Intelligent GIS: Automatic generation of qualitative spatial information

Jimmy A. Lee[1] and Jane Brennan[1]

[1] University of Technology, Sydney, FIT, P.O. Box 123, Broadway NSW 2007, Australia
janeb@it.uts.edu.au

Abstract. This paper reports on an extension to Geographic Information Systems (GIS) that can intelligently analyse and record qualitative information of the surrounding area when adding a feature to a map. This recorded qualitative spatial information can be utilised to perform queries such as path generation using landmarks. Although, there is a lot of research on qualitative spatial reasoning, none of the currently available GIS do actually incorporate this kind of functionality. There have been systems developed that do have functions for adding new features, or generating paths; however they do not generally analyse and record, or use, qualitative information. We have implemented a prototype illustrating our approach.

Keywords: Intelligent Systems, Spatial Reasoning, Knowledge Representation, GIS

1 Introduction

Most Geographic Information System (GIS) computations are quantitative, relying on visualisation tools and user interaction to provide the qualitative insights that users need. Although quantitative representations do allow powerful and efficient calculations, they fall short when users lack information about for example the exact positioning or the geometry of the features involved. It has been recognised that a quantitative approach is an inappropriate representation of human cognition and spatial reasoning (e.g. [7]). Most human reasoning about physical space appears to use a qualitative interpretation of that space. Therefore, purely quantitative methods are inadequate and qualitative methods are essential to effectively represent spatial information.

In recent years, there have been various proposals for spatial descriptions, which are qualitative rather than quantitative to overcome the problem of imprecise spatial information (see [3] for an overview). However, practical applications applying the results of these research efforts are rare. This paper reports on the development of an extension to a GIS. We examined several aspects of GIS functionality in terms of adding qualitative reasoning capabilities to these systems, and included qualitative feature adding and path generation to our GIS extension. When a new feature is added to the map, the system will automatically analyse and record the qualitative information of its surrounding area. This recorded qualitative spatial information will be utilised to perform intelligent queries such as path generation using landmarks. Although, there is a lot of research on qualitative spatial reasoning (QSR), none of the currently available GIS do actually incorporate this. There have been systems developed that do have functions for adding new features, or generating paths; however they do not generally analyse and record, or use, qualitative data/information.

A system that did generate qualitative path descriptions is the GRAAD system by Moulin and Kettani [9]. It provides an integrated logical and analogical framework for reasoning about time and space, while generating and describing the paths of a virtual pedestrian in a virtual environment. The analogical framework based on the notion of spatial conceptual map (SCM) specifies a spatial environment; and the logical framework based on a possible worlds approach carries out temporal reasoning and generates plans that are used to compute the virtual pedestrian's displacements in the SCM. It also generates natural language descriptions of the paths. In contrast, our system uses "real world" GIS data and also allows for adding features with qualitative spatial information inferred by the system.

2 Knowledge Representation in GIS

Data in GIS is generally represented in raster or vector format. These files are usually stored in a database together with associated information in the database table. Data in the table can be categorised into two types: quantitative information such as "20" or "40.5" and qualitative information such as "front" or "back". Users can query this information to retrieve particular details such as "which school is on the left of this traffic light". Our extension to GIS allows for qualitative inferences for added features and a qualitative path generation.

Many of the existing GIS do have functions to add features and some also allow for path generation. We reviewed several GIS and found that while all of them had functions to add features, none of them incorporate an intelligent agent to automatically record qualitative information of the surrounding area into the database table. Users would therefore have to enter this information manually. Some systems such as OpenMap® and GRASS® do not even have database tables as they use the Java method to store data. The MapObject® extension of the ArcGIS® package does have a path finding function using shortest path methods. None of the other systems have path generation available. In TakukGIS® and AccuGlobe®, data can also be stored in its GIS database and represented in a database table. There are create/edit feature functions.

2.1 Qualitative Spatial Reasoning in GIS

Research in QSR is motivated by a wide variety of possible application areas such as GIS [5], robotic navigation [8], spatial propositional semantics of natural languages [1, 10], engineering design such as qualitative kinematics [6] and many others. As Cohn [2] points out: "*The principal goal of qualitative reasoning is to represent not only our everyday commonsense knowledge about the physical world, but also the underlying abstractions used by engineers and scientists when they create quantitative models."(p 323)* However, these findings are rarely used in practical applications.

The extension we are bringing forward in this paper is, to our knowledge, the first one to actually add a functionality to a GIS that enables the addition of new features coupled with the inference of qualitative spatial information about the surrounding area.

2.2 AccuGlobe®

AccuGlobe® [4] is an ideal GIS software package for prototype implementation, as it is the only user-friendly free GIS software that has create/edit feature functions. The software includes functionality such as zooming, panning, measuring, selecting feature, drawing objects, etc. Information, either quantitative or qualitative, will be stored in a database table. However, when new features are added to the map, the system will not automatically analyse and generate the feature's relation to other objects in the map. Instead, the user has to manually enter the information into the database table.

3 Integrating Qualitative Aspects into GIS

We developed a prototype that integrates qualitative aspects into an existing GIS by allowing features to be added qualitatively. In addition, the prototype automatically analyses, and records qualitative information of the surrounding area. We also developed a road map representation that supports qualitative spatial reasoning based on landmarks, thereby providing an effective tool for path generation.

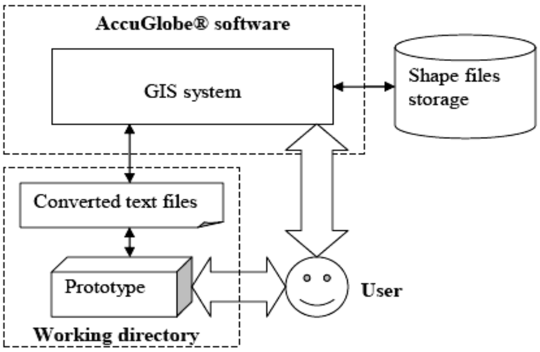


Figure 1: Integrative framework

We used PERL to implement our prototype. AccuGlobe® provides a functionality to export the feature attributes recorded in the shape file to a working directory, and even convert it to an easily readable and editable text file format. The prototype can perform any necessary action to the exported text files, instead of working directly on the shape files. AccuGlobe® will automatically convert the text file to a shape file during the import process. Figure 1 on the previous page shows the framework we used for the prototype.

3.1 Qualitative Database Table Design

Figure 2 shows a simple example of a database table as used in the shape files, illustrating feature entities and attributes.

Feature	**Type**	**Streetname**
A	House	Market St

Figure 2: Database table example

The words “Feature”, “Type”, and “Streetname” in the top row in bold are the feature entities. The words at the bottom row “A”, “House”, and “Market St” are the feature attributes.

3.2 Road Segment Layout

An appropriate road segment layout is very important as the path-finding algorithm generates its way through these road segments. After some trials and testing, we have come up with an easy and effective way to organise these segments. This layout also allows the inference of relative direction relations both when adding features to the map and when generating a path. Roads are generally already divided into segments in GIS data. However, we found it effective to use road intersections as segment dividers. In Figure 3a, there is only one road with no intersection, so there will be only one road segment, which will receive the unique segment id “0”. In Figure 3b, there is an intersection between the vertical road and the horizontal road. Therefore the horizontal road will be divided into two different road segments with the ids “0” and “0.1”. The vertical road also needs a separate id since it is also a road segment; its id is “1”. The layout displayed in Figure 3c follows the same logic.

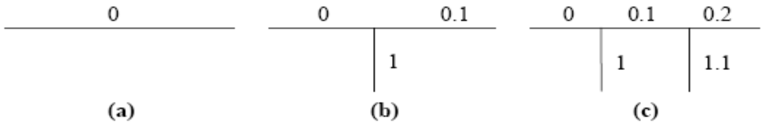


Figure 3: Division of road segments

There is a sequence and meaning for each segment id in order for the path-finding algorithm to work, and to support the inference of qualitative spatial direction relations. Each road segment is either vertical or horizontal. Moreover, all vertical road segments will have segment ids in odd numbers (*before the decimal point*) and all horizontal road segments will have segment ids in even numbers (*before the decimal point*). The numbering always starts at the left or the top for horizontal or vertical road segments respectively. The value before the decimal point is like the Y-axis in a graph having the smallest value at the top and the largest value at the bottom. The value after the decimal point is like the X-axis, with the smallest value on the left and the largest value on the right. Figure 4 shows this pattern clearly by grouping the segments accordingly. This pattern will allow the path-finding algorithm, which will be discussed in sub-section 3.4, to determine which direction to take in order to reach the destination. Any road network can be transformed into this layout because the layout does not consider distances. In practical terms, the road segment layout could be another map layer.

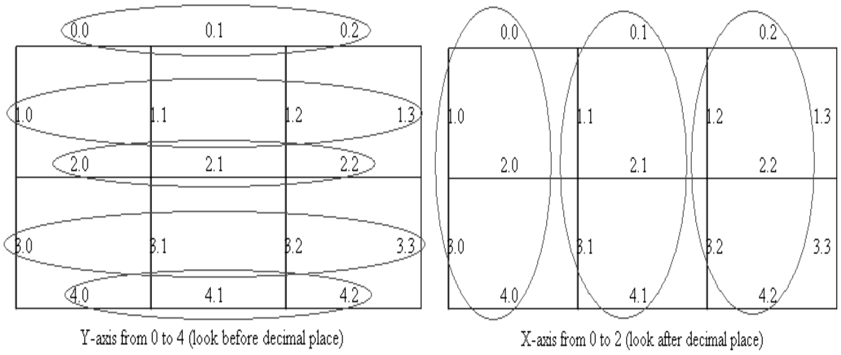

0.0
0.1
0.2
1.0
1.1
1.2
1.3
2.0
2.1
2.2
3.0
3.1
3.2
3.3
4.0
4.1
4.2
Y-axis from 0 to 4 (look before decimal place)
X-axis from 0 to 2 (look after decimal place)

Figure 4: X and Y-axis pattern in the proposed road segment layout

The sample data used for the prototype implementation includes mainly buildings, referred to as address features. In order to fully utilise the road segment layout, address features are issued with "house ids" that start with the value of the segment id on which they are located. Figure 5 illustrates the procedure of assigning house ids.

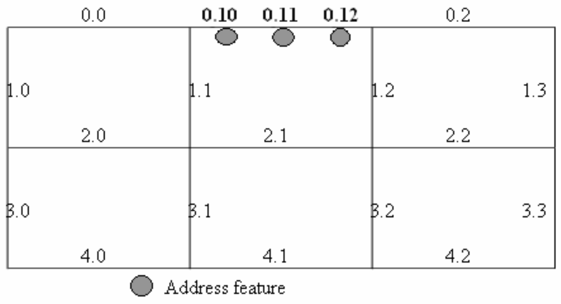


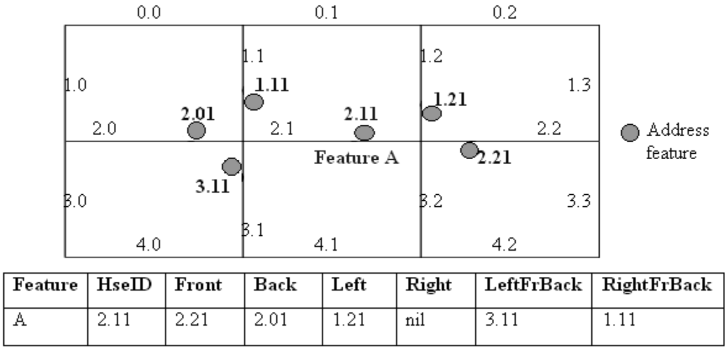
Figure 5: Address features added to road segment

Suppose that we want to add three address features on a horizontal road segment with segment id "0.1". The left most address feature that is located on this particular road segment will have the house id "0.10". The second left most address feature on the same road segment will have the house id "0.11", and the third left most address feature will have the house id "0.12". They are all preceded by "0.1" to indicate their road segment location. Figure 5 illustrates this clearly, the three address features are added onto the road segment with segment id "0.1". If a new address feature was added between two existing address features such as "0.10" and "0.11", the new house id would be chosen between "0.10" and "0.11", e.g. "0.101".

3.3 Adding Features by Storing and Inferring Qualitative Spatial Relations

We modified the attribute tables of an existing data set in order to accommodate the storage of qualitative relations that can either be entered or inferred. In addition to relative direction relations such as left and right, we also included a small subset of topological relations. In order to resolve any reference frame issues, we presume a bird’s eye view. For an address feature on a horizontal path, we will go from left to right, which means the right is always the front and the left is the back. For an address feature on a vertical path, we will go from top to bottom, which means the bottom is always the front and the top is the back.

Figure 6 shows an example of a modified attribute table. If we need to record the qualitative relations of “Feature A” (house id “2.11”) with the features of its surrounding area; we have to create new entities for these qualitative relations in the attribute table. With this recorded, qualitative information of the feature’s location in relation to other objects can easily be accessed by looking up the attribute table.



Feature	HseID	Front	Back	Left	Right	LeftFrBack	RightFrBack
A	2.11	2.21	2.01	1.21	nil	3.11	1.11

Figure 6: Attribute table with added qualitative relations’ entities

The entities “Front”, “Back”, “Left”, “Right”, “LeftFrBack”, and “RightFrBack” are added to the attribute table, as displayed in Figure 6. We can see that the entity “Right” has a nil attribute value. This is because by turning right at Feature A, there is no feature that can be reached; this is represented by a “nil” in the attribute table.

Figure 7 illustrates some address features laying on both the vertical path and horizontal path. House id “2.1” is located on a horizontal path and house id “1.2” on a vertical path. The attributes of their qualitative relations with the surrounding features, which can be extracted from the database table, are summarised below Figure 7. If the user goes the other way around, from bottom to top, or right to left, the relations will automatically be transformed (*e.g., the front becomes the back, etc.)* .

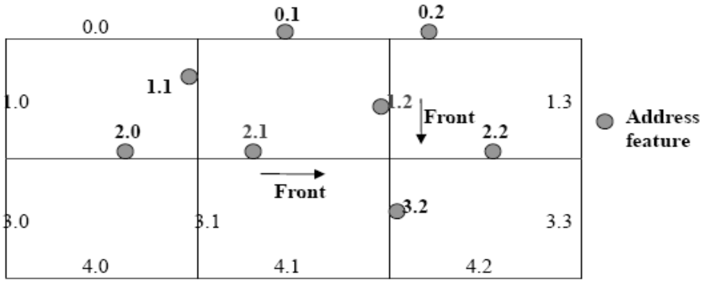


Figure 7: Qualitative relations of features

Horizontal path (house id “2.1”)	
Front	house id “2.2”
Back	house id “2.0”
Left	house id “1.2”
Right	house id “3.2”
Left from back	Nil
Right from back	house id “1.1”

Vertical path (house id “1.2”)	
Front	house id “3.2”
Back	Nil
Left	house id “2.2”
Right	house id “2.1”
Left from back	house id “0.1”
Right from back	house id “0.2”

3.4 Path Finding Algorithm

This section explores the simple path-finding algorithm we developed. Due to our road segmentation into vertical or horizontal segments, we can subset our path-finding algorithm into four functions: “odd-to-odd”, “odd-to-even”, “even-to-odd” and “even-to-even". The “odd-to-odd” function would be used if both the start and destination address features have odd house ids and therefore lie on vertical road segments in the map. A simple example of an odd-to-odd function is shown in Figure 8, where the start and destination address features have a house id “1.11” and “1.21” respectively.

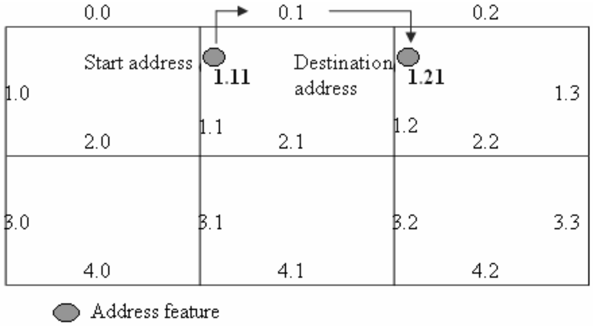


Figure 8: Odd-to-odd function example

In order to select the appropriate function, the system takes the values of house ids, “1.11”, and “1.21”. Both the values before the decimal point are “1”, which are odd. Hence, the odd-to-odd function would be selected. The odd-to-odd function itself determines which particular road segments are needed for navigation. The function specifies which position the destination address is in relative to the start address. It will check on the house ids again. The preceding decimal number for both is “1”;

hence they lay on the same level of the y-axis. After the decimal point, the start address has a value of “11” and the destination address has a value of “21”. The destination address’s value is larger than the start address’s value on the x-axis; the destination address is therefore on the right of the start address. The results in a path that goes straight to the top, turns right at the next available turn off, then continues to go straight until it finds the right turn with segment id “1.2”, along which it can locate the feature address with house id “1.21”.

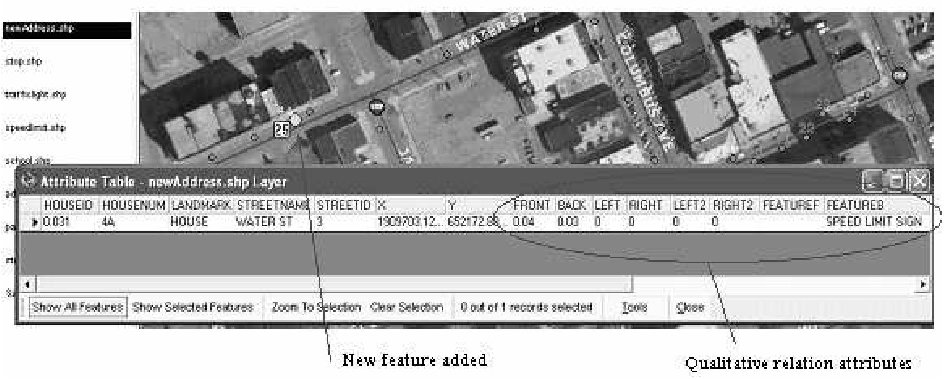
The other functions (e.g. *odd-to-even algorithm*) have the same underlying logic. Mainly they check on the values before and after the decimal points to determine into which direction the path should be generated.

4 Prototype Evaluations

For the software evaluation, we used a photographic image [4] that consists of several layers for different feature types each such as traffic lights, stop signs etc. We have successfully added features to the map qualitatively using topological relations descriptions. The prototype automatically generates and records the feature id, street id, street name, and the qualitative relations with the surrounding area.

4.1 Experimental details

We tested our GIS extension against several data entries. Once the data was added, we verified information in the database tables for inferred qualitative information. We did this for three different data entry types, namely for the quantitative, simple qualitative and qualitative on different streets addition of features to the map; and subsequently uploaded the newly inferred information into the image map.

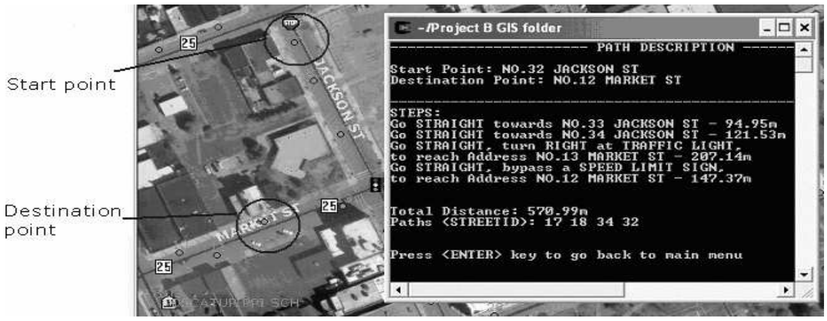


HOUSEID	HOUSENUM	LANDMARK	STREETNAME	STREETID	X	Y	FRONT	BACK	LEFT	RIGHT	LEFT2	RIGHT2	FEATUREF	FEATUREB
0.031	4A	HOUSE	WATER ST	3	1909700.12...	652172.88...	0.04	0.03	0	0	0	0		SPEED LIMIT SIGN

Figure 9: Added feature to map and feature attributes to database table

The following is an example of a simple qualitative addition of a new feature. When we add an address feature between No.3 and No.4 of Water Street; any further information required by the attribute table is automatically inferred by the system. Figure 9 shows the newly added feature with the reference addresses entered by the user and all the other attributes inferred by the system. The figure also shows the new feature being uploaded into the map.

We further evaluated the usefulness of our data representation by utilising this qualitative information for path description generation using direction relations and landmarks, as shown in Figure 10. This gave us an additional level of evaluation when doing this with newly added features. Without this qualitative information being recorded, it is not possible to effectively generate a path description as there is no sufficient information about the surrounding area of each feature.



```
------------------------ PATH DESCRIPTION ------
Start Point: NO.32 JACKSON ST
Destination Point: NO.12 MARKET ST

------------------------------------------------
STEPS:
Go STRAIGHT towards NO.33 JACKSON ST - 94.95m
Go STRAIGHT towards NO.34 JACKSON ST - 121.53m
Go STRAIGHT, turn RIGHT at TRAFFIC LIGHT,
to reach Address NO.13 MARKET ST - 207.14m
Go STRAIGHT, bypass a SPEED LIMIT SIGN,
to reach Address NO.12 MARKET ST - 147.37m


Total Distance: 570.99m
Paths <STREETID>: 17 18 34 32


Press <ENTER> key to go back to main menu
```

Figure 10: Path description

4.2 Limitations of the system

Despite its usefulness, our system does also have limitations that need to be discussed. As explained in Section 3.3, for an address feature on a horizontal path, we will go from left to right, which means the right is always the front and the left is the back. For an address feature on a vertical path, we will go from top to bottom, which means the bottom is always the front and the top is the back. There may be a problem if there are two address features lying side by side, one across the street of the other. There is no front, back, left or right relation in this case. Another limitation is the quantitative distance which is currently only used as part of the path generation output, but not during the reasoning process. If we were to use quantitative distance for the path generation, we were again to face problems when data is imprecise.

While every road network can be transformed into our road segment layout theoretically, our system does not yet provide the functionality for such a transformation and relies on manual transformation from existing road network into what is commonly known as “Manhattan Structure”.

For the generation of path descriptions, if there is a cluster of addresses between two locations (start and destination locations); the path description generated by the system could be too long to be useful, as every single address would be considered. A solution could be to automatically skip the path description of some addresses if the list is too long.

5 Conclusions and Outlook

In this paper we discussed the possibility of an extension to Geographic Information Systems (GIS) that can intelligently analyse and record qualitative information of the surrounding area when adding a feature to a map. We then presented a prototype implementation of such a system based on a road segment layout and path finding algorithm that not only support spatial reasoning but also allow the automatic inference and storage of qualitative spatial relations when new features are added to a map. This recorded qualitative spatial information can be utilised to perform queries such as path generation using landmarks, this particular function was also implemented in our prototype. An evaluation of the system confirmed the well known assumption that purely quantitative methods are inadequate and qualitative methods are essential to effectively represent spatial information.

Our current prototype is text based, but we are planning a graphical interface to allow user input for feature addition. The path generation still needs to be optimised as it is not necessarily the shortest path. This is a general problem that has received a lot of attention in GIS research.

References

1. Aurnague, M., Vieu, L.: A three-level approach to the semantics of space. C.Zelinsky-Wibbelt (ed.) The semantics of prepositions – from mental processing to natural language processing. Mouton de Gruyter. Berlin (1993)
2. Cohn, A.: The Challenge of Qualitative Spatial Reasoning. ACM Computing Surveys. Vol 27. No.3. (1995) 323-324, UK
3. Cohn A.G., Hazarika S.M.: Qualitative Spatial Representation and Reasoning: An Overview. Fundamenta Informaticae, volume 46, No.1-2, (2001), 2-32
4. DDTI (Digital Data Technologies, Inc.) (2005) http://www.ddti.net
5. Egenhofer, M., Mark, D.: Naïve Geography. Frank and W. Kuhn (eds.) Spatial Information Theory: A theoretical basis for GIS. Proc. COSIT'95. LNCS No. 988. Springer-Verlag (1995) 1-16
6. Faltings, B.: A symbolic approach to qualitative kinematics. Artificial Intelligence. 56(2) (1992)
7. Kuipers, B.: Modeling Spatial Knowledge. Cognitive Science 2 (1978) 129-153
8. Kuipers, B., Byun, Y.: A robot exploration and mapping strategy based on a semantic hierarchy of spatial representations. Journal of Robotics and Autonomous Systems (1991) 47-63
9. Moulin, B., Kettani, D.: Route generation and description using a logical and an analogical framework. Annals of Mathematics and Artificial Intelligence 24. (1998) 155-179
10. Sablayrolles, P.: Spatio-temporal semantics in natural language: the case of motion. M. Aurnague et al. (eds.) Semantics of Time, Space, Movement and Spatio-Temporal Reasoning, Working Papers of the 4th Intl. Workshop (1992) 69-88.