Image Based Navigation System for Pedestrians in an Indoor Environment

Kazeem Oyebode, Shengzhi Du, Barend Jacobus van Wyk and Karim Djouani
Department of Electrical Engineering, Tshwane University of Technology, Pretoria, South Africa.
dushengzhi@gmail.com

***Abstract*— Indoor navigation systems provide means to guide pedestrians to their various destinations. While many tools that take advantage of the Global Positioning System (GPS) for outdoor navigation exists, their usefulness is limited to the availability of GPS signal reception, which is usually poor in indoor environments. In this research, we propose a method that employs only images for indoor pedestrian navigation. In the proposed method, a map of the indoor environment is first transformed into a graph model where features of indoor environment are attached to graph nodes and their distances represented by the graph edges. Feature images of initial and destination locations are provided by the pedestrian who needs to be guided. These images are fed into the created graph model and thereafter the Speeded-Up Robust Features (SURF) is then used to find a match to these images to discover their corresponding graph nodes. Graph nodes are identified in a manner that corresponds to pedestrian localized position and destination. Leveraging on these nodes in the graph, models are proposed to find the shortest path to user's destination with instructions and graphical navigation path to enhance maneuverability. Experiment carried out on an indoor environment of the French South African Institute of Technology building, (Tshwane University of Technology) shows encouraging results.**

***Index Terms*— Image Localization; Image Matching; Image Processing; Pedestrian Navigation.**

I. INTRODUCTION

Navigation systems are of great advantage to users in traversing unfamiliar terrains with great accuracy. These systems are usually built to interact with the Global Positioning System (GPS). For outdoor users, such as commuters, the GPS device would guide them from any location to a desired destination with high precision. However, in indoor environments, the functionality of these devices is hampered due to highly degraded or unavailable GPS signals.

Early research in computer vision-based-navigation is carried out [1] and [2], where markers [3] or features snapped from the environments by the pedestrians are compared to those stored in a database. When a match is found, the pedestrian is localized. One of the problems with this approach is that additional infrastructures are required to be installed in the environment of interest in order to facilitate localization. This challenge was addressed by the introduction of sensors to vision-based navigation systems. One of such is proposed in [4], where two sensors are combined; the Inertia Measurement Unit (IMU) and the visual images for navigation. The IMU is used to track the navigation trajectory of the pedestrian, while visual images are used to improve the accuracy of the IMU. However, one of the drawbacks of the model is its cost, because two IMU devices have been used for their analysis with one of them installed on the user's foot. In addition, the pedestrian is expected to wear a camera on the chest. This indicates that the model may not be deployed in a real-life scenario. The IMU wearable device is also investigated in [5] and [6] for indoor navigation. Also, in [7], research is carried out on an indoor environment, where office numbers are used to indicate the initial position and destination of the pedestrian. However, this approach might not work in the absence of office numbers. In [8], a drone is developed to assist pedestrians to find their way in both indoor and outdoor environment, the drone is integrated with a mobile application, wherein a call can be placed to it. Research has also been carried out on pedestrian navigation using smartphones. In [9], a method is developed where user navigation is based on camera phone. Firstly, the video of the indoor environment is obtained and thereafter, key frames from the videos and their camera positions are obtained. These key frames are then used to localize the pedestrian (armed with a camera phone) with additional markers installed on the floor to aid robust localization. In [10], an image-based approach for pedestrian localization in an outdoor environment using image matching algorithm is investigated. In recent time, navigation systems have also been developed for pedestrians on the wheel chair. In [11], an intelligent wheel chair is proposed, where its navigation system is based on obstacle detection and avoidance. In [12], a pedestrian localization is proposed for outdoor systems where image descriptors are used to find a match between a query image and descriptors of images stored in the database.

A robust image matching algorithm is needed for localization as images acquired from pedestrians may be taken in a non-upright position, which means acquired images may have been rotated. There are various image matching methods in literature. One of such is the template matching algorithm. This model compares an image in question to reference images and the best match is picked [13] [14]. This is carried out by comparing their grayscale intensity values [15]. The Speeded-Up Robust Features (SURF) model [16] is also used for image matching and has the ability to match images of different orientation. The FAST method [17] uses extracted features (corners in images) for image matching. The Local Binary Pattern (LBP) has also been used for image recognition [18].

Some of the methods of pedestrian navigation proposed in literature requires that devices be mounted on the foot or the chest or a marker be installed in a given environment or might require expensive device such as drones. Hence, laborious work is needed on the part of the pedestrian to achieve indoor navigation [19]. This suggests that such systems may not be explored commercially. Therefore, in this paper, we propose a method for pedestrian navigation in an indoor environment

using only visual images. First, leveraging on graph theory, the environment is transformed/mapped into a graph model. Feature images in the indoor environment are attached to graph nodes while edges of the graph indicate the distances between nodes. When a pedestrian supplies images (initial and final locations) of the environment, image features are matched using SURF and the corresponding initial and destination nodes are discovered on the graph. The pedestrian is then localized on the graph and the shortest path to the destination is investigated, while providing sequence of images and navigation instructions that will be observed along the path.

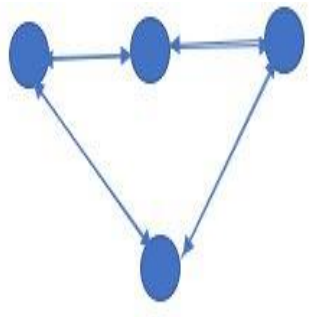
There are distinctions that can be observed from previous image based pedestrian navigation system. In [2], a localization problem is only solved where a pedestrian supplies an image, and this is matched against a collection of database images in an outdoor environment. Once an image match is found, the pedestrian is localized. In addition, the issue of false localization is not addressed. However, in our proposed model, we solved the localization problem for an indoor environment using graph theory and also tackle problems that may arise from false localization owing to identical images found in different locations. In addition, a robust and intuitive model for suggesting optimal navigation path that shows direction with labelled features in the graph path is put forward. In [4], a combination of sensors is used for aiding navigation (IMU and visual images), however, in our proposed model, only visual images are used for navigation. Also, in [9] an image-based navigation system is proposed leveraging on installed markers, but in our approach, markers are not installed in the considered environment. In [7], pedestrian localization is based on office numbers, however, not all offices have numbers. Section II of this paper puts forward the concept of graph theory and it application to indoor environment modelling. The following section evaluates the proposed model. Discussion and conclusion sections followed respectfully.

II. Indoor Navigation Using Visual Images

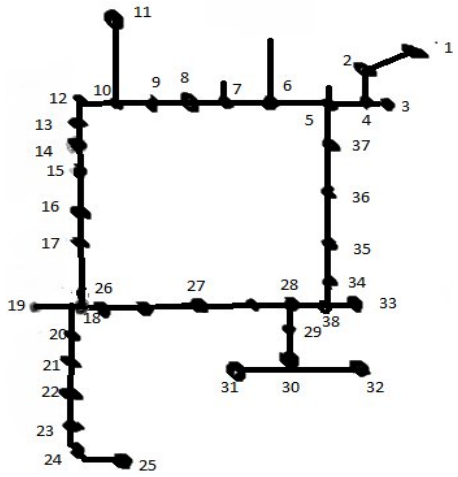
The proposed model comprises three stages. Firstly, a transformation/mapping stage where the indoor environment is transformed into a graph model in which images are assigned to node(s). Thereafter, the graph is transformed into a graph-map to give positions to identified nodes. Secondly, images supplied by pedestrians taken from the indoor environment are uploaded and their corresponding nodes are discovered using SURF. In addition, the positions (x,y) of these nodes in the graph-map are identified. The coordinates are then used to calculate the shortest path to the destination. Finally, the shortest part is realized and images and instructions aiding the movement of the pedestrian to the destination is given.

A. Generating a Graph-Map of FSATI

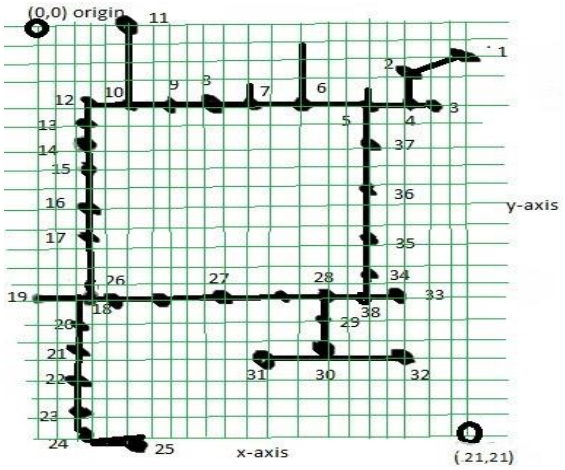
The FSATI indoor environment encapsulates lecture theatres, offices, toilets, printer room, laboratories and dining rooms, with an estimated size of 16,000 square meters. A graph G = (N, E) consists of nodes N and edges E as observed in Figure 1(a). In the figure, the circular shapes represent nodes, while the arrows represent edges. The edges are bidirectional, i.e. they can traverse either directions, as indicated. Intuitively, this concept of a graph can be used to model any environment where important features (images) in the environment are attached to nodes and edges are the pathways or directions leading to the adjoining node as observed in Figure 1(b).



(a)



(b)



(c)

Figure 1: (a) A graph with circles representing nodes and arrows representing edges, (b) A graph map of FSATI building where circles represent nodes and edges as lines connecting nodes, and (c) A graph map of FSATI building

In Figure 1(b), the nodes are assigned numbers, as observed. The graph has been manually drawn. An example of hand drawn maps for robotic navigation can be seen in [20]. While the graph in Figure 1(b) gives the layout of nodes and their pathways or connections in an indoor environment, they lack

positioning. In order to achieve positioning, the nodes are assigned coordinates using the 2d coordinate system. Therefore, the graph in Figure 1(b) where G = (N, E) is transformed into a graph-map (GM) where $GM = (N_{(x,y)}, E)$ in Figure 1(c). The x and y coordinates of each node can now be obtained and the distance separating two nodes can be derived. This distance is also referred to as edge weight.

B. Adding Images to GM Nodes

Each feature image acquired from the considered indoor environment is attached to a node n n ($n \epsilon N_{(x,y)}$). Nodes are assigned numbers (Figure 1(c)) (n ∈ {1,2…….38}) such that when a node number is found, its corresponding feature image(s) can be retrieved. Also, when one or more images attached to a node are matched by an input image, the node can be retrieved.

(a) (b)

(c) (d)

Figure 2: (a) A portion of the indoor environment, (b) Image attached to node 18, (c) Image attached to node 18, and (d) Image attached to node 18.

C. Pedestrian Localizarion and Destination Identification

The SURF model is used to match the images obtained from pedestrians to that attached to graph nodes. Once images are successfully matched, their corresponding nodes can be discovered on the graph model. Therefore, the shortest path to destination can be calculated. The SURF uses a box filter to convolve the image under question, using integral images and thereafter generates an image scale space. The next stage is to detect features in the image scale-space. These features (blobs) are detected using the determinant of the Hessian matrix (H)) [21]. Each element in the matrix holds the second-order derivate of the image V (Equation (1)). The determinant of the Hessian matrix (Equation (2)) is known to detect sudden changes in color or greyscale intensity of image V. These detected features (sudden change in color) are described in a vector form and are immune to rotation and translation. These features are then used to match images attached to graph nodes. The highest matched feature is discovered and its corresponding graph node is identified.

$$H = \begin{matrix} \frac{\partial^2 V}{\partial^2 x} & \frac{\partial^2 V}{\partial x \partial y} \\ \frac{\partial^2 V}{\partial x \partial y} & \frac{\partial^2 V}{\partial^2 y} \end{matrix} \quad (1)$$

$$det(H) = \frac{\partial^2 V \partial^2 V}{\partial x^2 \partial y^2} - \left(\frac{\partial^2 V}{\partial x \partial y}\right)^2 \quad (2)$$

The discovery of nodes aid pedestrian localization and path planning. However, the localization strategy adopted by SURF has its challenges as the pedestrian might be localized on a wrong node leading to false localization. This is the case when an input image I_i returns the highest matched feature of a wrong image attached to a node. Possibly the image is found in more than one location and the wrong node has been retrieved.

$$D = [d_{1,}\, d_2 \ \ldots\ldots\ldots d_z]^T \quad (3)$$

$$P = [\left(\frac{d_1}{d_2}\right),\ \left(\frac{d_2}{d_2}\right) \ldots\ldots\ldots \left(\frac{d_z}{d_2}\right)]^T \quad (4)$$

The probabilities of matching I_i with node images are obtained in Equation (4). When two or more matches have a minimum probability of 0.8, then their corresponding nodes (N > 1) are potential candidates for false or true pedestrian localization. The pedestrian is then instructed to input another nearby image, within the vicinity, until a single node (N == 1) is returned with a minimum probability of 0.8. This localization strategy is shown in Equation (5) and (6). I_f is also an input image that discovers the destination node I_f can be retrieved from a text to image conversion where a pedestrian selects a text say "printer room" and then converted to I_f attached to a node. Once the initial and destination nodes are retrieved, their corresponding positions ($S_{(x,y)}$ $E_{(x,y)}$) on the GM are known.

$$f(I_i) = \begin{cases} f(I_i), & P(N > 1|\, I_i) \quad at\ P \geq 0.8 \\ N, & P(N == 1|\, I_i)\ at\ P \geq 0.8 \end{cases} \quad (5)$$

$$f(I_f) = \begin{cases} f(I_f), & P(N > 1|\, I_f) \quad at\ P \geq 0.8 \\ N, & P(N == 1|\, I_f)\ at\ P \geq 0.8 \end{cases} \quad (6)$$

The localization strategy adopted by SURF has therefore been improved. Given any input image I_i its matched feature values across the entire images of nodes are collected in a vector form (Equation (3)), given that we have a total of z images attached to all nodes in the graph. Thereafter, each element in Equation (3) is normalized and transformed into a probability. Given that element d_2 has the highest number of feature-match with I_i Equation (3) is normalized to give Equation (4).

D. Algorithm for Shortest Path

The Dijkstra algorithm [22] finds the shortest path in a graph model but unfortunately, it fails when an edge weight is negative and also performs an exhaustive search amongst nodes. We take a different approach from the Dijkstra algorithm to suit the research work in two areas. Firstly, the shortest path is discovered bearing in mind the destination node in order to minimize exhaustive search. Therefore, when a path leading to the destination is collinear, this path is assumed the shortest path and the algorithm terminates. Given some initial and destination nodes at $S_{(x,y)}$and $E_{(x,y)}$respectively, a path is collinear if either Equations (7), (8) or (9) holds true.

$$S_{(x)} - E_{(x)} = w;\ \{w = 0\} \quad (7)$$

$$S_{(y)} - E_{(y)} = w;\ \{w = 0\} \quad (8)$$

$$((S_{(x)} - (S+1)_{(x)})^2 + (S_{(y)} - (S+1)_{(y)})^2)^{1/2} + ((E_{(x)} - (S-1)_{(x)})^2 + (E_{(y)} - (S-1)_{(y)})^2)^{1/2} == ((S_{(x)} - (E)_{(x)})^2 + (S_{(y)} - (E)_{(y)})^2)^{1/2} \quad (9)$$

However, when nodes leading to the destination do not satisfy neither equation, the algorithm then searches for any two paths leading to the destination and choose the shortest distance from a collection of possible paths discovered. One may suggest that selecting the path with minimal nodes encountered can provide the shortest path; however, this may be wrong as their Euclidian distances may suggest otherwise. Secondly, negative weight edges (distance between two nodes) are avoided using the Euclidean distance *d* in Equation (10). The shortest path algorithm is given Algorithm 1.

$$d = ((S_{(x)} - (S+1)_{(x)})^2 + (S_{(y)} - (S+1)_{(y)})^2)^{1/2} \quad (10)$$

The shortest path algorithm is given below.

1. **Input**: *GM = ($N_{(x,y)}$, E)*, start and end nodes ($S_{(x,y)}$, $E_{(x,y)}$ obtained from Equations (5) and (6) respectively
2. **Output***:* Shortest path between $S_{(x,y)}$ and $E_{(x,y)}$
3. **Begin:**
4. **Initialize**
5. List<node> *path* (Create a variable, *path*, that stores a single path from $S_{(x,y)}$ to $E_{(x,y)}$
6. List<List<node>> *paths* (Create a variable, *paths*, that stores a collection of possible paths from $S_{(x,y)}$ to $E_{(x,y)}$
7. *distances[]* (create variable *distance* to store a collection of distances in *paths*
8. *pathCounter* = 0 (counts the number of paths from $S_{(x,y)}$, $E_{(x,y)}$)
9. **function ShortestPath (***GM*, $S_{(x,y)}$, $E_{(x,y)}$**)**
10. **if** (Equation (7)**)**) || (Equation (8)) || ((Equation (9))
11. *path.add(n)* (add all the nodes from $S_{(x,y)}$, $E_{(x,y)}$)
12. *paths.add(path)*
13. **else**
14. *path.add(S)* (add start node)
15. ***while (****pathCounter < 2)*
16. **for each** node *n* in identified path *id* from $S_{(x,y)}$ to $E_{(x,y)}$
17. *path.add(n)*
18. **end for each**
19. *paths.add(path)*
20. *pathCounter = pathCounter + 1*
21. ***end while***
22. **end if**
23. **for each** *p* in *Paths*
24. **for each** node $n_{(x,y)}$ in *p*
25. $x_1 = n_{(x)}$, $y_1 = n_{(y)}$
26. $x_2 = (n+1)_{(x)}$, $y_2 = (n+1)_{(y)}$
27. *distance = distance* + $((x_1 - x_2)^2 + (y_1 - y_2)^2)^{1/2}$
28. **end for**
29. *distances [] = distance*
30. **end for**
31. find the index of the least value in *distances* and assign to *l*
32. *paths[l]* is the shortest path
33. **end if**
34. **end function**

Algorithm 1: Shortest path algorithm

In Algorithm 1, when a path leading to the destination node is collinear, the path from the initial to the final node is considered the shortest path. However, when the initial and final nodes are non-collinear, two paths leading to the destination node are discovered and the shorter distance to the destination is considered the shortest path.

E. Generating Route Guide And Instructions

Route instructions show pedestrian's current location in the graph and how to navigate to their destination. In addition, guided instructions are needed to help pedestrians know where and how to make a turn. Graphical images are provided to enhance pedestrian maneuverability. There are four scenarios considered. The first is when the pedestrian is localized on the constant y axis of the GM (Figure 1(c)) (pedestrian at $S_{(x)}$) and based on the shortest path, if $S_{(x)}$ is less than the next node $(S+1)_{(x)}$, then the pedestrian is instructed to face the right direction in the graph and proceed to walk (Equation (11)). Otherwise, the system instructs the pedestrian to face the left direction. Either way, a feature is shown that informs the pedestrian that he is on the right path. A red line is drawn in Figure 3 to highlight this path (node 19 to 31).

$$f((s+1),s) = \begin{cases} turn\ right, & S_{(x)} < (S+1)_{(x)} \\ turn\ left, & S_{(x)} > (S+1)_{(x)} \end{cases} \quad (11)$$

For example, in Figure 1(c), if the navigation path is from node 9 to 8, which is on a constant *y* axis, node 9 is $S_{(x)}$, and has *x*,*y* position (7, 4) and node 8 is $(S+1)_{(x)}$ with (9,4). The second node in this path is 8, its *x* position is 9, while its predecessor's node has its *x* value to be 7. Since 9 is greater than 7, then the pedestrian is instructed to turn right and move from node 9 to 8. A feature image of node 8 is shown to the pedestrian. However, if the navigation path was from node 8 to 9, then the pedestrian is instructed to move in the left direction.

The second scenario is when the pedestrian is localized on the constant *x* axis of the *GM* (Figure 1(c)). Based on the shortest path, if the *y* value of the initial node $(S_{(y)})$ is greater than the next node $(S+1)_{(y)}$, then the pedestrian is instructed to face the right direction in the map and proceed to walk. Otherwise, the pedestrian is instructed to face the left direction (Equation (12)). Either way, a feature is shown that guides the pedestrian and a red line is drawn in Figure 3 to highlight this path.

$$f((s+1),s) = \begin{cases} turn\ right, & S_{(y)} > (S+1)_{(y)} \\ turn\ left\ , & S_{(y)} < (S+1)_{(y)} \end{cases} \quad (12)$$

The third scenario comes to a realization when the pedestrian is currently on node s and s-1, s, s+1 have either constant *x* or *y* values. In this case, the pedestrian is instructed to keep moving straight as seen in Equation (13). A feature image is also presented as a guide and a red line is drawn in Figure 3 to highlight this path.

$$f((s-1),s,(s+1)) = \begin{cases} keep\ moving\ striaght, (S-1)_{(y)} == S_{(y)} == (S+1)_{(y)} \\ keep\ moving\ striaght\ , (S-1)_{(x)} == S_{(x)} == (S+1)_{(x)} \end{cases} \quad (13)$$

The fourth scenario is when the pedestrian is moving from a given point *a* to *b*, however both their *x* or *y* values differ (Equation (14)). An example of this scenario is when the pedestrian moves from node 1 to 2 in Figure 1(c). In this case, the pedestrian is instructed to move straight. A red line is drawn in Figure 3 to highlight this path.

$$f(s,(s+1)) = \{move\ straight, S_{(y)} \neq (S+1)_{(y)}\ and\ S_{(x)} \neq (S+1)_{(x)} \quad (14)$$

F. Navigating Graph and Image Sequence

Navigating graph and image sequence enables pedestrians to have a smooth navigation path to their destination. Navigation path is highlighted in red with additional instructions and numbered sequence of images matching node numbers on the graph. Assuming a pedestrian is on node number 19 and wants to navigate to the male's toilet (node 31). The pedestrian uses a mobile device to take an image of a nearby feature, on node 19; and therefore selects a text "males'

toilet", which is then transformed to a feature image. These images are used by Equation (5) and (6) for the discoveries of node 19 and 31. Algorithm 1 finds the shortest path (shortest path is 19-18-26-27-28-29-30-31), while this navigation path is highlighted in Figure 3 with feature images (Figure 4), seen along selected path. At node 19 (Figure 3), the "start" word is written, indicating the pedestrian has been localized. And on node 31, the "stop" is written, suggesting the destination node. Highlighting navigation path similar to Figure 3 also has an advantage of implicitly informing pedestrians, where to turn as not all scenarios of turns, in the real world, are addressed in the previous section. For instance, at round-about or at forks.

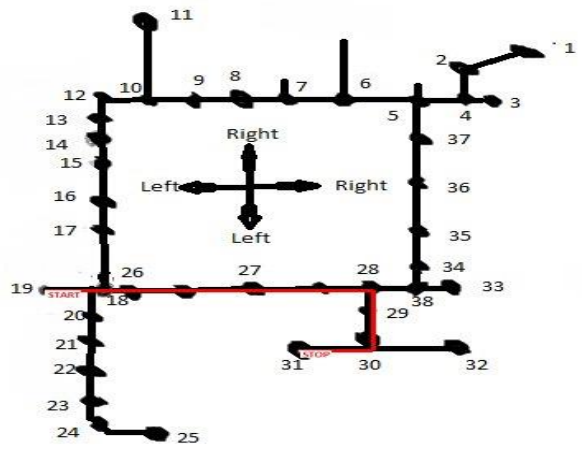


Figure 3: Graph highlighting navigation path in red

You are here

(a) Node 19

Turn Right

(b) Node 18

Keep moving straight

(c) Node 26

Keep moving straight

(d) Node 27

Keep moving straight

Node 28

Turn Left

(e) Node 29

Keep moving straight

Node 30

Turn Left

(f) Node 31

Figure 4: Instructions and sequence of images in navigation path selected in Figure 3

III. EVALUATION

The proposed model is evaluated on two fronts. The first is on the node identification capability when images are supplied. This is the ability to recognize nodes via images taken by mobile phones especially when only portions of the image are captured. The ability to recognize and match images is crucial as it aids localization and path planning from initial location to destination. Hence, a rigorous process is adopted to test the robustness of proposed model on pedestrian localization. 58 images [23] captured from mobile phones at various times, and of varying degree of rotation have been considered for the evaluation of pedestrian localization in the considered environment. Any of these 58 images may serve as either the source or destination image.

Evaluation metric used is the Pedestrian Localization Accuracy (PLA) (Equation (15)).

$$PLA = \frac{total\ number\ of\ currently\ classfied\ Images}{total\ number\ of\ Images\ referenced} \quad (15)$$

The second evaluation verifies the correctness of the shortest route discovered. We identify various navigation routes and benchmark them against the ground truth. The ground truth gives the shortest path and obtained by taking measurements on the actual environment.

Table 1
Comparison of Image Matching Models

Model	Image Number	Number of Correctly Classified	Number of Wrongly Classified	PLA (%)
Template Matching	58	20	38 34.5	34.5
Fast	58	14	44	24
SURF	58	48	10	82.8
Proposed Model	58	55	3	94.8

Table 1 shows the localized accuracies of different localization models investigated. The table shows that the proposed localization strategy outperforms other considered

models such as the FAST and the template matching methods. SURF model has the capability to match images of different orientation (Figure 5), thus giving it a score of 82.8 % for PLA. The proposed model exhibits better performance by ensuring that the input image provided by the pedestrian is unique before localization, which results in a PLA score of 94.8 %. The robustness of SURF is also demonstrated in Figure 5. Apart from the input image (Figure 5(b)) being slightly rotated, image features are not completely captured, yet it is able to localize accurately (Figure 5(c)). The input image is recognized and its corresponding node is retrieved (localization).

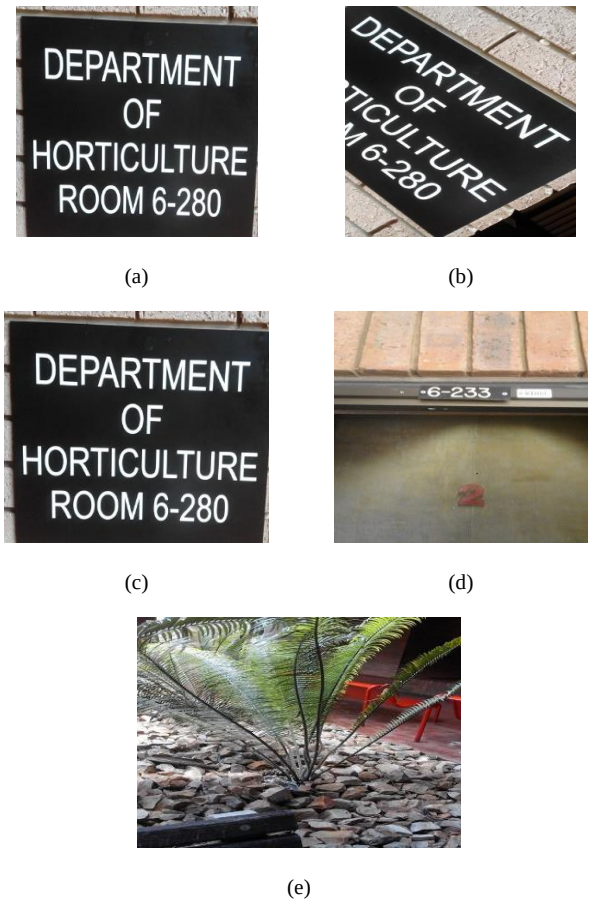


Figure 5: (a) Ground truth (image attached to node 33), (b) Input image from camera phone, (c) Returned image via SURF (localized at node 33), (d) Returned image via template matching (localized at node 7), and (e) Returned image via FAST (localized at node 37)

Table 2
Shortest Path Evaluation of Algorithm 1

Nodes (see Figure 1(c))	Shortest Path by Proposed Model	Ground Truth
21 - 11	25, 24, 23, 22, 21, 20, 18, 17, 16, 15, 14, 13, 12, 10, 11	25, 24, 23, 22, 21, 20, 18, 17, 16, 15, 14, 13, 12, 10, 11
19 - 32	19, 18, 26, 27, 28, 29, 30, 32	19, 18, 26, 27, 28, 29, 30, 32
1 - 28	1, 2, 4, 5, 37, 36, 35, 34, 38, 28	1, 2, 4, 5, 37, 36, 35, 34, 38, 28
11 - 27	11, 10, 12, 13, 14, 15, 16, 17, 18, 26, 27	11, 10, 12, 13, 14, 15, 16, 17, 18, 26, 27
6 - 27	6, 5, 37, 36, 35, 34, 38, 28, 27	6, 5, 37, 36, 35, 34, 38, 28, 27
3 - 31	3, 4, 5, 37, 36, 35, 34, 38, 28, 29, 30, 31	3, 4, 5, 37, 36, 35, 34, 38, 28, 29, 30, 31
19 - 35	19, 18, 26, 27, 28, 38, 34, 35	19, 18, 26, 27, 28, 38, 34, 35
35-20	35, 34, 38, 28, 27, 26, 18, 20	35, 34, 38, 28, 27, 26, 18, 20

Evaluation of Algorithm 1 is shown in Table 2. The shortest paths suggested by the proposed Algorithm are benchmarked against actual shortest paths in the environment. The test results show that all considered scenarios passed.

IV. Discussions

This research presents pedestrian navigation using only images. The proposed approach employs graph theory to model the considered indoor environment and thereafter transformed to a graph map (GM), where nodes acquire positions. The GM is then used to develop guides which aid pedestrians in traversing from their initial location to their destination with great accuracy. When non-unique image features are provided by the pedestrian for localization, there is a high possibility of false localization. Therefore, the proposed model instructs pedestrian to input another nearby feature until a match probability, not less than 0.8, of a single node is obtained. Thus, the pedestrian is localized.

V. Conclusion

The contributions in this research are three-folds. First, the recognition of supplied images by pedestrians for their localization on a graph node. Second, the path planning from the localized node to destination node. Finally, the introduction of navigation guide together with the sequence of images that will be encountered in the chosen path. These have been achieved by exploring the graph theory for indoor environment navigation. Indoor feature images are attached to graph nodes and are discovered (nodes) by matching images uploaded via mobile phones. In addition, the ability to transform the generated graph into a graph map, where nodes have coordinates, has provided the platform to develop models for the shortest path to destination and instructions for pedestrian navigation. The proposed navigation model, which does not require markers to be installed in the considered indoor environment, can also be extended to other scenarios such as shopping malls, especially for pedestrians on a wheel chair in order for them to optimize their movement.

References

[1] H. Aoki, 8. Schiele and A. Pentland, "Realtime personal positioning system for wearable computers," in ISWC, Proceedings of the 3 rd IEEE International Symposium on Wearable Computers , 1999.
[2] D. Robertson and R. Cipolla, "An image-based system for urban navigation"," in Proceedings of British Machine Vision Conference, 2004.
[3] H. Hile and G. Borriello, "Information overlay for camera phones in indoor environments," in 3rd International Symposium on Location- and Context Awareness, Lecture Notes in Computer Science, 2007.
[4] C. Keßler, C. Ascher, M. Flad, and G. F. Trommer, Gyroscopy and Navigation, vol. 3, no. 2, p. 79–90, 2012.
[5] F. Hartmann, D. Rifat and W. Stork, "Hybrid Indoor Pedestrian Navigation Combining an INS and a Spatial Non-Uniform UWB-network," in 19th International Conference on Information Fusion, Heidelberg, 2016.
[6] M. Placer and Stanislav Kovačič, "Enhancing Indoor Inertial Pedestrian Navigation Using a Shoe-Worn Marker," Sensors, vol. 13, pp. 9837-9859, 2013.
[7] R. Nitami, A. Suzuki and Yoshitoshi Murata, "Development of a Pedestrian Navigation System Without Additional Infrastructures," in

2014 International Conference on Indoor Positioning and Indoor Navigation, 2014.

[8] "SKYCAL," MIT, [Online]. Available: http://senseable.mit.edu/skycall/. [Accessed 2 2 2017].

[9] W. Elloumi ; K. Guissous ; A. Chetouani ; R. Canals ; R. Leconge ; B. Emile ; S. Treuillet, "Indoor navigation assistance with a Smartphone camera based on vanishing points," in "Indoor navigation assistance with a Smartphone camera based on vanishing points," International Conference on Indoor Positioning and Indoor Navigation, Montbeliard-Belfort, 2013.

[10] W. Zhang and J. Kosecka, "Image Based Localization in Urban Environments," in Proceedings of the Third International Symposium on 3D Data Processing, Visualization, and Transmission (3DPVT'06), 2006.

[11] E. Y. Kim, "Wheelchair Navigation System for Disabled and Elderly People," Sensors, vol. 16, no. 11, 2016.

[12] S. Lynen, M. Bosse and R. Siegwart, "Trajectory-Based Place-Recognition for Efficient Large Scale Localization," Int J Comput Vis , vol. 124, pp. 49-62, 2017.

[13] H. Peng, F. Long and Z. Chi, "Document Image Recognition based on template matching of component block projections,," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 25, no. 9, pp. 1188-1192, 2003.

[14] Q. I. Pham, R. Jalovecký and M. Polášek, "Using template matching for object recognition in infrared video sequences," in IEEE/AIAA 34th Digital Avionics Systems Conference (DASC), Prague, 2015.

[15] Z. Yang, "Fast Template Matching Based on Normalized Cross Correlation with Centroid Bounding," in International Conference on Measuring Technology and Mechatronics Automation, Changsha, 2010.

[16] H. Bay, A. Ess, T. Tuytelaars, and Luc Van Gool, "Speeded-Up Robust Features (SURF)," Computer Vision and Image Understanding, vol. 110, no. 3, pp. 346-359 , 2008.

[17] M. Trajkovic,. and M. Hedley., "1998. FAST corner detector," Image and Vision Computing, vol. 16:, pp. 75-87, 1998.

[18] A. Satpathy, X. Jiang and H. L. Eng, "LBP-Based Edge-Texture Features for Object Recognition," IEEE Transactions on Image Processing, vol. 23, no. 5, pp. 1953-1964, 2014.

[19] L. Ruotsalainen, H. Kuusniemi and R. Chen,, "Overview of methods for visual-aided pedestrian navigation," in Ubiquitous Positioning Indoor Navigation and Location Based Service, , Kirkkonummi, 2010.

[20] F. Boniardi, A. Valada, Wolfram Burgard, Gian Diego Tipaldi, "Autonomous Indoor Robot Navigation Using a Sketch Interface for Drawing Maps and Routes," in IEEE International Conference on Robotics and Automation (ICRA), Sweden, 2016.

[21] H. Bay., A. Ess, T. Tuytelaars, and L. Van Gool, "Speeded-up robust features (SURF)," Lecture Notes in Computer Science, vol. 3951, pp. 404-417., 2006..

[22] J. Lu and C. Dong,, "Research of shortest path algorithm based on the data structure," in International Conference on Computer Science and Automation Engineering, Beijing, 2012.

[23] K.O Oyebode, S. Du, B.J. van Wyk, and K Djouani,, "Dataset," 18 January 2018. [Online]. Available: https://drive.google.com/open?id=17BIoUzeE6yLycwn2HVqas8O9I0_HelbI. [Accessed 26 March 2018].