach we used in our design, is to simply use the whole frame as a coordinate system and remember the last coordinate of every single detected person. Now, similarly to histogram approach, whenever we detect a new person in the frame, we simply look throughout previously saved coordinates, and if we find that this new person's coordinates is relatively close to some previously saved person's coordinates, we simply interpret him as the same person we detected a second or or few seconds ago, otherwise we see him as a new person. Naturally, we must regularly clear our previously saved coordinates, so that newly arrived person would not simply, by taking a similar path, be interpreted as a person who is no longer in the monitored area. This approach of differentiating between people is is illustrated in Figure 3.12.

3.4 The Server

TODO

The server is the central part of the whole system. It is here the Raspberry PIs send all the gathered data, where data are stored and structured, and where Android devices can fetch information about rooms, people, and predictions on which exits a person is most likely to use.

All sorts of devices can be used to capture and process data. To simplify the process and to allow access for all types of devices, the server provides an API which allows the senders to focus on data capturing and processing.

As with the devices providing the data for the server, the devices which requests data from the server can be of different types. By providing an API these devices can focus on how to use and present the data, and let the server focus on how the data is fetched. This ensures that all data is available for everyone and that each request has a minimum of effect on other devices access to the data.

When data is received the server will store it in a database, making it easier to get relevant data for both positioning of occupants and predictions. An outline of the database is seen in figure 3.13

3.5 Prediction

3.5.1 Gathering statistical data

In addition to storing data about each occupant we also store data about how occupied each section of the recorded image is. The process is simple; we split the image of a room into sections, referred to as cells, and each time an occupant enters a cell the stored activity for the given cell is increased. Each room has a different and independent set of cells. This data tells us which sections of the room are most occupied. We use the activity data to perform predictions about an occupant's future actions using a custom prediction model.

3.5.2 Custom prediction model based on HMM

We have chosen to build our own prediction model heavily inspired by the hidden Markov model. Each cell in the image is a state where the state transition probability is the likelihood of an occupant moving to an adjacent cell and the output probability is the likelihood of an occupant going to an exit given any current cell. Unlike a regular hidden Markov model, we do not store probability values individually for each state, but rather do the necessary calculations each time a prediction is requested by using the stored activity values of each cell. Additionally, our custom model allows us to take several custom factors into account during calculations. These factors work as a rule set for likely or unlikely occupant actions:

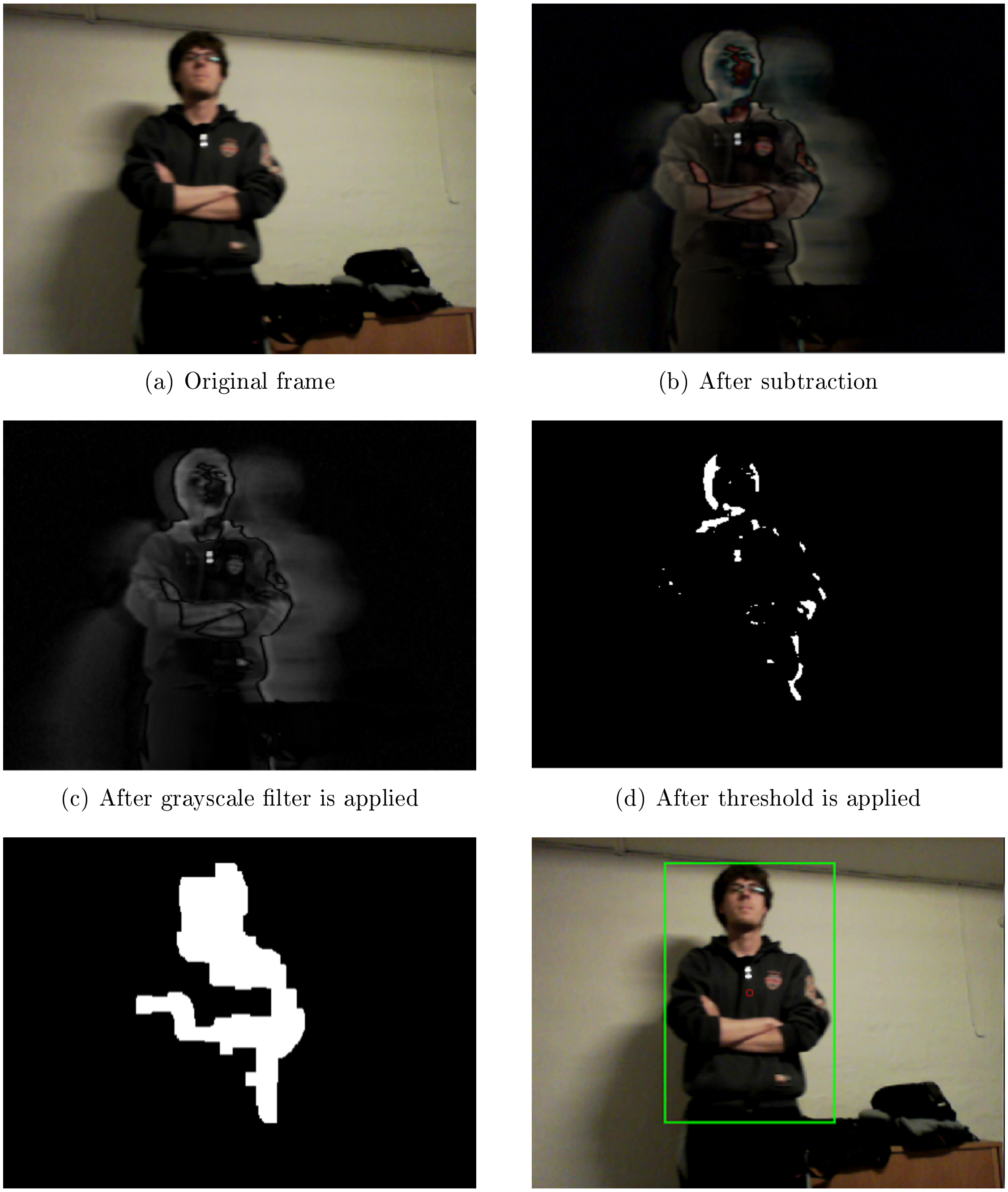
- An occupant entering a room from a given exit is less likely to exit the room at the given exit.
- An occupant is more likely to continue moving in his general direction and least likely to return to his previously visited cell.
- An occupant is more likely to move to the adjacent cell with the highest amount of previous activity.
- The likelihood of an occupant exiting at a given exit (unless it also serves as the occupant's entrance) is inversely proportional to the direct distance to the given exit, producing a magnet-like effect.

These factors have an influence on a final value of a cell or exit, which is used when calculating each probability. The sum of the probabilities of an occupant moving to each individual exit given a current cell is 100. The cell with the highest probability will be the predicted cell.

Figure 3.14 shows the custom model translated to a scenario. Each cell contains an identification number, an activity count and a value denoting whether or not the cell is an exit cell. The occupant entered the room at exit y_3 and the current state is x_{14}. a_{14-13} denotes the state transition probability of the occupant moving from state x_{14} to state x_{13}. b_{14-1} denotes the output probability of the occupant ultimately choosing exit y_1. Taking our custom rule set into account, the occupant is least likely to exit at y_3 and most likely to continue to x_{13}. Moving to x_9 would increase the probability of the occupant exiting at y_1 (b_{9-1}) significantly.

3.6 Android Application

TODO



(a) Original frame

(b) After subtraction

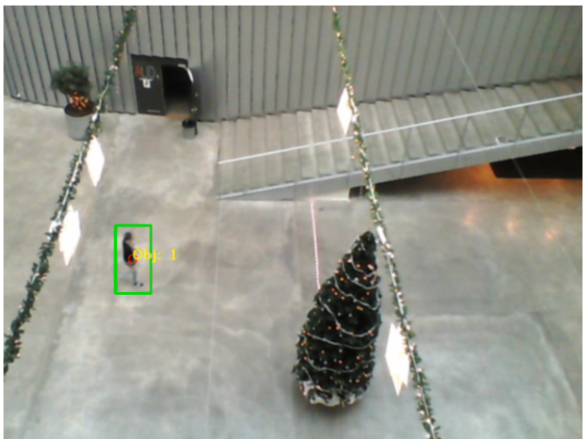
(c) After grayscale filter is applied

(d) After threshold is applied

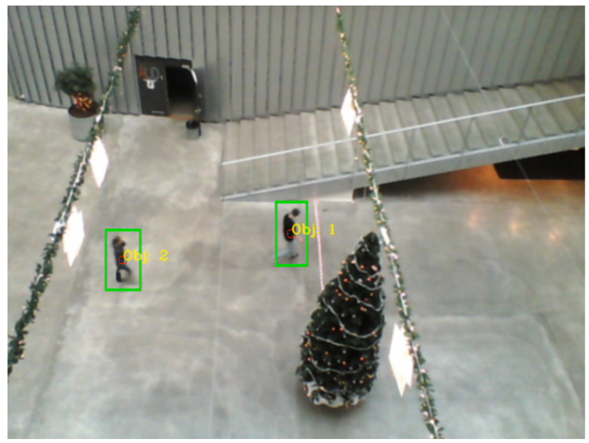
(e) After dilation and erosion

(f) After bounding box is drawn

Figure 3.11: Object detection example



(a) One person enters the monitored area



(b) Second person enters the monitored area

(c) Two people continue their walk through the monitored area

(d) First person leaves the monitored area

Figure 3.12: Object differentiation example in the monitored area

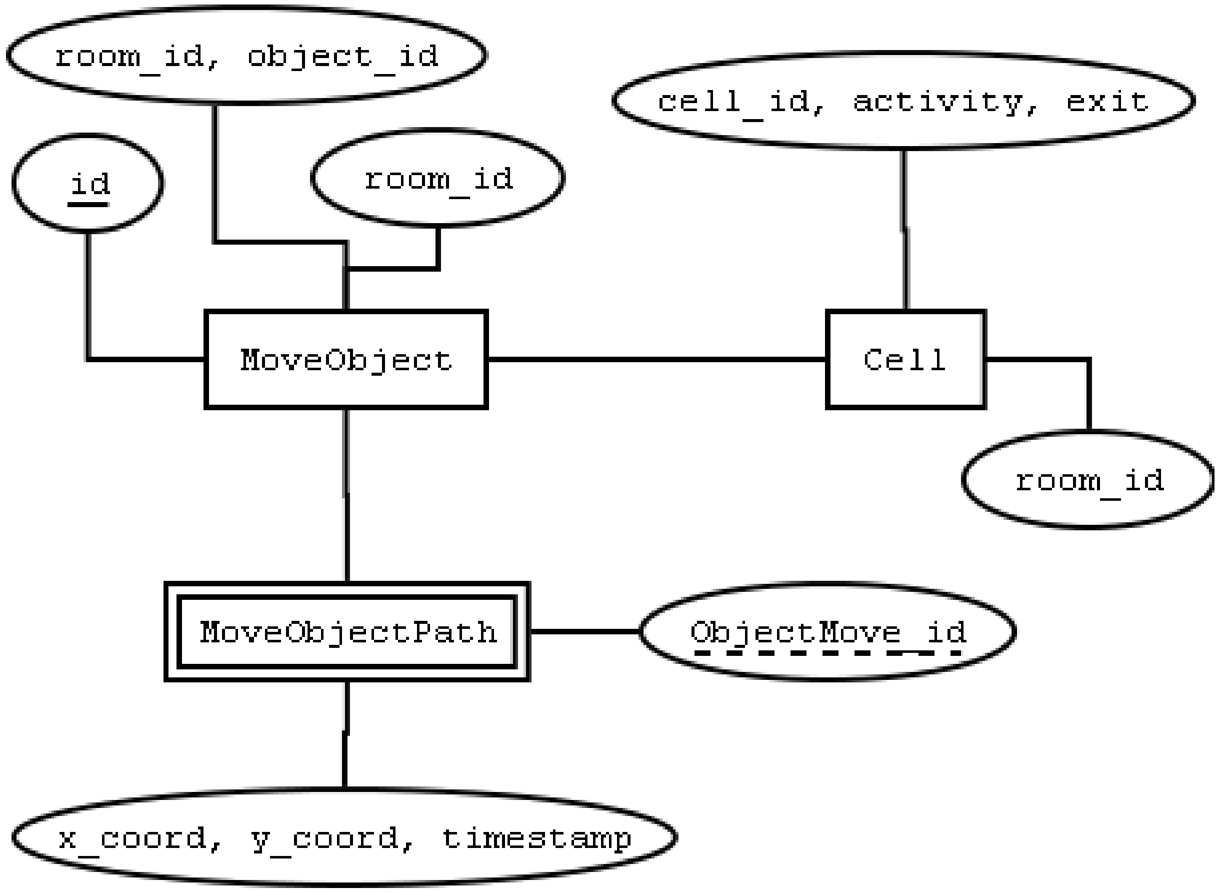


Figure 3.13: Database overview

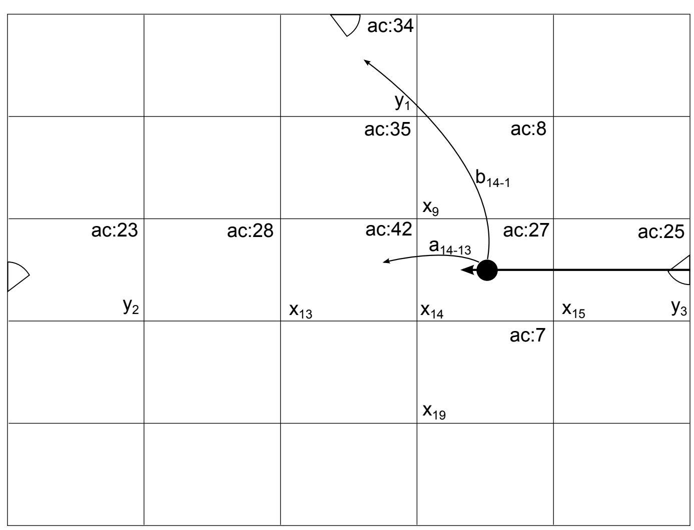


Figure 3.14: Applying the custom model to a scenario.

4 | Implementation

4.1 Image Processing

4.1.1 Library

As non of the group members had any previous experience with image processing, it was quite unrealistic to try to implement all of the image processing techniques needed for this project, and at the same time make them optimized enough, so it could work well and fast on Raspberry Pi computers with limited resources. For this reason the group decided to use an existing image processing library to ease the implementation process. After some research and experiments, we decided go with one of the most popular and well documented libraries called OpenCV [1]. OpenCV is an open source computer vision and machine learning software library that has more than 2500 optimized algorithms for image processing. Furthermore, it provides interfaces to multiple popular programming languages, including C++, C, Python and Java.

4.1.2 Programming Language

For determining which programming language to use for image processing on Raspberry Pi computers, skill and preference document (Appendix B.2) was created, where both, ITU and Strathmore University, teams indicated their skill and preference for various programming languages. The two languages that stood out the most were Java and Python, thus to choose one of them, we decided benchmark [2] them in order to compare their performance. The results are presented in the next section.

[1] OpenCV homepage: `http://opencv.org/`.

[2] Code used for benchmarks: Java - `http://itu.dk/people/tmis/javatest/`, Python - `http://itu.dk/people/tmis/pytest/`.

4.1.2.1 Performance Comparison Between Java and Python

Since the project dealt with a real-time vision application that had to process large amount of frames, we were interested in how fast Java and Python can perform different image processing algorithms. Benchmarks were performed on a laptop [3] and a Raspberry Pi computer [4] to give some perspective on how much faster a modern laptop is compared to a Raspberry Pi computer.

- At first we tried to benchmark how fast can Java and Python perform a simple matrix multiplication of two 300×300 size matrices. The results are illustrated in Figure 4.1. We can clearly see that Java was way faster than Python in this benchmark. It took Java less than half of a second to perform the multiplication of two 300×300 size matrices on a laptop, whereas it took more than 3 seconds to do the same in Python (Figure 4.1(a)). Multiplication on a Raspberry Pi computer (Figure 4.1(b)) was naturally much slower than on a laptop. Java was again much faster than it's counterpart by dealing with the task in less than 23 seconds, whereas Python was very close to hitting 2 minute mark to accomplish the same task.

  In conclusion, as it was expected Java convincingly won the first benchmark.

- The second benchmark involved testing how fast Java and Python can perform different image processing algorithms. For this benchmark we used the OpenCV library that we introduced earlier. OpenCV has Java and Python interfaces and all the computations are performed on the native level [5], hence we have some overhead that is equal to a cost of one or several API calls. Consequently, the purpose of this benchmark was to see, which language - Java or Python - has less overhead and can perform API calls faster.

  As we can see in Figure 4.2, the difference between Java and Python in this benchmark was rather small on both, laptop 4.2(a) and Raspberry Pi computer 4.2(b). This was rather surprising at first, since Java had a big edge in the first benchmark. One can speculate that Python's interface to OpenCV library has been implemented in a more efficient and optimized way than Java's, thus the outcome. In general, the difference was only in terms of a few milliseconds, however, it was a good enough reason for us to choose Python's interface to OpenCV library for image processing part in this project.

Make a small conclusion of this, the conclusion should not be in the

[3] Toshiba Satellite L855 Laptop (Intel Core i7-3630QM 2.40 GHZ, 4GB DDR3 1600MHz, Radeon HD7670M 2GB, Windows 7 OS).

[4] Raspberry Pi Computer (ARM1176JZF-S 700MHz, 512 MB memory, Broadcom VideoCore IV graphics, Linux Raspbian OS).

[5] In C++.

4.2 The Server

4.2.1 Communication

To reduce possible platform limitations the server communicates with all other devices through the TCP protocol, structuring the data in a JSON string.

The server works with two different structures:

1. Commands, and
2. Data Deliveries.

4.2.2 Commands

The command structure is used by the devices consuming the server's resources. The structure is very simple as it consists of the commands name which it wants to invoke and a sequence of parameters which each consists of the parameters name and value.

A generic example of a command can be seen in figure 4.3 for comparison the command "setExit" is shown in figure 4.4

In figure 4.4 "theCommandName" have been substituted with "setExit" and the parameters have respectively been substituted with "room_id" and "4ef9ad6e-5da3-4a2e-aa12-b32f70f3900e", "x_coord" and "5", and so forth.

There are x commands in total:

getRoomStatus

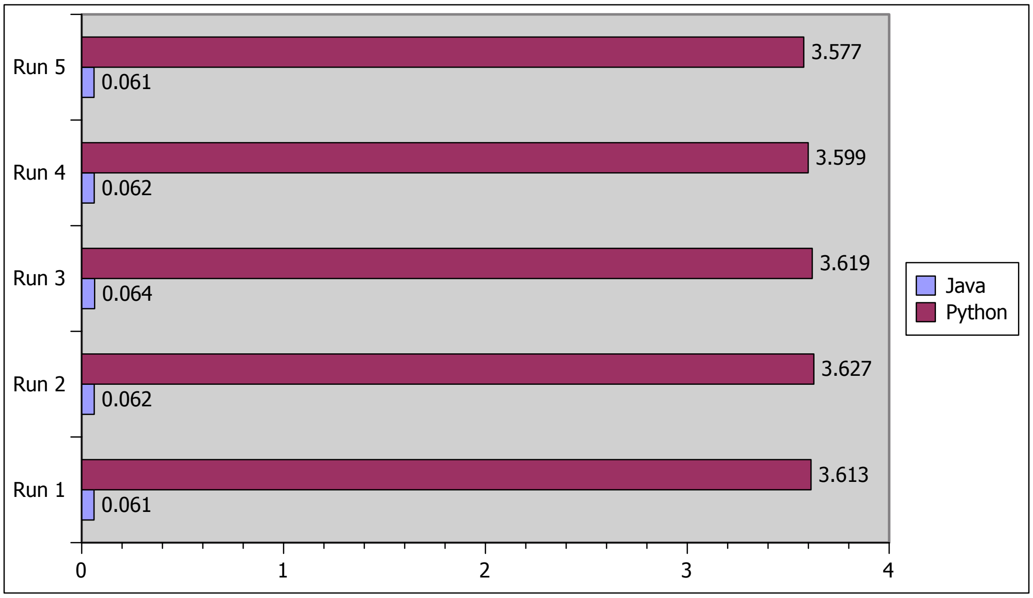
getRoomsAvailable

checkPredictions

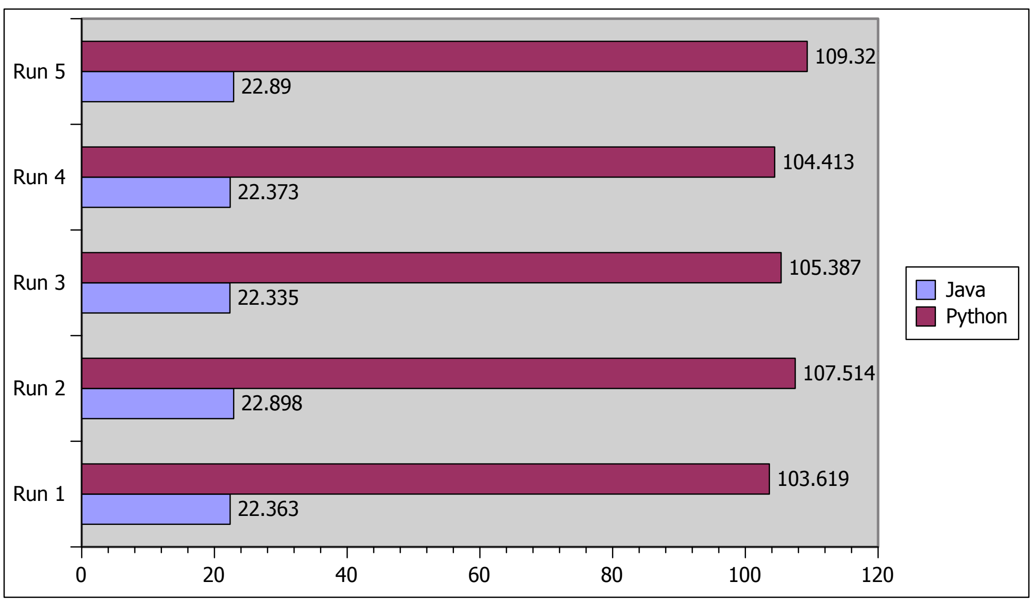
checkProbabilitiesToExits

setExit

4.2.3 Storage

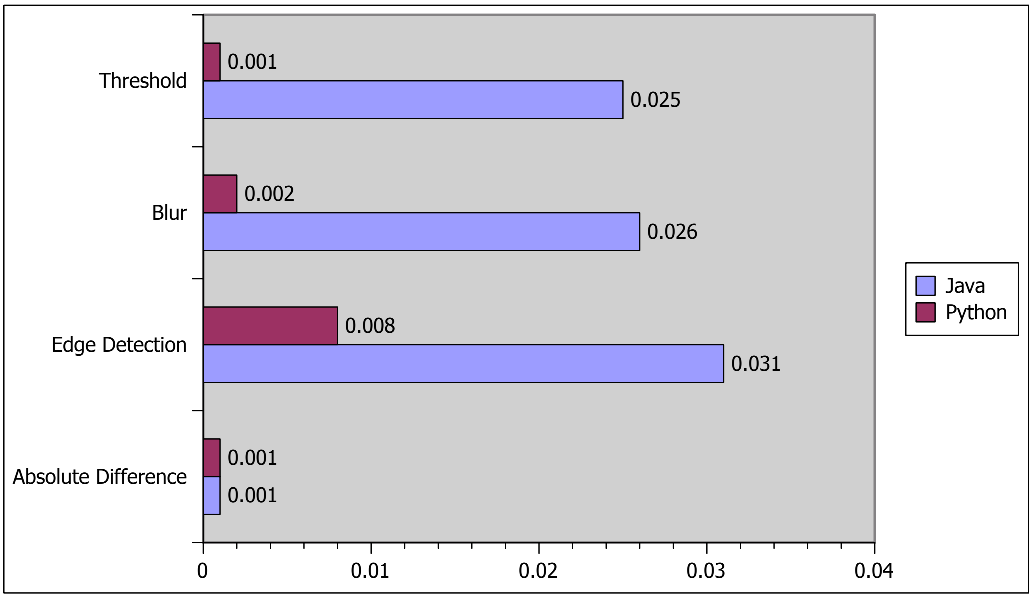


(a) Time (in seconds) needed to multiply two 300×300 size matrices in Java and Python on a laptop

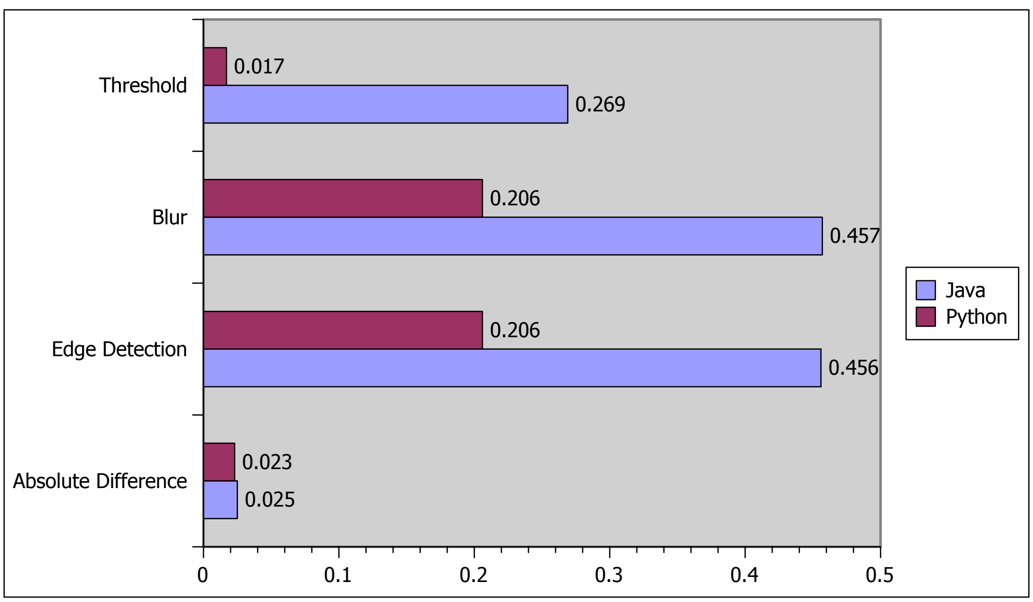


(b) Time (in seconds) needed to multiply two 300×300 size matrices in Java and Python on a Raspberry Pi computer

Figure 4.1: Benchmark of matrix multiplication of two 300×300 size matrices



(a) Time (in seconds) needed to process images in Java and Python on a laptop



(b) Time (in seconds) needed to process images in Java and Python on a Raspberry Pi computer

Figure 4.2: Benchmark of image processing

```
{"command":"theCommandName", "param": [{"pame": "paramName1", "value":"paramValue1"}, {"name": "paramName2", "value":"paramValue2"}, {"name": "paramName3", "value":"paramValue3"}, {"name": "paramName4", "value":"paramValue4"}]}
```

Figure 4.3: Generic JSON command example

```
{"command":"setExit", "param": [{"name": "room_id", "value":"4ef9ad6e-5da3-4a2e-aa12-b32f70f3900e"}, {"name": "x_coord", "value":"5"}, {"name": "y_coord", "value":"470"}, {"name": "is_exit", "value":"true"}]}
```

Figure 4.4: setExit JSON command

5 | Evaluation

5.1 Verification

5.2 Benchmarks

6 Collaboration

This chapter deals with the collaboration and process in the project. First we introduce some theory on project management and global collaboration. In the next step we report our project work: We introduce the project team, the initial situation and present the chosen project management approach, methods and tools. In addition we evaluate the collaboration tools we used in the project. Finally, we cover the issues we had to face during the project. These include progress issues, issues with the used project management techniques and collaboration issues. We analyze the collaboration issues based on the previous introduced theory. Furthermore we reflect on the process and collaboration and explain our learning outcomes.

6.1 Introduction

Test

6.1.1 Project management

There are several approaches how you can perform project management for software development projects. It is often distinguished between the traditional approach and the agile approach in project management.

The traditional approach in project management is dividing the project into distinguished phases, which are planned in advance. The phases are run sequentially. Theoretically, once a phase is complete it will not be revisited. This structure is also known as waterfall model in software development. The traditional approach plans the project in the beginning. All requirements and ressources are collected and planned in the beginning of the project to build up a project plan, which containes a detailed project tasks and milestones. These pre-planned project tasks are then worked off during the project. The project manager can keep track and report the progress of the project by comparing planned and actual progress.

The problem with this traditional approach is that it is not very flexible for unpredictable events. Furthermore, clients are often unable to state all requirements in the beginning. A change of a requirement or the adding of additional requirements during the project can have a major impact on the project plan and is often embedded in some kind of change management process.

The agile approach is in general more flexible than the traditional approach, especially when it comes to requirements and unpredictable events. The idea of an agile approach is to have many iterative planning and development cycles. The approach supports quick results, which are constantly improved by involve the client actively in the project progress.

An agile approach does not intent to have a completely pre-planned process like in the traditional approach. In software development projects an agile approach is preferred due to its advantages. A popular agile project management framework is Scrum.

An agile approach requires a strong collaboration between the project team members and healthy communication. It is also essential, that the project team members have the needed competencies and that the users and executives support the project .

Add reference

6.1.2 Global collaboration

Globally-distributed projects are becoming more popular for large software systems. The advantages of globally-distributed projects are that the

TODO

In the paper "Managing cross-cultural communication in multicultural construction project teams: The case of Kenya and UK" from E.G. Ochieng and A.D.F. Price

- Trust
- Individualism
- Power distance

6.2 Background information of the Project Team

The project team consists of two groups of students. One group is from Strathmore University located in Nairobi (Kenya). The other group is from the IT University (ITU) in Copenhagen (Denmark). The project team agreed on to name the two groups "Team Kenya" for the student group from Strathmore University and "Team ITU" for the student group from ITU. This helped to address each group in meeting reports, emails and conversations.

In the following paragraphs the background information for this project from both groups are introduced. The information explains the motivation and contribution of each group in the project and is used in arguments for why certain decisions were made and issues arise.

6.2.1 Team ITU

Team ITU started with four members, which are all in the Masters programme "Software Development and Technology (Software Engineering)" of the ITU[1]. The members are from three different countries: Lithuania, Germany and Denmark[2]. The communication language within Team ITU is English, which is not the mother tongue of any of the members. Two of the members of Team ITU already collected some negative experience with previous global collaboration projects and were not very interested on a global collaboration. They experience that the other groups in their previous global collaboration projects did not put much effort into the projects, so that they and their local team had all the workload. The members of Team ITU met each other the first time on the 27.8.2013.

The students of Team ITU have to complete the project under the course "Global Software Development Project", which is mandatory for their masters programme. The requirements and deadlines for the project are given in the course base from ITU (reference) and by the advisor of the project. The course is rated with 15 ECTS points, which corresponds to approximately 20 hours per week per student. The students of Team ITU have to hand in this report as a mandatory requirement.

Link to the course base

As the course for the Team ITU started in late August and the project team and topic was already known, Team ITU already started with the project work before Team Kenya. Team ITU and their advisor did not know when they would get the contact details from the student group in Kenya, so the advisor recommended already initializing the project and doing some research and thoughts on the project.

At the beginning Team ITU had received a different project topic by the advisor. The topic of the first project was "Vector Shooter"[3]. Team ITU put some thoughts into the project and spent time on defining the program, which they wanted to develop[4].

After three weeks the advisor had to change the topic of the project due to the collaboration with Strathmore University. Only one student from the Strathmore University was interested on the project "Vector Shooter". Team ITU could have spent time and effort in convincing the other students from Strathmore University to do the project "Vector Shooter". Despite

[1] In the beginning there were five members, but one left the project after two weeks, because he changed to another project and project team. This did not have an impact on the work as it was in the early stage of the project.

[2] Although there are minor differences between the nationalities, which could have an influence on the team work, we will not go into this, because it is out of scope for this report.

[3] In this project a game had to be developed, in which the calculation of a vector - based on an image - had to be made. The image should contain a person, who uses his hands to demonstrate a vector by taking a position like he uses a bow and arrow. The requirement was to use webcams, to capture the image, and to use RaspberryPis to detect the hands of the person and to calculate the vector. Furthermore, an Android application should be included in the program.

[4] Team ITU planned to develop an individual cannon game with an appropriate concept for a global collaboration project.

frustration, in consideration of the given deadline of the course and on recommendation of the advisor, Team ITU decided to not take this option.

Team ITU received the project topic and the contact details of the members of Team Kenya on the 17.9.2013. So the collaboration between the student groups did not start before this date.

6.2.2 Team Kenya

In the beginning Team Kenya consists of three members, which are all in the Masters programme "Telecommunication and Innovation" of the Strathmore University. The members are all from Kenya. The official languages in Kenya are English and Swahili. The members of Team Kenya met each other the first time on the 1.10.2013 (–> Appendix B: Global Meeting Report 1.10.2013 Page 1). One member had to leave the group in the last third of the project due to workload of other projects and obligations (–> Appendix B: Global Meeting Report 26.11.2013 Page 2).

The department, which is responsible for the masters programme "Telecommunication and Innovation", wants to convince the school management from Strathmore University to invest into the technology and idea of the project to improve environmental conservation. Therefore this collaboration project was initiated. (–> Appendix B: Global Meeting Report 3.12.2013 Page 2) .

Maybe in the Introductio Context Part?

For the students from Team Kenya the project is not included in any mandatory course and is completely voluntarily. The students from Team Kenya spend their free time on this project. There are no mandatory requirements or deadlines, which the Kenyan students have to achieve, except that they have to create documentation for their advisor to prove the progress of the project (–> Appendix B: Global Meeting Report 19.11.2013 Page 2). The motivation for the Kenyan students to participate in this project is on the one hand to support the environmental thought and on the other hand to use this project work as basis for their master thesis (Email 10.12.2013).

Add refer-nce

6.3 Project Management

6.3.1 Approach

As the course for the team ITU started in late August and the project team was already known, Team ITU already started with a project management technique. This was already established when Team Kenya got into the project.

Team ITU chose an agile project management approach, because it is more flexible, which was important due to the fact, that Team ITU did not knew the student group from Strathmore University neither their skills nor their requirements on this project. Moreover an agile project management approach is known for being more suitable for software development projects (*TODO: reference*).

Find reference

Team ITU planned to merge Team Kenya into their project management approach, because they did not come up with a different approach on how to organise the project. Team ITU asked several times for feedback or other suggestions according to project management methods and tools, but Team Kenya did not react at all or agreed with the suggestions of Team ITU. So Team ITU assumed that Team Kenya is fine with the project mangement approach. In general Team Kenya was very reticent in project management topics, although one student from Team Kenya seemed to be very interested in the project management and global collaboration challenges (–> Appendix B.3). The reason for this could be, that Team Kenya had not much time to spend on the project. In the end it also turned out, that the one kenyan student, who was the most interested in project management and global collaboration, left the project due to other obligations.

Team ITU considered introducing Scrum to the project. However, due to the inexperience and non-knowledge about this framework in both teams, this idea has been dropped. For Scrum it is necessary that every project team member knows the concept. Thus, each team member would need to learn Scrum, which in turn would have taken ressources from the implementation.

6.3.2 Project Organization

6.3.2.1 Timeline

Team Kenya agreed on to go with the deadlines from Team ITU as they do not have to meet any deadlines. (–> Appendix B: Global Meeting Report 24.10.2013 Page 1)

6.3.2.2 Project Team and Roles

Each member of the project team created a member profile to introduce themselves, which are attached in the Appendix B.1.

The members of Team ITU split up the tasks, such that different kind of roles arose. The roles were not set in the beginning, but emerged during the progress of the project. They define the major responsibilities. This does not

Collaboration master

Server Developer

Image Processing Developer

Prediction Model Developer

Team Kenya/Android Application Developers

6.3.3 Project Management Tools

6.3.3.1 Member profiles

Team ITU suggested to come up with some member profiles of each project team member to introduce each other in the beginning of the project. The intention of the member profiles was to set the first steps for building up trust within the distributed project team. The project team members could recognise the members as individual persons with own personalities and different backgrounds in the beginning. The profile members should also preclude misunderstandings, for example according to the genders and how to address a person.

One member of Team Kenya followed the suggestion in the beginning. The other two members provided their member profiles three weeks after Team ITU reiterated the request.

Because Team ITU did not receive the member profiles of all students from Team Kenya in the beginning, a first disappointment on the side of Team ITU could be noted. Team ITU already waited since end of august to meet the Kenyan group, with whom they supposed to work with. There was high curiosity on the side of Team ITU, which was not satisfied by Team Kenya. This probably already led to a higher mistrust, as two members from Team ITU, which already collected some negative experience with previous global collaboration projects, felt reassured.

6.3.3.2 Introduction and Kick-Off meeting

6.3.3.3 Skill-/Preference- and Motivation-Sheet

To see where the strengths, weaknesses and motivations within the project team are Team ITU came up with a Skill-/Preference-Sheet and Motivation-Sheet. The Skill- and Preference-Sheet focusses on the technical concerns like programming languages, version control tool, database- and other technologies. So when a skill in a technology is purely pronounced within the project team, the most preferred technologies can be considered. The Motivation-Sheet, on the other hand, focusses more on the different project work tasks and topics. With this information the roles within the project can be split up, so that every member is the most satisfied and therefore more motivated.

Every project team member was encouraged to fill out these tables. The results can be seen in the Appendix B.2 and B.3.

The outcome of these tables also influenced some decisions in the project. For example, the decision of which programming languages Team ITU choose to benchmark was based on this (*reference*). Also the choice of database technology and which version control system to use for the source code is based on the Skill-/Preference-Sheet. It also pointed out the possible roles of each project team member within the project (*reference*).

Add reference

Add reference

In the result it can be noticed that image processing, prediction models and RaspberryPis were quite unknown amongst the project team members. So each project team member was encouraged to do some research on these topics in the beginning.

6.3.3.4 "Out-of-Office"-Calendar

Because every project team member had different obligations besides the project, Team ITU came up with an "Out of Office"-Calendar, in which every project team member could publish for example exams, other hand-ins or other unavailability. The motivation of this calendar was to allow for better planning of meetings and assignments during the project. The problem was that no one really looked into this calendar except in the beginning of the project or when a certain issue aroused. The "Out-of-Office"-Calendar can be seen in the Appendix B.4.

So it was noticed by Team ITU that Team Kenya as well as Team ITU had some exams in October. This influenced the project plan according to when to probably start the implementation phase. In the first half of November Team Kenya had a second exam phase. This was not correctly mentioned in the calendar, so Team ITU was not aware of this situation in the beginning. Team ITU got informed of the exam phase by Team Kenya in a meeting, when Team Kenya excused themselves for the non-fulfillment of agreed assignments (–> Appendix B: Global Meeting Report 12.11.2013 Page 2). This led to amazement and disappointment on the side of Team ITU. Team ITU wondered why Team Kenya even agreed on the assignments the week before, when they knew that they probably do not have the time to fulfill these assignments. Team ITU expected that Team Kenya communicates such situations before they happen.

6.3.3.5 Milestone plan

Team ITU chose to create a very rough milestone plan to have an eye on the deadline given by ITU. A detailed milestone plan would contradict the idea of the agile project management approach. Team ITU did not know Team Kenya, their skills and requirements in the beginning and a detailed milestone plan probably would have been rescheduled several times. In the milestone plan Team ITU roughly pointed out when they want to be done with each phase of the project.

The following milestones where set in the beginning of the project.

Milestone	Description	Planned	Achieved
First Global Meeting	Introduction session with Team Kenya to get to know each other and sharing requirements and expectations of the project	01.10.2013	01.10.2013
Final definition of Requirements and Roles	Get all the main requirements together and splitting up the project tasks	31.10.2013	03.12.2013
Finishing Prototype	A complete prototype of the product	01.12.2013	15.12.2013
Report	A complete final report of the project	13.12.2013	16.12.2013

Table 6.1: Milestone Plan

As it can be seen in table 6.1 the planned milestones were mostly not achieved in time. The most significant difference between a planned and a achieved date can be seen at the second and third milestone, which contained a final defintion of requirments and roles and the development of a prototype. The reason for this is on the one hand that Team ITU was too optimisitic in the planning process of these milestones, as they assumed that the members of Team Kenya have the same work pace and working experience like they have.

Furthermore the milestone plan was not properly introduced and discussed with Team Kenya. But that is because Team Kenya did not show much interest and their response time was very slow, especially in the first two months of the project. On the other hand Team ITU could not encourage Team Kenya enough to do their part of the project in time or the results were not satisfactory enough, such that Team ITU could work with it. In consequence several iterations and new requests had to be made from Team ITU to Team Kenya, which slowed down the project progress a lot. Team ITU was to patient in the end, because they still hoped that Team Kenya will get a better drive after their exam period as Team Kenya make Team ITU believe.

Figure 6.1 shows the milestone plan, which was planned in the beginning of the project.

The milestone plan was mostly a guideline for Team ITU, because they had a deadline in contrast to Team Kenya. Team ITU shared the planed dates for half of the milestones with Team Kenya as they agreed to go with the deadlines from Team ITU. (–> Appendix B: Global Meeting Report 24.10.2013 Page 1) These shared milestones were according to when to finish the prototype and when to hand-in the mandatory final report.

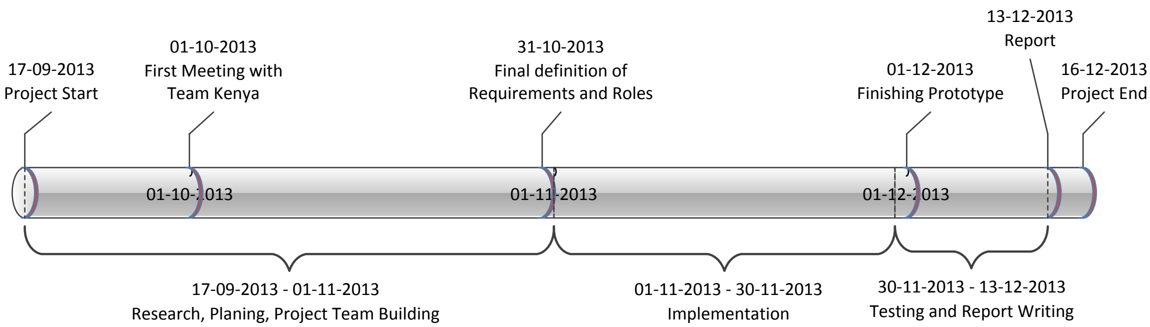


Figure 6.1: Milestone Timeline

6.3.3.6 Internal weekly meeting (Jour-Fixe)

Team ITU agreed on having an internal weekly meeting to update each other on the current progress, to discuss occuring topics, make decissions, work together and assign each other new assignments till the next meeting. Furthermore Team ITU also met with their advisor to check if they are on the right track. The concept of the weekly meeting equates to the concept of a Jour Fixe or a Daily Scrum .

Find reference

Find reference

There was a constant agenda for the meeting:

1. Status update of each team member
   (a) What has each group done during the past week according to the project?
   (b) Which achievements were made?
   (c) Which problems occurred?
   (d) Any other news
2. Discussions and Decisions (the topics in this section varied weekly according to the current progress)
3. Meeting with advisor
4. Assignments

This constant agenda gave some structure to the project. The team members knew what to expect from the meeting and where they can place their concerns.

The members of Team ITU were mostly fully represented in the internal weekly meeting.

6.3.3.7 Global weekly meeting (Jour-Fixe)

Team Kenya and Team ITU agreed on having a weekly Skype-Meeting (*reference*) to update each other on the current progress, to discuss occurring topics, make decisions and assign each other new assignments till the next meeting, just like the weekly internal meeting. The meeting took place every Tuesday afternoon/evening. According to a Doodle-Survey (*reference*) all project team members could reserve this time for the project meeting.

Add reference

Add reference

Each group could make a suggestion for the Agenda, but Team Kenya never used this opportunity. Therefore only Team ITU prepared an agenda, which was structured in the following way:

1. Status update of each group
   (a) What has each group done during the past week according to the project?
   (b) Which achievements were made?
   (c) Which problems occurred?
   (d) Any other news
2. Discussions and Decisions (the topics in this section varied weekly according to the current progress)
3. Assignments

It can be noted, that the structure for the agenda of the global meeting is similar to the agenda of the internal meeting. This is because the structure was already proofed by Team ITU before the global colaboration started and was considered to be suitable by all members from Team ITU. Although it was offered to Team Kenya, they never made alterations to the agenda.

After the first few meetings Team Kenya and Team ITU agreed to prepare the status update in advance, because it took quite a while for each group to present their status updates in the Skype chat. Unfortunately Team Kenya did not comply with this agreement except in the last meeting, where none of the Kenyans could attend due to other obligations or internet connection problems (Table 6.2).

As it can been seen in Table 6.2 almost every meeting took place. Team ITU was always in the majority. The attendance from Team Kenya was low mostly due to other obligations and internet connection problems.

6.3.3.8 Meeting reports

For every internal and global weekly meeting Team ITU wrote a meeting report. These meeting report documents all updates, news, discussions, decisions and assignments, which

Global Meetings	Attendance Team ITU	Attendance Team Kenya	Prepared Status update Team ITU	Prepared Status update Team Kenya
01.10.2013	4	1	no	no
24.10.2013	4	2	no	no
31.10.2013	3	2	no	no
12.11.2013	4	2	no	no
19.11.2013	4	2	yes	no
26.11.2013	4	1	yes	no
03.12.2013	3	1	yes	no
10.12.2013	4	0	yes	yes

Table 6.2: Attendance and prepared status updates of the global meetings

were discussed in the meetings. After publishing the meeting report, each project team member had the chance to correct wrong information or add missing information for the next couple of days. Afterwards the information and assignments in the meeting report were binding.

Project team members which could not attend the meeting had the chance to be updated with all necessary information. The report also serves to confirm the content of the meeting. Furthermore, the report is a source the team members can rely on. For example in case it comes to false allegations against a team member, the report could be used as a proof. The meeting report is also a reminder on the agreed tasks.

Team ITU expected that the agreed assignments were done or at least initiated to the next meeting. Since this never happened on the side of Team Kenya in the beginning, Team ITU started to put some deadlines on the most important assignments. But even those assignments were mostly not done in time, satisfactory or done at all, although Team ITU got specific confirmation by email on the published meeting report or no disagreement on the meeting report from Team Kenya.

This lack of collaboration and passive attitude of Team Kenya frustrated the members of Team ITU a lot, such that the motivation of Team ITU on collaborating with Team Kenya was gone by the half of the project. Also the expectations towards Team Kenya changed much lower expectations than in the beginning.

6.3.3.9 Time recording

Time recording is a quite popular tool amongst consultancies. Consultants record their times to keep track of project work they are performing for different customers. Based on this recorded times correct invoicing can me ensured. Some Human Resources Departments also use time recording to keep track of the presence of their employees. But it also can be used for other purposes.

Team ITU decided to record the time they spend on each part of the project. This time recording data helped to evaluate the spended time on a topic for each team member and for the whole group. This information can be a motivator in those situations, where it seems that no progress is done. If no visible progress was made within the project, the information of spended hours made it less frustrating. Furthermore, it also can encourage members to work on the project by either comparing themselves to other team members or assess whether the time they spend is appropriate. This worked out for Team ITU, because the attitude towards the project of each member was similar.

The only challenge of this tool is that you have to remember to record your time. To make it more easy Team ITU used a time recording software (See in section 6.3.4.3).

Project Parts	September	October	November	December	Total
Lectures/Advisory Meetings	4 %	12%	5%	3%	5%
Reasearch	58%	29%	11%	3%	16%
Internal Collaboration	9%	33%	20%	15%	20%
Global Collaboration	1%	23%	13%	4%	11%
Implementation and Testing	0%	0%	39%	15%	23%
Report	17%	1%	8%	59%	22%
Other	11%	2%	4%	1%	3%

Table 6.3: Results of the time recording for the Project "Occupancy Analyzer". The table shows the ratio of hours spend for each project part for each month. In addition the table also contains the ratio of hours for each project part for the whole duration of the project. The data is based on the recorded times in the Time Recording Software "Toggl" between the 17.9.2013 and 13.12.2013.

Team ITU was also interested in the result of the total hours spend in each project part in the end of the project. The results[5] can be seen in Table 6.3.

It can be noticed that research was a main task in the first half of the project, while in the second half of the project the focus was mostly on the implementation. This approximately conforms with the planned milestone plan in Table 6.1.

The amount for internal collaboration activities was more or less constant. An higher workload can be noticed in October. The same pattern can be seen on the global collaboration work. Also here is the highest work load in October. This is probably because the most communication and initialization according collaboration had to be set in the beginning.

[5] Note that these results contain approximately values due to the fact that not every time was recorded

Team ITU did not suggest this tool for the whole project team, because they had the feeling that it would be an overload for Team Kenya, which already had troubled to deliver progress within the project.On the other hand the result would have been very intersting and informing for Team ITU, as they presume Team Kenya does not put any effort into the project.

6.3.3.10 Surveys

6.3.4 Collaboration Software Tools

6.3.4.1 Communication

- Skype (excluding Google Hangout)
- Email

6.3.4.2 Sharing

- Google Drive (excluding Skydrive)
- Github

6.3.4.3 Time recording

- Toggl (excluding Excel)

6.4 Collaboration Issues

In this chapter Team ITU points out the major issues within the collaboration with Team Kenya and describe the attempts of avoiding these issues, when Team ITU noticed them.

6.4.1 Lack of Communication

One main issue during the collaboration with Team Kenya was the lack of communication between Team ITU and Team Kenya.

The lack of communication already started in the beginning of the project. Team ITU started the first contact with Team Kenya with an introduction email , in which they introduced

Reference to the Email

themselves with member profiles and some information about the project topic change they had to deal with. Furthermore Team ITU also suggested a date for a first face-to-face meeting. Since the members of Team Kenya neither introduced themselves nor responded to the suggestion for the meeting, Team ITU sent another Email one week after the first Email, in which they asked again if Team Kenya is up for a meeting on the suggested date. This time one of the kenyan students responded and a meeting date could be fixed. Team ITU did not get any response from the other members of Team Kenya. Also the kenyan student, who answered us, could not get in touch with the other members. Finally the first meeting took place with Team ITU and one member of Team Kenya.

Reference to the Email

This was already a disappointment for Team ITU and made little hopes for the future collaboration. Also during the project the response time to Emails was very slow (often it took several days until a response). Team ITU started to use deadlines for responses, but this did not make a difference in the response-time. Once in a while Team ITU also had to check if Team Kenya even get the Emails by sending another "Reminder"-Email.

Team ITU confronted Team Kenya with this issue in an Email and once again in a meeting. Team Kenya excused themselves for the slow responses and wanted to improve. But this improvement was not noticed by Team ITU.

Another two situations shows the lack of communication: Team ITU created a shared folder with Google Drive, where Team Kenya and Team ITU could exchange information. Team ITU sent invitations and an Email to Team Kenya, where they explained the documents, which they aleady had put into the shared folder. Team Kenya supposed to put also some information into the shared folder. In the next meeting, two weeks later, Team Kenya claimed that they could not share files with Team ITU, because they allegedly did not have access to the Google Drive folder. This response confused Team ITU quite a lot as they could notice that Team Kenya already put some files in the shared folder. Furthermore it was an poor excuse, because they could have sent us the information also via Email.

Team ITU requested Team Kenya to communicate such issues immediately in the future. Team ITU did not address the confusion about the supposedly not available access to the shared folder, because they did not want to accuse Team Kenya of lying.

Team Kenya also did not mention to Team ITU that they have a second exam phase in advance. Team Kenya told Team ITU about the exam phase while they were within the exam phase and could not bring up the results of agreed assignments.

6.4.2 Misconception of the project work and project team

6.4.3 Different priorities

6.4.4 Lack of Cooperation

6.4.5 Lack of Skills

6.4.6 Lack of Initiative

6.4.7 Prejudices

6.4.8 Other issues

6.4.9 Feedback from Team Kenya

- Misconception of the project work and project team
- Failure to comply with the assignments
- Communication
- Lack of skills
- Other exams/hand-ins
- Differing requirements
- Attendence of meetings
- Equipment
- Prejudice

6.5 Hypothetical Scenarios

- Assignments
- Communication
- Requirements

- Organisation by the universities (Requirments, clarification,)

7 | Discussion

8 Conclusion

9 References

[1] Y. Benezeth, P.-M. Jodoin, B. Emile, H. Laurent & C. Rosenberger. October 2012. *Comparative Study of Background Subtraction Algorithms*, pp. 5-6.

[2] Aditi Job & Shirish Halbe. October 2012. *Multiple Objects Tracking using CAMshift Algorithm in Open CV*, pp. 41-46.

[3] Gellert, A. & Vintan, L., 2006, *Person Movement Prediction Using Hidden Markov Models*, Computer Science Department, University of Sibiu, http://webspace.ulbsibiu.ro/arpad.gellert/html/SIC_HMM.pdf.

[4] Ashbrook, D. & Starner, T., *Using GPS to Learn Significant Locations and Predict Movement Across Multiple Users*, Georgia Institute of Technology, http://www.cc.gatech.edu/~thad/p/journal/using-gps-to-learn-significant-locations.pdf.

[5] Bhatia, N., 2010. *Survey of Nearest Neighbor Techniques*, p. 303. http://arxiv.org/ftp/arxiv/papers/1007/1007.0085.pdf

[6] Massimo Piccardi. 2004. *Background subtraction techniques: a review*, p. 1.

[7] Mohamad Hoseyn Sigari, Naser Mozayani & Hamid Reza Pourreza. February 2008. *Fuzzy Running Average and Fuzzy Background Subtraction: Concepts and Application*, p. 1.

[8] Herman Cheng, Zhicong Huang & Mark Kumimoto. Spring 2006. *Final Project Report - Image Processing Techniques*, pp. 2-4.

[9] Matt Siber. Fall 2005. *Resolution and File Size*, p. 1.

[10] Brian Lovell *Thresholding Techniques*. Spring 2003.

[11] R. Fisher, S. Perkins, A. Walker & E. Wolfart. 2003. *Dilation*.

[12] R. Fisher, S. Perkins, A. Walker & E. Wolfart. 2003. *Erosion*.

[13] Dr. Nicolas Pronost. *Introduction to Image Processing*, p. 9.

[14] Ahmed Elgammal. Spring 2008. *Histograms of Digital Images*.

A | Appendix A

B | Appendix B

B.1 Member Profiles

Member Profile of <Name>

General Information

Name:	Ann Chemutai
Surname :	Maritim
Gender:	Female
Country:	Kenya

Contact Information

Email:	annchemu@gmail.com
Skype-Name:	ann.chemutai

Education and Work

University:	Strathmore University
Programme:	MSc. Mobile Telecommunication and Innovation
Previous Education:	Bachelor of Business and IT
Programming Languages:	Java including mobile apps development, C++, PHP,
Work Experiences:	Software developer

Further Information

Hobbies:	Hiking, dancing, art
Comments:	Hope to collaborate with you guys fully

TEAM KENYA

MEMBER PROFILE: CECIL MIGAYI.

General info:

Name:	Cecil
Surname:	Migayi, Ochieng’
Gender:	Male.
Country:	Kenya.

Contact info:

Email:	migayicecil@gmail.com
Skype:	migcecil

Education & Experience:

University:	Strathmore university
Programme:	Msc. Mobile Telecommunication and Innovation.
Previous Education:	Bsc. Computer Science.
Programming languages:	J2me, android, php, REST, mysql

Further info:

Hobbies:	Gaming, football and rugby fan.
Comments:	

Member Profile of Elizabeth Wayua

General Information

Name:	Elizabeth Wayua
Surname :	Mutisya
Gender:	Female
Country:	Male

Contact Information

Email:	nwayua26@gmail.com
Skype-Name:	wylin26

Education and Work

University:	Strathmore University
Programme:	Masters of Science in Telecommunications Innovation
Previous Education:	Bachelor of Business and Information Technology
Programming Languages:	Java, HTML, PHP, Android
Work Experiences:	Website development at Strathmore University

Further Information

Hobbies:	Swimming, Reading

Comments:	Very passionate about this project, I would like to see it through to the end

Member Profile of <Name>

General Information

Name:	Christoffer
Surname :	Pagaard
Gender:	Male
Country:	Denmark

Contact Information

Email:	cwil@itu.dk
Skype-Name:	acronsard

Education and Work

University:	IT-University of Copenhagen
Programme:	Software Engineering
Previous Education:	Software Development and Network technician
Programming Languages:	C#, Java, F#, ASP.NET, ASP, VB.NET
Work Experiences:	Some mobile development.

Further Information

Hobbies:	Goalball and some development on different applications and sites
Comments:	Visually impaired.

Member Profile of Jacob Romme Rasmussen

General Information

Name:	Jacob
Surname :	Romme Rasmussen
Gender:	Male
Country:	Denmark

Contact Information

Email:	jrra@itu.dk
Skype-Name:	J7a7c7o7b

Education and Work

University:	IT University of Copenhagen
Programme:	Software Development and Technology – Software Engineering
Previous Education:	BSc. Software Development
Programming Languages:	C#, F#, Java, C
Work Experiences:	Nothing study related.

Further Information

Hobbies:	Gaming, Programming, Bicycling.
Comments:	Even a broken clock is right twice a day.

Member Profile of Theresa Brandt von Fackh

General Information

Name:	Theresa
Surname :	Brandt von Fackh
Gender:	Female
Country:	Germany

Contact Information

Email:	tebr@itu.dk
Skype-Name:	theresa.bvf

Education and Work

University:	IT University of Copenhagen
Programme:	Software Development and Technology
Previous Education:	B.Sc. Business Information Technology (in Germany)
Programming Languages:	C#, Java, Python, C, Delphi
Work Experiences:	3 years Systems Consultant, specialized in Identity and Access Management (in Germany)

Further Information

Hobbies:	Longboarding, Snowboarding
Comments:	

Member Profile of Tomas Miseikis

General Information

Name:	Tomas
Surname :	Miseikis
Gender:	Male
Country:	Lithuania

Contact Information

Email:	tmis@itu.dk
Skype-Name:	daghoster

Education and Work

University:	IT University of Copenhagen
Programme:	Software Development and Technology (Software Engineering track)
Previous Education:	IT, Communication and New Media – Aalborg University
Programming Languages:	Java, Python, Ruby, Scala
Work Experiences:	Freelance programmer

Further Information

Hobbies:	Cycling, Poker, Basketball
Comments:	No comments.

B.2 Skill-/Preference-Sheet

Skills/Preference	Theresa		Tomas		Jacob		Christoff		Ann		Cecil		Wayua	
	Skill	Pref.	Skill	Pref.	Skill	Pref.	Skill	Pref.	Skill	Pref.	Skill	Pref.	Skill	Pref.
Programming language														
C#	7	7	0	0	8	8	9	10	0	0	2	5	0	1
Java	6	8	7	8	7	7	8	9	8	9	6	5	6	6
C / C++	3	0	0	0	4	2	2	0	4	4	4	3	0	0
Python	5	9	6	8	3	2	0	4	0	0	3	4	0	0
Delphi	5	0	0	0	0	0	0	0	0	0	0	0	0	0
DBMS														
MS SQL	6	8	0	0	5	6	6	4	4	4	0	4	5	8
MySQL	3	7	8	10	6	7	7	6	8	8	8	3	7	7
Oracle	0	6	0	0	0	0	0	0	4	4	0	7	6	7
HSQLDB	1	5	0	0	0	0	0	0	0	0	0	0	0	0
PostgreSQL	0	0	6	5	0	0	0	0	0	0	0	0	0	0
IDE														
Visual Studio	5	8	0	0	8	8	10	10	8	8	2	5	3	2
Netbeans	0	0	6	0	5	5	6	9	8	8	6	6	6	7
Eclipse	5	8	10	7	7	6	7	0	9	3	6	6	6	7
IntelliJ IDEA	0	0	7	10	0	0	0	0	5	5	0	0	0	0
Other														
HTML	7	0	8	8	3	0	10	0	9	9	8	6	7	9
XML	6	6	6	4	6	5	5	5	5	5	6	7	4	3
UNIX/Linux	2	3	5	7	3	4	3	3	5	4	2	3	5	5
Raspberry PI	0	4	0	8	2	5	0	0			0	0	0	2
Networking (TCP/UDP/...)	2	2	6	6	2	2	7	7			2	3	4	5
Android App	0	3	7	6	2	4	7	2			6	6	6	5
Prediction Models	0	7	0	5	0	7	0	3			0	0	0	0
Image Processing	0	1	0	4	0	0	0	0	7	7	0	0	0	0
Version Control														
Git	1	5	7	7	4	4	7	7	6	6	1	0	4	5
SVN	4	5	3	5	7	7	7	7	6	3	0	0	0	0

0 = nope
10 = expert

B.3 Motivation-Sheet

Motivation	Theresa	Tomas	Jacob	Christoffer	Wayua	Cecil	Ann
Project management (local team)	8	3	2	8	10	1	
Virtual project management (both	8	3	2	8	8	2	
Collaboration Challenges	10	5	4	10	7	5	
Programming/Implementation	5	10	10	5	4	7	
Architecture/System design	9	8	7	10	9	5	
Requirements/Analysis	6	6	5	9	7	5	
User interfaces/GUI design	5	7	7	0	8	5	
Test/QA	4	4	4	4	9	5	
System documentation	5	6	5	3	8	4	
Writing report	8	6	7	6	7	4	
Support (repository, server etc.)	2	6	4	0	4	5	
Grade (10 means top grade)	9	9	8	9	8	9	

0 = No interest
10 = very interested

B.4 "Out-of-Office"-Calendar

Oktober 2013																															
Name	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**	**16**	**17**	**18**	**19**	**20**	**21**	**22**	**23**	**24**	**25**	**26**	**27**	**28**	**29**	**30**	**31**
Christoffer (cwil)														X	X	X	X	X	X	X									X		
Jacob (jrra)														X	X	X	X	X	X	X											
Theresa (tebr)														X	X	X	X	X	X	X			X		X	X	X				
Tomas (tmis)														X	X	X	X	X	X	X											
Wayua							X		X	X										X	X	X						X	X		
Ann							X		X	X										X	X	X						X	X		
Cecil							X		X	X										X	X	X						X	X		

November 2013																															
Name	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**	**16**	**17**	**18**	**19**	**20**	**21**	**22**	**23**	**24**	**25**	**26**	**27**	**28**	**29**	**30**	**31**
Christoffer (cwil)																															
Jacob (jrra)																															
Theresa (tebr)							X	X	X	X				X							X							X	X	X	
Tomas (tmis)																															
Wayua					X		X					X		X																	
Ann					X		X					X		X																	
Cecil					X		X					X		X																	

December 2013																															
Name	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**	**16**	**17**	**18**	**19**	**20**	**21**	**22**	**23**	**24**	**25**	**26**	**27**	**28**	**29**	**30**	**31**
Christoffer (cwil)														X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
Jacob (jrra)														X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
Theresa (tebr)					X							X		X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
Tomas (tmis)														X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
Wayua																															
Ann																															
Cecil																															

X	Not available (Work, Private, Busy, ...)	X	Holiday
X	Exam	X	Other Event

B.5 Global Meeting Reports

Global Meeting Report 01.10.2013

Team ITU

Date	Time	Duration	Attendees	Communication medium
Tuesday, 01.10.2013	03:15pm (CET) 4:15pm (EAT)	~ 0:30 min	ITU Team (4 total): • Tomas • Christoffer • Jacob • Theresa Kenya Team (1 total): • Wayua	Google Hangout

Agenda: Introduction

Preamble

This report is written by Team ITU. The connection wasn’t that good, so it was difficult to understand each other. The answers of Team Kenya may be incomplete. The suggestion of the Team ITU is to focus on communication via email and chat.

Items discussed

- Short introduction of the attendees
- Initial description of the project for each team is matching
- Upcoming exams for Team Kenya and Team ITU

Questions

- Do the teams know their members?
  - Team ITU: Have regularly meetings since beginning of September
  - Team Kenya: Haven’t meet as a group before, but will meet tonight
- Have the teams already started yet?
  - Team ITU: Started with the project “Vector Shooter” in the beginning of September. The project changed to “Occupancy Analyzer” on the 17th September, because no Kenyan team wanted to the project “Vector Shooter”. Since then Team ITU started to work on the topic of the new project.
  - Team Kenya: Have exams till next week, so they haven’t started yet
- Raspberry PIs already received?
  - Team ITU: Received three Raspberry PIs from the university to work with
  - Team Kenya: Have access to Raspberry PIs, but don’t get their own Raspberry PIs
- What is the Android phone for in this project?
  - Team ITU: Uncertain about why an Android phone is necessary for an “Occupancy Analyzer”. Hopes Kenya Team can provide more information on this.

  - Team Kenya: Mobile phone should be seen as an extension. It is less cumbersome to carry around a smart phone than a laptop.
- Do you have any experience with Raspberry PIs?
  - Team ITU: No one has worked with a Raspberry PI before, but started to teach themselves this technology
  - Team Kenya: No experience with Raspberry PIs

Assignments

- Wayua will meet with the other attendees from Team Kenya
- Team Kenya will write down an interpretation and expectation of the project to Team ITU and vice versa
- Preparing the Agenda for a next meeting

Global Meeting Report 24.10.2013

Team ITU

Date	Time	Duration	Attendees	Communication medium
Thursday, 24.10.2013	16:00pm (CET) 17:00pm (EAT)	~ 3:00 min	ITU Team (4 total): • Tomas • Christoffer • Jacob • Theresa Kenya Team (2 total): • Cecil • Ann	Google Hangout, Skype

Agenda: Introduction

Meeting progress

- Started at around 16:30 CEST (17:30 EAT), when the first Kenyan student (Cecil) attended
- Second Kenyan (Ann) student attended at 17:00 CEST (18:00 EAT)
- Started to use Chat in Google Hangout, cause the connection was not good enough
- At 17:15 CEST (18:15 EAT) the platform was changed to Skype, because Google Hangout was not suitable for Chat
- ITU student (Christoffer) had to leave at 18:30 CEST (19:30 EAT)

Misunderstanding solved

- Team Kenya thought that they only assist the ITU students in the project
- Kenyan and ITU students are in the same boat: No experience with Raspberry PIs or other related topic to this project
- Both teams are equally responsible in this project

Information from Team Kenya

- Team Kenya met with their advisor and put some pressure on the advisor to get their equipment (Raspberry PIs and cameras)
- Team Kenya has no "Supervisor", which is doing the project lead (same for ITU Team)
- Team Kenya does not have any deadlines or requirement for the project
- Team Kenya will go with the deadline of Team ITU

Information from Team ITU

- Deadline for the report is the 16.12.2013, so Team ITU want the implementation to be finished end of November
- The structure Team ITU have talked about is as follows: Raspberry PI will take a picture, detect the people and calculate the person's center position. It will then

send this coordinate to the server which will store it and provide access to it and the objects movements for the android app.

- Suggestion for server side language: Java or C#

Splitting main tasks

- Team Kenya will be responsible for Android App Part (including appearance and function) and will do mainly the research and implementation on the Android App
- Team ITU will be responsible for the RaspberryPI-Part (including Image Processing) and Server-Part (including Prediction Models). Team ITU will do mainly research and implementation on these parts.
- Both teams can bring in ideas and arguments for every task
- Team ITU will provide the data through a TCP stream wrapped in JSON (Team ITU has not yet started on the server part, so they are not sure yet how JSON response from server will be formatted)
- Team Kenya will start with dummy JSON data and maybe try to do some Mock-ups and get an idea of how the Android App could look and function

Assignments till end of the week (27.10.2013)

- Team Kenya: Filling out Skill-/Preference AND Motivation-Sheets
- Team Kenya: Filling out the Doodle (http://doodle.com/mmqcqumck76f95g5) for finding a regularly weekly meeting with Team ITU
- Team Kenya: Checks if every member of Team Kenya has access to the documents on Google Drive. If not, this is to report to Team ITU and they will send another invitation.
- Team Kenya: Reply to the rest of the questions, which were in the past emails from Team ITU
- Team ITU: Theresa will write a report of the meeting (this document)
- Team ITU: Theresa will plan the future regularly meeting

Global Meeting Report 31.10.2013

Team ITU

Date	Time	Duration	Attendees	Communication medium
Thursday, 31.10.2013	17:00pm (CET) 19:00pm (EAT)	~ 1:45 h	ITU Team (3 total): • Tomas • Christoffer • Jacob Kenya Team (2 total): • Wayua • Ann	Skype

Agenda: Status-Update

Agenda (planned):

1. Status update from Team ITU
2. Status update from Team Kenya
3. Decisions
    a. Version Control
    b. Programming Language
    c. Where to process which data?
    d. Client-Server vs. Peer-to-Peer Document
    e. Other
4. Assignments
5. Feedback
6. Other

Meeting progress

- Started at around 17:10 CEST (19:10 EAT)
- ITU student (Christoffer) attended at 18:00 CEST (20:00 EAT)
- ITU student (Theresa) and Kenyan student (Cecil) couldn’t attend the meeting

Status update from Team ITU

- Managed to figure out on how to process images and determine the path a person took:
    - Calculate the difference between the room background image and the image containing a person and get the difference image
    - Blurring and thresholding the difference image to get rid of the noise in the image
    - Determine person's area, and take it's middle/center x,y coordinates
    - Determine the whole path a person took (fairly accurately)
    - Did it with Lego figures walking around in a small shoe box.
- The web cameras are very simple, so they will be placed in a way to cover as much as possible room area

- Will probably use Programming Language “Python” on the Raspberry PIs, according to the Benchmark (Document “Java – Python Benchmark” in folder “Team ITU”)

Status update from Team Kenya

- Still don’t have the Raspberry PIs (depends on the University)
- Can borrow one from a research lab nearby if it's necessary
- Discussed specs of the UI:
  - Displaying the information from the server on the android app
  - Navigation through the android app
    - User should choose the building he wants to look at, and the specific room/floor
- Familiarized with JSON
- Concentrate on the design for now. Functionality part will be done later.

Suggestions/Information for Team Kenya

- Propose to simply have a background image of a room for the android app
- Team ITU pass x,y coordinates of a person to the server, which then can be updated on the android app to display the path or current position of the person within the room
- Team Kenya decides on the look of the Android app (colors and all)
- Will the application be able to always display the information in real time?
  - Team ITU: The information about the movement in a room will be detected and sent all the time to the server, as long as there are connection between RPs and the server. And then the android app can always get the information from the server about the movement within the room.
- How fast will the data in the server be updated? Is it every 5 seconds or minutes?
  - Team ITU: Not sure yet how often the server will be updated. It will be probably quite rapid (around a few seconds). It'll only be updated whenever occupancy is detected
- How do you know that the cameras are covering/scanning the entire room?
  - Team ITU: 3 web cameras and 3 Raspberry PIs are available for now, which will be placed in 3 different rooms. User could be able to choose which room he wants to look at

Decisions

- Version Control:
  - Git Repository will be used. According to the Skill-/Preference-Sheet it’s the most reasonable choice
  - Tomas provided a repository: https://github.com/meshake/occupancy_analyzer/

Assignments Team ITU

- Send GitHub Usernames to Tomas, so he can add everyone to the repository
- Research more on prediction models
- Try to predict which door/path a person is likely to choose when walking in the room
- Detect and process several people walking in a room

Assignments Team Kenya

- Send GitHub Usernames to Tomas, so he can add everyone to the repository
- Ask Cecil if Tuesdays evening would be suitable for a regularly meeting (instead of Thursdays), because Theresa can’t attend on Thursdays at 5pm CEST (too early)

- Check on the updates in the Skill-/Preference-Sheet and fill out missing information
- Mockups of the screen for the android app by next Thursday (7.11.2013) and share results with Team ITU
- Have a look into the document “Client-Server-Model vs. Peer-to-peer” in the folder “Team ITU” and add information/comments if you want to

Global Meeting Report 12.11.2013

Team ITU

Date	Time	Duration	Attendees	Communication medium
Tuesday, 12.11.2013	17:00pm (CET) 19:00pm (EAT)	~ 2:00 h	ITU Team (4 total): • Tomas • Christoffer • Jacob • Theresa Kenya Team (2 total): • Cecil • Ann	Skype

Agenda: Status-Update

Agenda (planned):

1. Status update from Team ITU (Image Processing, Prediction Models, Server)
2. Status update from Team Kenya (Mockups, Android App, Equipment)
3. Data and format (JSON) for the Android App
4. SCRUM
5. Other

Meeting progress

- Started at around 17:06 CEST (19:06 EAT)
- Kenya student (Ann) attended at 17:36 CEST (19:36 EAT)
- Kenya student (Wayua) was missing

Status update from Team ITU

Image Processing

- We can detect multiple people moving around the building and to some degree differentiate between them
    - Simply record last x,y coordinate of every detected object
    - Next time we detect movement, we simply go through previously detected objects (if there are any) and if they are within +/- 20 pixels of the last objects, we interpret it as the same objects. Otherwise it's a new object.
    - If several people are close to each other they will be detected as one entity, but otherwise this method works fairly well.
- When we detect all objects, we simply generate a JSON formatted string and pass it to the server for processing and storage.

- The format of the JSON string, which is passed between the RaspberryPIs and the Server is on Github: https://github.com/meshake/occupancy_analyzer/commit/a706b0f7b0c9ee9544af39291e077a1d0897ea44 (you have to be logged in to see this)
- Except all the integers on rip_id, room_id and object_id will be in UUID format

Prediction model

- With regards to the prediction models we started out by making our own interpretation of a Hidden Markov Model
- We can predict fairly well where a person's next position will be given a current position and a previous position
- Predicting which exit the person will take given any position is still a work in progress. As it is now, the plan is that it should be possible to ask for prediction information of a given occupant and one should receive probabilities of the occupant going to the various exits. The probabilities are calculated given stored occupancy data of course.

Server

- The server model can receive multiple connections from both Raspberry PIs and Androids
- The data base layer is close to be done
- What is missing is linking the JSON input to the database, but this is soon done with the Raspberry PIs part
- What is missing is that we (ITU and Kenya) have to agree on which data should be available for the Android and which format the JSON should be in

Setting up RaspberryPIs

- Installation and configuration guide for RaspberryPIs is on Google Drive (It’s a draft; record issues to Theresa if a step is unclear or if something is missing)

Status update from Team Kenya

- Team Kenya drew some Mockups for the Android App
- Will meet with university contact person tomorrow to ask of the status of receiving Raspberry PIs
- Team Kenya doing exams this week and another term project

Data Format for Android App

- Team Kenya:
  - Same JSON, which is used between RaspberryPIs and Server for Android App
  - Variables: Coodinates, room, floor(if applicable), date/time, total number of occupancies
- Team ITU:

  - Team Kenya can get a sequence of objects in a room (occupants) in a given timespan. These objects will have a start position and the path they follow around. From this info the amount of people in a room can be extracted easily
  - i.e. int getOccupants(startTime, endTime) will return the number of occupants detected in a room in the given timespan
  - The Android App calls the server with JSON request, and the server will give some JSON formatted response

Conflict in Requirements

- Team Kenya: Monitor room occupation and resources to be used i.e if the room had fewer people reduce resource to a certain degree for example if a room had a certain number of people you will be required to regulate air circulation to a certain degree etc.
- Team ITU: Prediction where the occupancies go (which room for example). So the resources can be started in advance (heat up the room in advance or something like that)

SCRUM

- Team ITU was interested in if Team Kenya knows the agile software development framework “SCRUM”, because Team ITU was discussing to use it in the project
- Because Team Kenya don’t know much about “SCRUM” and the knowledge about “SCRUM” in Team ITU is also rare, we decided to not use it

Assignments Team ITU

- Writing meeting report
- Continue with the image processing, RaspberryPIs, server and prediction model
- Will give feedback on the required methods from Team Kenya (see assignments for Team Kenya) in the next meeting (Tuesday, 19.11.2013)
- Write a status update on next Tuesday and send it to Team Kenya at least 15 Minutes before the meeting (19.11.2013 16:45CEST)
- Discuss and try to include the requirements of Team Kenya (see also Conflict)

Assignments Team Kenya

- Ann will send the Mockups, which Team Kenya drew, on Wednesday (13.11.2013)
- Clear requirements for the API-methods: Write down methods, which the Android App will call and send it to Team ITU till Sunday (17.11.2013)
  - What input-parameters and what output is needed
  - Format of the JSON response
- Start on programming on the Android app (UI, back-end) on Monday (18.11.2013)
- Write a status update on next Tuesday and send it to Team ITU at least 15 Minutes before the meeting (19.11.2013 18:45EAT)
- Ann: Check on the updates in the Skill-/Preference-Sheet and fill out missing information

- Send GitHub Usernames to Tomas, so he can add everyone to the repository Share Mockup results with Team ITU
- Optional: Have a look into the document “Client-Server-Model vs. Peer-to-peer” in the folder “Team ITU” and add information/comments if you want to

Reminder

1.12.2013 Finishing a first working prototype of the occupancy analyzer. After this Team ITU will concentrate on writing the mandatory report. Further improvements and testing will be made in parallel.

16.12.2013 Hand-in of the report (End of project)

➔ **Just 2 weeks for finishing the prototype!**

Global Meeting Report 19.11.2013

Team ITU

Date	Time	Duration	Attendees	Communication medium
Tuesday, 19.11.2013	17:10pm (CET) 19:10pm (EAT)	~ 2:00 h	ITU Team (4 total): • Tomas • Christoffer • Jacob • Theresa Kenya Team (2 total): • Cecil • Wayua	Skype

Agenda: Status-Update

Agenda (planned):

1) Discuss Status Update Kenya
2) Discuss Status Update ITU
3) Feedback to Kenya
    a. Feedback from Theresa
    b. Why do you want to know where a person is in the room?
4) Future Collaboration
5) Assignments for this week

Meeting progress

- Started at around 17:10 CEST (19:10 EAT)
- Kenya student (Wayua) attended at 18:10 CEST (20:10 EAT)
- Kenya student (Ann) was missing

Status update from Team Kenya

- Delegated duties:
    - Ann: User Interface
    - Cecil, Wayua: Development and documentation of the functional part (retrieve data from database and represent them)
- Decided to work with canvas to map object coordinates in the App (Representation of the room data)
- Thoughts on prediction:
    - Prefer to use alerts (android toast)
    - The predictions only become live for a short time before the target achieves them. i.e The android user will only see an alert such as “object1 is headed to room2,

heater started” or “object1 is headed to room2, AC start”. Once object1 has gone to room2, the prediction ceases to exist.
  - Prediction: What room we foresee the use going after he/she leaves the current
- Update on requirements: Documentation
  - University staff will check on the progress and require a documentation at the end of the project

Status update from Team ITU

- **Image Processing:** Same status as last week
- **Prediction model:** Looked into the Hidden Markov library (jahmm) for possible prediction. It works just as described on the wiki page about HMM. Not sure yet if we want to use this library or Jacob’s implementation on the prediction, since jahmm is not very flexible when it comes to taking into account if person turned around and etc.
- **Server**
  - Team ITU uses wamp for running the server locally
  - What the database more or less looks like:
    - Objects: (Raspi_Id, Room_id, Object_id)
    - ObjectPath: (X_Coord, Y_Coord, timestamp)
    - Datatypes:
      - All IDs are UUIDs
      - Coordinates are in Integer
      - Timestamps are in Long
- **Setting up RaspberryPIs**
  - Compiled OpenCV on the second Raspberry PI
  - Connected a webcamera to the Raspberry PI and build up a Livestream
- **Report:** Created a Latex document with the general structure and the frontpage of the report

Decisions/Disusssions

- **Using canvas to map the objects (room occupants) in the App**
  - To pin point people coordinates
  - Keep in mind: The coordinates are image coordinates and are inaccurate when translating to a top down view. The image will have an angle, but it is not something that should be considered too much as it will increase the project size too much.
- **Format of the JSON request**
  - Generic example:
    {"command":"commandName","param":[{"name":"param1","value":"60592575600000"},{"name":"param2","value":"1384877575145"}]}
    - For example “commandName” could be replaced with getRoomStatus, “param1” with startTime, “param2” with endTime, and “param3” with roomid
    - The RoomID must be an UUID in the request to the server (can be translated to a nicer representation by the App for the end-user)

- **Output of the method “getRoomStatus”**
  - Object list with number of objects (occupants) if any
  - The object entities contain an Object ID and the latest coordinate/position in the room (X and Y coordinate of the image)
    - **NEEDS CLARIFICATION:**
      Question to Team Kenya: If you send a request with a start time and an end time, you expect a list of objects, where each object represents a person with the latest coordinate. Why do you send a start time and an end time to the server in the first place, when you only want the latest coordinate of a person?
      When you want the end-user to provide a start time and an end time, you probably want the path of each person in that time span. That means that you will get several coordinates for each person. To clarify, please write down EXACTLY what you want. What input and what output (including the format). Is one method enough to cover all App functionalities?
  - Timestamps are in Long and must be converted by the App
  - Format of the coordinates
    - Integer, long or double doesn’t matter, since Team Kenya is not considering too much of accuracy
    - Team Kenya will give Team ITU an update on this if any changes in the format must be done (Deadline for changes: 1.12.2013)
- **Livestream of the current occupants in a room**
  - App queries the server for example every 5-10 seconds
  - App checks output of the query for detected objects
  - If there are detected objects, they will be displayed on the App
- **Servers will be set up locally**
  - Team Kenya can run the server locally
  - Reason: Due to the amount of maintenance, Team ITU won’t provide a public reachable server for Team Kenya
  - Team ITU develops the server application and will send Team Kenya a compiled jar file, so that Team Kenya can run the server application on a local server
  - Team ITU will send Team Kenya the database-data (data, which the raspberryPIs sent to the database)

Future Collaboration

Team ITU -> Team Kenya: For future collaboration we would want to encourage you to try to be more active, and propose your own solutions to problems, and not only wait for our proposals. You shouldn't be afraid to be more proactive, because we are at the same page as you, and we have 0 experience in image processing and prediction models, so everything is new to us, and we are not always right on certain things and certain ways we do things. Furthermore, we would like for you to put more effort by completing the tasks you have, and ALWAYS send email asking us for further explanation, if you have any questions so we could give you any feedback you need. You shouldn't wait till the next meeting, if something you're doing right now requires any information from us.

Assignments Team ITU

- Theresa will write the meeting report
- Feedback to Cecils document (ITUquestions.docx)
- Christoffer will implement the getRoomStatus method and will sent a compiled jar-file of the server application to Team Kenya afterwards
    - There will be instructions provided to set up the server application
    - Database-data will be provided
- Christoffer will set up a meeting with Cecil and Wayua after the jar-file is sent
- In General: Further development on the image processing, raspberryPis and server part
- Will talk with supervisor about this whole prediction thing. (What is the usage? Where does this requirement come from?)
- Write an status update on the day of the next regularly meeting (Tuesday, 26.11.2013) and send it to Team Kenya before the meeting

Assignments Team Kenya

- **EXACT definition of the methods needed from the server for the Android App**
    - **What name for each method. Just one name for each method.**
    - **What input for each method? List of EACH variable with a description and a format (including the formats we already decided on)**
    - **What output for each method? List of EACH variable with a description and a format (including the formats we already decided on)**
    - **Also have a look in the "NEEDS CLARIFICATION"-Part in this document**
- Installation of the server application (with the instructions from Team ITU), which Team ITU will send
- Meeting with Christoffer of Team ITU
- Start to implement the android app
- Write an status update on the day of the next regularly meeting (Tuesday, 26.11.2013) and send it to Team ITU before the meeting

Reminder

1.12.2013	Finishing a first working prototype of the occupancy analyzer. After this Team ITU will concentrate on writing the mandatory report. Further minor improvements and testing will be made in parallel.
16.12.2013	Hand-in of the report (End of project)

➔ Just 11 days for finishing the prototype!

Global Meeting Report 26.11.2013

Team ITU

Date	Time	Duration	Attendees	Communication medium
Tuesday, 26.11.2013	17:10pm (CET) 20:40pm (EAT)	~ 3:30 h	ITU Team (4 total): • Tomas • Christoffer • Jacob • Theresa Kenya Team (1 total): • Cecil	Skype

Agenda: Status-Update

Agenda (planned):

1) Discuss Status Update Kenya
2) Discuss Status Update ITU
3) Feedback on communication
    a. Meeting report
    b. Server installation
4) Open Assignments for Team Kenya:
    a. If you send a request with a start time and an end time, you expect a list of objects, where each object represents a person with the latest coordinate. Why do you send a start time and an end time to the server in the first place, when you only want the latest coordinate of a person?
    b. When you want the end-user to provide a start time and an end time, you probably want the path of each person in that time span. That means that you will get several coordinates for each person. To clarify, please write down EXACTLY what you want. What input and what output (including the format).
    c. Is one method enough to cover all App functionalities?
5) Setting up the server
6) What to use the Prediction for? (ITU Advisor said it’s a requirement from Strathmore)
7) Connection of the rooms
    a. should be configured on the Android App
8) Assignments for this week

Meeting progress

- Started at around 17:10 CEST (19:10 EAT)
- ITU student (Jacob) attended at 17:23 CEST (19:23 EAT)
- Kenya student (Ann) was missing: Cecil excused her for having difficulties with the internet connection
- Kenya student (Wayua) was missing: Won’t be in the project anymore

Status update from Team Kenya

- Update on Project team: Wayua has excused herself from the project, because she's quite busy. She will be no longer in this project.
- Team Kenya has now access to Raspberry PIs
- Cecil: Currently trying to setting up the local server (installation of JAR-File)
- Anne: Currently working on the android development (UI and canvas)
- Answers from Team Kenya according to the questions in the meeting report from last week
    - There’s no need to have start and end time if we want only the latest coordinates
        - start time and end time will return a lot of data. When we have so much data requested at once, the phone might not respond very well. In this case we will not need start and endtime as parameters, only roomID. In otherwords, when we focus on the latest update of a room, the phone will respond much better
        - Requesting room status only by roomID

Status update from Team ITU

Image Processing:

- The timestamp is still long “real” timestamp, which can be converted to date / UTC time
- Tresholding: RP will only send coordinates to the server, if a person moves significantly
    - The point of sending coordinates of a person only if they move significantly is to avoid sending "useless" data of people who don't technically walk, but are standing or sitting in one place and only move their hands/heads or something like that.
    - Prediction does not get impacted by this

Prediction model:

- Converted C#-Implementation of the prediction-model into Java
    - Because the server was written in Java so it made sense to redo the prediction in Java (originally in c# because it made some testing parts easier)
- Bug Fixing
- Answer from the Advisor:
    - prediction part was originally a request from Strathmore to make the android app more interesting
    - Team ITU should have received some detailed requirements from Team Kenya

Server:

- Connection-Fix between RP and Server: If there is no data update from the RP to the server for 5 seconds, the connection will be closed; A new connection will be built up when RP wants to send data to the server after 5 seconds
- RP will only send one coordinate for each object to the server instead of a coordinate-path
    - Example file of the JSON sent to the server from the RPs on Google Drive
- Created JAR-File of the server
- Bug Fixing

Decisions/Discussions

- Prediction (View from Team Kenya): When will a person leave the room?
    - Feedback Team ITU:
        - Why you want to know the probability of someone exiting the room in the near future? What is the near future?
        - What is the probability regarded to?
        - We don’t have information about speed of an occupant
- Prediction (View from Team ITU): On which exit a person will leave the room?
    - One room with many exits: Determine which exit, the person is likely to take
    - Server will return a list of exits with corresponding probabilities and the android app could just pick the exit with the highest value. That way the android app only show an alarm if the probability is greater than 60% or something like that. With a room with only one door, a person will take this exit with a 100% probability.
- Prediction is calculated on the server
- Android App requested prediction data from the server
    - Methods have to be defined
    - Server will give back coordinates
    - Android App can choose a labelling for the exits
- Prediction data on the Android App
    - Toast on the android app: Display prediction data as an alert just below the screen
    - Suggestion Team ITU: Check the prediction for one occupant. Server makes a prediction for the given occupant and returns the probabilities of the person going to each exit of a room
- Logic of the connection of the rooms has to be made in the android app
    - The server only knows that there're different rooms/corridors (corridors are also just rooms), but doesn’t save information about how they are connected to each other
    - This part of prediction ("person1 is in room 1 and is heading to room 2"), has to be set up in the android app
    - Exits are saved manually on the server
    - Android app needs to request those exists to build up a reasonable connection of the rooms (because the following does not make sense: Room1 has 1 exit. Room 2 and 3 can be entered from Room1)
- Suggestion for methods from Team ITU (at least required)
    - getRoomStatus: Containing the latest coordinate for each object
    - method where the android app gets information on which rooms are existing and where the exits are in each room
    - method to request the prediction data (which probabilities for each exit for one object in a room)

Instructions for setting up the server

1. Push windows button and push the "r" butten just after (win + r)
2. Then type "cmd"
3. Now navigate to the directory where you have put the file
    a. To go one directory up, type "cd.."
    b. to enter a directory type "cd directory-name"

4. When you're at the folder type "java -jar GSDServer.jar"

If java is not recognized as internal or external command:

- Install java
- Set up environmental variables
    - JAVA_HOME as variable and link it to your java folder (jdk) as value
- If the problem continues → google it
- If you still have problems with setting up the server, ask Team ITU (Christoffer)

Feedback on communication

- If questions are asked, do answer them as requested asap
- Give replies on meeting reports, especially if they are explicitly requested
- Why do we have meeting reports?
    - everyone has an committed statement on things we agreed on
    - everyone has the chance to correct/complete the information
- Always communicate when you need help and cannot finish your assignments

Assignments Team ITU

- Theresa will write the meeting report
- Implementation of the last requirements this week (afterwards only bug fixing, testing and writing the report)
- Update the JAR-file with the prediction part and the new requirements and send it to Team Kenya
- Write an status update on the day of the next regularly meeting (Tuesday, 3. Dec. 2013) and send it to Team ITU before the meeting
- Check and answer emails according to this project regularly in the next days and try to be visible on Skype as much as possible

Assignments Team Kenya

- Write down an exact definition of the ALL methods, which are needed on the server for the android app (till Wednesday, 27. Nov. 2013)
    - Team ITU will only implement requirements this week
    - Suggested structure: CommandName, Parameters (name, datatype), output, and description.
    - At least the suggested three methods (see Decisions/Discussions)
- Installation of the server application
- Write an status update on the day of the next regularly meeting (Tuesday, 3. Dec. 2013) and send it to Team ITU before the meeting
- Update of the android app mockups
- Summary on the prediction data
- Check and answer emails according to this project regularly in the next days and try to be visible on Skype as much as possible

Reminder

1. Dec. 2013	Finishing a first working prototype of the occupancy analyzer. After this Team ITU will concentrate on writing the mandatory report. Further minor improvements and testing will be made in parallel.
16. Dec. 2013	Hand-in of the report (End of project)

➔ **Just 5 days for finishing the prototype!**

Global Meeting Report 03.12.2013

Team ITU

Date	Time	Duration	Attendees	Communication medium
Tuesday, 03.12.2013	17:10pm (CET) 19:10pm (EAT)	~ 2:00 h	ITU Team (3 total): • Tomas • Christoffer • Theresa Kenya Team (1 total): • Cecil	Skype

Agenda: Status-Update

Agenda (planned):

1. Discuss Status Update Kenya
2. Discuss Status Update ITU
3. Strathmore university’s interest in an occupancy analyzer
4. Server
5. Assignments for this week

Meeting progress

- Started at around 17:10 CEST (19:10 EAT)
- ITU student (Jacob) was excused
- Kenya student (Ann) was missing
- Kenya student (Wayua) is not in this project any longer

Status update from Team Kenya

- User interface is done
- The android app is not done; still can’t get data from the server due an exception
- Started to work with the Raspberry PIs
- Currently testing with dummy data
    - Instead of UUID normal IDs
    - Integers as coordinates

Status update from Team ITU

Report:

- Started to write the mandatory report

Server:

- Implemented the server methods “GetAvailableRooms” and “CheckPredictions”
- Added additional table in the DB for storing the statistical information
- Methods are not tested and not ready yet (server-side)

Strathmore university’s interest in an occupancy analyzer

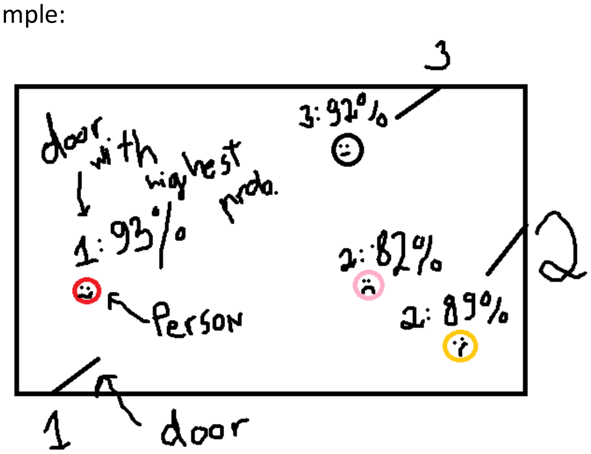
- The student center is called the 'green building' in terms of green thinking concept of environmental conservation
- The idea has been to use IT to enhance conservation by monitoring usage of e.g. water, lights etc.
- The school management has not yet been convinced enough to buy into the idea
- The department (which the MSC program is under) has taken this initiative to work with interested students to achieve this noble idea
  - this will be achieved in the beginning through research and the collaboration with ITU (this project)
  - it’s at its initial stages
- In addition: earlier in the year some computers were stolen in the same building, so there is need to improve surveillance

Decisions/Discussions

- 1. Suggestion for prediction data for the android app (not chosen!)
  - The server method “checkPrediction” should return something like this: for each person in the room there will be a coordinate of an exit, if the probability for this person and exit is over 60%
  - When the probability for an exit is less than 60%, return nothing
  - The server is doing the logic (which exit has a probability over 60%), because that would be better for the performance of the android app
  - In JSON something like this (just an abstract example!):

```
roomID: {
       objectlist: {
              person1: {exit1: {x:435, y: 644, timestamp:
       3753786523978}}
              person2: {}
              person3: {}
       }
}
```

- 2. Suggestion for prediction data for the android app (chosen)
  - The server method “checkPrediction” should return something like this: All the probabilities for each person and exit
  - Example:



  - In JSON something like this (just an abstract example!):

```
roomID: {
	objectlist: {
		person1: { exit1:0.93, exit2: 0, exit3: 0.07}
		person2: {exit3: 0.92, exit1: 0.05, exit2:0.03}
		person3: {exit2: 0.82 ...}
		person4: ...}
	}
}
```

  - The android app chooses the highest probability for each person (or chooses the probabilities which are over 60%)
  - The border of 60% can then be also configured on the android app
- Second suggestion is much more flexible, because the android app doesn't have to rely on the server for future functionality; The app can request all the prediction data and can decide what to do with it
- Team ITU will check if the preferred suggestion can be implemented and how the JSON will look like

Instructions for setting up the server

- The following exception means that the server application cannot connect to the MySQL database

```
C:\wamp\www\GSDServer-2013-11-22\GSDServer>java -jar GSDServer.jar
Runner: Starting.
Runner: Shutting down
Exception in thread "main" com.mysql.jdbc.exceptions.jdbc4.CommunicationsExcepti
on: Communications link failure

The last packet sent successfully to the server was 0 milliseconds ago. The driv
er has not received any packets from the server.
        at sun.reflect.NativeConstructorAccessorImpl.newInstance0(Native Method)

        at sun.reflect.NativeConstructorAccessorImpl.newInstance(Unknown Source)

        at sun.reflect.DelegatingConstructorAccessorImpl.newInstance(Unknown Sou
rce)
        at java.lang.reflect.Constructor.newInstance(Unknown Source)
        at com.mysql.jdbc.Util.handleNewInstance(Util.java:411)
        at com.mysql.jdbc.SQLError.createCommunicationsException(SQLError.java:1
117)
        at com.mysql.jdbc.MysqlIO.<init>(MysqlIO.java:355)
        at com.mysql.jdbc.ConnectionImpl.coreConnect(ConnectionImpl.java:2461)
        at com.mysql.jdbc.ConnectionImpl.connectOneTryOnly(ConnectionImpl.java:2
498)
        at com.mysql.jdbc.ConnectionImpl.createNewIO(ConnectionImpl.java:2283)
        at com.mysql.jdbc.ConnectionImpl.<init>(ConnectionImpl.java:822)
        at com.mysql.jdbc.JDBC4Connection.<init>(JDBC4Connection.java:47)
        at sun.reflect.NativeConstructorAccessorImpl.newInstance0(Native Method)

        at sun.reflect.NativeConstructorAccessorImpl.newInstance(Unknown Source)

        at sun.reflect.DelegatingConstructorAccessorImpl.newInstance(Unknown Sou
rce)
        at java.lang.reflect.Constructor.newInstance(Unknown Source)
        at com.mysql.jdbc.Util.handleNewInstance(Util.java:411)
        at com.mysql.jdbc.ConnectionImpl.getInstance(ConnectionImpl.java:404)
        at com.mysql.jdbc.NonRegisteringDriver.connect(NonRegisteringDriver.java
:317)
        at java.sql.DriverManager.getConnection(Unknown Source)
        at java.sql.DriverManager.getConnection(Unknown Source)
        at gsdserver.database.DbHelper.getConnection(DbHelper.java:175)
        at gsdserver.database.DbHelper.createDatabase(DbHelper.java:73)
        at gsdserver.Runner.main(Runner.java:35)
Caused by: java.net.SocketException: Permission denied: connect
        at java.net.DualStackPlainSocketImpl.connect0(Native Method)
        at java.net.DualStackPlainSocketImpl.socketConnect(Unknown Source)
        at java.net.AbstractPlainSocketImpl.doConnect(Unknown Source)
        at java.net.AbstractPlainSocketImpl.connectToAddress(Unknown Source)
        at java.net.AbstractPlainSocketImpl.connect(Unknown Source)
        at java.net.PlainSocketImpl.connect(Unknown Source)
        at java.net.SocksSocketImpl.connect(Unknown Source)
        at java.net.Socket.connect(Unknown Source)
        at java.net.Socket.connect(Unknown Source)
        at java.net.Socket.<init>(Unknown Source)
        at java.net.Socket.<init>(Unknown Source)
        at com.mysql.jdbc.StandardSocketFactory.connect(StandardSocketFactory.ja
va:259)
        at com.mysql.jdbc.MysqlIO.<init>(MysqlIO.java:305)
```

- This is due to the server hasn't been set up correctly

- Instructions for solving this problem: http://stackoverflow.com/questions/2983248/com-mysql-jdbc-exceptions-jdbc4-communicationsexception-communications-link-fai
- The following lines show how it should look like in the cmd, when everything is set up correctly:

```
Runner: Starting.
RaspberryInterpreter-Server: Started.
LeacherInterpreter-Server: Started.
LeacherInterpreter-Server: Listening.
RaspberryInterpreter-Server: Listening.
```

Assignments Team ITU

- Theresa will write the meeting report
- Will check on the preferred solution for the prediction data method and will send Team Kenya how the JSON will look like
- Finishing Implementation of the last requirements and testing
- Update the JAR-file and share it with Team Kenya
- Write a status update on the day of the next regularly meeting (Tuesday, 10. Dec. 2013) and send it to Team ITU before the meeting
- Check and answer emails according to this project regularly in the next days and try to be visible on Skype as much as possible

Assignments Team Kenya

- Set up the server correctly and installation of the server application. If there are any other problems occurring, contact Team ITU.
- Write a status update on the day of the next regularly meeting (Tuesday, 10. Dec. 2013) and send it to Team ITU before the meeting
- Check and answer emails according to this project regularly in the next days and try to be visible on Skype as much as possible

Reminder

16. Dec. 2013 Hand-in of the report (End of project)

➔ Just 12 days for finishing the report (end of project)!

B.6 Emails

B.7 Survey

Todo list

Some note or other.

Another note.

And another one.

Add my picture here.