Computer-Human Interaction Issues when Integrating Qualitative Spatial Reasoning into Geographic Information Systems

Carl P. L. Schultz
The University of Auckland
csch050@ec.auckland.ac.nz

Hans W. Guesgen
The University of Auckland
hans@cs.auckland.ac.nz

Robert Amor
The University of Auckland
trebor@cs.auckland.ac.nz

ABSTRACT

To allow the immense volume of spatial data currently available to be used effectively, people need intelligent query tools that are simple and intuitive. Standard query tools have a number of serious usability limitations, as they often rely solely on numerical approaches when dealing with spatial information. The qualitative reasoning community has addressed this issue, by providing powerful formalisms based on the way that humans deal with spatial information, however, integrating these methods into numerical systems raises a number of new CHI problems. This paper addresses three key CHI challenges when combining qualitative and numerical methods: (1) managing the subjective, ambiguous nature of qualitative terms, (2) providing a powerful, yet simple query system, and (3) effectively visualising a complex, fuzzy qualitative query solution. A qualitative GIS called TreeSap is presented, which demonstrates that, with the use of CHI principles, query tools can be both powerful and accessible to non-expert users.

Author Keywords

Graphical user interfaces, Fuzzy Qualitative Spatial Reasoning

ACM Classification Keywords

H5.m. Information interfaces and presentation (e.g., HCI): User Interfaces.

INTRODUCTION

With continuing advances in sensing and surveying technologies, rapidly increasing amounts of data are being collected each year, particularly in the spatial domain [1]. To make effective use of this immense volume of data, people need to be able to extract specific, relevant pieces of information with powerful, yet *accessible*, query tools [1,2]. This can only be accomplished by employing a combination of CHI principles. Standard Geographic Information Systems (GIS) have lacked focus in this area, with querying tools often being restricted and cumbersome to use. In particular, query systems are either extremely limited (e.g. web based GIS systems designed for public use often have very restricted query structures), or require knowledge in areas such as set theory and database languages (e.g. Structured Query Language, SQL), and are thus only available to a very narrow range of expert users.

The key problem is that modern GIS rely entirely on numerical methods when working with spatial information, which leads to a number of CHI issues [1,2]. People find numerical methods non-intuitive, for example, a statement such as “The café is at latitude 23 minutes, 8 degrees, and longitude..” is far less natural than “The café is opposite the art gallery on Symonds St” [1,3]. Further to this, numerical approaches cannot handle uncertainty in information, despite uncertainty being an intrinsic property of information about the physical world [3]. For example, it is impossible to define the boundaries of a coastline with absolute numerical accuracy, due to physical limitations of measurement precision, and the issue of information becoming out of date [4,5]. Another example is the inherent vagueness in a statement such as “The Forest is *near* the Pond”. Despite this, humans still reason with imprecise and vague spatial information [3].

In everyday situations, humans often reason about spatial information in a qualitative manner, in particular, working with uncertainty [1,5]. Qualitative Spatial Reasoning (QSR) is a field of study based on these principles, with the development of coarse, “commonsense” formalisms for reasoning about space. These methods have been strongly influenced by Allen’s qualitative temporal logic [3,4,6,7]. Allen presents a set of thirteen atomic temporal relations that describe relationships between time intervals. He describes key attributes for effective qualitative reasoning, that have been extended to the spatial domain [5]:

- The logic must handle *imprecision* in the data, given that people often express spatial information in a relative manner, with no reference to an absolute coordinate.
- *Uncertainty* in the data must be handled, so that a partial relationship between two features is accepted by the calculus, if the exact relationship is not known.

Freksa in [8] presents the notion of conceptual neighbours [8]. Neighbourhoods are defined by considering the continuous translation or deformation of objects within the given relation. For example, if two regions are not touching (*disconnected*), and one region is moved towards the other, the regions must first become *adjacent* before they *overlap*.

Reasoning about extending objects (i.e. beyond simple primitive shapes) has been approached in numerous ways. Jungert [9] presents symbolic projections for reasoning about 2D extended objects. The approach uses vertical and horizontal cutting lines to project the objects onto Cartesian axes. 1D strings are then derived for each axis, that indicate relationships between objects. These strings are used to infer information about extended objects [9].

Evans [10] proposed a triangular model for reasoning about extended objects, where right angled triangular areas mark out four qualitative relationship regions with respect to a reference object: front, right, back, and left. The vertex of each triangular area is positioned over the centroid of the reference object. Peuquet and Ci-Xiang [11] extended this model to handle relationships where the object sizes and shapes were not necessarily uniform, leading to a more cognitively intuitive interpretation of a scene.

Work on reasoning about qualitative distance includes Hong [12] and Hernandez [13], where space is divided into circular regions, centered around a reference object. As the distance from the reference point increases, distinctions between distance regions become increasingly coarse.

The primary aim of this work is to show that key problems in GIS (non-intuitiveness, imprecision and vagueness) can be resolved by applying CHI principles to assist the integration of QSR techniques into numerically based software applications.

QUALITATIVE FORMALISMS

Qualitative methods are a coarser, language based approach to working with information, and have been used to specify spatial relationships and properties [3,4,5]. A system can make use of multiple formalisms by applying the appropriate combination of formalisms for a given situation, and combining the results. Escrig and Toledo [14] apply this principle when integrating numerous QSR methods in the QNavSim robot navigator algorithm, where different qualitative orientation and distance schemes are used depending on the information available. Two QSR formalisms are discussed in this section, namely Qualitative Proximity (QP) and Region Connection Calculus (RCC).

Figure 1. Subset of the distance relationships defined in QP, where A and B are objects or regions

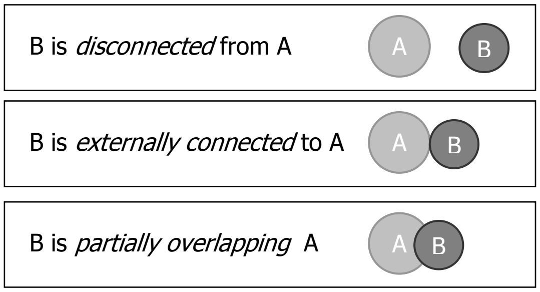


Figure 2. Subset of the region relationships defined in RCC8 [16], where A and B are regions

The QP formalism is an adapted version of the Fuzzy Proximity formalism described in [15], and is used to reason about **distance relationships** between spatial objects. The possible relationship types, in order of increasing distance, are: touching, very near, near, moderately near, moderately far, far, very far. Figure 1 illustrates two example relationships between a pair of objects, A and B.

Region Connection Calculus (RCC) proposed by Randell et al. [16] describes **relationships between spatial regions** based on their topological properties, and is thus independent of any coordinate system [16]. RCC8 is a system which selects a set of eight of these basic relations, such that the set covers any possible physical relation between two regions, and such that there can be no physical relation which is an instance of two basic relation types (jointly exhaustive, pairwise disjoint). A subset of the relations are illustrated in Figure 2 [4,16].

Applying Fuzzy Logic

To address the issue of vagueness in spatial information, we have combined qualitative methods with fuzzy logic [3,4]. To illustrate this, consider the following query: "Find all objects *within* A". As shown in Figure 3, a "within" membership value is assigned to every relationship that A shares with some other region, indicating how closely each relationship matches the "within" relationship type. More generally, the standard alpha notation can be used [3,4,15], where α_0 indicates the highest possible membership (a value of 100%, where the relationship is definitely considered a "near" relationship), and:

$$\alpha_1 > \alpha_2 > \alpha_3 > \ldots$$

indicating decreasing membership values, where the exact values of α_1, α_2, α_3,… can be determined according to the application [3,4,15].

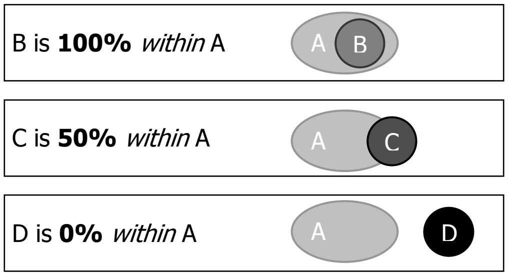


Figure 3. Results of the query "Find all objects *within* A". A fuzzy membership value is assigned to every relationship that A shares with another region, representing how well each relationship matches the definition of "within". B is considered definitely "within" A, however D is definitely not "within" A. C is considered partially "within" A.

Fuzzy membership values are thus assigned to relationship types using the conceptual neighbourhood approach proposed in [3,4]. Membership grades are assigned to relations according to the distance the relation is from a reference relation in the conceptual neighbourhood graph [3,4]. The further away a relation is from the reference relation, the lower its membership grade [3,4]. Figure 4 illustrates the assignment of membership grades to relations with respect to the "externally connected" (RCC8) and "near" (QP) relationship type.

Deriving Qualitative Relationship Networks

A complete network of qualitative relationships is constructed for each formalism [3,4], based on the raw numerical data [15]. Each network is then referred to by a system's query processor in order to support more sophisticated querying.

In RCC, the first step is to determine connection relationships between each pair of regions. This can be accomplished with geometric set operations in simple equations. For example, two regions are considered partially overlapping if the area of their intersection is greater than zero, but less than either region's individual area:

$$0 < \textit{intersection}(\text{A,B}) < \textit{area}(\text{A}),\ \textit{area}(\text{B})$$

Similarly, two regions are considered equivalent if the area of their intersection is the same as both of the regions' individual areas:

$$\text{intersection(A,B)} = \text{area(A)} = \text{area(B)}$$

The set of all relationships (between every pair of regions) makes up the complete RCC relationship network.

In QP, the first step is to take the distances between each pair of objects. For example, in a 2D scene, this can be accomplished by computing the minimum distance between each pair of objects, a and b, using:

$$\text{distance}(a,b) = \min\left(\sqrt{(x_b - x_a)^2 + (y_b - y_a)^2}\right)$$

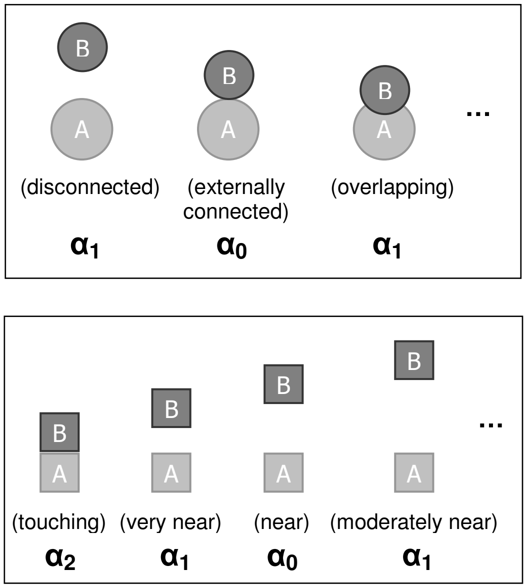


Figure 4. Extract from the set of RCC (top) and QP (bottom) relations, arranged according to their conceptual neighbourhoods. Membership values (alpha notation) have been assigned with respect to the "externally connected" (top) and "near" (bottom) relationship types.

The next step is to normalise the distance values to a number between 0 and 1. Thus, a decision needs to be made as to what value is considered a "near" distance (the issue of resolving ambiguity is explored in a later section). The normalised distance value (d) is then transformed into a fuzzy membership value. If the normalised distance value is 0, then the two objects are touching [15], resulting in a fuzzy value of 1. As the normalised distance value increases, we consider the two objects to be increasingly further apart [15], resulting in a decreasing fuzzy value (between 1 and 0). The following function [15] was used in this case, however, any function with similar properties may also be suitable:

$$\text{Fuzzy membership value} = \frac{1}{1 + d^2}$$

This fuzzy value is then mapped to one of the seven QP relation types (from "touching", to "very far away"). The set of qualitative distance relationships (between every pair of objects) makes up the complete QP relationship network.

GIS INTERFACE REQUIREMENTS

Usability and CHI are central issues for effectively integrating QSR into a standard GIS framework. The aim is to provide query tools with the following two key requirements.

Firstly, the query tool must be expressively powerful, so that a user can access and isolate any piece of relevant information from the data. If the query tool is not expressive enough, certain users will not be able to present the appropriate criteria to the system.

Secondly, query tools must be readily accessible to a large variety of users. Thus, the tool must be intuitive and simple to use, without requiring expert knowledge in any particular area.

Meeting both of these requirements is a central CHI problem. The difficulty is that, on one hand, the simplicity of the interface must not compromise the query tool's expressive power. On the other hand, many problems arise when a user must provide complicated input:

- A query may be malformed (e.g. a syntax error in SQL).
- A challenge is in teaching a user how to operate a query interface that allows arbitrarily complex input, at the same time minimising the potential for a misunderstanding of the system. The user thus requires constant feedback, and reassurance that the intended query is accurately represented by their actual input.
- A user may be required to learn and remember numerous commands and keywords (such as "select", "where", "drop index" from SQL), increasing learning time, along with the chance of a misunderstanding or a syntax error.
- A direct reflection of the underlying formalisms is also desirable, as it allows a user to develop an accurate understanding of how the spatial information is being managed. A user can then benefit from the full potential of the formalisms.
- A further issue relates to the structuring of a query. If a complicated query has poor structure, or has a format that is too general, then the query may become either ambiguous, or far too difficult to understand. On the other hand, if a query format is too strict, then it may lose the desired expressiveness.

The TreeSap GIS application was produced to address these CHI issues, by demonstrating a qualitative reasoning approach along with different visualisation methods. TreeSap's querying and visualisation approaches are discussed in the following sections.

TREESAP – QUALITATIVE REASONING GIS

TreeSap (Topographic Reasoning Application) is a desktop GIS application, that provides powerful spatial reasoning tools, with a strong emphasis on usability. TreeSap was produced as an example of how the interface into a powerful query tool can be intuitive and simple to use.

TreeSap applies QSR in two areas. Firstly, QSR is used to generate a qualitative relations network layer on top of the standard numerical data, which is used to facilitate more intelligent, powerful query support. This involves the CHI problem of resolving inherent ambiguity in qualitative terms, e.g. determining an exact definition for "near". Secondly, the design of TreeSap's user interface was driven by QSR, with the aim of allowing a wide range of users to take full advantage of the underlying QSR formalisms, in a simple and intuitive manner. This involves two fundamental CHI issues:

1. How can a user present a qualitative query to the system in a simple, yet clear and unambiguous manner, without sacrificing query expressiveness?
2. How can the system convey a query solution to a user that consists of fuzzy qualitative information on top of detailed numerical data, in a simple and intuitive manner?

QSR has been incorporated using a complete implementation of the QP formalism (the RCC8 formalism is in the process of being implemented into the TreeSap framework). The three CHI issues of disambiguation, querying, and visualisation are discussed in the following sections.

Resolving Ambiguity

Resolving ambiguous words such as "near" and "within" is essentially a CHI issue, in that the aim is to synchronise both the system's interpretation and the user's interpretation of a qualitative term. In the case of QP, a decision needs to be made as to what numerical distance is considered a "near" distance. In RCC, one issue is deciding what "within" means for a given region. That is, "within" could be interpreted as "surrounds" (e.g. the lake is within the forest), or "inside" (e.g. the train station is within downtown). Two methods for disambiguation are presented. The first approach refers to the numerical content of the data (this can be used as the automatic default interpretation), and the second allows a user to provide explicit definitions (this can be applied if the default settings are unsatisfactory).

Heuristics Applied to the Numerical Content of Data

Qualitative terms are relative to the numerical relationships between objects being considered. For instance, two uses of the word "near" can refer to two very different numerical scales, e.g. "Auckland is near Hamilton", and "the lecturer is near the blackboard". Thus, heuristics that only look at the numerical content of spatial data (distinct from the semantic content) can be considered, to help automatically resolve ambiguity.

One such heuristic, for 2D spatial data, is to determine the bounding box of the objects, and set the length of the longest side as the minimum "very far" distance. The use of "very far" is now meaningful, as at least one distance relationship will fall into this category (the distance relationship that defined the longest side), and likely more.

This approach is robust with reasonably uniform spatial datasets, as the fuzzy logic compensates for small differences in nearness settings. This is because the ordering of the weightings between distance relationships is maintained. For example, Figure 5 illustrates two results for the query "find all Cafés *near* Symonds St". In each case, slightly different distances are equated to the term "near". Despite this, both results indicate the same relevant information, specifically that "Cezze Bar" is the nearest

“near” distance setting	**288m**	**202m**
Cezze Bar	100%	83%
Yellow Crow Café	83%	66%
Rachino’s Café	0%	0%

Figure 5. Two results for the query “Find all Cafés *near* Symonds St.”, given different settings for a “near” distance. In both cases, the relevant information is the same, as the ordering is maintained.

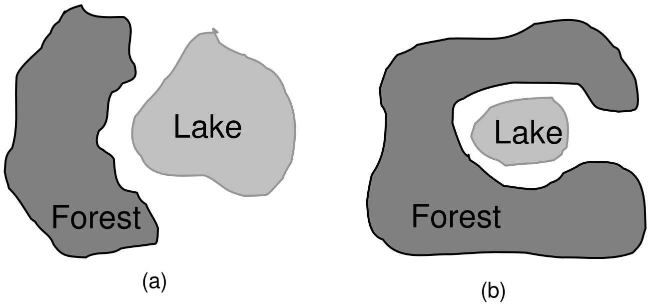


Figure 6. In both cases the lake is technically disconnected from the forest, however in (b) a more intuitive interpretation is that the lake is within the forest.

café, and it is closer than “Yellow Crow Cafe”, and that “Rachino’s Cafe” is not consider near at all. In general, a good “nearness” setting is one that spreads the data over many qualitative categories. A poor “nearness” setting groups the data into a small number of qualitative categories, leaving some categories entirely empty.

In RCC, one issue is when to apply the “within” relationship between two complex regions. While two regions might be considered disconnected in a strictly technical sense, this is not always conceptually satisfying. Figure 6 illustrates this, where the lake may or may not be considered “within” the forest. A heuristic is to consider cases where one region is much larger than another. In these cases, the “within” relationship is determined using the convex hull of the larger region. Further work will determine the effectiveness of this heuristic, along with other similar approaches in RCC.

Explicit Definitions Provided by the User

The meaning of a qualitative term, such as “near”, is inherently subjective, and thus any exact definition will vary between users. User subjectivity is difficult to take into account a priori, as a person’s understanding of a qualitative term depends on their unique experiences of the world.

For instance, a person’s understanding of a qualitative word can be affected solely by the semantic content of the data, e.g. Figure 7 illustrates a case where two (essentially identical) numerical region relationships have different qualitative interpretations, according to what they represent in the physical world. In the first case, the lake is considered within the forest, however, in the second case, the lake is not considered within the building. This is due to the experience of forests having more vague boundaries, compared to buildings.

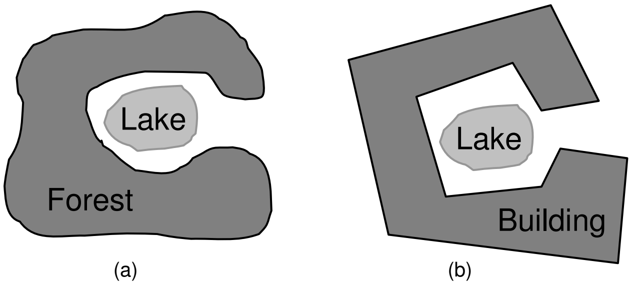
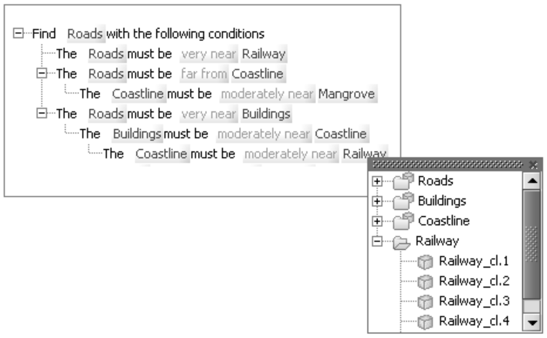


Figure 7. Two (essentially) numerically identical scenarios with different qualitative interpretations (a) the lake is within the forest, but (b) the lake is not within the building.



Find Roads with the following conditions
The Roads must be very near Railway
The Roads must be far from Coastline
The Coastline must be moderately near Mangrove
The Roads must be very near Buildings
The Buildings must be moderately near Coastline
The Coastline must be moderately near Railway

Figure 8. Screenshot of the interface used to specify a query. The user builds a natural language query tree using the mouse.

In QP, a solution is to let the user explicitly specify a “near” distance. In TreeSap the user is presented with a map, and is asked to specify what they consider to be a “near” distance, by selecting two features. Future work will explore whether this approach is feasible for very large datasets, and spatially non-uniform datasets. In RCC, one approach is to let the user specify how crisp or vague a boundary is for a given type of region (forests, buildings, suburbs, …). Future work will determine whether this is an effective approach, and identify ways in which this can be accomplished in an intuitive manner.

Qualitative Querying

TreeSap’s query interface, illustrated as a screen shot in Figure 8, places a strong emphasis on being intuitive and simple to operate, while providing a user with the full benefits of the underlying qualitative formalisms. The query tool is natural language driven, attempting to reflect the underlying qualitative formalisms as directly as possible, and is thus easily accessible to non-expert users.

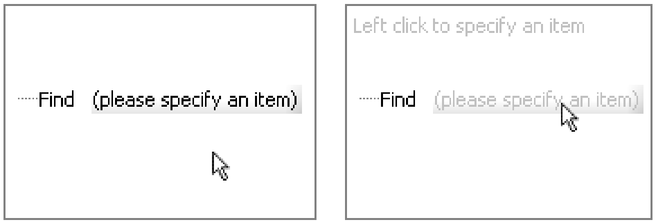
The user builds their query as a hierarchical tree structure of conditions, where nodes of the tree are search criteria, or constraints. These criteria are described in terms of a subject, and its qualitative spatial relationship with another object. The query tree allows nested relationships with no restriction on the depth of nesting, thus allowing for a query of arbitrary complexity. The hierarchical structure allows the complicated queries to be organised in a natural and intuitive manner.

Each nested condition acts to constrain its parent. For example, consider the condition from Figure 4 “The Roads must be very near Buildings”. The buildings are, in turn, constrained by the nested condition that states: “The Buildings must be moderately near Coastline”. A further level of nesting now constrains the coastline: “The Coastline must be moderately near Railway”.

A query is built up in stages, and at each stage the user is presented with the results of the partial query, visualised on a map. This constant feedback is important, as it allows a user to develop a thorough and accurate understanding of how the query tool works.

The query building process is entirely mouse driven, and thus has a number of usability advantages as follows:

- A query can never be malformed, due to the nature in which it is built. This is not the case for other approaches such as SQL, where the query could have syntax errors.
- There is no possibility for erroneous or invalid input, such as incorrect data types being entered into a field. Input is always valid, thus minimising the opportunity for a misunderstanding to develop.
- The user is given immediate feedback through the mouse-driven interface. When the mouse moves over an interactive component (such as a button), the component lights up, as illustrated in Figure 9. This implicitly suggests to the user that the component is interactive. This is reinforced by a message that explicitly tells the user what interactions the component supports. This feedback encourages a user to explore and familiarise themselves with the interface.
- All user communication and interaction (such as component highlighting, tool-tips, popup selection boxes, and messaging) happens near the mouse pointer. This simplifies the query building process, as it is likely that the user’s attention will be focused on the mouse. It also reduces user effort, by avoiding large eye and mouse movements between targets.
- Being mouse driven, the interface is easy to operate, as it does not require a user to learn or remember commands or keywords.



Left click to specify an item

Figure 9. Screenshots illustrating how interactive components provide the user with feedback. As the mouse moves over an interactive component, the component changes colour (right).

The nature of progressively building a tree which describes each of the user’s desired constraints ensures that the query building process is simple, while also providing constant feedback (the results of the query are immediately displayed whenever a condition is specified or altered). Further to this, the use of qualitative formalisms to describe the spatial relationships in the query make it intuitive and easy to learn. Future work will involve allowing a user to combine numerical statements with qualitative statements in a query, for example, “Find all Cafes *near* the Railroad, such that the Railroad is *within 5km* of Downtown”. Also, the current query structure uses an implicit AND to join the conditions. This will be extended to include other conditional operators, such as OR, and XOR, introducing new CHI challenges.

Qualitative Visualisation

Standard geographic data is often large and detailed. The data that results from a qualitative query is even more complicated, with the introduction of fuzzy values on top of the large, detailed datasets. Further to this, there is a need to reflect the innately ambiguous qualitative notions that are present in the underlying formalisms. The key challenge is to present this information to a user in an intuitive manner, while avoiding an approach that requires knowledge in disciplines such as mathematics, artificial intelligence, or database languages. TreeSap addresses the issue of conveying complex qualitative information to the user by demonstrating two visualisation strategies: using transparency, and using a display threshold.

Transparency

In the first strategy, transparency is used to represent how well a feature fulfills the query criteria, as demonstrated in Figure 10. Features that fulfill the query criteria to a high degree are displayed completely opaquely, while features that are less relevant to the query are displayed transparently. The level of transparency used is proportional to how poorly a feature fulfills the query criteria.

Figure 10. Screenshot of the transparency method used to visualise results of the query: "Find all Roads *near* a Specific Building (black circle)".

Using this method, all elements displayed to the user directly convey spatial information. Opaque features represent the solution to the query, and are therefore the most important pieces of information being displayed. These features appropriately attract a user's attention, by being displayed more distinctly than non-solution features. By displaying neighbouring, non-solution features very faintly, the user is implicitly given some spatial context, to assist in the interpretation of the solution. In this respect, transparency offers an intuitive and visually efficient technique for conveying qualitative information.

This strategy presents the user with a static snapshot of the solution, with all the information relating to the query result being provided in a single image. This method is thus ideal if it is required that the query solution be ported onto a hardcopy medium, such as a hardcopy report document, or a newsletter.

Threshold Display

A limitation of the transparency approach is that, while it can provide an instant overview of a query result, it does not effectively convey subtle trends and details. For example, the exact location with the highest solution quality is not always obvious, as subtle differences in transparency can be difficult to recognise. To address this issue, a second approach is proposed that uses a threshold to determine how much information is presented to the user at a given point in time.

Some features fulfill a query's criteria more than others. The notion of a solution quality is thus used to indicate how well each feature meets the given criteria. 100% indicates that a feature meets all the criteria, while 0% indicates that a feature does not meet the criteria in any way. The user can then control a display threshold by dragging a slider. All features that have a solution quality above the threshold are displayed opaquely, and any features with a solution quality that do not meet this threshold are not displayed at all. A scenario is illustrated in Figure 11, where more roads are displayed as the threshold is lowered, revealing an underlying trend.

Figure 11. Screenshot of the display threshold method used to visualise results of the query: "Find all Roads *near* a Specific Building (black circle)" for differing thresholds.

This strategy is a dynamic representation of a query result, with different parts of the solution being revealed at different points in time. A key aspect of this method is that the user has control over the dynamic property by adjusting the threshold, thus revealing trends and patterns in the way that the solution unfolds. For example, consider the scenario that a city council is looking for a site to transform into a reserve for growing native New Zealand kauri trees. After applying the appropriate query, it is observed that one small area meets the criteria by 100%, but a much larger area meets the criteria by around 78%. This second part of the solution will be expressed as a small, independent pocket appearing and growing rapidly, once the threshold has dropped to 78%. This suggests that, with minor adjustments, the larger area may be a more appropriate, long term solution. Thus, the display threshold approach offers a deeper understanding of the query solution.

FUTURE WORK

The ideas proposed in this paper are currently being applied to different domains. For example, a Qualitative Gantt (QGantt) application is being developed, to assist in project planning and management. A standard (non-qualitative) project management system involves a user defining a collection of tasks with exact start and end dates, despite the inherent uncertainty that people have about future events. QGantt allows a user to define tasks by specifying qualitative duration (very short,…, very long), along with a qualitative confidence for that duration (certain, very confident,…, very uncertain). The tolerance of a task is also affected by the uncertainty of its dependencies, thus interesting global analysis can be conducted, such as observing critical path sensitivity (task uncertainty can cause unforeseen delays, affecting the critical path), and earliest and latest possible project completion times (best and worst case for each task tolerance).

Formal usability surveys will be conducted for TreeSap's query interface, to allow a more accurate comparison with existing systems. TreeSap is currently being extended to include the RCC formalism, involving a number of different CHI issues, particularly resolving ambiguity.

CONCLUSION

Integrating qualitative spatial reasoning into a numerically based system raises a number of CHI problems. This paper addresses the CHI challenges of (1) managing the subjective, ambiguous nature of qualitative terms, (2) providing a powerful, yet simple query system, and (3) intuitively conveying a complex, fuzzy qualitative query solution to a user.

When resolving the ambiguity in qualitative terms two strategies are presented. An automatic approach is to use heuristics based on the numerical content of the data. Alternatively, the user can provide an explicit definition of a qualitative term, for example, by specifying two objects that they consider "near" on a map.

The query interface is natural language driven, and can thus be understood without requiring expert knowledge in any particular area. At the same time it allows arbitrarily complicated queries to be unambiguously presented, by organising constraints into a hierarchical structure (without restrictions of number of constraints, or the depth of constraint nesting).

Two visualisation methods are demonstrated, that assist in conveying complex fuzzy qualitative data. Transparency offers a static overview of the information, while threshold cutoff provides a dynamic approach, revealing subtle trends within the data.

ACKNOWLEDGMENTS

The authors would like to acknowledge Tim Clephane who co-developed the TreeSap system. This work has been funded by the Bright Future Top Achiever Doctoral Scholarship (Tertiary Education Commission, New Zealand).

REFERENCES

[1]Cohn, A. G., Hazarika, S. M. Qualitative Spatial Representation and Reasoning : An Overview. *Fundamenta Informaticae* Vol.46, No.1-2, (2001), pp.1-29.

[2]Frank, A. U. Qualitative Spatial Reasoning: Cardinal Directions as an Example. *Internet. J. GIS*, Vol.10, No.3, (1996), pp.269-290.

[3]Guesgen, H. W. Fuzzifying Spatial Relations. *Applying soft computing in defining spatial relations*, Matsakis P. and Sztandera L. (Eds.), Physica-Verlag, Heidelberg, Germany, (2002) , pp.1-16.

[4]Guesgen, H. W. Fuzzy Reasoning About Geographic Regions. *Fuzzy Modeling with Spatial Information for Geographic Problems*, Petry F.E., Robinson V.B., and Cobb M.A. (Eds.), Springer, Berlin, Germany, (2005), pp.1-14.

[5]Allen, J. F. Maintaining Knowledge About Temporal Intervals. *Communications of the ACM*, Vol.26, No.11, (1983) , pp. 832-843.

[6]Gooday, J. M., Cohn, A. G. Conceptual Neighbourhoods in Temporal and Spatial Reasoning. *Proc. ECAI-94*, (1994) , pp. 57-64.

[7]Guesgen, H. W. Spatial Reasoning Based on Allen's Temporal Logic. Technical Report TR-89-049, ICSI, Berkeley, California, 1989

[8]Freksa, C. Temporal Reasoning Based on Semi-Intervals. *Artificial Intelligence*, Vol.54, No.1-2, (1992) , pp.199-227.

[9]Jungert, E. Extended Symbolic Projections as a Knowledge Structure for Spatial Reasoning. *Proc. ICPR*, (1988), pp. 343–351

[10]Evans, T. G. A Heuristic Program to Solve Geometric Analogy Problems. in: M. Minsky (Ed.), Semantic Information Processing. MIT Press, Cambridge, MA, 1968

[11]Peuquet, D. J., Ci-Xiang, Z. An algorithm to determine the directional relationship between arbitarily shaped polygons in the plane. Proc Pattern Recognition Vol.20, No.1, (1987), pp.65-74.

[12]Hong, J.-H. Qualitative Distance and Direction Reasoning in Geographic Space. Ph.D. Thesis, Department of Spatial Information Science and Engineering, University of Maine, Orono, ME, 1994

[13]Hernández, D., Clementini, E., Di Felice, P. Qualitative Distances. in: A. U. Frank and W. Kuhn (Eds.), *COSIT-95*, Semmering, Austria, (1995) , pp. 45-58

[14] Escrig, M. T., Toledo, F. "Qualitative spatial reasoning : theory and practice ; application to robot navigation", IOS Press, Amsterdam, 1998

[15]Guesgen, H. W. "Reasoning About Distance Based on Fuzzy Sets". *Applied Intelligence*, Vol.17, (2002), pp.265-270.

[16]Randell, D. A., Cui, Z., Cohn, A. G. "A Spatial Logic Based on Regions and Connection". *In Pro KR-92*, (1992) , pp.165-176.