Westfälische Wilhelms-Universität Münster

ifgi
Institute for Geoinformatics
University of Münster

Master Thesis

Extracting Indoor Map Data From Public Escape Plans On Mobile Devices

Georg Tschorn

Supervisor: Prof. Dr. Christian Kray
Co-Supervisor: Klaus Broelemann

Institute for Geoinformatics, University of Münster

25th March 2013

Abstract

Due to the non-applicability of GPS devices and the higher need for accuracy, collecting indoor spatial data presents different challenges than mapping outdoor environments. Considering usability or technical requirements, existing approaches for data capture indoors are poorly suited for non-professional collaborators of volunteered geographic information (VGI) communities, such as OpenStreetMap (OSM). This work investigates an alternative to measuring what has been measured already. The idea is to extract map data from public escape plans by means of computer vision. An approach to automatically interpreting photos of escape plans on mobile devices is developed and evaluated.

Contents

Figures

Tables

1 Introduction

For a growing number of location-based applications indoor spatial data is an important requirement. However, opposing to outdoor spatial information, no centralized body or agency is charged with collecting or maintaining spatial information about indoor environments [1]. The amount of indoor data available in a consistent format is very low. Considering the success of volunteered geographic information (VGI) approaches, such as OpenStreetMap (OSM), for outdoor data, it is conceivable to adapt or extend such approaches to indoor environments. Although there already exists a detailed proposal for mapping building interiors using the OSM data structure [2], only a small number of indoor maps has been added to the OSM repository yet.

To circumvent the problem that GPS is no solution for indoor environments, various alternative approaches for mapping building interiors have been developed. But in terms of usability they can not compete with their outdoor siblings. Especially, since VGI communities usually consist of non-professionals to a large proportion, this is problematic.

However, one potentially fruitful source of indoor information can be found in most public buildings, mounted on walls. Escape plans contain architectural information such as footprint, doors and stairs, but also semantic information (e.g., room name or usage). The provision of escape plans is required by law due to location, extent and use of a building. Furthermore, their representation is standardized to some degree. Considering this, an automatic solution for extracting map data by means of computer vision appears to be feasible.

In this work, an approach for extracting basic indoor map data from images of public escape plans is developed, implemented prototypically on mobile devices, and evaluated. The aim is to find out if extracting indoor map data from escape plans is a feasible task to be performed automatically on mobile devices. First, the typical content and representation forms used are analyzed. Then, the prototype, developed based on the results of this analysis, is tested and evaluated, focusing on the technical feasibility of the approach. To get a first insight in the ability and willingness of people to use such a system, an exploratory user study is performed subsequently.

2 Background

This chapter provides the reader with the required background. Firstly, an introduction to laws and norms concerning escape plans is given. This is followed by a section on the most important image processing methods referred to in the course of this thesis. Thirdly, existing techniques for indoor mapping and relevant approaches making use of them are presented.

2.1 Escape Plans

Evacuation and escape plans can be found in most public buildings. In Germany, their provision is required by law due to location, extent or use of a building [3]. They need to contain graphical representations of the building outline, escape routes, location of first aid facilities, fire protection equipment, collecting points and the current location.

The representation of these properties is normed. Currently most escape plans follow the DIN 4844-3 [4], which exists since 2003. However, in 2010 a new international standard, the DIN ISO 23601 [5], was published. Until now, it is not required to follow the new standard. But it is intended to replace the DIN 4844-3 by the DIN ISO 23601, so that in future all escape plans will have to follow the new standard. Until then, both standards may be used.

According to DIN 4844-3, escape plans basically consist of the seven different colors signal white, signal black, signal red, signal green, signal yellow, brighter green and darker green (Figure 1). Signal white is used for the background of the plan. Texts, the floor plan or labels within the floor plan have to be signal black. Signal red is used for symbols indicating fire protection equipment. Signal green is used for symbols indicating first aid equipment or the directions of escape routes. Escape routes must be in a brighter green and vertical escape routes (e.g., stairs) must be in a darker green. To indicate the current location, a signal yellow symbol with a black border is used.

Figure 1: Escape plan according to DIN 4844-3 (from [51]).

Other restrictions exist in terms of scale and size. The scale of the floor plan should be 1:100. However, other scales are allowed as well, as long as all details are well perceivable. The size of the escape plan should be at least A3. Though there are exceptions as well. The format A4 may be used in plans situated in, for example, classrooms or hotel rooms.

The presentation proposed by the new DIN ISO 23601 is similar (Figure 2). In terms of color there are two differences. The first is that instead of signal yellow, signal blue must be used for the sign indicating the current location. The second is that there is no distinction between vertical and horizontal escape routes. Both are represented in a brighter green.

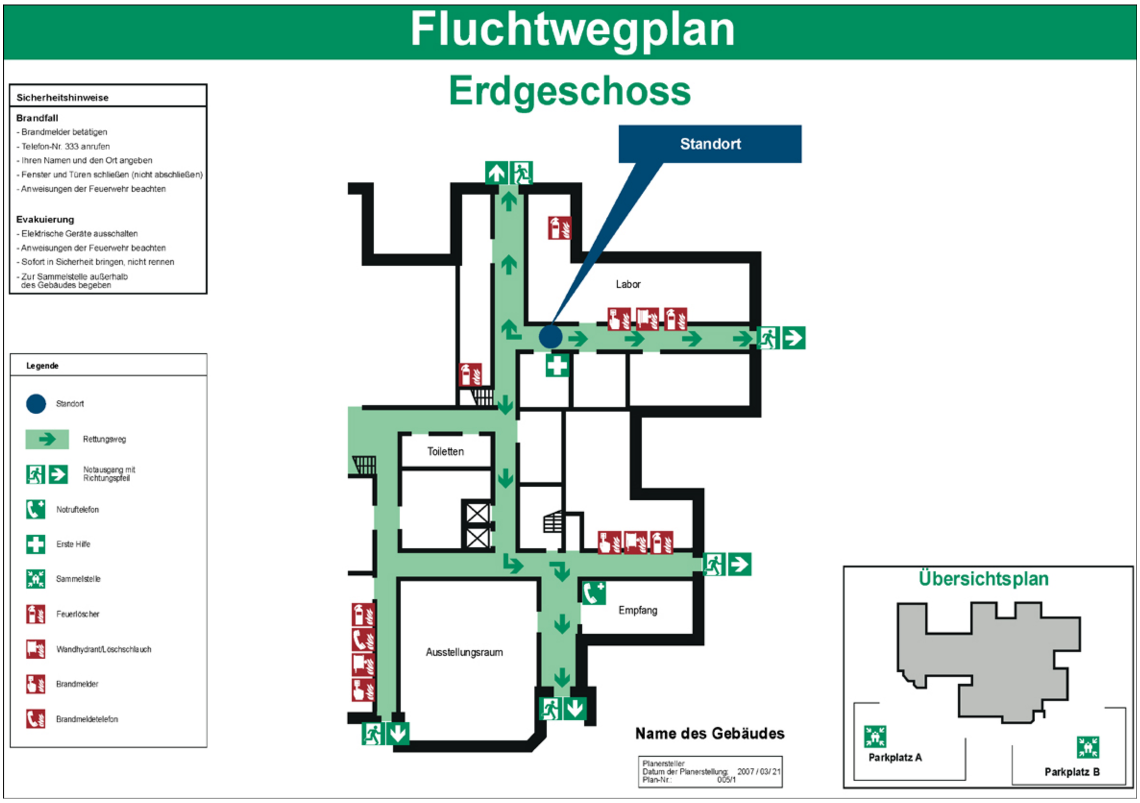

Figure 2: Escape plan according to DIN ISO 23601 (from [50]).

Also the rules for scale slightly differ. It has to be 1:250 for large buildings and 1:100 for small or medium buildings. For plans in single rooms (e.g., classroom) 1:350 may be used as well. Other changes occur for the emergency symbols. While the color is the same (white on background in signal red or signal green), most of the pictographs are slightly varying.

To sum it up, both standards insure a quite consistent way of representing relevant information for the case of emergency situations. However, they leave open space for individual solutions as well. Especially the regulations for the floor plan are vague. In principle any kind of floor plan drawing can be used for this part of the escape plan. Only the layout of the concerned building part, stairs and elevators need to be represented. Important elements should be highlighted and unnecessary elements should be removed. But there are no restrictions on how to draw particular elements and there is no definition of what is important or what is unnecessary.

Another consistency gap arises from the coloring conventions. The terms signal white, signal black, signal red, signal green, signal yellow and signal blue all refer to fixed color tones. In contrast to this, there are no fixed color tones for horizontal and vertical escape routes. The notions *brighter green* and *darker green* are used. But they do not refer to strictly defined colors.

2.2 Technical Background

This section gives an introduction to the most important image analysis methods and color representation forms referred to in this work.

2.2.1 Color Representation

The most common form of representing the color of one pixel is the RGB color space. Here, three values are used to specify colors by their red, green and blue components. However, for certain color image analysis tasks, representation forms that explicitly indicate hue, saturation and intensity (or brightness) are more suitable [6]. This is the case when particular objects should be identified by their typical color, as it is required later in this work. A color space providing this characteristic is the HSV color space. It consists of the three channels hue (H), saturation (S) and value (V), in which value specifies the intensity or brightness. If the task is to search for pixels with a certain color (e.g., for red), this can be done by analyzing the hue channel and selecting all pixels with hue values within a certain range. Compared to this, the same task performed in the RGB color space is far more complex. All three channels need to be observed, since they all affect the color tone perceived by human beings.

2.2.2 Morphological Operators

Morphological operators are introduced in [7]. Amongst many other applications, they can be used to process shapes in binary images or for noise removal. In this work four different morphological operations are used.

Dilation is applied to grow regions. In this paragraph, the example of dilation shall be used to explain the basic principle of morphological operations. Generally, they function by moving one structuring element step by step over each pixel of an image. Theoretically, the structuring element could have any shape. Figure 3 shows a structuring element of 3 by 3. In dilation, the structuring element is put with its center over each pixel and the value of the pixel is replaced by the maximum value that can be found in the area of the structuring element. In the case of a binary image, this can only effect pixels with a value of zero. If a zero pixel has a non-zero neighbor within the area of the structuring element, its value will be changed. Applied to all pixels, this results in a growing of the non-zero regions.

1	1	1
1	1	1
1	1	1

Figure 3: 3 by 3 structuring element.

Erosion is used to shrink regions. Instead of the maximum, the minimum value of the region covered by the structuring element is given to each pixel. Each non-zero pixel adjoining a zero pixel is set to zero. This results in shrinking non-zero regions. Furthermore, small regions may be deleted by the operator.

Opening is used to remove noise. It refers to applying erosion followed by dilation with the same structuring element. The effect of noise removal arises from the erosion that is executed initially. As stated before, small regions that might refer to noise (e.g., pixels without any neighbors), may be deleted by an erosion. After one region was deleted completely, it cannot be restored by the dilation performed subsequently.

Closing is used to fill holes. This is done by first dilating and then eroding. Small holes within non-zero regions may be filled completely by the dilation applied firstly, so that the erosion performed as second step has no effect on them. The holes can be considered as being closed.

2.2.3 Connected Components Analysis

Applied to binary images, connected components analysis refers to identifying and analyzing sets of connected non-zero pixels. There exist various algorithms to identify and label connected components, and to analyze certain properties and relations between them (e.g., [8–10]). Examples of features which can be analyzed and used for a classification are area, bounding box, minimal enclosing rectangle or shape.

2.2.4 Freeman Chain Code

Typically, connected components are represented by sets of points containing all coordinates of the component's contour. However, for some tasks it is more useful to convert those absolute coordinates into a form that indicates the relations between the contour points. The most common approach to do so is the Freeman chain code [11]. Instead of points, it consists of directions. For each pair of succeeding points, the direction from the first to the second point is computed and assigned to one of eight direction classes. Starting with direction 0 pointing to the top, followed by direction 1 pointing to the top-right, and followed by six more steps of 45 degrees, all possible directions are covered (see Figure 4). In other words, the classification scheme consists of all possible neighborhood relations two pixels of a raster image can have.

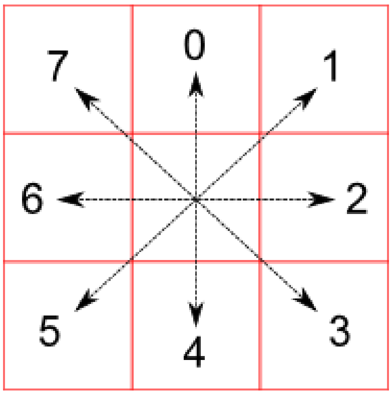

Figure 4: The eight directions of the Freeman chain code.

2.2.5 Chain Code Histogram

Freeman chain codes can be analyzed using the technique of chain code histograms [12]. They are obtained by filling a histogram consisting of one bin for each of the eight direction's frequencies. To achieve scale invariance, it can be normalized by deviation through the sum of all directions. If the sequence of points the chain code was created from includes successive points with a distance larger than one pixel, this needs to be considered by weighting the directions by the corresponding distances.

Chain code histograms are often used for shape recognition. Matches can be found by either comparing to a sample histogram or by checking certain predefined criteria for the proportions of the bins. Rotation invariance can be achieved by rotating the histogram and applying the matching criteria for each possible rotation. To rotate a chain code histogram by 45 degrees, its last bin simply needs to be moved to the first place, while all other bins move up one rank.

Besides being very fast and simple, the approach has the drawback that it does not account for the order. One distinct histogram may be obtained from two completely different shapes. Thus, it is restricted to be used in applications with relatively distinct classes [13].

2.3 Related Work

Within buildings it is not possible to rely on GPS. Therefore, various alternative approaches for indoor mapping have been developed. A rather lightweight approach applicable on mobile devices is to use trigonometric functions to calculate distances between corners of rooms. While the indoor environment is shown through the camera preview, the user can indicate when the device is pointing to a certain feature. Based on the input of orientation sensors, positions of corners can be computed. In both, [14] and [15] systems that rely on this method are described. The advantage of this approach the little computational costs and that it may work with hardware that almost every common smartphone provides. On the other hand, it is quite prone to inaccuracy and requires the effort to manually capture each corner of every room of a building.

Three-dimensional (3D) laser scanners provide much more accuracy. Based on point clouds, 3D indoor models can be reconstructed. Examples of systems that allow collecting indoor spatial data by this means constitute [16], [17] and [18]. The problem of such approaches is that laser scanners are quite expensive and thus hardly available to non-professionals.

A cheaper and more widespread alternative present depth cameras. [19] and [20] propose systems allowing for 3D reconstruction of building interiors using the consumer-grade range camera Microsoft Kinect. Compared to laser scanners, depth cameras have the disadvantage of being less accurate and having a field of view of around 60 degrees, while for laser scanners this usually is around 180 degrees. Furthermore, according to [21], sufficiently accurate measuring is only possible within a distance between 1 and 3 meters. Due to this, data collection is quite labor-intensive.

In absence of depth sensors, 3D point clouds may also be generated from image sequences. This can be done using the structure from motion pipeline. An approach to reconstructing building interiors, that makes use of this technique, is described in [22]. Alternatively, simultaneous localization and mapping approaches can be used. An example for this, which is reported to work on smartphones, can be found in [23]. The drawback of such approaches is their robustness. To reliably work, they require an environment that is sufficiently covered with features that can be used for matching the images.

However, the huge effort of measuring building interiors can be avoided by exploiting already existing information sources. A number of works dealing with automatic information extraction from architectural floor plans can be found in literature. In [24], [25], [26], [27], [28] and [29], approaches designed to generate 3D building models based on the information extracted from two-dimensional floor plans are described. Two more recent works, [30] and [31], solely focus on room shape detection. However, in these approaches as well, the problem is the robustness. While there exists a great variety of different representation standards, all of this works only address one particular or a restricted number of existing notions. Even more challenging is that architectural floor plans are usually property of the building owner. As such they are not publicly available. This is problematic for effectively collecting a large quantity of indoor spatial data by crowdsourcing.

In contrast to this, escape plans are open to the public. A first approach for extracting indoor map data from escape plans is described in [32–34]. First, the image is binarized by simple thresholding. Then, it is segmented into distinct regions by a connected components analysis. To identify symbols, the resulting regions are classified according to their aspect ratio and template matching is performed in regions with an appropriate aspect ratio. After identifying and removing all symbols, the remaining boundaries are skeletonized and the end points and branch points are identified. To extrapolate walls that are occluded by symbols, end points that are next to symbols are lengthened in the direction of the previous skeleton points. When the gaps are closed, a connected component analysis is performed again and the resulting regions are classified. Regions larger than a threshold are identified as rooms. To obtain a suitable threshold value, the transformation from image to world coordinates is computed beforehand. The approach assumes that a georeferenced version of the building outline is available from an external source. The transformation is obtained by matching the building outline found in the image with the georeferenced outline.

An improved version of the approach described previously [35] exploits the fact that signs can be distinguished from other plan elements by their color. To do so, first the color structure code algorithm [36] is used to cluster homogeneously colored regions. Subsequently, the symbol areas are detected using predefined thresholds for the colors in demand.

However, the robustness and accuracy of this approach were not investigated until now. Its applicability to the majority of escape plans and thus to the majority of public buildings is not proven yet.

3 Methodology

This work aims at investigating the feasibility of collecting indoor data by photographing escape plans. It is assumed that the basic prerequisite for this is a sufficient degree of standardization. Theoretically both, the old German DIN 4844-3 and the new international ISO 23601 standard provide a potentially useful backbone allowing for an automatic analysis. However, the most important question is how they are practically implemented. Therefore, the first step of this work was to find out which representation forms exist and which are most common.

Based on this, an approach exploiting the general characteristics of escape plans was designed. Instead of focusing on a few particular representation forms, the idea was to address properties that can be found in as many plans as possible.

The approach was implemented prototypically and evaluated subsequently, aiming to find out if it is possible to extract useful data from the majority of existing plans. This was assumed to be the case if the effort required to remove all errors in the resulting data sets is comparatively small.

The use case of the approach is seen as a tool that can be used by contributors of a VGI project. Therefore, to get an impression of the willingness and ability of potential users to work with the system, a usability study with ten participants was conducted.

3.1 Study on Representation Forms of Escape Plans

The existing norms for escape plans are fixing only a couple of properties. They still leave lots of space for individual solutions. To find out how this space is usually filled, 120 original escape plans stemming from 41 different buildings were analyzed. The goal was to find out what notions are commonly used, and to identify potentially exploitable extra information, not explicitly requested by the norms. It was expected that some general characteristics, preferably applying for as many plans as possible, and exploitable by the approach, could be identified. To do so, first all elements that represent spatial or other useful information were identified from the collection of plans. Then, for each identified element it was counted in how many of the plans it can be found and how often it is represented in a certain form.

Problematic for practical use is that most existing approaches for automatic analysis of architectural floor plans were designed to work with a restricted number of particular standards. Lots of them work under certain assumptions that may improve the performance. But potentially not all of them are true for a sufficiently large number of plans. For this reason, one more goal was to investigate the reliability of these assumptions for escape plans. In particular, this was the connection of the line strength to certain feature classes (e.g., thin lines are only used for symbols and not for walls), as assumed in [30]. Another assumption found in [26] was that thick lines are used to represent outer walls and thin lines are only used for inner walls. One further objective was to find out which notions exist to represent doors.

After it was found that the green tones for indicating escape routes are not explicitly defined, it was additionally decided to analyze the colors used for this features. This was done to find a general rule for distinguishing the three different feature classes depicted in green tones.

3.2 Prototype

After the escape plans were analyzed and some general characteristics were identified, a prototypical mobile application was developed (see Figure 5). It allows to extract room shapes, doors and the building outline. It was decided to focus only on these three features, since they are sufficient to create a coarse indoor map and can be found in every escape plan. As input the application needs to have a photo of an escape plan. The extracted data can be exported to an OSM XML file in the format proposed in [2] and further processed using the OSM editor JOSM[1]. The application was designed with the aim to support all layout types, common in Germany. Currently most existing escape plans follow the DIN 4844-3 standard. However, in future more and more plans will be in accordance with the international ISO 23601 standard. For this reason, plans according to both, the DIN 4844-3 standard and the ISO 23601 standard, are addressed.

1 http://josm.openstreetmap.de

Figure 5: Screenshots of the prototype. Top: the input image is loaded and the three parameters format, scale and doors need to be set. Bottom: extracted room shapes (yellow) and doors (red).

3.3 Evaluation

The resulting prototype was evaluated threefold. First to find out if the system works under typical conditions. Five escape plans were digitized under varying conditions. Three devices were tested for three types of lighting. One time, the plans were laminated. One time, uncoated paper versions of the plans were photographed. The resulting data sets were evaluated using the protocol described in [37]. Basically, the goal was to find out if the prototype works under all typical conditions. Moreover, it was investigated if particular settings provide better or worse results.

The second part of the evaluation was to investigate the prototype's overall performance. 30 original escape plans were digitized under real conditions. Again, the resulting data sets were evaluated using the protocol described in [37]. The aim was to find out if the system is able to extract data of sufficient quality from the majority of plans.

The third part was a small-scale user study. The ten participants had to digitize three original escape plans under real conditions using the prototype. The first one with guidance, the next two without. The aim was to find out if they are able to autonomously work with the system and to identify critical steps. Considering the important influence of the input image's quality to the result, one main goal was to find out if users generally proceed sufficiently accurate while photographing the escape plan. The resulting datasets of this test were evaluated using the protocol described in [37] as well. It was assumed that the less varying the results are, the more it can be relied on the ability of users to produce input images of sufficient quality. Moreover, the participants had to fill a questionnaire. The questions targeted to identify problematic steps and to get an idea of the people's interest to use such a system to collect indoor data.

4 Implementation

To realistically judge the feasibility of using mobile devices to automatically extract indoor map data from escape plans, an application was designed and implemented prototypically. Its source code and APK file are available on the attached DVD. Firstly, this chapter describes what requirements and limitations were determined beforehand. Then, the actual approach and its implementation is presented. The description starts with the requirements on the input image and is followed by the proposed processing pipeline. Furthermore, the data structure of the output and the most important parameters are explained. The last section is on implementational details, such as the platform and libraries used.

4.1 Requirements

The approach was designed to function with the most common types of escape plans identified before. It should allow to extract the shape of rooms, the building outline and the position of doors. It was found that there are many different representation forms in use. The methods to be chosen should address their general characteristics. Instead of focusing on a few particular representation forms, properties that can be found in as many plans as possible should be exploited.

While choosing the algorithms to be used, it needs to be considered that the application should run on mobile devices. Although the performance of smartphones has extremely increased over the last years, it is still limited. Therefore, using computationally expensive algorithms needs to be avoided.

Another requirement arises from the goal to use the application as a tool for contributors of a VGI community. To be attractive, the approach should not require too much user intervention. The algorithms to be chosen need to take this into account.

The idea is that the application is used to extract a coarse indoor map, which serves as starting point for the creation of a highly-detailed indoor map. Hence, the application's output needs to be in a data format that can be read by software that allows to easily further process the dataset.

4.2 Limitations

To identify what representation forms are commonly used, 120 escape plans were analyzed. Based on the results of the analysis and the restrictions found in the DIN standards, a number of assumptions, the approach can rely on, was determined. The following listing presents all preconditions that are exploited by the approach and need to be fulfilled to obtain useful results.

1. The floor plan is the largest connected element in the image.
2. Walls, doors and labels are depicted in black or gray.
3. Doors are represented by gaps, arcs or triangles.
4. Labels are not connected with other elements.
5. The background has the brightest color of the image.
6. Horizontal escape routes have the brightest green.
7. Vertical escape routes are generally larger than signs and indicated in green, but not bright green.
8. Elements that are not black, gray, bright green and no vertical escape routes must be signs.

Further requirements address the form and quality of the input image.

1. The image contains only the escape plan.
2. It has no perspective displacements.
3. There are no, or just little lighting differences.
4. All elements including thin lines are clearly distinguishable from the background.

Other limitations are concerning the user interface. The targeted use-case of the approach is seen as a tool to collect indoor map data for contributors of a VGI community. Certainly, it could only serve this purpose if it is easy to use. This requires a user interface that is well understandable. In principle, while working with a long processing pipeline consisting of many different algorithms, there can be a large number of parameters that need to be adjusted. Besides choosing the algorithms in a way that the number of such parameters is kept to a minimum, the user interface must allow to control the necessary parameters as easily as possible. However, this work focuses on technical issues. It is out of the scope to design an optimal user interface. The work on this aspect is limited to keeping the number of required parameters low.

4.3 Input Image

The prototype can load images from the following three sources.

1. The device's camera application to directly process photos just taken.
2. The image gallery to access pictures stored already.
3. The application CamScanner[2].

Since it is not always possible to photograph escape plans without perspective displacements or parts of the surrounding, it was necessary to provide image enhancement functions. For this reason the prototype can access the application CamScanner. It allows to crop images to remove the surrounding, not belonging to the escape plan. As well as perspective distortions and lighting differences can be removed. Providing these operations, escape plan images can be enhanced in a way that fulfills all requirements demanded by the prototype.

4.4 Processing Pipeline

The proposed approach can be subdivided into ten steps (see Figure 6). In the following sections each step is explained. It starts with downsampling the input image to increase the processing speed. Then, bilateral filtering is applied to smooth homogeneously colored areas. This is followed by separating the background and the foreground. Afterwards the building outline is detected and the superfluous surrounding is cropped. Using colors as main criterion, the foreground is segmented into the four classes *walls*, *sings*, *horizontal escape ways* and *vertical escape ways*. Then, knowing which pixels correspond to signs, the building outline is refined. In the next step, text elements are removed from the walls mask. After that, gaps within walls, that may result from removing signs, are detected and closed. Subsequently, doors are detected and possibly occurring door symbols are replaced by simple lines. The last step is detecting rooms in the refined walls mask.

2 https://play.google.com/store/apps/details?id=com.intsig.camscanner&hl=de

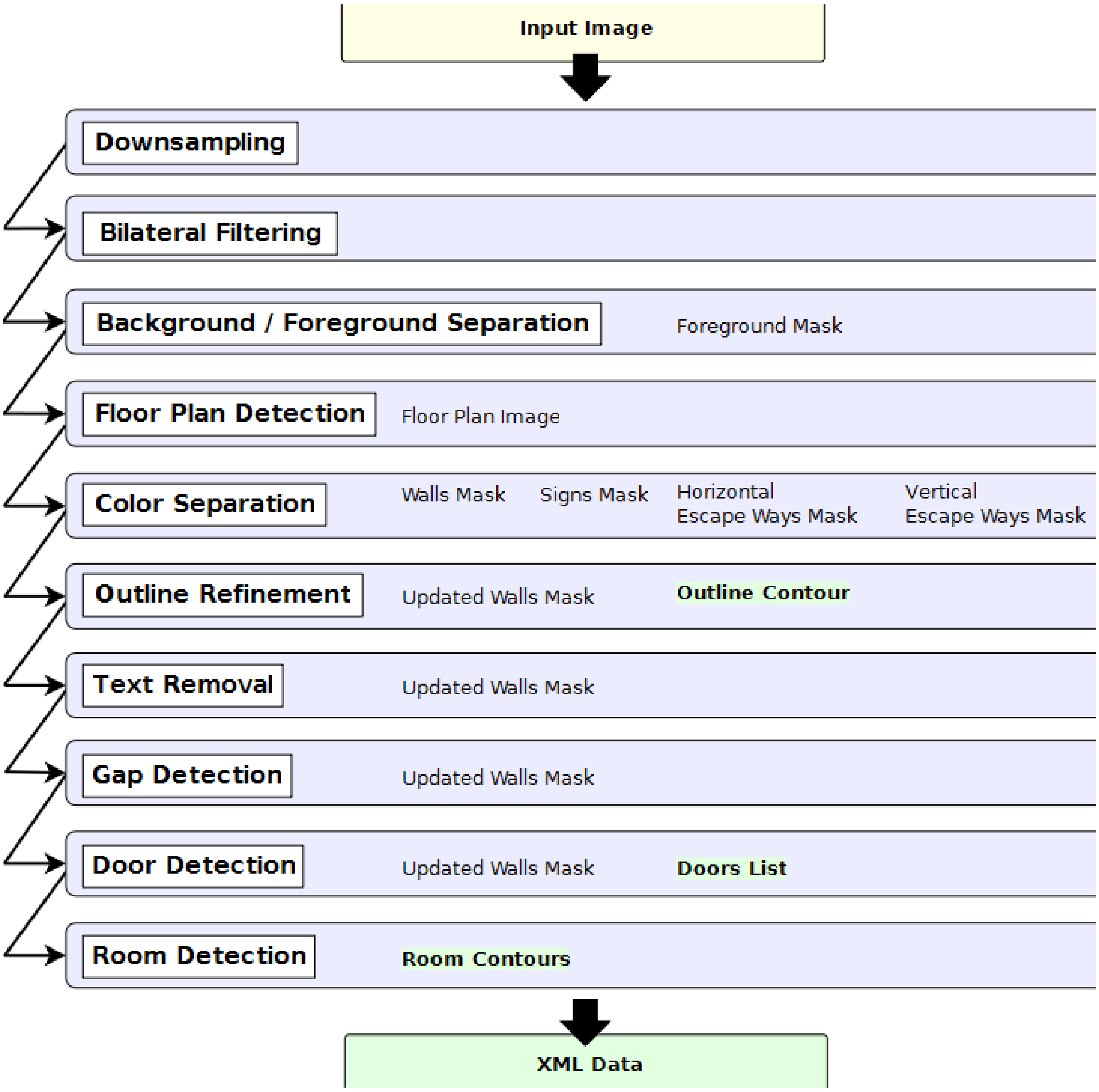

Figure 6: The proposed processing pipeline.

4.4.1 Downsampling

The first step of the processing pipeline is to downsample the input image. This is done by first smoothing the image with a Gaussian kernel and then removing every second row and column. The resulting image is four times smaller than the original. This increases the speed of the subsequent processing steps.

While this operation considerably reduces the amount of data to be processed, it only results in a very small loss of information. The Gaussian smoothing ensures that even small lines can not get lost. The only potentially critical effect is that features, which were separated before, may touch each other afterward. However, it was observed that performing this operation only once, normally has no considerably negative effect to the result.

4.4.2 Bilateral Filtering

After the image is downsampled, bilateral filtering is applied. Compared to other methods, this one comes with the advantage of being edge-preserving and not producing phantom colors [38]. The method only averages together perceptually similar colors. This is important for the results of the color separation.

While in other works the color structure code (e.g., in [35]), or mean shift segmentation are used to cluster homogeneously colored regions [39], here it was decided not to use neither of them. In contrast to bilateral filtering, both, color structure code [40] and mean shift segmentation [41], were observed to produce fake colors at transitions between two colors. For instance, while experimenting with the two algorithms, it was observed that they formed larger gray transition regions between colored signs and the white background. This is problematic since these regions might be incorrectly classified as walls subsequently.

4.4.3 Background / Foreground Separation

The separation of foreground and background is applied by thresholding the image. The image is converted to a grayscale image. All pixels with a value below the threshold value are considered as foreground. To obtain an appropriate threshold value, the assumption is made that the background is the brightest and at the same time the largest part of the image. The histogram of all values of the grayscale image is computed and analyzed. The minimum before the last peak is chosen to be the threshold value.

It was tried to use Otsu's thresholding method [42] as well, which is a standard method to obtain threshold values for images containing two different classes of pixels. However, the results were not satisfying. The threshold obtained tended to be too low, resulting in misclassifications of bright foreground elements as background. Moreover, adaptive thresholding (according to [43]) was tried. But here problems were observed, when the image contained large non-white (i.e. foreground) areas. In such cases, rather bright foreground elements, such as door symbols, often were falsely assigned to the background.

4.4.4 Floor Plan Detection

The method used for detecting the floor plan works under the assumption that the border of the floor plan is one connected foreground component, which is true for almost all escape plans analyzed in this work. Furthermore, it is assumed that the sought-after connected component is the one with the largest area. But since most escape plans have some kind of frame, which encloses the whole plan, further criteria were added. The sought-after connected component also needs to be smaller than a certain percentage[3] of the whole image. Additionally, its width and height have to be smaller than a certain percentage[4] of the width, respectively the height of the input image. When the floor plan is detected, its surroundings are removed and the image is cropped to the floor plan area's bounding box. This way, the speed of the following processing steps can be increased.

4.4.5 Color Separation

The image is separated into the four different color classes *black*, *darker green*, *brighter green* and *other colors*. The class *black* includes black and gray elements. Usually this should be the actual floor plan and labels. The class *dark green* contains vertical escape routes. The class *bright green* contains horizontal escape routes. The fourth class, *other colors*, could be any other color not included in the other three classes. Basically, this should be red, green, yellow or blue elements referring to emergency symbols or *you are here* symbols. For better understanding, later it will be referred to the four classes as *walls*, *vertical escape routes*, *horizontal escape routes* and *signs*.

The image is converted from RGB to HSV color space. All foreground pixels become assigned to one of the four classes. First, the pixels for the *signs* class are detected. This is done by looping through all pixels and selecting the pixels with a saturation and brightness higher than a certain threshold[5]. However, due to noise, some pixels situated on black elements, can fulfill these criteria as well. Therefore, a connected components analysis is performed and only components large enough to be signs are kept. In addition, small holes are filled, since in some cases they contain black elements (e.g., black pictographs on *you are here* symbols) that should not be mixed with other *wall* elements. The resulting mask is dilated to cope with signs that are surrounded by black contours, which again, should not be mixed with other *wall* elements.

3 For the prototype 80 % was used.

4 For the prototype 95 % was used.

5 For the prototype a minimal saturation of 100 and a minimal brightness of 70 were used.

The next step is to detect dark green and bright green elements. This is challenging as the common norms are not defining dark and bright green in an exact way. For this reason, in some escape plans, the brightest green is even darker than the darkest green of others. Additionally, this highly depends on device, camera settings and lighting conditions. To deal with this, pixels with a hue value that matches green and sufficiently high saturation and brightness values are selected initially. Subsequently, two thresholds are defined to further classify this set of green pixels. One threshold for the saturation values is obtained using Otsu's method [42], giving the saturation values for all green pixels as input. A second threshold value is computed in the same way for the brightness. All pixels with a brightness value higher than the second threshold are assigned to the *bright green* class. All pixels with a saturation value higher than the first threshold and a brightness value lower than the second threshold are assigned to the *dark green* class. Pixels with a saturation lower than the first and a brightness lower than the second threshold are neglected. Due to their low saturation and brightness, they have a high chance to belong to black elements.

Just as with the pixels classified as *other color*, some dark or bright green pixels could be situated on elements that belong to different classes. Therefore, here as well some structural constraints must be fulfilled. For each of the two greens, a connected components analysis is performed and only areas that are large enough are kept and small holes are filled.

Until now, only bright green and dark green are distinguished. As mentioned before, escape plans also include signal green, which is used for certain emergency signs. However, on many plans, the used dark green is very similar to signal green. Due to this, it is hard to distinguish green elements just by color. But it can be assumed that escape routes usually cover larger areas than most emergency signs. For this reason, for the minimal area size of *bright green* and *dark green,* a higher threshold than for the *signs* class is chosen. The *signs* class initially contains all colored elements, including those from all three green tones. After the *bright green* and *dark green* elements are detected, they are removed from the *signs* class. Green elements that are too small for being escape routes, are assumed to be emergency signs. They remain in the *signs* class.

All foreground pixels not belonging to *signs* or *escape routes* are assumed to be black or gray. Consequently, they are assigned to the *walls* class.

4.4.6 Outline Refinement

During the floor plan detection, it is not known if a certain foreground pixel refers to a wall or to another feature class. In plans where emergency symbols are situated outside of the actual building outline, they could be falsely classified as exterior wall segments. Due to this, it may need refinements. This is done by first morphological eroding the area of the floor plan. The erosion is necessary to reduce the likelihood for errors. At the borders of the foreground, pixels with just very little saturation can be found frequently. For this reason, the color classification is more error-prone in these regions. By shifting the contour more inside, one can circumvent this problem. Finally, a loop through the shrunken preliminary outline contour is performed. For each point, is checked if it belongs to the walls layer. If this is not the case, the point is removed from the contour.

4.4.7 Text Removal

Separating text from graphics is a well explored task in literature (e.g., [44–47]). In most approaches first the connected components are analyzed. Subsequently, to obtain suitable thresholds, statistics about size, aspect ratio or density are computed. These thresholds are used to segment images into one text and one graphics layer. Moreover, some algorithms deal with the problem of retrieving text components overlapping with graphic elements.

However, it was observed that in most escape plan images, text components can be identified using very simple criteria. To not unnecessarily increase the processing time, a lightweight approach was chosen. It is assumed that text components are small elements not connected with other elements, which is true for most of the plans analyzed. Following this assumption, a connected components analysis of the walls layer is performed. All components with a sufficiently small height and a small width are classified to be text components or other superfluous information and removed subsequently.

4.4.8 Gap Detection

In some escape plans, walls are occluded by emergency signs. Thus, it is necessary to fill the gaps that arise when signs are removed. In this step, wall ends that touch a sign are detected, and extended until they meet another wall. In [32], this is done by first skeletonizing the walls and then identifying end points hitting signs in the vectorized wall skeleton. But this approach has the disadvantage that skeletonization might be fairly time consuming if the image contains thick walls. Furthermore, as reported in [48], skeleton-based methods tend to be very sensitive to noise. For this reason, an alternative approach was chosen.

It starts with converting the contours of the pixels classified as walls to the Freeman chain code [11]. Then, for each contour, the neighborhood of each chain code element is analyzed using two local histograms (similar to [13]). One for the preceding and one for the following chain code elements. The idea is to identify wall ends by their characteristic of being situated between two contour sections of opposing directions (e.g., direction 0 followed by direction 4, as can be seen in Figure 7). Exploiting this, it is checked if the two histograms fulfill the following criteria.

- More than 20% of all elements in histogram A have a direction of 0.
- More than 20% of all elements in histogram B have a direction of 4.
- The difference between the percentages of all elements with direction 0 in histogram A and all elements with direction 4 in histogram B is smaller than 10.

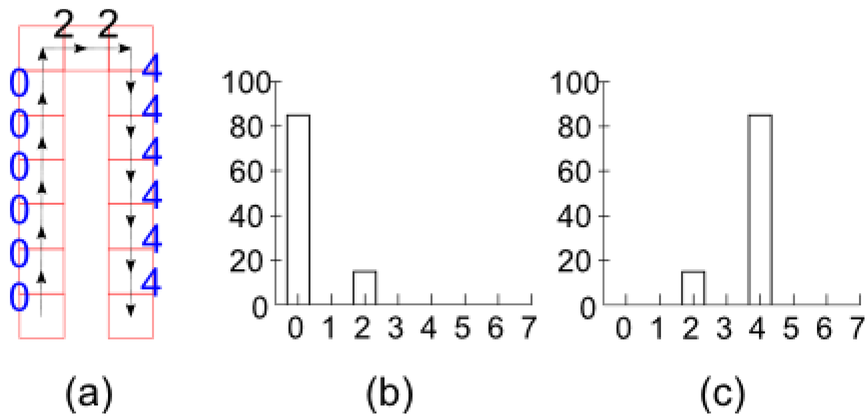

Figure 7: Freeman chain code at wall end (a) and referring chain code histograms (b) and (c).

If not all criteria are fulfilled, the histograms are rotated by 45 degree and the three criteria are checked again. This is repeated until all criteria are fulfilled or all eight possible rotations have been checked. If a rotation is found where all criteria are fulfilled, it is clear that this part of the contour constitutes either a convex or a concave curve. However, the interest lies in convex curves. To see if the curve is convex, the center of the curve is computed and it is checked if it is inside or outside of the contour. Then, two cases have to be distinguished. The contour could either be enclosing a wall or a set of connected walls. But it could also be enclosing a hole, surrounded by a set of walls.

Thus, the curve is convex if the center is inside the contour and the contour encloses a wall, or if the center is outside of the contour and the contour is a hole. If the curve is identified as being convex, it needs to be ensured that it encloses a wall end. For this reason, it is checked if the center is on a black pixel (i.e. on a wall). This is necessary because there can also be convex curves that constitute other features than wall ends (e.g., door symbols).

After all wall ends are detected, for each of them is checked if they touch a sign. This is the case if the first pixel after the wall end is part of a sign. The wall is extended until it touches another wall or a certain distance is exceeded. While extending the wall pixel by pixel, those that belong to signs are counted. If another wall could be reached before a threshold distance is exceeded, and more than 60 percent of all pixels passed belong to signs, the so created new wall is kept and the gap is closed.

4.4.9 Door Detection

In most literature found, door detection refers to detecting symbols consisting of an arc followed by a straight line or vice versa (e.g., [27], [30] and [49]). It was observed that the architectural drawings included in escape plans often use this symbol as well. However, at the same time, triangles or gaps are used in many plans. To cope with this, three different methods for door detection are used by the prototype.

The first addresses plans that represent doors by gaps within wall lines. It is based on the method used for detecting gaps that result from removing signs. In the prototypical implementation, the detection of gaps at signs and gaps representing doors is done in one step. When the wall ends are already detected, and all gaps resulting from sign removal are closed, the remaining gaps that have an appropriate size are considered to be doors.

Another method addresses both, the arc door symbol and the triangle door symbol. It exploits the fact that the Freeman chain code and local chain code histograms are already computed for the gap detection. The already calculated histograms can be reused, but in this step they are analyzed in a different way.

Comparing the chain codes of arched and triangular door symbols, it is observable that their properties are similar. As indicated in Figure 8, both have the property that the number of chain code elements with a direction of 1 is similar to the number of chain code elements with a direction of 4. This fact is exploited in the analysis of the two histograms. It is checked if the following criteria are fulfilled.

- More than 33% of all elements in histogram A have a direction of 1.
- More than 33% of all elements in histogram B have a direction of 4.
- The difference between the percentages of all elements with direction 1 in histogram A and all elements with direction 4 in histogram B is smaller than 15.

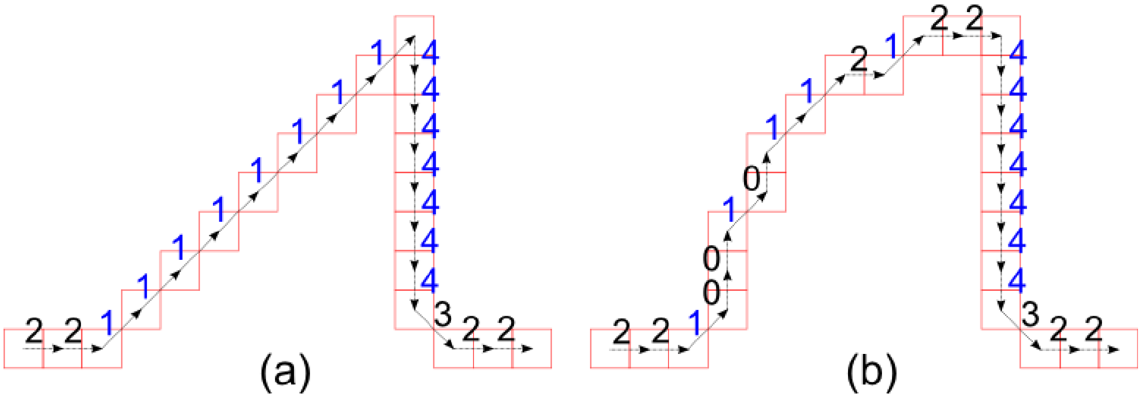

Figure 8: Freeman chain code of (a) triangle door symbol and (b) arc door symbol.

Certainly, the critical values used here might be further optimized. But they have shown to be accurate enough to work sufficiently well. The histograms are rotated by 45 degree and the three criteria are checked again if not all criteria are fulfilled. This is repeated as well until all criteria are fulfilled or all eight possible rotations have been checked. If a rotation is found where all criteria are fulfilled, this part of the contour is considered as a potential door symbol contour. However, depending on the symbol size, it is possible that the actual door symbol is not completely covered by that contour part, or that it covers more than the actual door symbol contour. Hence, it needs to be refined by finding its actual start and end points. This is done in two steps. First, the top point T is detected by looping all points of the preliminary door symbol contour. For each point, the sum of the distances to the first and to the last point of the contour are computed. The point where this sum has its maximum, is chosen to be the top point T. Subsequently, the optimal start and end point are computed. This is done by finding the two points A and B that maximize the distances from A' to the line AB and from B' to the line AB (see Figure 9). The algorithm starts with selecting A' as the first left neighbor of T, which has a distance of more than 1/6 of the contour's length. B' is selected as the first right neighbor of T, which has a distance of more than 1/6 of the contour's length. Temporally, A is chosen to be the first left neighbor of A' and B is chosen to be the first right neighbor of B'. The distances from A' to the line AB and from B' to the line AB are computed. Then, alternately A and B are shifted away from T. The procedure stops when the distances from A' to the line AB and from B' to the line AB reach their local maxima.

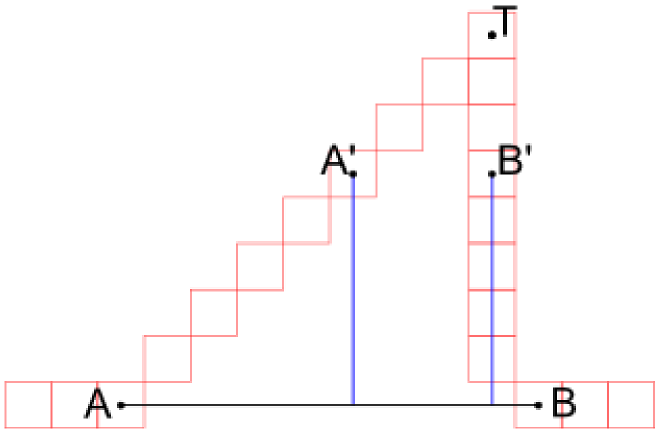

Figure 9: Refinement of the door symbol contour. The distances from A' to the line AB and from B' to the line AB need to be maximized (blue lines) by alternately shifting the points A and B away from T.

For detecting a door symbol, the so obtained optimized contour needs to fulfill two further criteria. Just as with the wall ends, the center of the contour part must be inside the contour if the contour encloses a wall. And the center must be outside of the contour if the contour is a hole. Finally, it is checked if the center is a black pixel. If this is not the case, a door symbol is detected. This last step is necessary to distinguish door symbols from wall parts that have a door symbol-alike shape. The detected door symbol is replaced by a line from the point A to the point B, as detected before.

The last of the three door detection methods is useful in cases where the others fail. For example, when a door symbol is interrupted for some reason. It takes advantage of how escape routes are represented. Often this is done by completely filling the rooms passed with dark green, respectively bright green (e.g., in the plan of Figure 2). For the most part, those areas colored in green directly border on walls. But where doors can be found, they adjoin white areas. The proposed method assumes doors wherever a green area touches a white area, while, at the same time, the border between the two areas is one straight line, not exceeding a certain length. In the prototype, this is realized by morphological dilating the green areas, followed by using the pixel-wise AND operation for a mask of all green and a mask of all white areas. After that, a connected components analysis of the resulting mask is done. For each obtained component, the smallest possible encompassing rectangle is computed. Components that have a rectangle with an appropriate aspect ratio and length are classified as doors.

This method is not applicable for buildings with escape routes through very wide rooms. If the green area of an escape plan does not fill the whole room, it has borders that do not touch walls or doors. This is why only components with a straight line-alike shape and a sufficiently short length are chosen. If the other two methods are used successfully before, this step is not necessary. However, since the other two methods may fail in some situations, applying this extra step generally decreases the chance of missing to detect particular doors.

4.4.10 Room Detection

After all symbols are removed, all gaps are closed and all door symbols are replaced, the actual room detection is a rather simple step. It can be done by a connected components analysis. The only thing of importance is to distinguish between rooms and smaller elements, such as stairs. As in [35], this is done by the size. Only components that are large enough can be rooms. In [35], an appropriate threshold is obtained by georeferencing the plan, so that the scale is known and it is easy to choose a realistic threshold value. However, in the present approach, the scale is estimated with the help of the scale restrictions given by the representation norms. This way it is independent of external information sources.

4.5 Output

The output of the prototype is an OSM XML file in the format proposed in [2]. The file is stored on the SD card and may be readily opened and further processed by the OSM editor JOSM. According to the proposed schema, each output dataset consists of one building relation object, one level relation object, one way object representing the building outline, several other way objects representing rooms and a number of node objects. Since the data is not georeferenced, a local reference system is used. It is obtained by using the image's pixel coordinates and scaling them by a factor estimated according to the scale restrictions given by the representation norms. Using JOSM, the building object can be georeferenced by moving it to the correct location and scaling and rotating it appropriately.

4.6 Parameters

As mentioned before, one objective of this work is to keep the number of parameters that need to be adjusted for each plan small. Many important parameters depend on scale (e.g., minimal room size). The fact that both common representation norms include regulations for the scales to be used can be exploited to estimate a transformation factor. Generally, all escape plans must be at least as large as DIN A3. The only exception are plans in single rooms (e.g., classrooms), which may fit DIN A4 as well. Additionally, the scale should be 1:100 if possible. If this it not possible due to the building's size, the rule to be followed depends on which DIN standard is used. According to the old DIN 4844-3, this may be any other scale that ensures all requirements to readability. The new DIN ISO 23601 is more precise. If 1:100 is not applicable, it requires to use 1:250. Furthermore, in single rooms, 1:350 may be used as well.

On that basis, it is possible to approximate the scale with just two input values the user needs to choose. The first value is the format. The prototype allows A4, A3, A2 and A1. The second value is the scale, where two values are possible, large or small. It is assumed that users are able to distinguish between the two scales 1:100 (large) and 1:250 (small) if they apply a simple rule of thumb. If the escape plan contains doors whose width approximately matches the radius of a one-cent coin, it could be assumed to have a scale of 1:100. If the doors are considerably smaller, the scale should be 1:250. The diameter of a one-cent coin is around 16 mm. In a scale of 1:100, this corresponds to 1.6 m. Usually doors should not be smaller than 0.8 m, especially when they are crossed by escape routes.

To compute an approximate scale value, the length of the shortest image side (in pixels) is divided by the length of the shortest side of the selected plan format (in meters). Subsequently, the result is multiplied by either 100 or 250, according to the user's choice. An exception is made in the case of DIN A4. Since for this format, a scale of 1:350 is allowed as well and the probability of a building to be too large for a representation in a scale of 1:100 is quite high, it is assumed that the scale must be either 1:250 (large) or 1:350 (small) in this case.

Especially since the old DIN standard does not explicitly specify which scale has to be used if 1:100 is not applicable, the approximation could be rather imprecise. However, with the new standard, in future it will be possible to rely on the assumption to have a scale of either 1:100, 1:250 or 1:350. On the other hand, it is not of great importance to have a very accurate approximation of this value.

In addition to format and scale of the plan, there is one more mandatory parameter. The type of door representation needs to be chosen. It could be either gaps or door symbols. By providing this setting, it can be avoided to search for both types. The probability for wrong results, as well as the computation time are reduced.

Adjusting other parameters is not mandatory, but was observed to be frequently necessary for good results. Most of them are concerning the colors. It can be that the colors of a certain plan do not match with the default parameters. Additionally, lighting, the camera used or the material of a plan can have an influence. Thus, in some cases it is necessary to adjust the hue, saturation or brightness thresholds, to obtain correct color segmentation results. Other adjustments can be necessary while detecting the floor plan.

4.7 Platform and Libraries

The prototype was developed for the Android[6] platform. The OpenCV[7] library, which provides a broad collection of programming functions for real time computer vision, was included to the source code. Amongst others, algorithms for contour finding, contour analysis, color space conversion, thresholding, morphological transformation and matrix operations were used from this library.

6 http://www.android.com

7 http://opencv.org

5 Evaluation

The evaluation was designed to address the following issues. The main goal was to test if the approach is robust enough to cope with the majority of representation forms used in escape plans. To do so, 30 images of escape plans were digitized using the prototype and the resulting data was analyzed. However, it was conceivable that the device used, lighting conditions or the material of the plan could influence the result. For this reason, the prototype was tested with a small set of escape plans under varying conditions beforehand. Although this work focuses on the technical feasibility, a small-scale user study was performed as well to get an impression of the willingness and ability of users to work with the proposed application.

5.1 Empirical Performance Evaluation

The first test was to digitize five escape plans under varying conditions. This included tests with three different devices, the three different lighting conditions daylight, artificial light and flash light, and plans with and without lamination. All 18 possible combinations of those properties were tested for each plan, resulting into 90 test images (available on attached DVD).

Keeping in mind the potential use case of the approach within a VGI project, three widespread mobile devices were chosen for the test. This included a Nexus S[8] with a 5-megapixel camera, a Galaxy Note[9] with an 8-megapixel-camera, and an iPhone 5[10], with an 8-megapixel camera as well.

Since it is required by law to install escape plans at locations where they are clearly visible, it was assumed that it would always be possible to take the photo from an optimal position and distance. For this reason, all test images were taken from a distance allowing to photograph the whole plan, but at the same time capturing as less as possible of its surrounding.

Especially because the approach heavily relies on color recognition, it was supposed to be influenceable by illumination. In principle, escape plans must be mounted on sites, where the lighting is good enough to make them well readable. For this reason, they

8 http://en.wikipedia.org/wiki/Nexus_S

9 http://de.wikipedia.org/wiki/Samsung_Galaxy_Note

10 http://en.wikipedia.org/wiki/IPhone_5

should not be found in dark environments. Though, depending on time of day and location, there could be daylight or artificial light. For the test, pictures were taken under both of those two typical conditions. Additionally, a series of pictures was taken using flash light. It was believed that there could be both, a positive influence from the increased brightness, but also a negative one, due to shifting the image's color.

To enhance the durability of escape plans, they usually are laminated or coated by another transparent material. To find out if there is an influence by the lamination, test images of laminated and non-laminated escape plans were made as well.

Lots of the materials used for coating the escape plans are strongly reflective. For this reason, in some cases it was necessary to take the picture from an angular perspective. To not pollute the test images by reflection artifacts, all flash light images were taken from an angle of 30 degrees.

The resulting data sets were evaluated using the protocol of Phillips and Chhabra [37], which was already used to report the accuracy of the two systems for architectural floor plan analysis, [30] and [49]. For each data set, detection rate, recognition accuracy, one-to-many count (over-segmentation) and many-to-one count (under-segmentation) of rooms were measured. Furthermore, the detection rate of doors, the number of wrongly detected doors and the accuracy of the building outline was determined.

The ground truth for each plan was defined manually. To obtain the detection rate of one plan, the number of rooms that match with only one detected room was counted and divided by the sum of all rooms. The recognition accuracy was measured by estimating the percentage to which the shape of a detected room matches with the ground truth. The recognition accuracy for one plan is the average of each room's accuracy measures. One-to-many count is the sum of all cases of over-segmentation. It occurs when one room matches with two or more detected rooms. Similar to this, many-to-one count is the sum of all cases of under-segmentation, meaning that two or more rooms match with one detected room. The recognition rate of the building outline was obtained in the same way as the one for the rooms. It is the percentage to which it matches with the outline defined by the ground truth.

Subsequently, another experiment was performed to find out if the approach can cope with the majority of representation forms, and at the same time, to see if it works under real conditions. This was to digitize 30 original escape plans found in public buildings (images available on attached DVD). The set consisted of a broad variety of plans. This included four small plans with less than 20 rooms, but also four quite large ones with more than 90 rooms. On average, the plans had 43 rooms. Furthermore, the collection covered a broad variety of representation forms. Since all images were taken in-situ, some of them had to be taken under difficult conditions. In particular, 8 images had to be taken from an angle greater than 30 degrees, due to reflecting materials used for coating the plans. The resulting data sets were evaluated using the same methods as in the first experiment.

5.2 Usability

To find out if users are willing and able to use the application, a small-scale user study was conducted. For the study, ten users had to digitize three original escape plans using the prototype. Afterward, they had to answer a questionnaire. The tests were done on the field, in three different buildings of the University of Münster. The test device was a Nexus S smartphone. All three escape plans were illuminated by artificial light and coated by a reflecting material.

While digitizing the first plan, each step was explained to the participants. They were told to capture the whole plan, but as little of its surrounding as possible with their photograph. They were advised to pay special attention to avoiding reflections. Using the application CamScanner, they had to indicate the plan's borders to crop the image appropriately and to remove perspective distortions.

Subsequently, the three parameters format, scale and door type had to be set. The participants were not explicitly told about right settings. Only the concepts and how to decide for the correct values was explained to enable them to make the decisions for their own. It was noted that the plans usually were in standard DIN formats, such as A4 or A3, and that the one fitting best in their opinion had to be chosen. Before setting the scale parameter, they were given a one-cent coin and explained that most plans either use a scale of 1:100 or 1:250. Then, they were told how it is possible to recognize the scale by comparing the radius of the one-cent coin to the width of the doors in the plan. Furthermore, it was said that there exist three forms to represent doors, triangles, arcs and gaps, and that they either have to select symbol or gap. After digitizing the first plan, the participants were told what the so obtained data could be used for.

The participants had to process the second and third plan without guidance. The time spend to collect the data was measured and it was captured which parameters were selected. Later, all resulting datasets were evaluated following the same protocol as the empirical performance evaluation.

The test ended with filling a questionnaire (see Appendix A). It was designed to investigate the interest of people to use the proposed system, to identify if all steps and parameters are well-performable, respectively understandable, and to get an idea if the computation time is tolerable.

6 Results

This chapter is structured as follows. First, the results of analyzing a collection of original escape plans are presented. This is followed by the results of evaluating the performance under varying conditions, the overall performance and usability of the prototype.

6.1 Representation Forms of Escape Plans

While analyzing the characteristics of 120 escape plans some rather fixed, but also some quite strongly varying properties were found. The greatest variety was concerning the layout of the included floor plans. As there exist lots of different standards for architectural floor plans, the same applies to floor plans in escape plans.

Since this work aims at extracting room shapes, the representation of walls was analyzed. It was found that in 51 % of the plans, inner walls were drawn as single lines (see Table 1), in 39 % of the plans as double lines and in 10 % as a mixture of single lines and double lines. In 66 % of all plans, the lines for inner walls were thin, but there were also 20 % with thick lines and 15 % where some walls were drawn as thick and some as thin lines.

	Line Type				Line Strength		
	Single (%)	Double (%)	Other (%)	Mixed (%)	Thin (%)	Thick (%)	Both (%)
Inner Walls	51	39	10	10	66	20	15
Outer Walls	27	20	54	17	42	27	32

Table 1: Representation forms of walls.

Especially for outer walls, a great variety of representation forms was found. The most common form were single lines (27 %), followed by double lines (20 %). But in 54 % of the analyzed plans, other forms were used as well. In many cases, more complex line structures were used to indicate windows or specific characteristics of walls. Furthermore, the lines were thin (41 %), thick (27 %) or both in one plan (32 %).

The consequence of this variety of line structures is that a distinction between inner walls, outer walls or other plan elements based on the nature of lines, as proposed in other works (e.g., [49]), does not seem to be applicable.

To represent doors, three basic forms were found (see Table 2). They were either depicted by arcs (66 %, see Figure 10), triangles (29 %) or gaps within walls (5 %). Furthermore, in some plans that used door symbols, such as arcs or triangles, walls were interrupted at the location of a door (referred to as *open*), and in others not (referred to as *closed*).

Type					Line Strength		
Arc Open (%)	Triangle Open (%)	Arc Closed (%)	Triangle Closed (%)	Gap (%)	Thin (%)	Normal (%)	None (%)
56	24	10	5	5	71	24	5

Table 2: Representation forms of doors.

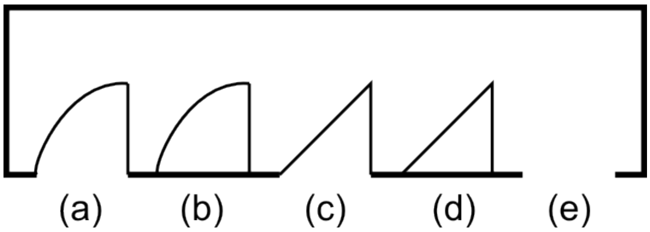

Figure 10: Representation of doors. (a) Arc open, (b) arc closed, (c) triangle open, (d) triangle closed and (e) gap.

Aiming at finding a criterion to distinguish door symbols from walls, for each plan, the line strength of doors and walls was compared. The result was that only in 71 % of the plans, the lines of doors symbols were thinner than those of walls, which means that this characteristic cannot be used for the distinction.

Although not mandatory, most plans (88 %) had labels indicating the room name, room number or its usage. What can be exploited to separate text from other plan elements, is that labels usually did not overlap with other elements.

Generally, escape plans should not contain unnecessary information. However, as stated before, it is not clearly defined, which information has to be considered as being unnecessary. While depicting furniture, such as tables in offices or seating areas, might help people to orientate, it could also be argued that it clutters the plan. Anyhow, it increases the complexity of an approach for automatic analysis. 39 % of the plans contained such information.

Since escape plans may use three different green tones, it was analyzed which tones are used and how they differ from each other. Green emergency signs and vertical escape routes often had almost the same color. On average, the green for signs, that theoretically should be signal green, had a saturation of 111 and a brightness of 56, while the green for vertical escape routes, which should be darker in theory, had a saturation of 108 and a brightness of 59. Furthermore, it was checked for each plan, if the color for vertical escape routes was actually the darkest green. However, this was only true for 61 % of the plans. For this reason, a separation from green signs and vertical escape routes just by color does not seem to be feasible.

In contrast to this, the color of horizontal escape ways is well distinguishable. In all plans, they were depicted in the green tone with the highest brightness. Additionally, in 93 % of the plans this green had the lowest saturation.

6.2 Prototype

In this section, the results of the prototype's evaluation are presented. This includes performance under varying conditions, overall performance and usability.

6.2.1 Performance under varying Conditions

The results of the first test show only little differences between the performances under varying conditions (see Table 3). On average, the detection rate obtained for the five test plans was 82.7 %. The best detection rate was 84.6 %, while in the worst case still 79.8 % of all rooms were matched with exactly one of the detected rooms. When it comes to recognition accuracy, the results are very close too. In the worst case the shape of each detected room matched on average to 88.5 % with the ground truth, and in the best case to 91.5 %. The average for all plans was 90.5 %. The same applies to the door detection rate, which was ranging from 76.8 % to 81.7 % and the outline accuracy ranging from 88.7 % to 95 %.

	Rooms				Doors			Outline
	Detection Rate (%)	1 - *	* - 1	Recognition Accuracy (%)	Detection Rate (%)	True Positive	False Positive	Accuracy (%)
All	82.7	1.2	1.2	90.5	79.1	17.8	3.7	92.8
Nexus S	79.8	1.3	1.3	90.3	76.8	17.0	3.8	88.7
iPhone 5	84.6	1.1	1.2	90.9	81.7	18.4	3.9	94.7
Galaxy Note	83.8	1.2	1.2	90.4	78.7	18.0	3.3	95.0
Without Lamination	82.5	1.2	1.2	90.2	79.3	17.9	3.8	91.2
With Lamination	82.7	1.2	1.3	90.9	78.5	17.8	3.7	93.8
Artificial Light	81.7	1.4	1.3	89.2	77.5	17.5	3.9	92.5
Flash Light	83.1	1.0	1.2	91.3	80.9	18.2	3.3	92.8
Daylight	83.4	1.3	1.2	91.1	78.8	17.7	3.8	93.0
Artificial without Lam.	81.7	1.4	1.3	88.5	77.4	17.6	3.9	90.3
Flash without Lam.	83.0	1.1	1.1	91.2	80.9	18.3	3.7	92.0
Daylight without Lam.	83.5	1.1	1.3	90.8	79.2	18.0	3.8	92.7
Artificial with Lam.	81.8	1.3	1.3	89.9	77.5	17.3	3.9	94.7
Flash with Lam.	83.2	0.9	1.3	91.5	81.0	18.1	2.9	93.7
Daylight with Lam.	83.2	1.5	1.1	91.4	78.4	17.5	3.9	93.3

Table 3: Performance of the prototype under varying conditions.

The greatest influence factor seems to be the device. Comparing the results of the iPhone 5 and the Galaxy Note to those of the Nexus S, considerable differences can be observed. Considering the detection rate and the outline accuracy, the performance of the Nexus S seems to be worse. This might be explained by the quality of the images taken with this device. It was observed that dark colors, such as dark green, tend to be darker on the photo than they really are. Thus, sometimes they could not be distinguished from black. The consequence was that vertical escape ways, which were dark green in the plan, looked black on the photo and were classified as walls. So those rooms were not detected or over-segmented, when an originally connected dark green region was interrupted by some pixels that had become too dark.

The same may occur for signs and may explain the variances in the building outline detection. The test included two plans with green or blue signs outside of the building outline. In a couple of images, they were very dark, so in several cases they had the same color as some walls.

Another rather critical plan element are door signs. In some plans they are depicted by very thin symbols that appear in a light gray on photos, although they are black in reality. Especially when the photo is a little blurry, it can occur that such door symbols are not even visible or interrupted. As observable in Table 3, the door detection rate seems to be influenced by the device as well. Similar to the problems observed for dark colors, some devices allow for a better graduation of bright colors than others. The consequence is that in some cases door symbols are distinguished from the background and sometimes not. Furthermore, this seems to depend on the lighting as well. While for photos taken with flash light, the door detection rate was 80.9 %, it was 78.8 for daylight and only 77.5 % for artificial light.

Comparing the other results of the three tested lighting conditions, it could be argued that the results obtained from the artificial light images are slightly worse. One reason for this might be that in comparison to the other two illumination settings, this one had the weakest light. Another interesting observation is that the results of the the flash light images are quite good, although they had to be taken from a distorted perspective to avoid reflection artifacts.

While designing this test, it was supposed that the result could be influenced by the coating of plans, especially when flash light is used. However, this does not seem to be true. The results for the tests with and without lamination do not show noticeable differences. Also, the comparison of images taken with lamination and flash light to images without lamination but with flashlight or other particular combinations does not show noteworthy influences.

6.2.2 Overall Performance

Table 4 shows the performances obtained while testing the prototype with a set of 30 real escape plans. On average, the detection rate was 78.5 %. In the worst case, it was 55 % and in the best case, 94.3 % were reached. The recognition accuracy was 81.9 % with a standard deviation of 10 %. In the worst case, the shapes of the detected rooms matched 54.9 % of the ground truth, 98.5 % in the best case .

	Number	Detected	Detection Rate (%)	1 - *	* - 1	1 - 0	0 - 1	Recognition Accuracy (%)
Average	43.1	48.8	78.5	3.1	2.3	1.4	5.1	81.9
Minimum	17.0	18.0	55.0	0	0	0	0	54.9
Maximum	124.0	140.0	94.3	14.0	8.0	7.0	22.0	98.5
Std. Dev.	30.2	31.5	9.7	3.3	1.9	2.0	5.6	10.0

Table 4: Room detection results.

The table reveals the system's tendency to produce more regions than actual rooms exist. The main reason is that sometimes other elements than walls were falsely classified as such. As mentioned before, many plans use very dark colors for several elements, which are hardly distinguishable from black. This can also explain the 3.1 occurrences of over-segmentation (1 - *) on average for each plan. In plans that represented doors as gaps, one more reason are mistaken doors where wall ends were pointing to another close wall. Actually, this is the method's criterion for detecting doors. Though apparently, relying to this criterion is not always sufficient. However, without extra knowledge, also for human observers it is hard to make the right choice for such wall constellations. While defining the ground truth, some situations where it was not possible to decide with absolute certainty occurred as well.

At the same time there were 2.3 cases of under-segmentation (* - 1) for each plan in mean. Most of their occurrences can be explained by one of the following reasons. Firstly, this may be caused by interrupted door symbols, which occurred in some images when very thin lines were used for the door symbols. Another reason are areas of escape ways. To cope with outliers inside of colored regions, dilation is performed and holes inside those regions are filled as well. This has the advantage that it counter-steers unnecessary splitting of such regions. But on the other hand, it may destroy thin lines that represent walls or doors. This may lead to under-segmentation. However, it could be argued that rooms, which are traversed by escape ways, are mostly hall ways. As such, a chain of particular hall way parts may be seen as one functional room. Especially since in many public buildings, they are only subdivided by doors for the matter of fire protection.

As well as for over-segmentation, a further reason for under-segmentation may occur for plans where doors are depicted as gaps. There can also be wall constellations where the system can not detect doors. For instance, when none of the walls ends at the location of a door. One more possible reason for under-segmentation are signs that occlude walls. The system extends walls with an ending that touches a sign area if there is another wall in its direction. But it can also be that the wall to which it should be connected, cannot be found in this direction. This can be the case when a sign is covering the walls of a room corner.

One to zero count (1 – 0) is the number of not detected rooms. On average, for each plan 1.4 rooms were not detected. Most of them occurred at dark green areas which were falsely classified as black. Some others were very small rooms, smaller than the threshold for the minimal room size.

Zero to one count (0 – 1) refers to detected rooms that do not have an equivalent in the ground truth. On the one hand, some of the on average 5.1 rooms for each plan occurred for buildings that have inner courtyards. Others were generated in plans where walls are represented by double lines. In some cases, the spaces between the lines were large enough to be considered as rooms by the system. Especially this error could be avoided in future by adding more strict criteria for room shapes. Further rooms were falsely detected outside of building outlines in sign areas, due to errors in the color classification.

Table 5 shows the performance of the door and outline detection. On average, the door detection rate was 64.5 %. The number of missed doors can be explained by three reasons already given for under-segmentation. Firstly, this is that the system mostly cannot detect doors within areas marked as escape ways. Secondly, this is the problem of very thin lines used for door symbols. Incomplete door symbols can not be detected. Thirdly, in plans with gaps, instead of door symbols, wall constellations may occur where the systems fails. In addition, the method does not work for door symbols that partly overlap with another wall. This can be problematic for doors situated in the corner of a room. For instance, if the door symbol consists of an arc and a line, while the line part overlaps with a wall, the system fails. Moreover, as well as walls, also door symbols may be occluded by signs. Although the system may fill gaps within door symbols as well, the method may fail especially when a part of the arc or the top of the symbol is covered.

	Doors				Outline
	Number	Detection Rate (%)	True Positive	False Positive	Accuracy (%)
Average	45.4	64.5	28.2	4.0	94.0
Minimum	5.0	29.0	3.0	0	70.0
Maximum	135.0	87.1	81.0	11.0	100.0
Std. Dev.	32.1	14.6	18.6	2.9	8.3

Table 5: Results of door and outline detection.

While on average, each tested plan contained 45.4 doors, also four faulty door detections occurred. As the system uses three methods for detecting doors, all of them may produce false results under certain circumstances.

The first method, the detection of appropriate gaps, may produce wrong results by falsely classifying certain gaps as doors.

The second method, which finds door symbols based on their contour, may classify certain wall constellations falsely as doors as well. In particular this happened in two plans quite frequently. The reason was that they contained a great number of pillars, which fulfilled the basic criterion used for detecting door symbols. Those pillars had a shape very similar to those of door symbols, while at the same time being hollow inside.

Also, the third door detection method, which assumes direct transitions from escape way areas to room interiors (i.e. no wall is in between) with an appropriate size as doors, produced false results in some particular cases. This occurred in plans with very thin walls. In some situations wall pixels were falsely classified as belonging to escape way areas. Consequently, wall segments that in fact were between the escape way area and the room interior, disappeared. In cases where the common border of these two areas had an appropriate size and form for being a door, false positive door detections occurred.

Detecting the building outline worked with an accuracy of 94 % on average (see Table 5). As already observed in the first experiment, problems occurred when signs where situated outside of the building outline. In plans with rather dark colors, sometimes pixels, that in fact were belonging to signs, were classified as being part of walls. Subsequently, while refining the detected building outline, the system considered those pixels as members of the building outline.

6.2.3 User Study

Table 6 shows that there were great differences between the datasets extracted by the ten participants of the user study. On average, the detection rate for the two plans digitized without guidance was 77 %. However, the best result (90.7 %) and the worst result (56.7 %) were far away from the average, as well as the standard deviation was quite high. The behavior of the recognition accuracy and the door detection rate was similar to this.

	Rooms		Doors	Outline
	Detection Rate (%)	Recognition Accuracy (%)	Detection Rate (%)	Accuracy (%)
Average	77.0	83.4	59.1	92.8
Minimum	56.7	71.4	43.3	87.5
Maximum	90.7	87.8	69.1	95.0
Std. Dev.	10.6	4.6	8.2	2.4

Table 6: Performance of the prototype in the user study.

On the one hand, the high variance can be explained by the characteristics of the two plans. Both had rather dark colors and quite thin door symbols. As observed before, these characteristics may cause problems to the system. Especially, due to the thin door symbols, the results were quite sensitive to camera shaking or irregular lighting. Moreover, the material of the plans was reflecting, while for both plans the lighting came from an angle that did not allow to take the photo from a straight perspective. Due to this, to obtain good results, it was of greatest importance to be very accurate while taking the photos. On the other hand, the high variance shows how differently the task of taking photos of escape plans can be understood and executed by different users. It was observed that for some participants it was no problem to find a perspective that does not produce reflections. While for others this was quite challenging. Although while explaining the task the importance of avoiding reflections was pointed out, 20 % of the images contained strong reflection artifacts (as in Figure 11b). Additionally, further 15 % were blurred.

Figure 11: Quality differences of images resulting from the user study. (a) Without reflections, (b) with strong reflection artifacts on the top left.

According to the questionnaire (see Appendix B), 50 % of the participants found it easy or very easy to photograph the escape plans without reflections, while 20 % found it difficult. A small difference can be seen between participants who took their photos without reflections and participants whose photos contained reflections. Participants, who took photos free of reflections, on average found it easier to avoid reflections than the others. For this reason, it appears likely that reflections occurred because the participants had problems to find the right perspective, and not because they were not aware of the importance to avoid them.

It was also investigated how well the participants understood choosing the right settings. As can be seen in Table 7, choosing the right format was no problem. All participants were able to decide for the right format for both of the plans. Answering the questionnaire, 90 % of the participants stated that they found it easy or very easy to asses the format. Only one participant found it difficult.

	Right Format (%)	Right Scale (%)	Right Door Type (%)	Time (min)
Average	100	85	100	2:49
Minimum	100	50	100	1:39
Maximum	100	100	100	4:44
Std. Dev.	0	22.9	0	0:52

Table 7: Behavior of the participants in the user study.

Judging about the scale of the plan was observed to be more critical. Beforehand, it was assumed that estimating the scale is a task that could only hardly be performed without any extra knowledge. For this reason the participants were informed how they can use a one-cent coin to better estimate the scale. Despite of this, three participants decided for the wrong scale at the second plan. However, having the extra knowledge that was taught them before, 70 % of the participants found it easy to estimate the scale. Only one participant evaluated it as difficult.

Indicating the representation form of doors to the system did not cause any problems. For both plans, all participants did the right choices. Moreover, all of them found it easy or very easy to select the right value for this parameter.

Besides aiming to investigate the ability to work with the system, another goal of the user study was to get an insight in the interest of people to use it. 90 % of the participants already knew about OpenStreetMap or other VGI projects. And even 70 % already collected data for a VGI project. However, only 30 % were interested in collecting indoor data in general, while 40 % were not, or absolutely not interested. Similar were the answers to the question, if they are interested to collect indoor data by photographing escape plans. 20 % were interested, 10 % were not interested and 30 % had absolutely no interest.

The last question was concerning the computation time. As can be seen in Table 7, on average the participants took 2:49 minutes to extract the data of one plan. This included around 43 seconds for the computation[11]. 50 % of the participants found that this

11 On a device with a 1 GHz single-core processor.

was quick enough or clearly quick enough, but also 30 % found that this is not quick enough or clearly not quick enough.

7 Discussion

The aim of this work is to find out if extracting indoor map data from escape plans is a feasible task to be performed automatically using mobile devices. It is assumed that this depends on the way plans are represented and their degree of standardization. The results obtained by analyzing 120 original escape plans show that lots of different notions are in use. On the one hand, certain characteristics are clearly fixed by standards. But on the other hand, the space remaining unaffected by standardization, is filled with a diverse mixture of notions. This is particularly true for the floor plan.

In principle, any existing standard for architectural drawings may be used. In fact, the escape plans of many buildings are created by reusing the building's already existing architectural plan. The only requirement is to remove unnecessary content. This is followed in some plans more, in others less. This might be linked to the fact that there is no clear definition of what is unnecessary content. However, with the exception of furniture (e.g., tables or seating), the floor plans usually seem to depict only very basic architectural entities, such as walls, doors, stairs, elevators or windows.

There already exist several works on the analysis of architectural floor plans. Many of them exploit characteristics that only occur for a number of particular standards. Some approaches assume that the line strength is connected with certain feature classes (e.g., [30]). For instance, they assume that thin lines are only used for symbols (e.g., door symbols). Alternatively, they assume that thick lines are always used to represent outer walls, and never for inner walls (e.g., [26]). The results of the analysis show that it cannot be relied on neither of these assumptions. Similarly, many works only address one representation form of doors (e.g., [30], [35] and [49]), while three different basic representation forms are found in the analyzed set.

In contrast to this, other properties are quite reliable. In particular, the colors to be used are rather well defined. Particularly, they help to distinguish the floor plan from other elements. Only the three green tones constitute a major obstacle. Due to the great variation of green tones used from plan to plan, assigning green features to their right meaning is quite challenging. However, this problem will disappear with the new standard, which includes just two green tones.

Another useful property constitute the scale restrictions. Especially the new standard helps to improve the conditions. The old standard only requires to use 1:100 if possible, and if not, any other smaller scale may be used. With the new standard it can be assumed to either have 1:100, 1:250 or 1:350 in some special cases. With the help of this knowledge, it is possible to estimate a transformation factor between pixels and the real world. This is needed to adjust certain parameters. This way, it is not necessary to have access to a georeferenced version of the building's exterior, as in [33].

All in all, the standardization of escape plans provides several useful properties that may be exploited by a system for automatic analysis by means of computer vision methods. While the characteristics of plans created according to the old standard already are quite sufficient, the situation could be improved by replacing the old plans by new ones created according to the DIN ISO 23601 standard. Especially since this is an international norm, which is intended to be implemented by many different countries.

The results of the evaluation show that the system can work under the typical conditions of the intended use case. It is conceivable that escape plans are mounted in bright rooms and close to windows. They can be situated in rooms without windows, so that there are only artificial light sources. They can also be in comparably dark rooms, so that flash light needs to be used. The test results of these three lighting conditions show only little differences. And what is of most import, there is no setting where the system is unusable. Indeed, it has to be noted that the test was limited to only one type of artificial light, which was fluorescent. In some buildings, other lamps can be found, which might produce negative effects, such as shifting the colors.

Also, the results for laminated and non-laminated plans do not show significant differences. This is even the case for flashlight images. The results are similar to those without lamination. Although it was necessary to photograph from a sharp angle, due to the material's characteristic of being reflective.

Only the device seems to be an important factor for the quality of the result. The system performs better for images with a great number of color graduations. Cameras that tend to depict pretty dark colors as black or pretty bright colors as white increase the probability of errors in plans with certain critical characteristics. Basically, these are very thin door symbols or dark colors for signs or vertical escape ways. However, at least the problem with vertical escape ways will disappear with the new standard, since they need to be indicated in bright green then as well. Moreover, only some plans have these characteristics and usually this only concerns smaller parts of the plans.

Table 8 compares the room detection performance with two systems for analysis of architectural floor plans described in [30] and [31]. Both were evaluated using the protocol of [37] as well. Of course, it is not possible to directly compare the present results to the others. The system described here deals with escape plans, while those are designed to work with architectural floor plans. Furthermore, the set of test plans in the referenced works was different and obviously had different characteristics. Instead of 43.1, on average, the plans of that collection only contained 9.3 rooms. Nevertheless, in the absence of other comparable works, and especially in the absence of other comparable evaluation results, this comparison still may help to better classify the present results.

	Number	Detected	Detection Rate (%)	1 - *	* - 1	Recognition Accuracy (%)
[30]	9.3	15.3	85	2	0.77	69
[31]	9.3	-	89	1.5	1.6	79
Proposed	43.1	48.8	78.5	3.1	2.3	81.9

Table 8: Room detection performance compared to performances of two systems for architectural floor plan analysis described in [30] and [31].

Considering that the plans used to test the other two systems contained only one particular representation standard, the results for the system proposed here may be seen as being quite good. Both, the worse detection rate and the better recognition accuracy can be explained by general characteristic differences of original architectural floor plans and floor plans found in escape plans. On the one hand, the complexity for escape plans increases due to extra information, such as emergency symbols and escape routes. Since those elements often hide wall segments, this may cause an increase of segmentation problems and could have a negative influence to the detection rate. On the other hand, escape plans usually contain less architectural information, since it is required to remove unnecessary contents. Potentially in original architectural floor plans a greater number of symbols and lines that do not directly refer to walls can be found. Consequently, an accurate detection of room shapes is hindered, which may explain the higher recognition accuracy archived by the system proposed here.

To decide if the obtained performance is sufficient, the overall performance results can be examined from a more practical perspective. It could be argued that the less post-processing is necessary, the better is the result. Table 9 reveals the percentages of rooms that require a certain refinement to obtain an error-free data set on average. Based on the proportion of correctly recognized, over-segmented, under-segmented and falsely detected rooms, the table shows how many rooms need to be merged, split, deleted or do not need modification. Additionally, it shows how many room shapes need no, minor or major refinements. Room shapes with 100 % accuracy are assumed to not needing refinements. Shapes with an accuracy between 80 % and 100 % need minor refinements. Those with not more than 80 % accuracy are assumed to require major refinements.

Segmentation Refinements				Shape Refinements		
None (%)	Merge (%)	Split (%)	Delete (%)	None (%)	Minor (%)	Major (%)
69	16	5	10	34	40	26

Table 9: Necessary refinements to obtain a data set free of errors. Each percentage refers to the proportion of rooms that require the corresponding operation on average.

As can be seen, on average, quite a number of operations is necessary to remove all errors. 16 % are sub-regions of rooms and need to be merged, 5 % need to be split and further 10 % need to be removed. Moreover, the shapes of 40 % of the detected rooms need minor refinements and another 26 % needs major refinements. Beyond this, missing rooms and missing doors need to be added, and on average, 14 % of the detected doors need to be removed.

Clearly, with this error rate, a direct use of the just extracted data, for instance in an application for indoor routing, seems not to be feasible. Especially since the number of missing doors and over-segmentation result in a set of rooms not connected to others. Furthermore, a routing application only would make sense if the extracted data sets contain at least basic semantic information, such as room names. But to ease the creation of coarse indoor maps, the results should be sufficient. The largest part of the extracted rooms can remain unchanged or requires only minor refinements. Considering that not all use cases require highest accuracy, it is not of greatest importance to always correct all room shapes with just minor distortions. The remaining effort required to correct all major errors is comparably small. Compared to other approaches, and especially to those that only require widely used technical equipment (e.g., [14] or [15]), the total effort to obtain a readily useable set of indoor data is considerably lower.

Moreover, the results of the user study show that generally, people are able to work with the system. Although the design of the prototype did not focus on usability, all participants were able to collect indoor map data with the proposed system. However, for future work it needs to be considered that the order to take a photo of an escape plan without reflections can be understood and executed very differently. Some users appear to be satisfied with their photograph when it still contains clearly visible reflection artifacts. Others proceed very accurately. They try to vary the perspective until they obtain a good result. In addition, two users stated that they did not think it was possible to completely avoid reflection at one of the plans. The user interface needs to account for this. It has to explicitly advise users to pay special attention to this.

Furthermore, a better solution to estimate the scale needs to be found. The intermediate solution tested here, to let the user decide which scale is used, is error-prone. For this, it is desirable to find an automatic solution. Potentially, this could be done by detecting features that have a rather fixed size, for instance doors.

The user interface needs to be extended as well. One important thing that needs to be added is a mean to adjust the threshold values for the color separation. Conceivable would be a method that allows the user to point areas that have a particular sought-after color. Although the floor plan detection was observed to work quite reliably in the tests, adding a mean to correct errors occurring in this method would be reasonable. It is of great importance that this step is free of errors. If the actual floor plan can not be found, no data extraction is possible. A method allowing the user to indicate regions belonging to the floor plan, but also to exclude superfluous parts, would be desirable. Another reasonable extension poses editing functionality. Means to allow for in-situ correction of errors or to enrich data sets with additional information could further increase the system's benefit.

Some participants of the user study were not satisfied with the calculation time. While the algorithms chosen preferably require as less computation time as possible, in the actual prototype not all algorithms are implemented in the most efficient way. For this reason, quite a potential for increasing the processing speed remains. Some methods are not implemented natively yet. As well as multi-threading potentially could be applied for some methods.

An aspect that needs to be further investigated, is the interest of people to collect indoor data, and in particular, to collect indoor data by photographing escape plans. Due to its small size, the results of the user study can not give a sufficient answer to this. It only gives small evidence that at least some people may have a general interest. Aside from performing a larger survey, the most efficient way to obtain a sufficient answer may be to further develop the system and to make it available to as many potential users as possible.

8 Conclusion

This thesis aims at investigating the feasibility of using mobile devices to extract indoor map data from images of public escape plans. To achieve this, a collection of original escape plans is analyzed and a number of general characteristics that can be exploited by computer vision methods is identified. Based on this, a prototypical system for map data extraction is developed. The results of the evaluation show that it can work under typical conditions. The quality of the extracted data sets is good enough to on average only requiring a comparably small number of major errors to be corrected. All in all, the effort to obtain instantly usable data sets is considerably smaller than such of other existing approaches. A user study proves that people generally are able to make use of the application.

This work shows the feasibility of the approach in principle. However, to make it readily usable, and this in particular for voluntary non-expert contributors of a VGI project, further work is indispensable. The development of an appropriate user interface is of special importance. Furthermore, the approach could be refined to better address some more specific characteristics of certain plans. Problems that occur while mostly relying on colors to identify particular features could be counteracted by increasing the exploitation of structural characteristics.

Moreover, additional feature types could be extracted. For instance, stairs and elevators could be detected. This is of particular interest when it comes to realization of systems to assist people with walking disabilities.

Finally, since many plans also contain labels for each room, the approach could be extended by an appropriate optical character recognition method to extract depicted room names and numbers.

References

[1] J. Rosser, J. Morley, and M. Jackson, “Crowdsourcing of building interior models,” in *Proceedings of the 15th AGILE International Conference on Geographic Information Science*, 2012, pp. 24–27.

[2] M. Goetz and A. Zipf, “Extending OpenStreetMap to Indoor Environments: Bringing Volunteered Geographic Information to the Next Level,” *28th Urban Data Management Symposium*, pp. 47–58, 2011.

[3] ASR A2.3, *Technische Regeln für Arbeitsstätten Fluchtwege und Notausgänge, Flucht- und Rettungsplan Zielstellung Anwendungsbereich*. 2011.

[4] DIN 4844-3, “Safety marking - Part 3: Escape- and rescueplan,” 2003. [Online]. Available: http://www.beuth.de/de/norm/din-4844-3/64374740.

[5] DIN ISO 23601, “Safety identification - Escape and evacuation plan signs (ISO 23601:2009),” 2010. [Online]. Available: http://www.beuth.de/de/norm/din-iso-23601/134503696?SearchID=479423473.

[6] G. Wyszecki and W. S. Stiles, *Color Science: Concepts and Methods, Qunatitative Data and Formulae*. New York: Wiley, 1982.

[7] J. Serra, *Image Analysis and Mathematical Morphology.* London: Academic Press, 1982.

[8] A. Rosenfeld and J. Pfaltz, “Sequential operations in digital picture processing,” *Journal of the ACM (JACM)*, 1966.

[9] S. Suzuki and K. Abe, “Topological Structural Analysis of Digitized Binary Images by Border Following,” *Computer Vision, Graphics and Image Processing*, vol. 30, pp. 32–46, 1985.

[10] L. Di Stefano and A. Bulgarelli, “A simple and efficient connected components labeling algorithm,” *Proceedings 10th International Conference on Image Analysis and Processing*, pp. 322–327, 1999.

[11] H. Freeman, “Computer Processing of Line-Drawing Images,” *ACM Computing Surveys*, vol. 6, no. 1, pp. 57–97, Jan. 1974.

[12] J. Iivarinen and A. Visa, “Shape recognition of irregular objects,” *Photonics East’96*, 1996.

[13] J. Iivarinen, M. Peura, J. Srel, and A. Visa, “Comparison of Combined Shape Descriptors for Irregular Objects,” *8th British Machine Vision Conference, BMVC ’97*, 1997.

[14] A. Westermann, G. Tschorn, P. Weiss, O. Ogundele, and D. Lasnia, “OpenFloorMap Implementation,” in *Geoinformatik 2011*, 2011.

[15] J. Rosser, J. Morley, and M. Jackson, “Mobile Modelling for Crowdsourcing Building Interior Data,” *ISPRS - International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, vol. XXXVIII–4/, pp. 29–32, Jun. 2012.

[16] A. Budroni and J. Böhm, “Automatic 3D Modelling of Indoor Manhattan-World Scenes from Laser Data,” *Proc. of the ISPRS Commission V Symposium*, vol. XXXVIII, 2010.

[17] B. Okorn and X. Xiong, “Toward Automated Modeling of Floor Plans,” *Proceedings of the Symposium on 3D Data Processing, Visualization and Transmission*, 2010.

[18] T. Liu, M. Carlberg, G. Chen, J. Chen, J. Kua, and A. Zakhor, “Indoor Localization and Visualization Using a Human-Operated Backpack System,” in *International Conference on Indoor Positioning and Indoor Navigation*, 2010.

[19] S. Izadi, D. Kim, and O. Hilliges, “KinectFusion: Real-time 3D Reconstruction and Interaction Using a Moving Depth Camera,” *Proceedings of the 24th annual ACM symposium on User interface software and technology*, 2011.

[20] P. Henry, M. Krainin, E. Herbst, X. Ren, and D. Fox, “RGB-D mapping: Using Kinect-style depth cameras for dense 3D modeling of indoor environments,” *The International Journal of Robotics Research*, vol. 31, no. 5, pp. 647–663, Feb. 2012.

[21] K. Khoshelham and S. O. Elberink, “Accuracy and resolution of Kinect depth data for indoor mapping applications.,” *Sensors (Basel, Switzerland)*, vol. 12, no. 2, pp. 1437–54, Jan. 2012.

[22] Y. Furukawa, B. Curless, S. M. Seitz, and R. Szeliski, “Reconstructing Building Interiors from Images,” *2009 IEEE 12th International Conference on Computer Vision*, pp. 80–87, Sep. 2009.

[23] G. Klein and D. Murray, “Parallel Tracking and Mapping on a Camera Phone,” *2009 8th IEEE International Symposium on Mixed and Augmented Reality*, pp. 83–86, Oct. 2009.

[24] C. Ah-soon, K. Tombre, C. Cnrs, and I. Lorraine, “Variations on the Analysis of Architectural Drawings,” in *Proceedings of the 4th International Conference on Document Analysis and Recognition*, 1997, pp. 1–13.

[25] R. Lewis and C. Séquin, “Generation of 3D building models from 2D architectural plans,” *Computer- Aided Design*, vol. 30, no. 10, pp. 765–779, 1998.

[26] P. Dosch, K. Tombre, C. Ah-Soon, and G. Masini, “A complete system for the analysis of architectural drawings,” *International Journal on Document Analysis and Recognition*, vol. 3, pp. 102–116, 2000.

[27] S. Or, K. Wong, Y. Yu, and M. M. Chang, “Highly Automatic Approach to Architectural Floorplan Image Understanding & Model Generation,” *Pattern recognition*, 2005.

[28] T. Lu, H. Yang, R. Yang, and S. Cai, “Automatic analysis and integration of architectural drawings,” *International Journal of Document Analysis and Recognition (IJDAR)*, vol. 9, no. 1, pp. 31–47, Aug. 2006.

[29] X. Yin, P. Wonka, and A. Razdan, “Generating 3D Building Models from Architectural Drawings: A Survey.,” *IEEE computer graphics and applications*, vol. 29, no. 1, pp. 20–30, 2009.

[30] S. Macé, H. Locteau, E. Valveny, and S. Tabbone, “A System to Detect Rooms in Architectural Floor Plan Images,” in *Proceedings of the 9th IAPR International Workshop on Document Analysis Systems*, 2010, pp. 167–174.

[31] S. Ahmed, M. Liwicki, M. Weber, and A. Dengel, “Improved Automatic Analysis of Architectural Floor Plans,” *2011 International Conference on Document Analysis and Recognition*, pp. 864–869, Sep. 2011.

[32] M. Peter, N. Haala, M. Schenk, and T. Otto, “Indoor Navigation and Modelling Using Photographed Evacuation Plans and MEMS IMU,” *IAPRS*, vol. XXXVIII, 2010.

[33] M. Peter, N. Haala, and D. Fritsch, “Using Photographed Evacuation Plans to Support MEMS IMU Navigation,” in *2011 International Conference on Indoor Positioning and Indoor Navigation (IPIN)*, 2011.

[34] N. Haala, D. Fritsch, M. Peter, and A. K. De, “Pedestrian Mobile Mapping System for Indoor Environments based on MEMS IMU and Range Camera,” *Archives of Photogrammetry, Cartography and Remote Sensing*, vol. 22, pp. 159–172, 2011.

[35] M. Peter and D. Fritsch, “Versatile Geo-referenced Maps for Indoor Navigation of Pedestrians,” *International Conference on Indoor Positioning and Indoor Navigation*, vol. 13, 2012.

[36] L. Priese and P. Sturm, “Introduction to the color structure code and its implementation,” 2003.

[37] I. Philipps and A. Chhabra, “Empirical performance evaluation of graphics recognition systems,” *Pattern Analysis and Machine Inteligence*, vol. 21, pp. 849–870, 1999.

[38] C. Tomasi and R. Manduchi, “Bilateral Filtering for Gray and Color Images,” *Sixth International Conference on Computer Vision (IEEE Cat. No.98CH36271)*, pp. 839–846, 1998.

[39] L. Luccheseyz and S. Mitray, “Color image segmentation: A state-of-the-art survey,” in *Proceedings of the Indian National Science Academy (INSA-A)*, 2001.

[40] V. Rehrmann and L. Priese, “Fast and robust segmentation of natural color scenes,” *Computer Vision—ACCV’98*, 1997.

[41] D. Comaniciu and P. Meer, “Mean Shift Analysis and Applications,” *The Proceedings of the Seventh IEEE International Conference on Computer Vision*,

vol. 2, 1999.

[42] N. Otsu, “A Threshold Selection Method from Gray-Level Histograms,” *Automatica*, vol. 20, no. 1, pp. 62–66, 1975.

[43] A. Jain, *Fundamentals of Digital Image Processing*. Englewood Cliffs, NJ: Prentice-Hall, 1986.

[44] K. Tombre, S. Tabbone, B. Lamiroy, and P. Dosch, “Text / Graphics Separation Revisited,” in *Proc. 5th IAPR Int’l Workshop Document Analysis Systems*, 2002, pp. 200 – 211.

[45] D. Bloomberg and L. Vincent, “Document Image Applications,” in *Morphologie Mathmatique*, Hermès - Lavoisier, 2008.

[46] P. P. Roy, J. Lladós, and U. Pal, “Text / Graphics Separation in Color Maps,” in *International Conference on Computing: Theory and Applications, ICCTA*, 2007, pp. 545–551.

[47] L. A. Fletcher and R. Kasturi, “A Robust Algorithm for Text String Separation from Mixed Text/Graphics Images,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 10, no. 6, pp. 910–918, 1988.

[48] K. Tombre, C. Ah-Soon, and P. Dosch, “Stable and Robust Vectorization: How to Make the Right Choices,” *Graphics Recognition Recent Advances*, pp. 3–18, 2000.

[49] S. Ahmed, M. Liwicki, M. Weber, and A. Dengel, “Automatic Room Detection and Room Labeling from Architectural Floor Plans,” *10th IAPR International Workshop on Document Analysis Systems*, pp. 339–343, Mar. 2012.

[50] Deutsches Institut für Normung, “Neuer Norm-Entwurf zur Kennzeichnung von Fluchtwegen,” 2007. [Online]. Available: http://www.din.de/cmd?level=tpl-artikel&cmstextid=64708&languageid=de&bcrumblevel=2.

[51] S. Sieder, “Flucht- und Rettungspläne,” 2009. [Online]. Available: http://www.schemas-plaene.de/Fluchtweg.htm.

Appendix A: Questionnaire of the User Study

In the following, all questions of the user study's questionnaire are listed. The original questions were in German.

1. Please indicate your age.

2. Please indicate you gender.

   Male ☐ Female ☐

3. Do you own a smartphone?

   Yes ☐ No ☐

4. How practiced are you in using smartphones?

   Very practiced ☐ ☐ ☐ ☐ ☐ Very unpracticed

5. Do you know OpenStreetMap or other VGI projects?

   Yes ☐ No ☐

6. Have you ever collected data for OpenStreetMap or similar projects?

   Yes ☐ No ☐

7. Would you be interested in collecting indoor data for OpenStreetMap or other VGI projects?

   Very interested ☐ ☐ ☐ ☐ ☐ Absolutely not

8. Would you be interested in collecting indoor data by photographing escape plans?

Very interested ☐ ☐ ☐ ☐ ☐ Absolutely not

9. How easy is it for you to take pictures of escape plans without reflections?

Very easy ☐ ☐ ☐ ☐ ☐ Very difficult

10. How easy is it for you to appropriately crop the image?

Very easy ☐ ☐ ☐ ☐ ☐ Very difficult

11. How easy is it for you to decide about the format (A4, A3, etc.) of the escape plan?

Very easy ☐ ☐ ☐ ☐ ☐ Very difficult

12. How easy is it for you to decide about the scale (small or large) of the escape plan?

Very easy ☐ ☐ ☐ ☐ ☐ Very difficult

13. How easy is it for you to decide about the representation form of doors (symbol or gap)?

Very easy ☐ ☐ ☐ ☐ ☐ Very difficult

14. Would you evaluate the computation time as being short enough?

Clearly yes ☐ ☐ ☐ ☐ ☐ Clearly no

Appendix B: Survey Results of the User Study

The following table contains the participant's answers to the questionnaire. In question 4, 7, 8, 9, 10, 11, 12, 13 and 14, 1 corresponds to the most left possible selection in the questionnaire, while 5 corresponds to the most right possible selection in the questionnaire.

	Question													
Participant	1	2	3	4	5	6	7	8	9	10	11	12	13	14
1	27	m	yes	1	yes	yes	2	3	4	1	4	2	1	2
2		f	yes	2	yes	yes	3	3	2	1	2	2	2	2
3	25	f	yes	2	yes	yes	2	3	3	1	2	1	2	4
4	25	m	yes	1	yes	yes	4	4	3	1	1	1	1	5
5	25	m	yes	3	yes	yes	5	2	2	1	1	3	1	3
6	24	m	yes	1	yes	yes	3	3	1	1	2	4	2	4
7	28		no	5	yes	no	2	5	3	2	1	2	1	2
8		f	yes	3	yes	no	5	5	4	1	2	2	1	3
9	25	m	no	3	no	no	5	5	2	1	1	3	1	1
10	27	m	yes	1	yes	yes	3	2	2	1	1	2	1	1

Appendix C: Source Code and Test Images

Attached to this work is a DVD containing the following items:

1. The source code of the prototype.
2. An APK file allowing to install the prototypical application on Android devices.
3. The images of the analyzed escape plans.
4. The images used for the test under varying conditions.
5. The images used for testing the overall performance.

Statutory Declaration

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

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

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

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

Münster, March 25, 2013

Georg Tschorn