Map Model Extraction from Image Floor Plans

Miroslav Opiela[1], Martina Hrehová[1] and František Galčík[1]

[1] *Faculty of Science, Institute of Computer Science, Pavol Jozef Šafárik University in Košice, 04001 Košice, Slovakia.*

Abstract

Indoor positioning systems commonly rely on map models to enhance localization accuracy. These models can be represented in either vector or raster formats. In this study, we propose a method that utilizes floor plan images, commonly provided in IPIN competitions, to process the map and create both vector and raster models. Manual annotation is used to identify walls, doors, and zones, while automatic methods based on convex polygons are employed to generate the map model. Additionally, we introduce a computer vision technique for automatic map annotation. This method significantly reduces the map processing time, reducing it from 40 minutes required for manual annotation to just 5 minutes with the automatic approach, followed by manual editing. Although the solution is not entirely conclusive, the map model can be reliably obtained with minor user adjustments.

Keywords

map, map model, floor plan, indoor positioning, computer vision, line detection

1. Introduction

Indoor positioning is diverse in terms of use-cases, devices, and solutions. Various positioning methods have appeared in recent years [1]. The IPIN competition is a reflection of these trends, offering different tracks that address specific aspects of indoor positioning. Notably, tracks focused on smartphone-based solutions provide images of building floor plans to be used for the positioning.

Lessons learned from competitions [2] and observation of solutions introduced by competitors clearly suggest the importance of fusion of multiple sources of information. The presence of the map model bounds the overall position estimation within the area of accessible locations. Moreover, the structure of the building with walls, corridors, and junctions may improve the positioning accuracy for various solution types. The map model is essential for many systems, especially for those using pedestrian dead reckoning approach calculating user's position relative to previous estimations.

The main objective of this paper is to propose a method for generating map models using floor plan images. It is possible to derive a data-driven approach using neural networks to perform this task. However, a labeled dataset of the required quantity may be challenging to obtain. Instead of building a robust solution for map model extraction from images, the automatic method with user adjustments or semi-automatic approach may be considered. In

Proceedings of the Work-in-Progress Papers at the 13th International Conference on Indoor Positioning and Indoor Navigation (IPIN-WiP 2023), September 25 - 28, 2023, Nuremberg, Germany

miroslav.opiela@upjs.sk; martina.hrehova@student.upjs.sk; frantisek.galcik@upjs.sk

0000-0001-8802-4442 (Miroslav Opiela)

this work, the automatic method for vector and raster model production is presented based on annotated map. The annotation of walls forming convex polygons is manual or automatic with manual user corrections. The solution combines established computational geometry methods with a computer vision approach, aiming for a fully automatic method that currently requires some user adjustments.

The paper provides a brief summary of related work regarding integration of map into solutions based on Bayesian filtering, and approaches using neural network or computer vision for floor plan image to vector conversion. The proposed system is introduced in Section 3 with definitions of inputs and outputs for selected methods. Section 4 summarizes the map annotation, including the computer vision method for automatic annotation. The annotated model with convex zones is transformed to the vector model and then to the two-dimensional grid in Section 5. Section 6 is focused on the computer vision method evaluation and the overall examination of the proposed method.

2. Related Work

In many solutions, Bayesian filtering handles inaccuracies introduced by noisy sensors. The process consists of two phases: the prediction of the system state and the correction [3]. In general, the uncertainty of the state estimation is increased in the first phase and the correction phase reduces the variability using obtained measurements. Typical implementations of filtering include Kalman and particle filters. The map is utilized in such systems, e.g., Fetzer et al. [4] limit the movement in the first phase. The transition is performed according to the mesh obtained from the map model. In [5], the map model is utilized in the correction phase to suppress unaccessible positions (walls, locations outside the building).

In the first aforementioned example, floor plans are processed manually in a custom 3D editor, and the navigation mesh is automatically created. The second example utilizes the two-dimensional grid, which was obtained using the method proposed in this paper. In general, simple floor plans are replaced by extended map representations including 3D map models. Chen and Clarke [6] present various examples of map formats with comments on supported geometry, semantic support, and spacial referencing. Li et al. [7] provide a survey on map formats and standards in context of indoor positioning.

The map model extraction involves the utilization of three primary types of approaches: manual, analytical, and data-driven. Manual annotation of maps is often performed in geographic information systems (GIS). Multiple sofware applications are available and provide an opportunity to visualize geographic data, e.g., ground truth and estimated positions from IPIN competitions are given in KML format, which could be loaded and displayed over the map background. These data, including points, lines, and polygons, may be created, edited and further processed. Nevertheless, it is not unusual to use a custom-built application for the map annotation.

Jaworski et al. [8] introduce an analytical approach that requires the user to setup the venue, followed by automatic walls and doors detection. The proposed algorithm consists of image preprocessing, circle Hough transform, and least squares method. The solution includes a tool for manual polygon drawing to label door masks. The evaluation also includes a building with not

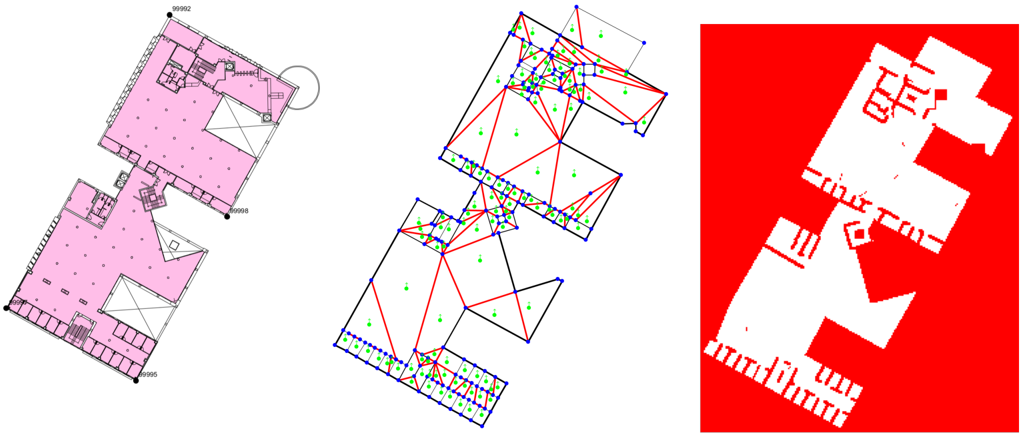

Figure 1: Left image is a floor plan of a library building in Spain provided by IPIN 2020 organizers. Middle image represents the vector model and is composed by manual annotation of connections forming convex polygons. Right image is a raster representation of accessible areas generated by the proposed method.

only straight and perpendicular walls. Pan et al. [9] propose a solution which extracts bearing walls and performs other steps for elements recognition. Tombre and Tabbone [10] suggest that contour-matching methods achieve better results than skeletonization-based approaches.

Even though data-driven methods are not applied in this paper, the trend in computer vision problems to be solved by machine learning methods is present also in floor plans processing. Dodge et al. [11] propose an approach for parsing floor plan images based on fully convolutional neural networks. Kim et al. [12] perform the raster to vector conversion using deep network and style transform.

Given the problem addressed in this study, the input consists of a raster floor plan image, while the output varies depending on the specific use case, often also taking the form of a raster image. Nevertheless, the majority of solutions typically undergo a conversion process to transform the input into vector representation before generating the output raster image, if required. Liu et al. [13] adopt learning-based approach where neural network transforms a rasterized image to a set of junctions. The proccess continues with the junctions aggregation into simple primitives using integer programming to produce a vector floor plan. It allows 3D model popup for indoor scene visualization. In [14], the deep segmentation and neural networks for detection are utilized to outputs a corresponding vectorized 3D reconstruction model.

In [15], room detection is a final step in processing the image floor plan. Basic building blocks (walls, doors) are detected using statistical patch-based segmentation followed by structural pattern recognition methods on the generated graph. In such graphs, the walls are vertices and edges denote the connection between two walls.

3. System Overview

The proposed solution consists of multiple steps to obtain a two-dimensional grid from a floor plan image through creating a vector map (Figure 1). This process is semi-automatic.

Input is considered to be a single image of the floor plan with additional information involving the map scale (ratio between a map pixel and a meter in real world), map rotation, and a georeferenced position (WGS-84) of the initial point in the image. The leftmost top corner is mostly convenient to be considered as the initial point and could be calculated if other position is given.

Final output is a two-dimensional regular square-shaped grid, where every grid cell denotes whether the respective area is accessible or inaccessible in the real building. Inaccessible positions are physically impossible to access (walls) or chosen to be off the limits (places outside the building). The map is tessellated into the grid automatically and the grid cell width is a parameter.

Moreover, this process produces an intermediate output, which is a vector map model consisting of points, lines and polygons. From structural point of view, the map model is formed by points and connections (lines) with assigned properties. These data are stored to fully represent the map and are used in following operations in order to obtain the full vector or the raster output.

Map of a single floor is composed of so-called zones. A zone is a basic unit in a form of a convex polygon. The convexity requirement of polygons is essential for the following automatic extraction of a vector model. In case of semantical requirements, rooms could be created as a combination of multiple zones, e.g., each zone is assigned to a room. During the process, zones are annotated with a single point anywhere inside the polygon which is later acquired automatically. Three types of connections are present in this solution - solid lines (walls), transitional lines (doors), and transparent lines (artificial borders for separating zones). These lines are segments connecting two distinct points in the map.

4. Map annotation

The input image is processed to obtain the vector and/or raster map. Walls and doors are annotated on the floor plan manually or automatically, using proposed computer vision approach.

4.1. Walls and Doors

In the image, walls and doors are annotated. Both are processed identically with a distinction in the property assigned to these annotated connections. This task may be performed manually using GIS software or a custom application. A list of connections is the final product of the annotation process. To simplify the task, it is recommended to have the original floor plan image as a background layer and to draw lines at the top of it. The output is formed by a list of points with their 2D positions in the image (in pixels) and a list of connections where every connection (of two endpoints) is a pair of two indices in the points list.

Even though the annotation is reliable, it is the most tedious part of the solution, especially in more advanced floor plans with hundreds of connections. The process may be replaced by a

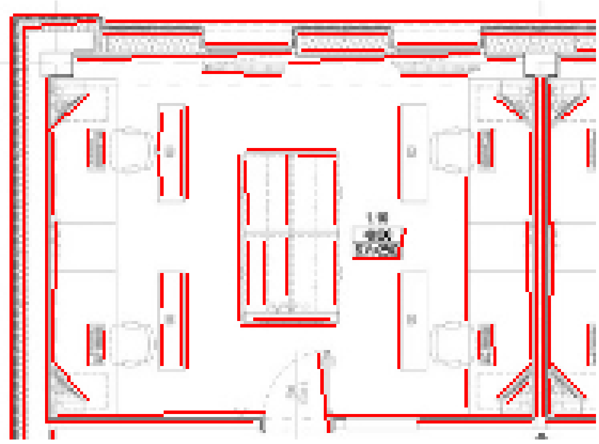

Figure 2: Red lines are denoted by line segment detection. There are walls with multiple detected lines which is later reduced using mean shift clustering of endpoints. The image is a floor plan of a faculty building in Slovakia.

computer vision approach. The output is corrected manually afterwards, i.e., the connections are verified manually and removed, added, or edited if necessary. The method is executed as follows:

1. The image is preprocessed with a focus on removing text to improve the accuracy of walls detection. The text is removed from the image using OCR (Optical Character Recognition) method, e.g. [16]. Moreover, rotating or mirroring the image may help to achieve better output.
2. Line segment algorithm (LSD) [17] is applied to obtain a list of lines in the image. Walls are often thick in the image, i.e., they are not represented by a line but a solid narrow rectangle. Figure 2 shows that the automatic detection may produce duplicated lines for these walls.
3. Endpoints of detected lines are clustered using mean shift algorithm to minimize redundant walls. The benefit of the mean shift is that it does not require the number of clusters to be defined and is based solely on the distance between samples. After the point clusterization, a new list of lines is derived from the LSD output in a way that every endpoint is replaced by its cluster position. Duplicates of lines are removed.
4. Points are aligned using mean shift algorithm applied separately on each dimension. The lines are restored similarly as for the two-dimensional clustering. This method suppresses the misalignment introduced by the LSD algorithm and preceding clustering. Mostly in buildings, multiple line segments (walls, doors) lies on the same line which is rarely the case if detection is inaccurate. Note that the input includes the map rotation information which may be utilized if walls are not mostly in west-east and north-south directions in the image.

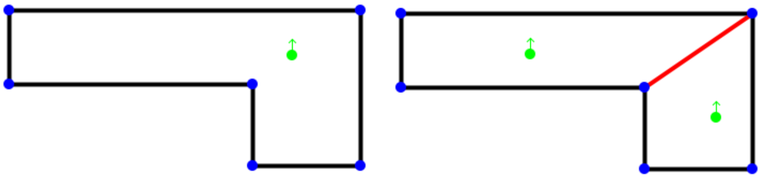

Figure 3: First image denotes a zone which is not convex. The convexity could be ensured by dividing the polygon with a single line (red color) to two separate convex zones.

4.2. Convex Zones

A polygon is convex if any line segment between two arbitrary points inside polygon is inside this polygon as well. Moreover, internal angles do not exceed $180°$. A zone is a convex polygon with connections as its sides. The requirement of convex zones enables the method to automatically create the vector map model. A zone is annotated in the floor plan with a single point which should be placed anywhere inside the polygon (ideally not too close to the border).

Alongside the vector model extraction, a few automatic operations are performed to ensure that the annotation is correct, i.e., internal angles are not reflex angles, there are no missing connections, and every polygon contains at least one connection which is transparent or transitional.

If a zone is not convex and cannot be easily transformed to the convex polygon by adjusting positions of points, the zone should be divided into at least two zones using transparent connections (Figure 3). This method does not influence the semantical representation of the map but ensures the property of convex polygons.

Zones are disjoint and every position may belong to at most one zone.

5. Vector Model and Grid

The annotated map is processed automatically to obtain a vector model or the raster map as a grid of accessible or inaccessible positions.

5.1. From Convex Zones to Vector Model

Annotated model consists of connections and points either as connection endpoints or zone denoting points. The output vector model is a list of zones. Every zone is a sequence of n points $(C_0, C_1, \ldots, C_{n-1}), n > 2$. There is a connection between all succeeding points (connection $C_iC_{i+1}, i \in (0, \ldots, n-2)$ and between the last and the first point $C_{n-1}C_0$).

A zone is constructed from a list of all connections in the annotated map and the zone point $Z[x_z, y_z]$ in a following way:

1. First connection C_0C_1 is selected from a list of all connections such that there is an intersection between line C_0C_1 and the vertical line ZZ_0, where $Z_0[x_z, 0]$, Z is zone point, and the distance between the intersection and Z is minimal.

2. Next point C_{i+1} for given C_i is selected from the list of connections, where C_i is one endpoint of the connection, excluding the connection $C_{i-1}C_i$. Connection C_iC_X with minimal angle $C_{i-1}C_iC_X$ is selected as the connection C_iC_{i+1}.
3. The process is repeated until the connection $C_{n-1}C_0$ is found, i.e., $C_{i+1} = C_0$.

To eliminate the impact of the connection direction, the vector product $C_{i-1}C_i \times C_iC_x$ is calculated and according to the direction, the conjugate angle is considered if the corresponding angle is greater than $180°$. Therefore, the first point of a zone sequence C_0 may be selected from either endpoint of the first connection.

5.2. From Vector Model to Grid

The list of zones in form of sequences of points is transformed to the two-dimensional grid. The grid cell size is defined in advance. The output may be visualized as an image of size $(W/S, H/S)$ where W, H are width and height of the original floor plan image in pixels and S is the single grid cell side. However, output in more convenient way for further processing may be required, e.g., the list of zeros and ones denoting accessible and inaccessible positions alongside with the overall size of the grid.

The grid is derived from the vector model as follows:

1. Set grid scale based on the given parameter and create a two-dimensional array of numbers with a predefined value 1 for inaccessible grid cells.
2. Iterate over all connections forming zones. Calculate array indices (grid cells positions) for respective connection endpoints. Apply line drawing algorithm between these grid cells. The value 2 is assigned to all cells alongside the line segment which is a wall and the value 3 is used for doors and transparent connections.
3. Iterate over all zones. Starting from the zone point, apply flood-fill algorithm to fill zone area bounded by values 2 or 3. The algorithm puts value 0 to grid cells to denote accessible areas. Flood-fill is implemented using breadth-first search with four neighbours for every grid cell.
4. Finally, mark all grid cells representing transitional or transparent connections (with value 3) as accessible (value 0).
5. If the separate distinction of walls is not necessary, replace walls (value 2) with inaccessible information (value 1). Moreover, true and false values may be used instead 0 and 1.

Line drawing is performed using naive algorithm where the y position is calculated for all given x ($x_1 \leq x \leq x_2$):

$$y = y_1 + d_y \times (x - x_1)/d_x$$

where (x_1, y_1) and (x_2, y_2) are endpoints of the line segment and difference values are calculated $d_x = x_2 - x_1$ and $d_y = y_2 - y_1$.

To ensure that no gaps arise that would cause the flood fill algorithm to fill places outside the zone, not only the position (x, y) is denoted as line point but also $(x + 1, y)$. The line is thicker and more robust for further processing. If $d_y > d_x$, it is recommended to exchange x and y, i.e., y is iterated from y_1 to y_2 and the respective x is calculated.

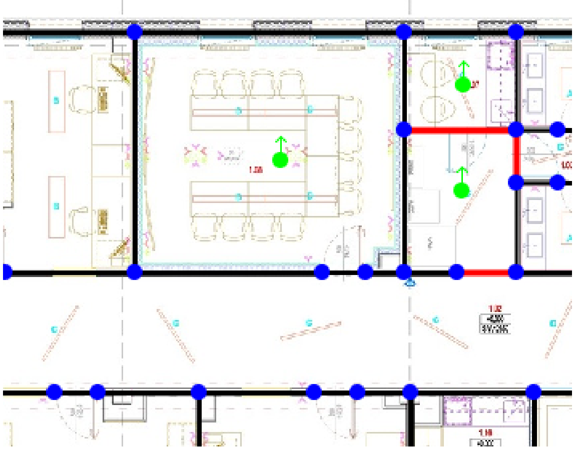

Figure 4: Walls and zones annotation with an original floor plan as the background. Black lines represent walls. Red lines denote doors or transparent connections added to achieve the convexness of polygons. Green dots inside polygons identify zones.

6. Evaluation

Four different multifloor buildings were processed using proposed approach with manual annotation of walls. Three buildings were from IPIN competition: Atlantis shopping mall in Nantes, France (IPIN 2018), CNR research institute in Pisa, Italy (IPIN 2019), and library in Castellón, Spain (IPIN 2020). Map models for these buildings were created with no prior knowledge of specific buildings. Author of this paper visited first two venues afterwards (at the onsite competition). Some map elements were challenging to process offsite, e.g., balconies and outdoor staircases on IPIN 2019 maps. Moreover, a faculty building in Slovakia was prepared for indoor positioning evaluation.

6.1. Application used for Floor Plans Annotation

The annotation of walls and zones was performed in a custom Java application. Main principles are the same as for GIS software. An automatic calculation of zones and check of polygon convexness were incorporated into the application. Moreover, the two-dimensional grid is exported from the vector representation. Although common geographic formats exist, the application stores the vector map model in a custom text-based format. Distances and positions are presented in centimeters, simplifying the process and aiding users in verifying the correctness of the map model.

Annotation of walls (including doors) is the most tedious part of the solution. The application provides features to streamline the construction of convex polygons, such as aligning points on a line or creating a point on a line. The most challenging aspect of manual annotation is ensuring the convexness of multiple small rooms that share the same endpoint. The zones annotation is a quick task when walls are present. Figure 4 shows an example of annotated map over a floor plan image.

The process of manual annotation in the application can be summarized as follows. The

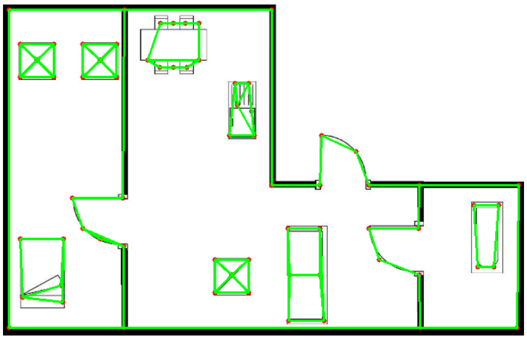

Figure 5: Incorrect wall annotation. Three walls are detected at doors and furniture is labeled as well. Image is from ROBIN dataset.

original image is loaded with the appropriate resolution as the background. All walls and doors are annotated by creating points and connections between them. L-shaped corridors are split using transparent connections. Zone points label all corridors and rooms. Automatic counting and validation of zones are performed. Non-convex zones are rectified by adjusting points or adding transparent connections to divide the zone into two zones if needed. The process is repeated until all desired zones are accurate. Finally, the grid is exported and visualized in the output image.

6.2. Line Detection Evaluation

The evaluation of computer vision method proposed in this paper was accomplished on 20 selected floor plans from ROBIN (Repository of Building plaNs) [18], CubiCasa5k [19], and CVC-FP [20] datasets. The main aim was to observe recurring tendencies and problems instead of a statisical evaluation. These 20 images consist of various number of walls (from 14 to 50), doors (between 3 and 15), and rooms (from 3 to 13).

In general, text separation using Keras OCR [16] improved the final output. However, the method was unsuccessful on two images and incomplete on two images out of 5 from CubiCasa5k dataset and incomplete in 5 of 9 images from CVC-FP. Images from ROBIN dataset do not contain any text. More problems occurred in large, complex buildings with numerous junction points, where multiple connections share the same endpoint or with zone polygons consisting of a large number of connections. Nevertheless, the importance of text separation depends on specific floor plan and was not examined further in these experiments.

Different methods were compared for the line detection. The automatic process becomes more complex when different maps require various parameter configurations using Hough transform. Therefore, line segment detection was applied. The resolution of images was downgraded to 650x650 pixels to obtain better results. Thick walls were often labeled by two lines which is resolved using mean shift algorithm. The bandwidth parameter for this method was set to 19. The alignment of points was performed by mean shift algorithm with bandwidth 10 individually for x and y axes.

The worst results were obtained on images from CubiCasa5k dataset. Only one image met the declared requirements. Door detection was successful on 50% images. Wall detection achieved unsatisfactory results especially in images with furniture which draws lines in incorrect places (Figure 5).

ROBIN dataset results provides the most representative outputs due to the simplicity of floor plans. Door detection was problematic on the CVC-FP dataset, as doors are visualized with thin lines that are difficult to detect. These observations helped to improve the proposed method to achieve the best possible result for this approach.

Apart from text separation and resolution changes, the input images were not preprocessed. In the IPIN competitions, the focus was on corridors rather than rooms, so furniture was not a primary concern in such scenarios. Therefore, no method for detecting or separating objects in floor plans was included.

6.3. Overall Evaluation

The automatic evaluation did not find all required lines with 100% accuracy. However, the proposed method expects the complete labeled model with walls, doors, and convex polygons. The output may be manually repaired by editing, adding, and removing lines in the aforementioned application. The detailed experiment was performed on IPIN 2019 map consisting of more than 800 lines. Manual annotation required 40 minutes for experienced user. Automatic method produced 892 distinct points and 870 lines. Manual adjustments took 5 minutes to achieve the desired model accuracy. The majority of the time was spent identifying lines that needed editing. In order to streamline this process, a tool that can identify problematic lines, such as those not aligned in preferred directions, highlight points with only one connection, and assess zone completeness, would be highly beneficial.

Exact comparisons in terms of points and lines count can be challenging to perform as the annotation process may vary in execution. While the key structure is provided, specific lines may be split into multiple smaller connections. Achieving zone convexity can be accomplished by adjusting points on the same larger line or by dividing polygons into smaller units, and this process can be carried out in different ways. As a result, the improvement is substantial in terms of the time spent by a trained person.

The zone reconstruction and grid creation performed as expected for all tested maps. However, an issue arose with low-resolution floor plan images, resulting in inaccessible grid cells appearing unexpectedly. This problem mainly occurred near short connections and can be easily resolved through adjustments to the annotated model, applying automatic post-processing techniques, or altering the resolution of the input image.

The proposed method's key advantage is its high level of automation. With enhanced line annotation, the solution progresses towards becoming fully automatic. However, its weakness lies in its lack of robustness, as the process relies on the specific building image style, necessitating customization for each image type. Even the manual annotation process demands minor adjustments in the application to tackle new challenges presented by different buildings. Moreover, the requirement for zone convexity introduces an additional layer of intricacy to the map annotation process. While in other types of solutions, this step may not be necessary, here it becomes essential to ensure that points on the same line are aligned or transparent lines are introduced to fulfill the convexity property.

Conclusion

This paper addresses the challenge of extracting map information from raster images. The proposed method focuses on automatically extracting a vector model from annotated maps and converting it into a two-dimensional grid. The annotation process is typically done manually using a custom Java application, which can be time-consuming.

To improve efficiency, the paper introduces the automatic annotation method based on computer vision techniques such as line detection and mean shift clustering. While the automatic method may not provide a complete output, it significantly reduces the time required for manual adjustments compared to traditional annotation. In the specific scenario using the IPIN 2019 competition map, the method only required 5 minutes of correction instead of the usual 40 minutes of manual annotation by experienced user.

Although the proposed computer vision approach automates a significant portion of the annotation process, certain parts still require manual execution. Achieving a fully automatic method remains a significant challenge. However, alternative semi-automatic methods could further reduce the overall time needed for the map extraction. For example, users could click on specific elements like doors, and the system could automatically label them using pattern recognition or template matching. Contour finding in the image is another potential approach.

In the future, it would be beneficial to conduct a broader evaluation to test the robustness of the proposed method on larger and more complex maps, as well as to identify any limitations. Nevertheless, the proposed method simplifies the manual process of annotating maps, contributing to the enhancement of indoor positioning systems and improving the accuracy of user or device localization.

Acknowledgments

This paper was supported in part by the Slovak Grant Agency, Ministry of Education and Academy of Science, Slovakia, under Grant 1/0177/21, and in part by the The Cultural and Education Grant Agency, under Grant 012UPJŠ-4/2021.

References

[1] G. M. Mendoza-Silva, J. Torres-Sospedra, J. Huerta, A meta-review of indoor positioning systems, Sensors 19 (2019) 4507.

[2] F. Potortì, J. Torres-Sospedra, D. Quezada-Gaibor, A. R. Jiménez, F. Seco, A. Pérez-Navarro, M. Ortiz, N. Zhu, V. Renaudin, R. Ichikari, et al., Off-line evaluation of indoor positioning systems in different scenarios: The experiences from ipin 2020 competition, IEEE Sensors Journal 22 (2021) 5011–5054.

[3] V. Fox, J. Hightower, L. Liao, D. Schulz, G. Borriello, Bayesian filtering for location estimation, IEEE pervasive computing 2 (2003) 24–33.

[4] T. Fetzer, F. Ebner, M. Bullmann, F. Deinzer, M. Grzegorzek, Smartphone-based indoor localization within a 13th century historic building, Sensors 18 (2018) 4095.

[5] M. Opiela, F. Galčík, Grid-based bayesian filtering methods for pedestrian dead reckoning indoor positioning using smartphones, Sensors 20 (2020) 5343.

[6] J. Chen, K. C. Clarke, Modeling standards and file formats for indoor mapping., GISTAM (2017) 268–275.

[7] K.-J. Li, S. Zlatanova, J. Torres-Sospedra, A. Pérez-Navarro, C. Laoudias, A. Moreira, Survey on indoor map standards and formats, in: 2019 International Conference on Indoor Positioning and Indoor Navigation (IPIN), IEEE, 2019, pp. 1–8.

[8] W. Jaworski, P. Wilk, M. Juszczak, M. Wysoczańska, A. Y. Lee, Towards automatic configuration of floorplans for indoor positioning system, in: 2019 International Conference on Indoor Positioning and Indoor Navigation (IPIN), IEEE, 2019, pp. 1–7.

[9] G. Pan, J. He, R. Fang, Automatic floor plan detection and recognition, in: 2017 2nd International Conference on Image, Vision and Computing (ICIVC), IEEE, 2017, pp. 201–205.

[10] K. Tombre, S. Tabbone, Vectorization in graphics recognition: to thin or not to thin, in: Proceedings 15th International Conference on Pattern Recognition. ICPR-2000, volume 2, IEEE, 2000, pp. 91–96.

[11] S. Dodge, J. Xu, B. Stenger, Parsing floor plan images, in: 2017 Fifteenth IAPR international conference on machine vision applications (MVA), IEEE, 2017, pp. 358–361.

[12] S. Kim, S. Park, H. Kim, K. Yu, Deep floor plan analysis for complicated drawings based on style transfer, Journal of Computing in Civil Engineering 35 (2021) 04020066.

[13] C. Liu, J. Wu, P. Kohli, Y. Furukawa, Raster-to-vector: Revisiting floorplan transformation, in: Proceedings of the IEEE International Conference on Computer Vision, 2017, pp. 2195–2203.

[14] X. Lv, S. Zhao, X. Yu, B. Zhao, Residential floor plan recognition and reconstruction, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2021, pp. 16717–16726.

[15] L.-P. De Las Heras, S. Ahmed, M. Liwicki, E. Valveny, G. Sánchez, Statistical segmentation and structural recognition for floor plan interpretation: Notation invariant structural element recognition, International Journal on Document Analysis and Recognition (IJDAR) 17 (2014) 221–237.

[16] F. Chollet, et al., Keras, 2015. URL: https://github.com/fchollet/keras.

[17] R. G. Von Gioi, J. Jakubowicz, J.-M. Morel, G. Randall, Lsd: A line segment detector, Image Processing On Line 2 (2012) 35–55.

[18] D. Sharma, N. Gupta, C. Chattopadhyay, S. Mehta, Daniel: A deep architecture for automatic analysis and retrieval of building floor plans, in: 2017 14th IAPR International Conference on Document Analysis and Recognition (ICDAR), volume 1, IEEE, 2017, pp. 420–425.

[19] A. Kalervo, J. Ylioinas, M. Häikiö, A. Karhu, J. Kannala, Cubicasa5k: A dataset and an improved multi-task model for floorplan image analysis, in: Image Analysis: 21st Scandinavian Conference, SCIA 2019, Norrköping, Sweden, June 11–13, 2019, Proceedings 21, Springer, 2019, pp. 28–40.

[20] L.-P. de las Heras, O. R. Terrades, S. Robles, G. Sánchez, Cvc-fp and sgt: a new database for structural floor plan analysis and its groundtruthing tool, International Journal on Document Analysis and Recognition (IJDAR) 18 (2015) 15–30.