

Designing and Evaluating a UI for Helping Users Uncover and Create
Missing Road Segments on a Geowiki

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

Yanjie Liu

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE DEGREE OF
MASTER OF SCIENCE

Advisor: Loren G. Terveen

November 2013

© Yanjie Liu 2013

Acknowledgements

I would like to acknowledge the support of many people who made my thesis possible. First and foremost, I would like to thank my advisor, Loren Terveen, for his guidance throughout this process. I would like to thank Brent Hecht and Yuqing Ren, my committee who have reviewed my thesis and given me useful advice. I also want to thank other people in Cyclopath team for their support along the way, and Cyclopath users who gave me a lot of valuable feedback. I would also like to thank my family and friends who encouraged and supported me during my thesis work.

Abstract

Incomplete map data is a persistent problem in map applications, especially those that provide route-finding services. In order to solve this problem, map matching algorithms that take advantage of GPS traces to detect missing map data are emerging and developing quickly in recent years. A number of applications have used this technique to update and repair maps.

Cyclopath is a geowiki system designed for cyclists that has been deployed in the Twin Cities in Minnesota for five years. In addition to being a collaboratively edited geographic system, it is also a computational system that computes bike-friendly routes for cyclists. Like other map applications, it too has suffered from the problem of incomplete map data, resulting in that users are not getting the best bike routes.

In this work, we designed, implemented, and evaluated a user interface for Cyclopath that lets users step through a set of map update suggestions computed by a map matching algorithm and automatically updates the map as necessary based on user decisions. The interface was evaluated by a user-based experiment to test and validate its user acceptance and usability. The results indicate that this interface is able to effectively help users uncover and fix missing blocks in the map.

Table of Contents

List of Tables	v
List of Figures	vi
1 Introduction	1
1.1 Background	1
1.2 Research Motivation	3
2 Related Work	5
2.1 GPS Data Mining Applications	5
2.2 Geowikis and Eliciting Volunteer Work	7
3 Cyclopath	11
3.1 Cyclopath Platform	11
3.2 Users and Volunteers in Cyclopath	15
4 Conflation Feature Design and Implementation	17
4.1 Interaction and UI Design	17
4.2 Implementation	27
4.2.1 Implementation Environment	27
4.2.2 Technical Implementation and Rationale	27
5 Evaluation	32
5.1 Research Questions	32
5.2 Methods	32
5.3 Analysis and Results	36
5.3.1 Suggestion Presentation	37

5.3.2 Map Editing.....	41
5.3.3 Other Findings.....	47
6 Conclusions and Future Work.....	50
Bibliography.....	52
Appendix	54
Cyclopath Interview Script	54

List of Tables

Table 3.1: Geographic and non-geographic features in Cyclopath (Blocks are atomic segments of the roads and trails that make up the transportation network.).....	13
Table 5.1: Interview Participants	34
Table 5.2: Difficulties of map editing in the four cases	46

List of Figures

Figure 3.1: Cyclopath web Client	12
Figure 4.1: Flowchart of conflation process at each suggestion	18
Figure 4.2: Interface after map matching finished	20
Figure 4.3: One suggestion with one new block (between N Oxford St and Chatsworth St) and two modified blocks (on N Oxford St and Chatsworth St respectively)	22
Figure 4.4: One suggestion with two new blocks connected together. One is between W Englwood Ave and W Seminary Ave, and another is between W Seminary Ave and W Hubbard Ave.	23
Figure 4.5: One suggestion with two new blocks connected with a modified block in the middle.....	23
Figure 4.6: Interface after ignore one suggestion	25
Figure 4.7: Selecting blocks where actually rode	25
Figure 4.8: Interface after finish selecting blocks	26
Figure 4.9: Interface after accept one suggestion	26
Figure 4.10: Track panel when conflation is in progress	29
Figure 4.11: Grouping relevant new blocks and new intersections together	29
Figure 4.12: Highlighting selected suggestion.....	30
Figure 5.1: Suggestion with its tooltip	41

Figure 5.2: Two sub cases that a new block is at the correct position but its geometry is not correct 43

Figure 5.3: Two sub cases that the geometry of the new block is the same as in fact, while its position is not exactly the same as where it should be 44

Figure 5.4: Neither the geometry nor and position of the new block is exactly the same as in reality 45

Chapter 1

Introduction

1.1 Background

Over the past decade, open collaboration systems have been emerging and developing quickly as powerful methods for creating community-maintained artifacts of lasting value and providing value to others. For example, Wikipedia was created through extremely huge amounts of contributions by millions of editors, and Yahoo! Answers has received over one billion answers and is visited by over tens of millions of users per month. The number of participants and contributions of an open collaboration system it maintains appear to be highly correlated with its survival and success. One of the important challenges in open collaboration systems is eliciting sufficient contributions from community members. Users in online communities may have difficulty finding opportunities to make contributions, or cannot understand the value of their contributions to the community. Thus, eliciting more volunteer work has become critical in helping open content communities live longer.

The open collaboration system that this project focuses on is for cycling community. Cycling is popular as an enjoyable form of commuting, recreation, and for health and fitness, not only because of its low cost but also because it is

environmentally friendly. However, cycling experiences are often not satisfying. In the physical world, finding a good cycling route is not easy because transportation networks are mainly designed for motor vehicles. There are no bike lanes on many roads, and construction often occupies bike lanes. In the digital world, most maps and navigation services and applications are designed for driving, public transit, and walking. There are very limited online resources and platforms that provide support particularly for cyclists. Thus, it is important to develop support for the cycling communities.

Looking to support cycling communities, recent websites combining map-based interfaces with open content models started to emerge. Geowikis such as OpenStreetMap allow users to provide and edit geographic content collaboratively. In order to provide cyclists better support, researchers in GroupLens have developed Cyclopath, a geowiki that provides route-finding services for cyclists in the metropolitan area of Minneapolis-St. Paul, Minnesota, USA. Users of Cyclopath are able to not only get personalized bike routes, but also edit the map itself in order to improve and keep it updated. Cyclopath also has a mobile app that allows users to request routes from Android phones and record GPS traces for their rides. Cyclopath is a user-maintained geographic system where user contributions are the key factor for its survival. Furthermore, Cyclopath has many areas that need geographic volunteer work, such as areas with missing road segments, wrong road geometries, and missing road names. A key challenge is to elicit work that is both sufficient and focused where needed.

1.2 Research Motivation

Due to the increasing pervasiveness of GPS and remote sensing technology in a variety of devices in our daily life, GPS data is becoming increasingly abundant and has become available to many of geographic applications. One potential use of GPS data is to generate and update the geometry and connectivity of road maps, using what are known as “map generation” or “map inference” algorithms. On the other hand, GPS data can also enhance our understanding about users in these geographic applications, such as their behavior and familiarity with different parts of the map. On Cyclopath, collecting GPS tracks (we call it “track” in Cyclopath) is a relatively new feature. Users can record a track while on a ride and view the information (such as length, duration, etc.) of tracks after saving them from the mobile app. Therefore, there is still a lot of room to gain more value from GPS data recorded by users and explore its potential on Cyclopath.

On a map application for cyclists like Cyclopath, the most common task performed by users is to find good routes that match their personalized requirements. But the map often has some missing shortcuts, sidewalks, alleyways, and bike trails, which brings some obstacles to routing. Fortunately, Torre et al. [21] of the Cyclopath team created a map matching and building algorithm (we call it the conflation algorithm in this thesis), which can help address this issue. The algorithm has been demonstrated to be able to not only match GPS observations to blocks, but also detect and create missing blocks. However, the conflation

algorithm is imperfect in some aspects: a) When testing to see if the algorithm can detect blocks if they were removed, they found that it was hard to find the right cutoff distance to make a trade-off between false negatives (blocks that should have been created but were not) and false positives (blocks that should not have been created). b) The algorithm degrades if the map data set has too much missing data. c) There's still room for improvement in the geometry creation part of the algorithm in order to better handle noise. These flaws are hard to solve from the algorithm side alone. A human-computer cooperative approach should be explored.

The work described in this thesis attempted to address this problem in this direction. We designed and implemented an interactive interface, which lets users step through a set of map update suggestions computed by a map matching algorithm and automatically updates the map as necessary based on user decisions. We evaluated the interface, finding that this interactive interface is able to help users understand the suggestions and make the correct decisions between accepting or rejecting suggestions. We also proved that this interface provides users opportunities to edit the map in a simpler way.

The rest of the thesis is organized as follows: Chapter 2 describes related work. Chapter 3 introduces Cyclopath and the context for this research. Chapter 4 contains the design and implementation of the conflation user interface. In Chapter 5, we present the design of the experimental evaluation, and analyze the results in detail. Chapter 6 draws the conclusions of this thesis and discusses some ideas about potential future work.

Chapter 2

Related Work

2.1 GPS Data Mining Applications

In recent years, more and more users have started recording their outdoor movements with GPS traces for different reasons, such as life logging, sports activity analysis, and travel experience sharing. Meanwhile, a number of studies have been conducted based on mining GPS traces. These works include mining individual and multiple users' location histories for identifying their activities at certain locations, inferring the users' movement among different locations, and recommending personalized information for users. Liao et al. [8] described a system that can build personal maps customized for each user automatically and infer his daily activities and movements from raw GPS data. Marmasse et al. [11] predicted where a person may be going from a list of previously visited destinations using a Bayes classifier, as well as HMM and histogram matching. On the other hand, by taking advantage of multiple users' location histories, some recommender systems, such as CityVoyager [20], etc, have been designed to recommend points of interest for users. Takeuchi et al. attempted to recommend shops to users based on their individual preferences estimated by analyzing their past location histories. Zheng et al. [24] demonstrated by using the location data based on GPS and users'

comments at different locations that they could recommend interesting locations to users and possible activities that could be performed there.

By mining GPS data, we are able to not only understand human behaviors and make recommendations for them, but also refine and update digital maps. A number of Websites or applications, such as TomTom¹ and Waze², which leverage crowdsourcing to enable people to collaboratively update their map, have appeared on the Internet. By uploading GPS traces to these communities, people are able to visualize and manage their GPS logs on a Web map and make necessary changes to the map dataset manually by referring to aerial imagery and their GPS traces. In addition, automated map matching and inferencing using GPS traces has gained much attention recently. Part of that is because large-scale GPS traces are increasingly abundant and map matching algorithms are becoming more and more accurate. Most existing map inference methods rely on densely-sampled and uniformly distributed GPS traces. Biagioni et al. [2] presented an extensible map inference pipeline that is able to mitigate GPS error, apply to less-frequently traveled roads, and scale to large datasets. CrowdAtlas [23] is a system that automatically and continuously updates existing maps based on people's travels, either individually or crowdsourced. Waze has the most similar features to our conflation feature. It can detect missing roads by taking advantage of users' GPS tracks and report to users. But Waze does not have an interface to allow users to verify the suggested new roads and add them to the map automatically. In contrast to these applications, our conflation feature on Cyclopath is able to utilize single tracks from individual users to

¹ www.tomtom.com/mapshare/tools

² www.waze.com

detect missing map data and let users step through a set of map update suggestions computed by the map matching algorithm. This feature can also automatically make the necessary map editing operations indicated by the suggestions after users accept them.

2.2 Geowikis and Eliciting Volunteer Work

In the Web 2.0 era, there have been a growing number of geographic-based open collaboration communities -- called geowikis. The use of geowikis and the fields it involves extend to many aspects of our lives. FixMyStreet³ is being used throughout the UK, allowing users to enter a postcode, locating the problem, such as potholes, broken street lights and similar problems, on a map and entering the details of the problem, which are then forwarded to local governments. SeeClickFix⁴ allows anyone to report and track non-emergency issues all over the world on the map. Problems are fixed by publicly broadcasting to the appropriate parties (such as community groups, media organizations and governments) for resolution, working collectively to raise the profile of key concerns, or by taking direct action. Beside these community-oriented sites, mapping-oriented sites like Google My Maps enable users to create personalized annotated maps by collaboratively editing geographic points, paths, and polygons, all of which can be

³ www.fixmystreet.com

⁴ seeclickfix.com

annotated with text, images, and videos. Google Map Maker⁵ allows users to directly edit Google Maps data (in some countries) and submit those changes for inclusion in the public map. The most famous geowiki system, OpenStreetMap (OSM)⁶, is an ongoing project building a worldwide street map, designed to create and provide free spatial data. It is based on an open content philosophy, and combines geographic data gathered from users' volunteered work, allowing users to create, edit, download, and use OSM data in their applications. There are also great amounts of academic research on geowikis. Mashhadi et al. [12] investigated crowd-sourcing as a sustainable means of maintaining accurate information about points of interests in dynamic urban environments, which demonstrates the efficacy of geowikis in information update and maintenance. In the research conducted by Mooney et al. [14] about the collaborative nature of spatial data collection and editing in OSM, they found that there is no explicit social network structure to the OSM spatial database. This could provide us inspiration for improving the social factors in geowikis by strengthening the interaction among users based on their contributions. Finally, Cyclopath is a geowiki system designed for cyclists. One of the previous studies on Cyclopath evaluated of the effectiveness of the Cyclopath geowiki model [17]. It showed empirically that both geowiki and computational geowiki features are necessary for bicycling. Cyclists have useful knowledge that is available only from cyclists and cyclist-oriented automatic route-finding is enhanced by user input.

The success of open collaboration communities largely lies in people's participation

⁵ www.google.com/mapmaker

⁶ www.openstreetmap.org

in building the content, but getting users to volunteer their efforts is an ongoing challenge. Much work has explored techniques that motivate participation, such as comparisons [5, 6, 7], setting specific and challenging goals [10], familiarity [3, 4, and visual prompts [18]. Anderson, who leads the SETI@home and BOINC projects [1], proposed that credit is the primary incentive for many participants, which has been widely used by many forum sites. He also proposed using visualizations through a screensaver application to give feedback about participation. Ling et al. [9] describe several experiments that evaluated techniques based on factors like uniqueness and goal-setting. Cosley et al. have developed automated techniques for eliciting work. They find that asking users to do familiar work is the most effective way to motivate their participation in Wikipedia and in MovieLens, which means encouraging users to edit articles that are related to those they have edited in Wikipedia and asking users to edit movies they have rated in MovieLens. Priedhorsky et al. [18] proposed two techniques to elicit and focus user work: using familiarity to direct work opportunities, and visually highlighting where work is needed. In another related work with regards to specializations of contributions on Cyclopath, Masli et al. [13] found that there tend to be two obvious types of geographic shapes where users may make most contributions: area - editing is focused on areas, such as the neighborhood surrounding one's home, and route - editing is focused in a "linear" way, like along a route of a work commute or favorite exercise trail. This gives us strong evidence that users tend to make contributions along their rides where they might have better local knowledge. GPS

tracks are a strong indicator of people's familiarity with a route or an area and a possible good prompt for cyclists to recall their cycling experience. By making use of GPS traces, we should be able to elicit more user work by taking users to a familiar place where they might have better knowledge.

The related work above has shown us opportunities for developing a user interface that helps users discover and fix map problems by taking them to the places along their track where they are familiar with. This is just the area this thesis work has focused on. At the same time, this might be a great opportunity, if further explored, for eliciting more volunteer work to make contributions to a successful open collaboration community of Cyclopath.

Chapter 3

Cyclopath

This chapter gives an overview of Cyclopath, which this thesis is focused on, especially its system platform as well as the user group and the volunteering work on it.

3.1 Cyclopath Platform

Cyclopath is a web-based mapping application and is targeted to the route finding needs of cyclists. Compared to other map applications, Cyclopath is out of the ordinary because it is a geowiki, which means that the map data can be edited by any user. The original data in Cyclopath was imported from the Minnesota Department of Transportation. Since 2008 when it went live for the public, Cyclopath has become a community resource and a booming research platform. There is a wide range of research done on it, including generating personalized bike routes [19], analysis of user behavior [15], introducing tags into Cyclopath [22], task specialization on Cyclopath [13], and how volunteer work can be elicited [18]. The right side of the interface is the map that shows roads, points, and regions with highlights if they have annotations. The left panel is used for a number of associated functions including editing properties of items selected on the map.

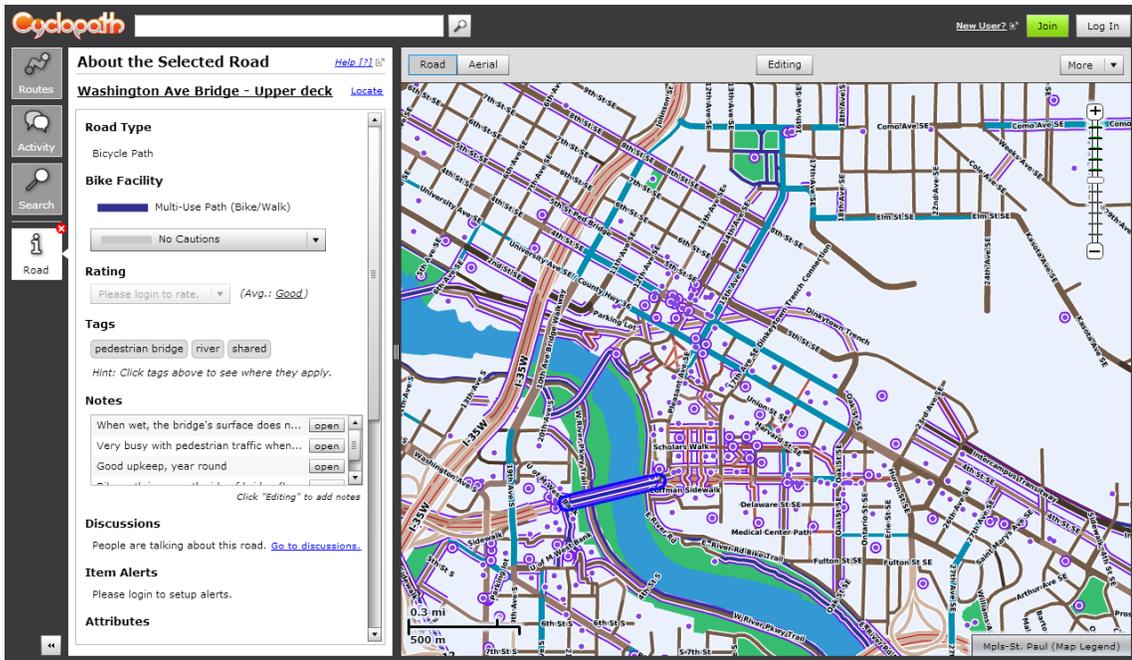


Figure 3.1: Cyclopath web client

As a geowiki, Cyclopath gains much of its value from the contributions made by its users, including geographic and non-geographic data. Geographic information consists primarily of roads, trails, and points of interest, while annotations, ratings and discussions serve as non-geographic information. Table 3.1 shows the statistics of different features on Cyclopath on May 2013.

As the main components of transportation networks on Cyclopath, roads and trails can be added, edited and deleted by users. Points of interest, such as bike racks, restaurants, water fountains or other locations considered relevant, are used as starting and end points in routing and serve as landmarks when browsing the map. Regions include cities, neighborhoods, and other defined geographic regions, such as Minneapolis or University of Minnesota.

Feature	Number
Blocks	15,6373
Points	3,821
Regions	483
Tags	826
Notes	3,119
Bikeability Ratings	166,721
Discussions	405

Table 3.1: Geographic and non-geographic features in Cyclopath (Blocks are atomic segments of the roads and trails that make up the transportation network.)

Non-geographic features are closely linked to geographic features and they are of high value on Cyclopath:

- (a) **Tags:** Tags are short textual labels that can be attached to points, blocks or regions and they are used in two systems functions: Point and region tags can help users filter points and regions that are displayed on the map based on their preferences. Block tags can help modify route finder preferences to provide every user with a personalized route.
- (b) **Notes:** Notes are another important part of Cyclopath. Notes can provide information about the points of interest and blocks. These can be used when doing routing, especially for providing information for users about unfamiliar areas.
- (c) **Bikeability:** Users can rate the bikeability of individual blocks or trail

segments in a scale from impassable to excellent. Ratings can enhance the route finder's computations. Every user may have their own rating for any block, and the system also provides an aggregated rating based on all of the personal ratings entered by users.

- (d) **Discussions:** Discussions are another type of non-geographic feature which allows users to talk directly to each other on Cyclopath. It also offers a link to the specific place that people talk about, so that users can easily find it on the map.

Cyclopath also has a mobile extension which is an Android app that gives users access to Cyclopath functionality while on the go. They can get routing information whenever on a ride, not having to plan a route and trying to remember it before starting off. Besides, the app is able to collect users' GPS traces and upload them to the Cyclopath server. Before the conflation feature discussed in this project is released, we offer only very basic features for tracks, allowing the user to record tracks and view basic information such as date and length.

From a previous study [18], we know that the Cyclopath database contains thousands of errors, including approximately 7,000 missing X intersections and 6,300 missing T intersections. Missing X intersections are places where two blocks cross each other geometrically but no network node exists, while missing T intersections means places where a dead-end block came within 20 meters of intersecting another block. Although these potential intersections can be identified automatically, judgment from users is required to determine whether a node is actually appropriate because a missing

X intersection may consist of one road on a bridge over another, and a missing T may consist of two roads which come close but don't actually meet. Moreover, there are also a number of missing roads (such as sidewalks, alleyways, and bike trails) on Cyclopath. If we are able to detect these errors and suggest them to users for verification, these errors might be easily fixed.

3.2 Users and Volunteers in Cyclopath

At the time of this research, Cyclopath had over 5,100 registered users and 1,000 editors. Like other open content systems, Cyclopath has a small group of core contributors who do much of the work, but most users do little or even none (e.g., 611 logged-in users have saved at least one revision, but only 14 have saved more than 100).

On the mobile side, more than 240 users have logged in to Cyclopath Mobile. Requesting routes is the most frequent feature that has been used on mobile. Users have requested more than 600 routes and recorded more than 100 tracks from their devices, but all of these tracks are recorded by only 27 of our mobile users.

In previous work, Panciera et al. [16] surveyed Cyclopath users to identify their motives for participation and contribution. The most common reason contributors gave for joining Cyclopath was to edit content, while consumers tend to join to get bicycling information. As users contributed more, their motivations diverged more sharply from those of consumers. Further, contributors gave

different reasons for beginning and for continuing to contribute on Cyclopath. They often started editing in order to fix specific problems they noticed, while they continued editing because they wanted to benefit other cyclists and because of a general commitment to bicycling. To better motivate contributors for continuing their contributions, creating some specific work opportunities for them to act on might be an appropriate way.

Chapter 4

Conflation Feature Design and Implementation

4.1 Interaction and UI Design

Due to the limitations of the conflation algorithm, it cannot do the whole job without human assistance. The main goal in this research project is to design and implement a user interface for conflation to let users be involved in this activity. In this process, the user is presented with the step-by-step suggestions that are generated by conflation algorithm, in which possible missing blocks are pointed out. The user can go through the process, accepting, rejecting or modifying what the system suggests at each step.

After roughly sketching out the general concept of the conflation feature, we developed the concept into an interaction design by creating an interaction flow (as shown in Figure 4.1):

After identifying the general interaction flow, we created a low fidelity prototype by using a mockup tool (Balsamiq Mockups) for brainstorming, collecting various possible design solutions. During the prototyping process,

several iterations of design were implemented to optimize the design, and meanwhile, the information structure was refined, and the framework was defined. The prototype is shown in Figure 4.2- 4.9.

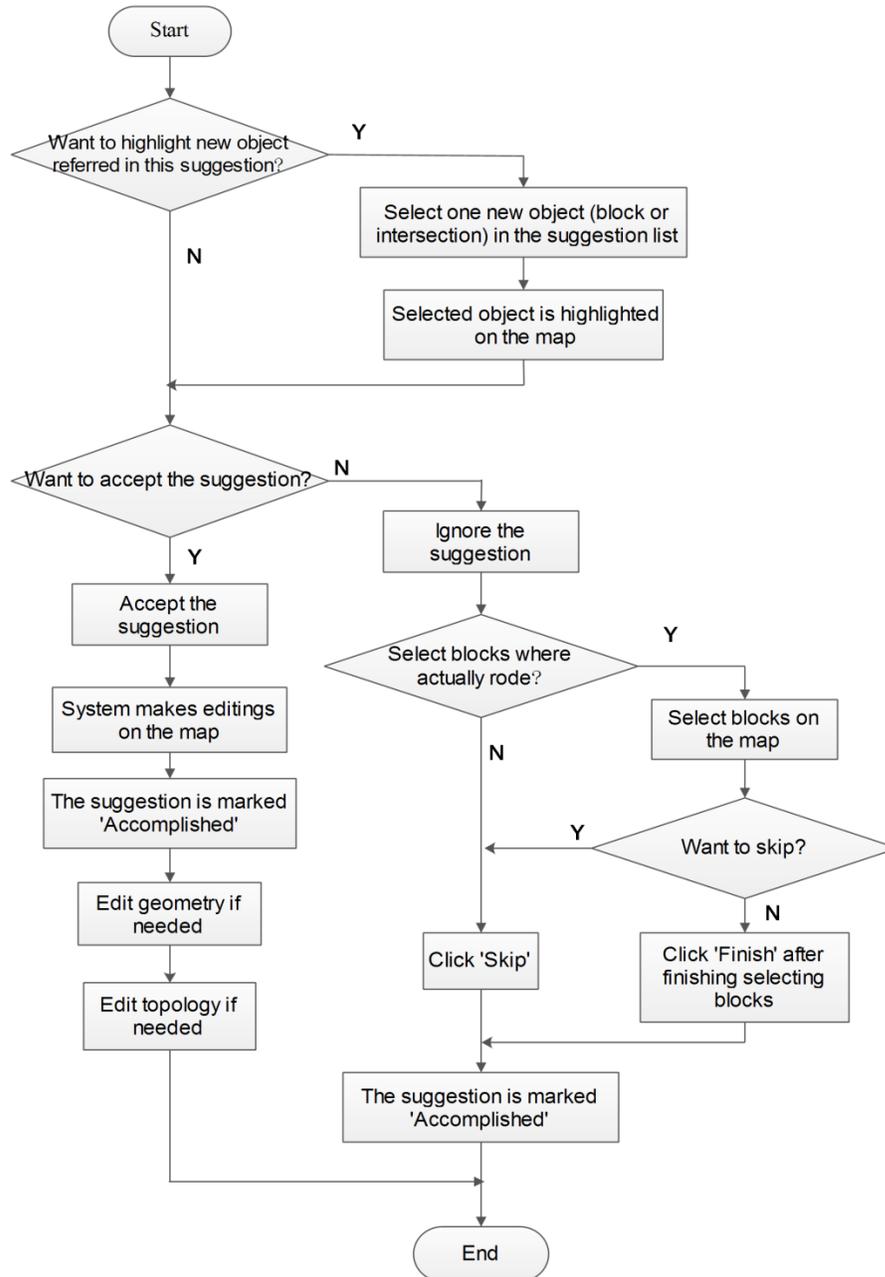
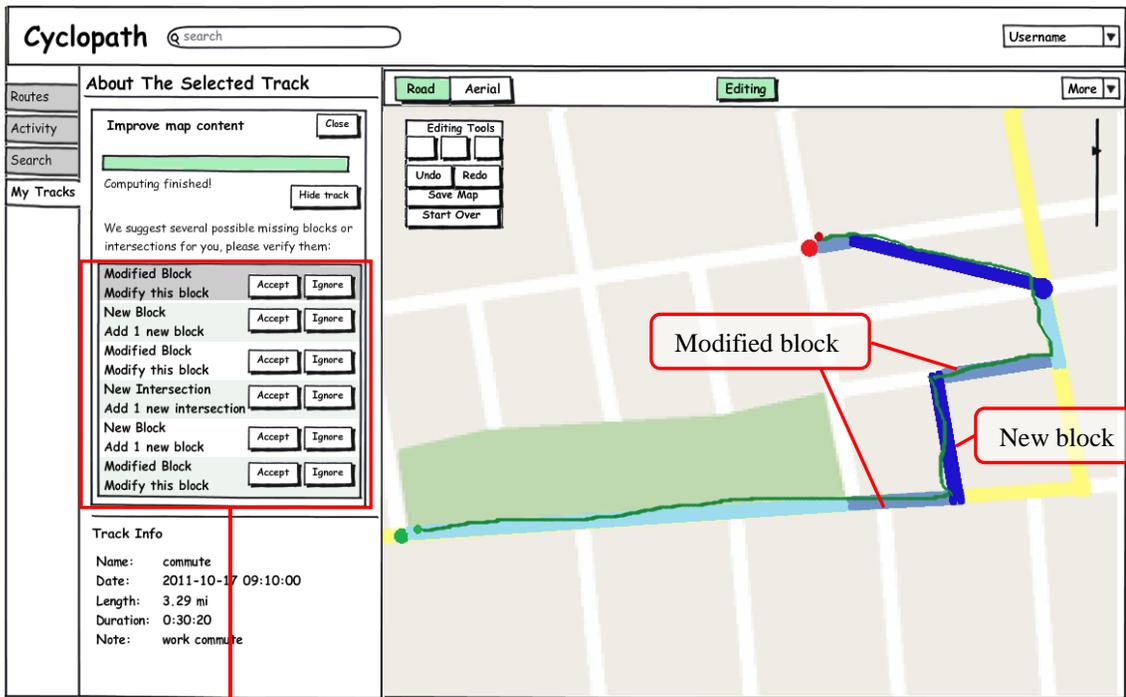


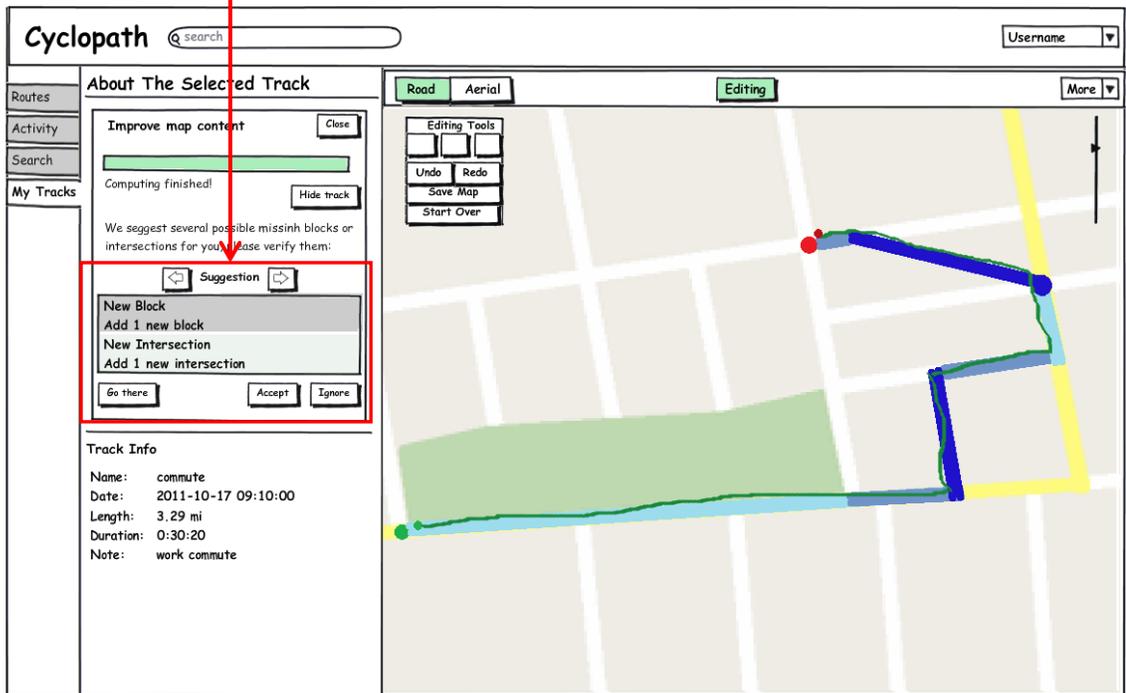
Figure 4.1: Flowchart of conflation process at each suggestion

In the conflation task, users will see how their track matches to the map data. Parts of the track can match to roads on the map perfectly, while other parts will not. For example, the user may ride on a trail or road that does not exist on our map -- in this case the system will suggest some new roads. Sometimes the system's guess is right, while sometimes it is wrong. Thus user verification is quite important.

As shown in Figure 4.2 and Figure 4.3, the system generates a route by matching the track to the map data. Segments of the route have three colors: light blue represents the parts where the track matches to the route perfectly (normal part of route); dark blue indicates the places where the system suggests new blocks; royal blue colors where an existing block needs to be modified (i.e. extended, split or both). The criteria for using these colors are based on the colors of the existing revision interface on Cyclopath, where new blocks are shown in dark blue and blocks whose geometry has changed are indicated with royal blue. For each suggestion, the user has two options: accept – accept to add the suggested new blocks, or ignore – reject because the suggestion is wrong or because the user cannot tell whether the suggestion is right or not. When blocks, the route, and the track lay one above another, it is hard to see objects in lower map layers clearly. Therefore, we also allow users to hide the track in order to see the objects below it.



(a) Initial version



(b) Final version

Figure 4.2: Interface after map matching finished

From the first iteration to the final iteration of our prototype (as shown in Figure 4.3), several important revisions were made. These design decisions can serve as key focuses in the user evaluation:

- In the initial design, all suggestions were displayed within one list. Even though this design could allow users to view all suggestions at the same time and switch to highlight any of the suggestions on the map easily, it appeared to be providing too much information in one list and made users feel overwhelmed. Therefore, we revised the design, showing one suggestion at a time, and in this way allowing users to handle the suggestion one-by-one.
- We had both new blocks and modified blocks in the suggestion list in our first design iteration. Modified blocks are blocks that need to be either extended or split, which are always attached to new blocks. For example, in Figure 4.3, in order to add the new block in the middle, two existing blocks (N Oxford St and N Chatsworth St) which the new block will be connected to, have to be split. New blocks are the main tasks to be accomplished, so it makes sense to focus the suggestion interface around these tasks. Also, from what we know about the users on Cyclopath, only a few of them are proficient with map editing tools and most of the users do not quite understand how map editing works (e.g. adding one new block to make it connect to another existing block, which requires adding a vertex, splitting the existing block, and connecting them together). For these reasons, we simplified the presentation of the content in the suggestion list, only showing

users new blocks.

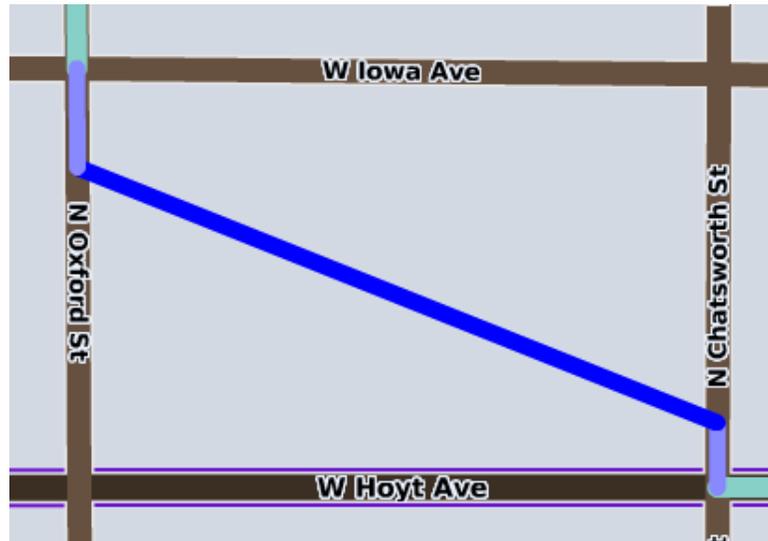


Figure 4.3: One suggestion with one new block (between N Oxford St and Chatsworth St) and two modified blocks (on N Oxford St and Chatsworth St respectively)

- Another important refinement we made during the prototyping process was the grouping of relevant suggestions. For the new block and the modified blocks which are connected with each other, we take them (new and modified) together as a set of relevant blocks. We decided to designate this set of blocks as a single, coherent suggestion. There are two different cases where we need this kind of grouping:
 - 1) Two new blocks are directly connected together (as shown in Figure 4.4). It is quite easy to understand why we should group them together in this case, because this usually happens when there are two segments of roads missing on the map.
 - 2) Two new blocks are connected via one modified block (as shown in

Figure 4.5). In this case, the two new blocks are located very close to each other and they are connected to the same modified block, so taking them as one single suggestion is reasonable.

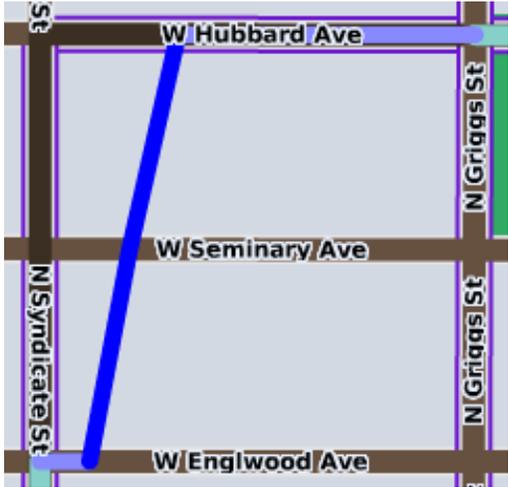


Figure 4.4: One suggestion with two new blocks connected together. One is between W Englewood Ave and W Seminary Ave, and another is between W Seminary Ave and W Hubbard Ave.

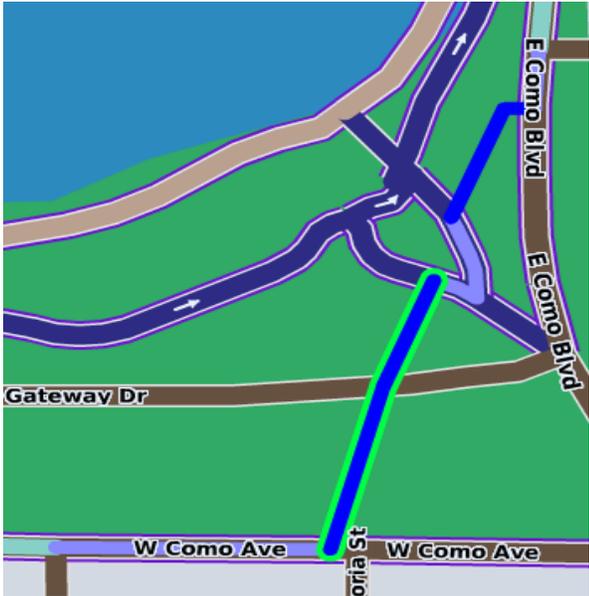


Figure 4.5: One suggestion with two new blocks connected with a modified block in the middle

Figure 4.6 is the interface after the user ignores one suggestion (e.g. add a new block and a new intersection). All of the suggested new blocks and new intersections along with modified blocks of this suggestion will be removed from the route. Afterwards, we let users select the blocks where they actually rode on, and these blocks become a normal part of their route (as shown in Figure 4.7 and 4.8).

If users want to accept a suggestion (as shown in Figure 4.9, e.g. add a new block), then the system will automatically do all of the necessary operations (e.g. Create a new block, create a new intersection, split a block, etc.). This is much simpler than using existing map editing tools and manually making the changes. Besides, we do not make the accepted part covered with the normal route color (or being normal part of the route) after the system finishes editing operations. This is because if the suggested new block is not in the right position or its geometry is not as what users expect, it will need to be further modified.

After one suggestion is accepted or ignored, it will be marked as “accomplished”. In the whole process of this activity, we allow users to stop and exit at any point of this task.

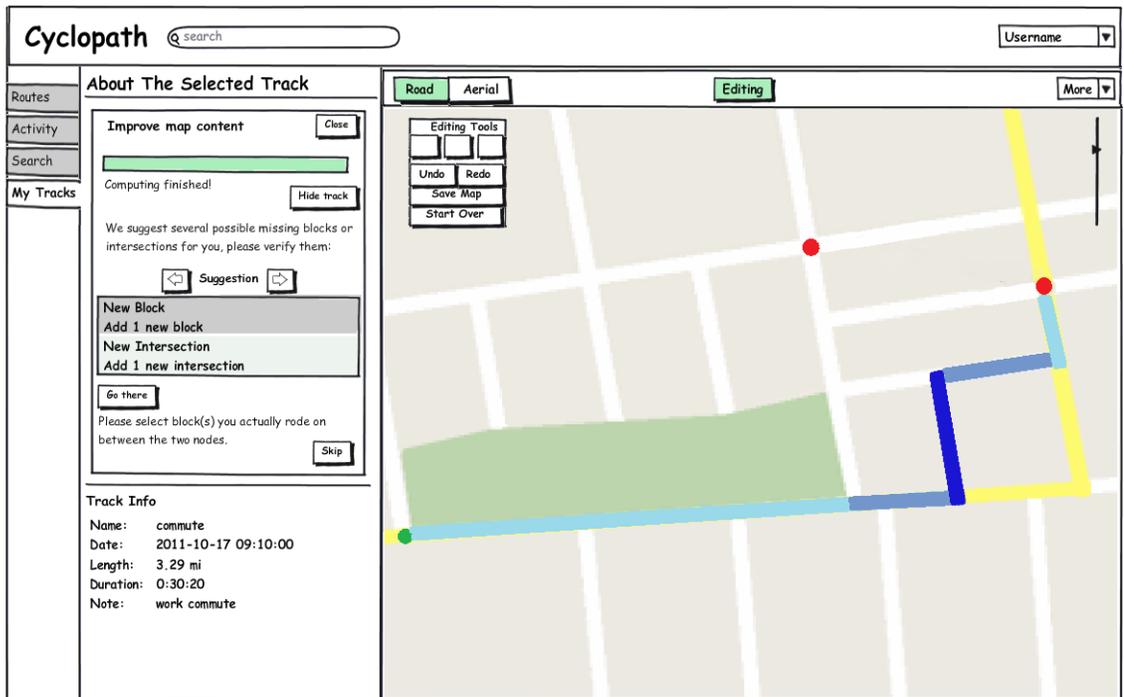


Figure 4.6: Interface after ignore one suggestion

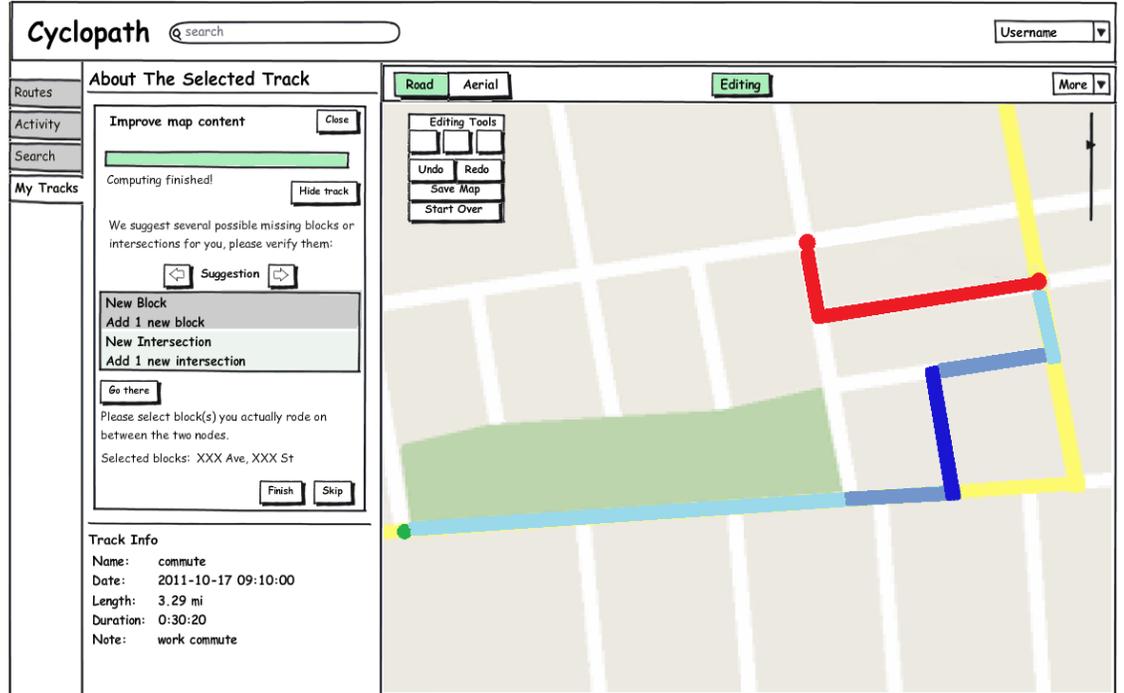


Figure 4.7: Selecting blocks where actually rode

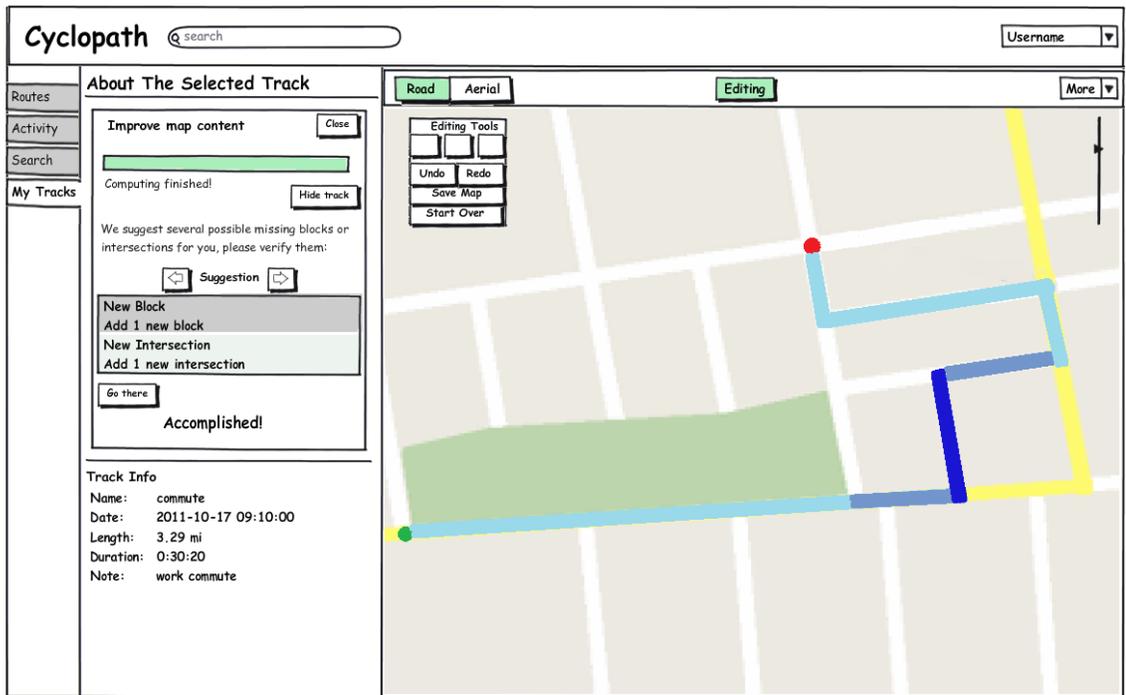


Figure 4.8: Interface after finish selecting blocks

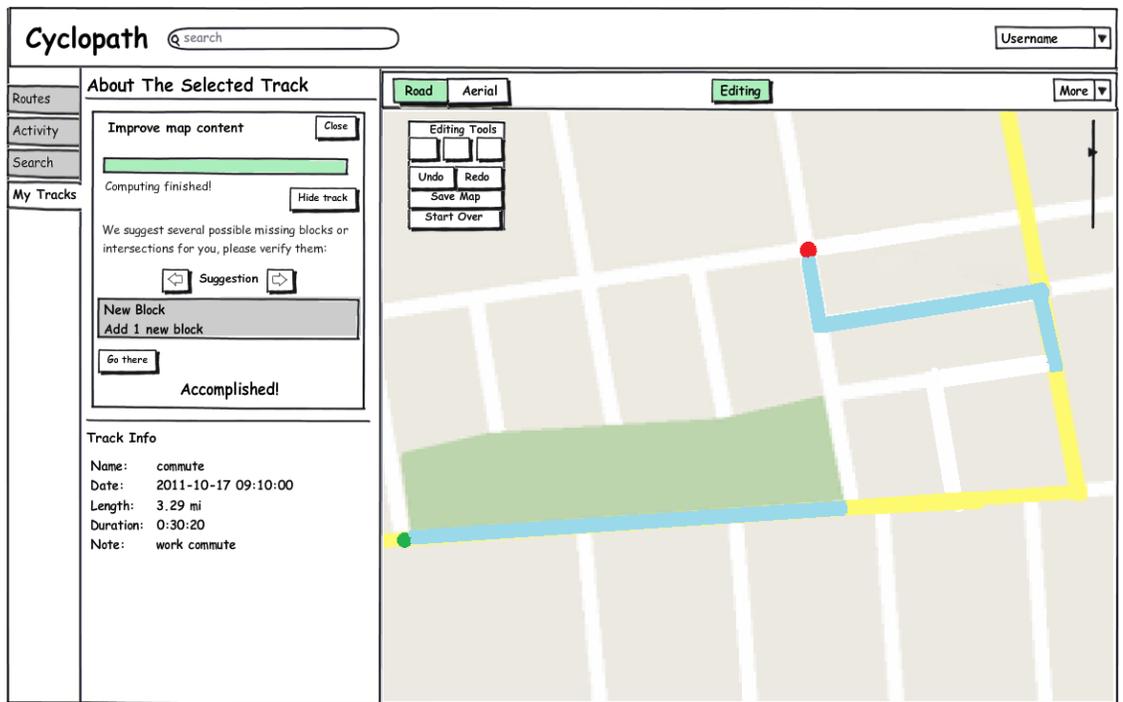


Figure 4.9: Interface after accept one suggestion

4.2 Implementation

This section explains the environment in which the conflation feature was implemented. Then, the implementation of some representative screens of the interface designed in section 4.1, as well as the technical rationale behind them is presented.

4.2.1 Implementation Environment

On Cyclopath, the chosen programming language is Python for the server application and Flash/Actionscript for the web client. The server fetches and stores most of this data directly in our PostgreSQL database. When requesting routes - these are passed on using a raw Internet socket to a route daemon, which is a standalone service where all the current transportation network information is stored in memory. Conflation is done by another "service", just like the route finder daemon. The conflation job is stored in the database and the service looks at the job queue in the database to fetch conflation results. The implementation work in this thesis is mainly on the web client.

4.2.2 Technical Implementation and Rationale

When we implemented our design, going from the low fidelity prototypes to the final system, the design was refined again. Since a prototype is limited in functionality, there are some gaps between the prototype and final deliverable. In

this section, we mainly discuss some of the design refinements, when taking more UI design fundamentals (e.g. consistency, error preventing, etc.) into consideration in the process of implementing the design from prototypes to the real system.

Figure 4.10 shows interface of the track panel, in which the layout is mostly consistent with other item panels on Cyclopath. Tracks, just as watched regions, are considered private information in Cyclopath. That is why we use pink (which means private on Cyclopath - other users cannot see this information) as the background color. In addition, the red progress bar is updated as the conflation computing runs.

When there are new blocks or new intersections that are relevant to each other, we group them together and place them in one suggestion. One difficulty here is how to resolve all the new blocks, new intersections, and modified blocks from the source xml file of conflation results and make necessary groupings. Conflation results sent from the server consist of a sequence of route steps, and each route step has two node ids representing the location of the two end nodes of a block on the map. If there is one new block whose end node is of the same value as of another new block, then the two blocks are connected together; thus we want to place them in one group. As for new intersections, if two modified blocks are connected by the same node id, then there must be a new intersection between them. Figure 4.11 is an example of grouping two new blocks with one new intersection together.

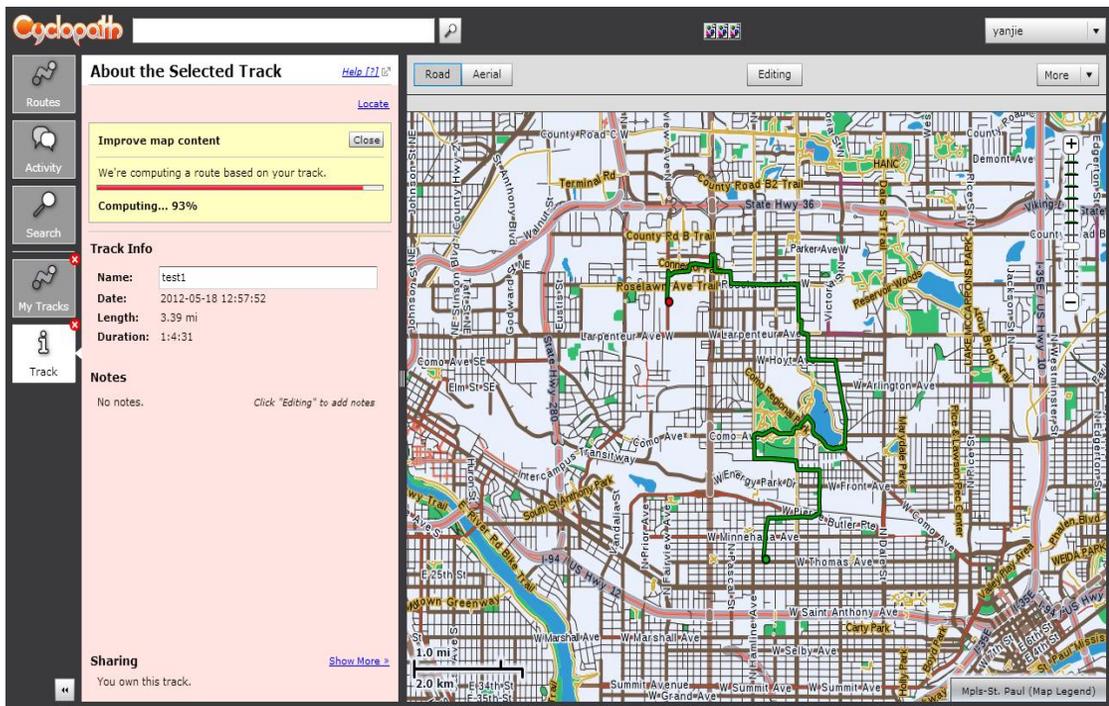


Figure 4.10: Track panel when conflation is in progress

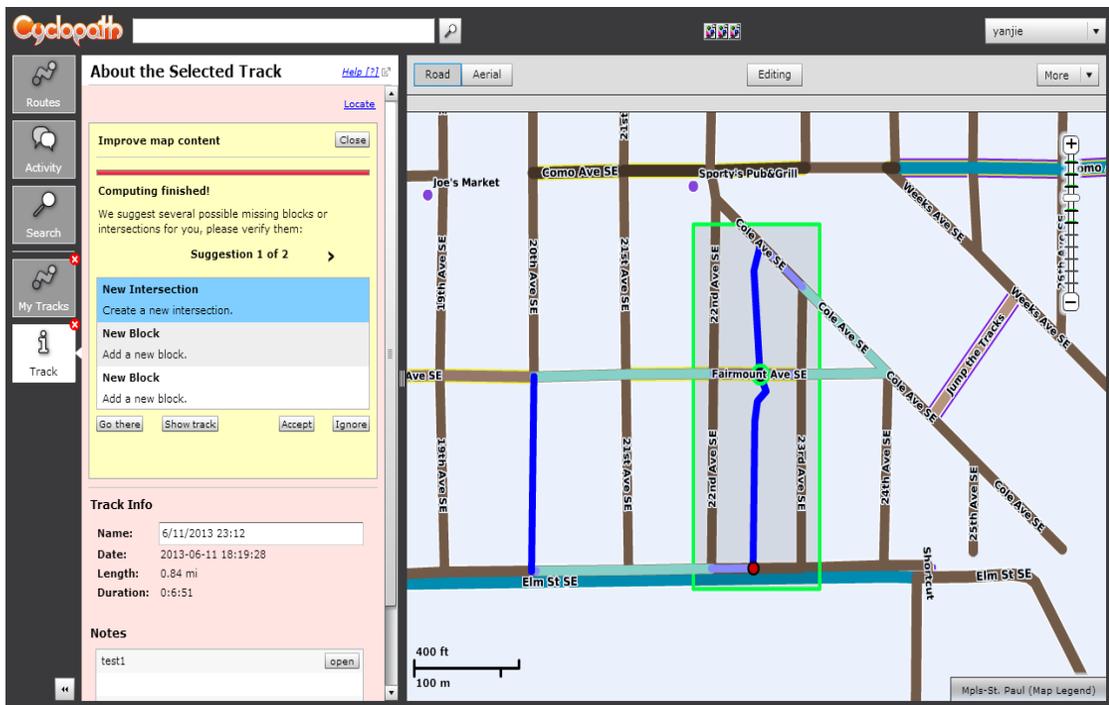


Figure 4.11: Grouping relevant new blocks and new intersections together

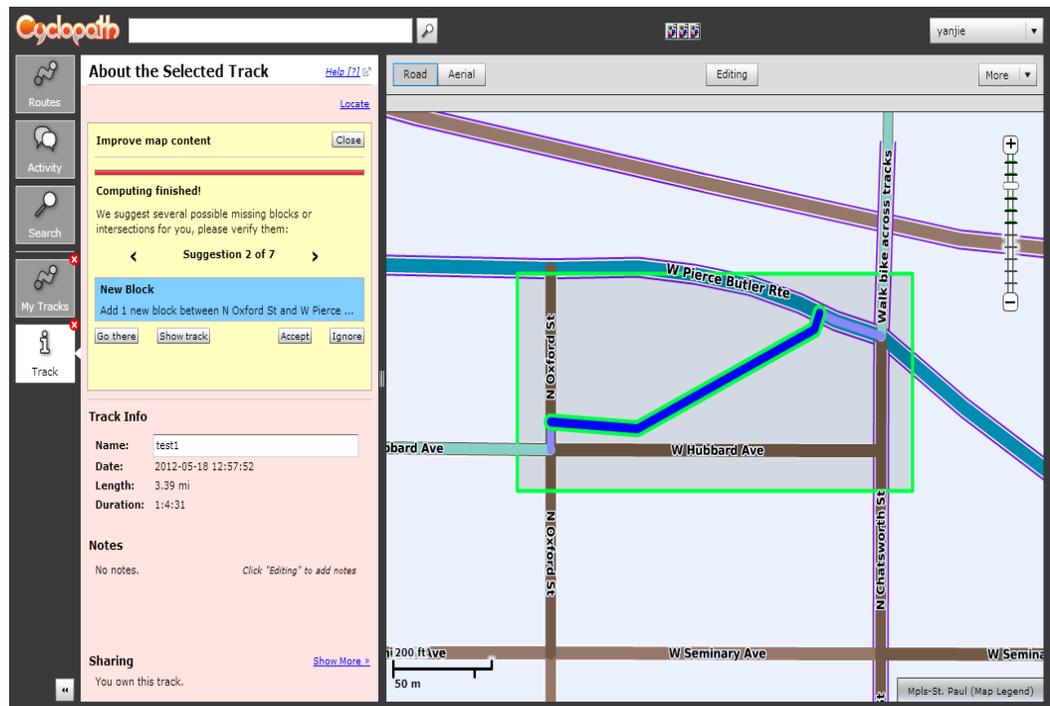


Figure 4.12: Highlighting selected suggestion

When accepting one suggestion, we want to make sure the user is looking at it in the current viewport and does not make any arbitrary decisions. Thus, we require the rectangle to be within the current viewport when the user accepts the suggestion. If this prerequisite is not met, the user will get a warning message.

In order to reduce the possibility of making mistakes or getting error message, we do not let the user do any operations on the interface when the system is automatically processing the editing operations after the user accepts a suggestion. During this process, it usually takes about 5-6 seconds to finish all the map editing operations (including adding new blocks, creating new intersections).

It is important to provide users a good mapping between the suggestions on the left

panel and their corresponding locations on the map where the suggestions refer to. Therefore, we allow users to highlight each suggestion area (green rectangle) on the map when switching to suggestions on the left panel (as shown in Figure 4.12). When clicking on a specific new block or new intersection within a suggestion list, the selected object is also highlighted in green outline. Furthermore, “Go there” guides users to the specific location referred to by the suggestion.

Chapter 5

Evaluation

This chapter describes an evaluation of the conflation user interface. We did user testing of the interface in the lab to investigate user acceptance and usability and determine the success or failure of the design and implementation. We also collected some feedback regarding the advantages and disadvantages of the design of the interface to improve the design in the future.

5.1 Research Questions

Generally speaking, only a few mapping applications have functionality that helps users discover and fix incomplete map data; thus little research regarding this kind of interactive technique is available. This evaluation tried to examine the design in the target context of use to better understand its user acceptance and usability with the intention to support the improvement of the conflation feature. A user experiment was conducted to examine the following two specific research questions:

- Is this interactive technique acceptable and usable for Cyclopath users?
- Which aspects of the interface were either positively or negatively perceived by the users and therefore might be subject to future improvement or extension?

5.2 Methods

In order to gain in-depth knowledge about user acceptance and usability of the conflation user interface in helping them better rectify the map, the evaluation consisted of two stages with different methods: face-to-face interviews, and a remote user study.

In the first stage, we conducted eight user interviews. These were semi-structured, one-on-one interviews with the objective of evaluating the interface qualitatively. Each session lasted approximately one hour. The participants included both core and non-contributors of Cyclopath, and both Cyclopath Mobile and non-Cyclopath Mobile users, as we tried to get as diverse a set of users' real-life scenarios in as possible. Details of the participants are shown in Table 5.1 (The number of revisions demonstrates how many edits the user has made. The number of activities on Cyclopath Mobile reflects to what extent that a user uses our mobile app. Larger number means that the user is an experienced user on Cyclopath Mobile.).

To solicit volunteers for the study, we emailed 350 Cyclopath members, chosen randomly from those who had logged in during the past year, and who had given us permission to send them emails for research purposes. In this email, we introduced the general process of the user study, in which participants needed to record at least one track using our test version of the Cyclopath Mobile app before coming to our interview. 20 users replied to the invitation email, of whom 12 consented to participate and 8 of them were finally included in the study. The other 8 users also had interest in participating, but were not able to participate because they did not have Android phones. Each participant was given \$10 as incentive for participation.

Participants	Revisions	Activities on Cyclopath Mobile	Tracks Recorded on Cyclopath Mobile
Subject 1	25	390	0
Subject 2	2	2246	0
Subject 3	245	3890	0
Subject 4	129	40	0
Subject 5	56	8876	4
Subject 6	1	0	0
Subject 7	4	8854	0
Subject 8	47	0	0

Table 5.1: Interview Participants

The user interviews were one-on-one interviews with a moderator, an interviewee, and a note taker in the lab. Each interview consisted of a set of interview questions and tasks. We began with some ice-breaker questions about their latest cycling experience, followed by questions related to riding behavior, familiarity of roads and trails in the Twin Cities, their experiences recording rides on mobile phones or GPS devices, and their use of the Cyclopath web client and mobile app. Afterwards, we asked them to show us what they usually do on the Cyclopath web client during a regular visit, with particular attention to their familiarity with using the map editing tools.

This set of initial interview questions was followed by the tasks portion of the study, where we asked participants to: (1) choose one of their own tracks (if any) and start the conflation task with it; (2) verify each suggestion provided by the system during the

conflation task. If they did not have any tracks in their account, users were asked to use our test track and follow steps (1) and (2). Before the conflation task started, we would introduce a little bit about how the activity worked. During the task session, we asked participants to speak out (think aloud) what they were thinking when they performed each operation on Cyclopath.

After completing the task, users were asked a set of follow-up questions about their experience using the conflation feature, motivations for recording tracks on Cyclopath after knowing about this new feature, and what they liked or disliked about it. Aside from giving users a chance to express usability concerns about the interface, we asked questions about potential benefits they gained from using the conflation feature. During the interviews, in addition to notes taken by the researchers, audio was also recorded. Additionally, during the user tasks, we recorded the user's computer screen. These records, together with our observations, gave us a rich amount of data to work with and sort through after the evaluations. A copy of the interview script is provided in the appendix. Please note that since the interviews were semi-structured, all the interviews were slightly different, although with the same focus.

The objective of the first stage was to obtain concrete feedback regarding the interface in a real-world scenario by using users' own tracks, and get their actual feeling by face-to-face conversation and our observations. But the results gained from first stage were not sufficient to measure to what extent this interface is usable because the tasks were not the same for all participants. In the second stage, we let all the users run the conflation task with the same test track, generating the same set of suggestions for every

participant. The objective was to help us know whether this interface is able to help users understand the suggestions from the system and make the right decisions, as well as what difficulties they have in further editing the map. We mailed the eight participants involved in the first stage and a few other users who were interested in participating in the interviews in the first stage but were not able to participate because they do not have an Android mobile phone. We sent them the link and instructions for participating in the second round of the user study, where users were asked to run the conflation task with our test track from their own computers without coming to our lab. We modified the interface a little bit to support obtaining users' decisions towards each suggestion and their comments that explained the specific reasons behind their decisions, as well as what difficulties they had in further editing. The users could submit their feedback on Cyclopath immediately after accepting or rejecting each of the suggestions, while their thoughts about the suggestions were still fresh in their minds. In the end, eight users, including five from first stage interviews and three new participants, were involved in the second stage of the study.

5.3 Analysis and Results

The two major contributions of the conflation algorithm are detecting missing blocks and creating them as necessary. So our conflation interface should be able to support the two primary functions of the algorithm -- presenting the map update suggestions in an easy to understand way and simplifying the process of editing the map

after the user has accepted the suggestion. In this evaluation, we explored the usability of our interface in helping users uncover and fix map problems in two aspects – suggestion presentation and map editing.

5.3.1 Suggestion Presentation

In order to evaluate the usability of our interface in helping users understand the suggestions the system provides and make the right decisions, we collected feedback on whether users accepted or rejected each suggestion as well as the reasons behind their decisions. We tested this by asking participants to run the conflation task with our test track, for which there were seven suggestions for edits. Three of the suggestions were correct, three of them were wrong, and the other one was doubtful since the new block in that suggestion went through a parking lot, for which different people might have different opinions. In total, we collected 54 instances of feedback about the 7 suggestions from 8 participants.

To test whether the suggestion presentation in our interface was able to help users make the right decisions, we counted the number of correct and incorrect decisions. We excluded 8 from 54 feedback comments when calculating the accuracy of decisions made by participants, because they were all cases where there was no right or wrong answer. Overall, the 46 instances of feedback gathered from the 8 participants in this study show that about 85 % of the decisions from our participants were correct. Of the 7 wrong decisions, 6 were false negatives, which means there should be a new block but

they rejected to add a block there. The most common reason was that people thought a possibly private path should not be on a public map.

“It appears that the path taken is a shortcut across private property and should not be reflected in a public map.”

For the only false positive decision made by one of the participants, there was a GPS tracking error in that section of the track. The suggestion was obviously wrong because the second half part of the suggested new blocks went through someone’s house. So overall, 85% correct decisions demonstrated that our interface was effective in helping users make the right decisions.

When verifying each suggestion, users tend to validate its correctness based on their local knowledge and satellite view. If there is a real path under or close to the new block we are suggesting, then they will want to accept the suggestion. If the suggested new block passes through any building or other kind of barrier, then they will ignore the suggestion without any doubts.

In addition, we also paid special attention to how users reacted to the grouping technique of relevant suggestions on the interface. Two participants mentioned that it is reasonable to group relevant suggestions together, thus simplifying what a user needs to do. However, they also said it should allow users to take different actions with each new block as well.

“I am sure there was a reason for showing two blocks in one suggestion. But each one ought to be separately choosable in terms of action (accept or reject).”

For the text presentation of suggestions in the left panel, it was in line with our expectations that users were only thinking about new blocks and did not care about the modified blocks which were associated with the new blocks. No one questioned why modified blocks are not showed in the suggestion list, and they could understand the suggestions very well and make the right decisions. So our simplification of text suggestions is useful in making suggestions easier to understand.

Compared to Google Maps, Cyclopath lacks lots of map data in parking lots. Thus, in this evaluation, we also tested whether users would like to add roads for parking lots or not. Unlike motor vehicles, cyclists are able to route through parking lots. There was only one participant that accepted to add new blocks for the parking lot in our suggestions. Our experiment results therefore suggest that people tend not to add roads in a parking lot because it is routable for cyclists everywhere in it.

We also gained some additional valuable feedback about how the interface could be improved from our participants:

- **Routes cannot be hidden.** Although we allow users to hide the track to see the route (especially the new block part), it is still not satisfactory because they are not able to hide the route in order to see what is under the suggested new block on the aerial photo when verifying each suggestion.

“I would like to have been able to hide the road part of the map and just see the aerial. The road part of the map including the suggested new segment obscures the aerial photo.”

In our current design, considering this issue, we have already made the new block and modified block thinner than the normal part of the route, but it appears to not be as ideal as we imagined. Thus in the future design, we are going to refine this part. The simplest solution might be enabling routes to be hidden.

- **No undo operation.** Participants were also interested in having the ability to undo and redo a selection, regardless of whether the selection was “accept” or “ignore”. The system should allow users to revert the map back to its original state if they change their mind about the selection.
- **No middle option between accept and ignore.** In our initial design in the very beginning, we had two options for users for each suggestion: 1) accept and change, 2) ignore. In order to simplify the description of the options, we made them into “accept” and “ignore” in our current design. We did this because we believed users are aware that the map is editable anytime and that they are allowed to make any modifications they wish after the system finishes all the necessary operations when accepting a suggestion. Thus the middle option didn’t seem to be necessary. But two of the participants mentioned that it would be nice to have a middle option (accept and adjust) to let users know that they are able to adjust the new block after it is created. In the current design, it’s not obvious that they are able to do so. To improve the design of this part, instead of adding a middle option, we should have a good prompt to inform users that they can adjust new blocks after they are created automatically.

- **Incomplete display of suggestions.** Users need to move the cursor over the suggestion in order to view the complete content (as shown in Figure 5.1), which is not convenient. We could fix this problem by showing the detailed description of each suggestion in two lines in the future.

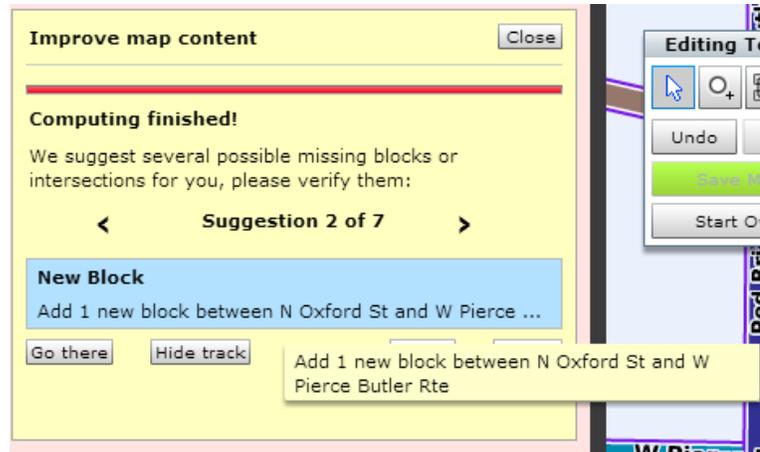


Figure 5.1: Suggestion with its tooltip

5.3.2 Map Editing

With the conflation feature, the system is able to automatically make all the necessary operations in editing the map as the suggestion indicates to users. We wanted to investigate the effectiveness of the interface for map editing, as well as what challenges users had in further editing the map to make it conform to the reality after they accepted a suggestion.

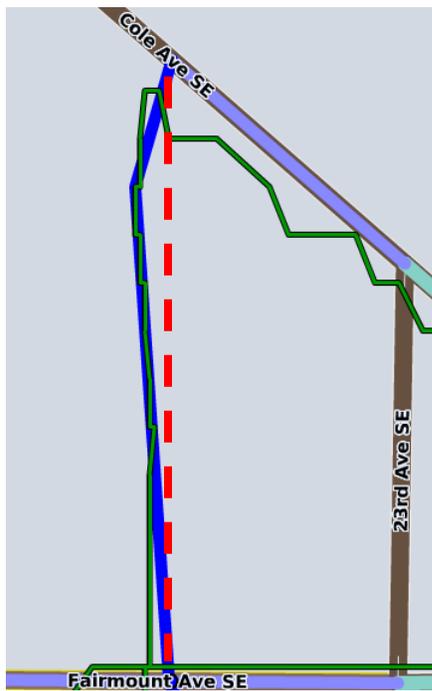
Participants believed the conflation interface provides them a much simpler way of editing the map, by saving the complicated intermediate editing steps, even though there are still a few challenges at different levels for them in further editing the map.

In order to know better of what the most common challenges users have when editing new blocks after the system automatically creates them, we first summarize what necessary editing operations users need to do. Comparing the new blocks created by the system automatically with the real blocks, we find four cases. Case 2-4 are shown in Figure 5.2-5.4. The dark blue lines are new blocks suggested by the system. The red dashed lines represent the actual missing paths. The green lines are the GPS tracks:

- (a) The geometry and the position of the new block are both the same as what they should be in reality. There is no doubt that users want to accept to add the new block and there is no need for them to do any further editing in this case. However, this seems to seldom happen because there are always some bias errors with GPS observations.
- (b) The new block is at the right position (if both of the two end nodes of a block are at the right position, then we define it as being located at the right position), while its geometry is not correct (as shown in figure 5.2)
- (c) The geometry of the new block is correct, while the position is not exactly the same as where it should be (as shown in Figure 5.3)
- (d) Neither the geometry nor the position of the new block are the same as in reality, but they fit somewhat. (As shown in Figure 5.4, for example, the new block should be moved downward a little and the geometry should be as the red dashed line indicates.) This case includes all of the situations of the previous two cases, so we only discuss the second and third case in detail below.

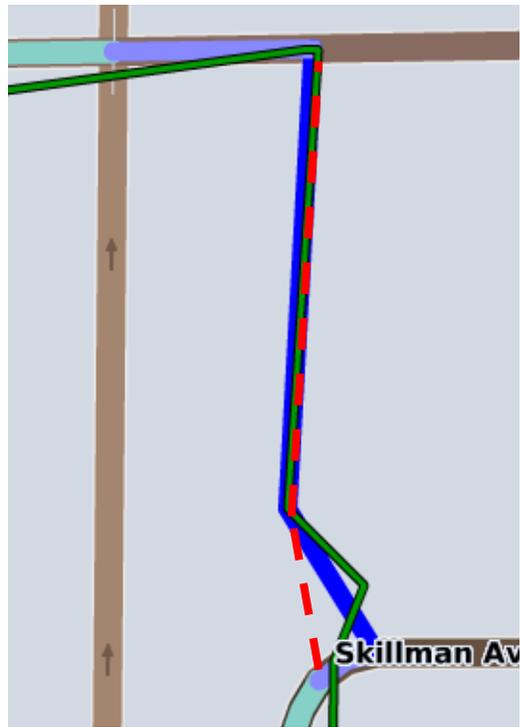


Case 2-a: The new block should be a curve road segment instead of a straight line in this case

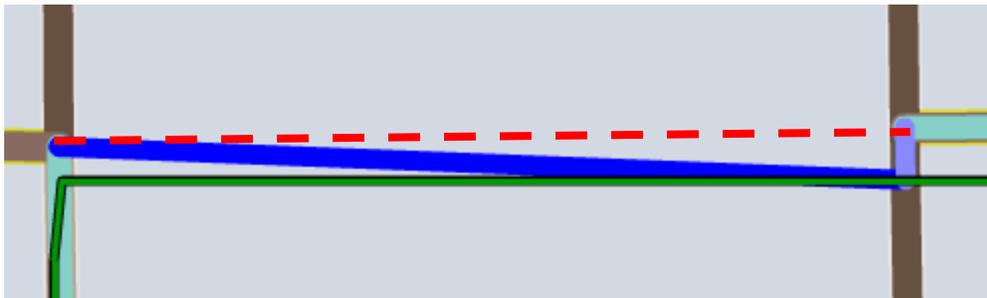


Case 2-b: The new block should be a straight line instead of a broken line.

Figure 5.2: Two sub cases that a new block is at the correct position but its geometry is not correct



Case 3-a: The intersection connecting the new block with Skillman Ave should be skewed to left a bit in this case.



Case 3-b: One of the end nodes of the new block should be connected to an existing intersection.

Figure 5.3: Two sub cases that the geometry of the new block is the same as in reality, while its position is not exactly the same as where it should be.

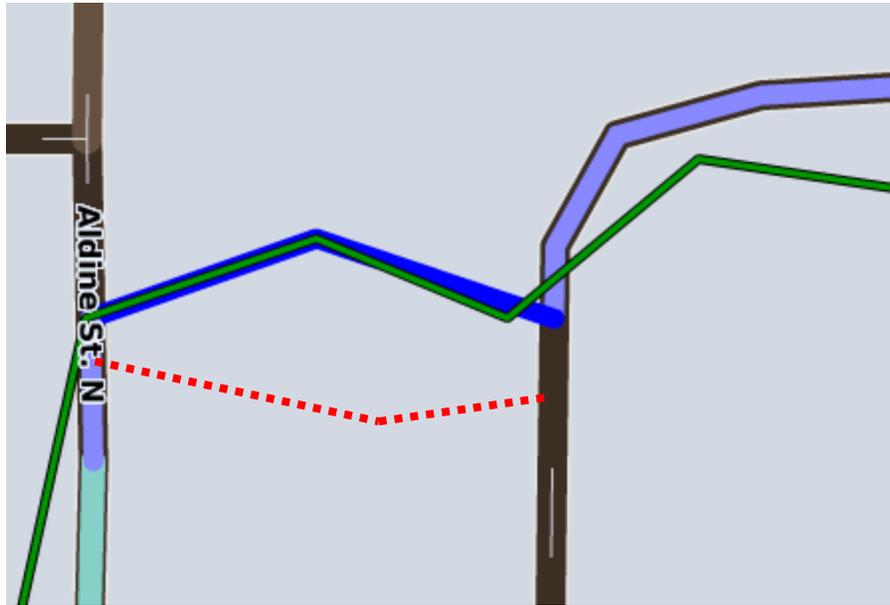


Figure 5.4: Neither the geometry nor the position of the new block is exactly the same as in reality.

Regarding the first case, users do not need to do anything after they accept to add the new block, while for the latter three cases, they need to do some kind of editing -- either modify its geometry or change its position or both. We summarized all the necessary operations users need to do in map editing to rectify what the system has suggested. In case 2-a, users need to add vertexes to the new block and drag it to make it a curve or a polygonal line. In case 2-b, users may have to drag the block to make it straight or delete extra vertexes. For case 3-a, dragging is the only required operation. In case 3-b, the system suggested a wrong intersection which should be removed. Users need to first move the ends of the new block to connect it with an existing intersection and then merge the two blocks that were just split by the new intersection into one block.

Case	Difficulty
2-a	Normal
2-b	Easy
3-a	Easy
3-b	Hard

Table 5.2: Difficulties of map editing in the four cases

Based on our observations, we analyzed and summarized the difficulty levels of all the cases (as shown in Table 5.2). Users who were experienced in editing the map on Cyclopath could generally use editing tools easily to drag the new block to the right position and change its geometry to make it fit the reality even if we did not give them any instructions. Meanwhile, users who hadn't edited the map before or had only done so a few times could quickly find a way to drag the block. However, they would need some help from the researchers to complete editing if they wanted to delete extra vertexes of a new block, which is done using a tool that is not easily discoverable (pressing the delete button on the keyboard). Case 3-b is a difficult task for both novice users and experienced users. If users want to accept the suggestion and then drag the new block to connect it with the existing intersection, they will need to uncheck the "Enable sticky intersections" option in the settings panel first (used to allow dragging of an intersection as a whole), and then drag the new block to connect it to the right intersection. We found that most users had no idea how to do this, and some of them just left two intersections overlapping one another. In the future, we would like to add some prompts and instructions to assist users to complete these editing tasks where they have

difficulties. We can see that one way to improve this design is to add some descriptions of the editing operations in the user's language, such as "make it a curve", "make it straight", or "connect it with another intersection", and then give them some simple instructions of the concrete editing steps.

5.3.3 Other Findings

In addition to the two major aspects we centered on, we also gained some other valuable feedback and interesting findings on both the interface and the algorithm. Although the interface, together with the conflation algorithms, revealed the effectiveness of helping users discover missing map data and create them as necessary, fixing GPS errors is a great challenge that we need to overcome to provide better suggestions. The quality of GPS data can be affected by the environment in which the ride occurs. The locations that the mobile devices detect can be significantly displaced in areas where many tall buildings exist. There are mainly two types of errors in GPS traces: missing points (due to losing GPS signal) and displaced points (due to GPS error). We can see two ways of addressing these issues: a) the system automatically detects outliers by analyzing the speed trend – Currently, the algorithm already does this, but the threshold is a bit high in order to include bus rides. One extension to this would be to detect whether the user is riding a bike or the bus. Another approach is to detect if an increase of speed only happens for a very short amount of time, b) allow users to edit their track before starting the conflation task. They could delete parts of the track and then draw on the map the paths they actually rode, or drop some pins on the map, and

then the system connects those points to construct several line segments and build a complete track. There was also one participant that mentioned that this algorithm is only able to detect missing blocks, but does not work for removing a block which should not be on the map. In short, these ideas give us implications in thinking of and improving our current algorithm and design of the interface.

There were three participants who wanted to add some information (name, road type, bikeability) for new blocks, which suggests it would be better to have a prompt to ask users to do so. We could consider automatically opening the panel for new blocks to encourage users to enter the information.

When asked about what results they expect to see after they accept or ignore a suggestion, most users thought it was matching their expectations –new blocks are created after accepting a suggestion and suggestions are removed from the map after ignoring them. Two of the participants thought they would also like to see the new blocks become part of the route after they are created automatically by the system. But we believe that editing the newly added blocks (including geometry, topology, and some information, such as name, bikeability, etc.) is quite necessary. If the route is on top of the new block, it is hard to see clearly what is under the route.

Another question which we are interested in is what users want to do with the route if they can get a complete route from their ride after completing the conflation task. It appears that they do not have a strong desire to save the route to their own route library or make it public to other Cyclopath users. Some of them would like to save it as their private data. By matching tracks to routes, users could more easily compare multiple

rides of the same route if we could help them track their cycling times. This will be a very useful support for user's personal training. The finding above also indicates that even though users have privacy concerns that prevent them actively making contributions, our interface can still effectively break through these barriers to get them to make public edits to the map. It is always a good thing for us if we can get users to do tasks that benefit the community eventually while doing tasks that they believe mostly benefit themselves.

To summarize, this conflation interactive interface is acceptable and usable for our target users. We proved it is able to help users find more map problems that need to be fixed, and help users understand the suggestions the system provides and make the right decisions. It is also able to help users edit the map in a much simpler way. The conflation feature elicits work to be done by users and provides them with a convenient way for contributing content to Cyclopath.

Chapter 6

Conclusions and Future Work

In this work, we designed and implemented a user interface that helps users uncover and fix missing road segments by taking advantage of GPS tracks on Cyclopath. This interface lets users step through a set of map update suggestions computed by a map matching algorithm and automatically updates the map as necessary based on users' decisions. It was evaluated and verified to be able to help users understand the suggestions the system proposes and to assist them to make the right decisions about the suggestions. Our design for simplifying map editing is favorable for not only users who are experienced in map editing but also inexperienced. Users generally do not have too much difficulties and challenges in further editing the map after the system's automatic editing operations. But in some cases when the editing task is cumbersome or the way to do it is not obvious, they will fail to achieve the goal. We also gained some valuable feedback from the users with regards to some user interface problems, which could be directions for future work.

However, our work also has some limitations. Due to some problems with users' mobile phones and our mobile app, not all the participants were able to save their own track successfully for our interviews. Instead, they evaluated the interface with our test track. Thus their scenarios of use when evaluating our interface were not exactly the same as in their actual use, given that they were not necessarily familiar with the area

around the track. Additionally, the participant sample size was not quite ideal for any quantitative inquiries for the restrictions in resources and time.

While the results provided answers to the research questions in this thesis, it also poses some valuable open questions to be explored for future work. The work in this thesis is the first step in studying what potential benefits we can obtain from the conflation technique, in which we designed and built a usable interface for the conflation feature. In the future, we plan to release this feature on Cyclopath for public use and trace its usage within a certain period of time. Then further user studies, could be conducted to explore more interesting research questions such as how to motivate people to use the conflation feature, what types of information they tend to input for the blocks and points along their tracks, and potential user behavior evolution through time.

Bibliography

- [1] D. P. Anderson. BOINC: A system for public-resource computing and storage. In *Proceedings of the Workshop on Grid Computing*, pages 4–10, 2004.
- [2] J. Biagioni and J. Eriksson. Map inference in the face of noise and disparity. In *Proceedings 20th ACM SIGSPATIAL*, pages 79–88. ACM, 2012.
- [3] D. Cosley et al. Using intelligent task routing and contribution review to help communities build artifacts of lasting value. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 1037-1046. ACM, 2006.
- [4] D. Cosley et al. SuggestBot: Using intelligent task routing to help people find work in wikipedia. In *Proceedings of IUI*, pages 32–41, 2007.
- [5] L. Festinger. A Theory of Social Comparison Processes. *Human Relations*, 7, 2 (1954), pages 117–140.
- [6] B. Frey et al. Social comparisons and pro-social behavior: Testing “conditional cooperation” in a field experiment. *American Economic Review*, December 2004, 94, 1717–1722
- [7] F. Harper et al. Social comparisons to motivate contributions to an online community. In *Persuasive Technology*, pages 148–159, 2007.
- [8] L. Liao et al. Building Personal Maps from GPS Data. In *proceedings of IJCAI MOO05*, pages 249-265, 2005.
- [9] K. Ling et al. Using social psychology to motivate contributions to online communities. *Journal of Computer-Mediated Communication*, 10, 4 (2005), art. 10.
- [10] E. A. Locke, and G. P. Latham. Building a practically useful theory of goal setting and task motivation: A 35-year odyssey. *American Psychologist*, 57: 705–717, 2002.
- [11] N. Marmasse, and C. Schmandt. A User-Centered Location Model. *Personal and Ubiquitous Computing*, pages 318-321, 2002.
- [12] A. Mashhadi et al. On the accuracy of urban crowd-sourcing for maintaining large-scale geospatial databases. In *Proceedings of the 8th International Symposium on Wikis and Open Collaboration*. ACM, 2012.
- [13] Mikhil Masli, Reid Priedhorsky, and Loren Terveen. Task specialization in social production communities: The case of geographic volunteer work. *Proc. ICWSM*, 2011.
- [14] P. Mooney, et al. How social is OpenStreetMap? In *Proceedings of AGILE 2012*.
- [15] Katherine Panciera, Reid Priedhorsky, Thomas Erickson, and Loren Terveen. Lurking? cyclopaths?: a quantitative lifecycle analysis of user behavior in a geowiki. In *Proc. CHI*, 2010.
- [16] K. Panciera, M. Masli, and L. Terveen. ”How should I go from to without getting killed?”: motivation and benefits in open collaboration. In *Proceedings of the 7th International Symposium on Wikis and Open Collaboration, WikiSym '11*, pages 183–192. ACM, 2011.

- [17] R. Priedhorsky et al. The computational geowiki: What, why, and how. In *Proceedings of the 2008 ACM conference on Computer supported cooperative work*, pages 267–276. ACM, 2008.
- [18] Reid Priedhorsky, Mikhail 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.
- [19] Reid Priedhorsky, David Pitchford, Shilad Sen, and Loren Terveen. Recommending routes in the context of bicycling: algorithms, evaluation, and the value of personalization. In *Proc. CSCW*, 2012.
- [20] Y. Takeuchi et al. CityVoyager: An Outdoor Recommendation System Based on User Location History. In *Proceedings of UIC'2006*, pages 625-636, 2006.
- [21] F. Torre, D. Pitchford, P. Brown, and L. Terveen. Matching GPS traces to (possibly) incomplete map data: bridging map building and map matching. In *Proceedings of ACM SIGSPATIAL GIS 2012*, pages 546–549. ACM, 2012.
- [22] Fernando Torre, S Andrew Sheppard, and Loren Terveen. bumpy, caution with merging: an exploration of tagging in a geowiki. In *Proc. GROUP*, 2010.
- [23] Y. Wang, X. Liu, H. Wei, G. Forman, C. Chen, and Y. Zhu. CrowdAtlas: Self-Updating Maps for Cloud and Personal Use. In *Proceedings of the 11st International Conference on Mobile systems, Applications and Services*, pages 469–470, 2013.
- [24] W. Zheng, Y. Zheng, X. Xie. and Q. Yang. Collaborative location and activity Recommendations with GPS History Data. In *Proceedings of the 19th International Conference on World Wide Web*, pages 791-800. ACM, 2010.

Appendix

Cyclopath Interview Script

This script gives general structure, focuses and question style of the interview. Since this is a semi-structured interview, we also asked questions that are not on this list as follow-up questions in our interview

Cycling

1. Can you tell me something about your latest ride? (Purpose? Weekdays or weekends?)
How long have you been riding in the Twin Cities?
What are your regular purposes of riding?
Are there some rides you do only during the week or on weekends?
How well do you feel you know the local trails and roads?
2. Have you ever use any tools to record a ride (Cyclopath, My Tracks...)
What types of tools did you use?
Why did you do this?
What did you do with your track?
Do you do this sort of thing often?

Cyclopath

1. Do you remember when was your first time using Cyclopath/Cyclopath Mobile?
How did you know Cyclopath/Cyclopath Mobile?
How often do you use Cyclopath/Cyclopath Mobile?
What are your regular purposes of using Cyclopath/Cyclopath Mobile?
2. Please log into Cyclopath and show us what you do on a regular visit.
What features on Cyclopath do you like most? Why? How do you usually use these features? (In what situation? Before, during or after cycling?)
Have you ever edited some information to the map?
 - If yes, please show me how you usually do this. How do you usually find a place that needs changes? What are some challenges you've faced while editing Cyclopath?
 - If no, why?

Tasks

1. Choose a track you have on Cyclopath and take a look at the information of the track. Please tell me about that experience:
How did you feel about the process when you recorded it on Mobile? Is there any information you need but is not provided?
2. Start the conflation task and tell me what you think when you perform each step (think-aloud):
Instructions: Now I would like to have you do an activity with the track. In this activity, you'll know how your actual track matches to the data on the map. Parts of your track will match to the roads on the map perfectly, while some will not. For example, you may ride on a trail that does not exist on our map, and then our system will suggest some new roads.

If the subject is not active enough to tell me his/her thoughts, the following questions will be probably asked:

- How do you understand each suggestion? (New block, new intersection, modified block)
- What is your expecting result of each action? (Accept, ignore)
- How do you think the one-click action to perform multiple editing operations?
- Please tell me the reason why you want to accept/ ignore in each step.
- Would you like to make the route complete and save it to your own library/share it with others? Why?
- After you complete the task, can you think of any reason you want to track your ride on Cyclopath?
- What do you think about the whole process of the task? Is there something you like/dislike? Why?
- Is there any other improvement you think we could make?