Optimal Path Routing

Loay.K.Abood and Ahmed.A.Shkara
Department of Computer Science, University of Baghdad, Baghdad, Iraq
www.ahmedsk_2007@yahoo.com

Abstract

Analyzing Google map to find optimal path routing is an important and advanced strategy in geographic information system (GIS), it provides strong decision support for the users in service area.

Optimal path routing system built to allows the users, businesses, and militaries to find the optimal path (shortest and fastest path) between any two points (locations) on a road networks (edges) of Google map image.

إيجاد المسار الأمثل

لؤي كاظم عبود و أحمد عبد الزهرة شكارة
قسم علوم الحاسبات، كلية العلوم، جامعة بغداد، بغداد، العراق

الخلاصة

تحليل خرائط قوقل لإيجاد المسار الأمثل تعتبر ستراتيجية متقدمة و مهمة في مجال نظم المعلومات الجغرافية (GIS) , حيث تساعد المستخدم في مجال الخدمات على إتخاذ قراراته.

نظام أيجاد المسار الامثل بني ليساعد المستخدمين ورجال الأعمال والعسكريين على إيجاد المسار الأمثل (أقصر وأسرع طريق) بين أي نقطتين يتم تحديدهما على شبكة الطرق الظاهرة على صور خرائط قوقل .

1. Introduction

The proposed system built to work dynamically by; capturing the raster Google map images directly from Google maps explorer, and by using image segmentation the road networks will be extracted from the image, and by using some morphological operators the road networks will be binaries and skeletonize, and by using conversion raster to vector method the nodes (with their coordinates and indexes) and edges (with their weights) were extracted (the vector map representation) and stored in a dynamic GIS database. After that, the system helps the user to locate any two points on the road networks as initial and distant nodes. Then, the optimal path will be determined by applying Dijekestra's algorithm on vector map (dynamic GIS database), and the optimal path "road" will be drawn in raster map image by red color. In addition to that the system has the ability to modify the optimal criteria on selection by editing the map itself or any wanted property or configuration.

In the following algorithm we will be shown how to find optimal path routing on raster Google maps. First, an overview of the algorithm is given. Then, the partial steps are described in detail and results on different examples are shown. Finally, it will be discussed how the algorithm can be extended to support various kinds of applications.

2. Algorithm

Finding the optimal path between any two points on a roads map (Google map) can be categorized into three stages as in flow:

Stage 1: Image Map Acquisition

1.1- Browsing Google map to get good view for roads map

1.2- Capturing raster Google map image.

1.3- Editing image to updating road network

Stage 2:Pre-Processing Google Map Image

2.1- Automatic roads network extraction by image segmentation.

2.2-Binarised input roads network

2.3-Applay Morphological operation for refining extracted roads network.

2.4- Skeleton Roads network.

2.5- Smoothing and enhancing skeleton road map

2.6- Convert Skeleton road map from raster to vector.

Stage 3: Optimal Route Finding

3.1- Locate two points as source and destination.

3.2- Using Dijkstra's algorithm to determine the optimal path routing.

In the first stage the image acquisition, where explorer software is used to browse, capturing, and editing raster Google map. Editing process covered labeling, modifying roads image. In the 2nd stage (preprocessing road image) the road image converted to road map (raster to vector). Vector image contain roads map as nodes and edges through multiple processing process includes morphological operators. The final third stage, optimal rout finding is determined according to suitable algorithm (Dijkstra's method).

3. Stage 1: Image Map Acquisition

To acquire a road image of specific city around the Earth, Google map software was used. But the initial problem is that this software is not supported with capturing or saving road images, for this reason explorer or software arrangement was proposed to full fill this task. An additional facility like editing is also included in this program. Figs (2) represent a capture road image to the City of Baghdad, Iraq – as an example.

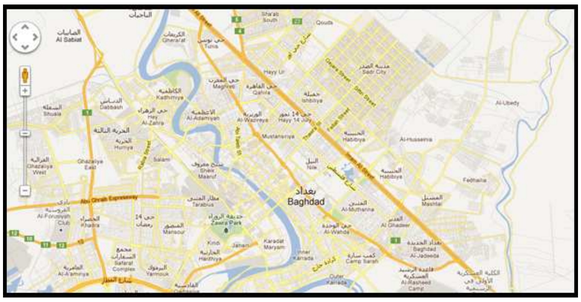

Figure 2- Raster Google Map Of The City Of Baghdad, Iraq.

4. Stage 2: Pre-Processing Google Map Image

The main goal of this stage is to convert road image to road image vector. This process is accomplished through multiple processes as fallow:

4.1 Automatic Roads Network Extraction By Image Segmentation.

In this work, roads network is extracted based on Image filtering using RGB pixel values, Where were gradient colors of yellow and green, and the rest of the colors were converted to white (see Fig .3).

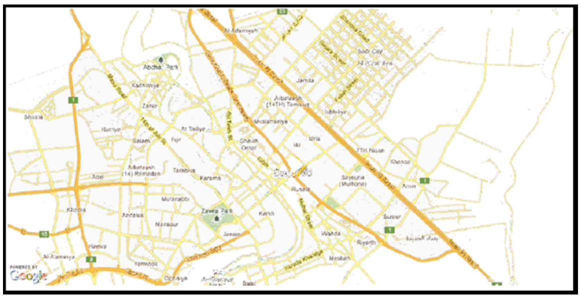

Figure 3- Roads network**.**

4.2 Binaries roads network image

The next step is to convert the extracted image into two class foreground roads and background (the rest of the image) through binarization (threshold) see Fig (4).

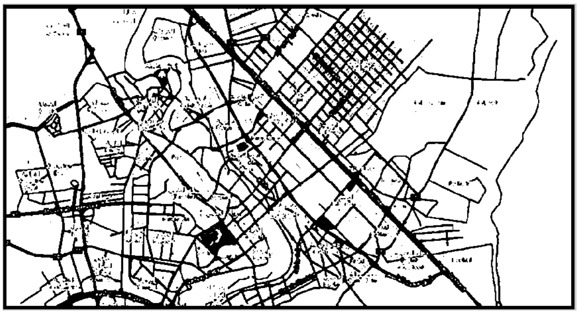

Figure 4- Binaries Roads Network Image.

4.3 Apply Morphological Operation For Refining Extracted Roads Network.

Morphological operation after that will be used to refine the binary image by applying Open operator to remove any object fewer than 10 pixels (see Fig .5). To get more pure and adequate image we have repair the gaps and disconnected generated from the previous operators by applying Dilation operator (see Fig. 6) and, Bridge operator (see Fig. 7) respectively. Finally Closing operator is used to remove small objects and holes generated from removing labels and signs [1], (see Fig 8).

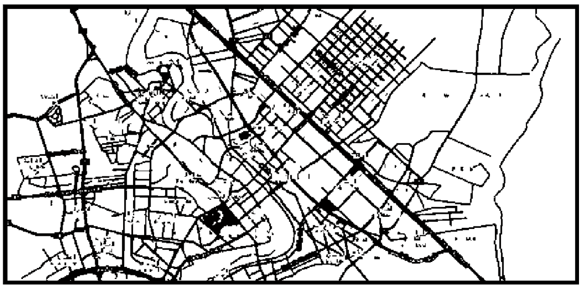

Figure 5- Morphologically Open Operation

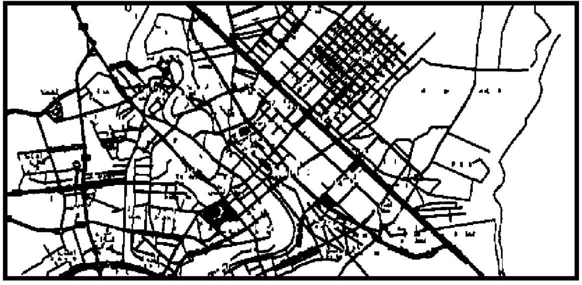

Figure 6- Morphological Dilate.

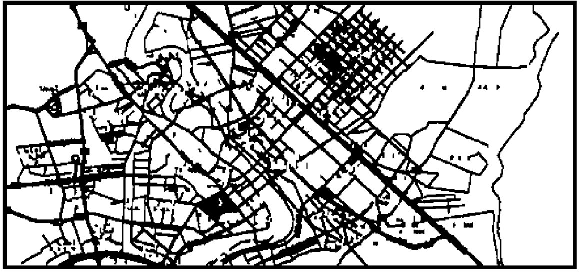

Figure 7- Morphological Bridges Operation.

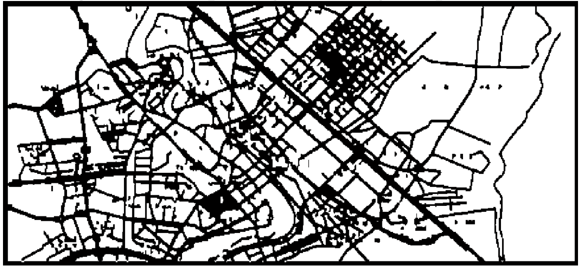

Figure 8- Morphological Closing Operation

4.4 Morphological skeleton operation

After all these operators, Skeletonization process is begun to extract road skeleton, this process can be performed in many different algorithm.(see Fig. 9)

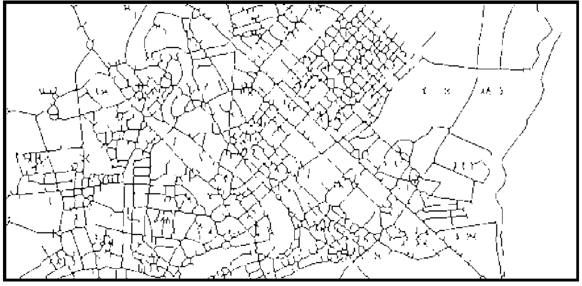

Figure 9- Morphological skeleton operation

4.5 Smoothing And Enhancing Skeleton Road Map

Apply Skeletonization ,generate unwanted artifacts that affect on the result of the next stage, therefore morphological operators are needed by applying logical operator between the skeleton images and binary images (using AND operation) to improve the image ,followed by open operator to get clearness skeleton image (see Fig . 10)

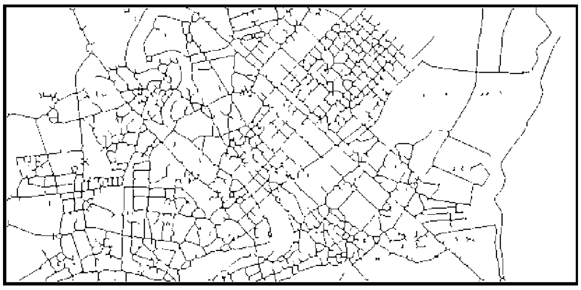

Figure 10- Smoothing And Enhancing Skeleton Road Map.

4.6 Convert Skeleton Road Map From Raster To Vector:

The purpose of conversion of raster images into vector representation is to get mathematical described for paths (roads) which is used later to finding shortest path. This process is accomplished through two processes as fallow:

4.6.1 Extracted The Nodes

Extracted the nodes from skeleton road map begins by scan and analyzes image line by line from top to button as in the following algorithm:

1- start

2- Create dynamic two dimension array (named – Nodes).

3- Loop: scan image from top to button.

4- For each bit (pixel) in the bitmap that is turned on, apply 8-connected mask to calculate the number (sum) of neighborhood pixels. With the same mask, test if the pixel is in the middle of the line or not, and we can do that by multiply the elements on each diameters in the mask with each other, and find the summation for the multiplication results, after that if the result is one then the pixel is in the middle of the line else it is not.

5- If (the number of neighborhood pixels does not equal two) and (the pixel not in the middle of line) then it is Node, therefore save it is coordinate(x,y) and number of neighborhood pixels in Nodes array .

6- End loop

7- End

8- The previous algorithm yielding complete dynamic nodes array Such as in table (1) :

Table 1- Nodes Array Representation

index	Coordinate X-excess	Coordinate Y-excess	No- of neighborhood pixels
1	X1	Y1	8
2	X2	Y2	1
. .	. .	. .	. .
n	Xn	Yn	5

4.6.2 Representation Of Vector Graph:

The representation of the vector graph in the computer can be done in several ways. Usually the most convenient approach is the one used here, *which are acquired in the Geographic Data Files (GDF) format [2].* We will represent the graph as an array. In the previous step we collected the nodes and stored them in an array of nodes. Note that we don't find the relationship between these nodes. In other words, every two nodes in skeleton image represent the beginning and end of a single straight line and we store this relationship in dynamic array that contains all lines in our graph. To do that, we have developed the following algorithm:

```
1-  start
2-  Loop1: until end of nodes array.
3-  Read coordinate(x,y) of node .
4-  Read the number of neighborhood pixels.
5-  Use coordinate (x,y ) for access directly to pixel in skeleton image.
6-  Loop2: until the number of neighborhood pixels equal to zero.
7-  Apply 8-connected mask on pixel and track the first neighborhood pixel that found.
8-  Apply 8-connected mask on neighborhood pixel
9-  If neighborhood pixel is not node then delete it ,and increase width counter by one ,and keep tracking (as a Snake )
10- If neighborhood pixel is node then teak it coordinate and search for it on ( nodes array ) and if you finding it save it index in dynamic vector ( named Yn), and save the index of organ node in another dynamic vector (named Xn) , and save the number of width variable in third dynamic vector (named Wn) .
11- End loop 2.
12- Decrease number of neighborhood pixels by one.
13- End loop1.
14- End
```

<u>Note</u>

We Separated between Collecting of nodes and Representation of vector graph, because the first operation need to Sequential processing, and the other to direct processing.

If we use the three previous dynamic arrays to plot a vector Graph we will get Fig.13.

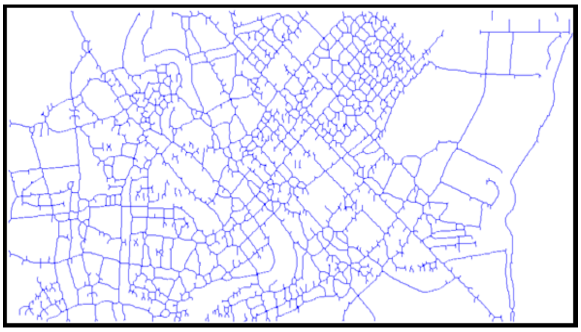

Figure 13-Vector Road Network

5. Stage 3: Optimal Path Finding

To calculate the optimal path routing we use vector skeleton map information as in flow:

5.1- Locate Two Points As Source And Destination.

In traditional applications there are several ways let the users enter the start and end pointes of the path. As an example, the application can asks the user to enter the source and destination cities name, or streets name, or any two specific locations or data. Her we use a simple method to enter node A and B; this method depend on feedback, were the system let the user enter the start and end pointes by click on specific vertices that embedded in Skelton road map and close those vertices with streets roads vertices that's in raster Google map. This method work as in the following:

1- Copying nodes image in to 2D array have the same size with image, and load it to memory.

2- Load and display original Google map image, and use full crosshair cursor to scan embedded vertices.

3- Use (moving cursor event) to determine X, Y coordinate of current cursor position on displayed Google map image.

4- Compeer X, Y cursor position with X, Y index of array, and if there is one (1) in array location, then display hint (OK to click) on the title of image, otherwise, do not do anything.

5- After the hint appears, the user can click to mark the specific vertex using right mouse button click event.

6- If the user wants to clear map he can the click left mouse button event.
7- If the user wants to save the result or printing it or doing any option he can do that by selecting the suitable option from option bar.
8- If the user wants to consul all the operations and return to (optimal browser) he can push on (return to Google) button.
Not that, in this way we prevent the user from clicking on anywhere except on the streets map (see Fig.14).

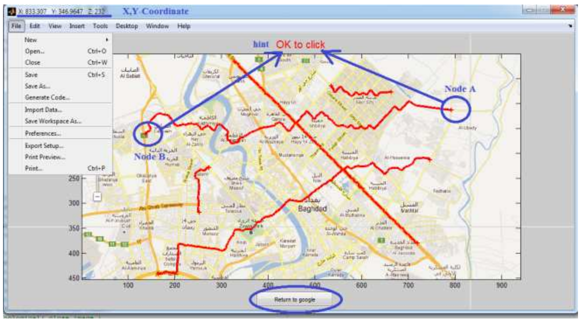

Figure 14- Calculations The Optimal Path Between Start And End Nodes (In Red Color)

5.2 Using Dijkstra's Algorithm To Determine The Optimal Path Route.

The optimal path is calculated with an Dijkstra's algorithm [*3*]. The optimal shortest path is that path that connects the start and end point with the skeleton and has the shortest length and suitable cost. Fig. 14 shows an example of a shortest path in red color.

6. Results & Discussions

Fig. 14 shows how we can find any optimal path between any two points on the roads network by using the edges of the skeleton, and this path is represented with a thick red line.

Conclusions

From this research, many points are noticed and concluded. The following are the most important ones:
1-The system has been tested on a large number of dataset, and obtained very satisfying results, but it doesn't always find exactly the shortest path, because the shortest path calculation depends essentially on the extracted skeleton. Therefore in some special situations, the system finds the second or the third best path ,but always a path which has a length that does not differ very much from the shortest path.
2-The interactive editing and updating on saved map image is a very important factor. For example the Google map may contain old or new or closed roads. This can lead to unnatural results; therefore updating will eliminate this problem. However, the interactive editing can be done very effectively, because an operator has only to use a paint application and add or remove or close all unnecessary roads from the map.

9. References

[1] Gonzalez, C.Rafael , and Woods E.Richard **2003** " Digital Image Processing", 3rd ed., Addison-Wesley,.
[2] Winston, H. J, **2000** "Artificial Intelligence", Addison-Wesley, London,.
[3] Dijkstra E.W. **1959**,"A Note on Two Problems in Connexion with Graphs", Numerische Mathematik 1,269-271 ,.