Techniques for Improving Routing by Exploiting User Input and Behavior

A THESIS
SUBMITTED TO THE FACULTY OF THE GRADUATE SCHOOL
OF THE UNIVERSITY OF MINNESOTA
BY

Fernando Torre

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE DEGREE OF
DOCTOR OF PHILOSOPHY

Loren Terveen

October, 2014

Acknowledgements

In no way has this work been a one-man job. Many people have helped, supported, and challenged me along the way. I would therefore like to begin by acknowledging and expressing my gratitude to those who have contributed in some way to the making of this thesis.

First of all, I am grateful to have had Loren Terveen as my advisor. He challenged me often and was patient when I was slow to figure things out myself. His advice and guidance were instrumental in helping me get where I am today. I am also thankful to my thesis committee, John Riedl, John Carlis, Yuqing Ren, and Brent Hecht, for their guidance and feedback. I count myself lucky to have had such smart and at the same time kind and helpful members in my committee.

My research would have not been possible without the Cyclopath team. I thank Reid Priedhorsky, who started the project, for giving me an opportunity to join the team and for his detailed and thoughtful feedback. I am also thankful to Andrew Sheppard, for his help in the implementation of tags, to Phil Brown, for his help building the Cyclopath Android app, to David Pitchford, for his help implementing the initial versions of the map matching algorithm, to Yanjie Liu, for her help conducting interviews in China, to Landon Bouma, for all of his help with getting the landmarks experiments released, and to the rest of the team for all of the valuable discussions and feedback. I also thank the Cyclopath community, who have contributed greatly to this resource and on many occasions made themselves available for interviews.

I would also like to thank two research groups in particular. First, I am grateful for having been a part of the excellent GroupLens research group. This lab has a certain quality that simply attracts excellent professors and students. It was great to be part of a group that values both life-work balance and above-average research work.

Second, I am also indebted to Professor Zhengjie Liu and the Sino-European Usability Center at Dalian Maritime University, who opened their doors to me for research and collaboration. Their hopsitality was first class.

These years would have been much more difficult without the unwaiving support of my friends, family, and church community. In particular, I could not have done it without the support of my wife, Mariangellys Rodriguez. Throughout my studies, she has not failed to inspire and encourage me, especially when progress seemed to be at its slowest.

Finally, this work was made possible by NSF grants IIS 05-34692, IIS 08-08692, and IIS 09-64695, by a GAANN fellowship, by NSF EAPSI grant 1209673, and by SAFEA funds from the Chinese government.

Dedication

To Eugenio Torre and Lydia Gonzalez, parents who instilled in me the curiosity and wonder that got me here.

Abstract

This dissertation explores innovative techniques for improving the route finding process. Instead of focusing on improving the algorithm itself, I aim to improve the other factors that make the route finding experience better: personalization, map data, and presentation. I do so by making extensive use of user input (both explicit and implicit) and crowdsourcing strategies. This research uses Cyclopath, a geowiki for cyclists in the Twin Cities, MN, as a case study for the various techniques explored.

The first challenge is the lack of personalization in route finding algorithms. Aside from start and end points, users can rarely specify their riding preferences; algorithms usually know very little about users. However, user preferences can greatly affect their ideal routes. I studied the use of community-shared tags that allow users to specify preferences for those tags instead of doing so for each individual road segment, allowing them to easily express preference for a large number of roads with little effort. Correlation between individual road segment ratings and ratings deduced from tag preferences was evidence of the utility of this technique for making personalization easier.

The second challenge is missing data. The best routing algorithm is only as good as the map data underneath it. Unfortunately, maps are often incomplete. They might not have updates on the latest construction, they might be missing roads in rural areas or they might not include detailed information such as lanes, trails, and even shortcuts. I present an HMM-based map matching algorithm that uses GPS traces recorded by users to generate potential new road segments. Tests within Cyclopath confirmed the abundance of missing roads and the ability of this algorithm to detect them.

Finally, I look at the issue of unnatural presentation of routes. The way computers relay route directions is very different from humans, who use landmarks most of the time. However, gathering useful landmarks can be difficult and is often limited to points of interest. In this research, I tested methods for crowdsourcing different types of landmarks. Integrating landmark suggestions into route directions allowed users to contribute information within a relevant context. I show that POIs are not sufficient to represent landmarks and that there is no objective truth regarding which landmarks are more useful to users.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

Every month approximately one billion users visit Google Maps [1]. In 2011, Google's mobile Maps application boasted 150 million mobile users [2]. At that time, Google Maps Navigation was already guiding users 12 billion miles per year.

These statistics are quite impressive, but they don't even include other well-known competitors such as MapQuest and Bing Maps. Taking it a step further, these numbers are only representative of online route finding.

We are constantly asking ourselves questions such as "How do I get from here to there?", "Is there a faster way to get there?", "Is there a safer way to get there?", and even "Is there a different way to get there than I usually take?".

The ability to navigate has always been an essential part of our lives. With time, maps, stars, and compasses have been mostly replaced by GPS systems and online maps; but the end goal remains the same: to get the best route from point A to point B.

1.1 Challenges

With computers, the process of finding routes has become as fast and efficient as ever. Researchers have focused heavily on creating algorithms that can quickly find the shortest or simplest routes. However, challenges still remain in the route finding process. In this dissertation, I focus on tackling the following issues:

Missing Data The most efficient route finding algorithm will still not produce the best route if it is missing information about the road network. Unfortunately, maps are often incomplete. Roads get built and removed. Maps designed for one purpose, frequently driving, may not contain information about shortcuts, sidewalks, bike trails, and alleyways that are crucial for other purposes, such as bicyling or walking. Obtaining up-to-date, accurate data can be a difficult and expensive process.

Lack of Personalization Finding the best route is not always an objective process. For each user, the fastest route might not be the best route. Users' subjective preferences can affect whether they prefer to navigate via highways, whether they are in the mood for scenic routes, whether they prefer roads that they are already familiar with, and so on. However, most popular routing systems do not take users' preferences and contexts into account.

Unnatural Presentation Research shows that when humans give route directions, only about 15% of the direction elements are **not** related to landmarks [3]. However, route finding systems rarely include references to landmarks. Often this is the result of lacking information about which objects and features would actually serve well as landmarks. If this information was available, there would be more potential to make route directions feel more natural to users.

1.2 Framework

In order to better illustrate where these challenges fall in the route finding process, I present the framework in Figure 1.1. In general, the route finding algorithm takes map data and user preferences as input and produces one or more lists of route directions as output.

Map Data The most important input to any route finding algorithm is the map itself. An accurate and complete map is essential to the route finding processes. However, map data does not consist solely of a network of connected road segments. For example, information about traffic jams, speed limits, stop signs, and traffic lights is especially useful for finding the fastest route instead of the shortest route. Information about tolls and public transportation fares is useful for finding inexpensive routes. Other information such as elevation, road type, and even points

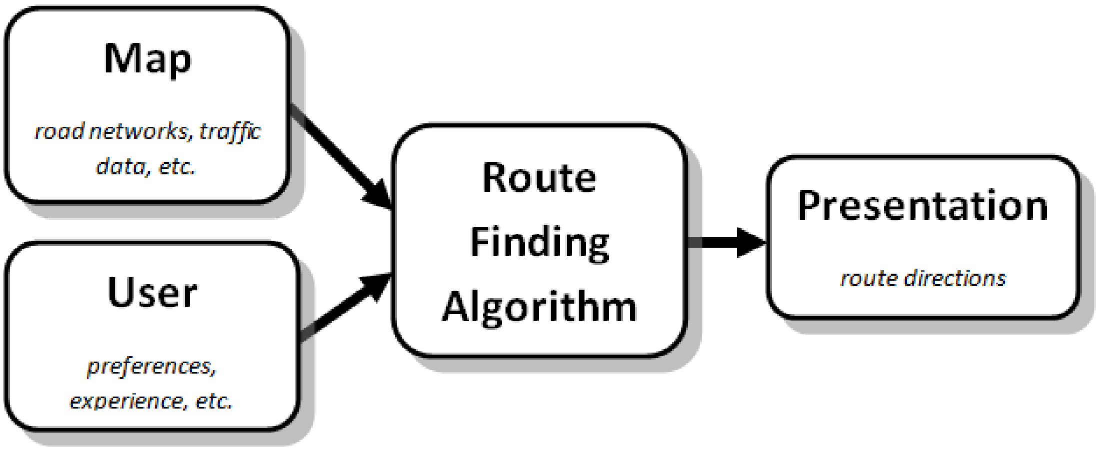


Figure 1.1: Route Finding Framework. The route finding algorithm takes map data and user preferences as input and produces a list of route directions as output.

of interest could be used to find easy, safe or even scenic routes. These data are all part of the map input which can be used to enhance route finding algorithms.

User Input Other than specifying start and end points to a route, user input is often neglected in route finding algorithms. However, users' subjective preferences can greatly affect their ideal route. Users may have preferences about types of roads (e.g. some users prefer to avoid highways) or about specific road segments (e.g. a specific dangerous intersection or a poorly-lit road). Furthermore, the user's route history can be a valuable indication of preference and familiarity. This gives route finding systems the ability to let users choose between navigating through paths that they are comfortable with and exploring new paths.

Route Finding Algorithm This is the core of the route finding process and the part that has been most deeply studied. The map and user inputs provide the pieces to the puzzle and the algorithm not only puts them together correctly, but does it as quickly and efficiently as possible. The goal is often to find a route that minimizes one or more variables, such as distance, time or cost.

Presentation At the end of the route finding process, the resulting route is presented to the user. In some cases, users can also be presented with multiple routes from which to choose from. Route presentation consists of two main parts: a visual representation of the route on a map and a list of directions.

The visual representation of the route displays the geometry of the route within the context of the map. It serves as a preview and lets users visualize properties of the route, such as length, compass orientation, road shapes (curves, T intersections, etc.), highways, and nearby landmarks (such as parks, rivers, and other points of interest). This representation can also show data used as input for the route finding process, which may have influenced the algorithm's decisions. For example, many route finding systems use color gradients on the roads to display traffic data.

The list of directions shows the sequence of roads and turns users must follow. Information such as length of each leg, duration of the route, and even cost (when there are tolls on the route) can be presented to users here. Furthermore, route directions can use information about the road segments and nearby landmarks to present the directions in a more rich and natural way.

1.3 Research Goals

My main goal in this thesis is to improve the route finding process. However, I do not focus on the route finding algorithm, but instead on improving the inputs and outputs of the process.

What makes this work unique is the use of users' explicit and implicit inputs to improve each of those components. Through the use of crowdsourcing techniques, mobile access, and peer production systems I give users the tools to make the most out of their route finding process.

The main contributions of this dissertation are the following:

Improving Personalization I study the use of community-shared tags for road segments that allow users to give routing preferences for road segments that share similar attributes. This approach lets users specify preference for large amounts of road segments easily (e.g. "I like roads with bikelanes", "I dislike unpaved roads"). It also benefits greatly from peer contributed information, as it uses tags applied to roads by the whole community.

Improving Map Data I present a map matching algorithm that uses GPS traces from users' rides to generate a list of potential new road segments to be presented to users. This approach helps improve the map by exploiting users' ride behavior, letting users contribute effortlessly to the map data.

Improving Presentation I present crowdsourcing solutions and interfaces for gathering landmark information. These landmarks are used to enhance route directions, making them more natural and useful. The crowdsourcing approach allows users to specify the landmarks and information that are relevant to their own contexts and experience.

1.4 Cyclopath

All of the work in this research is done through the Cyclopath system [4]. This section gives an overview of Cyclopath and why it was used as the research platform for this dissertation.

1.4.1 System Overview

Cyclopath[5] is a geowiki launched in the summer of 2008 that provides route-finding services for bicyclists in the metropolitan area of Minneapolis-St. Paul, MN, an area of roughly 8,000 square kilometers and 2.3 million people. Being a geowiki means that any user can edit the map, thereby creating a community-generated cycling map.

Users on Cyclopath can edit road segments (blocks), places of interest (points), and regions (such as cities). They can also add tags and notes to both blocks and points. Additionally, users can rate the bikeability of blocks. These ratings are used by the route finder to compute bike-friendly routes. Users can also create, save, and share routes. Finally, since Cyclopath is a wiki, users can monitor and revert any changes to the system. Figure 1.2 shows a screenshot of the Cyclopath web application.

In 2011, I led the development of a mobile Cyclopath extension for Android in order to provide location-aware services and help us gather GPS data from cyclists. Mobile users are able to browse the Cyclopath map, make simple edits (such as tags, notes, and more recently block ratings), find routes in the system, and record GPS traces of their rides. Figure 1.3 shows three screenshots of the Android application.

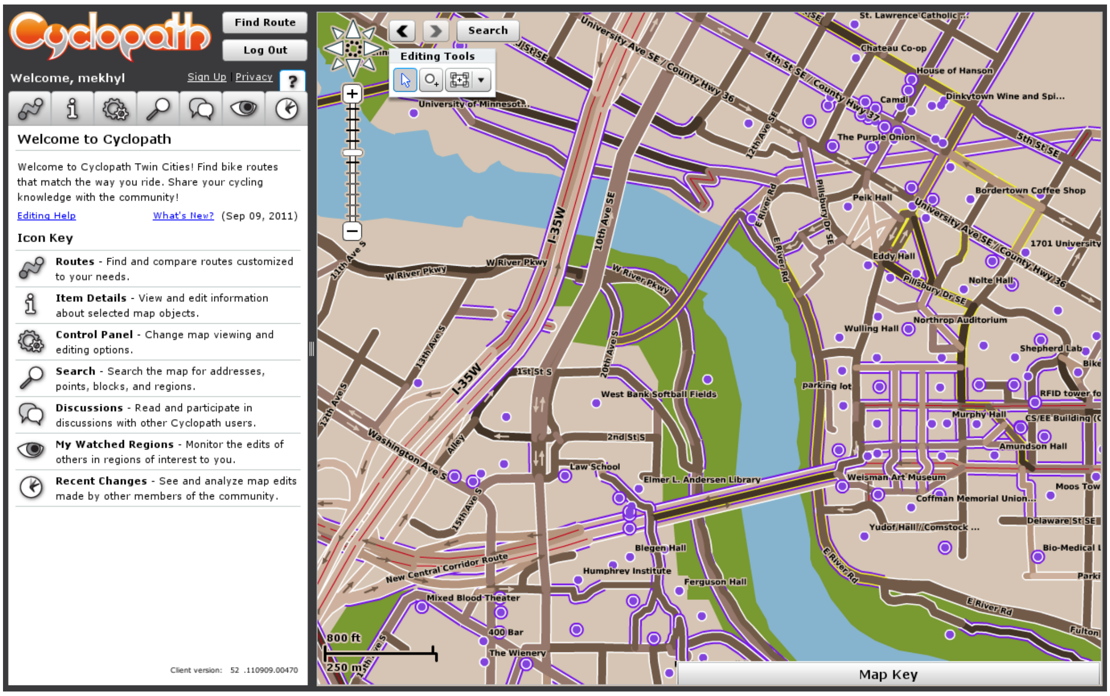


Figure 1.2: Cyclopath Web Application

At the time of the writing of this thesis, Cyclopath has more than 6,000 registered users and has been visited by more than 60,000 distinct IP addresses. The Android application is currently installed in more than 650 Android devices. Users have requested more than 130,000 routes (about 150 per day during biking season). They have also edited the map more than 20,000 times.

The Cyclopath road network currently contains more than 156,000 road segments covering a distance of more than 22,000 miles in the Twin Cities area. There are over 3,000 points of interest and over 400 regions. The map also has over 700 unique tags applied to over 27,000 blocks and points and over 3,200 unique notes applied to over 8,000 blocks and points. Users have also added over 80,000 bikeability ratings to blocks.

1.4.2 Research Fit

Cyclopath has multiple qualities that make it an ideal venue for researching methods to improve route finding.

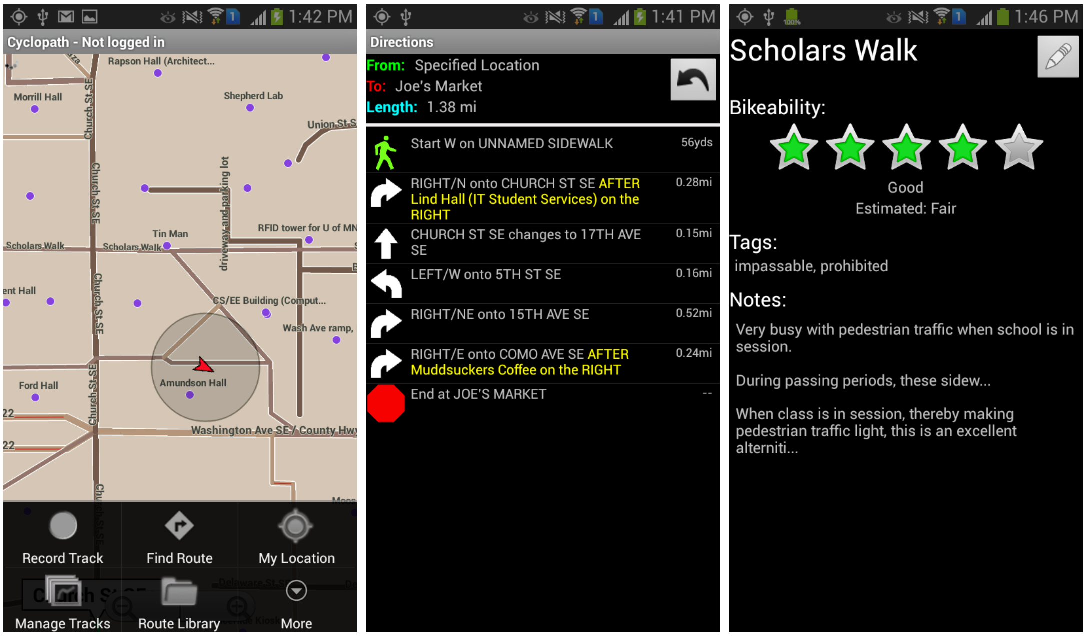


Figure 1.3: Cyclopath Android Application

Route Finding Focus Cyclopath's most used feature by far is the route finder. A system where route finding takes central stage is perfect for exploring ways to make route finding better. Improving this process will consequently improve the general user experience for all users of the system.

Cycling Context Cycling provides a unique and interesting context for researching route finding. First, most online maps are focused on finding routes for cars. They are often missing bike-related facilities and properties. Second, cyclists come with widely ranging opinions, preferences, and experience. In their case, user preferences can play a much more active role in the route finding process. An experienced cyclist commuter is more likely to be willing to ride through busy streets than a recreational cycling beginner. Finally, bike route directions can be harder to follow. Bike paths and bike lanes are not always labeled as clearly as roads. And unofficial paths, such as shortcuts and parking lots may also require better instructions.

Wiki Model The fact that Cyclopath is a wiki makes it a perfect platform for experimenting with crowdsourcing techniques in order to improve the map. Users are allowed and encouraged to improve every aspect of the map and consequently their own routes.

Total Access and Control Finally, one of the main advantages of using Cyclopath as a research platform is that since we built the system from the ground up, we have complete access to the data and interface. This allows us to study users more in-depth and to run experiments with more ease and control than with other platforms. For example, complete access to the database and to users logs allowed Panciera et al. to study differences between registered and anonymous user behaviors [6] and Masli et al. to study task specialization, including differences between registered and anonymous users [7]. Access to the user interface allowed Priedhorsky et al.[8] and Masli et al.[9] to compare different elicitation and compliance techniques. More recent work has extended Cyclopath's interface in order to aid transportation planners in analyzing proposed changes and therefore make better decisions [10].

1.5 Related Work

This section gives a quick overview of research work related to the route finding process described in the framework above and to some of the general techniques used in this research. Subsequent chapters in this thesis provide more relevant, in-depth related work as needed.

1.5.1 Route Finding

Given the pervasiveness of navigation tasks in our daily lives, it is no wonder that route finding algorithms have received a lot of attention among researchers. There is a large body of work detailing algorithms for finding shortest paths in the fastest way possible [11, 12].

More recently, route finding algorithms have been tackling new challenges aside from simply finding shortest paths. Some examples include live navigation (e.g. how to efficiently recalculate the current route in the presence of a traffic jam), multi-criteria routing (e.g. finding the best train path while taking into account not only distance, but also ticket costs and number of transfers [13, 14]), multimodal routing [15] (e.g. finding a route that contains a combination of cycling and taking the bus), and time-dependency (e.g. finding the quickest routes by taking into account traffic, speed, and departure times) [16, 17, 18].

Although most research work concerning route finding has focused on the algorithm, a growing body of work has also focused on improving the inputs and outputs of the route finding process.

Improving Map Data

There are two main approaches to improving data on a map. Specifically, I am interested in techniques for adding (and correcting) road network information.

The first family of approaches uses GPS traces to build map data automatically, such as [19], [20], and [21]. However, these algorithms have two significant limitations. First, sufficient amounts of high-precision GPS data are not always easy to obtain. In our system, for example, GPS data is obtained from users voluntarily using our application on their mobile device. Therefore we have a limited amount of GPS data, and these data are also usually not as accurate as data obtained from specialized GPS devices. Second, these algorithms do not work in the context of already-existing road network data. Even the incremental approaches have to build on previously obtained GPS data. Therefore, correcting existing map data remains a difficult task.

The second approach to improving map data is by letting users explicitly edit the data. Systems that let users contribute VGI (Volunteered Geographic Information), such as OpenStreetMap[22], Google MapMaker[23], and Cyclopath, fall into this category. This approach lets users not only add new road segments, but also fix and remove existing segments. Users can also add other types of information not available with GPS traces, such as road type and points of interest. Lastly, the community can also review changes made by other users.

Unfortunately, this process can be difficult and tedious to newcomers who are not familiar with the map editing tools. In Cyclopath, for example, we found that it was the experienced users who devoted a higher proportion of their efforts to editing road segments[7], which are more complicated than other types of work in Cyclopath.

In our approach we aim to make the road segment editing process easier by combining both approaches: extract possible new road segments from users' GPS traces while allowing users to confirm these suggestions and fix them as necessary. In this way we exploit users' implicit (GPS traces) and explicit (map edits) inputs in order to improve the underlying road network.

Improving Personalization

Personalizing the route finding process is a more recent area of research. Earlier work has looked at using previous experience to personalize future routes[24]. More recent work has looked at the need for "subjective human experience" in route finding [25] and at the effectiveness of personal road segment ratings and aggregation techniques from the field of recommender systems for providing personalized routes[26, 27].

Unfortunately, even when using aggregation techniques, these approaches require most road segments to have at least one rating or expressed preference in order to be effective. In sparse datasets such as map networks, this can be hard to achieve, especially in less urban areas of the map. For unfamiliar routes, users will depend almost exclusively on other users' preferences and not their own. Our approach allows users to use tags to give preferences to more general attributes of roads in order to easily express preference for a much larger amount of road segments.

Improving Presentation

Computer-generated route maps and directions tend to be very precise. However, they are very different from those designed by humans, including hand-drawn maps. Due to these differences, the presentation of computer-generated routes often feels unnatural to users.

The main reason for these divergences is the fact that most route presentation algorithms do not distinguish well between necessary and unnecessary information for navigating a route. For example, computer-generated route maps often include irrelevant information such as names of cities and far-away roads. Maps drawn to scale can also be providing extraneous information in some cases. Although they allow users to judge distances better, maps drawn to scale may fail to clearly show important and confusing route decision points that are too small to distinguish from far away. Systems such as LineDrive[28] solve many of these issues by simplifying and generalizing the visual display of route maps.

As for textual route descriptions, humans often use landmarks when communicating route directions [29, 30]. However, most computer-generated route directions fail to include landmarks, often because of a lack of information available. Some previous work has focused on the selection of landmarks from a list of possible points of interest by using models of landmark saliency [31]. In this work, I explore crowdsourcing and mobile techniques for the acquisition and validation of landmark information from users.

1.5.2 Peer Production Communities

Online communities that depend on user-generated content are often referred to as peer production communities. These communities allow systems to leverage user contributions in order to build shared resources. For example, Yahoo! Answers and StackOverflow allow the community to collaboratively share answers to common questions. SourceForge allows users to build software together. And Newgrounds brings users together in order to produce collaborative animations.

On the more transparent end of the spectrum of peer production communities are wikis, which allow users to not only contribute information, but also to review and oversee other users' contributions as well. The most famous example is Wikipedia, which boasts more than 4 million articles in the English language alone[32].

Most relevant to this work are the special class of wikis that let users collaboratively edit geographic data, otherwise known as geowikis. Site such as SeeClickFix[33] let users report issues of a geographic nature. Collaborative maps such as OpenStreetMap let users edit almost every aspect of the map, including roads, points of interest, and geographic features such as rivers, lakes, and forests.

All of the work in this thesis is done through a geowiki. This gives us the best opportunities to gather explicit and implicit inputs from users in order to improve the map and, consequently, route finding. Understanding how peer production communities work and how to motivate users to contribute is thus crucial to this research.

1.5.3 Mobile Crowdsourcing

One technique employed in this project is the use of mobile systems to obtain data from users. The technique of delegating tasks to mobile users is often referred to as mobile crowdsourcing. In general, crowdsourcing has the advantage of outsourcing a lot of work to a lot of people at little or no cost. Mobile crowdsourcing additionally takes advantage of location to improve the selection of users for specific tasks. It assumes that users have better access to information about a location when they are actually at (or have been at) that location. Opportunistic mobile crowdsourcing systems are even able to capture data without direct intervention from the user [34]. Applications such as Waze[35] employ opportunistic crowdsourcing to help gather live traffic data.

In this work I employ opportunistic techniques to gather GPS data and experiment with non-interruptive elicitation approaches to gather landmark information.

1.6 How this Thesis is Organized

The rest of this thesis is organized as follows. Each of the next three chapters focuses on the contributions to each area of the route finding process. Chapter 2 describes the implementation and analysis of community-shared tags for improving route personalization. Chapter 3 describes an HMM-based map matching algorithm that can detect missing blocks from GPS traces in order to improve map data. Chapter 4 describes the research concerning the use of landmarks to improve route presentation: the motivations behind the use of landmarks for this thesis and experiments on crowdsourcing the acquisition and validation of landmarks. Finally, Chapter 5 ends with some thoughts on future work and implications that arise from this research.

A significant part of the work presented in this dissertation has been published before in the following research papers which I have authored:

- Torre, Fernando, S. Andrew Sheppard, Reid Priedhorsky, and Loren Terveen. "bumpy, caution with merging: An exploration of tagging in a geowiki." In *Proceedings of the 16th ACM international conference on Supporting group work*, pp. 155-164. ACM, 2010.

- Torre, Fernando, David Pitchford, Phil Brown, and Loren Terveen. "Matching

GPS traces to (possibly) incomplete map data: bridging map building and map matching." In *Proceedings of the 20th International Conference on Advances in Geographic Information Systems*, pp. 546-549. ACM, 2012.

- Torre, Fernando, Yanjie Liu, Zhengjie Liu, and Loren G. Terveen. "Local Knowledge Matters for Crowdsourcing Systems: Experience from Transferring an American Site to China." In *ICWSM*. 2013.

Chapter 2

Improving Personalization through User Tag Preferences

2.1 Introduction

Tagging has become a ubiquitous information management technique on the Web. Users can apply tags – short textual labels – to items of interest and use these tags to browse and search for items. The design of tagging systems and the analysis of how tagging vocabularies evolve are active research areas [36, 37, 38, 39, 40, 41].

This chapter reports on a study of tagging in Cyclopah. Tags in our system allow for not only the annotation of information, but also the indication of routing preferences, which consequently helps improve route finding via personalization.

Tags in Cyclopath present a new and distinctive context for the design and analysis of tagging systems in general for several reasons:

- **Geographic.** Users tag geographic objects in an interactive map. There are two distinct domains of geographic objects: *blocks*, atomic segments of the roads and trails that make up the transportation network, and *points*, places added by users that serve as the start and end points of routes and aid navigation.
- **Rich annotation ecosystem.** We introduced tags into Cyclopath about nine months after the site went live. Two other annotation techniques had been available from the beginning: *notes*, free-form textual comments attached to points or

blocks, and *ratings*, subjective opinions of the bikeability of blocks.

- **Novel application.** We modified the Cyclopath route finder to let users express routing preferences with tags and thus fine-tune the routes they received.

I pose three research questions to understand the tagging behaviors that have emerged in Cyclopath:

RQ1. Vocabulary. *How do the tagging vocabularies that evolve for the two object domains – blocks and points – compare? What factors drive the evolution of tagging vocabulary in the Cyclopath domain?* Looking at the vocabulary in a tagging system is one of the most common ways to gain an understanding of how tags are used in the system. I found that the two object domains in Cyclopath developed almost wholly distinct vocabularies with quite different characteristics; I argue that the differences emerge because the two types of objects support quite different tasks.

RQ2: Ecosystem. *How do users employ the three annotation techniques tags, notes, and ratings? Do they express overlapping or distinct content?* Having three closely related techniques made it natural to investigate the interactions and relationships between them. I found significant content overlap between notes and tags. I also computed the notion of tag-derived preference and found a positive correlation between tag-derived preferences and ratings.

RQ3: Specialization. *How do individuals balance annotation roles, techniques, and objects? Are they specialists or generalists?* The next step was to study tags (and annotations in general) at the user level. I found strong specialization: users tend to either use or apply tags, not both, and annotate either blocks or points with either notes or tags.

2.2 Related Work

2.2.1 Geowikis and geotagging

Many web sites let end users edit geographic content. Google Maps lets users edit the locations of searched-for places and add new places. Google My Maps goes further, enabling collaborative editing of geographic points, paths, and polygons, all of which can be annotated with text, images, and videos. Sites like FixMyStreet and SeeClickFix let users plot the location of potholes and similar problems on a map. Open Street Map is a large project to build a worldwide street map with a wiki. Google Map Maker lets users directly edit Google Maps data (in some countries) and submit those changes for inclusion in the public map. There is a growing body of scholarly work on these systems, including multiple studies of Cyclopath [6, 8, 5] and an analysis of how FixMyStreet facilitates citizen-government interaction [42].

This research extends previous work by examining the design choices required to add tagging to a geowiki and analyzing a year's worth of user tagging behavior.

It helps to clarify the difference between this work and geotagging: adding geographical metadata (typically latitude and longitude) to content like web pages and photographs (as in Flickr). This work looks at almost the opposite situation: adding tags to existing geographic data.

2.2.2 Tagging

Tagging is an active research area, including studies of algorithms to suggest tags to users [43, 40], evaluations of tag clouds [39], and studies of tagging in an enterprise information sharing system [44]. Two strands of work are most relevant to this work: analysis of the vocabularies that emerge in tagging systems and studies of how tags relate to associated techniques like recommender systems.

Vocabulary analysis. Research includes information theoretic analysis of the overall structure and evolution of tagging vocabularies [36], empirical analysis of what constitutes "quality" for tags [45], and several schemes for categorizing tags [37, 40]. I use the categories introduced by Sen et al. [40] to analyze the Cyclopath tagging vocabulary.

Tags and associated techniques. A large body of work explores various ways to integrate tags and recommender systems, e.g., using tags to infer user preferences for items and improve recommendation algorithms [46, 47, 48] and using tags to explain recommendations [49]. Other work has investigated the relationship between the text of web pages and the tags applied to web pages, finding that 50% of tags were present in the text of web pages to which they were applied [38]. In this research, I examine the relationship between preferences expressed through tags and through ratings, and I compare the content and usage patterns of tags to textual annotations.

2.2.3 Specialization

Prior research on a variety of online communities has found that users take on specialized roles. In online discussion forums, Turner et al. [50] and Welser et al. [51] identify different roles that users assume, notably "Question Person" and "Answer Person". In Wikipedia, Welser et al. [52] mapped out various roles that contributors can play, such as technical editors, substantive experts, vandal fighters, and social networkers. Bryant et al. [53] found that Wikipedia editors shifted concerns as they became more experienced, evolving from a focus on topics about which they had some personal expertise to taking on different types of "community maintenance", e.g. monitoring for vandalism and enforcing policies like "Neutral Point of View".

Compared to prior work, I investigate a particular type of specialization: how Cyclopath users balance their choice of annotation roles, techniques, and objects.

2.3 The Cyclopath Tagging System

2.3.1 Design Choice: Wiki-tags

We faced a number of choices when we designed the Cyclopath tagging system. Most notably, we had to decide whether tags should be normal objects in the wiki model. For our purposes, the key properties of wiki objects are:

- They are *public*. All objects may be changed or deleted by any user. Objects do not have owners.
- Each set of changes (and deletions) results in a new *revision* that captures the changes. There is only one current version of an object, but previous versions are saved. Any user may revert (undo) any revision, rolling back to a previous version.

We can interpret the implications of making tags normal wiki objects in the framework of Sen et al. [40]. Wiki-tags would be completely open or public on the *sharing* dimension. For example, if the tag *construction* is applied to a block, this is visible to

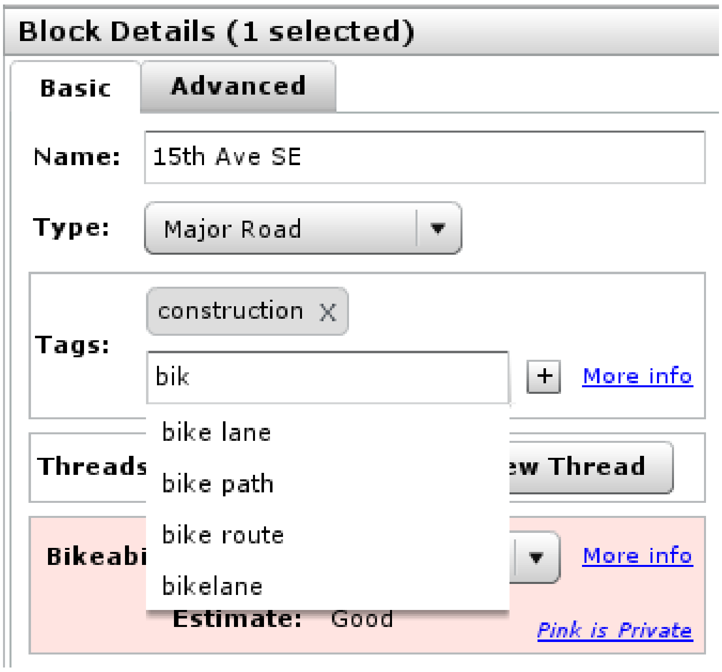


Figure 2.1: Tag application interface. The tag *construction* has already been applied to this block. The user is entering a new tag (so far "bik"), and auto-complete shows existing tags that begin with the text entered so far.

all users. Wiki-tags are owned by the community, and there is a single set of tags for any object; thus wiki-tags have *broad scope*. For example, the tag *construction* can be applied at most once to a given block.

We could have implemented tags as a separate add-on to the wiki or by extending the wiki model (e.g., by remembering the user who applied a tag to an object, and making that tag application only editable by that user). However, we prefer to follow the wiki model whenever possible. Moreover, we wanted to promote the use of tags that were useful to the community. Therefore, implementing tags as wiki objects was a good fit for our goals.

2.3.2 Entering and using tags

Tags are applied to points or blocks using a rather standard interface (see Figure 2.1). Users type into a text field, and an auto-complete function suggests tags that have been applied to this type of object. This encourages users to reuse tags.

In addition to enhancing user exploration, tags are used in two system functions. (a) Point tags can be used to filter the display of points on the map. For example, a user could request to see only points that had one of the tags *air pump* or *bike rack*. Filtering is rather straightforward, and I do not discuss it further in this chapter. (b) Block tags can be used to tailor route-finding preferences.

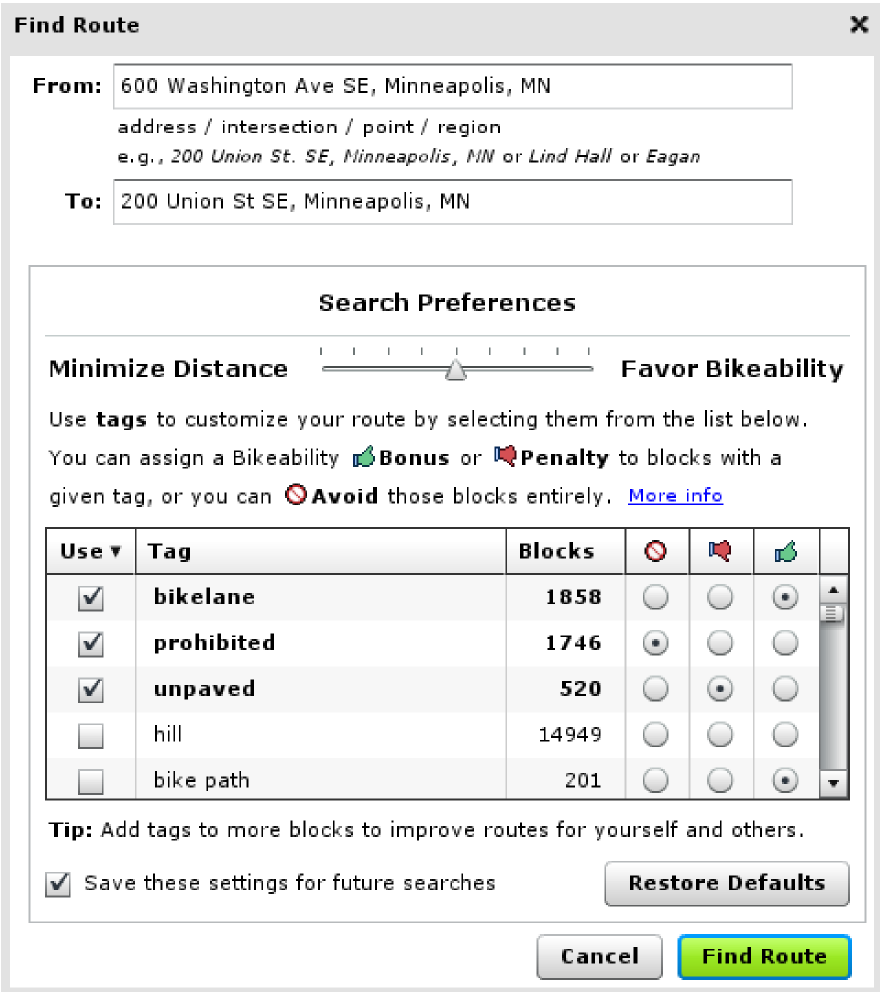


Figure 2.2: Route finder and tag preferences interface. In this example: (a) blocks with the *bikelane* tag are favored, (b) blocks with *unpaved* are penalized, (c) blocks with *prohibited* are avoided completely, and (d) all other tags (in particular, *bike path* and *closed*) have no effect.

Route-finding preferences. Before we introduced tags, our route-finding algorithm estimated the bikeability of a block by considering user ratings if available, and by using objective features (such as block type, speed limit, and shoulder width) otherwise. We believed that tags would be an effective complement to ratings. A user can have only one rating for a given block – one of five choices from "Impassable" to "Excellent" – but this does not capture why a user likes or dislikes a block. Hence, we modified the route-finding interface to let users express preferences for or against certain tags (see Figure 2.2). Users may assign selected tags a bonus or a penalty or state that they be avoided altogether. (User tag preferences are saved between route requests.) This innovative use of tags gives users more expressivity and control. For example, a user might assign a penalty to the *hill* tag in most requests, but assign this tag a bonus when in the mood for a challenging workout. The route finding algorithm respects any specified tag preferences. If a block has any tag that the user wants to avoid, that block will not be added to the route. If a block has a tag that has been assigned a bonus (or penalty), the block's bikeability estimate will be incremented (or decremented).

2.3.3 Annotation framework

Tags have both similarities and differences with the two other Cyclopath annotation techniques, notes and ratings. Like notes, tags can be applied to blocks or points, are public, and aid users in evaluating points, blocks, and routes. However, like ratings, tags are used by the route finding algorithm. Table 2.1 summarizes the three annotation techniques.

2.4 Tagging Use and Data

Tagging was added to Cyclopath in April 2009. In the subsequent year, 178 users applied tags, creating a vocabulary of 239 distinct tags that had been applied over 1,900 times to over 1,400 distinct blocks and points. (There would be more tag applications if tags had *narrow* (multiple tag applications per user for each object) rather than *broad* (a single set of tag applications for each object) scope [40]). Over one quarter of all revisions included tag edits. Over 600 users had filtered with tags and over 2,000 users had expressed route finding preferences with tags. As common in open content systems,

Annot.	What	Who	Used by
ratings	blocks	private	**machine** route finding algorithm
notes	blocks, points	public	**human** evaluate blocks and points
tags	blocks, points	public	**human and machine** filter points express route preferences evaluate points and blocks route finding algorithm

Table 2.1: Framework for Cyclopath annotations.

many more users consume information (here, using tags for filtering and route finding) than produce information (here, applying tags).

The tagging vocabulary contained a number of tags that were applied automatically. Prior to the introduction of tagging, blocks could have any of three binary features: *unpaved*, *bike lane*, and *closed*. We automatically converted these features into tags, thus creating a small initial tag vocabulary consisting of 2,470 applications of these three tags. Later in 2009 we wrote code to automatically apply two additional tags. *prohibited* was applied to about 1,700 blocks (mainly expressways) where it is illegal to ride a bicycle. *hill* was applied to almost 15,000 blocks with a slope greater than 2%, based on existing measures of hilliness and existing applications of the tag in the system. We also set default route finding preferences to avoid both the *closed* and *prohibited* tags. We refer to this set of five tags as auto-applied tags.

In my analysis of tag applications, I ignore all auto-applied tags and any tags created by members of the Cyclopath research team. In my analysis of tag preferences, I ignore preferences for *closed* and *prohibited*, because these tags have default preferences in the route finding dialog.

Table 2.2 summarizes the tagging data analyzed. In the next three sections, I sketch my methods (a combination of manual coding and quantitative analysis), present my findings, and discuss their implications.

	Blocks	Points
tags	91	163
tags (non-auto-applied)	86	163
tag applications	20,030	973
applications of non-auto tags	940	973
objects with tags	19,525	668
objects with non-auto tags	862	668
tag applications removed	626	12

revisions to Cyclopath since tags went live	3406
revisions involving tags	919
users that have applied tags	178
users that have removed tag applications	49
users that have filtered using tags	623
users that have used tag routing preferences	2089

Table 2.2: Usage statistics.

2.5 RQ 1. Vocabulary

In this section, I analyze the tagging vocabulary that evolved in Cyclopath and identify factors that influenced its development. A *tagging vocabulary* is the set of distinct tags used in an online community. For example, if the tag *low traffic* has been applied twice, *rocky* five times, and *gravel* once, then the vocabulary would be the set {*low traffic*, *rocky*, *gravel*}.

As mentioned earlier, blocks and points are intrinsically different and play different roles in Cyclopath: blocks are the components of routes, and points are landmarks and the start and end points for routes. Further, block tags and point tags serve different purposes: block tags can fine-tune route preferences, while point tags filter the map. Thus, I provisionally treat block tags and point tags as two separate vocabularies and do several analyses to see if this separation is warranted.

2.5.1 Tag diversity

The first characteristic that I examined for the two tagging vocabularies was diversity, the ratio of the number of distinct tags to the number of tag applications. High diversity means that each distinct tag has been applied just a few times; low diversity means that each distinct tag has been applied many times. Cyclopath point tags had a diversity of 0.18, and block tags had a diversity of 0.09. Therefore, the mean point tag had been applied about 5 times, and the mean block tag about 10 times. However, the mean number of applications is a misleading statistic: both vocabularies followed the expected non-normal "long tail" distribution. Indeed, the reason the point tag vocabulary was more diverse is because it had a longer "tail", i.e., tags that had been applied just once.

Contributing to the lower diversity of block tags are the inherent nature of blocks and a specific feature of the Cyclopath user interface. Contiguous sequences of blocks often share a feature – for example, several blocks in a row may be bumpy or scenic. Users can select a set of blocks, then apply the same tag to the entire set with a single UI action. On the other hand, points must be evaluated separately, and point tags applied one by one.

A difference in diversity tells us about the global structure of the two tagging vocabularies. However, it does not tell us about similarities or differences in their content. I examine this issue next.

2.5.2 Overlap

A natural way to compare the two tag vocabularies is simply to compute the overlap (intersection). As background, I note that the auto-complete feature (see Figure 2.1) could bias user tag application decisions. To avoid biasing users in favor of developing separate tag vocabularies for points and blocks, we initially designed auto-complete to suggest tags applied to both blocks and points. However, after about six months of use, only 7 out of 109 tags occurred in both vocabularies. To better support emergent practice, we then modified auto-complete to suggest only tags from the relevant vocabulary for each object. The total overlap at the time of this study was then 10 tags out of 239 distinct tags.

For those tags that appeared in both vocabularies, the majority of their applications were in just one. For example, the tag *steep hill* had been applied to 20 blocks, but just one point. The tag *food* had been applied to 67 points, but to just two blocks (which perhaps have many restaurants alongside them).

Since blocks and points are different types of objects, it is not surprising that different terms are used to describe them. However, perhaps a more abstract categorization of the two vocabularies would reveal deeper similarities – or clarify their differences. I did a content categorization to investigate these possibilities.

2.5.3 Content Categorization

I did three categorizations of the tagging vocabularies:

- **Factual, subjective, personal.** Sen et al. [40] used these categories to analyze the MovieLens tagging vocabulary. They also found that different categories were useful for different user tasks.

- **Cycling content.** Priedhorsky et al. [5] categorized Cyclopath notes, with a fundamental distinction between notes that were and were not directly related to cycling. They found that while most point notes were not related directly to cycling, most block notes were.

- **Part of speech.** A very general way of distinguishing vocabularies is in terms of the parts of speech of the individual tags.

For each content categorization, three coders were given classification rules and categorized the tags independently. Tags that did not fit into any of the categories were classified as *other*. After independent classification, the tags were assigned their final categories using majority rule.

Factual, subjective, personal categorization

Building on the definitions by Sen et al., I define factual, subjective, and personal tags as follows:

1. **Factual** tags express objective properties of an item. For example, in Cyclopath, *bridge* and *one way* are factual block tags, and *bike shop* and *pizza* are factual point tags.

2. **Subjective** tags express opinions. This class can be quite broad, since even if nearly everyone would agree on the application of a tag, we classified it as subjective if disagreement was reasonable. Some examples are *scenic* and *dangerous* for blocks and *bad coffee* and *awesome* for points.

	Blocks	**Points**
factual	58%	91%
subjective	35%	5%
personal	1%	0
other	2%	2%
no agreement	3%	2%

(a) factual/subjective/personal

	Blocks	**Points**
bike-specific	70%	6%
non-bike-specific	21%	91%
other	1%	1%
no agreement	7%	2%

(b) cycling content

	Blocks	**Points**
noun (is-a)	33%	49%
noun (has-a)	20%	37%
adjective	34%	5%
verb	2%	4%
other	3%	2%
no agreement	8%	3%

(c) parts of speech

Table 2.3: Results for content categorizations.

3. **Personal** tags have the user who applied them as the only intended audience. An example of a personal tag in Cyclopath is *home*.

The results of this categorization are shown in Table 2.3a. All three coders agreed 78% of the time for blocks and 90% of the time for points, while at least two of the coders agreed 97% of the time for blocks and 98% of the time for points. In comparison, the distribution of tags in MovieLens was 63% factual, 29% subjective, 3% personal, and 5% "other" [40].

There were few personal tags in either vocabulary; indeed, most personal tags that were applied were removed by other users. One such example is the tag *work1*, for which 74 applications to blocks were removed.

I expected that community ownership of tags would lead to a very high proportion of factual tags. And factual tags indeed are the majority in both vocabularies. However, while over 90% of point tags are factual, the distribution of the vocabulary for block tags is more comparable to that of MovieLens (where tag applications are owned by users, not the community), and there is a sizeable minority of subjective block tags. We can consider two possible explanations for this. First, perhaps Cyclopath users in general simply agree that a particular block is *bumpy* or *dangerous*, even though it is theoretically possible to disagree. Second, this may indicate an intrinsic problem with the Subjective category or at least our application of it. Maybe "subjective" is subjective – or maybe we were too strict in our application.

Why do we see a different distribution for point tags? To answer this question, I refer back to Sen et al. They identified five user tasks supported by tags in MovieLens: self-expression, organizing, learning, finding, and decision support. Surveys showed that different classes of tags were more useful for different tasks. Factual tags were particularly useful for learning and finding, while subjective tags were useful for self-expression and decision support. The learning and decision support tasks are most important in Cyclopath: learning is supported by map exploration, and decision support occurs during route finding and evaluation. Point tags are not used during route finding. Therefore, they are mainly used for learning, which explains the high percentage of factual tags. On the other hand, block tags must support both learning and decision support, which may account for the higher proportion of subjective tags.

Cycling content categorization

I also distinguished tags by whether they were bike-specific:

1. **Bike-specific** tags are directly related to the practice of bicycling. Block tags like *caution with merging* and *moderate traffic* and point tags like *free air* and *covered bike parking* fall into this category.

2. **Not bike-specific** tags are everything else. These include block tags like *river* and *storefronts* and point tags like *church* and *fast food*.

Table 2.3b shows the results for this categorization. All three coders agreed 40% of the time for blocks and 90% of the time for points, while at least two coders agreed 93% of the time for blocks and 98% of the time for points.

As with notes [5], a much higher proportion of block tags were bike-specific. This is because when planning and evaluating routes, cyclists are concerned with properties of the route (or the blocks that comprise it) that affect bikeability. On the other hand, points are of more general interest as possible destinations, landmarks, or resources along the way; therefore, cyclists likely are more interested in intrinsic information about the places themselves, e.g., is it a good place to get food or an interesting place to visit?

Parts of speech categorization

I used the following rules to categorize by part of speech:

1. **"Is-a" noun** tags fit the pattern "this block/point is a(n) X". Examples include *alley* and *highway* for blocks and *bakery* and *gas station* for points.

2. **"Has-a" noun** tags fit the pattern "this block/point has X". Examples include *curb* and *no shoulder* for blocks and *internet* and *coffee* for points.

3. **Adjective** tags describe a property of a block or point. Examples include block tags like *quiet* and *rough* and point tags like *expensive* and *awesome*.

4. **Verb** tags describe actions relevant to a point or block, such as the block tag *avoid* and the point tag *get the fudge cake*.

The results for this categorization are shown in Table 2.3c. All three coders agreed 56% of the time for blocks and 62% of the time for points, while at least two of the coders agreed 92% of the time for blocks and 97% of the time for points.

Once again, the categorization yielded quite different results for both vocabularies. The most striking difference is in the use of adjectives. About a third of all block tags were adjectives, while only 5% of tags applied to points were adjectives. These adjective block tags give information about what it's like to ride there. On the other hand, point tags often describe what a place is or what can be found there, uses well-supported by nouns.

2.5.4 Results Summary

Cyclopath is unusual in that two separate tagging vocabularies with very different characteristics emerged. The block tag and point tag vocabularies differed not only in their specific terms, but also in the type of content and distribution. Three content categorizations revealed unique, yet mutually supporting points of comparison. Point tags tended to be factual nouns and not bike-specific, while block tags were a mix of factual and subjective nouns and adjectives and were mostly bike-specific. I trace these distinctions to intrinsic differences between points and blocks, the different roles they play in Cyclopath, the different roles played by block and point tags, and user interface differences.

2.6 RQ 2. Ecosystem

Tags, notes, and ratings comprise an ecosystem of annotation techniques in Cyclopath. Table 2.1 (in Section 2.3 above) summarizes the important properties of each technique. Here I investigate similarities and differences in the usage patterns of tags vs. notes and tag preferences vs. ratings.

2.6.1 Tags vs. Notes

Tags and notes have obvious similarities. Both describe blocks and points, and both are public, owned by the Cyclopath community. As is typical, Cyclopath tags are mostly single words, with a few short phrases. On the other hand, notes can be arbitrarily long texts. For example, here is a Cyclopath block note: *Buses go very fast along here, but they generally give you plenty of space. No stop signs!*, and here is a point note: *They fix and give away commuter bikes, on volunteer time and donations. Good source of used parts.* I examined two relationships between tags and notes: content overlap and usage substitution.

Content overlap. I examined cases where both tags and notes had been applied to a single block. For example, the tag *dangerous* had been applied to a certain block, and that block also had the following note:

> *The bike lane on the left side abruptly disappears in Dinkytown and reappears*

> *after passing through w/ dangerous parking situation. It reappears only to disappear again at the ramps to 35W. Here it is totally unsafe...*

Thus, the tag *dangerous* appeared literally in the note.

The overlap analysis was simple: for each block b with tags $t_1, ..., t_N, N > 0$ and notes $n_1, ..., n_M, M > 0$, check whether t_i appears in any of the notes $n_1, ..., n_M$, and the same for points. 31% of block tag applications and 26% of point tag applications were mentioned in a corresponding note.

Usage substitution. Given the similarity of tags and notes, I wondered how the introduction of tags 9 months after Cyclopath went live affected note usage. I thus calculated the mean number of note edits per revision before and after tags were introduced. There was a significant difference in the mean number of note edits per revision before and after tags were introduced: 0.87 vs. 0.66 (t-test; p¡0.01). This suggests that tags assumed some of the functions notes had served.

2.6.2 Tag Preferences vs. Ratings

As I explained earlier, both ratings and tags play a role in the Cyclopath route finder, creating a context for comparing the two. I now show how the tag preferences users express when requesting routes can be used to derive user preferences for blocks, then investigate whether these tag-based block preferences correlate with the preferences users express through ratings.

Tag Preference/Ratings Correlation

Ratings let users express specific bikeability preferences for specific blocks: *I think this block of Summit Avenue is "excellent"* or *I think this block of Hennepin Avenue is "poor"*. Tags let users express generic preferences for the types of blocks they want in a route: *I prefer blocks with the tag "bikelane" (a bikeability "bonus") and dislike blocks with the tag "bumpy" (a "penalty")"*. Consider the following situation, as illustrated in figure 2.3: a user u has rated a block b as "excellent"; the tag *bikelane* has been applied to b (not necessarily by u); u has expressed a "bonus" route-finding preference for the tag *bikelane*. u's (directly expressed) ratings-based preference for b and u's (derived) tag-based preference for b appear consistent. This makes sense: if a user likes blocks

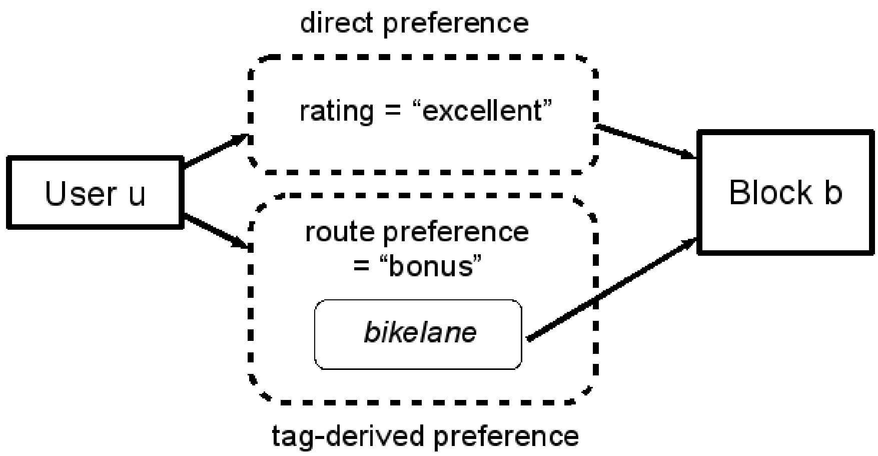


Figure 2.3: The relationship between ratings (directly expressed preferences) and tag-derived preferences.

with *bikelane*s, shouldn't the user rate block b highly?

My goal was to test this intuition formally. I first introduce some terminology: a user has a *tag-derived preference* for a block b if there exists at least one tag t such that (a) t has been applied to b, and (b) the user has expressed a route-finding preference using t. I then compute the global correlation between ratings and tag-derived preferences by aggregating across all users, blocks, and tags. I next show how to compute *tag-derived preferences.*

Tag-derived preferences

The intuition for computing tag-derived preferences is simple. Suppose the tags t_i, and t_j have been applied to a block b. If a user u expressed a "bonus" preference for t_i, this should lead to a positive tag-derived preference for b, and if the user expressed a "penalty" or "avoid" preference for t_i, this should lead to a negative tag-derived preference for b. And if the user expressed opposing preferences for t_i and t_j, i.e., "bonus" for one and "penalty" or "avoid" for the other, this should lead to a more-or-less neutral tag-derived preference for b. To formalize these intuitions, we must make two decisions: (a) how to translate tag preferences to numeric values,[1] and (b) what to count as a user having a preference for a tag. I consider each of these in turn.

[1]Recall that ratings already are on a numeric scale of 0 ("impassable") to 4 ("excellent").

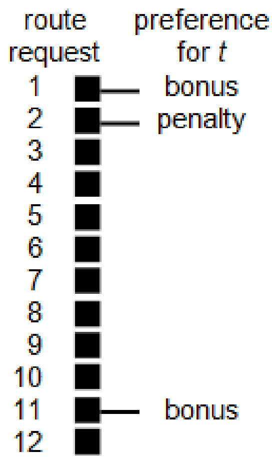


Figure 2.4: Example use of tag preferences in route requests used for illustrating three ways to count tag preferences.

Translating tag preferences to numeric values. We must specify precisely how to go from *bonus*, *penalty*, and *avoid* to numeric values. It is easy to specify constraints on possible values. (a) Values should range from -1 to 1. (b) A mathematical relationship that must hold among the three preferences is: $-1 \leq \text{avoid} < \text{penalty} < 0 < \text{bonus} \leq 1$. (c) Since the "avoid" preference is as negative as possible, it makes sense to set its value to -1. (d) "Penalty" and "bonus" are opposites (conceptually and in the Cyclopath route finder), so it makes sense for "penalty" to be equal to "$-$bonus. However, there are no obviously correct values for "penalty" and "bonus". Therefore, I experimented with a number of different values, including 0.75, 0.5, and 0.1. When I computed correlations between tag-derived preferences and ratings using the different values, the results were virtually identical unless I gave "penalty" and "bonus" scores close to 0. When I present the correlations below, I present them using two sets of values for ("avoid", "penalty", "bonus"): (-1, -0.75, 0.75) and (-1, -0.1, 0.1).

Tag preferences: what to count. What it means for a user to have a preference for a tag turns out to be a bit complicated, since users may change their preferences with each route request (recall that preferences are saved between requests). Figure 2.4 illustrates this situation. A user has issued a total of 12 route requests, expressing a "bonus" preference for the tag t on the first request, changing this to a "penalty" on the second request, then returning to a "bonus" on the eleventh request. t could be

	end-state	decision-average	use-average
(-1,-0.75,0.75)	0.259	0.278	0.283
(-1,-0.1,0.1)	0.149	0.211	0.225

Table 2.4: Correlation between tag-derived preferences and ratings. Two different numeric translations of the tag preferences (avoid, penalty, bonus) are used.

a tag like *hill*, and this pattern could reflect alternation between a weekend preference (*bonus hills for a better workout*) and a weekday preference (*avoid hills for a less sweaty commute*). There are three plausible things we could count to compute tag preferences:

1. **End-state.** Define a user's preference for a tag as the most recent preference expressed for the tag. The intuition is that users may experiment for a while, but have one true preference that they eventually reach. In figure 2.4, only "bonus" would be used to compute the preference for tag t.

2. **Decision-average.** Define a user's preference for a tag by weighting equally each decision to change a preference for the tag. (This and the following definition do not assume that a user has one true preference). In figure 2.4 there are three decisions regarding t: two to "bonus" it and one to penalize it. Thus, "bonus" would be weighted twice as much as "penalty" in computing the preference for t.

3. **Use-average.** Define a user's preference for a tag by weighting how many times it was used in a route request. In figure 2.4, "bonus" was used in three route requests and and "penalty" in nine requests. Thus, "penalty" would be weighted three times as much as "bonus" in computing the preference for t.

Results and Analysis

The results of the correlation between tag-derived preferences and ratings are given in Table 2.4. The correlation is positive, but weak to moderate (results were similar for all numeric translations of tag preferences we used). However, the average correlation obscures individual cases where the correlation is particularly good or bad. Considering such cases is instructive in understanding the relationship between tag-derived preferences and ratings.

Good correlation. There is a block of the Minnehaha Creek Trail that includes the tags *stairs* and *pedestrian bridge*. One user gave the tags *stairs* and *pedestrian bridge* penalties and also rated the block as *poor*. Another user also gave *stairs* a penalty and rated the block as *poor*. In another case, a block along the Kenilworth Trail with the tags *bikelane* and *bike path* was rated as *excellent* by six users who also gave these two tags a bonus. It is probable that the qualities represented by the tags in these two examples influenced the user's preferences for the blocks.

Poor correlation. There is a block of Dale St. that has the *bikelane* tag. From the pool of users who rated that block, only *bonus* was assigned to *bikelane*. But most users rated the block as *poor* or *fair*, while nobody rated it as *excellent*, resulting in a strong negative correlation between the tag-derived preferences and the ratings. A possible reason for this can be inferred by looking at a note attached to this block: *Surface of road is very bumpy and full of pot holes. Watch where your tires go!*. If the tags *bumpy* or *pot holes* had been added to this block, the tag preferences might have reflected the user ratings much better! Therefore, this correlation failure seems to be caused by missing data. A second reason for poor or negative correlation may be implicit priorities among tags: for example, a user might normally avoid *unpaved* bike paths, but will make an exception if they are *scenic*. Finally, some tags like *construction* reflect time-limited properties of a block, but we do not see evidence of users changing their ratings to reflect such temporary changes in the bikeability of a block.

2.6.3 Results Summary

I showed that tags and notes have significant overlap in content. Rather than seeing this as wasteful redundancy, I view it as confirmation that tags and notes have distinct utility: tags provide quick descriptions of objects and let users fine tune the route finder; notes provide detailed information useful for learning and evaluation. In addition, there was a positive correlation between user preferences expressed directly as ratings and derived from the use of tags in route requests. I further identified situations where the correlations were particularly positive or negative. I suggest follow-up research on both sets of results at the end of this chapter.

Role	**Count**	
consumers	379	89%
producers	26	6%
non-specialists	21	5%

(a) tagging role

Technique	**Count**	
notes	52	39%
tags	39	29%
non-specialists	43	32%

(b) annotation technique

Technique	**Object**	**Count**	
notes	blocks	170	84%
	points	14	7%
	non-specialists	18	9%
tags	blocks	36	50%
	points	23	32%
	non-specialists	13	18%
notes & tags	blocks	183	77%
	points	32	13%
	non-specialists	23	10%

(c) annotation object

Table 2.5: Specialization in annotation behavior.

2.7 RQ 3. Specialization

The findings for the previous research question illuminate the overall relationship among the use of tags, notes, and ratings. However, I also investigated similar issues from the perspective of users. Specifically, we examine how users balance their tagging role (producer vs. consumer), public annotation technique (tags vs. notes), and annotation object (blocks vs. points): are they specialists or generalists?

2.7.1 Tagging role: producer vs. consumer

Most users of open content systems only consume information, and only a small minority actually contribute it. I examined whether this was true for Cyclopath tagging. Specifically, I looked at tagging actions concerning blocks – either applying tags or using tags to express route finding preferences.[2] I considered only users who made at least 5 tagging-relevant actions with blocks – any combination of applying tags and using tags to specify route finding preferences. I categorized users as specialists if at least 75% of their tagging-relevant actions was either production or consumption. Table 2.5a presents the results. As expected, the vast majority of users (89%) were information consumers. However, it was interesting to see that the majority (55% or 26 of 47) of the remaining users were production specialists; they produced value while taking little value in return. This finding is related to work on social roles in online discussion forums that identifies "answer people", users who primarily answer rather than ask questions [51]. However, prior research did not have access to private user behavior such as message reading; thus, it is possible that "answer people" received value from reading answers even if they didn't pose many questions of their own. Because we have access to the entire Cyclopath usage history, we can analyze behavior like message reading or preference setting that is typically unavailable to researchers [6].

2.7.2 Annotation technique: tags vs. notes

Tags and notes are both publicly visible, shared annotations. The previous section showed that both techniques are used, and that the introduction of tagging decreased the use of notes. But do individual users combine both techniques? Or do they specialize?

I compared tag applications and note edits made after tags were added to Cyclopath. I considered the set of users who had made at least five public annotations (tag applications or note edits) and defined users as specialists if at least 75% of their annotations were either tags or notes. The results are shown in Table 2.5b. I make several observations. First, a majority – 68% – of users specialized in one technique, with note specialists outnumbering tag specialists 52 to 39. Second, a healthy proportion of users edited both notes and tags. This may be because both types of edits are easy to make, and they serve complementary purposes, as I sketched in Table 2.1 in Section 2.3. Finally, I did followup analysis to investigate how the introduction of tags into Cyclopath affected users who had edited at least 5 notes prior to this time. This gave an utterly mixed picture. The vast majority of such users (99 out of 122) had become inactive by the time tags were introduced. Of the other 23 users, 8 remained note specialists,

[2] I did not examine tagging actions concerning points because I did not log point filtering behavior.

8 became tag specialists, and 7 became generalists. Additional work is required to understand what drove individual decisions to retain an existing practice or adopt a new one.

2.7.3 Annotation object: blocks vs. points

Finally, I wondered whether users preferred to annotate one type of object, either blocks or points. As in the previous analysis, I considered users who had a total of at least 5 tag edits or note applications. I defined users as specialists if at least 75% of their annotation were associated with either points or blocks. I did this analysis three ways: with tags alone (i.e., only for users who applied at least 5 tags), with notes alone (only for users who edited at least 5 notes), and with tags and notes in combination (counting users whose total of note edits + tag applications was at least 5). Table 2.5c presents the results and lets us make several observations. First, all three analyses reveal that a high proportion of users are specialists. Second, there was a noticeable difference in the relative proportion of point and block specialists when I only considered notes and when I only considered tags. For notes, 92% of specialists (170 out of 184) specialized in blocks. For tags, the ratio is much more even: 61% of specialists (36 of 59) specialized in blocks.

2.7.4 Results Summary

I found strong specialization in usage of the public annotation techniques tags and notes. (a) Most users consume annotations; only a small minority produce them. This result is consistent with and extends prior research on open content systems. (b) Most users who do annotate favor one technique, with notes more popular. (c) Users also specialize in their choice of objects to annotate, and most of those who specialize are block specialists. We speculate that this is because blocks play a central role in Cyclopath (they are the building blocks for routes), and there are many more blocks than points (over 150,000 vs. about 2,400).

2.8 Implications

This work raises several interesting followup possibilities.

First, there are more opportunities to analyze the ecosystem of Cyclopath annotation techniques, notably deeper exploration of users' route preferences and how they express them. User changes to their tag preferences and ratings form an intriguing context for studying these questions. I hypothesize several different types of preference changes:

1. A more-or-less **permanent change** in preference: for example, as cyclists gain experience, they may be willing to ride on roads they previously considered too busy or dangerous.
2. A **temporary change** in preference: for example, a cyclist might avoid bumpy and unpaved bike paths in general, but will seek them out when trying out a new mountain bike.
3. A **change in external conditions**: for example, a traffic detour might turn a previously quiet and easily bikeable road into a dangerous adventure.
4. **Experimentation**: for example, a user might request a route, dislike the results, and then (re-)rate some blocks or modify some tag preferences to try to get a better route.

I conjecture that certain usage patterns indicate different types of preference changes. Do users request a route, change a preference, then immediately re-request the same route? If so, this may indicate experimentation. (I already found that 48% of all tag preference changes occur during route re-requests.) Does a user maintain a preference for a long time, then change it and keep the new one for a long time? If so, this may indicate a true change in preference. Does a user have a preference, change it for a route request, then change it back for the next request (of a different route)? This may indicate a temporary change in preference. Appropriate methods to investigate these issues include both quantitative analysis of logged usage data and user interviews.

Second, my findings suggest a number of semi-automated methods to infer new annotations.

1. **Notes to tags.** An obvious way to infer new tags is to mine the text of notes. The easiest case is when an existing tag is contained within a note. For example, one note in Cyclopath reads: *A nice scenic stretch to include on any route. Very*

nice shoulders and traffic isn't heavy. An already existing tag is *scenic*. Others include *low traffic* and *moderate traffic*. Slightly more sophisticated techniques could be used to infer these tags. Finally, a more difficult (but potentially more useful) step would be to infer new tags, e.g., to infer *pot holes* from the note *Surface of road is very bumpy and full of pot holes.*

2. **Tag preferences to ratings.** Certain tags like *bike lane* and *rough* are used frequently and consistently in route preferences. This usage suggests a pattern for inferring ratings to suggest to a user: *You penalize blocks tagged gravel; there are 39 such blocks that you have not yet rated; would you like to rate them all "poor"?*. Inferring ratings is particularly useful since the more users enter ratings, the better Cyclopath does at finding routes.

3. **Geoinference.** The geographic nature of Cyclopath raises unique and attractive opportunities for inferring tags: infer tags that have been applied to geographically nearby objects, and infer tags that have been applied to topographically connected objects (e.g., other blocks of the same trail or road).

All these methods are fallible. Thus, we would need to experiment with mixed-initiative dialog models for users to monitor and approve/reject inferred tags. At least two possible models are attractive. *Basic wiki model*: Inferred tags are applied automatically. Users monitor for changes using normal wiki techniques (watch regions, recent changes) and then revert inappropriate tags. *Work Hints Model* [8]: Inferred tags are maintained as potential tag applications and presented to users through a visual interface. Inferred tags are applied only if a user approves.

2.9 Summary

We introduced tags into the Cyclopath geographic wiki for bicyclists. We chose the relatively rare design option of making tags owned by the community (broad scope) rather than individual users (narrow scope). We made this choice because it is consistent with the wiki model and to promote the goal of evolving a tag vocabulary that would be useful to the entire community. We also modified the Cyclopath route finder to let users use tags to fine tune their routing preferences.

I studied the first year of tagging behavior in Cyclopath, finding that: (a) two quite distinct vocabularies emerged, one around geographic points and one around blocks in the transportation network; (b) while tags have evolved a distinct usage pattern, their use has some overlap with the use of other annotation techniques, and this raises interesting design opportunities; (c) user annotation behavior is highly specialized.

Most relevant to the goal of this thesis, the correlation between tag preferences and ratings was an indication of the potential of tags for specifying preferences for roads. Tags give users an easy way to specify these preferences for a large number of blocks quickly. Instead of users having to tell the system which blocks they like or dislike, they are able to say which general properties of blocks they like or dislike, resulting in more powerful route personalization. The number of block properties available evolves naturally with the needs of the community thanks to the shared nature of the tags.

Chapter 3

Improving Map Data through Conflation of GPS Traces

3.1 Introduction

As GPS-equipped devices become more common, larger amounts of GPS trace data become available to geographic applications. An important challenge and opportunity for these applications is to make sense of all of this new geographic information. Integrating GPS data into these applications can enhance understanding not only of the map and its structure, such as road networks, but also about users, such as their behavior and familiarity with different parts of the map.

One useful technique to help make sense of GPS data is map matching. Map matching consisting of finding a corresponding block in a map for a GPS observation. This lets us situate and analyze GPS data in the context of a road network. Researchers have used this technique to help them study characteristics of a map, such as traffic data [54] (how many vehicles travel through a specific road segment?), and characteristics of the users, such as route choice models [55] (which blocks do users prefer to travel through?).

Unfortunately, while rich GPS data is often available, maps themselves are often incomplete. Roads get built and removed. Maps designed for one purpose, frequently driving, may not contain information about shortcuts, sidewalks, bike trails, and alleyways that are crucial for other purposes, such as bicyling or walking. And many times, map data has simply not been gathered yet, as is often the case with VGI (Volunteered

Geographic Information) in maps such as OpenStreetMap [56, 57, 58].

This can greatly limit the usefulness of map-matching. For example, suppose a bike path next to a busy road is missing. Traffic data studies would incorrectly include bike traffic as part of the main road's traffic. Route choice model studies could also incorrectly reach the conclusion that cyclists prefer to ride on busy roads. If we simply try to map every observation to existing blocks, we can end up with many incorrectly matched blocks or with observations that do not correspond to any block. In order to overcome the limits of matching to incomplete data, we need to be able to account for missing road segments.

The ability to find missing road segments not only improves the map-matching process, but consequently improves all processes that make use of map data, including route finding. In the previous example, route finding systems, even those focused on cyclists such as Cyclopath, would either route cyclists through that busy road or route them through an alternate, less optimal path.

In this chapter, I present extensions to an HHM-based algorithm for matching traces to maps. This algorithm is able to detect missing blocks and add them as necessary. The basic concept is to match when possible and build when necessary.

3.2 Related Work

3.2.1 Map matching

The most common algorithms for matching GPS traces to blocks are geometric and topological algorithms. These range from the very simple, considering only the distance from each observation to each block, to the more advanced, taking other factors into account such as perpendicular distance to block and links between blocks to ensure a continuous route. Recently, a popular technique has been the Multiple Hypothesis Technique (MHT) [59, 60], which maintains a list of multiple possible routes and at the end of the sequence of observations selects the route with the best score. For an overview of many of these map matching algorithms, the reader is ecouraged to read the summary by Quddus et al. [61].

Another recently popular technique for map matching has been using Hidden Markov Model (HMM) algorithms [62]. These algorithms have been shown to be robust even in the absence of high-precision GPS data, which is often the case with handheld devices. This technique treats road segments as states with transition probabilities between each other and maps the sequence of GPS traces to the most probable sequence of HMM states. As described in Hummel's work, this approach also inherently provides means for detecting u-turns and erroneous map topology. Others have expanded on this algorithm by using Viterbi to decode the best sequence and adding travel time constraints [63] and by handling outages [54]. Although HMM provides the potential to detect missing map data, these implementations fail to address that problem. The algorithm presented in this chapter is based on this approach.

3.2.2 Map building

There exist many algorithms to infer road networks from GPS traces, including [19] and [20]. These algorithms are able to produce precise information (such as lanes and intersections) from high-precision GPS data. Some require the whole set of GPS traces and others are able to build the map incrementally as more GPS data is added [21].

As I mentioned in the introduction to this thesis, these algorithms have two significant limitations: the difficulty of obtaining sufficient amounts of high-precision GPS data and the lack of the ability to work in the context of already-existing road network data. My algorithm, however, is able to work both with low-precision data (by using a probabilistic model) and with existing road network data (by trying to match when possible).

Algorithms that work with existing road networks are usually referred to as map refinement algorithms. Some of these are used to build lane information for existing digital maps [64]. These, however, also require high-precision data and may not actually add missing road segments, but focus only on adding information to those already present.

3.2.3 Mobile Sensing and Biking

Biketastic is another relevant project [65]. Biketastic is a mobile sensing application that aims to make route sharing and experimentation a richer experience. Similar to us, they use GPS trace data to improve their application experience. One key difference,

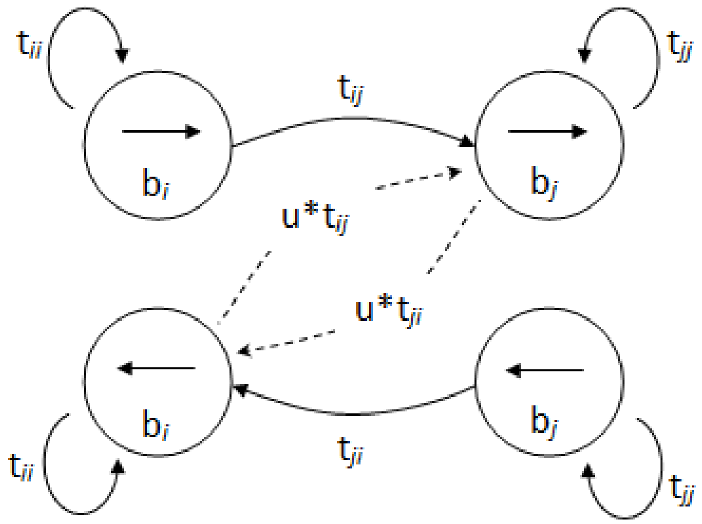


Figure 3.1: An example of an HMM process. Nodes represent blocks in the road network. There is an edge from one node to the next if it is possible to travel from the first node's block to the second node's block (that is, the two blocks are connected by an intersection). In the HMM, t_{ij} represents the probability of a transition from b_i to b_j, and u represents the probability of backtracking (making a u-turn).

however, is that they do not match the GPS traces to the road network algorithmically. Instead, they simply smooth the GPS data and overlay it on the map, leaving it up to users to do the matching visually.

3.3 Map-matching using HMM

My map-matching algorithm builds on the approach used by Thiagarajan [54]. This approach uses HMM to model road segments as hidden states that produce visible observations (GPS traces). Figure 3.1 shows an example of an HMM process in a road network. In an HMM process, we know what the resulting observations are, the probability of a state producing a certain observation (the emission probability), and the probability of a state transitioning to another state (the transition probability). The idea is then to find the most likely set of states that could have resulted in a given set of observations. Thiagarajan describes the advantages of this approach in the following way: “...it is robust to position samples that lie closer to one road segment than the one from which they were observed, and is able to capture the idea of a continuous route instead of a sequence of segments.”

In this section I provide a brief overview of the settings for running map matching using HMM and Viterbi decoding. For a more detailed description of these settings, I encourage you to take a look at Thiagarajan’s paper.

When applied to road networks, the emission probability for a given road segment and GPS observation is often modeled as a normal distribution based on the distance from the observation to the road segment. In my tests I use a standard deviation of 10m. For transition probabilities between road segments, similar to Thiagarajan, I set the probability of transitioning to a connected block or staying on the same block to 1/(dmax+1), where dmax is the maximum out-degree of the transportation graph.

In addition, I allow U-turns by multiplying the transition probability by a backtracking probability for blocks which we just transitioned from. This is denoted in Figure 3.1 by u. This is especially useful when working with GPS traces recorded by cyclists, given that often cyclists’ rides are exploratory in nature (and it is also easier to U-turn with a bike than with a car). The bigger the backtracking probability, the higher the possibility that the algorithm will consider that the user backtracked. To avoid accidental matches to incorrect nearby blocks, such as in Figure 3.2, I keep this probability at an extremely low value (currently 1E-20).

Before running the matching algorithm, I pre-process the tracks to remove outliers (observations that would suggest that the user is traveling unrealistically fast). I also discard tracks with less than 10 observations or that do not cover a long enough distance, since these are usually the result of experimenting with the mobile app and not actually proper rides. Once pre-processing is done, I use the Viterbi decoding algorithm to find the sequence of blocks with the highest probability of producing the given observations [66].

3.4 Extension to Find Missing Blocks

Extending the algorithm in the previous section to handle missing blocks means starting with an incomplete HMM and extending it as we go. We need to detect when a new

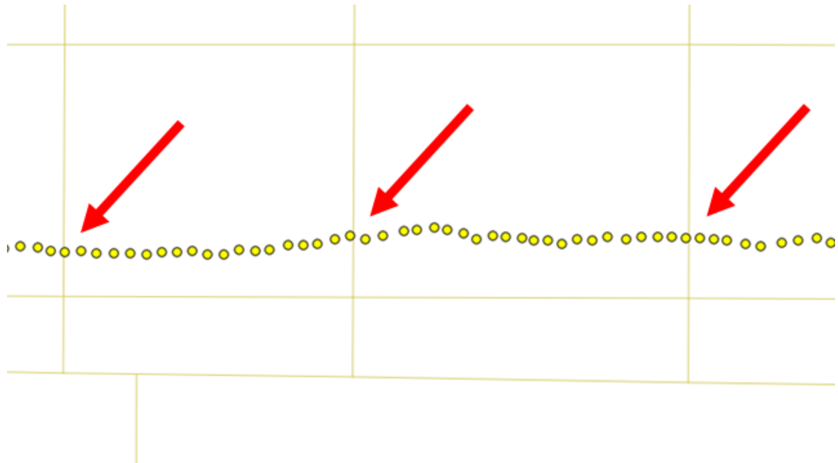


Figure 3.2: In this example from an actual track in our system, there is a high concentration of points close to blocks that are not part of the correct path. If u-turns are not allowed, the algorithm will not match to that block because it has no nearby block to transition to afterwards. But if u-turns are allowed, the algorithm might think that the user took that path and then returned to the main path. In order to avoid these erroneous sidetrips, I set the backtracking probability to an extremely low value, so that backtracking is only considered as a last resort.

state (block) is required to better fit the given observations. In essence, this is a hybrid between map-maptching and map-building: match when possible, build when needed.

3.4.1 Algorithm extensions

The main design challenge for a hybrid matching / building algorithm is to determine the condition that defines when to match and when to build. My approach is straightforward: if there are no blocks that we can transition to, add a new block.

There are two cases in which the condition for adding a new block is fulfilled: (1) all blocks have an emission probability of 0 for a given observation (there are no blocks nearby) or (2) all blocks with an emission probability higher than 0 have transition probabilities of 0 (nearby blocks are not connected to the available paths). In order for blocks to have emission probabilities of 0, I define a cutoff distance d. In other words, only blocks that are within d of a given observation are considered at that step of the algorithm. d essentially becomes a parameter representing how strict we want the algorithm to be.

The original Viterbi algorithm goes through the sequence of GPS observations and for each step calculates the possible paths that could have created the observations up to that point. With my extensions, at each step, if any of the conditions for creating a new block are met, the following actions also take place:

1. It starts with the current GPS observation and move backwards in the sequence of observations until one of two conditions are met: (1) it reaches the first observation in the sequence or (2) it reaches the observation that was last seen within s of a block in the matched path, where s is the standard deviation for GPS error. It is essential that it only stops at a block that the algorithm has visited before. Otherwise, the newly created block may get connected to a block that is not accessible in the path of the algorithm.

2. Once it has the starting observation for the new block, it moves forward until it reaches an observation that is within s of a any block. At this point it has the start and end observations of the new block to be created.

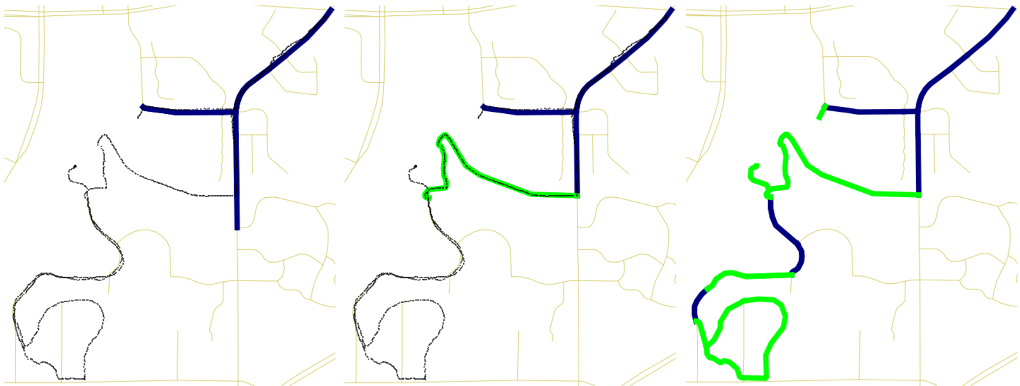
3. The next step is to create a block geometry using the GPS observations selected in the previous steps.

4. It then connects the new block to nearby blocks to ensure that transitions will be possible between them. If the first or last observations in the new block are close to an intersection or end of a block, it simply snaps the end of the new block to that point. Otherwise, it splits the closest block at the point nearest to the GPS observation and creates a new intersection at that point.

5. Afterwards, it simplifies the block geometry to reduce the impact of noise.

6. Finally, after the new block has been created and properly connected, the algorithm is now able to continue with the extended HMM. But because the presence of a new block would have affected the emission and transition probabilities of nearby blocks, it rewinds itself to the first observation where the new block or any blocks connected to it (which might have been split) had any emission probabilities. In other words, it rewinds to the first point where any of the affected blocks were considered as options.

(a) Area of map with incomplete data. (b) Section of GPS observations from a track ridden by a user. (c) Map matching with no u-turns or detection of new blocks.



(d) Map matching with u-turns, but no detection of new blocks. (e) Creation of first new block. (f) Final match. New blocks are shown in green.

Figure 3.3: Map matching and building

7. Once rewound, it continues the matching as usual, until the next time it encounters the need for adding a new block.

3.4.2 Map matching example

Figure 3.4 shows an example of map matching with detection of new blocks. Figure 3.3a shows the map area where the track was recorded. In the Cyclopath map, residential areas such as this one tend to have a significant amount of missing blocks. A section of a track ridden by a user of our system is shown in Figure 3.3b. The track is a good example of a case where the user backtracks and also veers off the existing blocks in our system.

If we do map matching without taking into account u-turns or missing blocks, the algorithm will only match the blocks highlighted in Figure 3.3c. Even if the algorithm can handle u-turns, the result is still not good enough, as shown in Figure 3.3d.

Once detection of missing blocks is enabled, the algorithm is able to add a new block as shown in Figure 3.3e. This new block creates a new intersection at the block where it started and then snaps to end of the first block it encounters. The algorithm can now continue as normal until the next time it needs to create a block.

Figure 3.3f shows the final matched ride. A total of nine new blocks added as a result of my algorithm are shown in green.

3.5 Experiments

3.5.1 Data

I tested my algorithm using the Cyclopath map of the Twin Cities, which at the time had slightly more than 155,000 blocks covering more than 20,000 miles. I used GPS data collected by our mobile Cyclopath Android app. After pre-processing GPS trace information and removing tracks that were too short, I had 128 GPS tracks. Each track had an average of 1450 GPS observations. The average length was 1.75 miles and the average duration was 8.5 minutes, with the longest ride taking 50 minutes.

3.5.2 Finding New Blocks

Methods

I first tested the algorithm by running it on the current map dataset to see: (1) the amount of new blocks it could find and (2) the length of those new blocks. This gives us an idea of how much benefit we could expect from this algorithm in terms of new block information.

The value of the cutoff distance d can significantly affect the results of the algorithm. Therefore, I ran the algorithm with several different values for d. I used multiples of

d	**meters added**	**blocks added**
50	117.1	0.5
40	146.5	0.8
30	207.4	1.3
20	414.7	3.0
10	842.7	10.8

Table 3.1: Average number of meters and blocks added for each track for each cutoff distance.

the standard deviation of the error (10m) in order to facilitate the understanding of the effects.

Results

Table 3.1 shows the results for cutoff values between 10 and 50 meters. As expected, a smaller d means the number of false positives (blocks that *should not* have been created) will increase, but the number of false negatives (blocks that *should* have been created but were not) will decrease.

Although these numbers help give an indication of the usefulness of the algorithm, in order to verify which newly created blocks are false positives and which ones are true positives a gold standard is required.

In principle, we could evaluate the results by visualizing each new block with aerial photos; however, this approach does not scale. In order to better evaluate our algorithm, I decided to try the following idea: remove existing blocks from our map dataset and see if the algorithm can find them with the given GPS tracks.

3.5.3 Tests Removing Existing Blocks

Methods

Existing blocks in our system can serve as a type of truth to aid in testing our algorithm. The idea is that if the matching algorithm matches to a certain block when run normally, if we remove that block, the algorithm should be able to recreate the block from the GPS observations. The accuracy of the algorithm for blocks in our system can be a predictor for accuracy for new, unknown bocks. This is a similar idea to hiding user

	1 block removed		2 blocks removed	
d	**false negatives**	**false positives**	**false negatives**	**false positives**
50	33.5%	5.4%	51.5%	6.0%
40	30.7%	6.3%	48.7%	7.2%
30	25.7%	7.7%	45.1%	9.6%
20	16.1%	9.7%	36.0%	13.8%

Table 3.2: Percentage of test runs with false positives and false negatives.

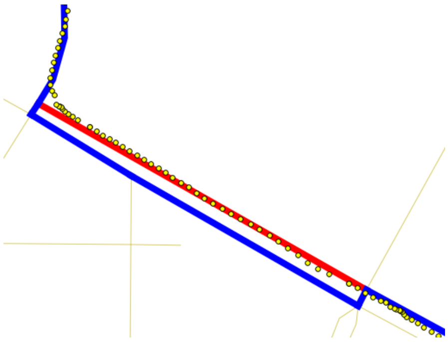
d	**tests**
50	1544
40	1532
30	1501
20	1397

Table 3.3: Number of tests run by removing one block for each cutoff distance.

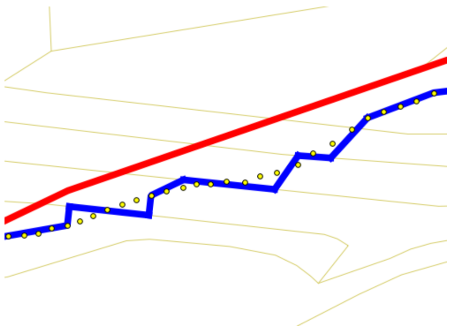
ratings when testing recommender systems [67]. The idea is that performance predicting those ratings serves as a predictor of performance predicting future ratings.

To test this, I first took every matched ride from the previous section and listed the sequence of unique blocks matched. From each sequence, I removed the first and last blocks, as these blocks are often matched on the basis of very few observations and are therefore matches of low confidence. I then removed from each sequence those blocks that were created by the algorithm when matching, since these blocks were created from the GPS tracks I was trying to test. Finally, I ran the algorithm for each track, once for each block in the sequence of blocks to be removed for that track. As an example, if after removing the first and last blocks and blocks created by the matching algorithm I was left with eight blocks, then I would run the algorithm eight times for that track. Each time I would remove exactly one of the eight blocks and see if the algorithm could correctly identify it.

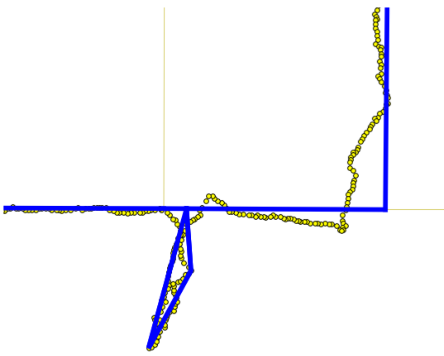
Since more blocks are created when using smaller cutoff distances and I did not test removing those newly created blocks, this means that for a smaller cutoff distance the amount of blocks to test with was fewer. The amount of tests run for each cutoff distance is shown in Table 3.3.



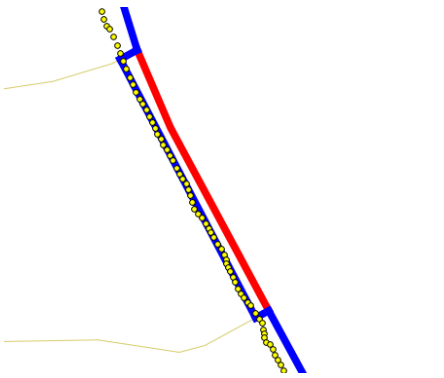
(a) An example of false negatives, where observations match to nearby blocks and the true block is not created.



(b) An example of false positives, where more blocks are created than actually exist.



(c) An example of incorrect block geometry resulting from a user making a u-turn on a new block.



(d) An example of a case where the intersection is offset from the GPS observations.

Figure 3.4: Map matching and building

Results

The results are shown in Table 3.2. I define false positives as test runs where more than one block was created and false negatives as test runs where no blocks were created. As expected, as the cutoff distance is decreased, the amount of false negatives also decreases as the algorithm is able to find new blocks more accurately, but the amount of false positives also increases, as new blocks get added when they shouldn't. I now take a closer look at the causes of false negatives and false positives.

- **False negatives.** False negatives occur when the aglorithm is able to find a different block to match to, such as in Figure 3.4a. The bigger the cutoff distance, the more blocks that will be considered and therefore the higher the probability that another nearby block can replace the removed block in the sequence of matches. One common cause is parallel blocks, such as bike paths, which often follow along main roads. Thus, a motivation for reducing false negatives is to able to find bike-related facilities such as these.

- **False positives.** False positives occur when the algorithm created more blocks that it should have. This is especially the case for overpasses, such as the one in Figure 3.4b. It is difficult to know for certain if an intersection exists at the point where two road segments intersect. Unless the user turns at an intersection, we can't tell if it is possible to do so.

Opportunities for Improvement

Detecting missing blocks is much easier than actually creating them. In this project I did not focus on creating high-quality blocks, so that left the algorithm with some space for improvement in that area.

- **Noise.** One case where the current algorithm could do a better job is in simplifying geometries of new blocks, especially when a lot of noise is involved. This is often an issue when the users wanders around the end of his ride or when the users u-turns on a new block, such as in Figure 3.4c.

- **Intersections.** As discussed earlier, correctly creating blocks near intersections is also an issue. This is the case not only for overpasses, but also when the intersection is slightly offset from the observations, such as in Figure 3.4d.

- **Bias.** Another issue to keep in mind when creating new blocks is that new geometries are biased towards the first observations used. In most map building algorithms, new observations are integrated with older observations to create the final blocks. But in this algorithm's case, it creates new blocks using only information from the current track.

d	**tests**
50	1420
40	1409
30	1378
20	1277

Table 3.4: Number of tests run by removing two blocks for each cutoff distance.

The advantage of a geowiki is the fact that it can leverage user input in order to make up for many of these deficiencies. Our system could potentially ask users to fix geometries created with noisy GPS data, decide whether a new block intersects or not with older blocks, and update old geometries based on new GPS data.

Users can also do a better job of detecting false positives and false negatives. The system could potentially allow users to interactively adjust the cutoff distance in order to change the amount of new blocks created by the algorithm.

3.5.4 Removing more than one block

Methods

In order to get an idea of how the algorithm might perform in situations with less data available, I tried the same technique as in the previous section, but removing two blocks in a row instead of just one. The amount of tests run is shown in Table 3.4 and the results for number of tests with false negatives or false positives are shown in Table 3.2.

Results

Both false positives and false negatives increased significantly when I increased the amount of missing blocks by just one. This is an indicator that the algorithm does not necessarily scale well for map data sets with too much missing data. This results from using a single set of GPS traces to create a representation of the road network.

3.6 Discussion

3.6.1 Using GPS Data

My main goal in this project was to use GPS data to improve the Cyclopath map and, consequently, route finding. GPS data itself (without matching to the road network) can already potentially be used to:

- aid geographic editing by overlaying traces on the map,
- elicit contributions from users when they are at the actual locations, reducing the need to recall information later on, and
- learn what areas of the map users are familiar with (which also helps us to elicit work from users more effectively [8]).

However, by matching GPS traces to blocks, we could better analyze riding patterns and preferences and detect missing blocks automatically.

3.6.2 Adding Human Component

Automating the map matching process can make editing the map easier and less intimidating for users. However, as seen in the results above, human intervention is still helpful and even necessary. Issues such as finding the right distance cutoff, identifying false positives and false negatives, and fixing geometry of new blocks are tasks that humans could perform.

As part of this effort, Yanjie Liu created an experimental interface that allowed users to view suggested new blocks and help confirm their geometries [68]. The details of the implementation and tests are outside the scope of this dissertation, but you are encouraged to read Liu's thesis if you are interested in this follow-up study. The main idea is that the map editing process was indeed improved and in many cases required only one click to add new blocks to the system.

3.6.3 Using Matched GPS Data

Now that we are able to match GPS traces to blocks, features that take advatange of this relationship can be built. Here is a list of possible uses for matched GPS data:

- **Riding preferences**. We could compute what types of blocks users generally prefer to ride on. We could learn a user's preferences and use them to compute better routes for them in the future.
- **Route sharing**. Currently users have to manually create routes that they want to share. If we can match traces to blocks, we could generate routes automatically and make sharing easier.
- **Cycling log**. We could create tools that allow users to track and visualize their cycling history and compare multiple rides through the same route.
- **Campaigns**. We could create and track campaigns that ask users to ride certain routes. This could be done to get users to become familiar with an area of the map that needs editing or it can be something as simple as a race to help build community.

3.7 Summary

In this chapter, I presented an extension of an HMM-based map matching algorithm for the purpose of not only matching GPS observations to blocks, but also detecting and creating missing blocks. I then tested my algorithm on the Cyclopath map with data gathered through a mobile extension of the Cyclopath geowiki.

The new algorithm was able to find significant amounts of missing data in our map, even when cutoff distances were set to conservative values. When testing to see if the algorithm could detect current blocks in the system if they were removed, I confirmed that there was a trade-off between false negatives and false positives as the cutoff distances were varied. I also found that this algorithm degrades as the amount of missing data is increased. Finally, although missing block detection works well, the geometry creation part of the algorithm still needs to be improved in order to better handle noise and intersections.

With the rising popularity of systems that depend on Volunteered Geographic Information, it is important to develop algorithms that can handle missing map data. This chapter provides one such bridge between map matching and map building algorithms. It is ideal for geographic applications that are in constant evolution, such as geographic wikis. Using a combination of a user's implicit data (GPS traces) and explicit input (verification of suggested missing blocks), maps such as ours can be improved and route finding be made more precise.

Chapter 4

Improving Route Presentation through Crowdsourcing of Landmarks

4.1 Introduction

So far I have discussed methods for improving the inputs to the route finding process. By providing richer and more complete map data and by allowing users to specify more preferences, we can produce routes that are more accurate and reflective of a user's optimal route. However, even if we find the perfect route, it might be hard to understand or remember. We might be including a lot of redundant information (such as sections of the route that a user already knows) and excluding information that could help make the route directions more natural.

In a study of Cyclopath in China [69], detailed later in this chapter, I noticed how landmarks played a key role in cyclists' route finding and map browsing processes. With this observation and the goal of improving route presentation in mind, I decided to explore the use of landmarks in route directions. Specifically, how can we make use of Cyclopath's unique crowdsourcing capabilities to improve the presentation of route directions?

I begin this chapter with a description of the user studies we performed in China and the findings that arose from that study, in particular those related to the use of landmarks. I follow with an overview of landmarks in general and within Cyclopath, including challenges such as how to represent, acquire, and validate landmarks. Finally, I detail a number of experiments I did in order to test various techniques for crowdsourcing landmarks and discuss their results.

4.2 Study of Cyclopath in China

Cycling is one of the most common forms of exercise, recreation, and transport around the world. Yet the geographic, structural, and cultural characteristics of a location can significantly impact the cycling experience. So how do these differences in turn affect geographic crowdsourcing applications? What is their impact on the success of such applications? Although the importance of culture in Human Computer Interaction has been recognized, this is a research area that is still young and developing [70, 71]. Most studies of culture in HCI have focused on user interface usability issues, but cultural effects on open collaboration systems introduce even more challenges beyond usability issues due to the social interactions between users. In order to explore these questions further, I took Cyclopath, a geowiki for cyclists in the Twin Cities and, with the help of Chinese colleague Yanjie Liu, created a Chinese version for cyclists in Dalian, China.

Despite the U.S. and China's cultural differences, they are two of the most populous and economically powerful countries in the world. For this reason, China has been a common point of comparison with the US in research involving online cultural issues and differences [72, 73, 74]. Most studies have focused on communication between Chinese and American collaborators [75, 76, 77, 78, 79]. In my case, I was particularly interested in the issues of self-disclosure [80] and authority [81] to see if there existed a clash between culture and open collaboration online.

4.2.1 Twin Cities and Dalian

The relatively flat Twin Cities metropolitan area covers about 8,000 km^2 and has a population of about 3.3 million people. Even though the winters are cold and cycling numbers drop during those months, Minneapolis has been named No.1 bike city in the United States by Bicycling Magazine [82] and Minnesota the No.2 bike-friendly state by the League of American Bicyclists [83].

Located in the northeastern province of Liaoning in China, Dalian is a mountainous region covering an area of about 13,000 km^2 and with a population of almost 6.2 million people (almost 3.6 million in urban areas). Although not as cold as Minnesota, Dalian also experiences seasonal temperature differences that affect the cycling experience. Near the beginning of the 21st century, Dalian's infrastructure was changed in favor of cars, moving cyclists to the pedestrian space [84]. Its heavy traffic and poor cycling infastructure therefore make Dalian a not very bike-friendly city. Dalian's different culture, geography, and infrastructure and its similar area, population, and seasonal changes made it an ideal city to compare with the Twin Cities.

4.2.2 Research Questions

RQ1. Contributed Content. What type of content would users in Dalian contribute and how would it be different from the content contributed by users in the Twin Cities? This would help us better understand the cultural, geographic, and infrastructure differences between the two cities.

RQ2. Usage Patterns. Would there be any differences in usage patterns, such as when browsing the map or requesting routes? Significant differences when interacting with the system could suggest design changes that need to be made to better support Chinese users.

RQ3. Authority and Trust. Would there be any evidence of the role of authority in the system and how it affects trust? The hierarchical and collectivistic nature of Chinese culture may affect users' trust of peer-produced content and may lead to a desire for greater authority, crucial issues for crowdsourcing applications.

4.2.3 Methods

To prepare Cyclopath for Dalian, we translated over 800 English phrases in the Cyclopath UI and populated the road network with map data imported from Open Street Map. Given that this was an experimental study, some features such as elevation data and aerial images, were not included. Figure 4.1 shows a screenshot of the final interface.

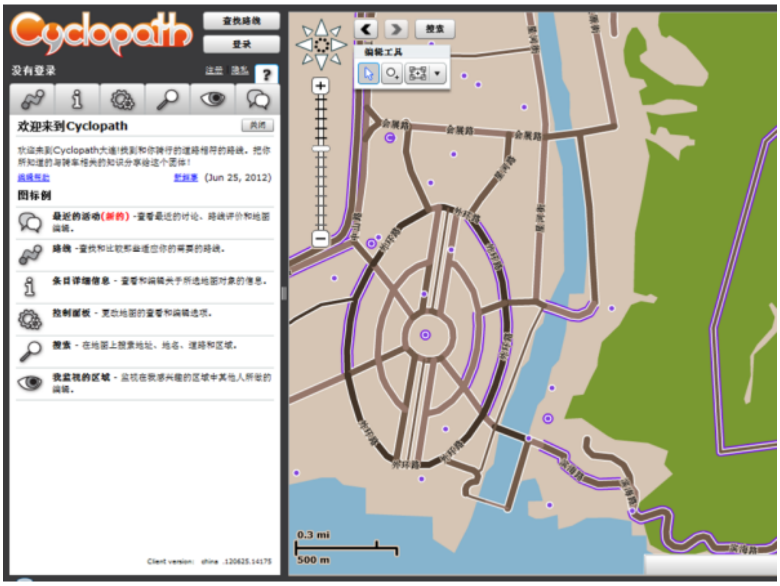


Figure 4.1: Chinese version of Cyclopath UI

With the help of the Sino-European Usability Lab at Dalian Maritime University, we conducted user studies of about one hour with 40 cyclists in Dalian. The studies were conducted in Chinese by a Chinese member of the research team while I watched from a separate observation room. This was done in order to avoid bias that results from a foreign researcher conducting the study, where the native users in some cultures might feel more inclined to hide negative feedback.

User studies consisted of a set of interview questions and tasks. We began with some ice-breaker questions about a recent cycling trip, followed by questions related to riding behavior, their involvement in the local cycling community, and their experiences with online communities.

This set of initial interview questions was followed by the tasks portion of the study. This portion was divided into four main tasks, each followed by a small set of questions about the task:

T1. **Entering points.** Subjects were asked to enter four points of interest on the map. These points would be visible to subsequent subjects.

T2. **Route-finding.** Subjects had to request a route of their choosing and give us feedback about the computed route.

T3. **Adding tags and notes.** Subjects were asked to enter four tags and four notes. These annotations could be done on blocks, points or a combination of both. As with points, tags and notes would be visible to subsequent subjects.

T4. **Adding bikeability ratings.** Subjects had to rate the bikeability of at least 10 blocks, on a five-point scale ranging from *Impassable* to *Excellent*.

After completing these tasks, users were asked a set of questions about their experience with Cyclopath, motivations to use the system, potential benefits gained from using the system, anonymous use of the system, and trust of information contributed by others.

We recruited participants with the help of local bike clubs. Subjects were 18 or older and had cycled at least three hours or 20km during the year preceding the study. Participants were compensated in cash for their time.

Half of the 40 subjects were students. 36 subjects were men and 4 were women. 12 subjects had more than 5 years of experience cycling in Dalian. When asked about why they cycled, 38 said they did so for recreation purposes, 22 said they cycled for training, and 16 selected commuting as one of their reasons.

4.2.4 Results

RQ1. Contributed Content

Subjects in the study entered a total of 168 points, 98 tags on blocks, 81 tags on points, 83 notes on blocks, and 78 notes on points. I manually coded this data and compared it to the results of previous US studies [5, 85]. The following were the key factors I found that affected the types of data contributed:

Geography. Given that Dalian is a mountainous region, as opposed to the generally flat Twin Cities, many more notes on blocks were descriptions about slope (22.5% of all block notes). Additionally, 18.1% of all notes on blocks related to scenery. This is in part due to Dalian's many mountain and sea views. For many of our subjects, whose primary purpose for cycling was recreation, thinking about which routes would be most enjoyable often led them to think about geographical features and sceneries.

Culture. As an example, Dalian subjects did not enter points related to arts (compared to 10.9% for US users). This may be in part because of a noticeable subculture in the US of people who value both arts and eco-friendly activities such as cycling. However, it was not uncommon to find points related to culturally popular places such as Karaoke venues.

Landmarks. What serves as useful landmarks in Dalian is different than in the US. For example, Dalian users entered fewer points related to food (16.8% vs. 36.4%). Restaurants in Dalian tend to be numerous, but with similar offerings and small in size. This makes it harder to remember restaurant names and makes them less useful as landmarks. In contrast, subjects entered more community resources (43.1% vs. 17.1%), such as hospitals and schools, presumably because they served as useful landmarks.

Cycling infrastructure. While the amount of cycling-related data entered by users was similar to the US studies, the type of content was often different, reflecting the different cycling infrastructures of both cities. For example, there were no mentions of bike lanes in Dalian, even though that is one of the most popular tags in the US system. Bike racks, which accounted for 12.7% of all notes on points in the previous US study, also went unmentioned in Dalian. Bike racks are harder to come by in Dalian, so cyclists might be used to other alternatives for securing their bicycles.

Discussion I found that users in Dalian have a significant amount of cycling-related knowledge to contribute. But there were many cases where the type of data contributed was quite different from that in the US. For example, marking bike racks was not as important for cyclists as specifying properties of road and trail segments like slope and scenery. These differences highlight the utility of crowdsourcing systems such as Cyclopath, where local knowledge is leveraged to provide users with information and insights that are relevant to them. Attempting to control or limit the type of information that users can contribute could hinder the capture of knowledge concerning geography, culture, and infrastructure required to meet user needs.

RQ2. Usage Patterns

As we conducted our studies, we observed an interesting pattern in the users' interactions with Cyclopath and answers to interview questions: *landmarks* played a major role for navigation and route finding.

One of the first signs of these patterns was that when asked to add points of interest, most users would go for the same well-known landmarks. Afterwards, they would tend to add new points around that landmark. Users expressed the importance of landmarks for finding (or adding) other nearby landmarks:

"There are no landmarks. I have to zoom in enough to see the small characters on the points. But in Baidu Map, I can see the name on those landmarks even when it is in low zoom level. For example, if I want to add a point near Dalian Maritime University (DLMU), I'm used to finding DLMU first."

"If there is no other point around the point I want to add, it will be a little difficult for me. Because sometimes I can't remember the name of the blocks but can remember the bus stop signs near it and some landmarks."

The second sign was the low use of addresses for requesting routes. Most users didn't know the addresses of their desired destinations and so preferred to route to nearby recognizable landmarks. In order to better facilitate this, we actually switched the order of the tasks so that users could add points first, whereas the original script had users asking for routes first. Users expressed to us the utility of landmarks when getting directions and finding new routes:

"Adding points is important. The map is currently not very full of points. When I go out if I can't find landmarks I can't find my way."

"If I want to ride on a brand new route, I will check the map to find some landmarks before I start off."

Discussion Navigation patterns have been found to vary from location to location depending on the cultural, geographic, and infrastructural properties of the place. These patterns can even vary within the same country [86]. Dalian cyclists' use of landmarks is evidence of infrastructural and geographic differences between Dalian and the Twin Cities. The smaller amount of grid-like road networks in Dalian, which is often a result of roads going along mountains, rivers, and similar large obstacles, result in a need for more navigation aids.

The use of landmarks in Dalian has important design implications for Cyclopath. First, cyclists use landmarks to orient their exploration of the map; however, in the current UI, when the map is zoomed out, main roads, bodies of water, and green spaces are shown – but no landmarks. Therefore, modifying the display to make some landmarks always visible, especially when zoomed out, would be useful. Second, landmarks should be used when showing computed routes, both displayed as part of the route visualization on the map and included in step-by-step text instructions, e.g., "Turn right after Xinghai Square".

RQ3. Authority and Trust

Hofstede's culture model, which categorized cultures based on five dimensions [87], is frequently used to help explain cross-cultural differences. The dimensions *Individualism* and *Power Distance* seem particularly relevant to the subject of authority and trust in an open content system like Cyclopath. Individualism refers to the degree to which a culture emphasizes an individual's reliance on the self. Collectivistic cultures such as China might be more motivated to contribute to crowdsourcing applications for the benefit of the community (or by social pressure), but may also avoid contributing for fear of disrupting the harmony of the group. Power distance refers to the extent to which less powerful people in a society accept inequality of power and consider it normal. Users from hierarchical societies might have different expectations of the role of authority in such systems.

Given the traditional respect for authority and hierarchy in Chinese culture, I was interested in seeing how this affects users' trust within Cyclopath. Because more extensive and real-world usage of the system is required to truly dive into these issues, a direct comparison with the US site was not possible. Therefore, I focused on finding preliminary evidence of these factors during our interviews.

When thinking about online trust in this section, I do not focus on how our site affects users' perception of trustworthiness about the site, but on how our system supports trustworthy behavior [88]. I am interested in knowing if users are willing to consume information contributed by other users. As in all situations where trust is required, there is a certain risk and uncertainty involved when doing so.

When asked whether they trusted content contributed by other users, only about a third of users (32.5%) in our study did. If we look only at subjects who were students, this increases to 45%, suggesting that there might be a higher tendency to trust contributed content among students, who are generally younger and more exposed to online content.

Authority Trustworthy behavior can be incentivized by the existence of authorities who supervise contributions and sanction behaviors that decrease trust. Comments from a number of subjects touched on the theme of authority, specifically through a desire for content contributed by professionals and for administrators to play a role.

> *" I'm worried about the limits of authority. An administrator should supervise the changes made by users."*

> *"I hope to have content written by both official organizations and the users."*

Reputation Another mechanism that can motivate trustworthy behavior is reputation. As explained by Riegelsberger, reputation mechanisms require stable identities and the traceability of actions. Stable identities are supported in Cyclopath through usernames (although users are not forced to log in in order to contribute). Because Cyclopath is a wiki, all past contributions of any user can also be traced. Although not an explicit reputation measure, users' work histories serve as reputation builders and evidence of past, trustworthy contributions.

Given that it is harder to establish reputation for anonymous users, we asked participants whether they trusted content contributed by anonymous users. Half of the users said they would trust content contributed by logged-in users more, but the reasons for this varied:

Experience. Some users correlated logging in with experience. To them users who have used the system for a long time will naturally decide to log in.

> *"I trust logged-in users better because they have used the system longer."*

Accountability. Some users felt that logging in gave users a sense of responsibility for their edits that anonymous users didn't have.

> *"I believe logged-in users more, because they can have a sense of responsibility "*

Community. Anonymous usage affects the sense of community among users, something very important in a collectivistic country such as China.

"I trust logged-in users more than anonymous users... When you conmmunicate with a logged-in user, it is like you are communicating with a person. But when you communicate with an anonymous user, it feels like you are communicating with a computer."

Discussion So how can we motivate users to build trust? First, we need to help users trust peer-contributed content, especially by anonymous users. Second, we need to motivate users to log in so that their contributions are trusted more.

Techniques such as feedback, points, and ranks are good ways to not only motivate users to log in and use the system, but also increase other users' trust. Making user rank and experience more transparent helps support the Chinese views on hierarchy and thereby improve trust. Users were already thinking about these potential solutions during our interviews:

"I hope to have a function to rate on other user's comments, just like on Taobao. We usually trust the seller who has a high prestige."

"Just like forums, the system can provide some credits to motivate the users to post more content."

4.2.5 Implications

We can derive various design implications, especially important when designing across different cultures or groups, from the results of this study.

Contributed Content The differences in contributed content give us insight into what is important to locals and reminds us of the importance of local knowledge for crowdsourcing applications and the dangers of limiting contributions too much.

Usage Patterns Information that is more important to a specific group of users, such as landmarks, should be made more salient.

Authority and Trust When designing for different cultures, the importance of privacy and trust must be taken into account. This can affect whether the system requires users to log in, encourages them to do so or avoids it altogether. It can also lead to differences regarding how transparent to make user history and contributions in crowdsourcing systems.

The fact that landmarks played important roles in both type of contributions and usage patterns (browsing and navigating the map) motivated me to further explore the role of landmarks in our system. How could we better support their use in Cyclopath? How could we improve route directions using landmarks? Before detailing my approach to integrating landmarks into Cyclopath, I first give a brief description of landmarks in the context of route finding.

4.3 Landmarks

Landmarks can be defined as cognitively salient, prominent features in the environment [89] that often serve as reference points. They help us in the development of mental representations of the environment [90] and in the communication of route directions [30, 29]. They are essential tools for navigating the world around us.

Navigation landmarks can serve different purposes during the traversal of a route, as described by [91]. They can be used at decision points, where reorientation is needed. For example, "Turn right after the gas station". They can also serve as route marks, confirmation of going the right way. Think of "You will pass a middle school on your left" or "If you see a McDonalds, you went too far". Finally, distant landmarks, such as mountains, are used for overall guidance, fulfilling a compasslike role.

In order to determine which landmarks are more useful than others, a model of landmark saliency, such as the one defined by Raubal et al.[31] can be used. In their framework, landmark attractiveness depends on the landmark's visual, semantic, and structural properties. Visual attraction is determined by attributes such as shape, color, and visibility. Semantic attraction is determined by factors that give the landmark meaning, such as cultural and historical significance. Structural attraction results from the role or location of the landmark in the structure of the environment. For example, intersections (such as a T-intersection) and boundaries (such as a river or highway) carry structural significance.

The benefit of using landmarks to make route directions richer and more natural to users should be evident. However, obtaining landmarks information that is current, useful, and relevant can be difficult.

Crowdsourcing landmarks is a relatively new solution to gathering landmark information. Previous research has used geo-referenced photos to extract landmark information [92]. Another research project explored crowdsourcing landmarks through the OpenStreetMap geowiki [93]. Although similar to my approach (crowdsourcing landmarks through a geowiki), there are a few key differences: 1) I experiment with mobile crowdsourcing techniques to help elicit more relevant contributions, 2) Their project expected users to enter complicated details about landmark saliency, while I simplify the process for users by only requiring basic information (name, location, and sometimes type), but still allowing users to enter more rich data through tags and notes, 3) I implement route directions that utilize crowdsourced landmarks and allow other users to validate given landmarks.

4.4 Framework

My goal when building landmarks into our system was to have:

- A landmark-enhanced navigation system that feeds on user-contributed landmark information.
- A crowdsourcing approach that utilizes location information to better elicit contributions.
- An integrated elicitation strategy that exploits and connects the various contexts from which a user interacts with the application.

The following potential framework, shown in Figure 4.2, puts these goals together and represents the various types and contexts of landmark information contributions.

4.4.1 Landmark Contribution Types

Ideally, users would be able to contribute three types of work:

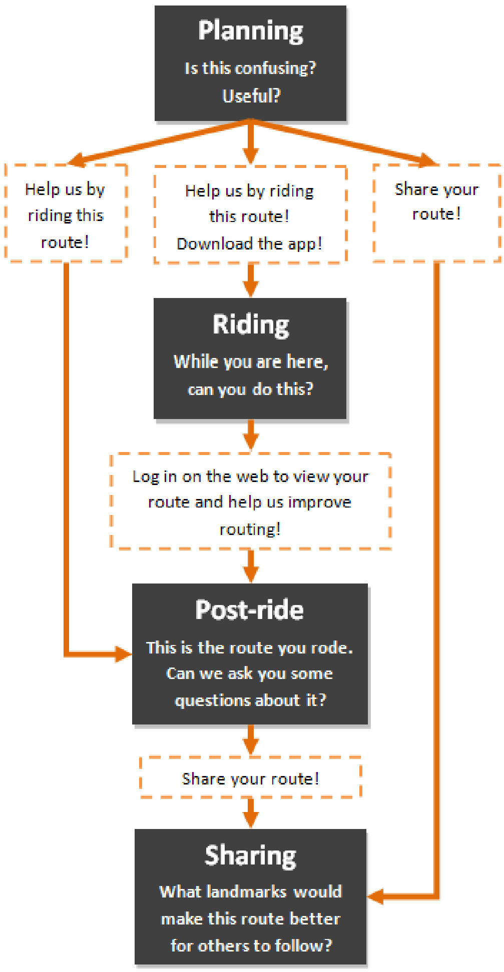


Figure 4.2: Landmark Contribution Framework

Landmark Need *"This intersection is confusing."* Users would be able to specify areas on the map or directions on a list that are confusing.

Landmark Input *"This landmark is here."* Users would be able to add the actual landmarks.

Landmark Validation *"That landmark is useful."* Users would be able to say if landmarks presented in route directions are useful.

4.4.2 Contribution Contexts

Users in our system would also be able to contribute these types of work within four different contexts, which take advantage of situations when users are likely to interact with landmark information. These contexts can also be used to encourage users to interact with landmarks through other contexts (i.e. to move from context to context), as represented with dashed rectangles in Figure 4.2.

Route Preparation The user has requested a route and is looking at the route directions and route display on the map. This is a good time to ask if certain parts of the route look confusing and if certain landmarks are useful. We can also remind the user that if they ride or share the route, they could help to improve it, thereby leading to the other contexts, as shown in Figure 4.2.

Riding The user is on the road. This is when users have the most access to landmark information. However, it is also the hardest time in which to interact with users. Interrupting them while they ride is a challenge. Once users finish riding, we can remind them to review their completed route in order to contribute landmark information.

Post-Ride The user visits our system after riding. The user is now familiar with the route and is more easily capable (i.e. does not need to stop cycling) of doing the work.

Route Sharing The user is sharing a route. At this point, users might have the incentive to improve the route with landmarks in order to make it more useful for others. As during the planning state, the user is also looking at the route directions and route display on the map, but now the goal is different, as it is based on the perceived needs of the community and not the user's own.

4.4.3 Research Aims

In this project, I did not aim to implement the complete framework, but instead aimed to first explore the feasibility of this approach for crowdsourcing landmarks. In order to do this, I did two experiments, one to explore a mobile crowdsourcing solution for the *Riding* and *Post-Ride* contexts and, based on those results, another to test a crowdsourcing approach during the *Route Preparation* and *Route Sharing* contexts.

The following are some of the design issues I explored when implementing and running these experiments:

Landmark Representation How should landmarks be represented in the system? First, POIs are often the simplest type of landmarks to understand (e.g. the gas station); however, there are other types of landmarks that might be useful but not as easy to represent with a POI, such as major highway crossings and T intersections. Second, other research, such as the OSM study in [93], use a complicated model of landmark saliency to describe the features that make a landmark useful (more salient). However, this information can be hard to recall and can make the contribution process cognitively harder for users. Therefore, my goal is to make entering landmarks as simple as possible.

Landmark Acquisition When are the best situations to gather landmarks? While users are riding is when they have the most access to accurate landmark information, but interrupting them is a difficult task. Can users still enter useful landmark data after a ride or when looking at route directions? Also, although users likely need to be familiar with a route in order to add new landmarks to our system, can a user that is unfamiliar with a route still select useful landmarks from a list of potential landmarks in our system?

Landmark Validation How should users tell us whether landmarks are useful or not? My goal is not necessarily to find out what makes good landmarks in general, as this might vary depending on the context (e.g. a landmark might be visible when riding north through an intersection, but may be hidden by a tree when riding south) and on the user (e.g. new cyclists might not recognize that they are crossing over highway I-35W or over the Mississippi river). Still, users should be able to tell us whether landmarks added by other users or selected by our system are useful to them in their current context.

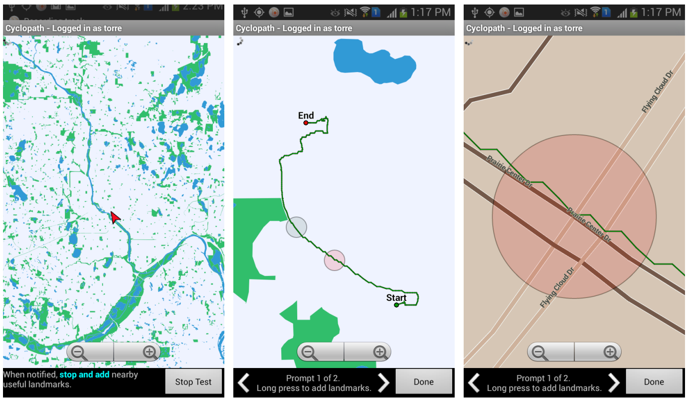


Figure 4.3: Landmarks Field Test

4.5 Mobile Crowdsourcing Field Test

My first attempt at crowdsourcing landmarks was based on a mobile solution using Cyclopath's Android application. My main probing questions were:

- Will users actually stop while riding to contribute landmark information? Or is a different approach, asking users to add landmarks after their ride, better? Stopping while cycling might be too inconvenient for most users.
- Are POIs good enough to capture most landmarks? POIs are simpler for users to add, but might not be versatile enough.

4.5.1 Mobile Field Test

Methods

I first implemented a landmark crowdsourcing interface for the mobile app as a probe to begin exploring our questions (see Figure 4.3). Users who agreed to participate were assigned to one of two categories when recording a GPS track:

Interruptive Solution Users in this condition were prompted while riding so that they would stop and add landmarks at nearby intersections.

Non-Interruptive Solution Users in this condition were prompted while riding so that they would look around (without having to stop) and pay attention to nearby landmarks. They would later receive an email reminding them to add the landmark information requested. Following a link in the email they would be able to see on the Cyclopath map their ride and the locations where prompts were received.

Users were prompted at most three times per ride. Those who completed this experiment were awarded a $10 Amazon gift card.

Results

A total of 14 users completed the test. Of those, 7 users experienced the interruptive solution, 5 users had the non-interruptive solution, and 2 users tried out both solutions. Users in the interruptive condition received a total of 43 prompts, while users in the non-interruptive condition received 14 (some technical errors prevented some trials in this condition from working, thereby resulting in lower numbers). Although not enough quantitative data to draw strong conclusions, I noticed the following: 1) Users in the interruptive condition added landmarks for 16% of the prompts, while users in the non-interruptive condition did so for 46% of the prompts; 2) Two users in the non-interruptive condition did additional work, one of them adding landmarks for intersections that had not been prompted and another attempting to edit block geometry; 3) Some landmarks were not actually POIs, but instead intersection information, and one user entered the same landmark twice for a building that spanned two intersections, suggesting that POIs might not be sufficient for capturing the landmark information that Cyclopath users are interested in.

4.5.2 Interviews

Given that the data was not enough, I followed the field test with user interviews in order to get more detailed feedback about their experiences.

Methods

I conducted 30-60 minute interviews with six of the users who had tried the landmark crowdsourcing interface (two users each from each condition and the two users who experienced both conditions). Those who participated in the interview were awarded a $20 Amazon gift card. The interview consisted of four parts:

Part 1 General ice-breaker questions about recent cycling experience, cycling behavior, and experience with Cyclopath applications.

Part 2 Questions related to their use of landmarks in routes they had recorded on Cyclopath.

Part 3 Questions related to their experience with the experiment UI, especially concerning the interruptive and non-interruptive variations.

Part 4 Explanation of research (including different experiment conditions) and additional feedback.

Results

The following are some of the interesting findings from the interviews:

- *Users felt that the interruptive solution was inconvenient and could even be dangerous.* Among other things, users said it was dangerous and difficult with traffic, it felt demanding, they felt pressured, it was a distraction, and it sometimes took them a while before they could come to a stop, at which point they couldn't see where they had been prompted.

- *The non-interruptive condition felt natural to users.* One user, not realizing that he was now in a different condition, thought the interface had improved. The only worry users had was recalling too many prompts. In our test, we never prompted users more than three times during their rides.

- *A combined approach, letting users decide when to add the information might be best.* One user expected to be able to add pictures of landmarks, which is only possible when stopping during a ride. Other users said prompts were easy to miss, which is less of a problem when users can view all prompted locations later.

- *Sound and voice would be good enhancements.* Some users mentioned that they use headsets when riding and hearing the prompts and instructions spelled out might help avoid missing them. Additionally, being able to enter landmarks by voice commands (in the interruptive condition) or leave voice annotations to review later (in the non-interruptive condition) were mentioned by users as useful potential additions.

- *In some cases users were unsure of what to do.* One reason was not knowing what types of landmarks we were asking them to add. Making it clear how landmarks are used in the route directions should help. Another case was when there were no obvious landmarks, such as when navigating through residential areas. A solution would be to let users explicitly say that there are no useful landmarks at the given intersection.

- *When asked to describe their routes verbally, users tended to add notes about safety.* Cyclopath already allows users to add notes to blocks. Making those notes more accessible from the route directions list would help make route presentation more useful and natural.

- *When describing their routes, users mentioned many types of landmarks that were hard to represent as POIs.* Examples included: highways, stop signs, traffic lights, road attributes (such as hills), large buildings, regions (such as university campus), and even road geometry (such as curves and roundabouts). The implication here is clear: POIs, although the simplest type of landmark to crowdsource, are not sufficient for making route presentation more natural.

4.6 Landmarks in Route Directions

Although mobile crowdsourcing can be a powerful technique given users' proximity to the places where work is needed, it can fall short if there is not enough usage. In Cyclopath, unfortunately, track recording is not often used by users. This experiment led me to believe that on Cyclopath there is much more potential to engage users and elicit landmark contributions in the *Route Preparation* and *Route Sharing* stages, given that the route finder is the most popular feature in Cyclopath.

The next step was then to create a user interface for displaying landmarks in route directions and allowing users to edit them. This approach would allow us to simultaneously elicit landmark information while improving route presentation (which is our end goal).

Based on the field test and user interviews, the following are some points to keep in mind when integrating landmarks into route directions:

- It should include types of landmarks other than simply POIs. Among some of these possibilities are intersection attributes (if available), byway geometry, network graph properties, and regions for larger objects.
- Tags and notes could be useful too. Tags from blocks can help show block attributes that aid navigation, tags from points can help better describe POIs, and notes can help make advice from other cyclists more accessible.
- Route instructions should integrate landmarks in such a way that users can hide or show specific landmarks. This would allow users to teach the system which landmarks are actually useful or not.

4.6.1 Methods

The aim of this experiment was to test an interface where users could provide landmark information directly from the list of route directions. The interface would allow users to select from a list of possible landmarks detected automatically by our system or add their own.

Research Questions

My main questions while designing this study were:

- **Landmark Types.** Which types of landmarks will users select or add more often? This could help improve automatic landmark selection. Furthermore, if POIs are not selected more often, this could point to shortcomings in current approaches that abstract all landmarks as POIs. New landmarks can also tell us which types of landmarks we are still not able to capture.

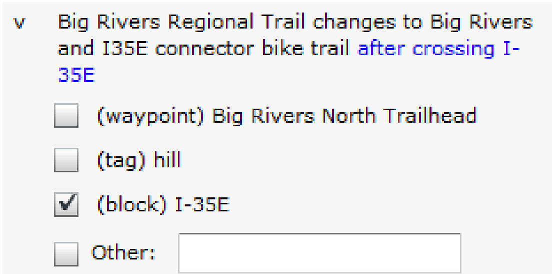


Figure 4.4: Landmark Selection

- **Familiarity.** What is the role of familiarity in the amount and type of landmarks selected? Familiarity with a route is most likely required to add new landmarks, but when selecting from a list of possible landmarks it may not be as necessary.

- **Agreement.** Do users generally agree on which landmarks are useful? In other words, do they tend toward an objective truth?

Experimental Design

In this experiment, users were asked to select landmarks for five routes. These five routes were chosen randomly from a list of 10 preselected routes. These routes were chosen to have a good variety of landmarks and about 10-20 steps with suggested landmarks.

For each route, users were first asked to rate their familiarity with the route by selecting one of the following options: *Unsure*, *A lot*, *Somewhat*, and *Not at all*.

Users were then able to browse the route on the map and also the list of route directions. The list included suggested landmarks at decision points that users could checkmark in order to include as part of the route direction, as shown in Figure 4.4.

The following are the types of landmarks that users could select:

- **POIs.** POIs are a natural choice to represent landmarks. They appear in route directions with information about its location relative to the turn (e.g. *"Turn left onto Hennepin Ave. before McDonalds on the right"*).

- **Blocks.** I included major roads and highways that the user would cross before arriving at the decision point (e.g. *"Turn right onto Hennepin Ave. after crossing I-35W"*). Smaller blocks were often not well-known or notable enough to be useful.

When turning onto a major road, the road name already appears in the route direction, so including it as a landmark would be redundant.

- **Tags.** Tags on blocks often represent block properties that can be useful landmarks, such as a hill, a bike lane or the fact that the road a user is turning onto is unpaved (e.g. *"Turn right onto Hennepin Ave (high traffic)"*).

- **T intersections.** T intersections are a feature of the road network infrastructure that carry structural attraction and can therefore serve as useful landmarks (e.g. *"Turn left onto Hennepin Ave at the T"*).

- **Other.** In order to capture landmarks that we missed or do not exist in our system, I included a freeform textbox for users to enter any other landmarks they thought could be useful.

There were some other types of landmarks that would have been useful to suggest, but were not possible due to missing information or implementation difficulty:

- **Parks, Lakes, Rivers.** Regions such as bodies of water and parks can be very useful, especially in the Twin Cities, where the Mississippi River crosses and countless lakes and parks abound. Unfortunately, most of these regions in our system are missing names. Furthermore, including these correctly in directions, which requires more advanced geometric analysis, is out of the scope of this experiment.

- **Intersection Properties.** Information about intersections, such as stoplights and stop signs is currently not available in our system, but could help turn intersections themselves into potential landmarks.

4.6.2 Results

Before describing my analysis of the results, I introduce some general statistics from the experiment and describe the various perspectives from which I looked at the results. I analyze the results of the 22 users who edited landmark information for at least one route in our experiment. Table 4.1 summarizes the general statistics.

- **Route Instance Perspective.** This perspective helps analyze the data by looking at each instance of a route being shown to a user as a work instance. Each

Users	22
Routes	10
Route instances	69
Suggestion cases	239
Suggestions	1652
Landmarks selected	479
New landmarks added	162

Table 4.1: General statistics for the landmark selection experiment.

of the 10 preselected routes was shown to at least 5 different users. In total there were 69 instances of routes shown to users, with one route having a total of 11 users.

- **Suggestion Case Perspective.** The total number of landmarks that could be suggested to users was 239 (an average of almost 24 landmarks per route). In this section I will refer to these landmarks as *cases*, where each suggestion case could be presented to multiple users.

- **Individual Suggestion Perspective.** At this level, I focus on each individual landmark suggestion. In total, the system made 1652 landmark suggestions to users. Of these, users selected a total of 479 landmarks. Users also added 162 landmarks that were not included in the suggestions.

Landmark Type

The first data I looked at was which types of landmarks users tended to choose more often from the suggestions and which types users added as new landmarks. Table 4.2 summarizes the number of landmarks suggested and selected.

As can be seen in the results, T intersections were added significantly more often than the other types of landmarks. This result starts to hint at the fact that POIs might not be a sufficient abstraction for landmarks, likely missing out on a substantial amount of useful information.

In order to further investigate this hypothesis, I also analyzed the text for all 162 new landmarks added by users as part of the "Other" category. The results of the textual categorization are shown in Table 4.3. The categorizations are not mutually

Type	Suggestion Cases	Suggestions	Landmarks Selected	% Selected
Points	72	537	142	26.4%
Blocks	33	216	42	19.4%
Tags	106	714	202	28.3%
T Intersections	28	185	93	50.3%
Other	0	0	162	

Table 4.2: Landmarks suggested and selected by type. In total, 20% of all suggestions were accepted by users. The difference in percentage of landmarks selected is significant ($\chi^2 = 38.6,\ P < 0.0001$).

Type	Example	Total	Percentage
Block property	*Steep hill*	30	18.5%
Notes	*difficult to find sign prior to intersection*	30	18.5%
Block	*Cross Minnehaha Ave*	29	17.9%
Block type	*Bike path*	23	14.2%
Infrastructure	*before roundabout, first left*	19	11.7%
POI	*Orchestra Hall*	15	9.3%
Intersection property	*Stop sign*	14	8.6%
Corrections	*ignore this: go straight, not left.*	14	8.6%
River	*Cross the Mississippi River*	6	3.7%
Spam	*sdfsdf*	6	3.7%
Region	*at end of park*	5	3.1%

Table 4.3: Categorization of new landmarks added by users. Some notes referred to multiple types of landmarks, so the categories are not mutually exclusive.

exclusive, as some of these notes contained references to multiple landmarks.

Although some of the new landmarks, such as corrections, are not landmarks, we can still see that users barely limited themselves to using only POIs as landmarks. In fact, POIs were used less than 10% of the time. These results led me to the following two conclusions:

1. POIs are not a sufficient representation for route direction landmarks. Their use is limited to mostly objects on the route and cannot represent many other useful properties of routes, including infrastructure and block properties which are commonly used by people when describing route directions.

2. There is still plenty of potential landmark information that is missing or we are

Familiarity	Routes	Suggestions	Selected	Percentage	New landmarks added
Not at all	18	350	164	46.9%	20 (1.1 per route)
Somewhat	29	740	168	22.7%	71 (2.4 per route)
A lot	19	487	136	27.9%	55 (2.9 per route)

Table 4.4: Landmarks suggested, selected, and added by familiarity. The difference in percentage of landmarks selected is significant ($\chi^2 = 47.35,\ P < 0.0001$).

not yet able to capture in our system. For example, block properties, such as hills, could lead to new tags for those blocks. Intersection properties, mentioned 8.6% of the time, should also be included in our system in the future. Our landmark selection algorithm could also be smarter about infrastructure, looking at properties such as curves and number of street crossings.

Familiarity

In order to explore the role of familiarity in crowdsourcing landmark information, I looked at the amount of landmarks selected or added by users depending on their familiarity with each route. This can help us identify differences in contributions between experts (those familiar with a route) and non-experts.

The results by familiarity are shown in Table 4.4. Because users specified familiarity as *Unsure* only three times and given that these cases would not tell us much about familiarity, I omit those results, looking only at data where the familiarity value was *Not at all*, *Somewhat*, and *A lot*.

From these results, there are two trends we can see. The first is that unfamiliar users generally selected more landmarks from those suggested (almost 47% compared to about 23-28% for "experts"). This result could be due to different factors. Perhaps users who are not familiar with a route will feel more confident following a new route with more landmarks available. Or maybe these "non-experts" are not good judges of what makes good landmarks and therefore select more landmarks than needed.

On the other hand, the second trend that we see is that users familiar with routes added more new landmarks than those unfamiliar with the routes. "Experts" added more than twice the amount of landmarks than "non-experts". This result emphasizes the usefulness of familiarity for adding new information to the system.

Threshold	Agreed Yes	%	Agreed No	%
100%	1	0.4%	32	13.4%
75%	11	4.6%	124	51.9%
66%	16	6.7%	146	61.1%
50%	42	17.6%	197	82.4%

Table 4.5: How often a certain percentage of users agreed on whether a landmark should be selected or not.

In essence, the data suggests that selecting landmarks from suggestions and adding new landmarks might require different levels of familiarity and might also support different goals. Landmark suggestions can aid users exploring new routes, while landmark input can let "experts" improve the map and, consequently, future suggestions.

Agreement

Although we see different landmark use strategies between "experts" and "non-experts", this data doesn't tell us whether users in general or even experts agree with each other. If landmark selection tends toward agreement about which landmarks should be used, this could be an indication of an objective truth, at least for most landmarks. For this reason, I took the analysis one step further and compared agreement between users.

One simple method for looking at agreement is to define thresholds for percentage of agreement on each landmark suggestion. Table 4.5 shows for how many cases users agreed 100% of the time, at least 3/4 of the time, at least 2/3 of the time, and at least half of the time.

Complete agreement was rare, 13.8% of all cases and only once for actually adding a landmark. However, about 56% of all cases had at least 3/4 agreement and about 68% had 2/3 agreement. If we go by simple majority vote, 17.6% of all suggested landmarks would be added.

These results suggest that aggregating landmark selection data might be a plausible strategy. However, we are more interested in actually selecting landmarks than in not selecting them. And there are few landmarks where we have strong confidence about selecting them automatically. Deciding on a threshold can also be somewhat arbitrary. Furthermore, these results don't tell us whether different groups agree more between each other (such as experts agreeing with other experts) and how much of their

Familiarity	Yes	No	Overall
All	33.5%	73.5%	62.1%
Not at all	43.3%	63.0%	55.3%
Somewhat	32.0%	82.6%	72.3%
A lot	26.1%	74.7%	62.3%

Table 4.6: Raw agreement proportions based on familiarity of users with routes.

agreement is by chance. This strategy also assumes that all landmarks hold the same utility for all users, which as we saw in the previous familiarity results, might not be correct.

One method that can give us a more complete picture about agreement between users is to calculate raw agreement, i.e. how often users agree in general. I calculated agreement for multiple raters using the techniques described in [94] and [95]. In essence, agreement between raters is measured by the proportion of agreeing pairs out of all the n *(n - 1)* possible pairs. Table 4.6 shows the results for agreement specific to each category, where "yes" refers to positive agreement, or agreement on whether to actually add a landmark and "no" refers ot negative agreement, or agreement on whether to not add a landmark. I also divided the results by familiarity, in order to investigate whether experts would tend to agree more often than non-experts.

These results present a mixed story. Overall agreement seems moderate at 62.1%. However, given that selecting landmarks is likely to occur less often than not selecting them, we look to agreement specific to categories in order for a better picture. As expected, agreement on when to actually select landmarks is lower, at 33.5%. Users tend to agree more often on when not to select landmarks, but simply because they are selecting less landmarks in general. This can also help explain why it seems that unfamiliar users (or "non-experts") tend to agree more often on when to select landmarks. This is in part driven by the fact that they are selecting more landmarks overall, as shown in the previous section. For overall agreement, although unfamiliar users agree less often than other users, the trend is not clear, particularly because users who were somewhat familiar with routes had higher overall agreements than both "experts" and "non-experts".

In order to answer whether users are tending toward an objective truth, raw agreement measures are not enough. The reason is because part of this agreement might be

Route	Agreement	kappa
1	70.5%	0.04
2	50.9%	0.01
3	58.5%	0.03
4	67.7%	-0.03
5	61.1%	0.04
6	56.8%	-0.11
7	56.2%	0.05
8	66.8%	0.09
9	54.4%	0.06
10	75.6%	0.27

Table 4.7: Fleiss' kappa statistic for all 10 routes. A kappa of 1 indicates complete agreement, while a kappa of -1 indicates complete disagreement.

due to random chance, especially because there is a tendency toward the "No" category. To overcome this, Cohen [96] proposed the kappa coefficient, a widely-used statistical measure that takes into account chance agreement. Although there is disagreement on whether kappa statistics really are "chance-corrected" or not [97], they remain useful tools to verify that agreement exceeds chance levels.

Cohen's kappa can only be used to measure agreement between two raters. However, in our experiment we had multiple users "rate" each landmark. Therefore, we use Fleiss' kappa [94], which works for multiple raters. Because the experiment didn't have the same users rating the same routes, it makes more sense to calculate the kappa for each route individually, in which case every user "rated" the same landmarks. The results for calculating Fleiss' kappa for each of the 10 routes are given in Table 4.7.

A kappa of 1 would mean perfect agreement, while a kappa of -1 would equate to negative agreement and a kappa of 0 to agreement by chance. Except for the last route, all values are extremely close to 0. A commonly used, albeit not necessarily agreed upon, scale of kappa values would classify these results as barely *slight agreement* [98]. In other words, the agreement results are very likely the result of chance.

These results suggest that there is no objective truth when it comes to which landmarks are more useful than others. Familiarity with a route might translate to different landmark needs, but even familiarity alone did not greatly impact agreement, as shown above. There might be some landmarks that users generally agree upon more often, but for the majority of them, it may depend on not only a user's familiarity with the route, but also on their own personal preferences. One important implication of these results is that we cannot rely on "experts" to select high-quality landmarks, given that their goals and needs are different from other users and even from each other.

4.7 Implications

We can draw out several implications from these results:

- POIs, often used in systems as the *de facto* landmarks, are not sufficient. The ecosystem of useful landmarks is quite diverse and as such requires different landmarks to be represented differently in mapping systems and in route directions.

- "Experts" are essential for adding new landmark information. It is not surprising that users who are familiar with a route are able to add more information about landmarks along that route. Using edit logs and GPS traces, our system could estimate familiarity and automatically detect these experts.

- We cannot rely on "experts" to decide which landmarks to actually include in route directions. "Non-experts" still need the ability to select their own landmarks, given that there is no objective truth when it comes to useful landmarks. We could probably suggest some landmarks based on agreement statistics, but we can't depend on these aggregation techniques completely.

4.8 Summary

In this chapter, I described my research on using landmarks to improve route directions by making them more natural and informative.

The research in this chapter was inspired by a study of the use of Cyclopath in a Chinese setting. Users in that study used landmarks extensively to browse and navigating the map, motivating me to find ways to better integrate landmarks into route directions.

My first attempt at crowdsourcing landmarks using mobile devices showed promise, but lacked the necessary amount of usage. Taking into account that route finding is the most heavily used feature in Cyclopath, I decided that the best context in which to ask for landmark information would then be when looking at route directions.

The experiment consisted of asking users to select landmarks from a list of suggestions at different steps in a route or to add their own. The results showed that: 1) POIs are insufficient to represent the diversity of landmarks users employ to describe their routes, 2) users familiar with routes were able to add more new landmarks, while users unfamiliar with routes would tend to add suggested landmarks more often, and 3) users in general did not agree much on which landmarks should be added, suggesting that there is no clear objective truth regarding usefulness for most landmarks.

Recognizing the insufficiency of POIs, the role of familiarity, and the absence of an objective truth can help us design landmark crowdsourcing and selection strategies that are more flexible and personalized than currently existing methods.

Chapter 5

Conclusion

5.1 Summary of Contributions

In this thesis, I outlined and explored a series of techniques for improving the route finding process. Instead of improving the route finding algorithm itself, these techniques make use of user input to improve route preferences, map data, and route presentation. In the past three chapters I detailed the following contributions:

- **Improving Route Personalization.** I studied the use of community-shared tags that allow users to easily express preference for a large number of roads in the system with little effort. Correlation between block-specific ratings and ratings deduced from tag preferences was evidence of the utility of this technique for personalizing the route finding process.

- **Improving Map Data.** I presented an HMM-based map matching algorithm that could detect missing blocks from users' GPS traces, thereby improving the map using users' implicit input (their riding behavior). Tests with the data in our system confirmed that there were a lot of missing roads that could be detected using this algorithm.

- **Improving Route Presentation.** I tested a method for crowdsourcing landmark information by integrating landmark suggestions into route directions. This method allows users to contribute information while also enriching route directions. I also showed that POIs are not sufficient to represent landmarks and that there is no objective truth regarding which landmarks are more useful.

5.2 Implications

In this section I discuss the implications of the research I have presented in this thesis. I aim to show why the reader should care about the results of each contribution. I will also outline several design implications for mapping systems and, taking it a step further, online communities and crowdsourcing systems in general.

5.2.1 Improving Route Personalization

With community-shared tags, we were able to generalize preferences to make them more efficient for users to specify. Instead of having to specify a preference for every single road segment, users can specify general preferences for all segments that share a common trait. This is especially useful in domains where user preferences and abilities play big roles, such as cycling and hiking.

Implications for mapping systems

Make user preferences part of the routing algorithm. Including users in your considerations for your algorithm seems like the straighforward implication of this research. However, there are many ways to go about doing so. For example, looking at previous riding history in order to deduce route preferences and allowing users to manipulate the route once it has been computed are two possible strategies.

The limitations of these examples are that they don't allow users to explain their preferences and, consequently, makes generalizing them more difficult. For this, tags are an ideal solution, as they are simple, yet expressive enough.

One caveat when using tag preferences is that you need to encourage tagging among the community. Otherwise, tags will fail to fulfill their role if they are not applied enough. In Cyclopath, a tag that is applied to only one road segment is just as good as simply rating that block. Furthermore, you also need to encourage a common vocabulary in order to lower the cognitive burden on users when having to rate many similarly named tags.

One good technique for enouraging tagging is to use tags computationally as in our system. When tags are used for more than just information purposes, they become more useful and thus users will draw more benefit from applying more tags. In Cyclopath, for example, tags are used for route finding, searching, and filtering.

Implications for other systems

Look for opportunities where community shared information can be used to improve user-specific recommendations. In Cyclopath, tags, which were shared, were used to improve route recommendation. This technique is not limited to recommending routes. For example, Sen et al. used this approach to recommend movies based on how users rated tags applied to other movies [46]. In this way their recommendation algorithm could deduce that certain users liked moves *because* they were animated or *because* they included a specific actor. Similarly, tag preferences can be used to improve task recommendations on sites like Wikipedia and product recommendation on sites like Amazon.

5.2.2 Improving Map Data

My algorithm for matching GPS traces to map data is a bridge between map matching and map building. It exploits users' implicit contributions (GPS traces) to automatically find possible missing data. This in turn gives it the potential to suggest work to users, falling at the intersection between automatization and user contributions.

Implications for mapping systems

Ease the process of users editing your map. Editing a map is not simple. Users in Cyclopath tend to have a more difficult time editing road geometry than other types of edits. Yet that is one of the most crucial types of work needed in Cyclopath. Even if users know the methods required to edit the map, they might not know where their skills are needed. This is an even bigger issue for new users, who can get discouraged if they believe their knowledge or skills are not needed.

In a process such as extracting data from GPS traces, you can ease user contributions in two ways. First, users can volunteer to provide GPS data, which requires very little effort on their part if they are already traveling through the needed routes. Second, you can suggest roads to add based on previously stored GPS data. In most cases, these suggestions can be made even more simple by making them a one-click action: "Do you want to add this block here?".

Don't assume that every GPS point maps to an existing block in your map data. This warning stems from the fact that we were able to find so many instances of missing data in our system. If your goal for matching GPS traces to map data is analysis, beware of trying to match every single observation to a road segment, as the real road segment could be missing. This could in turn lead to incorrect conclusions.

There are a few strategies that can be followed to deal with or minimize the impact of missing data. First, you can ignore any matches of low confidence. By doing this, you avoid dealing with GPS points for which you are not sure what they map to. However, you may miss important data or patterns, especially if these points are many. A second option is to analyze data at a higher level, instead of at the road segment level. This lets you pool GPS points into slightly larger regions. However, the loss in precision might also miss important findings, especially if you are interested in information at the street level or even lane level.

Finally, my recommendation is to follow an approach similar to my research, where you create new blocks to match the data to. This approach still allows you to decide on a case-by-case basis whether to actually include the new block in the analysis or not.

Implications for other systems

Learn from user behavior and use that to elicit more contributions. Doing so not only helps match users to tasks, but also helps make contributions easier for new users.

For example, you can learn from user behavior to understand where their interests or knowledge lie. Do users spend most of their time reading math articles on Wikipedia? They might have interest in editing other math articles. Do users write a lot of reviews on Amazon? They might be interested in rating other reviews too.

In geographic systems, familiarity with an area (which can be deduced from GPS traces) can be an indication of knowledge and capability of contributing. In recommendation systems, if a user constantly disagrees or modifies the system's recommendations, what information could the system elicit from the user that could help that and other recommendations? For example, in route finding systems, if a user drags the given route to modify it, the system could ask the user to explain why the new route is better. Similarly, if users in Amazon don't buy the recommended or most popular option, it could help to ask the user to compare the two items and explain why they opted for the apparently subpar option.

5.2.3 Improving Route Presentation

Including landmarks in route directions has for a long time been known to make directions more natural, similar to how humans describe routes. My approach crowdsources landmarks at the moment where they are most relevant, while looking at directions within the context of a route. It takes the simple approach of not burdening the users with having to explain why the landmarks are good, but letting them simply select the landmarks useful to them.

Implications for mapping systems

Crowdsourcing context is important. For mapping applications the context where users are the most exposed to the information needed on the map is when they are traveling through those areas of need. However, this is also when getting their attention might be hardest and in some cases even dangerous. Futhermore, users might spend relatively little time within that context when using mapping applications.

Understanding what the best moment to elicit contributions is requires knowledge of how and for what reason users use the system. Eliciting contributions when they are interacting with that information, such as landmarks and route directions, can lead to increased interactions and contributions. Keep an eye, however, on whether quality decreases too much when contributions are increased. Depending on your end goal, eliciting contributions while users have the ability to provide higher quality information, albeit less often, might be a necessary trade-off.

Don't use only POIs as landmarks. Whether you want to improve route directions, map browsing or any other aspect of the map that involves landmarks, keep in mind that the definition of a landmark is much more than just an object on the map. The challenges remain, however, on how to obtain these other types of landmarks. Infrastructure information, for example, needs to be computed by looking at the structure and geometry of the road network. Block properties, on the other hand, require editors to be able to add properties, such as tags, to blocks.

There are also challenges on how to include diverse landmarks in route directions. POIs are commonly used because they are easy to locate from an intersection. However, other landmarks such as parks can be harder to include: the directions might take the user through the park, alongside the park, right up to the park, etc. When designing route directions, you must also decide whether to include confirmation landmarks (those that help users either confirm that they are going the right way when there is a long distance between decision points and those that help users realize quickly that they went past a decision point). Examples of these could be "You will pass Minehaha Park on your right" and "If you see a gas station on your left, you went too far".

Implications for other systems

Be aware of the limitations of relying on experts. Familiarity and knowledge about a topic are essential components for effective task distribution and elicitation in online communities. However, we must be careful when identifying "experts". These users can be extremely useful in communities that depend on crowdsourced content. However, we must also understand up to what point we should rely on them. Here are some cases where the limitations of experts can be apparent:

1. When users of different abilities may have different preferences. This is analogous to designing user interfaces for experts and for beginners. Experts benefit more from advanced features, while beginners benefit from a simple interface. This is also the case in the cycling domain, where beginner cyclists have different needs and worries than expert cyclists.

2. When experts disagree. If experts cannot agree between themselves, this can be a sign toward the absence of an objective truth.

3. In systems completely driven by personal preference. Most recommender systems fall into this category. For example, there are no experts when it comes to deciding what the best movie of all time is.

5.3 Future Work

Given the scope and time constraints of this thesis, there are several avenues of further research that I was not able to pursue. In this section I detail some of these possible future paths.

5.3.1 Improving Route Personalization

Conditional tags. There are properties of blocks that are not permanent (e.g. constructions) or that are only valid under certain conditions (e.g. high traffic, snow-related obstacles). One potential solution would be to create tags for which conditions can be set. For example, tags such as *high traffic* could be set to be activated only for certain hours of the day. Tags such as *construction* could be set to expire after a certain time span. These techniques could potentially help make users' preferences more accurate.

Tag suggestions. In order for tags to be useful as a proxy for specifying preferences, they need to be applied to as many blocks as possible. Suggesting tags for users to apply is a possible strategy to get users to contribute more tags. Tags can be suggested by looking at longer textual notes attached to blocks or by looking at other blocks with similar properties or proximity. Tags regarding traffic could also be deduced from GPS traces or route finding histories.

Explaining recommendations. Tags can be used to explain recommendations to users [49]. It would be interesting to explore what this would look like in the context of route recommendation. Users could potentially benefit from knowing that a route was selected *because* of their preferences for certain tags. They could also benefit from the reverse, knowing why alternate routes were not selected. This technique could help users make better decisions when personalizing their routes and even improve route presentation by showing the roles played by route traits that are important to users.

5.3.2 Improving Map Data

Real-world algorithm usage. Releasing Yanjie Liu's interface [68] for suggesting missing blocks to users as a feature in our live website would be the logical next step. It would be interesting to see how users use it and if it in anyway encourages more recordings of GPS routes.

Real-time mobile map matching. Real-time map matching is useful for accurate real-time navigation. One advantage of the algorithm I presented is that for the most part it analyzes and matches GPS observations as they come in. The one exception is when creating new blocks, where the algorithm has to backtrack. What changes to the algorithm would be needed to allow the mobile application to show users real-time matching while they record their GPS tracks? How would real-time navigation have to be modified to accurately describe directions when users ride through blocks that our missing in our system?

Better block creation. Although users would be able to edit the geometry of our block suggestions, it would make the process easier if our algorithm could do a better job when creating new blocks. This includes better handling of intersections, offsets, GPS noise, and cases where the user backtracks over a new block (in which case the new path is duplicated).

GPS recording campaigns. As this is a feature that depends on users recording their bike rides, encouraging users to actually go out and record as much as possible is important. Experimenting with different types of campaigns would be an interesting avenue for research. One possibility is to use *exergaming* techniques, were users are encouraged to exercise while also including gaming elements, such as points and competitions.

5.3.3 Improving Route Presentation

Landmark validation. In my experiment, I found that raters didn't seem to agree that much with each other. But if we explicitly asked other users to rate selected landmarks as useful or not useful, would these ratings follow the same pattern (little correlation)? Are landmarks selected by "non-experts" still considered useful by others? What about new landmarks added (mainly be familiar users)? A feedback mechanism is the next step in our system for crowdsourcing landmark information.

Interruptive mobile crowdsourcing. More research is needed to explore effective interruptive mobile crowdsourcing techniques. I saw in my experiments that: 1) getting users to participate is hard, 2) getting their attention while traveling is also hard, and 3) asking them to stop and contribute information is inconvenient and in some cases could even be dangerous. Multi-modal techniques that use audio and speech recognition might be one way to deal with some of these problems.

Complete landmark framework. In the last chapter, I presented a more complete framework for crowdsourcing landmarks. Implementing the whole framework could reveal new interactions and use patterns. Would users stick to only contributing landmark information in some contexts? Or would they contribute differently in the different contexts? What behavior differences would emerge between experts and non-experts? Are there any parts of the framework that fail to become useful in the landmark crowdsourcing process?

More landmark types. How can we make landmarks even more diverse while still maintaining their simplicity? What are the challenges to integrating regions (parks, rivers, lakes, malls) as landmarks? How can we best integrate confirmation landmarks that occur far from route decision points?

5.4 Concluding Remarks

A fast, efficient algorithm does not by itself make a great route finding experience. Route finding is also about flexibility for optimization and personalization, accurate and diverse map data, and rich and natural route presentation. All of these factors together influence the user's final perception of quality. My hope is that in this thesis I have been able to show that route finding is much more than just a routing algorithm.

References

[1] Tom Chivers. The story of google maps. *The Telegraph*, 6 2013. `http://www.telegraph.co.uk/technology/google/10090014/The-story-of-Google-Maps.html`.

[2] Jonathan S. Geller. Google vp marissa mayer dishes google mobile stats, 150m mobile maps users. *Boy Genius Report*, 3 2011. `http://bgr.com/2011/03/13/google-vp-marissa-mayer-dishes-google-mobile-stats-150m-mobile-users/`.

[3] Marie-Paule Daniel and Michel Denis. Spatial descriptions as navigational aids: A cognitive analysis of route directions. *Kognitionswissenschaft*, 7(1):45–52, 1998.

[4] Cyclopath. `http://cyclopath.org/`.

[5] Reid Priedhorsky and Loren Terveen. The computational geowiki: what, why, and how. In *Proceedings of the 2008 ACM conference on Computer supported cooperative work*, pages 267–276. ACM, 2008.

[6] Katherine Panciera, Reid Priedhorsky, Thomas Erickson, and Loren Terveen. Lurking? cyclopaths?: a quantitative lifecycle analysis of user behavior in a geowiki. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 1917–1926. ACM, 2010.

[7] Mikhil Masli, Reid Priedhorsky, and Loren G Terveen. Task specialization in social production communities: The case of geographic volunteer work. In *ICWSM*, 2011.

[8] Reid Priedhorsky, Mikhil Masli, and Loren Terveen. Eliciting and focusing geographic volunteer work. In *Proceedings of the 2010 ACM conference on Computer supported cooperative work*, pages 61–70. ACM, 2010.

[9] Mikhil Masli and Loren Terveen. Evaluating compliance-without-pressure techniques for increasing participation in online communities. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 2915–2924. ACM, 2012.

[10] Mikhil Masli, Landon Bouma, Andrew Owen, and Loren Terveen. Geowiki+ route analysis= improved transportation planning. In *Proceedings of the 2013 conference on Computer supported cooperative work companion*, pages 213–218. ACM, 2013.

[11] Riko Jacob, Madhav Marathe, and Kai Nagel. A computational study of routing algorithms for realistic transportation networks. *Journal of Experimental Algorithmics (JEA)*, 4:6, 1999.

[12] Daniel Delling, Peter Sanders, Dominik Schultes, and Dorothea Wagner. Engineering route planning algorithms. In *Algorithmics of large and complex networks*, pages 117–139. Springer, 2009.

[13] Matthias Müller-Hannemann and Mathias Schnee. Finding all attractive train connections by multi-criteria pareto search. In *Algorithmic Methods for Railway Optimization*, pages 246–263. Springer, 2007.

[14] Yann Disser, Matthias Müller-Hannemann, and Mathias Schnee. Multi-criteria shortest paths in time-dependent train networks. In *Experimental Algorithms*, pages 347–361. Springer, 2008.

[15] Angelica Lozano and Giovanni Storchi. Shortest viable path algorithm in multimodal networks. *Transportation Research Part A: Policy and Practice*, 35(3):225–241, 2001.

[16] Kenneth L Cooke and Eric Halsey. The shortest route through a network with time-dependent internodal transit times. *Journal of mathematical analysis and applications*, 14(3):493–498, 1966.

[17] Randolph W Hall. The fastest path through a network with random time-dependent travel times. *Transportation science*, 20(3):182–188, 1986.

[18] CO Tong and AJ Richardson. A computer model for finding the time-dependent minimum path in a transit system with fixed schedules. *Journal of Advanced Transportation*, 18(2):145–161, 1984.

[19] Stefan Schroedl, Kiri Wagstaff, Seth Rogers, Pat Langley, and Christopher Wilson. Mining gps traces for map refinement. *Data mining and knowledge Discovery*, 9(1):59–87, 2004.

[20] Lili Cao and John Krumm. From gps traces to a routable road map. In *Proceedings of the 17th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*, pages 3–12. ACM, 2009.

[21] Rene Bruntrup, Stefan Edelkamp, Shahid Jabbar, and Bjorn Scholz. Incremental map generation with gps traces. In *Intelligent Transportation Systems, 2005. Proceedings. 2005 IEEE*, pages 574–579. IEEE, 2005.

[22] Open Street Map. `http://www.openstreetmap.org`.

[23] Google. Google map maker. `www.google.com/mapmaker`.

[24] Bing Liu. Intelligent route finding: combining knowledge, cases and an efficient search algorithm'. In *ECAI*, volume 96, pages 380–384. Citeseer, 1996.

[25] Benjamin B Bederson, Aaron Clamage, and Catherine Plaisant. Enhancing in-car navigation systems with personal experience. *Transportation Research Record: Journal of the Transportation Research Board*, 2064(1):33–42, 2008.

[26] Harald Holone, Gunnar Misund, Håkon Tolsby, and Steinar Kristoffersen. Aspects of personal navigation with collaborative user feedback. In *Proceedings of the 5th Nordic conference on Human-computer interaction: building bridges*, pages 182–191. ACM, 2008.

[27] Reid Priedhorsky, David Pitchford, Shilad Sen, and Loren Terveen. Recommending routes in the context of bicycling: algorithms, evaluation, and the value of personalization. In *Proceedings of the ACM 2012 conference on Computer Supported Cooperative Work*, pages 979–988. ACM, 2012.

[28] Maneesh Agrawala and Chris Stolte. Rendering effective route maps: improving usability through generalization. In *Proceedings of the 28th annual conference on Computer graphics and interactive techniques*, pages 241–249. ACM, 2001.

[29] Kristin L Lovelace, Mary Hegarty, and Daniel R Montello. Elements of good route directions in familiar and unfamiliar environments. In *Spatial information theory. Cognitive and computational foundations of geographic information science*, pages 65–82. Springer, 1999.

[30] Steffen Werner, Bernd Krieg-Brückner, Hanspeter A Mallot, Karin Schweizer, and Christian Freksa. Spatial cognition: The role of landmark, route, and survey knowledge in human and robot navigation1. In *Informatik97 Informatik als Innovationsmotor*, pages 41–50. Springer, 1997.

[31] Martin Raubal and Stephan Winter. *Enriching wayfinding instructions with local landmarks*. Springer, 2002.

[32] Wikipedia. `http://www.wikipedia.org/`.

[33] SeeClickFix. `http://seeclickfix.com/`.

[34] Georgios Chatzimilioudis, Andreas Konstantinidis, Christos Laoudias, and Demetrios Zeinalipour-Yazti. Crowdsourcing with smartphones. 2012.

[35] Waze. `http://www.waze.com/`.

[36] Ed H Chi and Todd Mytkowicz. Understanding the efficiency of social tagging systems using information theory. In *Proceedings of the nineteenth ACM conference on Hypertext and hypermedia*, pages 81–88. ACM, 2008.

[37] S. Golder and B.A. Huberman. The structure of collaborative tagging systems. In *Proc. Information Science*, 2006.

[38] Paul Heymann et al. Can social bookmarking improve web search? In *Proc. WSDM*, pages 195–206, 2008.

[39] A Walkyria Rivadeneira, Daniel M Gruen, Michael J Muller, and David R Millen. Getting our head in the clouds: toward evaluation studies of tagclouds. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 995–998. ACM, 2007.

[40] Shilad Sen, Shyong K Lam, Al Mamunur Rashid, Dan Cosley, Dan Frankowski, Jeremy Osterhouse, F Maxwell Harper, and John Riedl. Tagging, communities, vocabulary, evolution. In *Proceedings of the 2006 20th anniversary conference on Computer supported cooperative work*, pages 181–190. ACM, 2006.

[41] Jennifer Thom-Santelli, Michael J Muller, and David R Millen. Social tagging roles: publishers, evangelists, leaders. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 1041–1044. ACM, 2008.

[42] Stephen F King and Paul Brown. Fix my street or else: using the internet to voice local public service concerns. In *Proceedings of the 1st international conference on Theory and practice of electronic governance*, pages 72–80. ACM, 2007.

[43] Nikhil Garg and Ingmar Weber. Personalized, interactive tag recommendation for flickr. In *Proceedings of the 2008 ACM conference on Recommender systems*, pages 67–74. ACM, 2008.

[44] David R Millen, Jonathan Feinberg, and Bernard Kerr. Dogear: Social bookmarking in the enterprise. In *Proceedings of the SIGCHI conference on Human Factors in computing systems*, pages 111–120. ACM, 2006.

[45] Shilad Sen, F Maxwell Harper, Adam LaPitz, and John Riedl. The quest for quality tags. In *Proceedings of the 2007 international ACM conference on Supporting group work*, pages 361–370. ACM, 2007.

[46] Shilad Sen, Jesse Vig, and John Riedl. Tagommenders: connecting users to items through tags. In *Proceedings of the 18th international conference on World wide web*, pages 671–680. ACM, 2009.

[47] Karen HL Tso-Sutter, Leandro Balby Marinho, and Lars Schmidt-Thieme. Tag-aware recommender systems by fusion of collaborative filtering algorithms. In

Proceedings of the 2008 ACM symposium on Applied computing, pages 1995–1999. ACM, 2008.

[48] Yi Zhen, Wu-Jun Li, and Dit-Yan Yeung. Tagicofi: tag informed collaborative filtering. In *Proceedings of the third ACM conference on Recommender systems*, pages 69–76. ACM, 2009.

[49] Jesse Vig, Shilad Sen, and John Riedl. Tagsplanations: explaining recommendations using tags. In *Proceedings of the 14th international conference on Intelligent user interfaces*, pages 47–56. ACM, 2009.

[50] Tammara Combs Turner, Marc A Smith, Danyel Fisher, and Howard T Welser. Picturing usenet: Mapping computer-mediated collective action. *Journal of Computer-Mediated Communication*, 10(4):00–00, 2005.

[51] Howard T Welser, Eric Gleave, Danyel Fisher, and Marc Smith. Visualizing the signatures of social roles in online discussion groups. *Journal of social structure*, 8(2):1–32, 2007.

[52] Howard T Welser, Dan Cosley, Gueorgi Kossinets, Austin Lin, Fedor Dokshin, Geri Gay, and Marc Smith. Finding social roles in wikipedia. In *Proceedings of the 2011 iConference*, pages 122–129. ACM, 2011.

[53] Susan L Bryant, Andrea Forte, and Amy Bruckman. Becoming wikipedian: transformation of participation in a collaborative online encyclopedia. In *Proceedings of the 2005 international ACM SIGGROUP conference on Supporting group work*, pages 1–10. ACM, 2005.

[54] Arvind Thiagarajan, Lenin Ravindranath, Katrina LaCurts, Samuel Madden, Hari Balakrishnan, Sivan Toledo, and Jakob Eriksson. Vtrack: accurate, energy-aware road traffic delay estimation using mobile phones. In *Proceedings of the 7th ACM Conference on Embedded Networked Sensor Systems*, pages 85–98. ACM, 2009.

[55] Jeffrey Hood, Elizabeth Sall, and Billy Charlton. A gps-based bicycle route choice model for san francisco, california. *Transportation letters*, 3(1):63–75, 2011.

[56] Mordechai Haklay. How good is volunteered geographical information? a comparative study of openstreetmap and ordnance survey datasets. *Environment and planning. B, Planning & design*, 37(4):682, 2010.

[57] Jean-Franois Girres and Guillaume Touya. Quality assessment of the french openstreetmap dataset. *Transactions in GIS*, 14(4):435–459, 2010.

[58] Dennis Zielstra and Alexander Zipf. A comparative study of proprietary geodata and volunteered geographic information for germany. In *13th AGILE international conference on geographic information science*, volume 2010, 2010.

[59] F Marchal, J Hackney, and KW Axhausen. Efficient map-matching of large gps data sets-tests on a speed monitoring experiment in zurich. *Arbeitsbericht Verkehrs-und Raumplanung*, 244, 2004.

[60] Nadine Schuessler and Kay W Axhausen. Map-matching of gps traces on high-resolution navigation networks using the multiple hypothesis technique (mht). *October*, pages 1–22, 2009.

[61] Mohammed A Quddus, Washington Y Ochieng, and Robert B Noland. Current map-matching algorithms for transport applications: State-of-the art and future research directions. *Transportation Research Part C: Emerging Technologies*, 15(5):312–328, 2007.

[62] B. Hummel. Map matching for vehicle guidance. In *Dynamic and Mobile GIS: Investigating Changes in Space and Time*. CRC Press, 2006.

[63] John Krumm, Eric Horvitz, and Julie Letchner. Map matching with travel time constraints. Technical report, SAE Technical Paper, 2007.

[64] Seth Rogers, Pat Langley, and Christopher Wilson. Mining gps data to augment road models. In *Proceedings of the fifth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 104–113. ACM, 1999.

[65] Sasank Reddy, Katie Shilton, Gleb Denisov, Christian Cenizal, Deborah Estrin, and Mani Srivastava. Biketastic: sensing and mapping for better biking. In *Proceedings*

of the SIGCHI Conference on Human Factors in Computing Systems, pages 1817–1820. ACM, 2010.

[66] A. Viterbi. Error bounds for convolutional codes and an asymptotically optimum decoding algorithm. *Information Theory, IEEE Transactions on*, 13(2):260 –269, April 1967.

[67] Jonathan L Herlocker, Joseph A Konstan, Loren G Terveen, and John T Riedl. Evaluating collaborative filtering recommender systems. *ACM Transactions on Information Systems (TOIS)*, 22(1):5–53, 2004.

[68] Yanjie Liu. *Designing and Evaluating a UI for Helping Users Uncover and Create Missing Road Segments on a Geowiki*. PhD thesis, UNIVERSITY OF MINNESOTA, 2013.

[69] Fernando Torre, Yanjie Liu, Zhengjie Liu, and Loren Terveen. Local knowledge matters for crowdsourcing systems: Experience from transferring an american site to china. In *Seventh International AAAI Conference on Weblogs and Social Media*, 2013.

[70] B. Aryana and T.A. ritsland. Culture and mobile hci: A review. In *Norddesign 2010 Conference*, volume 2, pages 217–226, 2010.

[71] Ravi Vatrapu and Dan Suthers. Culture and computers: A review of the concept of culture and implications for intercultural collaborative online learning. In Toru Ishida, Susan Fussell, and Piek Vossen, editors, *Intercultural Collaboration*, volume 4568 of *Lecture Notes in Computer Science*, pages 260–275. Springer Berlin / Heidelberg, 2007.

[72] Chen Zhao and Gonglue Jiang. Cultural differences on visual self-presentation through social networking site profile images. In *Proceedings of the 2011 annual conference on Human factors in computing systems*, CHI '11, pages 1129–1132, New York, NY, USA, 2011. ACM.

[73] Wei Dong and Wai-Tat Fu. Cultural difference in image tagging. In *Proceedings of the 28th international conference on Human factors in computing systems*, CHI '10, pages 981–984, New York, NY, USA, 2010. ACM.

[74] Zhenhua Dong, Chuan Shi, Shilad Sen, Loren Terveen, and John Riedl. War versus inspirational in forrest gump: Cultural effects in tagging communities, 2012.

[75] E. Ilana Diamant, Susan R. Fussell, and Fen-ly Lo. Where did we turn wrong?: unpacking the effect of culture and technology on attributions of team performance. In *Proceedings of the 2008 ACM conference on Computer supported cooperative work*, CSCW '08, pages 383–392, New York, NY, USA, 2008. ACM.

[76] Duyen T. Nguyen and Susan R. Fussell. How did you feel during our conversation?: retrospective analysis of intercultural and same-culture instant messaging conversations. In *Proceedings of the ACM 2012 conference on Computer Supported Cooperative Work*, CSCW '12, pages 117–126, New York, NY, USA, 2012. ACM.

[77] Hao-Chuan Wang and Susan Fussell. Groups in groups: conversational similarity in online multicultural multiparty brainstorming. In *Proceedings of the 2010 ACM conference on Computer supported cooperative work*, CSCW '10, pages 351–360, New York, NY, USA, 2010. ACM.

[78] Hao-Chuan Wang, Susan F. Fussell, and Leslie D. Setlock. Cultural difference and adaptation of communication styles in computer-mediated group brainstorming. In *Proceedings of the 27th international conference on Human factors in computing systems*, CHI '09, pages 669–678, New York, NY, USA, 2009. ACM.

[79] Leslie Setlock and Susan Fussell. Culture or fluency?: unpacking interactions between culture and communication medium. In *Proceedings of the 2011 annual conference on Human factors in computing systems*, CHI '11, pages 1137–1140, New York, NY, USA, 2011. ACM.

[80] Chen Zhao, Pamela Hinds, and Ge Gao. How and to whom people share: the role of culture in self-disclosure in online communities. In *Proceedings of the ACM 2012 conference on Computer Supported Cooperative Work*, CSCW '12, pages 67–76, New York, NY, USA, 2012. ACM.

[81] Qinying Liao, Yingxin Pan, Michelle X. Zhou, and Fei Ma. Chinese online communities: balancing management control and individual autonomy. In *Proceedings*

of the 28th international conference on Human factors in computing systems, CHI '10, pages 2193–2202, New York, NY, USA, 2010. ACM.

[82] Steve Friedman. #1 bike city: Minneapolis. *Bicycling*. `http://www.bicycling.com/news/featured-stories/1-bike-city-minneapolis`.

[83] Frederick Melo. Minnesota ranks as no. 2 bike-friendly state. *TwinCities.com*, 5 2012. `http://www.twincities.com/localnews/ci_20680858/minnesota-is-nations-second-most-bike-friendly-state`.

[84] J. Zacharias and J. M. Zhang. Estimating the shift from bicycle to metro in tianjin. *International Development Planning Review*, 30:93–111, 2008.

[85] Fernando Torre, S Andrew Sheppard, Reid Priedhorsky, and Loren Terveen. bumpy, caution with merging: An exploration of tagging in a geowiki. In *Proceedings of the 16th ACM international conference on Supporting group work*, pages 155–164. ACM, 2010.

[86] J Li and K Zhang. Regional differences in spatial frame of reference systems for people in different areas of china. *Perceptual and Motor Skills*, 108:587–596, 2009.

[87] G. Hofstede. *Culture's consequences: Comparing values, behaviors, institutions and organizations across nations.* Thousand Oaks: Sage Publications, Inc., 2001.

[88] J. Riegelsberger et al. The mechanics of trust: A framework for research and design. *International Journal of Human-Computer Studies*, 62(3):381 – 422, 2005.

[89] Matt Duckham, Stephan Winter, and Michelle Robinson. Including landmarks in routing instructions. *Journal of Location Based Services*, 4(1):28–52, 2010.

[90] Alexander W Siegel and Sheldon H White. The development of spatial representations of large-scale environments. *Advances in child development and behavior*, 10:9–55, 1975.

[91] Alexandra Millonig and Katja Schechtner. Developing landmark-based pedestrian-navigation systems. *Intelligent Transportation Systems, IEEE Transactions on*, 8(1):43–49, 2007.

[92] Yen-Ta Huang, An-Jung Cheng, Liang-Chi Hsieh, Winston Hsu, and Kuo-Wei Chang. Region-based landmark discovery by crowdsourcing geo-referenced photos. In *Proceedings of the 34th international ACM SIGIR conference on Research and development in Information Retrieval*, pages 1141–1142. ACM, 2011.

[93] Kai-Florian Richter and Stephan Winter. Harvesting user-generated content for semantic spatial information: The case of landmarks in openstreetmap. In *Proceedings of Surveying & Spatial Sciences Biennal Conference*, 2011.

[94] Joseph L Fleiss. Measuring nominal scale agreement among many raters. *Psychological bulletin*, 76(5):378, 1971.

[95] John Uebersax. Raw agreement indices. *Statistical Methods for Rater and Diagnostic Agreement*. `http://www.john-uebersax.com/stat/raw.htm`.

[96] Jacob Cohen. A coefficient of agreement for nominal scales. *Educational and Psychological Measurement*, 20(1):37–46, April 1960.

[97] John S Uebersax. Diversity of decision-making models and the measurement of interrater agreement. *Psychological Bulletin*, 101(1):140, 1987.

[98] J Richard Landis and Gary G Koch. The measurement of observer agreement for categorical data. *biometrics*, pages 159–174, 1977.