Localisation with sketch based input

Matthias Kopczynski, Hannover

Abstract

This article is dealing with an alternative method of localization. It proposes to use sketch based input as a description of the users location or a remote place. Sketches are assumed to be map like drawings in two dimensions that can be assembled on a computer screen for direct interpretation by a processing tool. For the localization process a common representation of the sketch and the reference data must be derived from some basic properties of both data sets. Sketches are an imprecise representation of the thoughts a person has about a situation and emphasizes the relations between objects. This relations are of a topological nature and do not represent a precise geometrie. The properties are leading to a data structure for the search process that is organized as a semantical network, modelling the objects and relations as nodes of a graph. This graph is the foundation on which a search algorithm can be build. It aims on finding a sketched situation in a reference data set by scanning through the reference and finding corresponding nodes of the semantical networks. Therefore a constraint tree search approach is used. It is growing a state tree while exploring potential assignements of nodes. The search space is restricted by constraints that are applied to potential assignements of objects in the sketch and in the reference. Some examples are shown to illustrate this process.

Introduction

For quite some time our society enjoys the improvements of technologie in the sector of telecommunication. In the year 2002 approximately 70% of all citizens of the Federal Republic of Germany were owners of a mobile phone (Statistisches Bundesamt 2003). While only a short time ago the mobile phones were only useful for the exchange of short messages or speech, today an increasing number is extended to universal devices, the difference between computer and mobile phone is disappearing more and more. Industrie and research has realized that the position of a user is of big interest for several services. This services are developed under the term „location based services“ (LBS).

Common methods for position determination of mobile phones rely on an existing net of cells with antennas controlled by a net provider. The position of a user in this setup is a relatively uncertain determined point which can be derived from the cell that is actively communicating with this phone. For many applications this procedure is sufficient but by far not for all. For this reason an increasing number of GPS receivers are integrated into the used devices to allow a much more precise position determination and real navigation solutions. For example in the USA the manufacturers are bound by law to allow a reliable location of mobile phones in cases of emergency.

Especially in the sector of pedestrian navigation this positioning information in combination with data of the environment is interesting. This additional data supports the user in realising where he is and which landmarks he can use for orientation if he wants to follow a planed route (Corona, 2001).

The use of GPS has some drawbacks because it is not operable in densly build areas. This will change with the introduction of GALILEO and the increasing number of available satellites.

But what happens if someone wants to find the current place given by a description or if he is not present at the given place? One possible solution of this problem is to let the user draw a sketch of the place in mind. This sketch is automatically interpreted by the computer and the system tries to find out where the sketched situation is located. This method would also be useful for easy routing or for specifying query areas in a spatial search engine. How this idea can be achieved is outlined in the next sections.

Sketches

At first a definition of the term "sketch" is helpful because the general definition of this term is very comprising. You easily can think of sketches which don't help in finding a place like naturalistic drawings of a sculpture or a drawing which was intended to explain the usage of a machine:

> *"traditionally a rough drawing or painting in which an artist notes down his preliminary ideas for a work that will eventually be realized with greater precision and detail. The term also applies to brief creative pieces that per se may have artistic merit."*
>
> (www.britannica.com, 2003: "sketch")

More suitable is the restriction to the term of a „map sketch":

> *"only approximately true scaled handdrawing without any exactly measurements."*
>
> (translation from www.wissen.de, 2003: "Kartenskizze")

Interesting is now how a sketch is assembled from basic elements. The drawing surface limits the sketch to two dimensions. Lines are drawn on this surface and combined to build a picture. Subject of the drawing is the spatial distribution of objects and their relations. A sample sketch is shown in figure 1. There are several possible ways for expressing the occurence of an object. It ranges from realistical drawings to abstract pictograms. Figure 2 shows a house drawn in three degrees of complexity. On the left a naturalistic symbol is used while in the middle only the shape is preserved. The right symbol will not be recognized as a house without further explanation but can be drawn very fast. Investigations have shown that abstract pictograms are prefered while realistical drawings are rarely

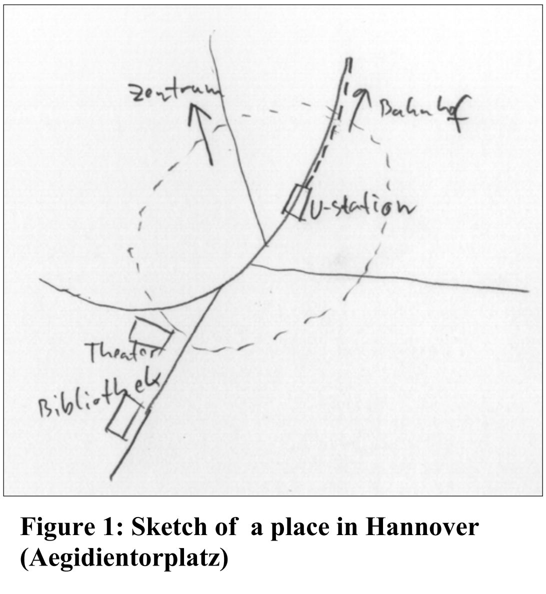


Figure 1: Sketch of a place in Hannover (Aegidientorplatz)

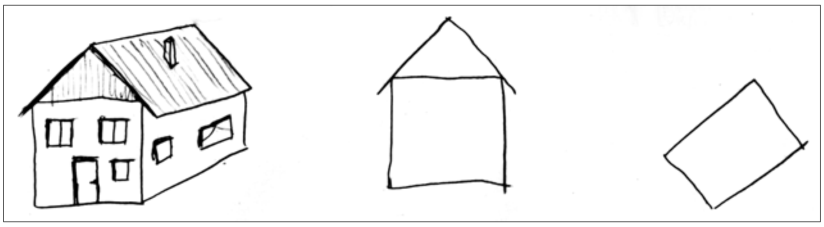


Figure 2: Levels of symbol complexity

used (Blaser, 2000). Every object can be assigned to a class using its appearence or by annotating it with text. All objects are arranged on the drawing surface in a way that their neighbourhood gives hints on the neighbourhood relations in nature. But exceptions, where the relative position in nature can only be reconstructed by interpreting textual annotations, are not uncommon.

The easiest method to create a sketch is to draw it onto a piece of paper. But for the introducted applications it makes more sense to have the possibility of a direct input of a sketch into the computer. PDA's and TablePC's which support comfortable and fast drawing on their screen with a pen and provide a quite realistic feeling of the drawing action are available today.

The electronic drawing surface extends the creative possiblities of a potential user. For example some kind of keyboard input is available for text creation, drawing steps can be undone and frequently used object types can be offered as drag and drop icons of symbols which can easily be moved to the drawing surface. The limiting borders of paper sheets aren't of any importance when using such a device.

Problem description

Suppose the user has assembled a sketch of the situation in his mind and now wants to know where the situation is located.

To find something in general needs something to exist where it can be found. Here, this can be a reference data set that contains all relevant information for comparison with a sketch. The sketch is then a partial set of the reference if one or more solutions of the query exists (Figure 3). As a reference data set the geometrical and semantical contents of a GIS can be used. But because sketches are only a rough approximation of reality, to find a solution for a query is not an easy task. First of all a common representation of the sketch and the reference must be found that makes them comparable and to provide enough information to have a good representation of the data for the query.

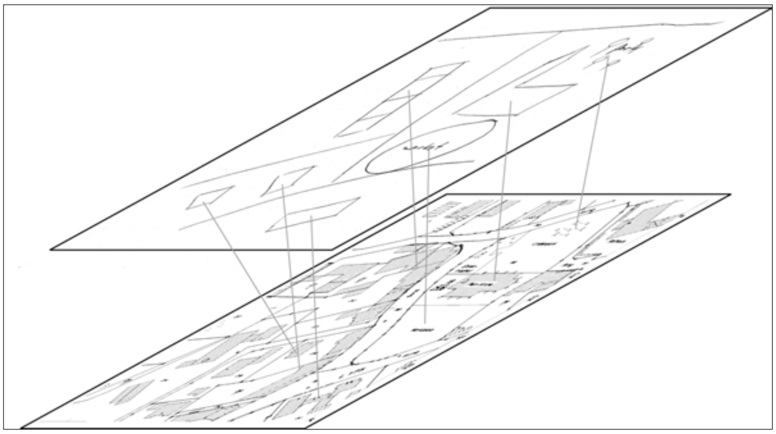


Figure 3: Sketched situation as part of a reference data set

The sketch data and reference data are of very different nature and some processing must be performed to have a comparable representation of both data sets. A sketch must be scanned for contained objects using patterns which describe how an object of a certain type is usually drawn. The patterns can be constructed manually or may be by machine learning from sample sketches. Additional information on the contained objects can be derived from the textual annotations in the sketch.

Following to this a process must derive topological and geometrical relations between objects from both datasets. Those relations then build a comparable basis on which a search algorithm can work. This articles focuses on the construction of a proper data structure and an algorithm being able to find a sketched situation in this data structure.

Data properties and general approach

At first an investigation on the properties of the sketch data and the reference data is needed. Like indicated earlier in this article a sketch is not a geometrical exact description of a situation but a manifestation of the thoughts of someone who has drawn the sketch. Humans think in term which are relatetd with each other with associations (Spitzer, 2000). How someone would place terms on a drawing surface depends on the associations in his mind.

A situation tends to be drawn in a simplified manner. A group of houses e.g. kann be shown by a single symbol, this way area like phenomenons are shrinking to point symbols. But this doesn't apply to every situation, for example when someone wants to express that something is inside or outside an area.

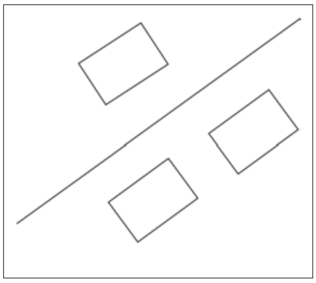


Figure 4: Separation of buildings

For objects with line shapes it is not necessary to copy its natural shape. Much more important are the points where intersections and turn-offs are located. An important information is whether two objects are separated by a line object in between, affecting their neighbourhood relation (Figure 4). Only prominent sharp bends of the road could be from interest in order to identify a road or a section of the road.

Usually a sketch only contains the characteristical objects and relations of a situation. Everything else is left out to win space on the drawing surface and to emphasize important objects. If a sketch describes a route only the important roads and most visible objects which can be used for orientation are drawn. Non visible objects or not conspicuouse enough objects are left out, even if they cover a noticable area in nature. From the cartographical point of view for the reconstruction of a sketch a process is needed which inverses the operations of generalisation.

At the beginning of the interpretation process only an unordered pile of lines, texts and symbols with given coordinates in respect to the drawing surface is given. First aim of the interpretation is to identify objects and to extract topological and geometrical relations between them (Figure 5). An extensible list of examples for the relations contains neighbourhood, distance, direction, clustering, parallelity, orthogonality, line intersections and contained in (Egenhofer, 1993). How the objects can be extracted and how the topological and geometrical relations are calculated is not in the scope of this paper.

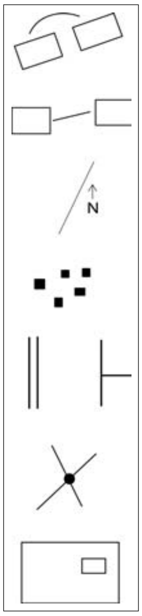


Figure 5: Topological relations

A sketch usually doesn't exceed a certain degree of complexity because it must be kept understandable to a potential sketch reader. The number of objects and relations is also limited, leading to a small data set.

The reference data by contrast is very different in nature. It can get very large, because it must provide all details of the area where the search service is be provided. It is a geometrical exact description of the area, where the coordinates are given in a national or international reference frame. A typical source of such data is a GIS, which contains readily interpreted objects and their attributes. This fact simplifies the conversion into a commom representation because no automatical object detection is needed. However, it is necessary to extract the same topological and geometrical relations between the objects to get a comparable basis.

Data structure

The properties of the data are the basis to derive a suitable data structure which allows to perform fast queries. This section wants to outline the principles of the data organisation providing only a skeleton that must be extended and adjusted for a real implementation.

The basic elements in such a structure must be relations between objects. Every Object has to carry a unique name. Objects in the sketch and in the reference must be named semantically equivalent, making a check for coincidence possible. Additionally, every object is assigned to a class which defines common properties in a set of objects. A case only allowed to appear in sketches is the existence of anonymous objects which don't carry a name but are assigned to a class. Anonymous objects can lead to multiple equivalent solutions in a query, because many objects in the reference can match these objects in the sketch, but they allow the definition of sketched patterns where only some general conditions about the waned area are defined. The same is true for relations between objects. In the sketch relations are always anonymous while in the reference they must have a unique name. The next section uses the term "concept" when referring both to objects and relations.

Concepts must be connected to each other with links to form a statement. To indicate that a house is a neighbour of a road, the concept of the object „house“ must be linked to the relational

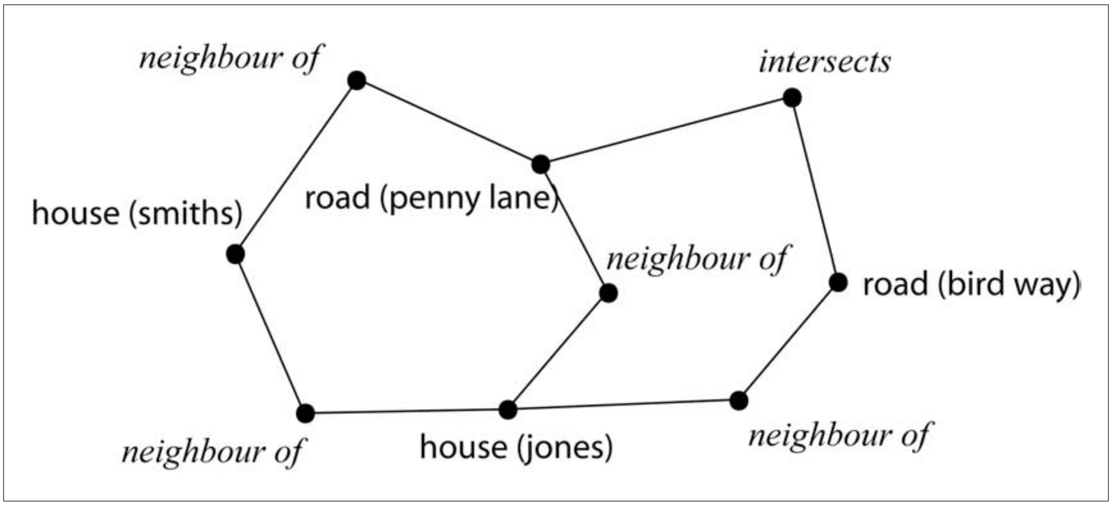


Figure 6: Semantical network of objects and relations. The relations are shown in italic, the objects all have unique names

concept „is neighbour of“ which itself is linked to the concept „road“ as shown in figure 6. The concept „is neighbour of“ shows how the concepts „house“ and „road“ are related. The house and the road would additionally carry unique names allowing to reference them.

A larger number of relations modeled this way referring to the same set of objects is called a semantical network. Usually only the concepts of objects are nodes in a network of this sort and the relations are assigned to the links. Here the relational concepts are also nodes and the links don't carry any properties. This allows to model relations between more than two concepts but more important is that it simplifies a searching algorithm based on this data structure. Note that this requires an alternating use of relational and object concepts.

Algorithm

Now it is possible to search sketched situation which are stored as semantical networks of objects and relations in a large reference data set which is also stored as a semantical network of objects and relations. A well known algorithm for pattern matching tasks, the constraint tree search (CTS) is used and adapted for this special case. This method is searching a state space for valid solutions. Depending on one state new states are created and checked if they are valid. Only valid states are followed, building a state tree with a growing number of branches. This tree is completely searched until one or all valid solution are found. The runtime of the algorithm strongly depends on the number of possible and valid states which result at each branch of the state tree and is always exponential. If this is limiting the practical usage depends on the general conditions of the problem and the data model (Grimson, 1993)

The states of this special process in our case are formed by a list of concept pairs. The basic assumption is that each concept of the sketch data can be assigned to a concept of the reference data to build a valid solution. Two concepts can be assigned one another if their names are identical or if their classes are matching. The last option is only valid if in the sketch data no name for the object was given. At the beginning of the query process two assignable concepts from both data sets must be found to form a valid state. This needs only to be done for one concept of the sketch but has to be done for all concepts of the reference in worst case. For this step the run time is depending linear on the number of concepts in the reference. But it is possible to limit the searching space if additional information is used, e.g. the class of a concept. This can exclude a lot of concepts to be proved in advance.

The algorithm is controlled by a depth search in the graph of the sketch data. All links are followed in one direction from its start node to its end node recursively. Depending on the following tests the depth search can be continued or taken back one step. In every step one link is inspected. For its start node it is always true that an assigned node in the reference exists. This explaines why a first pair of nodes is needed before starting the query algorithm. This property of the start nodes is true for every step of the query.

When the end node in the sketch is not assigned already, all links which originate from the start node of the reference are iterated. Links marked as already used can be ignored. A check is needed for the end node in both the sketch and the reference to prove if an assignment is allowed. If this is the case both links and nodes must be marked as used and an assignment between the end nodes must be established. Then the depth search can be continued by a recursive call with the next step. When returning from the recursion, all this assignments between nodes must be removed and the marks for used links and nodes must be cleared to restore the old state. Returning from the recursion means that the query was truncated by an invalid state or a solution was found and stored. A solution is complete if all links of the sketch are marked as used.

One possible case is that the end node in the sketch is already assigned to an end node in the reference. Then it must be checked if the start node in the reference has a link to this end node. If this happens the depth search can be continued recursively after marking the link as used. After returning from the recursion it must be taken care to restore the old state by removing marks. If no matching link is found in the reference data set, a valid solution does not exist on this track and the actual recursion step is quit. Figure 7 shows how a query works. Steps 1 to 4 are exploring the search space but in step 4 no legal assignment of links is possible. Steps 5 and 6 do backtracking until an unexplored state can be reached. The final solutions is found after exploring alternative paths in step 9. Note that this is not a unique solutions and further exploring of the state space would find more valid assignments.

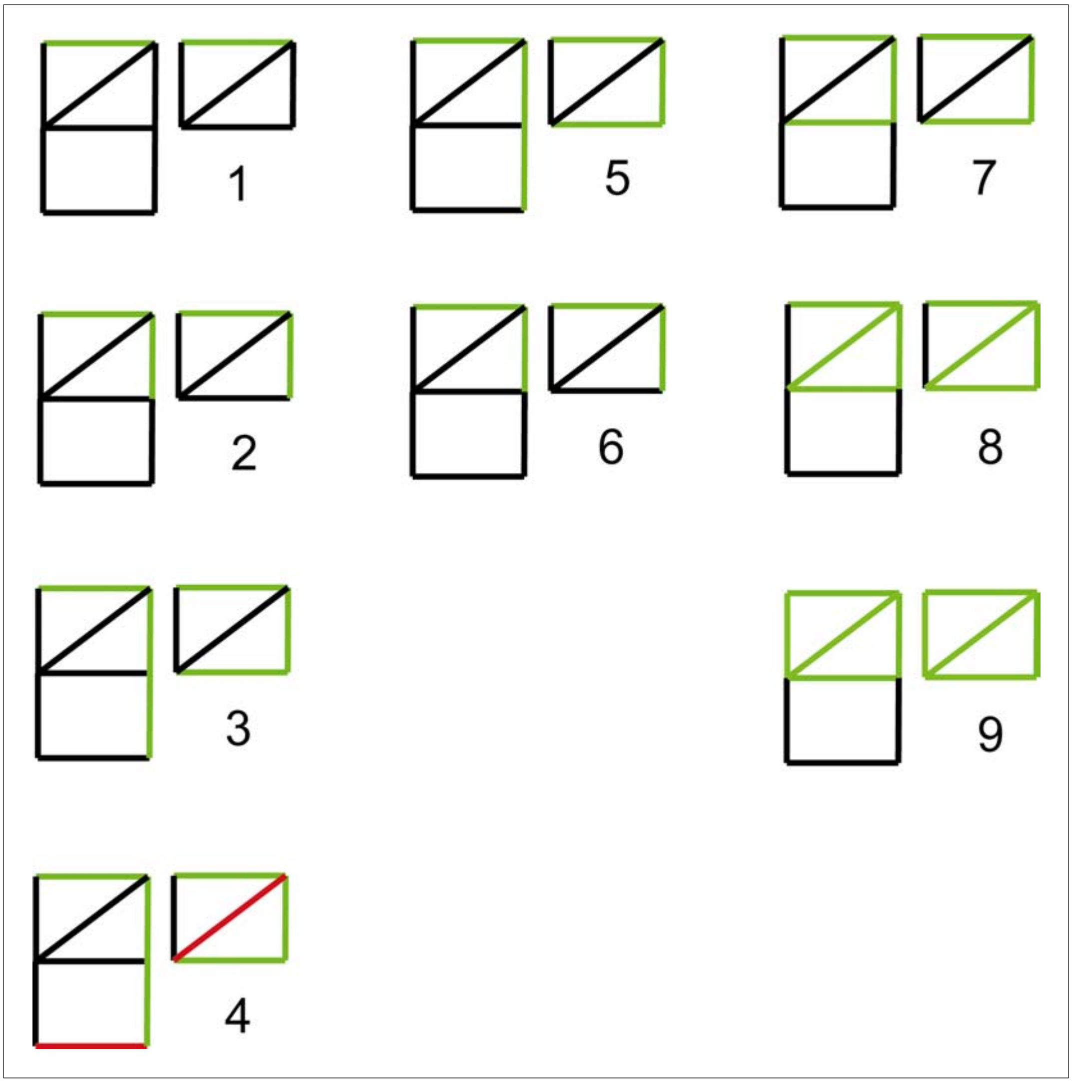
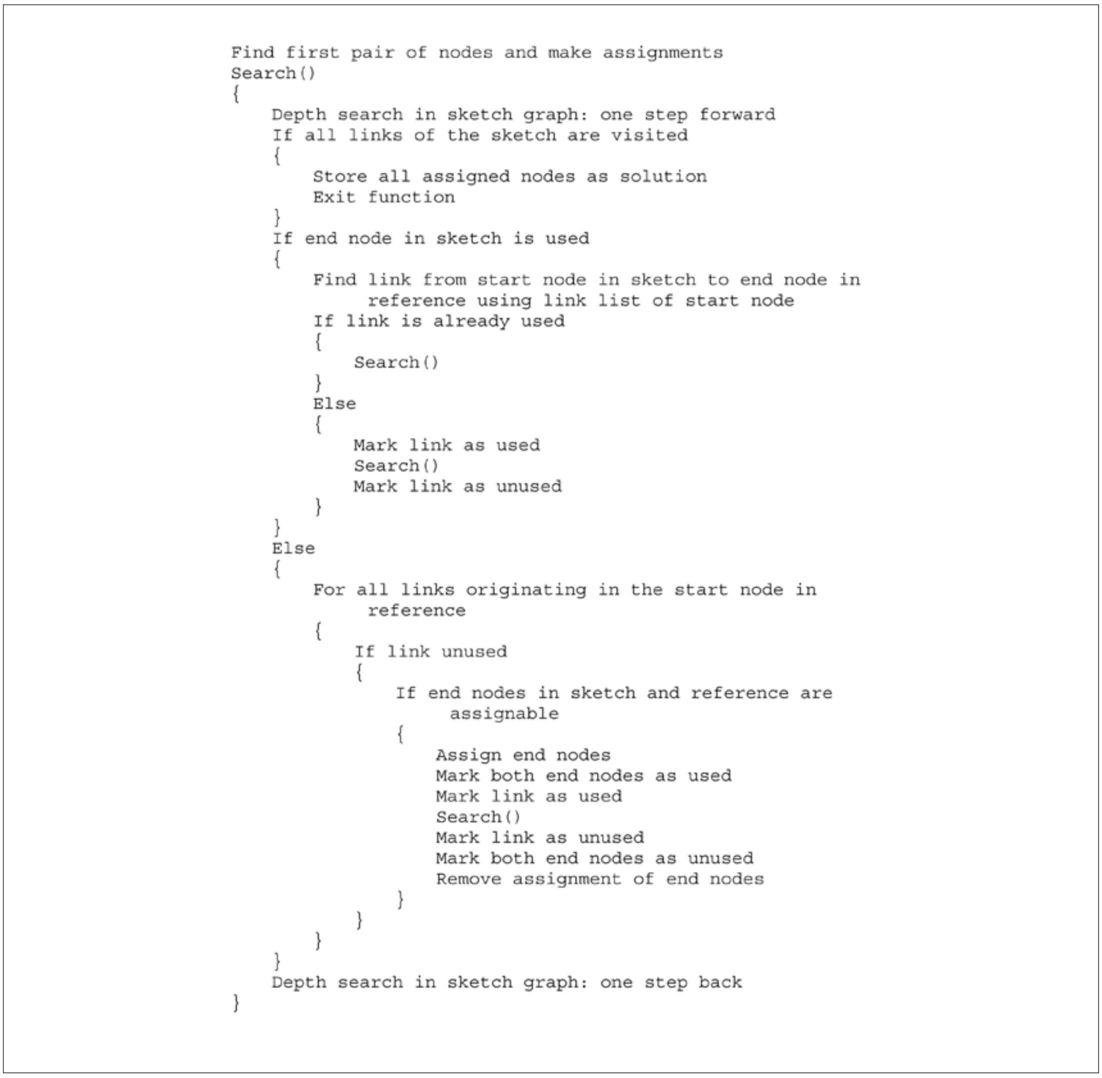


Figure 7: Iterating through the state space. The left network in every step is the reference data set. The right network is always the sketch network.

The algorithm can be written in pseudocode as shown in figure 8.



```
Find first pair of nodes and make assignments
Search()
{
    Depth search in sketch graph: one step forward
    If all links of the sketch are visited
    {
        Store all assigned nodes as solution
        Exit function
    }
    If end node in sketch is used
    {
        Find link from start node in sketch to end node in
             reference using link list of start node
        If link is already used
        {
            Search()
        }
        Else
        {
            Mark link as used
            Search()
            Mark link as unused
        }
    }
    Else
    {
        For all links originating in the start node in
             reference
        {
            If link unused
            {
                If end nodes in sketch and reference are
                     assignable
                {
                    Assign end nodes
                    Mark both end nodes as used
                    Mark link as used
                    Search()
                    Mark link as unused
                    Mark both end nodes as unused
                    Remove assignment of end nodes
                }
            }
        }
    }
    Depth search in sketch graph: one step back
}
```

Figure 8: Pseudocode for the query algorithm

The runtime of the algorithm strongly depends on the structure of the graphs building the semantical networks. In general it can be assumed that the sketch graph is small which limits the number of potential links to follow and limits the needed query time. Important is further if it the graphs are dense or thin. Thin graphs provide less possible tracks to follow in each node what speeds up the query. Are the links connecting only local concepts, the algorithm searches only in a small amount of nodes of the reference. Nodes far away can then be assumed to be unimportant for the query and are not visited during the searching process. All this must be taken care of when implementing a data model for the semantical networks.

Results and further work

The shown algorithm has been implemented in a program. Some small tests with simulated road data showed already a reliable detection of patterns in a reference data set. Only connections between turn-offs and intersections were modeled. Only the classes intersection, connection, same sided turn-off and opposite sided turn-off were used. The tested area and patterns were small so that no final conclusion regarding the practical run time and reliability can be drawn (Figure 9). But an analysis of the visited paths supports the assumption that many of the combinations are dropped as invalid and the search tree is strongly pruned.

More precise answers will be available when an automatic conversion of data sets into semantical networks is possible. This will also allow more complex models with more classes, giving a more realistic view of the world.

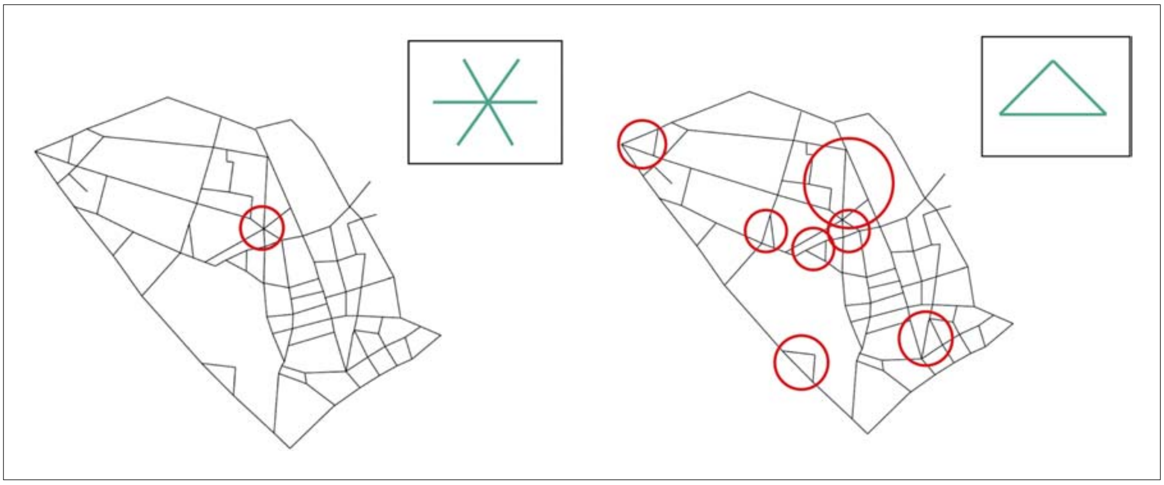


Figure 9: Sample data. The left pattern is unique but the right pattern occurs multiple times

At this point I don't want to neglect the development of a sketch input tool that will provide a web interface for the easy construction of sketches on a computer. It will be possible to produce sketches of realistic situations and use them as input to the query. This will answer the question if a realistic sketch can limit the number of solutions to a useful amount. One result of the first tests was that general enough patterns returned solutions in the amount of the reference data set. It would be helpful to detect this cases before the searching process to save searching times and to prompt the user for a more precise sketch.

The sketching tool will be used in the SPIRIT project as a contribution to the multi modal user interface of a spatial search engine. The search engine will allow to use spatial query terms and select only those web site which can be located in an area covered by these terms. Several different interfaces, including the sketching interface, are developed to enter the spatial query terms and use them for a query (Jones, 2002)

References

Blaser, A. D. (2000): **SKETCHING SPATIAL QUERIES, The Graduate School, University of Maine**

Blaser, A.D., Sester, M., Egenhofer, M. J. (2000): **Visualization in an early stage of the problem-solving process in GIS, Computer and Geosciences 26, p. 57-66 , Elsevier Science Ltd.**

Bollmann, J., Koch, W. G., Hrsg. (2001): **Lexikon der Kartographie und Geomatik, Spektrum Akademischer Verlag Heidelberg, Berlin – (Topics: "Skizze", "Kartenskizze")**

Corona, B.; Winter, S. (2001): **Guidance of Car Drivers and Pedestrians, Technical Report, Institute for Geoinformation, Technical University Vienna, Austria.**

Egenhofer, M. (1993): **A Model for Detailed Binary Topological Relationships, Geomatica 47 (3&4), p. 261-273, 1993.**

Encyclopedia Britannica (2003): **Internet: http://www.britannica.com - (Topic: "Sketch")**

Grimson, W. E. L., Lozano-Pérez, T., White, S. J., Noble, N. (1993): **Recognizing 3D Objects Using Constraint Search, Three-Dimensional Object Recognition Systems, A.K. Jain and P.J. Flynn (Editors), Elsevier Science Publishers B.V.**

Jones, C.B. et al (2002): **Spatial Information Retrieval and Geographical Ontologies: An Overview of the SPIRIT project, 'SIGIR 2002: Proceedings of the 25th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval August 11-15, 2002, pp.387 - 388., ACM Press, Tampere, Finland**

Spitzer, M. (2002): **Lernen, Spektrum Akademischer Verlag Heidelberg, Berlin**

Statistisches Bundesamt Deutschland (2003): **Ausstattung privater Haushalte mit Informations- und Kommunikationstechnik in Deutschland, Internet: http://www.destatis.de/basis/d/evs/budtab2.htm**

www.wissen.de (2003): **Internet: http://www.wissen.de - (Topic: "Kartenskizze")**