Automatic understanding of sketch maps

Ph. D. Proposal for The International Research Training Group on Semantic Integration of Geospatial Information

Klaus Broelemann

February 1, 2011

Contents

1 Introduction

During the last years there has been a huge increase in the number of users of Geographic Information System (GIS) as well as the capabilities of these systems. While the increase in capabilities often leads to a more complex user interface, many of the new users do not have strong GIS expertise. Thus there is a need for more intuitive and natural interfaces. This need can be met by sketch maps [1], which are hand-drawn maps that show spatial data in a schematic way.

Sketch maps are a natural way to share spatial information between humans. Unfortunately, sketch maps are not designed to be easily read by electronic systems. To use sketch maps for interaction with a GIS the system has to be able to understand sketch maps. Therefore, efforts in object recognition and semantic integration have to be made. As the user does not want to "explain" the sketch map to the GIS, the GIS has to understand sketch maps automatically.

1.1 Use Case

Use cases for sketch maps and automatic processing of sketch maps can be found in the common scenario of the International Research Training Group on Semantic Integration of Geospatial Information (IRTG)[1]. Main concern of this scenario is the disaster management after earthquakes. The earthquake that struck L'Aquila, Italy, in April 2009 is used as example for this scenario.

One essential demand for disaster management is the need for up-to-date maps of the affected region [2]. Due to changes caused by destruction, such maps are not available directly after an earthquake. In this situation, people who have been in affected areas can provide new information. A good way to communicate this spatial information about the destroyed areas are sketch maps. Especially, automatic processing of sketch maps can help to provide up-to-date maps for the disaster management. This can be done by integrating the information of sketch maps into existing metric maps or by combining several sketch maps to new maps. To save time and manpower, the processing should be done automatically, or at least semi-automatically.

In a disaster scenario, the support by special technology can not be guaranteed for people spread over the area. In addition, earthquakes cause a breakdown of infrastructure, including power supply and internet. Thus, if possible, the sketching should not depend on technical input devices. Papers and pens are more likely to be available and, hence, better adapted for drawing sketch maps in disaster situations.

On the contrary, automatic processing of sketch maps depends on technical devices. As mobile phones with cameras are available for everyone, they can provide a possibility for sketch processing in disaster scenarios. Since the communication between distributed people is based on mobile phones[2], a fast recovery of the necessary infrastructure is plausible. This will enable the exchange of sketch maps from different parts of the

[1] For more information about the IRTG common scenario see the IRTG wiki at `http://irtg-wiki.uni-muenster.de/mediawiki/index.php/Common_scenario`.

[2] According to Zock [3], people in Haiti used SMS to report problems.

affected area. Due to limited bandwidth a processing on the mobile phone and thus a reduction of data is beneficial.

1.2 Overview

For my Ph. D. research I propose to develop algorithms for automatic offline semantic recognition and integration of objects in images of hand-drawn sketch maps on mobile devices. The goal of the recognition is to transform a low-level pixel representation into a high-level semantically enabled object representation. These objects are representations of geographic features like buildings, trees, lakes or streets.

For my work, I will use images of sketch maps as input and process these images. Other work that starts with more abstract representations of sketch map like the work of Jia Wang and Malumbo Chipofya can use the output from my methods as input. Thus, in a sketch processing chain my work will be the first part.

In the following paragraphs, I will give a brief overview of the aspects that characterize my work.

Sketch Maps

According to Davis [4], there is a difference between diagrams and sketches. While diagrams are formal representations, sketches are informal figures. In the context of maps there is a difference between metric maps and sketch maps. Metric maps are a formal image of the world, originating from a projection of the world. In contrary, sketch maps are representations of cognitive maps [5].

Hand-drawn

There are two ways of creating sketch maps. One is to draw the sketch map by hand, the other one is to use a computer program for drawing. The difference between both is not that one of them exists on paper and the other one as a digital file: Hand-drawing can also be done on a tablet PC using a pen and digital maps can also be printed. The main difference is the accuracy of the drawing style. Sketch maps drawn at the computer have smooth lines and computer-rendered annotations. Hand-drawn lines are seldom smooth and hand-written text differs from person to person and even from time to time for the same person.

Offline Processing

Sketch recognition can be classified into online and offline recognition. While online recognition is done with the knowledge of the drawing process, offline recognition works on a complete sketch. Normally, digital pen-based input devices like tablet PCs and PDAs lead to an online recognition, whereas images, like scanned or photographed sketch maps lead to an offline recognition. Technically, online recognition can make use of information like segmentation into strokes, order of the strokes and even speed of the pen. For offline recognition, this information is not available and the recognition has to rely on pixel data. Most of current recognition techniques are online recognition techniques. In contrary, I will perform offline recognition. This will enable the use of sketch maps in situations without digital devices for pen-based input.

Mobile Implementation

One goal of my research is to find methods that can be run on a mobile device like smartphones. By doing so, sketch map processing will be supported for situations, where no other digital device is available. That can be the case of spontaneously asking for the way to some destination, but also in above described disaster situation, where rescue squads can not carry big devices while entering crisis areas.

To be able to work on smartphones, algorithms have to deal with images taken by the camera of a smartphone. Although the quality of smartphone cameras is increasing, there are still problems like inhomogeneous illumination.

Compared to PCs, smartphones have low memory and computation resources, which is a drawback for the development of mobile algorithms. One way to deal with the restricted resources is to transfer the image to a server and process the sketch on the server. In situations with stable network connections, this is a good way to combine mobile devices with computational expensive algorithms. For crisis situations, this is not applicable, since many people try to call someone and the disaster might also destroy parts of the infrastructure. Thus, I propose the development of algorithms that can completely be run on smartphones.

Integration

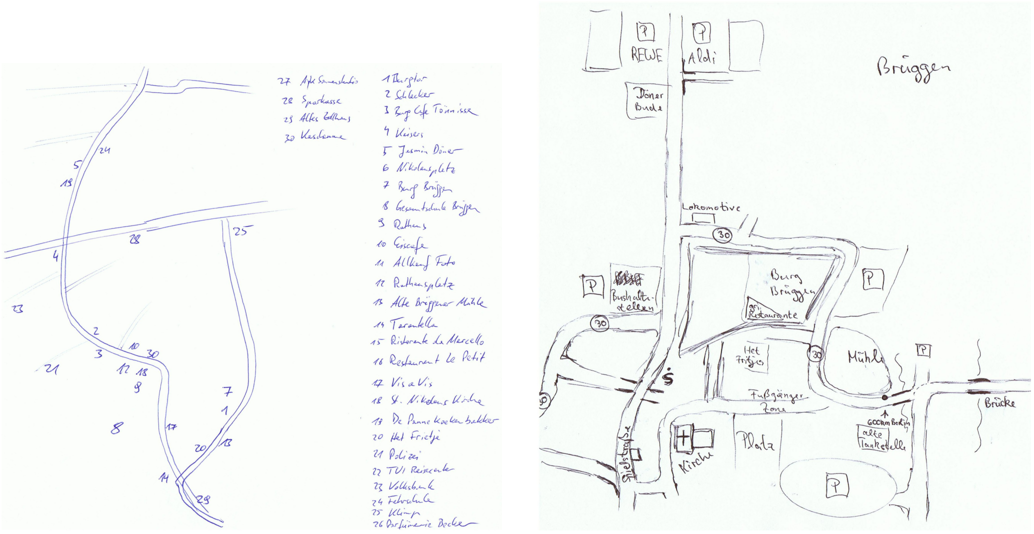
One goal of sketch map understanding is to integrate the sketched information. For the integration, two maps have to be aligned.

For my work, I will investigate the capability of pattern and graph matching to align sketch maps. This distinguishes my work from other work that also uses cognitive knowledge about sketching habits for alignment.

Supported Maps

There are different types of sketch maps, depending on the target application of the maps. Two of these types are route maps and survey maps.

> **Route maps** These maps are intended to display a route from one location to another. Hence route maps especially show landmarks “that can be used for orientation purposes, or [. . .] that prime upcoming decision points” (Golledge [6], p. 10). An example for a route map is displayed in Fig. 1(a).
>
> One use case for a route map can be found in the IRTG earthquake scenario: hand drawn sketch maps can display routes to people in need. In addition such maps can provide helping units with information about secure routes.



(a) Route Map (b) Survey Map

Figure 1: Examples for sketch maps

Survey maps In contrast to route maps, survey maps do not represent a single route but give an overview of a region of interest. Which objects are displayed depends on the purpose of the map and the drawer's knowledge of the region. Fig. 1(b) displays an example of a survey map.

One way to use survey maps is to register[3] sketch maps to metric maps and in doing so integrate the sketched information into the metric maps. It is also possible to combine several sketch maps to build a map for regions that are hardly supported by metric maps. The result of both can be used in the earthquake scenario to provide a survey of the current situation for helping units.

Both types of maps can be drawn in similar ways, but there are crucial differences for my work. In order to be able to align maps by graph matching, there has to be a graph of sufficient size for the matching. Since route maps often contain just a few streets, I will concentrate my work on survey maps.

2 Related Work

In this section, I will give an overview over related work. After a brief description of existing sketching applications, I will concentrate on work related to the subgoals for unterstanding of sketch maps: Text detection and optical character recognition (OCR), vectorization, structural recognition, and semantic recognition and integration.

[3](Image) registration is the process of transforming different images of the same scene into a common reference system. In this context a metric map defines the reference system and the registration transforms the sketch maps into this system.

2.1 Sketching Applications

It has been shown [1] that sketch maps are a natural way for untrained people to interact with GIS. To support an easy-to-use interface, several ideas, methods and systems to interact with sketch maps have been presented. One of them is "query-by-sketch" [7, 8, 9, 10, 11, 12], a way to formulate spatial queries to a database by sketching the desired constellation of spatial objects. Existing systems using query-by-sketch are still restricted to pen-based input methods. Forbus *et al.* [13] presented a system called CogSketch for sketch understanding, based on the earlier developed nuSketch [14]. Both systems require manual labeling of objects instead of using sketch recognition techniques. They even depend on user input to identify the start and the end of drawing one glyph.

2.2 Text Detection and Optical Character Recognition

As sketch maps often contain text[4] automatic recognition systems benefit from text interpretation. Therefore, both text detection and recognition should be used. The former one comprises methods that find text in images, segment images into text and non-text parts or determine text attributes like orientation and skew. The latter one deals with pixel representations of text in order to get a string representation and thus understands the image of text.

In the field of automatic map understanding, there are two applications for text detection: used as a preprocessing step for structural recognition, text detection can filter out text pixel and, thus, reduce distortions for object recognition. In combination with character recognition, text detection methods can be used to combine objects with annotations. Text detection has been an active field of research for the last twenty years, approaching more and more complex scenes. First text detection methods were designed to distinguish between text blocks and rectangle images [15, 16]. Since sketch maps mix graphics and text, these methods are not applicable for sketch maps. Later, research led to algorithms that can find text in images and graphics. According to Zhao [17] there are four main categories of text detection methods: connected component analysis, edge bases techniques, texture based techniques, and frequency based techniques.

A well-known method was proposed by Fletcher and Kasturi [18], using connected components for text detection. The disadvantage of this algorithm is the incapacity to detect text that is connected to the background. Tombre *et al.* [19] improved this algorithm to detect text that is partially connected to graphic elements. Li *et al.* [20] presented an algorithm to detect and recognize text in topographic maps. The algorithm is even able to detect text that intersects with map elements, but is restricted to trained fonts.

OCR has been subject of research for over forty years [21]. Thus, many algorithms exist to recognize a wide range of different types of text. Relevant for automatic recognition of sketch map objects are algorithms to detect handwritten text.

There are many OCR methods for handwritten text [22] that can be applied to text in sketch maps. As many of these methods depend on assumptions like straight text

[4]According to Blaser[9] 60% of objects drawn of a survey are annotated by text.

lines, the methods are not expected to recognize all text in sketch maps.

2.3 Vectorization

Vectorization methods have a long history in computer vision. There goal is the transformation of pixel based images into sets of parametric curves, such as straight lines, elliptic arcs or Bézier curves. One aspect of vectorization algorithms is the class of curves that can be detected.

Many vectorization methods are based on a preceding thinning. Thinning is a well-known technique in computer vision, which reduces lines to the width of one pixel. A survey on different thinning methods was given by Lam *et al.* [23]. Methods that use thinning typically contain three main parts [24, 25]: thinning, segmentation and parameter estimation. Due to the thinning, it is possible to trace lines from junction to junction. The segmentation part divides these thinned lines into segments that can be represented by one parameterized curve. For each segment, the parameter estimation finds parameters to approximate the segment by a curve. A comparison of methods that use thinning and methods that do not use thinning was done by Tombre and Tabbone [26].

Hilaire and Tombre [25] presented a vectorization method that initially separates the image into layers of lines of different thickness before thinning each layer. Though the approach is theoretically able to detect different types of curves, the authors only applied it to straight lines and circular arcs. Other curves can only be detected with high computational costs. Other methods detect Bézier curves. One algorithm presented by Chang and Yan [27] is explicitly designed for hand-drawn graphics. Another one from Masood and Sarfraz [28] promises accurate detection. Ferri and Griffoni [29] proposed as vectorization method for sketch objects that can detect closed polylines.

2.4 Structural Recognition

The aim of structural recognition for sketch maps is to find objects within images of sketch maps and to characterize by attributes like their shape, size, curvature of their borders or their location. A lot of work has been done in computer vision on structural recognition in general and for particular applications. This section will give a brief overview over work related to structural recognition of sketch map objects.

Sketch Recognition

A lot of work has been done in the field of online sketch recognition. Beside domain specific systems, there has been some work on multi-domain solutions. Hammond and Davis presented with LADDER [30] a well-known language to define domain specific sketch recognition systems. Later, Paulson and Hammond [31] developed a system to recognize simple primitives that can be combined into more complex objects. SketchML, another language for sketch recognition, was provided by Avola *et al.* [32]. Most online sketch recognition relies on the assumption that the drawing of one object is finished before the drawing of the next one is started. To overcome this drawback, Hammond and Davis [33] presented a method to recognize interspersed sketches. AgentSketch, a framework for agent-based sketch recognition has been developed by Casella *et al.* [34]. This framework is able to deal with both online and offline recognition.

While sketch recognition focuses on online recognition, nearly no work is done in the field of offline recognition. Notowidigdo and Miller [35] developed a system for offline sketch interpretation that is able to recognize rectangles, diamonds, circles, and arrowheads. Kara *et al.* [36] presented a more flexible system that can be trained to detected sketched symbols. For sketch maps, this can be used to detect symbols for trees or To close the gap between online and offline recognition, Qiao and Yasuhara [37] presented a method to gain dynamic information from handwritten images.

Road Recognition

One particular case for structural recognition is the recognition of roads. Roads can be found in several types of images, e. g. aerial images, satellite images and also different types of maps. Different approaches have been made to find roads in such images.

One way to find roads is *road tracking.* The base idea is to find starting points within streets. Beginning with these points road tracking algorithms find more points, following the road and thus construct polylines within streets. One of the first road tracking algorithms was designed by Groch [38] for aerial images. This method depends on differences in gray level values and only detects streets of a given width. A newer algorithm using similar techniques was presented by Hu *et al.* [39]. Both algorithms are designed to detect roads in aerial and satellite images and depend on some assumption, like fixed width of streets, that can not be made for sketch maps. Thus, both algorithms ca not directly be applied to sketch maps.

2.5 Semantic Recognition and Integration

To understand sketch maps, semantic information is needed. The goal of semantic recognition is to detect the semantics of structures.

One basis for semantic recognition is the knowledge about sketching habits, especially the knowledge, which semantics can be represented by which structures. Blaser [9, 40] presented surveys on peoples sketching habits. He got quantitative results, which can be useful for semantic recognition.

Semantic recognition can not deal with isolated objects but has to take in account the relations between the objects. As sketch maps are distorted in several ways, a promising approach is to use topological relations [41, 42].

Semantic recognition of sketch maps has to deal with a lot of possibilities and uncertainties. A promising approach is to use probabilistic recognition. One work for probabilistic understanding of sketches has been done by Alvarado and Davis [43]. They dynamically constructed Bayes networks using information about strokes. This work can not be applied for offline sketch recognition, since the stroke information is only available for pen-based input.

2.6 Mobile Computing

With the upcoming of smartphones, mobile computing became an fast-growing field of research. The challenge is to work on devices with restricted resources that in addition do not interact by mouse and keyboard. Jiang *et al.* [44] presented a collection of methods and applications for mobile devices.

In the field of mobile sketch map processing, Carduff and Egenhofer [45] proposed a system for mobile query-by-sketch. In this system, the sketch processing will be done on a server, instead of processing the sketch on the mobile phone.

3 Research Questions

How can structures be recognized in images of sketch maps?

Sketch maps contain lots of lines that are grouped together to structures like rectangles, circles, symbols or parallel lines. Thus, structures are geometric elements of sketch maps without semantic interpretation. Recognizing structures in images of sketch maps means to group the lines together and to detect which kind of structure they build. Beside shapes like rectangles and ovals, this also includes finding line networks that can later be interpreted as streets, rivers and railways.

One part of my research is to find methods for recognizing structures in sketch maps. In doing so, I have to deal with typical drawing problems like line intersections and disconnected lines that belong to the same structure.

How can semantic interpretation be inferred from detected structures and be used for integration?

Structures are sketch map elements without any semantics. For sketch understanding and integration it is essential to know the semantics of objects. Since the same structure can have several meanings, there is no well-defined mapping from structures to their semantics. To develop methods for inferring the semantics of objects will be one goal of my research. Since there might be some incorrectly recognized structures, these methods for semantic recognition should be robust up to a certain degree of errors.

Part of my research about semantic interpretation will be to investigate how detailed the semantics of objects can be recognized. For example, an object can be recognized just as a building or it might be possible to recognize it as a hospital. Furthermore, the semantics of objects will not only be recognized, but also integrated with data from other maps. This includes the registration of sketch maps against other maps.

Does semantic recognition need modifications for use on mobile devices?

One of the reasons for drawing sketch maps is the absence of computers that provide access to GIS. In contrast to computers mobile devices are available at nearly every time. These devices can also provide access to GIS, but can not provide the comfort. Due to

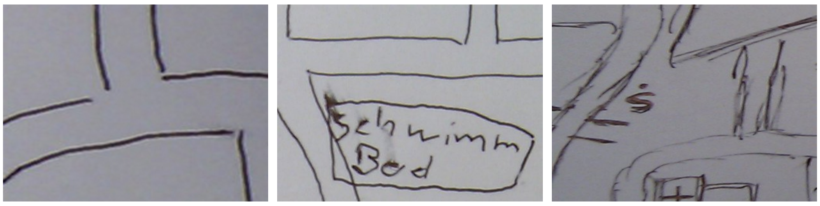


Figure 2: Examples for imprecision in drawings. From left to right: gaps between lines, intersecting objects, and distortions of lines

small display sizes of mobile devices, user interaction with GIS is limited. Hence mobile devices would benefit from the ability to understand hand drawn sketch maps.

There are two ways for mobile devices to understand sketch maps. On one hand the mobile device can send the image of a sketch map to a server. The server processes the sketch map and returns the results to the mobile device. On the other hand the mobile device can recognize the objects of the sketch map on its own. Due to the limited resources of mobile devices the second way needs more efficient algorithms. The advantage is a reduction in amount of data that is to be sent. The description of sketch maps via objects is much more compact than the description via pixels of an image. Thus processing sketch map images on mobile devices reduces the bandwidth that has to be used for communication with a server.

4 Approach

4.1 Challenges

Imprecise drawings for vectorization and structural recognition

The main intention of sketch maps is to communicate geographic information between humans. The focus of sketch maps is on the schematic illustration and not on a precise drawing. In addition sketch maps are often drawn free-hand without the use of rulers. Both the schematic focus and the free-hand drawings, lead to imprecision. At this point the kind of imprecision that is challenging for object recognition is the imprecision at the basic drawing level. There are other kinds of imprecision like imprecision in schematization and imprecision by fuzzy concepts.

Typical imprecision in drawings are disjointed lines that are intended to be connected, distortion of lines, line connections by mistake and overlapping objects. In addition people might make mistakes while drawing and correct them afterwards, like drawing small streets they had forgotten before. Figure 2 shows some examples for imprecise drawings.

Different drawing styles for structural recognition

There is no drawing standard for sketch maps. This leads to several different ways to draw the objects on sketch maps. For example, streets can be drawn single lined, but

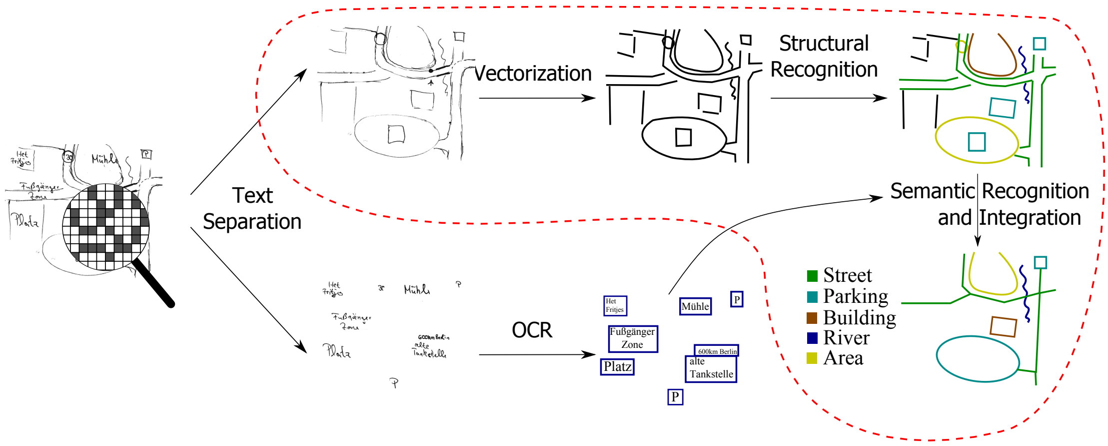


Figure 3: Proposed steps for automatic recognition of sketch map objects: Text Separation, Vectorization, Structural Recognition, Semantic Recognition and OCR. Focus of my work are the red framed parts.

also double lined. Another example are landmarks like trees and buildings that can be symbols, but also labeled shapes. There are two ways to deal with different drawing styles: on one hand the user's input can be restricted to one drawing style and on the other hand the algorithms have to be designed flexible and robust enough to deal with different drawing styles. Although the second way leads to more complexity the developed algorithms should restrict the drawing style as little as possible.

Semantic ambiguity for semantic recognition and integration

There is no well-defined mapping that maps the symbols to the semantics of these objects. For example, two parallel lines can be a street, but also a river. Arrows can mean a direction, but can also connect a label with an object. Rectangles can be buildings, but also parking places. Thus, semantic recognition and integration has to deal a huge number of possible semantic interpretations.

Mobile implementation

There are two challenges for algorithms on mobile devices. The first one are restricted memory and computational power compared to desktop computers. This restrictions leads to a need of efficient algorithms. The second challenge concerns image-based algorithms. Many smartphones do not have flashlight and a low or medium quality of the camera. Thus, algorithms for mobile phones have to deal with inhomogeneous illumination and noise.

4.2 Methodology

The automatic recognition of sketch maps objects can not be done in a single step. I propose to divide the task in substeps as follows: Text Separation, Vectorization, Structural Recognition, Semantic Recognition and Integration, and OCR. For my research, I will focus on the three steps Vectorization, Structural Recognition, and Semantic Recognition and Integration. Figure 3 gives an overview of the steps and the dependencies between them.

Text Separation

Written annotations are widely used elements of hand-drawn sketch maps [9, 14]. The annotations can be connected to other objects and even intersect with them. These intersecting annotations disturb the object recognition and makes it more difficult. Even disconnected text might lead to false recognition when treated as graphical object. To reduce the impact of text on the recognition of objects, I use an initial separation of text and graphical elements.

Besides the reduced distortion, the separation enables a different treatment of text and graphics. While the recognition of graphical objects focuses on shapes and lines, text recognition is done by OCR.

As mentioned in section 2, there are several methods for the separation of text and graphics that are applicable in this context.

Thus, the development of a new method for text/graphics separation will not be subject of my research. Instead, I will use a state-of-the-art algorithm that can deal with text in sketch maps. Such an algorithm has to find hand-drawn text, even if the text intersects with graphic elements. One algorithm that is able to do so was presented by Tombre *et al.* [19].

Vectorization

The result of the text/graphics separation step are two pixel-based images, one containing the text elements, the other one the graphic elements. This pixel-based representation is not suitable for the detection of structures like streets. Thus the graphics images will be transformed into a more abstract representation. One way to do so is to vectorize the image, which will provide elements like lines and arcs. These elements contain structural information and hence are useful for further recognition steps.

As mentioned in section 2, there are many existing algorithms for vectorization. Many of these algorithms like the one presented by Hilaire and Tombre [25] need accurate lines and are not suitable for hand-drawn lines. For the processing of sketch maps, I follow the ideas of two algorithms: Ferri and Griffone [29] presented an algorithm that is able to find closed contours in sketch maps and represents them as polylines. This algorithm can also be used to find objects. The other algorithm was developed by Chang and Yan in order to find Bézier curves [27]. Thus, it can be used to vectorize round structures in sketch maps.

Still there remain open problems for my work: due to some drawing styles the vectorization might result in a high splitting of lines into several segments. One example is given in Fig. 2. A reduction of the splittings leads also to a smoothing of the detected curves and, as a result of the smoothing, to a loss of information that might be useful, e. g. to detect wiggly lines. Other lines that curse problems are dotted lines, that vectorization methods detect as a number of single dots, but should be represented as one line.

Structural Recognition

The aim of the structural object recognition is to find and describe structures within the image of a sketch map. For this step, only geometric attributes of the structures are considered without any semantics in the context of maps. For example, recognized rectangles can be described by orientation and size. But it might also be possible to find other attributes, like the curvature at the corners.

The first step of structural recognition is to segment the curves from the vectorization step into structures. This can easily be done for perfect[5] sketch maps, which have all lines of a structure connected and all structures disconnected. As soon as an algorithm has to take account of the above mentioned problems with hand-drawn sketches, the recognition becomes more challenging: lines of intersecting structures have to be assigned to one structure, gaps between lines have to filled, connected structures can have common lines, and so on.

I propose to identify these problems and to find specific methods to solve them. A basic approach how to deal with intersecting objects can be found at by Hilaire and Tombre [25]. There a method is proposed to identify lines that are fragmented be intersecting or connected lines. Based on this method, it might be possible to develop an algorithm that detects intersection between objects, like shown in the second image of Fig. 2. To identify gaps between two lines, I propose to consider the directions of lines that have nearby endpoints.

The next step of structural recognition is to identify the type of structure. I propose to use pattern recognition methods [46] using properties of the structure as features. There are many possible properties like the number and distribution of corners, angles at the corners, curvature of the structure and its compactness. Though there are more possible properties, I will start with these for recognition.

With mobile implementations in mind, I will also have a look an computational efficiency of the developed algorithm.

Semantic Recognition and Integration

Structural recognition just finds geometrical structures. These structures have to be enhanced with semantic information to be of use for sketch map understanding. Semantic recognition has the goal to detect the objects in the sketch map, that means to find out

[5] In the sense of recognition.

what the structures represent. These objects can then be integrated into other maps like a metric map.

Since structures can have different meanings, it is crucial for semantic recognition not only to look at isolated structures, but also to examine the relations between structures. To derive the semantics of objects from their relation to each other, rules have to be found that allow reasoning from structures and their relations. Though I propose to use relations between objects inspired by previously proposed topological relations for spatial objects [9, 42, 41], the usage is completely different from the one of Blaser [9]. In Query-By-Sketch, Blaser and Egenehofer proposed to use topological relations to find a constellation with similar relations in a database. His analysis was limited to spatial analysis only, since no semantic information was available. The goal of my work will be to enhance spatial information by detecting the underlying semantics of objects. The results could for example be used to enhance the database query methods presented by Blaser.

To define the rules, which semantics can be derived from which constellation of structures and relation between them, I propose to use an ontology [47, 48, 49] for the sketching domain. A good basis for designing such an ontology are the quantitative results of Blaser's survey on sketching habits [9, 40].

Due to the huge amount of uncertainties for semantic recognition, like different possible interpretations for structures and relations, I propose to perform a probabilistic recognition. There are different ways to deal with such networks of uncertain interpretations like Bayesian networks [50]. Alvarado and Davis [43] presented a method to use Bayes networks for sketch understanding. As many other sketch understanding systems, their method relies on pen-based inputs and thus can not be applied for offline sketch understanding.

For the alignment of sketch maps and metric maps, I propose to use graph matching methods [51] based on the street graph. The graph matching has to be able to deal with typical distortions of sketch maps, like missing streets and wrong length and angles. Since the street network often is the most complex element in survey maps, it could be enough to use the connections between streets to align sketch maps and metric maps.

Like the structural recognition, semantic recognition and integration should run on mobile devices. Therefore, I will examine the efficiency of the used algorithms. If necessary, the computational efficiency has to be improved at the expense of accuracy of the results.

4.3 Evaluation

For evaluation I propose to build a reference database of sketch maps. This database will contain a huge set of hand-drawn sketch maps with object localisation and recognition that is done manually by humans. Each map will be repeatedly processed by different persons to detect uncertainties in human recognition. The developed algorithms will be tested against the certain ground truth.

There are plenty of sketch maps available in the Spatial Intelligence Lab (SIL) that can be used to build such a database. Since my work will be focused on survey maps of a given size, it might be necessary to create some additional sketch maps for the database. As pattern recognition classifiers need training data, there will also be a set of symbol patterns that can be divided into training and testing sets to evaluate the results of the used classifiers.

Beside the evaluation of the whole process, the steps will be evaluated, too. This comprises the localization of objects, the recognition of single symbols, the whole structural recognition and the semantic recognition at different levels of accuracy. This will make the outcomes of my work more comparable to other work that can be used for parts of the algorithm.

5 Expected Outcomes

The result of this research will be a set of algorithms that can detect and classify objects within a sketch map semantically. These algorithms will also return the detected street network and additional attributes of the objects. The identification of possible attributes will be part of my research. The research about possible will be inspired to the work of Jia Wang and Malumbo Chipofya, since recognized objects with their attributes can be used as input for their work. Thus, it is also important to know which kind of input they need.

Another result of my research will be a mobile implementation of the parts of my algorithms that are suitable for mobile environments.

The evaluation requires a database of sketch maps and ground truth data. This database will be an outcome that can be used to compare the performance of sketch map recognition systems.

6 Research Schedule

Task	Time	Chapters
Initial Literature Search	1 Month	-
Vectorization	5 Months	1
Structural Recognition	6 Months	1
Semantic Recognition and Integration	10 Months	1 – 2
Evaluation	4 Months	1
Finalize Writing	4 Months	2
	30 Month	6 – 7

7 Current State of Work

At the current point of my research, there is a prototype implementation for recognizing streets in sketch maps. This implementation covers structural and semantic recognition parts for streets. The structural part is to find parallel line segments. The subsequent semantic recognition detects the parallel line segments that represent streets. To do so, the relation to other line segments is considered. In addition, a paper [52] describing this algorithm for street network detection has been submitted for review to the workshop on graph-based representations in pattern recognition.

I also adviced the computer vision part of a joint project seminar about object recognition and spatial analysis in sketch maps. The seminar is still going on, but I expect some promising results for structural recognition of objects.

8 International Research Training Group

The goal of my research is to recognize the semantics of sketch map elements and in doing so enable the semantic integration of different sketch maps. Since the IRTG brings people together that deal with semantic integration I will benefit from the program.

In addition there are other IRTG students who are working on sketch maps. Their work utilizes sketch map objects for further processing, but does not deal with recognition of objects. Thus, my work will add an initial step to the whole process of dealing with sketch maps.

I propose two supervisors for my research. Since a huge part of my research will use computer vision methods my first supervisor will be Prof. Jiang. The second supervisor is Prof. Schwering, because my work will be integrated into the research that is done by the SIL.

References

[1] I. Schlaisich and M. Egenhofer, "Multimodal spatial querying: What people sketch and talk about," in *1st International Conference on Universal Access in Human-Computer Interaction*, pp. 732–736, 2001.

[2] P. Mishra, "Maps and disaster management," *Economic and Political Weekly*, vol. 37, no. 47, pp. 4676–4677, 2002.

[3] M. Zook, M. Graham, T. Shelton, and S. Gorman, "Volunteered geographic information and crowdsourcing disaster relief: A case study of the Haitian earthquake," *World Medical and Health Policy*, vol. 2, no. 2, 2010.

[4] R. Davis, "Magic paper: Sketch-understanding research," *Computer*, vol. 40, no. 9, pp. 34–41, 2007.

[5] M. Billinghurst and S. Weghorst, "The use of sketch maps to measure cognitive maps of virtual environments," in *Proceedings of the Virtual Reality Annual International Symposium (VRAIS'95)*, VRAIS '95, (Washington, DC, USA), pp. 40–, IEEE Computer Society, 1995.

[6] R. Golledge, "Human wayfinding and cognitive maps," in *Wayfinding behavior: Cognitive mapping and other spatial processes* (R. Golledge, ed.), pp. 5–45, John Hopkins University Press, 1999.

[7] M. Egenhofer, "Query processing in spatial-query-by-sketch," *Journal of visual languages and computing*, vol. 8, no. 4, pp. 403–424, 1997.

[8] A. D. Blaser and M. J. Egenhofer, "A visual tool for querying geographic databases," in *AVI '00: Proceedings of the working conference on Advanced visual interfaces*, (New York, NY, USA), pp. 211–216, ACM, 2000.

[9] A. D. Blaser, *Sketching spatial queries.* PhD thesis, University of Maine, 2000.

[10] V. Haarslev and M. Wessel, "Querying GIS with animated spatial sketches," in *Proceedings of the 1997 IEEE Symposium on Visual Languages (VL '97)*, VL '97, (Washington, DC, USA), pp. 197–204, IEEE Computer Society, 1997.

[11] M. Kopczynski, "Efficient spatial queries with sketches," in *Proceedings of ISPRS Technical Commission II Symposium*, pp. 19–24, 2006.

[12] F. Ferri, P. Grifoni, and M. Rafanelli, "The sketch recognition and query interpretation by GSQL, a geographical sketch query language," in *CIT*, pp. 34–38, 2005.

[13] K. D. Forbus, A. Lovett, K. Lockwood, J. Wetzel, C. Matuk, B. Jee, and J. M. Usher, "Cogsketch," in *AAAI*, pp. 1878–1879, 2008.

[14] K. Forbus, J. Usher, and V. Chapman, "Qualitative spatial reasoning about sketch maps," *AI magazine*, vol. 25, no. 3, pp. 61–72, 2004.

[15] K.-C. Fan, C.-H. Liu, and Y.-K. Wang, "Segmentation and classification of mixed text/graphics/image documents," *Pattern Recognition Letters*, vol. 15, no. 12, pp. 1201 – 1209, 1994.

[16] F. M. Wahl, K. Y. Wong, and R. G. Casey, "Block segmentation and text extraction in mixed text/image documents," *Computer Graphics and Image Processing*, vol. 20, no. 4, pp. 375 – 390, 1982.

[17] M. Zhao, S. Li, and J. Kwok, "Text detection in images using sparse representation with discriminative dictionaries," *Image and Vision Computing*, vol. 28, no. 12, pp. 1590 – 1599, 2010.

[18] L. A. Fletcher and R. Kasturi, "A robust algorithm for text string separation from mixed text/graphics images," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 10, no. 6, pp. 910–918, 1988.

[19] K. Tombre, S. Tabbone, L. Pélissier, B. Lamiroy, and P. Dosch, "Text/graphics separation revisited," in *Document Analysis Systems V* (D. Lopresti, J. Hu, and R. Kashi, eds.), vol. 2423 of *Lecture Notes in Computer Science*, pp. 615–620, Springer Berlin / Heidelberg, 2002.

[20] L. Li, G. Nagy, A. Samal, S. Seth, and Y. Xu, "Integrated text and line-art extraction from a topographic map," *International Journal on Document Analysis and Recognition*, vol. 2, pp. 177–185, 2000.

[21] H. Fujisawa, "Forty years of research in character and document recognition–an industrial perspective," *Pattern Recognition*, vol. 41, no. 8, pp. 2435 – 2446, 2008.

[22] T. Plötz and G. Fink, "Markov models for offline handwriting recognition: a survey," *International Journal on Document Analysis and Recognition*, vol. 12, pp. 269–298, 2009.

[23] L. Lam, S. Lee, and C. Suen, "Thinning methodologies-a comprehensive survey," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 14, pp. 869–885, 1992.

[24] R. W. Smith, "Computer processing of line images: A survey," *Pattern Recognition*, vol. 20, no. 1, pp. 7 – 15, 1987.

[25] X. Hilaire and K. Tombre, "Robust and accurate vectorization of line drawings," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 28, pp. 890–904, 2006.

[26] K. Tombre and S. Tabbone, "Vectorization in graphics recognition: To thin or not to thin," in *ICPR*, pp. 2091–2096, 2000.

[27] H.-H. Chang and H. Yan, "Vectorization of hand-drawn image using piecewise cubic Bézier curves fitting," *Pattern Recognition*, vol. 31, no. 11, pp. 1747 – 1755, 1998.

[28] A. Masood and M. Sarfraz, "Capturing outlines of 2D objects with Bézier cubic approximation," *Image and Vision Computing*, vol. 27, no. 6, pp. 704 – 712, 2009.

[29] F. Ferri and P. Grifoni, "Vectorization of graphical components in sketch-based interfaces," in *Databases in Networked Information Systems* (N. Bianchi-Berthouze, ed.), vol. 2822 of *Lecture Notes in Computer Science*, pp. 231–244, Springer, 2003.

[30] T. Hammond and R. Davis, "LADDER, a sketching language for user interface developers," *Computers & Graphics*, vol. 29, no. 4, pp. 518 – 532, 2005.

[31] B. Paulson and T. Hammond, "Paleosketch: Accurate primitive sketch recognition and beautification," in *Proceedings of the 13th International Conference on Intelligent User Interfaces*, pp. 1–10, ACM, 2008.

[32] D. Avola, A. Del Buono, G. Giorgio, S. Paolozzi, and R. Wang, "SketchML a representation language for novel sketch recognition approach," in *Proceedings of the 2nd International Conference on Pervasive Technologies Related to Assistive Environments*, (New York, NY, USA), pp. 31:1–31:8, ACM, 2009.

[33] T. A. Hammond and R. Davis, "Recognizing interspersed sketches quickly," in *Proceedings of Graphics Interface 2009*, (Toronto, Ont., Canada, Canada), pp. 157–166, Canadian Information Processing Society, 2009.

[34] G. Casella, V. Deufemia, V. Mascardi, G. Costagliola, and M. Martelli, "An agent-based framework for sketched symbol interpretation," *Journal of Visual Languages & Computing*, vol. 19, no. 2, pp. 225–257, 2008.

[35] M. Notowidigdo and R. Miller, "Off-line sketch interpretation," in *Proceedings of AAAI Fall Symposium on Intelligent Pen-based Interfaces*, 2004.

[36] L. B. Kara and T. F. Stahovich, "An image-based, trainable symbol recognizer for hand-drawn sketches," *Comput. Graph.*, vol. 29, no. 4, pp. 501–517, 2005.

[37] Y. Qiao and M. Yasuhara, "Recovering dynamic information from static handwritten images," in *Frontiers in Handwriting Recognition, 2004. IWFHR-9 2004. Ninth International Workshop on*, pp. 118–123, IEEE, 2004.

[38] W.-D. Groch, "Extraction of line shaped objects from aerial images using a special operator to analyze the profiles of functions," *Computer Graphics and Image Processing*, vol. 18, no. 4, pp. 347 – 358, 1982.

[39] J. Hu, A. Razdan, J. C. Femiani, M. Cui, and P. Wonka, "Road network extraction and intersection detection from aerial images by tracking road footprints," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 45, no. 12, pp. 4144–4157, 2007.

[40] A. Blaser, "Geo-Spatial Sketches," tech. rep., Univeristy of Maine, Boardman Hall 321 Orono, Maine 04469, USA, June 1998.

[41] M. J. Egenhofer and R. Franzosa, "Point-set topological spatial relations," *International Journal of Geographical Information Science*, vol. 5, no. 2, pp. 161–174, 1991.

[42] R. Praing and M. Schneider, "Efficient implementation techniques for topological predicates on complex spatial objects," *GeoInformatica*, vol. 12, pp. 313–356, 2008.

[43] C. Alvarado and R. Davis, "Dynamically constructed bayes nets for multi-domain sketch understanding," in *IJCAI-05, Proceedings of the Nineteenth International Joint Conference on Artificial Intelligence*, pp. 1407–1412, 2005.

[44] X. Jiang, M. Y. Ma, and C. W. Chen, eds., *Mobile Multimedia Processing: Fundamentals, Methods, and Applications*, vol. 5960 of *Lecture Notes in Computer Science*, Springer, 2010.

[45] D. Caduff and M. J. Egenhofer, "Geo-Mobile Query-by-Sketch," *Int. J. Web Eng. Technol.*, vol. 3, no. 2, pp. 157–175, 2007.

[46] R. O. Duda, P. E. Hart, and D. G. Stork, *Pattern Classification (2nd Edition)*. Wiley-Interscience, 2 ed., November 2000.

[47] D. Fensel, *Ontologies: a silver bullet for knowledge management and electronic commerce*. Springer Verlag, 2004.

[48] T. Gruber, "A translation approach to portable ontology specifications," *Knowledge acquisition*, vol. 5, pp. 199–199, 1993.

[49] J. F. Sowa, *Knowledge Representation: Logical, Philosophical, and Computational Foundations*, ch. 2. Pacific Grove, CA, USA: Brooks/Cole Publishing Co., 2000.

[50] S. Russell and P. Norvig, *Artificial Intelligence: A Modern Approach*, ch. 13. Prentice Hall, 3rd ed., 2009.

[51] D. Conte, P. Foggia, C. Sansone, and M. Vento, "Thirty Years Of Graph Matching In Pattern Recognition," *International Journal of Pattern Recognition and Artificial Intelligence*, vol. 18, no. 3, pp. 265–298, 2004.

[52] K. Broelemann, X. Jiang, and A. Schwering, "Automatic street graph construction in sketch maps." Submitted to the workshop on graph-based representations in pattern recognition, 2011.