An Augmented Reality Facet Mapping Technique for Ray Tracing Applications

Varun Kumar Siddaraju
Ingram School of Engineering
Texas State University
San Marcos, USA
varunsiddaraju@gmail.com

George Koutitas
Ingram School of Engineering
Texas State University
San Marcos, USA
george.koutitas@txstate.edu

Abstract—**This paper presents a novel spatial mapping technique that is capable of extracting the vector map of an indoor environment based on images captured from a smart phone camera. The extracted vector map follows the facet model concept and can be used as input in ray tracing algorithms for indoor wireless channel predictions. The algorithm computes the coordinates of the walls and doors of each room of the indoor environment and creates the facet model of the entire 3D space by applying edge and corner detection on the wall images of each room. The output of the algorithm is a facet model that can be used by ray tracing algorithms which are embedded in Augmented Reality (AR) applications. The overall process provides a better human-to-network interface and an improved user experience that is expected to provide a new way for indoor network planning of residential 5G systems.**

Keywords-augmented reality; spatial mapping; facet model; ray tracing; indoor networks; channel prediction.

I. INTRODUCTION

The computation of indoor vector maps and spatial mapping are active research fields for various Augmented Reality (AR) applications. Characteristic examples are gaming, interior design, property advertising, indoor security, indoor navigation, that all require information of the indoor space in order to overlay holograms. Spatial mapping requires high-end cameras like RGB depth (RGB-D) and Simultaneous Localization and Monocular (SLAM) cameras and this increases the overall cost of the system [1]. Spatial mapping is the process of analyzing the 3D space and transforming it to a set of vertices coupled to other information such as vertices normal and vertices type. In most applications, this transformation is very useful since a user can place holograms and avatars in the real space and interact with them. In some occasions, other applications may require a simplified spatial mapping where the overall objective is just to create the vector map of the walls and doors of the indoor environment without the need for indoor clutter information. This paper proposes a novel technique that can provide 3D mapping of indoor spaces utilizing the facet concept and only requires the use of a commodity smart phone cameras.

Different types of AR algorithms and limitations for real-time imaging are discussed in [2]. The presented applications mainly concern the use case of military, medical, gaming, interior designing and advertising. A survey of AR technologies and applications is also presented in [3][4]. With the increase of various AR applications, the need for more sophisticated spatial mapping and 3D indoor mapping algorithms also increases. Spatial mapping is usually performed by RGB-D cameras [5][6] and simultaneous localization and monocular (SLAM) cameras [7][8]. The RGB-D camera captures 3D RGB images with their depth details. SLAM cameras, simultaneously map the indoor environment with localization of indoor environment features and clutter. Both RGB-D cameras and SLAM cameras are integrated within expensive AR devices.

The next generation of communication networks, namely, the 5G networks, are expected to create new opportunities for mobile AR applications [9]. One characteristic application is network visualization and human-to-network interaction. For example, with the use of an AR application a user can visualize the results of ray tracing simulations that are overlaid on top of the physical space. This is very important for 5G networks where short range communications are expected to create important indoor network planning challenges [9]. To perform field strength prediction, ray tracing algorithms use the facet model where the indoor environment is represented in a vector format with facets incorporating data of the coordinates and the material structure of each facet [10]. An example of the use of indoor vector maps for ray tracing algorithms is given in [11].

The proposed algorithm uses a simple camera of a smart phone device that captures images of individual walls and is capable of constructing a simplified 3D map of the indoor space. The 3D map is a facet model that can be used by indoor channel estimation algorithms. The AR application then overlays the ray tracing results to enable a better human to network interaction. The facet model is created by identifying the coordinates and sizes of the walls and doors of the indoor environment. This process incorporates image processing techniques responsible for the edge and corner detection. The proposed solution uses the *Canny* edge detector to extract wall and door boundaries [12]. Corners on the found edges are detected using the concept of *detect minimum eigenvectors* algorithm. An interesting analysis and comparison of corner detection techniques is given in [13]-[15]. Based on the detected corners of the walls and doors of individual rooms, the entire indoor environment is synthesized to create a full 3D vector representation. A Graphical User Interface (GUI) is also developed to enable an easier interaction between the user and

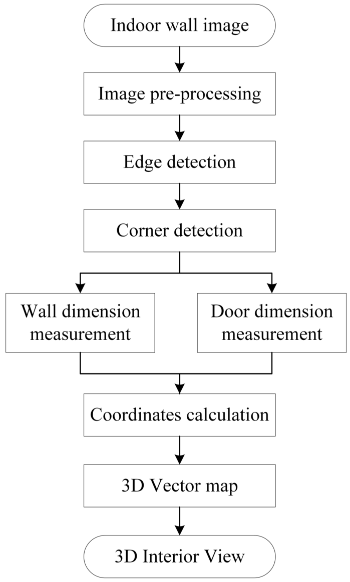


Figure 1. Flowchart of 3D indoor facet mapping.

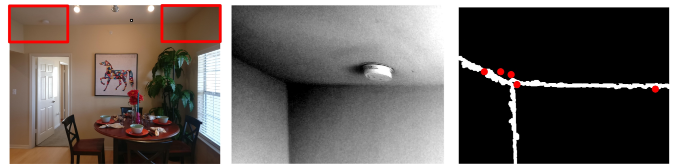
the application. The overall objective of the proposed solution is to create the necessary foundations for the efficient network planning and positioning of femtocell stations with the use of a typical smart phone device and AR applications.

The rest of the paper is structured as follows. In Section II, we present the system description with overview of the new algorithm. In Section III, we present the facet model with 3D model construction and its data structure. In Section IV, we present the results with detailed Graphical User Interface (GUI), computed facet model and ray tracing visualization output. Finally, we conclude the paper in Section V with the algorithm applications and future work.

II. System Description

A. Overview

The algorithm processes images of the indoor environment, identifies the walls and doors positions, computes the coordinates and creates the facet model for each wall and door. For the purpose of this investigation, window detection was omitted. This process is performed for each room of the indoor space and the found facets are combined in a data structure to represent the entire indoor environment. This process can be considered as a simplified spatial mapping technique that neglects the detailed furniture clutter since it is not significantly affecting signal propagation. The input of the algorithm are the images of every wall but also the height of the ceiling. The images can be captured using a standard camera of a typical smart phone device, without the need of using an expensive depth camera. The input images are then pre-processed to enable an efficient edge and corner detection process which is important for the identification of the vertices and coordinates of the walls and doors. The 3D Cartesian coordinates of a room are calculated using the length, width and height of the room which is computed once the wall and door vertices are detected. Using these coordinates, the 3D vector map or else the facet model can be constructed and become available to third party applications such as ray tracing and AR.



a) b) c)

Figure 2. a) The two regions of interest on the original wall image, b) Pre-processed image of top left region, c) detected edges and candidate corners.

The detailed overview of the proposed solution is presented in Figure 1 and is analyzed in the following sections. For efficient performance of the algorithm, the following assumptions should stand:

- Capture photo of the wall from the center of the room by standing parallel to the wall
- The captured image must be clear without any clutter near the top corners of the walls
- If the wall is large, the user can use the panorama function of the smartphone device to capture the entire wall in a single image file

In practice, the aforementioned conditions are usually met in most typical residential units. It should be noted that the proposed technique cannot be used for large corporate offices, since a wall is usually large enough and cannot fit in one photo screen.

B. Image pre-processing

The image pre-processing is the first step of the overall technique and prepares the images of the room for the edge and corner detection phase. For the efficient edge and corner detection, the input image is converted into a grayscale image [5]. The second step of the pre-processing phase is to crop selected regions of interest from the gray scale image.

For the purpose of our investigation these are the top and left corners of the wall as shown in Figure 2. The regions of interest are used to minimize unwanted edge and corner detection and reduce the computational demands of the algorithm. The last part of the pre-processing phase corresponds to a down sampling of the image pixel size procedure on the cropped images that further reduces the computational demands of the process. Usually, the image can be convolved with a Gaussian filter to reduce the number of unwanted edges [9]. The smoothing process [9] is given in the following formula:

$$S[i,j] = G[i,j;\sigma] * I[i,j] \quad (1)$$

where $I[i, j]$ denotes the input image of pixel size ixj, $G[I,j;\sigma]$ denotes Gaussian smoothing filter and $S[i, j]$ denotes the array of smoothed data and σ is the gradient level of the filter. Image is down-sampled to different resolutions like 1280x768, 960x720, 640x480 and experimented for best corner detection results. Images with low resolution 640x480 help to reduce the number of false corner detection compared to higher resolution images. A 5x5 size Gaussian filter is used for efficient edge and corner detection [13]. The overall process of the image pre-processing is demonstrated in Figure 2 a) and Figure 2 b). The next phase of the proposed solution is to process those images

 ISBN: 978-1-61208-623-1

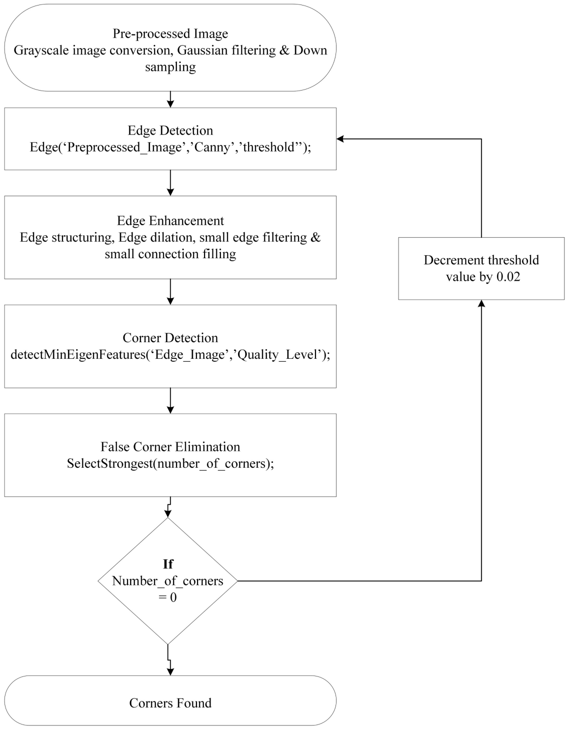


Figure 3. Flowchart of edge detection and corner detection.

for the edge and corner detection, as shown in Fig 2. c).

C. *Edge and corner detection*

The edge and corner detection of the wall image is the most crucial part of the algorithm. This is because, corner detection is directly related to the coordinates of the wall of the room, and thus the development of the facet model. The edge and corner detection flowchart is given in Figure 3. The first part of the algorithm is to perform edge detection upon the preprocessed input wall images by implementing the edge *Canny* method [9]. The *Canny* method calculates the gradient using the derivative of a *Gaussian* filter and uses two thresholds to identify strong and weak edges. With this approach, the edge detection of unwanted noisy parts of the image is minimized. The *Canny* method uses a threshold to distinguish between strong and weak edges. For the purpose of our investigation, the edge is detected according to the following function.

$$EM = edge\ (S, Canny, \delta, \sigma) \qquad (2)$$

where *S*, denotes the pre-processed image, 'Canny' denotes the edge detection algorithm, δ is the threshold used and is a two element vector, σ is the standard deviation of the *Gaussian* filter and is a scalar and *EM* denotes the edge map of the wall image. The *EM* image is a binary matrix with *1s* representing the points where an edge is detected. The threshold value is a sensitivity value, and is used to ignore all edges that are not stronger than the selected threshold. It is a scalar value that specifies the standard deviation of the Gaussian filter. The initial threshold was set to $\delta=0.4$ and if no corners found, it

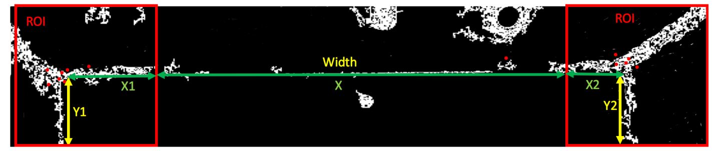


Figure 4. Corner points matching between two regions of interest of a wall image.

decrements by 0.02. The standard deviation was set to $\sigma=sqrt(2)$.

The corner detection is the second step of the process during which the edge map of the image is processed for the identification of the candidate corners. The output of the corner detection algorithm is a set of potential points that can be considered corners of the walls, as shown in Fig 2. c). The red mark corresponds to the set of potential points. It is obvious that the corner point that falls on the intersection of the three edges is the preferred wall corner. The identification of the final corner is described in the next section of the appear. For the purpose of our investigation, the *detectMinEigenFeatures* corner detection algorithm [14] was used. This is a function of MATLAB and has the following structure:

$$Corner = detectMinEigenFeatures\ (EM, q, G); \qquad (3)$$

where *EM* denotes the edge map in gray scale (binary), *q* is a scalar value between [0, 1] and denotes the corner strength and quality. Larger values of *q* are used to eliminate erroneous corner points. For the purpose of our investigation, the value was set $q=0.5$ because the pre-processing phase of the image eliminates the majority of erroneous points. The function returns an object file called *Corner* that incorporates location of corners in pixel coordinates *i*, *j* and the corner metric value, C_{metric}. Larger corner metric indicates a strongest candidate for a corner [13]. Parameter *G* is the Gaussian filter dimension and is an odd integer value in the range [3, inf]. For the purpose of our investigation, we set $G=3$. The Gaussian filter is used to smooth the gradient of the input image. The minimum Eigen values of the corner detection algorithm is computed using the following formula [14]:

$$C_{metric} = \sum \begin{bmatrix} I_x^2 & I_x I_y \\ I_x I_y & I_y^2 \end{bmatrix} = \begin{bmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{bmatrix} \qquad (4)$$

where I_x denotes the horizontal gradients of the edge map, I_y denotes the vertical gradients of the edge map, I_xI_y denotes the edges on diagonal. C_{metric} denotes the matrix with two *Eigen* values λ_1, λ_2 characterized by their shape and size of the principal component ellipse inside each filter of an image were computed. According to the used parameter *q* the output of the corner detection algorithm may not provide any candidate corner points. In that case, the algorithm reduces the corner quality parameter *q* with a step of 0.05 until corner points are detected. This process is also presented in Figure 3. The corner detection phase ends with the detection of at least one or more strong candidate corner points with a corner metric value above

 ISBN: 978-1-61208-623-1

a) b)

Figure 5. a) Detected wall boundary. (b) Detected wall corner on a wall with an unclear top right corner

the quality level. The same procedure is performed for the bottom corners of the wall. Thus, the output of the corner detection process is a set of corner points for each region of interest of the wall. For the top left part of the wall, the output is a set of points (x_i,y_i), $i \in TL$ where TL indicates the number of found corners for this region of the wall. Respectively, for the top right part of the wall the potential corner points are (x_j,y_j), $j \in TR$. The bottom left part includes the candidate corner points (x_m,y_m), $m \in BL$. Finally, the bottom right part of the image includes the candidate corner points (x_n,y_n), $n \in BR$.

D. Computation of wall width

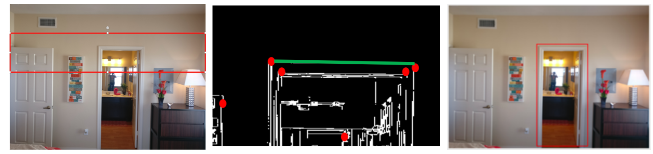
This part of the algorithm provides an estimation of the wall width according to the detected candidate corners. These corner points may include both good candidate corners but also erroneous corners. In order to avoid the negative effects of the erroneous corners in wall width measurements, the best candidates should be determined. For that reason, each corner point in all regions of interest are compared with each other. The corner points from the top part of the wall that have approximately the same y value $y_i \sim y_j$, where i and j are the two candidate corner points from the top right and top left part of the wall, are preferred. In addition, the corner points from the top left and bottom left part of the wall that have the same x value $x_i \sim x_m$, where i and m are the two candidate corner points from the bottom and top left part of the wall, are preferred. Similarly, the same procedure occurs for the bottom left and right and also for the top and bottom right part of the walls. The final corner detection is computed according to:

$$i^*, j^*, m^*, n^* = \min_{i,j,m,n} \left[|x_i - x_m| \cdot |y_i - y_j| \cdot |y_m - y_n| \cdot |x_j - x_n| \right] \tag{5}$$

The overall process is shown in Figure 4. The width, w, of a room wall is calculated by measuring the pixel distance between the two final corners.

$$w = \frac{h}{|y_{i^*} - y_{m^*}|} \cdot |x_{i^*} - x_{j^*}| \tag{6}$$

where $h/|y_{i^*} - y_{m^*}|$ is the pixel resolution r_p measured in meters/pixel. The pixel resolution can be computed according to the height of the wall, h, which is defined by the user and the number of pixels between the two corners. In a mathematical form, this is presented in (6). The detected wall boundary is demonstrated in Figure 5.



a) b) c)

Figure 6. (a) Input image with region of interest (ROI) selected. (b) Detected door corners over edge map. (c) Detected door on a wall

E. Door detection

The door detection process follows a similar approach where an edge and corner detection algorithm is used to find the location of the boundaries of the door [17]. An illustration of the overall process is given in in Figure 6. For the door detection, the region of interest is focused above the half of the wall and below the third quarter of a wall. This is because, most doors found in typical residential units have these height values. To increase the efficiency of the door corner detection algorithm, the following conditions were assumed:

- Preferred door corner should have y-axis value relatively equal to standard door height of 2.1 meter. Thus, $r_p \cdot |y_i - y_m| = 2.1$m.
- Two corner points should have relatively same y-axis values. Thus, $|y_i - y_j| \sim 0$.
- Two corner points should be separated relatively by standard door width 0.9 meters. Thus, $r_p \cdot |x_i - x_j| = 0.9$m.

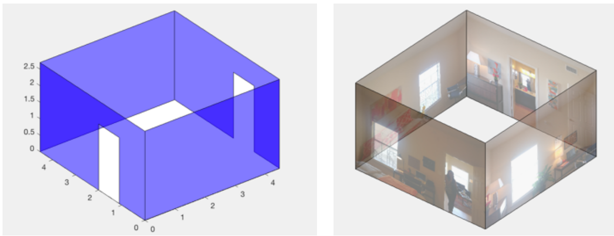
Similar to the wall detection process, the algorithm first identifies the position of the door boundaries, computes the door dimension and defines the coordinate values of its corners.

III. THE FACET MODEL

A. Constructing the 3D environment

After the successful wall and door width detection, the final coordinates of the room can be stored in a facet model format. The vector map is represented by its facet where each wall and door is defined by four coordinate points x,y,z. These coordinates indicate the respective corners. The facet representation of a single room is presented in Figure 7. For more enhanced experience, it is possible to overlay the picture as texture on the facet as presented in Figure 7b.

Using the same method and principles, the 3D vector map of the remaining rooms of the indoor environment can be constructed. One difficulty for this case, is the positioning of the rooms to form a realistic indoor environment, close to the real one. For the purpose of our investigation, we assume that the user takes four pictures per room to cover the 360 space and takes the pictures in a clockwise manner. Once the user completes this process for one room, then the user takes the pictures of the adjacent room, starting from the wall that is shared with the previous room. In that way, there is always a "calibration" or orientation point that allows the algorithm to reconstruct and attach the facet of each room and form a realistic indoor environment. This process is presented in Figure 8. In this figure, the first room is marked as initial and attached to the adjacent room according to the shared wall of the two rooms. In the next iteration, the second room becomes



a) b)

Figure 7. a) 3D vector map. (b) 3D indoor environment interior view.

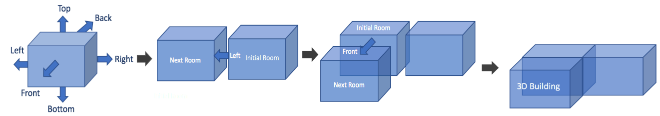


Figure 8. Integrating individual room blocks into a building based on the direction of next room with respect to initial room.

the reference room and the third room is attached according to their shared wall. This process is followed until the user captures images of all rooms of the indoor environment and the indoor environment is fully constructed.

B. Data structure

The data structure of the facet model is presented in Table I. The indoor environment in composed by a set of individual rooms. Each room has a number of walls and each wall may have a number of doors. The elements of the room structure store all the details of the indoor environment like wall width, room number, room position, wall image, wall coordinates and door coordinates. The room position field is used to determine the position of the room according to the previous one. The wall image is used as a texture and is overlaid on the facet

TABLE I. DATA STRUCTURE OF THE FACET MODEL

Room Wall Position	**Room Properties**	**Description**
Room(i).Wall(j)	Room_Position	Top, down, left, right, front & back position
---------||---------	Wall_Image	Respective room wall image
---------||---------	Width_Pixel	Wall width in pixel size
---------||---------	Width	Wall width in meter
---------||---------	Height	Wall height in meter
---------||---------	Coordinates	Wall (x,y,z) coordinates as a set of four corners
---------||---------	Door	Door (x,y,z) coordinates as a set of four corners

model to enhance the user experience. The pixel size is used for the computation of the dimensions of the walls and doors length and width and may also be used for future applications. The wall coordinates and door coordinates represent the vector format of the facet and is the most valuable element of the structure, used by the ray tracing algorithm. Finally, each facet incorporates its constitutive parameters that are used for the computation of the diffraction, reflection and transmission coefficients of the ray tracing model. For the purpose of our investigation, the wall was assumed to be made by brick material and the doors by wood material. The constitutive parameters of these materials can be found in [12]. The details of the indoor environment can be fetched using the 'Building Details' button of the main GUI, as described in the following section of the paper.

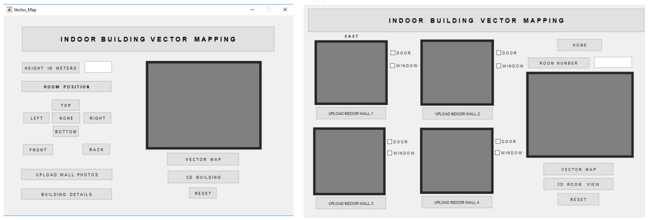
IV. RESULTS

A. Graphical User Interface (GUI)

A GUI was designed to make the use of the developed app easy and user friendly. The user can enter the standard height of the ceiling that is used as reference for the pixel resolution definition. The user also enters the room position that is used as a reference point for the construction of the 3D space. Finally, the user uploads the images for each wall of the indoor environment by using a secondary GUI as indicated in Fig 9. The user can upload four individual wall images per room and indicate if there is a door in the room. The door checkbox was used to reduce the computational cost by eliminating unwanted door detection processes. Once the user uploads the data to the system, the facet model is computed. Within the GUI, there is a button to indicate if there is a window in a wall. For the purpose of our investigation, windows were not incorporated in the facet model and is something that will be integrated in future versions of the algorithm.

B. Augmented Reality to Ray Tracing

The scenario under investigation is presented in Figure 10. A two-bedroom student dorm apartment was examined that has three main rooms. The facet model of the apartment was successfully reconstructed when the user uploads the twelve images of the walls of the three rooms. A commodity smart phone device was used to capture the images. The user spent approximately 3 minutes to take the photos and upload into the system using the GUI. When the user uploads the images to the system, the algorithm performed the pre-processing phase by down sampling and applying Gaussian filters. The input images were down sampled to different resolutions, and the best performance was met when the resolution was set to 640x480 from. It was found that the most suitable corner detection technique was '*DetectMinEigenFeatures*' of MATLAB since it provided the most accurate results and is widely used by the research community. The found coordinates of all rooms were integrated together to form the facet model of the entire indoor environment. The processed images are then embedded on to their respective walls to form a 3D interior view which is as demonstrated in Figure 10 b). It should be noted that the user inputs at the GUI, such as the height of the ceilings, the position of different rooms and the existence of doors on walls, reduced the computational cost by approximately 35%-40%. This is because, the algorithm did not search for doors in case there was no door at the room and made a more efficient positioning of the rooms to form the entire indoor space.



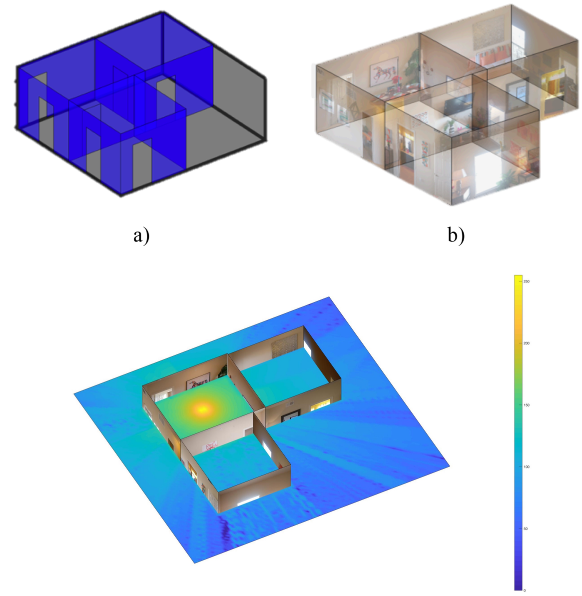
a) b)

Figure 9. a) Main GUI of indoor building vector mapping, b) Secondary GUI for uploading images of a room and mapping individual rooms.

The final step of the proposed system is to use the facet model of the indoor space as input to a ray tracing algorithm [11]. The ray tracing algorithm models the propagation of the electromagnetic waves using the Geometric Optic (GO) technique and decomposes the total field strength as sum of individual rays each one carrying a different amplitude and phase. The amplitude was computed as a combination of multiple reflection, transmission and diffraction coefficients. For the purpose of our investigation the used frequency was assumed to be of the order of the 6GHz band of 5G systems. The results are presented in Fig 10 c). It is observed that the walls and doors of the environment interact with the electromagnetic waves and change the signal strength. With the use of the proposed system, the user is able to take 12 images of the walls of the house and with just a few clicks be able to visualize the signal variation and channel condition of the 5G femtocell station inside the house. This process opens new frontiers in indoor network planning that can be performed by non-technical users and non-experts in the field. In addition, it creates new opportunities for the education of indoor channel modelling with the use of Augmented Reality (AR) devices and applications.

V. Conclusion and Future Work

This paper presented a novel image processing algorithm that is capable of creating the facet model of an indoor environment based on images captured by typical smart phone cameras. The algorithm can be considered as a simplified spatial mapping technique that leverages Augmented Reality (AR) technologies and principles. The application of the algorithm was focused on ray tracing and wireless indoor channel prediction. With the evolution of 5G networks and AR application, it is expected that there will be a great need for integrating network planning and visualization algorithms with AR technologies. It was found that the proposed solution could be used for standard indoor residential houses, but it is not efficient for large or complex indoor spaces. The proposed solution applies edge and corner detection algorithms on the images of the walls and identifies the coordinates and dimensions of the basic electromagnetic clutter, which are walls and doors. The coordinate system was based on the facet model that is used by most of the ray tracing and channel estimation algorithms. It was found that in less than 3 minutes a user could obtain signal strength estimations in a 3-bedroom house just by uploading *.jpg* images of the walls of all rooms.



c)

Figure 10. a) 3D Vector map of a building. b) 3D Interior view of a building, c) Implementation of a Ray Tracing algorithm on the facet model.

References

[1] T. Gupta and H. Li, "Indoor mapping for smart cities — An affordable approach: Using Kinect Sensor and ZED stereo camera," International Conference on Indoor Positioning and Indoor Navigation (IPIN), pp. 1-8, 2017.

[2] N. I. A. M. Nazri and D. R. A. Rambli, "Current limitations and opportunities in mobile augmented reality applications," International Conference on Computer and Information Sciences (ICCOINS), pp. 1-4, 2014.

[3] D. Chatzopoulos, C. Bermejo, Z. Huang, and P. Hui, "Mobile Augmented Reality Survey: From Where We Are to Where We Go," IEEE Access, pp. 6917 - 6950, 2017.

[4] M. E. C. Santos, "Augmented Reality Learning Experiences: Survey of Prototype Design and Evaluation," IEEE Transactions on Learning Technologies, vol. 7, pp. 38-56, 2014.

[5] L. C. Chen, N. V. Thai, and H. I. Lin, "Real-time 3-D feature detection and correspondence refinement for indoor environment-mapping using RGB-D cameras," IEEE International Symposium on Industrial Electronics, Taipei, Taiwan, pp. 1-6, 2013.

[6] X. Xu and H. Fan, "Feature based simultaneous localization and semi-dense mapping with monocular camera,"Image and Signal Processing, BioMedical Engineering and Informatics (CISP-BMEI), International Congress on, pp. 17-22, 2016.

[7] J. P. Collomosse, "Real-time environment mapping for stylised augmented reality," The 3rd European Conference on Visual Media Production (CVMP), pp. 184-184, 2006.

[8] S. Damodaran, A. P. Sudheer, and T. K. Sunil Kumar, "An evaluation of spatial mapping of indoor environment based on point cloud registration using Kinect sensor," Control Communication & Computing India (ICCC), International Conference on, pp. 545-552, 2015.

[9] S. Singh and R. Singh, "Comparison of various edge detection techniques," 2nd International Conference on Computing for Sustainable Global Development (INDIACom), New Delhi, pp. 393-396, 2015.

[10] X. Ge, "Multipath Cooperative Communications Networks for Augmented and Virtual Reality Transmission," IEEE Transactions on Multimedia, vol. 19, no. 10, pp 2345-2358, 2017.
[11] G. Koutitas, A. Karousos, and L. Tassiulas, "Deployment Strategies and Energy Efficiency of Cellular Networks," IEEE Transactions on Wireless Communications, vol. 11, no. 7, pp. 2552-2563, 2012.
[12] F. S. D. Adana, O. G. Blanco, I. G. Diego, J. P. Arriaga, and M.F. Catedra, "Propagation model based on ray tracing for the design of personal communication systems in indoor environments," IEEE Transactions on Vehicular Technology, vol. 49, no. 6, pp. 2105-2112, 2000.
[13] S Singh and R Singh, "Comparison of various edge detection techniques," Computing for Sustainable Global Development (INDIACom), 2nd International Conference on, pp. 393-396, 2015.
[14] X. C. He and N. H. C. Yung, "Corner detector based on global and local curvature properties," Optical Engineering 47, no. 5, pp. 1-4, 2017.
[15] P. Ram and S. Padmavathi, "Analysis of Harris corner detection for color images," International Conference on Signal Processing, Communication, Power and Embedded System (SCOPES), pp. 405-410, 2016.
[16] Bastanlar, Yalin, and Y. Yardimci, "Corner validation based on extracted corner properties", Computer Vision and Image Understanding, 112, pp. 243-261, 2008.
[17] X. Yang and Y. Tian, "Robust door detection in unfamiliar environments by combining edge and corner features," IEEE Computer Society Conference on Computer Vision and Pattern Recognition - Workshops, San Francisco, CA, pp. 57-64, 2010.

 ISBN: 978-1-61208-623-1