Trends in
Applied Sciences
Research

Developing Cognitive Map using Blueprint Map

Hazim A. Farhan, Hussein H. Owaied and Suhaib I. Al-Ghazi

Faculty of Information Technology, Middle East University, P.O. Box 42, Amman, 11610, Jordan

Corresponding Author: Hazim A. Farhan, Faculty of Information Technology, Middle East University, P.O. Box 42, Amman, 11610, Jordan Tel: 00962 64790222 Fax: 00962 64129613

ABSTRACT

This study presents methodology to develop a cognitive map using blueprint map. This methodology based on the Functional Model of Human System philosophy and the Framework Model of Intelligent Robot. The proposed model will be presented as knowledge to be added to the knowledge base of the intelligent robot as dynamic knowledge created during the navigation and interaction of the robot in the environment. Building Cognitive Maps can facilitate the intelligent robot work without the need of direct human interaction. It could also be used to reduce the time to reach a target object in a new environment and will need less information from the environment to build knowledge which means less space of knowledge stored and then retrieved. Therefore, save time for retrieve the knowledge from the knowledge base and is efficiently and economically in term of processing. Cognitive Maps supported by knowledge from architectural blueprints maps will also increase the ability of the intelligent robot to sweep more sophisticated buildings. While most of the previous researches, for indoor intelligent robot mapping, assuming few rooms for the building to be surfed. The proposed method will utilize the intelligent robot to sweep a building without taking in consideration the size of it.

Key words: Artificial intelligent, cognitive map, expert systems, knowledge-based systems, intelligent robot

INTRODUCTION

In Recent years intelligent robot mapping attracted many researchers in the field of intelligent robotics and artificial intelligence. Thrun (2002) made a good survey discussing the techniques that have been used on intelligent robot mapping. There are many concepts have raised, some of it researched concepts. The psychologists thought about how human thinks and build maps and addressed it with the machines architecture like Vasudevan *et al.* (2007) while others tried to find a new mapping concepts to perform human jobs far away from human thinking concepts like Kuipers (2000), Remolina and Kuipers (2004) and Goel *et al.* (1999).

However, recent studies assures that to make intelligent robots perform human jobs need to use the concept of human intelligence. Chandler (2009) and the psychologists thought about how an individual represent the problem space for the Cognitive Mapping which are the bases of the proposed model to develop the Cognitive Mapping.

Many definitions were written for cognitive mapping, such as Stephanie (1998) and Endo and Arkin (2003) but the simple definition written by Yeap and Jefferies (2000) is the process by which one's mental representation of the environment is acquired and manipulated. Also it has been

enhanced by many researchers gaining advantages from other fields like using finger print for places to enhance the process of cognitive mapping include certain characteristics of developed map (Vasudevan *et al.*, 2007).

Depending on maps during robot mapping field some studies have done using geographical maps during mapping, they tried to put a theory to topological maps (Remolina and Kuipers, 2004; Schonberg *et al.*, 1995).

Intelligent robots presently needs to be more dynamic, reliable and having the ability to be mobilized even within indoor environments or buildings that is abandoned without the needs of guidance. This ability will facilitate the intelligent robot reaching its target in the simplest and shortest way. The mentioned ability is more demanded for the intelligent robots with special jobs such as police or fire fighters intelligent robots where human guidance is not possible. The talent which previously described can be hardly reached without external knowledge recourses, such as architectural maps (architectural blueprints maps) or geographical maps. There are no satellite images for indoor environments, thus geographical maps are not useful inside such environments.

Therefore, this study concentrates on the architectural blueprints maps where the outdoor environments are completely covered. There are a lot of differences between indoor and outdoor environments, such as more objects, elements and obstacles regarding the indoor environments.

Difficulties of the indoor environments directly proportional with the offensive environments and emergency situations, the accuracy of the intelligent robot should be increased comparing to the matters mentioned previously. Indoor environments vary from simple to very complicated environments but even simple indoor environment are very hard to be swept by the intelligent robot or even an adult human in time manner without having simple knowledge about the indoor environment it is facing.

Since knowledge is the human brain soul, in order to solve problems and communicate with others, the needs of applying the human brain model on the intelligent robot increased by the demand of building an intelligent robot. The knowledge based system must be applied by covering one of its fields, i.e., receiving, thinking and storing knowledge.

INTELLIGENT ROBOT MAPPING

Thrun (2002) made a good survey discussing the techniques that have been used on intelligent robot mapping and many concepts have been identified. The psychologists concentrate on how human thinks and build maps and addressed it with the machines architecture (Vasudevan *et al.*, 2007) while others tried to find a new mapping concepts to perform human jobs far away from human thinking concepts (Kuipers, 2000; Remolina and Kuipers, 2004; Goel *et al.*, 1999). Recent study about that was to build a network of camera systems linked with a centralized unit that give direction for a Town Robot by Ishiguro *et al.* (1997). However, recent studies assure that to make intelligent robots perform human jobs we need to use the concept of human intelligence (Chandler, 2009).

Hall and Hall (1985) defines an intelligent robot as one that responds to changes to its environment through sensors connected to a controller. Much of the research in robotics has been concerned with vision and tactile sensing. For example, one of the most important considerations in using a robot in a workplace is human safety. A robot equipped with sensory devices that detects the presence of an obstacle or a human worker within its workspace. This is an important in the current implementation in most robotics work cells.

To reach the best performance for a robot to be intelligent, the human functional model has been studied which can be implemented as knowledge-based system (Owaied and Mahmoud, 2007).

This work based on the study of human system in term of functions, since the framework model is for a Robot doing a job as human in a specified and specific domain. Human in many situations needs to have basic information about the place or building that he can navigate inside it. Usually start building knowledge about the spatial objects either from a person who knows the place or using the information gained from reading the architectural blueprint map of a building. In this context the intelligent robot has been utilized to read the architectural blueprint map of a building that will be very important step in intelligent robot mapping and navigation. Therefore, the aim to develop intelligent robot which can read and gain knowledge from the architectural blueprints maps of indoor environments to build its own cognitive maps as dynamic knowledge to be added to the knowledge base.

COGNITIVE MAPPING

Many definitions have been written for cognitive mapping (Stephanie, 1998; Endo and Arkin, 2003). However, Tolman (1948) defined it as, inferred the existence of cognitive maps by recording the spatial behavior of a maze-running rat who took a short cut to the final destination by running across the top of a maze instead of following a route through it. Others studied cognitive mapping on dynamic environments like (Hahnel *et al.*, 2003).

Using architectural blueprint maps in building cognitive maps can facilitate even humans to better do their jobs and to better build their cognitive maps. Research on this field has discovered that even two years of work in the same building do not build a correct cognitive map like persons who used blueprints for the building from the first time. Survey made for school students by Tolman (1948) for outdoor environments but also shows problems in directions of areas without maps. The study shows that "When students at U.C. San Diego were asked to draw the direction between San Diego and Reno, they incorrectly indicated that San Diego was west of Reno. Indeed, it is surprising to learn that Reno is in fact west of San Diego. After all, California is on the western coast of the United States and Reno is far inland, in Nevada. A glance at a map reveals that the coast of California, far from running north-south, in fact cuts eastward as it cuts southward. Stevens and Coupe attributed their findings to hierarchical representations of space. People do not remember the absolute locations of cities. Instead, they remember the states cities are part of and the relative locations of the states.

Then they infer the relative locations of cities from the locations of their superset states. Where the research of Moeser (1988) discussed that: The series of studies reported in this article examined the cognitive mapping systems of student nurses who had worked in the hospital for various periods of time. After inspecting several different measures, it was concluded that the student nurses had failed to form survey-type cognitive maps of the building even after traversing it for two years. A control experiment was tested, using naive subjects who were first asked to memorize floor plans of the building. These naive subjects performed significantly better on objective measures of cognitive mapping than did the nurses with two years experience working at the hospital.

Another related research for Kuipers and Byun (1993) showed the huge amount of time needed by an intelligent robot in to explore a new building without human guide or supporting maps and move from room to room inside the building. Depending on maps during robot mapping field some studies have done using geographical maps during mapping (Motilal and Konolige, 2006) and others tried to put a theory to topological maps (Remolina and Kuipers, 2004; Schonberg *et al.*, 1995).

However, Schmidt *et al.* (2007) discussed that: When animals (including humans) first explore a new environment what they remember is fragmentary knowledge about the places visited. Yet, they have to use such fragmentary knowledge to find their way home. Humans naturally use more powerful heuristics while lower animals have shown to develop a variety of methods that tend to utilize two key pieces of information, namely distance and orientation information.

Knowledge based-system witnessed a huge research area and the study Frame Model for Intelligent Robot as Knowledge-Based System (Owaied, 2010), describes its architecture and it was as inspiration for using the knowledge-based system architecture in the proposed project. The project will be as enhancement to the cognitive mapping dealing with it as knowledge based system and extending its knowledge base with the knowledge gained from reading architectural blueprints of buildings which is the core of this work.

PROPOSED METHODOLOGY FOR DEVELOPMENT COGNITIVE MAP

The methodology of creating the cognitive map using blueprint map consists of many algorithms; the algorithms will be used by the inference engine of the knowledge based-system of the intelligent robot. These algorithms are used for gaining knowledge from the architectural blueprint maps, to build the cognitive map. These algorithms respectively are:

- Threshold algorithm
- Flood fill algorithm
- Start point finding
- End point finding
- Main path finding
- Targeted room finding
- Similarity algorithm
- Pattern matching

In the purpose of demonstration, testing and declaring the algorithms, Architectural blueprint map for offices building at Regus Whitefields Centre, Bangalore, India is used.

Figure 1 presents the labeling architectural blueprint process for the center floor of the building mentioned above building. Some algorithms were tested using MATLAB R2009b to prove where others are straight forward proved in pseudo code outputs.

Threshold algorithm: Architectural blueprint maps most likely have a lot of noise, especially if it was scanned from a hard copy or old architectural maps. Thus gray scale colors required using threshold to certain colors in order to decrease and utilize the processing time.

Therefore, the preprocessing dedicated threshold algorithm is necessary to simplify working on the architectural map. The enhanced threshold algorithm has been developed in this study, to match color ranges (black white, red, blue) expected by the robot inference engine.

As shown in Fig. 2, Starting from the first pixel in the Architectural blueprint map (0,0), threshold algorithm checks if the pixels color sets in the range of the white color, if so it converts the pixel to perfect white (FFFFFF). But if not it checks if the pixels color sets in the range of the black, it converts the pixel to perfect black (000000), otherwise it checks if the pixels color sets in the range of the red and converts the pixel to perfect red (FF0000). Also it checks if the pixels color
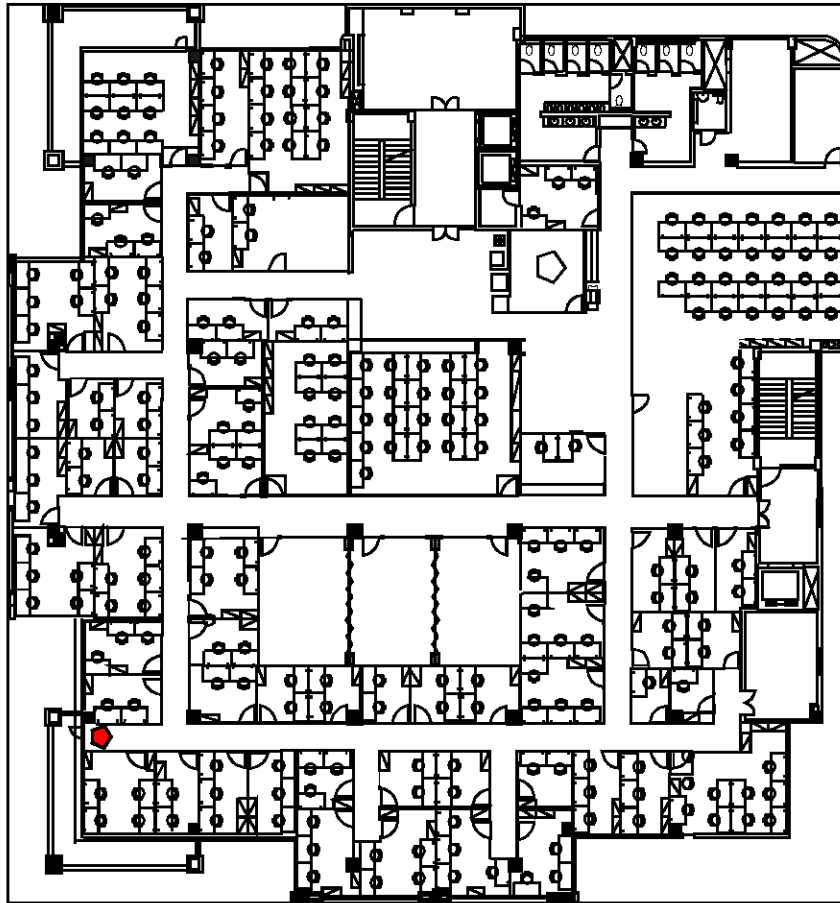
Fig. 1: Labeled architectural blueprint, center floor at regus whitefields centre, India

sets in the range of the blue, converts the pixel to perfect blue (0000FF). But if none of the above ranges matches the color then it converts the color of the pixel to perfect black (000000).

The mentioned color ranges sets on the knowledge-base of the robot based on RGB starting from perfect black (000000), ending with perfect white (FFFFFF). Previous step loops for each pixel in the Architectural blueprint map, narrowing the colors of it to the four colors mentioned above, Fig. 3 show the output after applying the code on the map using MATLAB R2009b.

Flood fill algorithm: Flood Fill, is an algorithm that is commonly used by pictures editors and the first picture editor start using it was Adobe Corporation, the pseudo code mentioned here is code for 8-Way Recursive Method from Vandevenne (2004), it is mostly used in selecting a set of connected pixels with the same color within a bordered figure.

The 8-Way Recursive Method works like this: at the start position, you plant a seed. Each seed gives the pixel at its position the new color and then plants a new seed at its 8 neighbors. Each of these new seeds will draw a pixel again and plant even more seeds but only if fulfills the following conditions:

```
// This is an enhanced thresholding algorithm that
// convert image from all the colors to only
// Four colors, black (000000), white (ffffff), red (ff0000), blue (0000ff)
```



```
for each pixel P in the image
    if P = one of the white ranges then
        color (P) = ffffff
    end if
    if P = one of the black ranges then
        color (P) = 000000
    end if
    if P = one of the red ranges then
        color (P) = ff0000
    end if
    if P = one of the blue ranges then
        color (P) = 0000ff
    end if
    if P = any other ranges not from above
        color (P) = 000000
    end if
end for
```
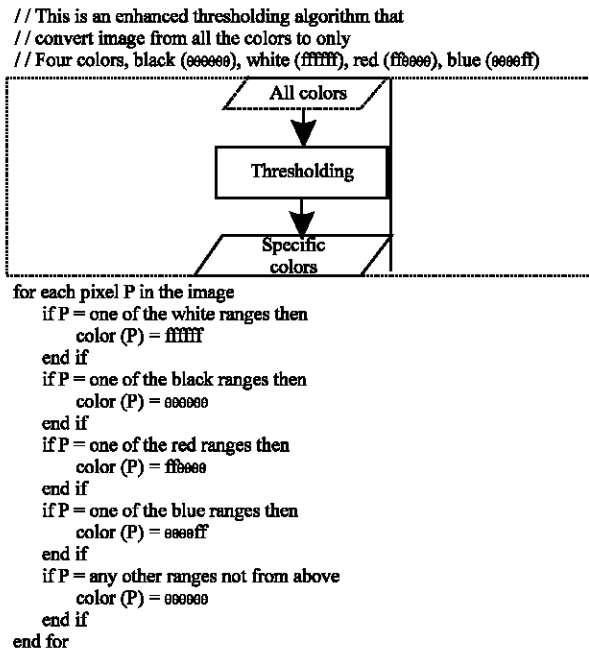
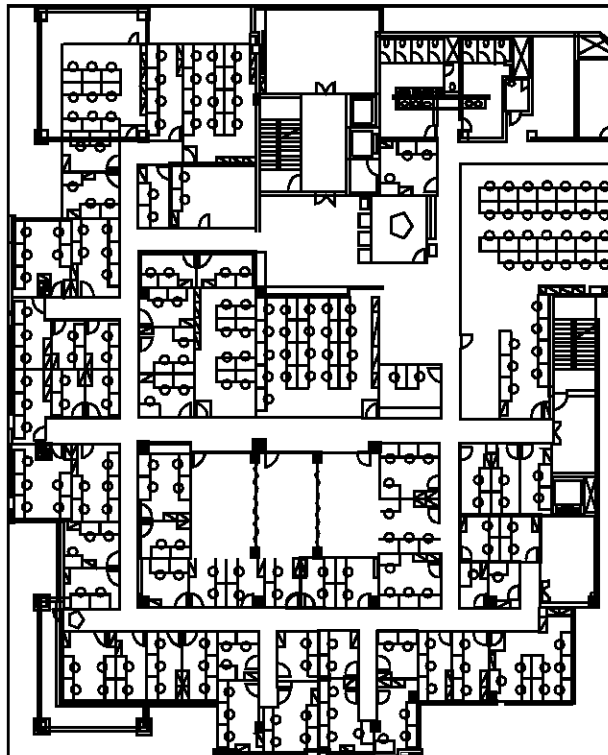Fig. 2: Threshold pseudo code

Fig. 3: Blueprint after threshold using MATLAB R2009b

- **The pixel is inside the screen:** Edges of the screen also count as edges of the area to be filled
- **The pixel has old color:** If it hasn't, it's either a border of the area were filling, or a pixel that has already been given the new color

- **The pixel doesn't have new color:** This condition is only needed if old color is the same as old color; otherwise it'll keep running forever since newly drawn pixels will again have old color and thus seeds will be planted again and again forever

The function will keep calling itself more and more until eventually all pixels are filled. If the new color is not specified then it is the same as old color.

The purpose of this algorithm is to put the selection in a non symmetry matrix so that each pixel in the mask represents a pixel in the figure with the same coordinates. The non symmetry matrix is called a mask.

In the algorithm, masks are used as tagging method side by side as a sub image within the architectural blueprint map while working with the sub image will not reflect the pixel in the main map. Figure 4 shows the algorithm Pseudo code.

Start point finding: As mentioned on the scenario, intelligent robot owners should specify start and end point, as shown in Fig. 5, to the intelligent robot; start point should be a red label that represents the intelligent robot start position in the targeted building.

To recognize start point, an algorithm was made (Fig. 6) to find the only red pixels set in the architectural blue print map. Thus starting from the first pixel on the map, if the pixel is red (FF0000) the algorithm put it on startPointMask which is a mask. The mask represents the set of all red pixels in the map (the red label). After that the red pixel in the map converted to white pixel (FFFFFF), this step decreases later calculations.

End point finding: This algorithm is pretty similar to the previous one, to recognize the end point, the algorithm shown in Fig. 7 to find the only blue pixels set in the architectural blue print map. Thus starting from the first pixel on the map if the pixel is blue (0000FF) then put it on endPointMask which is a mask that represent the set of all blue pixels in the map (the blue label). After that the blue pixel in the map converted to white pixel (FFFFFF), this step decreases later calculations.

```
//Recrusive 8-way floodfill, crashes if recrusion stack is full
function floodfill8 (x, y, int new color, int old color)
        If x>= 0 && x<w && y>=0 &&<h && screen Buffer [x] [y] = = old color
                                    && screen Buffer [x] [y] != new color then
                screen Buffer [x] [y] = new color//set color before starting recrusion!
                floodfill8 (x+1, y,     new color, old color)
                floodfill8 (x-1, y,     new color, old color)
                floodfill8 (x,          y+1,       new color, old color)
                floodfill8 (x,          y-1,       new color, old color)
                floodfill8 (x+1, y+1,  new color, old color)
                floodfill8 (x-1, y-1,  new color, old color)
                floodfill8 (x-1, y+1,  new color, old color)
                floodfill8 (x+1, y-1,  new color, old color)
        end if
        return screen Buffer
end function
```

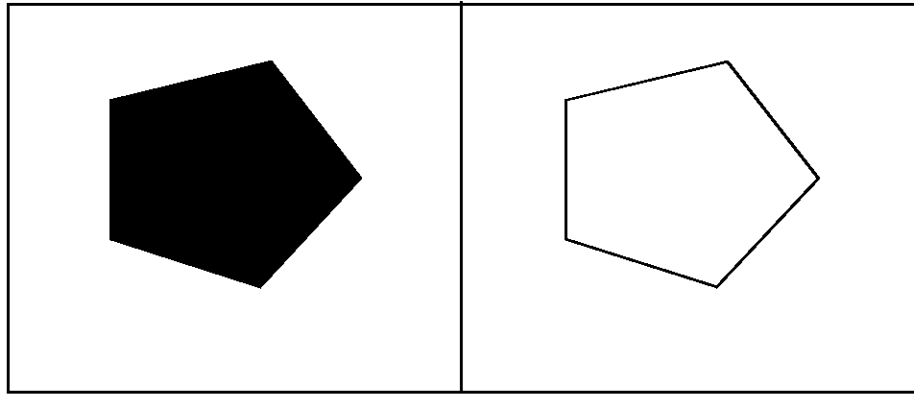Fig. 4: Flood fill algorithm pseudo code
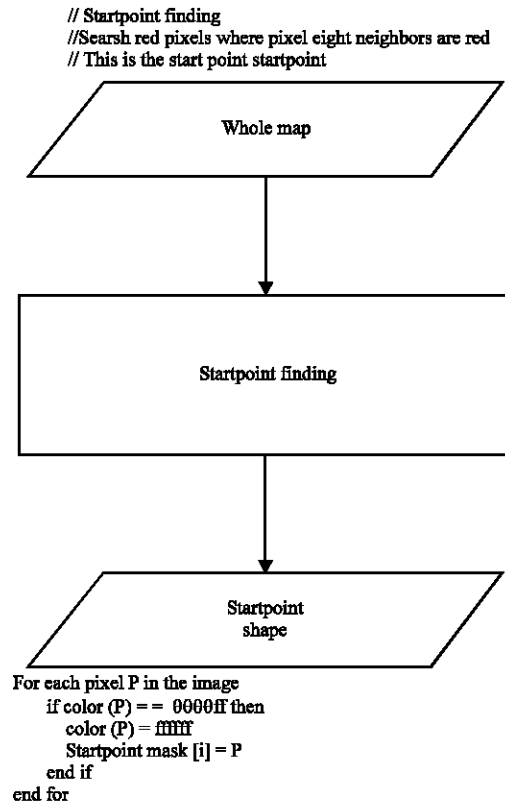
Fig. 5: Start (right) and end (left) point labels

Fig. 6: Start point finding algorithm pseudo code

Main path finding: In the scenario start point should be in the main path of the targeted room. While the start point relays on the main path, then all of its pixels are part from the main path pixels.

As shown in Fig. 8, to find the main path, the algorithm needs to get any point from the mask startPointMask where the positions of all start point pixels are stored. Therefore, sent along with the main path color (white), to the floodfill8 function which will capture the entire main path

```
// Endpoint finding
```

Whole map

end point finding

end point
shape

```
For each pixel P in the image
    if color (P) == 0000ff then
        color (P) = ffffff
        endpoint mask [i] = P
    end if
end for
```
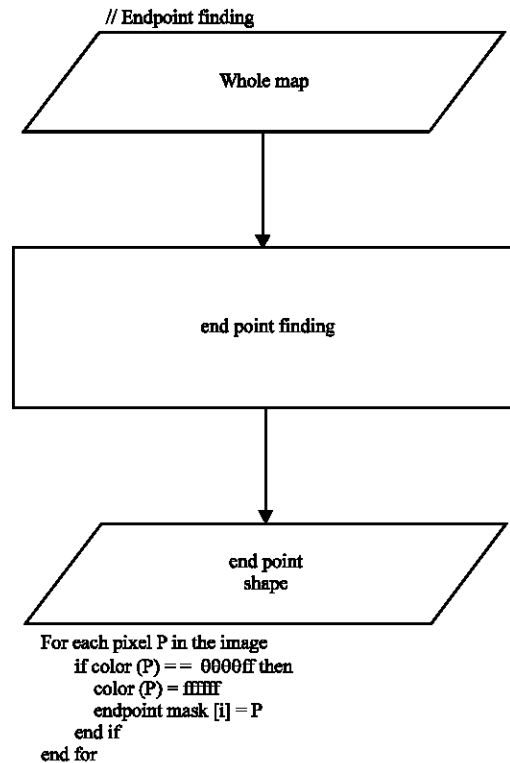
Fig. 7: Endpoint algorithm pseudo code

pixels which are white pixels disclosed inside the walls, doors windows of this main path. The function returns a mask that is captured by the main path finding algorithm as a mask called mainPathMask. Figure 9 views a declaring shape for the mask.

Targeted room finding: Scenario has specified that end point should be in the targeted room which is the room that is separated from the main path by only one door. While the end point relays on the targeted room, then all of its pixels are part from the targeted room white pixels.

To find the targeted room, the algorithm in Fig. 10 needs to get any point from the mask endPointMask where the positions of all end point pixels are stored; this position is then sent along with the targeted room background color (white), to the floodFill8 function which will capture all the targeted room background pixels which are white pixels disclosed inside the walls, doors and windows of this targeted room. The function returns a mask that is captured by the targeted room finding algorithm as a mask called targetedRoomMask. Figure 11 views a declaring shape for the mask.

Similarity algorithm: As shown in Fig. 12, for each white pixel the algorithm finds the white shape that contains it, each shape gets its own mask, then sent to a function that calculate the size of the mask, if the returned size is bigger or smaller than specific size, then the mask will be ignored, because as experienced, doors size set between that range. If returned size sets within that range then it might be a door or other architectural symbol, so to make sure that it is a door before tagging it, the algorithm contains pattern matching.
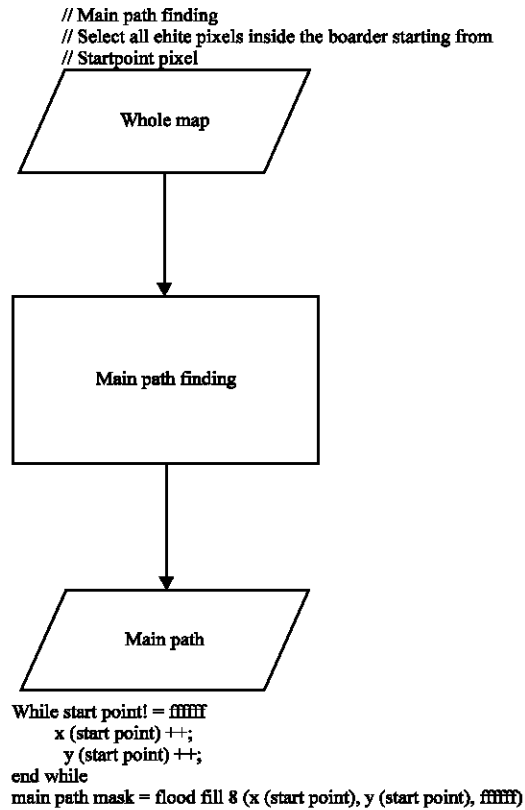
```
// Main path finding
// Select all ehite pixels inside the boarder starting from
// Startpoint pixel
```

Whole map

Main path finding

Main path

```
While start point! = ffffff
        x (start point) ++;
        y (start point) ++;
end while
main path mask = flood fill 8 (x (start point), y (start point), ffffff)
```
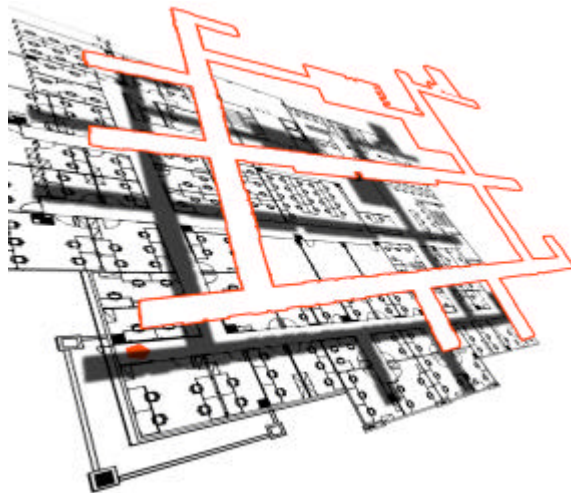
Fig. 8: Main path finding algorithm pseudo code

Fig. 9: Main path extracted from the blueprint after applying the algorithm

PATTERN MATCHING

The main idea of pattern matching is checking for specific patterns in images, if patterns almost rely on the checked image, then the image is similar to the patterned image. In this algorithm,

// Targeted room finding

Whole map

Targeted room finding

Main path

```
While end point ! = ffffff
        x (end point) ++;
        y (end point) ++;
end while
targeted room mask = flood fill 8 (x (end point), y (end point), ffffff)
```
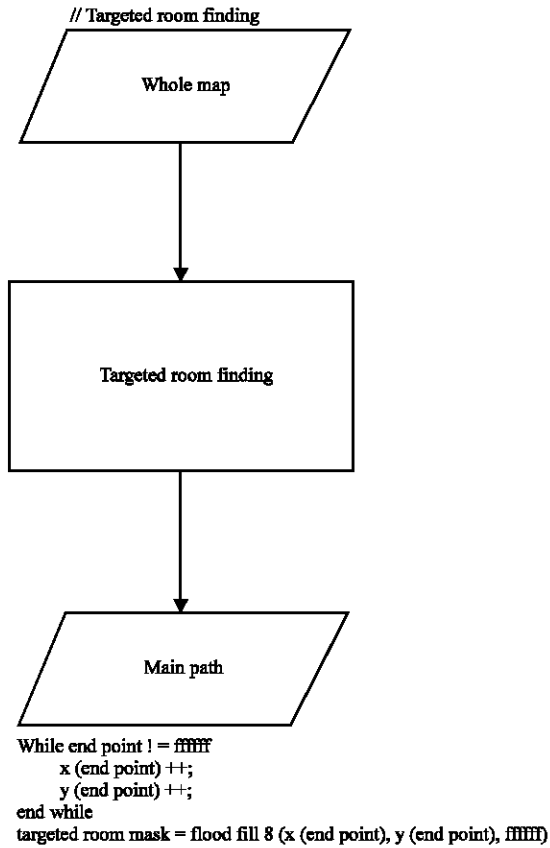
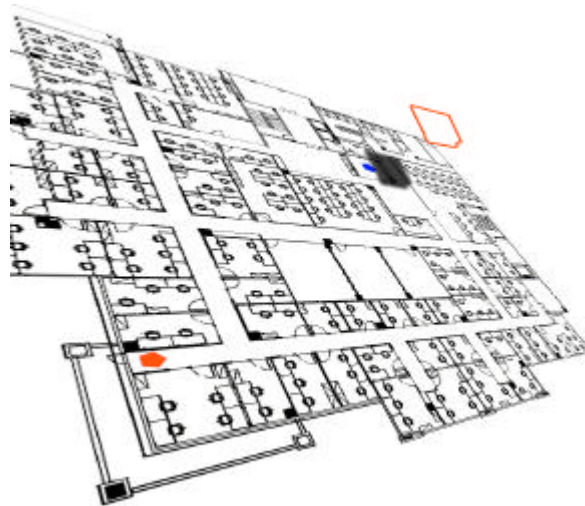Fig. 10: Targeted room finding algorithm pseudo code

Fig. 11: Targeted door extracted from the blueprint after applying the algorithm

pattern matching is enhanced to do a specific job, utilizing processing time. Matching the suspected doors' background with specific pixels (white, nonwhite) for right to left, left to right, bottom to up and top to bottom doors.

```
// Similarity
```



```
while white shape not scanned
if doorSize > = flood fill8 (x(pixel in white), y (pixel in white), ffffff > = door size then
// horizontal right to left checking
if P1, P2, P3, P4 ... Pn = = ffffff  and P1, P2, P3, ... Pn = = 000000 then
for all pixels in shape
color (P) = 00ff00
end for
end if
//hoizontal left to right checking
if P1, P2, P3, P4 ... Pn = = ffffff and P1, P2, P3, P4 ...  Pn = =000000 then
for all pixels in shape
color (P) = 00ff00
end for
end if
//horizontal top to bottom checing
if P1, P2, P3, P4 ... Pn == ffffff and P1, P2, P3, P4 ... Pn = 000000 then
for all pixels in shape
color (P)  = 00ff00
end for
end if
// horizontal bottom to top checking
if P1, P2, P3, P4 ... Pn == ffffff  and P1, P2, P3, P4 ... P n == 000000 then
for all pixels in shape
color (P) = 00ff00
end for
end if
whiteShape scanned
end if
end if
```
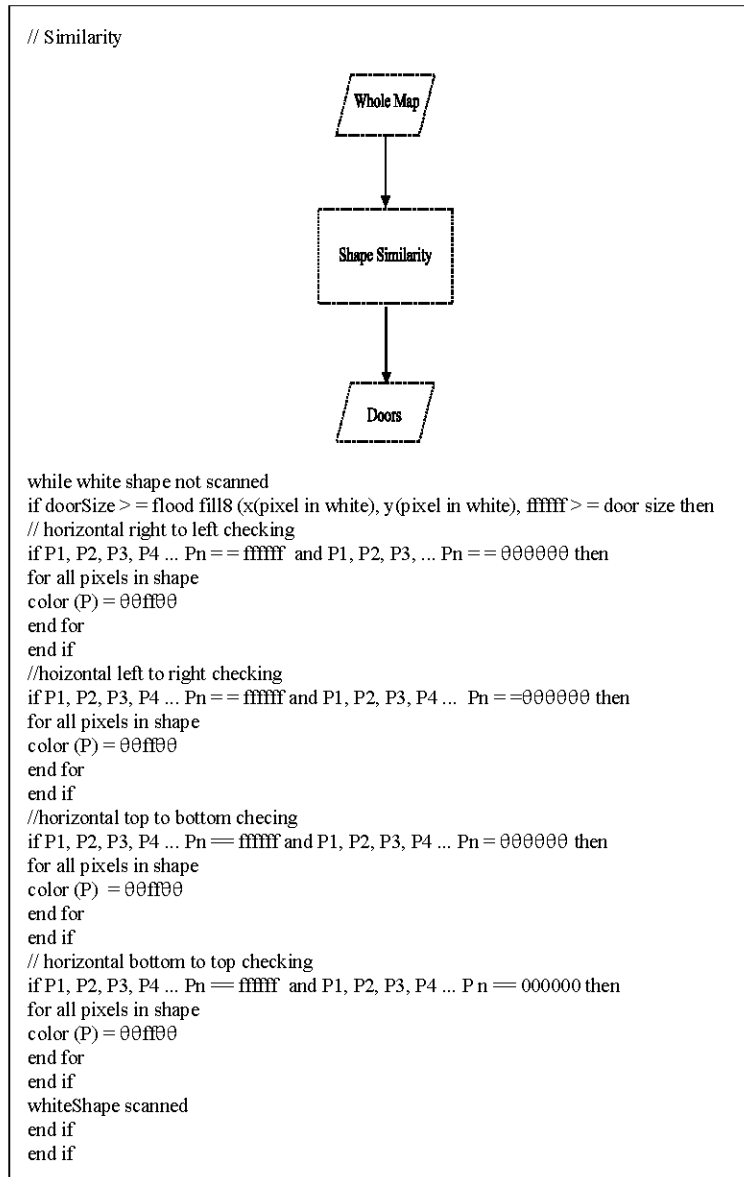
Fig. 12: Similarity algorithm pseudo code

The algorithm takes the current suspected door's mask background and compares it with the mentioned patterns (right to left, left to right ...). Therefore, if one of the patterns matched the mask then the mask is a door, the patterns are put to satisfy the heel and the semi heel background door shapes. For a detailed description for the above, to do the comparison, each pattern contains a matrix for a specific white and nonwhite (no pixel). Let's assume the following; left to right pattern, has (Ps, Pg, Pr ... Pn) white pixels and (Pw, Pd, Pf ... Pn) non white pixels, if the two conditions are satisfied where the mentioned pixels are in the door background mask, then the shape is the background of a door, as seen in Fig. 13, the pseudo code of the algorithm.

THE DEVELOPED COGNITIVE MAP

Based on the knowledge which were gained from the architectural blueprint maps, the intelligent robot start building cognitive map for that building (the building that the intelligent

```
        //doors on main path and targeted room
while green shape not scanned
            for each border pixel in the green shape
                if x (P) +4 = = MainPath then
                            door on Main = green shape
                    end if
                    If x (P) +4 = = TargetedRoom then
                            door on Main = green shape
                end if
            end for
    end while
    for all door on Main and door on Room
        for each pixel in the shape
                if P eight neighbors ! = θθ ff θθ then
                    if door on Room then
                            target door  =  P
                    else
                            doors [j] = P
                    end if
                    end the current loop
                else if one of P neighbors = = θθ ff θθ then
                        boarder Pixel [I] = P
                end if
            end for
                remove boarder Pixel array
                re-loop
    end for
```

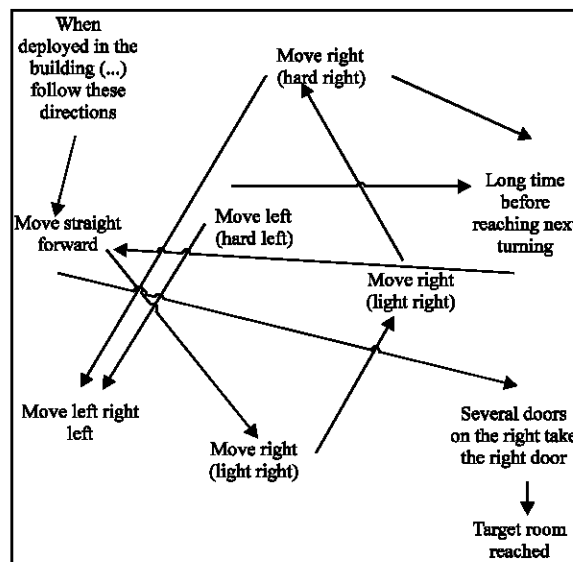Fig. 13: Pseudo code of the pattern matching algorithm

Fig. 14: Directions cognitive map

robot was supplied with its map). The cognitive map is built as a graph where each node present certain knowledge. Figure 14 shows a simple example of the directions extracted from the architecture blueprint map presented in the cognitive map.

The cognitive shown in the Fig. 14 is simplified in order to declare the idea of cognitive mapping. In real time it is more sophisticated map, containing relations with the definition of concepts used in each node (left, right, hard, light, normal, turning …). It is also connected with questions related to "where am I", "what am I doing" … and so on.
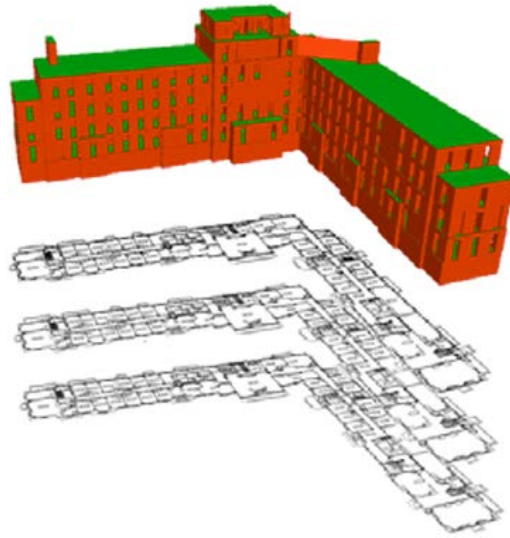
Fig. 15: Multi floored building architectural map (Alex *et al.*, 2009)

After building such map when the real time came after being deployed, the intelligent robot, start updating the cognitive map based on the realty. Adding more nodes based on what it sees and recognizes from objects and avoided obstacles.

CONCLUSION

This study focused on the development of algorithms for building cognitive map for intelligent robot based on architectural blueprint maps, showing how the cognitive maps could be generated from architectural maps. The main features of the developed algorithms that helping intelligent robots work without the need of direct human interaction. These algorithms take the architectural blueprint maps and analyze it based on the importance of the information needed to be gained from the map. The mentioned algorithms were created to make the robot understand the architectural blueprint maps and convert it to knowledge as a cognitive map inside the knowledge based system of the intelligent robot. As a future work, it is highly recommended to start thinking about more algorithms that extends the job of the algorithms created or developed in this thesis to cover multi floored buildings dealing with each floor as a layer connected with the other layers (floors) as by connecting points (leaders, stairs, Elevators ... etc.) (Fig. 15).

On the other hand algorithms need to be developed to be more dynamic dealing with more sophisticated buildings with more sophisticated job routers (separated paths, targeted room inside another room).

REFERENCES

Alex, L., A. Stere, K.N. Kutulakos, D.J. Fleet, S.J. Dickinson and K. Siddiqi, 2009. TurboPixels: Fast superpixels using geometric flows. IEEE Trans. Pattern Anal. Machine Intel., 31: 2290-2297.

Chandler, D.L., 2009. Rethinking artificial intelligence. Broad-Based MIT Project, Massachusetts Institute of Technology University, USA. http://web.mit.edu/newsoffice/2009/ai-overview-1207.html.

Endo, Y. and R.C. Arkin, 2003. Anticipatory robot navigation by simultaneously localizing and building a cognitive map. Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems, Oct. 27-31, Las Vegas, Nevada, USA., pp: 460-466.

Goel, P., S.I. Roumeliotis and G.S. Sukhatme, 1999. Robot localization using relative and absolute position estimates. Proceedings of IEEE/RSJ International Conference on Robotics and Intelligent Systems (RIS'99), Kyongju, Korea, pp: 1-7.

Hahnel, D., R. Triebel, W. Burgard and S. Thrun, 2003. Map building with mobile robots in dynamic environments. Proc. Int. Conf. Robotics Auto., 2: 1557-1563.

Hall, E.L. and B. C. Hall, 1985. Robotics: A User-Friendly Introduction. Saunders College Publishing, Holt, Rinehart and Wilson, Orlando, FL.

Ishiguro, H., H. Ishiguro and K. Kimoto, 1997. Town Robot-Toward Social Interaction Technologies of Robot Systems. Kyoto University, Japan.

Kuipers, B. and Y.T. Byun, 1993. A Robot Exploration and Mapping Strategy Based on a Semantic Hierarchy of Spatial Representations: Toward Learning Robots. MIT Press, Cambridge, MA., USA.

Kuipers, B., 2000. The spatial semantic hierarchy. Artificial Intell., 119: 191-233.

Moeser, S.D., 1988. Cognitive mapping in a complex building. Environ. Behav., 20: 21-49.

Motilal, A. and K. Konolige, 2006. Real-Time Localization in Outdoor Environments using Stereo Vision and Inexpensive GPS. SRI International Ravenswood Ave Menlo Park, California.

Owaied, H.H. and M.A.A. Mahmoud, 2007. Functional model of human system as knowledge base system. Proceedings of International Conference on Information and Knowledge Engineering, June 25-28, CSREA Press, Las Vegas, Nevada, USA., pp: 158-161.

Owaied, H.H., 2010. Framework model for intelligent robot as knowledge-based system. Special Issue of Ubiquitous Computing Security Systems. http://ubicc.org/files/pdf/Dr.hussein-353_353.pdf.

Remolina, E. and B. Kuipers, 2004. Towards a general theory of topological maps. Artificial Intell., 152: 47-104.

Schmidt, J., C.K. Wong and W.K. Yeap, 2007. Spatial information extraction for cognitive mapping with a mobile robot. Proceedings of the 8th International Conference on Spatial Information Theory, Sept. 19-23, Melbourne, Australia, pp: 186-202.

Schonberg, T., M. Ojala, J. Suomela, A. Torpo and A. Halme, 1995. Positioning an autonomous off-road vehicle, by using fused DGPS and internal navigation. Proceedings of the 2nd IFAC Conference on Intelligent Autonomous Vehicles, June 1995, Helsinki, pp: 226-231.

Stephanie, R.A., 1998. Community-level socioeconomic status effects on adult health. J. Health Social Behav., 39: 18-37.

Thrun, S., 2002. Robotic mapping: A survey. Exploring Artificial Intelligence in the New Millenium, Morgan Kaufmann. http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.9.7791&rep=rep1&type=pdf.

Tolman, E.C., 1948. Cognitive maps in rats and men. Psychol. Rev., 55: 189-208.

Vandevenne, L., 2004. Lode's computer graphics tutorial. http://www.student.kuleuven.be/~m0216922/CG/index.html.

Vasudevan, S., S. Gachter, V. Nguyen and R. Siegwart, 2007. Cognitive maps for mobile robots-an object based approach. Robotics Autonomous Syst., 55: 359-371.

Yeap, W.K. and M.E. Jefferies, 2000. On early cognitive mapping. Spatial Cognition Computation, 2: 85-116.