POLITECNICO DI MILANO
Corso di Laurea Magistrale in Ingegneria Informatica
Dipartimento di Elettronica, Informazione e Bioingegneria

TransitTrace: Facilitating Urban Wayfinding via an Ambient Display Visualization

Tesi di:

Massimo De Marchi
matricola 816512

Relatore (Politecnico di Milano):
Pier Luca Lanzi
Co-relatore (University of Illinois at Chicago):
Angus Graeme Forbes

A.A. 2014/2015

To my family,

my friends,

and the memory of my dear friend Diego.

ACKNOWLEDGMENTS

I want to express my gratitude to my advisor Angus Forbes. Thank you for finding me the opportunity to work on this project, and for stimulating my passion for Visualization.
I would like to thank Professor Jakob Eriksson for giving to me the opportunity to work on this project, for proposing useful ideas and providing excellent feedbacks.
I would like to thank Professor Pier Luca Lanzi for helping and inspiring during this master's degree.
Special thanks to Paul Murray, Giorgio Conte and Kyle Reese-Brogi Almryde for helping me to revise this writing. Special thanks also to Lynn Thomas for helping me and assisting me whenever the need.
Last but most importantly, thanks to my family, who gave me the opportunity to be here. Thanks Papà, Mamma and Alessandro for always being my greatest supporters.

MDM

PREFACE

This thesis is the result of a part of the research activity that has been conducted for the ”CybeR-Enabled Demand-Interactive Transit” project, funded by the National Science Foundation and carried out by University of Illinois at Chicago and Northwestern University. The project aims at creating theories and tools to characterize, analyze, operationalize, and evaluate a hybrid system that integrates conventional fixed-route service with demand-interactive service. The broader impacts of the project are related to the promotion of transit for more efficient, equitable and sustainable transportation.

My personal research centers on improving the availability of transit information to passengers. The goal of this thesis is to study, design and develop a new visualization tool that improves the ease of use of the Public Transport System, and promotes the Public Transport System itself.

TABLE OF CONTENTS

TABLE OF CONTENTS (continued)

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

PTS	Public Transport System
GTFS	General Transit Feed Specification
CTA	Chicago Transit Authority
JSON	JavaScript Object Notation
CPU	Central Processing Unit
GPU	Graphics Processing Unit
REST	Representational State Transfer

SUMMARY

People every day rely on Public Transport Systems for their daily commutes or for their journeys through new cities. The availability and usability of information that are relevant to the traveller is very important to enable wayfinding. Many visualizations exist that support wayfinding, and these visualizations make a transport system easier to use. However, due to the high complexity of the transport networks and the needs of different travelers, designing visualizations that effectively support wayfinding is a challenging task.

This thesis presents TransitTrace, an interaction-free visualization designed for ambient displays that enables casual and highly effective exploration of the transit network. The visualization shows current and future movements of transit vehicles in the vicinity of the user, as well as future transfers to other lines. The idioms employed for the visual design allow the user to focus on routes that are relevant to her, even when a high number of routes are displayed. In this way, the visualization can serve multiple users with different needs at the same time.

I show how partial trajectory drawing can be used to show connections to transfer lines and how this technique limits the amount of visual clutter. I also explain the usage of an animation that effectively enables route planning. Furthermore, I discuss the complexities of dealing with transit network data, and I describe how the system has been designed to address the various issues.

In conclusion, I report a case study to explain how the visualization is used to plan routes, and I discuss limitation of the approach and future work.

AMPIO ESTRATTO

Questa tesi presenta il risultato di una parte dell'attivitá di ricerca che é stata condotta per il progetto "Cyber-Enabled Demand-Interactive Transit", finanziato dalla National Science Foundation e realizzata in collaborazione da University of Illinois at Chicago e Northwestern University.

Ogni giorno milioni di persone utilizzano i sistemi di trasporto pubblico per raggiungere le loro destinazioni. Le autoritá del trasporto fanno ampio uso di visualizzazioni per rendere i mezzi pubblici piú accessibili e piú semplici da usare. Al giorno d'oggi le informazioni vengono presentate all'utente prevalentemente tramite mezzi di comunicazione passivi, ovvero dispositivi privi di interazione, come ad esempio ambient display che si possono trovare nelle vicinanze di una stazione oppure nelle vicinanze di luoghi di interesse quali centri commerciali. Questi display vengono utilizzati per fornire agli utenti informazioni utili come ad esempio la disponibilitá di mezzi e la stima del loro orario di arrivo ad un dato stop. Tuttavia, a causa della complessitá delle reti di trasporto e alla mancanza di interazione, le informazioni fornite da questi dispositivi sono piuttosto limitate. Questa tesi presenta *TransitTrace*, una visualizzazione che sfrutta ambient display passivi per fornire ai viaggiatori informazioni piú dettagliate, e per aiutare i viaggiatori ad esplorare e conoscere un sistema di trasporto pubblico. Prima di tutto, vengono analizzate le difficoltá nella progettazione di una rappresentazione visiva che supporti funzionalitá complesse, quali la pianificazione di un percorso, quando le circostanze escludono le solite forme di interazione che permettono la semplificazione di tali funzionalitá. Viene poi proposta una soluzione che combina accuratamente la rappresentazione visiva con un' animazione al fine di massimizzare le opportunitá di eseguire attivitá di pianificazione del percorso attraverso un processo overview+detail passivo. Piú nel dettaglio, la visualizzazione mostra le posizioni attuali dei veicoli nelle vicinanze dell' ambient display, e attraverso un' animazione ne simula gli spostamenti nell'immediato futuro. La visualizzazione consente all'utente di concentrarsi sui mezzi che sono rilevanti al fine di raggiungere la destinazione desiderata, anche nel caso in cui siano visualizzati un grande numero di mezzi; in questo modo l'ambient display puó essere utilizzato contemporaneamente da piú utenti con diverse esigenze. Viene infine descritto un caso di studio che dimostra come la visualizzazione puó essere utilizzata per pianificare percorsi.

CHAPTER 1

INTRODUCTION

Public Transport Systems (hereafter, PTS) are one of the most important infrastructures of a city. To improve the accessibility and usability of PTS is one of the major concerns of urban planners and city administrations because such improvements can lead to economical, environmental and social benefits. An effective usage of PTS in fact, results in more sustainable cities and ensure that citizens are able to move.

Every day millions of people use PTS for their daily commutes, or to travel to new cities. These systems are inherently complex and travelers make use of different navigation tools to help them to plan their trips. A study conducted by the Israeli Ministry of Transport identified system legibility as one of the greatest barriers to use the PTS (1) and this highlight that there is room for visualization researchers to help solve PTS problems. In fact, visualization allows people to offload internal cognition and memory usage to the perceptual system where a significant amount of information can be processed in parallel at the preconscious level (2). This enables humans to process and understand more complex data such as public transport networks.

Many visualizations already exist that improve PTS accessibility, such as Beck's London tube map (3) which is widely adopted by most PTS throughout the world. Very common are also static timetables and live updated displays that show estimated times of arrival of incoming trips at a station. One of the most used navigation tools is Google Maps (4). It allows travelers to plan their journey from an origin to a destination using three possible route selection parameters, that are best route, fewer transfers or less walking. Many transport authorities provide also tools to visualize vehicles positions on a map (5) or trip planner web applications (6). The limits of these tools is that they are only able to show a small amount of information and they limit the possibilities that the user has. These tools also limit the knowledge that the users have of the transport system and thus they are responsible of how the system is used and perceived.

I present a novel visualization technique that combines animation and trajectory depiction to enable multiple user to plan their routes without interaction. The project has been inspired by the CTA Web-based slideshow named DIY Transit Info Display (7). This initiative encourages owners of commercial services to feature displays the makes helpful information for the customers. We can imagine these non-interactive ambient displays placed anywhere in the city such as inside malls, museums, universities and workplaces so that travelers can pass by them and get useful routing information or they can stop and explore the system. The visualization is not interactive due to the nature of the target display and the fact that multiple users should be able to use it at the same time. Therefore, enabling each user to effectively plan routes is even more challenging. The absence of interactivity lead to an assumption that defines what information should be visualized: the user is interested in catching transports that pass within the immediate vicinity of the ambient display location during the time she is watching the visualization. Another implication of the absence of interaction is that the final destination of each user is not known. Therefore we should visualize routes to all the reachable destinations. It is possible, of course, to take advantage of various form of interaction, such as touch screens or integration with mobile devices, whenever these features are available, but the exploration of these options is beyond the scope of this thesis.

The aim of this thesis is to develop a visualization that enables travelers to plan route and to explore the possibilities that PTS offers. Therefore the technique that I propose let the user freely plan her journey based on the availability of nearby rides and shows all the possible alternative that she has to get to a certain destination.

For the sake of showcasing the visualization, it has been contextualized in the city of Chicago. The PTS of Chicago is held by CTA, and it currently features 126 bus routes and 8 trains routes, making it one of the most complex PTS world-wide. The Chicago PTS transit feed data is freely available at the CTA web portal (8).

CHAPTER 2

PREVIOUS WORK

In this section some previous works are discussed. Many visualizations researches can be applied to the domain of PTS, since to enable route planning several different aspects of the PTS can be visualized. For example, the user might want to see direction of vehicles, estimated time of arrival and reachable destinations among others. In this discussion I group some of the techniques suitable to visualize these pieces of information into three main categories: Isochrone Maps, Schematic Maps, and Visual Analytics of Movement.

2.1 Isochrone Maps

Isochrone maps (9) are a traditional visual representations used in transportation to show regions of equal travel time from a given departure location. Isochrone maps are typically visually encoded using colored contours and can be easily overlaid on geographical maps. Widener et al.(10) and O'Sullivan et al. (11) used this technique to represent accessibility information at a coarse resolution. In fact, the estimation of precise travel time is difficult to calculate, yet it can be very useful for a traveler in understanding how much time it takes to reach a specific destination. Isochrone maps cannot reveal other mobility factors and therefore are often used in combination with other techniques. For details see Zeng et al (12).

2.2 Schematic Maps

Schematic maps are the outcome of a process of simplification and/or distortion. Beck, a pioneer in this field, proposed a redesigned map of the London underground which today is still widely adopted to display metro routes (3). Many works in the literature have been inspired by Beck's work and aim at producing easy-to-use schematic maps.

Clark argued that travel time through a network is more important than travel distance (13). He proposed a schematic map in which space represents travel time. However, route planning based on his map is challenging due to the large amount of overlapping lines. More recently, a work by Wang et al introduced a *focus+context* technique that allows interactive exploration of metro maps in small displays (14). The outcome of focus+context methods are maps that highlight user routing information, while partially showing related information. This method is effective in orienting the user, but does not work if the user wants to explore different possibilities. Usually, focus+context maps are implemented using distortion techniques such as fish-eye lens. Haunert argued that map distortion should, whether it is possible, should be applied where the transit network is sparse. He proposed a graph-based approach to compute a new spatial mapping that better uses the available space (15).

2.3 Visual Analytics of Movement

Analysis of movement is currently an important research topic in visualization, and in the past few years, many tools to visualize movement data have been developed. A great introduction to this topic is provided by Andrienko et al. (16). Visual analysis of movement is closely related to PTS visualization since visualizing movement data is a core part of it.

Andrienko et al. list three visualization categories for movement data: *direction depiction*, *summarization* and *pattern extraction* (17). Zhong et al. (18) and Goncalves et al. (19) describe common techniques that are used to visualize movement data which include *static maps*, *space-time cubes*, *animated maps*, and *small multiples*. Work by Nguyen et al. (20) address the problem of representing the temporal information on the routes, namely, visually representing the different trips that occurs in the day along the same route. They proposed using a space-time cube in which two of the three axes represents the spatial information and the third axis gives the temporal information. This visual encoding represents a *normalized bus trip*, that is, a bus trip obtained with a shape-based aggregation of all the bus trips of a given route. By picking a time on this third axis the user can infer all the other information.

Animation techniques play an important role in visualizing movement data. A visualization tool that exploits animated maps is *TRAVIC* (21), it shows live world-wide vehicle movements using smooth animations. The focus of this work was to show many transportation modes, since animated maps are typically limited to one mode transportation (e.g. trains, bus, etc.). Another interesting work that uses animation is *Shangai Metro Flow* by Nagel and Gross (22). In their work they encouraged users to reflect on their perception of the environment as well as the visualization aesthetics, which was perceived as very pleasant. Visual aesthetics is important to consider especially when the visualization is targeted for the general public.

Janetzko et al. combined many techniques to show movement data from different perspectives. In their work they applied trajectory abstraction on a map and provided a simplified graph visualization with coherent topology. They also used a small-multiple technique to show changes in movement of the observed subjects (23). Small multiple diagrams are often used to represent data that change over time, as they offload memory to spatial location, accelerating both search and recognition. Conversely, animations require the human to memorize previous state to enable comparison. Although small multiples seems to be a better solution to visualize data that changes over time, their use is constrained by the available display area.

Flow Maps are another common visual encoding for geospatial movement data. Flow maps are a combination of maps and flow charts used to display object movements (e.g. vehicles, goods, etc) between different geographic regions. A pioneer of this technique is Tobler (24) but in his work visual clutter easily occurs when applied to a large dataset. An effective method to avoid visual clutter is *edge bundling*. Phan et al. (25) proposes the *Flow Map Layout* method to automatically cluster nodes into a tree hierarchical structure and then bundle neighboring flow lines to present the general flow trend. Buchin et al. (26) proposes a spiral-tree-based method to compute crossing-free flows that improves the flow maps readability.

CHAPTER 3

THEORETICAL FOUNDATIONS

In this chapter I describe some theoretical concepts that are important when it comes to design visualization tools for public transportation. A great introduction to some of these concepts is provided also by Colin Ware (27).

3.1 Wayfinding theory

To enable people move through an environment we need to think of how people build an understanding of an environment over time and how they use this understanding to seek information. We call *wayfinding*, or alternatively *pathfinding*, the cognitive process that people perform to compute and follow a path within an environment. The term was first used by the architect Kevin Lynch (28) referring to maps, street numbers and directional signs as ”way-finding” devices. Muhlhausen (29) explained that wayfinding requires more than maps and street numbers, namely it involves understanding the spatial organization of a place.

Woyciechowicz (1) described wayfinding as the process of collecting information from an environment to know where we are relative to where we want to go to and how to get there. Therefore a concept that is directly linked to wayfinding theory is *spatial cognitive maps*.

3.2 Spatial Cognitive Maps

Spatial cognitive maps is the term used to refer to human internal representation of an environment. Siegel and White (30) suggested that the formation of a cognitive map is the result of a three stages process. According to Siegel et al. in the first stage we learn information about key landmarks of the environment we are exploring; this kind of knowledge is also referred to as *declarative knowledge*. In the second stage we develop routing instructions from one landmark to another; the knowledge that we build in this second step is both *procedural knowledge* and *topological knowledge* since we develop this information by mean of sequential instructions and connections between different locations. The final step of this map formation process is the actual development of a cognitive map that let us also to estimate distances between places and to navigate from one place to one another even if we do not have the exact routing information. But in their work Siegel et al. ignored the importance of map technologies, in fact as proved by Thorndyke and Hayes-Roth (31), people can easily build cognitive maps directly from actual maps.

Another interesting insight comes from the study of Colle and Reid (32). The study suggests the importance of overview maps since the formation of cognitive maps is easier when the viewer can see everything at once.

Darken et al. (33) argue that perspective views are not quite effective to support the generation of mental maps since there are evidences that terrain features are not encoded in memory as three-dimensional structures; this is an important information to take into consideration when choosing the visual encoding for a geospatial visualization tool.

Kosslyn (34) proposed a theory of spatial knowledge based on two kind of knowledge, namely *categorical* and *coordinate*. With these two term Kosslyn encoded two ways in which humans build their cognitive maps. The former is due two the actual experience of going through an environment. The latter is due to visual imagery obtained with an overview.

3.3 Legibility measures

The study of Woyciechowicz (1) has identified system legibility as a barrier to use the PTS. To compare the quality of different networks solutions, transportation researchers developed a set of measures to evaluate the legibility of a network.

Their work is based on some insights of previous works in wayfinding and cognitive map theories. In particular they identify wayfinding as the combination of three kind of knowledge: *survey knowledge*, *procedural knowledge* and *landmark knowledge*. They also give definitions of these terms, namely survey knowledge is the ability of using a cognitive map, procedural knowledge is the ability to follow directions and landmark knowledge is the ability of building paths through landmarks.

The set of measures developed by the authors are based on how easy is to exploit these three kind of knowledge within a certain transportation network. Here I briefly sketch the most important:

- *Pattern repetition*: if some pattern is present in the network, that make it easier to navigate the network.
- *Simplicity*: it makes more easy for the user to exploit survey knowledge since less overlaps occurs between different lines.
- *Use of arterial road*: arterial road are more familiar to passenger and therefore they help to better exploit procedural knowledge.
- *Joining of transit hubs to landmarks*: this way the transit system becomes the link that connects different activity centers and this make it more easy for a passenger to navigate the transit system.

These measures could be important when studying a PTS to decide what visual encoding would better fit the considered scenario. For example we can imagine a preliminary study of the problem using these measures that highlights certain characteristics of the system, and therefore gives important insights on what is the most effective visual encoding.

3.4 Landmarks

Landmarks are significant physical, built, or culturally defined geographical features that stand out from their surroundings and help to locate geographic positions (35).

To decide which geographical features should be highlighted as landmarks is a challenging task. In fact, it is difficult to rank the saliency of different landmarks and often the relevance of a landmark is based on cultural aspects.

In the literature many approaches exist that develop formal models for the automatic extraction of landmarks from existing GIS dataset.

Raubal et al. (36) proposed a set of measures to formally establish the relevance of a geographic feature. The measures are based on the visual, semantic, and structural attraction of the feature and are used to extract landmarks from a dataset.

Elias (37) and Elias et al. (38) works describe knowledge discovery and data mining methods for landmarks extraction. These approaches are limited to the identification of buildings as landmarks, but also other terrain features such as parks and bridges could by identified as landmarks (39).

CHAPTER 4

DATA ANALYSIS

Today more and more the transport authorities are monitoring their infrastructures at an increasingly unprecedented level of detail and they are making their data available to the public. This opens the way for research in visualizing and analyzing these data to improve the PTS. These data can include geospatial, temporal and categorical information.

In order to understand the domain, a common terminology is defined below and will be used in this document.

- *Transportation network*: directed graph structure where node are stations and edges are available lines that connect nodes.
- *Transit route*: a sequence of nodes and edges with two locations that are defined as the departure location and the arrival location (alternatively, origin and destination).
- *Transit line*: a Public Transportation Service offered by a certain transport mode (e.g. bus, trains, etc).
- *Trip*: an individual ride on a certain road, therefore it is characterized by a temporal information.
- *Transfer*: a connection between one or more routes.
- *Journey*: a passenger travel from an origin to a destination within the PTS.

The PTS data is available in various formats, the most popular being *General Transit Feed Specification* (GTFS) (40). These feeds represent offline transit information and are updated every time the schedules changes (typically a few times a year). Many transit authorities have started to provide also real-time data collected by vehicles GPS. Google has released also a realtime version of GTFS (41).

An important and widely used source of geospatial data is *Openstreet Map* (42). *Google Maps* (4) is also a well known source of geospatial data. These services allow to get both raster tiles and vector tiles with a custom amount of features.

4.1 GTFS

The system that I have designed is based on the GTFS format. In the past few years, the General Transit Feed Specification has become the most popular format to model schedule data of transport systems.

The dataset consists of a set of tables, each containing different aspect of the PTS, such as trip stop times, stop locations and the availability of a service at any given time. The format can describe weekly schedules as well as day-wise trips. For each trip it provides also the sequence of stations as well as the stop times and the polyline that represents the path that connects each of the stations. Moreover, several attributes are available such as route description, route color and wheelchair accessibility among others.

A GTFS feed is composed by a set of text files collected in a ZIP file. To better understand chapter 6 I provide here a brief summary of the most important files.

- **stops.txt**. Contains the list of all stops along with their id, names and geographic locations.
- **routes.txt** Contains the list of all routes along with their id, names and identifying color.
- **trips.txt** Contains the list of all trips along with their respective id, route id, service id and direction.

- **stop_times.txt** Contains the schedules for all the trips, namely the stop times at each station along with the stop id, the stop sequence number within the route, the pickup type and the current traveled distance at the given stop.
- **calendar.txt** Contains the list of all the services along with their weekly availability.

The data is provided in a tabular format but its very nature is spatial since all vehicle positions and stop locations are defined in a geographic reference system. Nevertheless, we can see the PTS as a complex network in which nodes are represented by stations and edges are represented by trips. This last consideration will be useful for the design of the system (chapter 6).

4.2 Chicago Public Transport System

Chicago has been chosen as the context of the visualization. Its PTS is managed by the Chicago Transit Authority (CTA) and it is the nation's second largest. CTA has 1,865 buses that operate over 126 routes serving 11,104 bus stops, and 1,356 rail cars that operates over 8 routes serving 146 stations. The service operates 24 hours each day and on average, everyday 1.7 million rides are taken on CTA. During the rush hours, more than 1,000 means of transport are moving in the city.

These statistics give an impression of the amount of data that needs to be handled if we want to visualize all these vehicles on a map. For example, if we want to draw markers to show the realtime position of each vehicle, we might need to draw more than 1,000 markers every time we update the visualization. Clearly this large amount of data bring some problems.

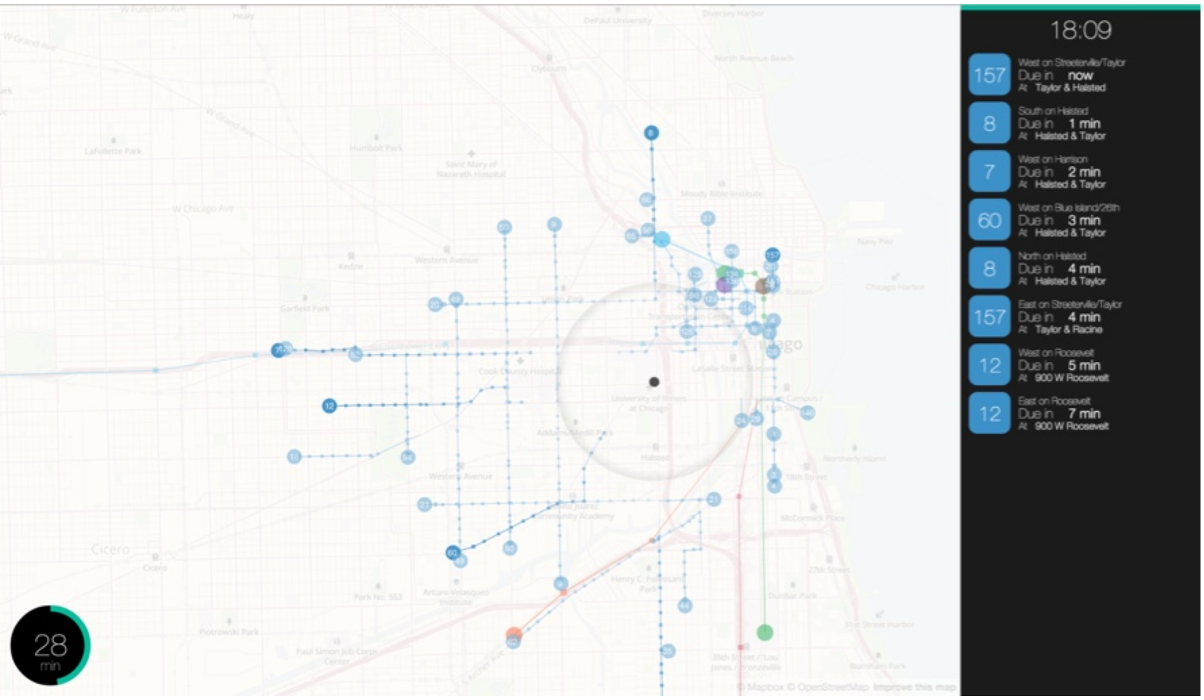
The primary challenge lies in displaying an amount of information that a human can process. Indeed, the risk is that if we visualize too much information the visualization will be cluttered and useless.

Another challenge lies in the performance issues that arise from the various technology bottlenecks. In particular, data need to be downloaded from the Internet, and we should limit the amount of data that needs to be downloaded to avoid both long latencies and high consumption of users' traffic. Moreover, finding efficient solutions to draw all the markers on the map is critical.

CHAPTER 5

VISUAL DESIGN

In this section I describe the tasks that the visualization should support and then I derive a set of design goals. Finally I describe the visualization techniques that I use to satisfy the design goals.



157 West on Streeterville/Taylor Due in now At Taylor & Halsted
8 South on Halsted Due in 1 min At Halsted & Taylor
7 West on Harrison Due in 2 min At Halsted & Taylor
60 West on Blue Island/26th Due in 3 min At Halsted & Taylor
8 North on Halsted Due in 4 min At Halsted & Taylor
157 East on Streeterville/Taylor Due in 4 min At Taylor & Racine
12 West on Roosevelt Due in 5 min At 900 W Roosevelt
12 East on Roosevelt Due in 7 min At 900 W Roosevelt

Figure 1: Overview of the visualization.

TABLE I: SEARCH TASKS GIVEN WHAT THE USER KNOWS

	Route known	Route unknown
Destination location known	Lookup	Browse
Destination location unknown	Locate	Explore

5.1 Tasks Analysis

Route planning is the main broad task the visualization should support. However, to be able to precisely establish design goals, it is necessary to identify the sub-tasks that are part of the wayfinding process. To understand what tasks the visualization should fulfill we need to know what the user is looking for and what the user already knows.

Table I illustrates that different search tasks are based on what the user already knows. If the user is able to estimate the location of her destination on the overview map, and she already knows the routing information from origin to destination, then she probably is looking for the closest station and the estimated time of arrival to that station. Conversely, if she identifies her destination on the map but she does not know how to get there, she probably wants to browse the visualization to find a route. When the user is not able to identify the intended destination on the overview map, but she knows what routes passes by the destination, then she probably wants to know the stations in the immediate vicinity and estimated time of arrival to those stations. She might want also to see the path on the map, and even look for alternative routes. When the user does not even known the routing information, then she probably wants to explore the visualization, looking for the information she needs.

To refine the tasks we also need to consider what we know about the user, and make some reasonable assumptions to guide the design of the visualization. Unfortunately, only few assumptions can be made about the target user since the visualization will be displayed on non-interactive ambient displays placed in public places.

While we know the start position for each route, namely the position of the ambient display, we cannot make assumptions about the final destination. For this reason, the visualization should enable users to see what the reachable areas of the city are.

The time that it takes to arrive at the destination is an information that the user is usually interested in. Furthermore, when the user has found a vehicle that passes in the vicinity of the intended destination, she wants to find out which stations that particular vehicle stops at.

For transport modes such as buses, stops are usually very close to each other. This does not necessarily hold for trains, and for this reason this information is very important because it lets the traveller know how close she can get to the destination.

It is important also to think of what the route selection parameters are, since with this tool I aim at satisfying the needs of different travelers. For example, a worker usually is looking for the route that takes him to the workplace in the shortest amount of time. Conversely, a tourist is looking for routes that take her close to important city landmarks and interesting places. To put it briefly, the problem does not make it possible to formulate assumption about the route selection parameters, therefore few assumptions as possible are made.

I assume that the user is looking for means of transport that passes within the immediate vicinity of the ambient display. Moreover, travelers are usually interested in knowing what the estimated times of arrival are for vehicles in the their vicinity, and where the stations these trips stop at are. Travelers may also have preferred modes of transport, and for this reason, the visualization should allow the user to plan a journey with the preferred transport mode in mind. Sometimes travelers have to catch more than one trip to get to their destination, therefore the users should be able to see connections of one trip to others.

Since the purpose of this work is to improve PTS accessibility, I also aim at designing a visualization that enables users to find new possibilities that they were not aware of. As shown by Thudt et al., serendipity is a trigger of exciting yet unexpected discoveries that plays an important role in information seeking (43). Therefore it is also desirable that the visualization fosters serendipitous discoveries.

5.2 Design Goals

Based on the analysis of the tasks, I defined the following set of goals to guide the design:

G1 Show nearby means of transport. The visualization must highlight what vehicles are in the surrounding area. More in detail, the visualization must highlight all the vehicles that the user can feasibly catch from the position of the ambient display. Obviously, a reasonable time frame should be decided to limit this search. Based on the assumptions we made, the user will be looking for transports that will depart in a relatively short amount of time (i.e. within 15/20 minutes). However, this parameter should be validated through user studies.

G2 Vehicle identification. To effectively perform route planning, the user should be able to identify the transport she has to catch. This means that the visualization must provide information about bus numbers and train names and/or colors.

G3 Show stations. The visualization must clearly point out not only station in the vicinity, but also the proceeding stations so that the user can see how close she can get to her destination.

G4 Show time dimension. One important aspect of route planning is timing information. The visualization must allow the user to infer at what time means of transport are at a certain location. Moreover, she should be able to determine estimated time of arrival of nearby transports as well as estimated time of arrival to a specific destination.

G5 Show transports paths. To follow a path is the core task of the wayfinding process. Therefore, the visualization must clearly show paths. This also lets the user determine the vehicle directions, the vehicles passing by, and ultimately the possible destinations.

G6 Highlight transfers. The visualization must highlight where transfers to other vehicles occur. The user should be able to determine how many transfer each route takes. In particular, the user should be able to locate routes that requires no transfers.

G7 Differentiate between transport modes. The visualization must clearly show what the transport modes are. The users should be able to plan journeys using their preferred transport mode, since some users may prefer some transport modes over others.

G8 Reduce visual clutter. Geographic visualization of public transport data tend to get messy easily. The visualization should show as many options as possible in order to satisfy all users. In summary, the visualization has to show a high amount of information without occluding the visual representation.

G9 Foster serendipitous discoveries. In an effort to improve PTS accessibility and increasing traveller awareness towards all available options, the visualization should be playful and should encourage users towards engagement of the application.

5.3 Overall Design

The user interface of the Web application is illustrated in Figure 1. The application is composed by a map, an info side bar on the right part of the screen and a timer on the bottom left part of the screen. The position of the ambient display is indicated on the map by the black circle. The map is where the animation of vehicles take place, hence it is the most important part of the visualization and all the techniques employed to enable route planning are described in detail in Section 5.4.

The info side bar displayed on the right part of the screen resembles a classical times table that displays estimated times of arrival of incoming vehicles. For each incoming vehicle, the info side bar reports also the closest station with regard to the ambient display location, and the direction of the vehicle.

The timer shown in the bottom left part of the screen report the current elapsed time within the animation. The number displayed represents the number of minutes from the time that the user is watching the visualization. The timer has been designed to give the user a hint about the time that it takes to get to the intended destination. Moreover, the green radial bar let the user to quickly determine the elapsed time without even reading the number.

5.4 Visualization Techniques

In this section I present the visualization techniques that have been employed to accomplish the design goals.

TransitTrace has been designed to visualize a high number of transit routes on an overview map. A quick analysis of the CTA data set reveals that there could be more than 1,000 vehicles in service at the same time. From the analysis of the tasks we know that it is not necessary to show all the trips that are in service. Yet, there could be a high number of routes that are relevant to the traveler.

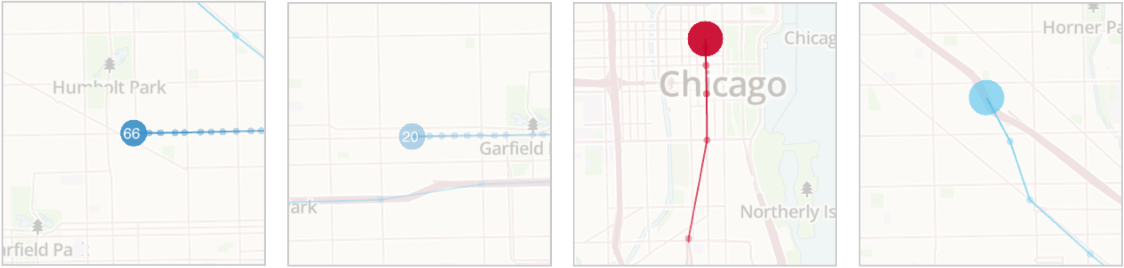
Since the visualization will be installed on non-interactive ambient displays and will serve multiple users at the same time, it is not possible to use one of the main visual design guideline, that is overview first, then details on demand (44). The visual techniques described in this section have been designed to overcome this problem.

I named the visualization TransitTrace because the core visual component are the traces left by the transports. These traces provide the most useful information to the traveler because they show vehicle movements over time and transfer to other vehicles. Moreover, the future movements of transports over time is simulated through an animation, while the level of details present on the map is adapted to the number of vehicles.

5.4.1 Visual Encoding

Transports are visually represented as colored circular markers that leave a trail as they move on the map. I assign to all buses the same blue color and I color each train with its identifying color, which is available in the GTFS data set. Moreover, to emphasize the transportation mode, train markers size is greater than buse markers size (**G7**). For each bus, I also report its number on top of the marker (**G2**).

To differentiate nearby vehicles from the vehicles that connect to them, I apply color interpolation and transparency (**G1**). More in detail, nearby transports are shown as fully opaque while the other



(a) (b) (c) (d)

Figure 2: Figures (a) and (b) illustrate the visual encoding used for buses. It is possible to see that the color of the bus in Figure (b) is neutral because it is not in the vicinity of the user. Figures (c) and (d) illustrate the visual encoding of trains. The foregoing discussion about the colors holds also in this case.

transports are semi-transparent and their color is interpolated with a neutral color such as white. The result is shown in Figure 2.

The trails that the means of transport leave behind are the core part of the visualization. In fact, trails disclose the vehicle trajectories (**G5**), allowing the user to see the vehicle directions. Although the GTFS dataset provides the exact shapes of each vehicle trajectory, they are not used to draw vehicles trails. In fact, exact shapes on an overview map are meaningless, since they do not provide any useful information and they make it more difficult to follow vehicle paths. Conversely, the trajectories are obtained through the interpolation of station positions. This simplification improves the trajectories legibility, since the lines represent direct connections from station to station. In fact, in this way the trails bring to light the real direction and make it more easy to predict future destinations of vehicles.

To reduce the visual clutter (**G8**), I designed the trails so that they gradually fade out. Moreover, only the parts of the vehicle path that are relevant to the traveler are shown. For example, vehicles that

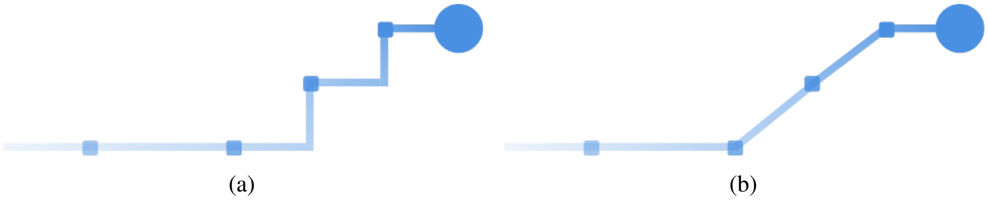

(a) (b)

Figure 3: Figure (a) shows a vehicle trail using the exact shape for the trajectory. Figure (b) shows a trail obtained through interpolation of station locations.

connect to vehicles in the immediate vicinity of the ambient display are visualized on the map only after they have reached the transfer station. This dramatically reduces the amount of overlapping between different vehicles trails leading to a more clear representation.

Each transport trail is long enough to intersect with the trails of means of transport that connect to it. This helps to determine possible transfers (**G6**) to other vehicles. However, this feedback is not precise since not necessarily an intersection between two trails represents a feasible transfer. In fact, there might not be enough time for the user to walk from one station to the other. To overcome this limitation I display a visual feedback when there is an intersection between two trails and the transfer is feasible. Figure 4 shows a proof of concept of the transfer visual feedback.

Stations are visually represented as small rounded rectangle markers that appear as the transport markers pass by. The most recent station marker on a trail is magnified. This helps the user to identify where a vehicle stops when two or more trails are close to each other. The visualization does not show the name of the stations, since too many stations are present on the map at the same time. Nevertheless, stations position are shown because they let the user know what destination she can get to (**G3**).

(a) (b) (c)

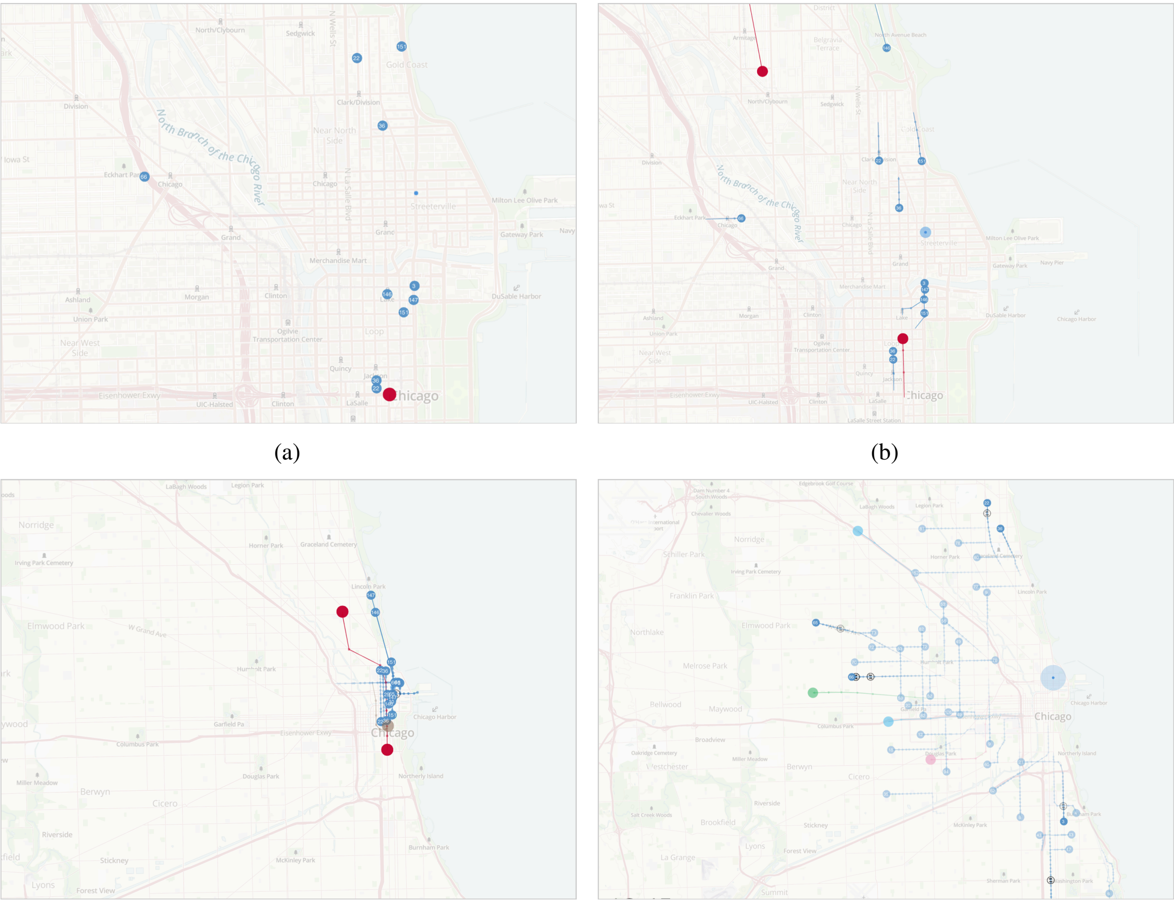
Figure 4: The sequence of figures illustrates how transfers to other vehicles are highlighted. Figure (a) represents the situation before the transfer station. Figure (b) shows that when the vehicle passes by a station where there is at least one possible transfer, a representative marker pops up and fades out as the connecting trip passes by (figure (c))

The look and feel of the visualization has been designed to be minimalistic to reduce the visual clutter, but playful to attract the user to observe and perhaps discover useful information she was not aware of (**G9**).

5.4.2 Spring Animation

Visualizing the movement of many means of transport at once while supporting route planning is a challenging task. To address this task I propose an animation (Figure 5) that simulates vehicle movements forward and backward in time.

The animation resembles the play and rewind of a video tape. As the animation plays forward, the user sees which vehicles pass by the intended destination. As the animation plays in reverse, the user can trace the vehicles back toward the origin. I refer to this technique as ”Spring Animation” because it reminds also the behavior of a spring.



(a) (b)

(c) (d)

Figure 5: The figures show the various steps of the simulation of the network evolution. Figure (a) illustrates the initial state of the visualization. In this state all the vehicles are shown at their actual real-time position. In this stage we have a higher level of details (i.e. more streets and landmarks) since the view is zoomed to a bound that contains the vehicles that are approaching the location of the ambient display. In figures (b) and (c) we can see that the means of transport gradually expand on the map, moving in several directions. At this stage the zoom level is adapted to an overview of the city of Chicago. Finally in figure (d) the simulation is close to the end. When it reaches the final state, the simulation plays in reverse until it gets to the updated initial state.

The animation starts from the current vehicles positions, to let the traveler see where vehicles are relative to her position. During the first 30 seconds, the animation gradually progress over a time span of an hour. In this way the user can see how the vehicles will move and if any of these vehicles lead to the desired destination. The current elapsed time of the animation is displayed on the bottom left corner of the screen (see Figure 1). After 30 seconds, the animation reverses, allowing the user to trace back the route from an intended destination. The reverse animation makes the visualization effective for route planning. Without the reverse animation, it is difficult for the user to remember where vehicles come from. The reverse animation let the user trace back the vehicles to their origin, allowing to reconstruct the entire route by spotting transfers and ultimately to determine where the first vehicle she should catch is.

The ease function of the animation has been chosen to emphasize certain parts of the animation. In its initial status, when it is showing the current position of vehicles, the animation should run slowly, being almost still. In fact, the user should be able to see the actual vehicle positions, and she should be able to spot the closest station to her position.

Then the animation should progressively speed up to until it reaches a certain threshold speed. The threshold speed should be set to be high enough to play the animation in a short time, and slow enough to enable the user to follow vehicles and to let her read bus numbers. The behavior of the animation playing forward is modeled by the exponential function illustrated in Figure 6.

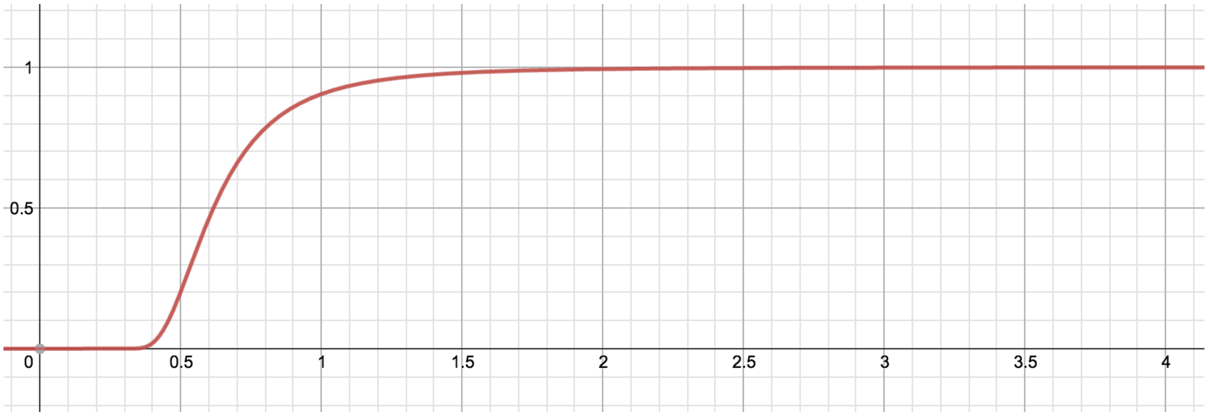


Figure 6: The figure illustrates the animation easing function that has been employed. As described by the curve, at the beginning the animation is relatively still, and this let the traveler understand what vehicles are in her vicinity.

The exponential function illustrated in Figure 6 is described by the following math function:

$$\Delta t = f(t) = \begin{cases} e^{-\frac{1}{5 \cdot t^2}} & \text{if } t > 0 \\ \\ 0 & \text{otherwise} \end{cases} \tag{5.1}$$

Where the increment of the animation time Δt depends on the current animation time t. When the animation plays in reverse the user should be able to trace back the route from an intended destination. At this stage, we do not need to emphasize any part of the animation, therefore an inverse linear function has been chosen as ease function for the second part of the animation. When the animation plays in reverse, its speed is higher than the speed of the animation playing forward.

5.4.3 Dynamic Level of Detail

In the initial state of the animation, only a small number of vehicles are present on the map, namely those within the immediate vicinity of the ambient display. Conversely when the animation progress, many vehicles appear on the map distributed along the many possible destinations. As the vehicles move away from the initial position, an overview of the entire city is necessary to see them all. The zoom level that allows to look at all the vehicles positions does not allow to have a closer look of the user location surroundings. However, to effectively plan a route a traveler should be able to understand how to reach the stations where to get on the means of transport. To overcome this limitation, the zoom level of the map is constantly changed to show higher portion of the map as the vehicles move further away from the ambient display position. Conversely, in the initial state of the animation all the vehicles are within the immediate vicinity of the user location; this allows to show their position with a greater level of details as illustrated in Figure 5(a).

CHAPTER 6

SYSTEM DESIGN

In this chapter I describe how I design the system in order to overcome technical problems that come with the complexity of the transit data.

The input of the system is a GTFS dataset, and the output is the visualization. The are two main problems in the process that transforms the GTFS data set in the final visualization.

The first problem relies on the manipulation of the data to obtain the information that are relevant for the visualization, during a given time at a certain place. Since the visualization should be accessed by anyone, it has been designed as a web-based application. The manipulation of the dataset that contains all the schedules requires a level of performances that cannot be achieved by a modern browser running Javascript. For this reason, a server side preprocessing is required and a client-server interface has been designed.

The other problem relies in the rendering of the relevant data on the screen. To obtain a smooth visualization of all the vehicles that has to be displayed on the map, some graphical performances are also required, and this is why I develop the visualization using WebGL, a JavaScript API for computer graphics rendering, that allows GPU hardware-acceleration.

The following sections explain in details the various steps of how the final visualization is obtained from the dataset.

6.1 Architecture Overview

The visualization I designed has to run on ambient displays. To make the visualization portable and accessible it is developed as a web-based application. This means that the display has to be backed up by a computer connected to the Internet, and the visualization is displayed on the screen through a web-browser.

Areas like Chicago have highly complex transportation networks. Its GTFS dataset now comprise 47,716 trips contained in the **trips.txt** file. The schedule for each of these trips is contained in the **stop_times.txt** file that, as of this writing, includes 2,911,188 entries and weights 180MB. The whole GTFS set of files sum up to 200MB and any modern browser running javascript cannot hold an amount of data as large as this. In fact, just a simple iteration over the whole list of schedules take seconds, and sometimes the browser would even crash. I highlight also that the fact that the browser crashes makes it impossible to transform the dataset into a more efficient data structure directly within the web application.

The visualization requires a smooth animation, and to be smooth the animation should run at least at 20 frame per second. This means that each update of the visualization should not take more than 50ms. Each update requires to update the model and to perform a draw call. In order to update the model we need to query the dataset to obtain the vehicles that should be displayed in the current spatio-temporal bounding box. Clearly it is not feasible to update the model of the animation in such a short amount of time given that just a simple iteration over the schedules requires seconds. Moreover, the client does not need to hold the entire dataset

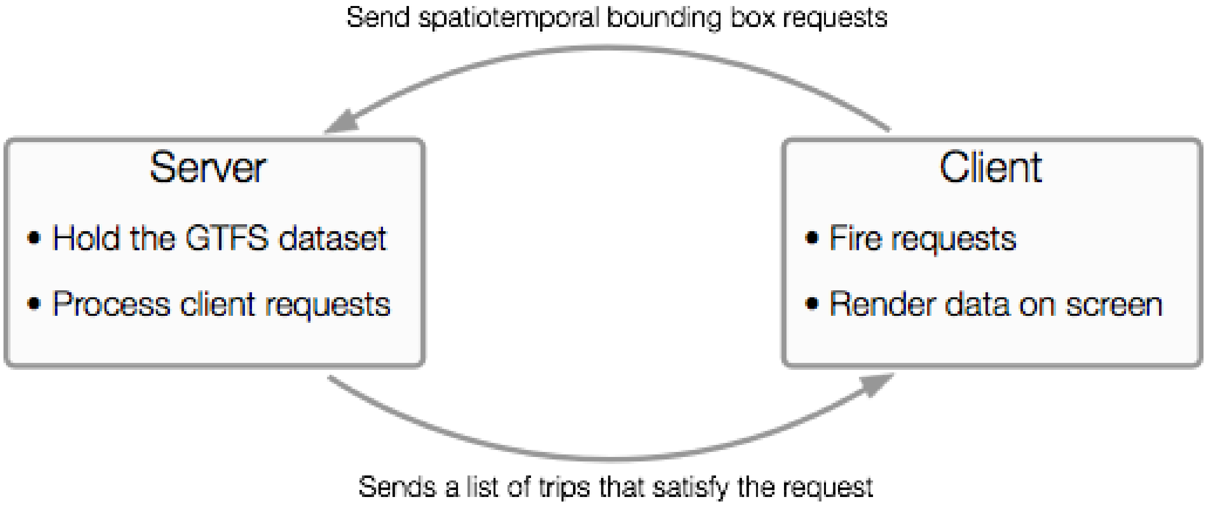


Figure 7: Architecture overview.

since only a subset of the trips has to be visualized. This lead to the conclusion that a server side preprocessing of the data is required to make the visualization feasible.

The architecture that I designed offloads the data preprocessing to an external server. More in detail, the client is responsible to fire data requests to the server and to perform the draw calls needed to render the visualization on screen. The server holds the GTFS dataset and processes the client requests returning a list of trips that satisfies the request parameters. In this way we reduce both the amount of data that the client needs to load, and the computational time needed to update the model of the animation. Moreover, the server can update the dataset every time the schedules are updated, and this operation is transparent to the client.

6.2 Client-Server Interface

A simple approach to visualize vehicle movements on a map is to fire a positions request every time that we want to update the position of the vehicles. This approach is the most commonly used to display live maps and it has the advantage that the client can be very thin. However, this method implies that without new positions request, vehicles position will not be updated. Since I aim at obtaining smooth animation, the requests have to be fired frequently and this results in a heavy network traffic. Even if we can guarantee a stable connection, the response time of the server could exceed the short timespan that we have to update the animation.

The interface that I design to overcome these shortcomings, uses a look ahead approach. More in detail, every time the client fires a request, not only it will ask for the list of trips that are on the map at a certain time, but it will also ask for all the trips within a certain timespan. In this way, the client can hold a buffer of the trips and consequently it can reduce the number of requests. Another benefit of this approach is that it is possible to pipeline the entire process, namely meanwhile we animate the current data we can fire a request for the next timespan. This guarantees that the animation runs smoothly and continuously.

The subset of trips that are visualized is not only constrained by the temporal bound, but also by a spatial bound. In fact, we need to visualize only the trips within the immediate vicinity of a certain location (the ambient display location) and the possible transfers from these trips. The transfers themselves are defined as the trips within the immediate vicinity of another trip stop. Therefore, the server will filter out also the trips that are not within these spatial boundaries.

To summarize, the request will be specified in terms of a *spatio-temporal bounding box* and a number transfer options.

Definition 1. *A spatio-temporal bounding box* B_{st} *is a tuple (*O_{lat}*,* O_{lon}*,* R_v*,* W_{day}*,* T_s*,* T_e*,* W_s*), where*

- O_{lat} *is the latitude of the origin*
- O_{lon} *is the longitude of the origin*
- R_v *is the maximum radius that defines the immediate vicinity*
- W_{day} *is the day of the week*
- T_s *is the start time*
- T_e *is the end time*
- W_s *is the walking speed*

Definition 2. *The transfer options are specified as s tuple (*T_t*,* R_t*), where*

- T_t *is the maximum waiting time for a transfer*
- R_t *is the maximum radius that defines the bounds of the area where to search for transfers*

Based on definitions 1 and 2, a RESTful service has been designed. The client can request the trips using the RESTful service provided by the server. The requests take the form of an HTTP GET.

`GET /api/trips/:lat/:lon/:rad/:dayofweek/:begin/:duration/:walkingspeed`

where:

- <`:lat`> is O_{lat}
- <`:lon`> is O_{lon}
- <`:rad`> is R_v
- <`:dayofweek`> is W_{day}
- <`:begin`> is T_s
- <`:duration`> is T_e - T_s
- <`:walkingspeed`> is W_s

The server after processing the request will respond with a JSON file containing all the trips that satisfy the conditions specified by the request. The response is organized by trip id, and for each trip several information are provided, such as the route id, the type of the means of transport and the list of the stops. Each stop will contain several information, such as its location, its arrival and departure times, and a list of possible transfers to other lines. An example of a typical response is illustrated in Figure 10.

```
{”445084170210”:
    {
     ”routeId”:”66”,
     ”routeLongName”:”Chicago”,
     ”direction”:”East”,
     ”stops”:
            [
             {
               ”arrivalTime”:{”hh”:5,”mm”:19,”ss”:58},
               ”departureTime”:{”hh”:5,”mm”:19,”ss”:58},
               ”stopId”:”510”,
               ”stopSequence”:1,
               ”pickupType”:0,
               ”lat”:41.89492553,
               ”lon”:−87.77484321,
               ”name”:”Chicago & Austin Terminal”
               ”transfers”: [
                             {
                               ”tripId”:”445084168325”,
                               ”stopIndex”:66
                             },
                             ...
                       ]
             },
             ...
            ],
     ”closestStopIndex”:66,
     ”hop”:0,
     ”type”:3
     },
    ...
}
```

Figure 8: JSON response example.

6.3 Data Structures

The GTFS format specifies data in a tabular fashion. This format is not suitable to perform efficient searches, therefore to shorten the response time of the server we need to transform the dataset into another data format that allows efficient searches. The nature of public transportation data is spatial. In fact, to display movements of means of transport within the immediate vicinity of the installation, we need to retrieve all the vehicles for which their trajectory intersects the spatio-temporal bounding box of the request.

A transportation network can also be seen as a directed graph $G = (V, E)$, where each node $v \in V$ models a station and each edge $e \in E$ models a time-dependent connection between two stations within the trajectory of a given trip. Two stations can be connected by multiple edges if they belong to the trajectories of more than one trip. The interval of each edge e_{th} of the graph is defined by a pair of functions (t_t, t_h), where $t_t(e_{th})$ returns the time of the tail of the edge and $t_h(e_{th})$ returns the time of the head of the edge. For each edge e of the graph, $t_h(e) > t_t(e)$. A trip can be defined as an alternating sequence of vertices and edges v_0, e_{01}, v_1, e_{12}, v_2, ..., v_{k-1}, $e_{(k-1)k}$, v_k, where $\forall m, n, p$ with $m > n > p$, $t_t(e_{nm}) = t_h(e_{pn})$.

This data structure enables efficient search of relevant movement as described in section 6.4. In fact, we visualize vehicles for which at least one segment of their trajectory intersects the spatio-temporal bounding box. This is equivalent to find the stops that are within the spatial boundaries and retrieve the outgoing edges that are within the temporal boundaries.

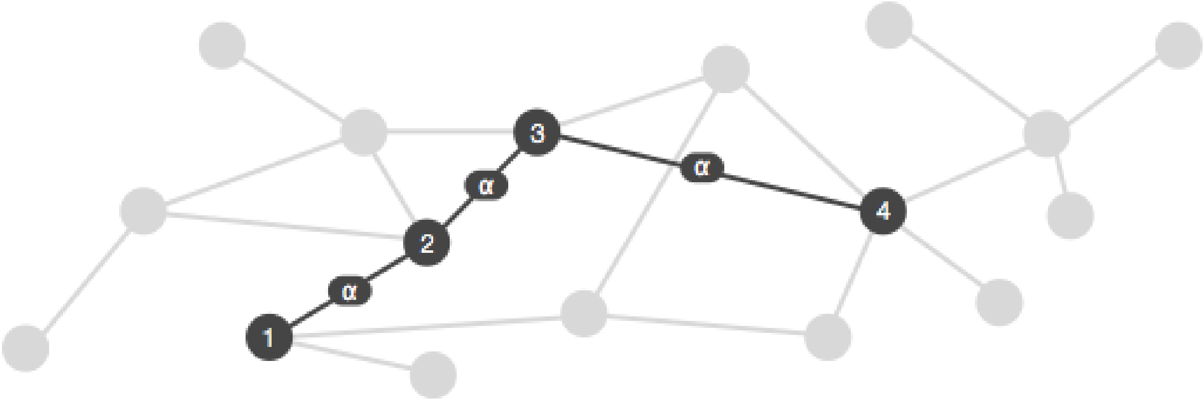


Figure 9: A trip is a temporal path through the transportation network, where all edges refer to the same trip id.

Each edge of the graph stores also the trip id that it refers to. Other information regarding a trip, such as the route id and the mean of transport type are held in a separate structure to avoid redundancy. The structure that keeps this information is a dictionary indexed by the id of the trip to allow fast retrieval of information regarding a trip.

6.4 Processing Client Requests

Every time a client fires a request, the server has to perform a search through the network to retrieve all the relevant trips. In this section I refer to the concept defined in Section 6.2 to describe how the search algorithm works. To describe the algorithm we also need to define what a transfer is.

Definition 3. *A transfer is a tuple (D,B, T_D, T_B), where*

- *D is the stop where a passenger disembarks from the previous vehicle*

- *B is the stop where a passenger boards the next vehicle*
- *T_D is the time when a passenger disembarks from the previous vehicle*
- *T_B is the time when a passenger boards the next vehicle*

At this stage of the project, only one transfer is shown. Therefore a transfer is relevant if D is a stop where a vehicle that is within the immediate vicinity of the ambient display stops at time T_D, and B is a stop where the next vehicle stops at time T_B. The difference between T_B and T_D should not exceed the maximum time T_t allowed for a transfer and should be enough to guarantee that the passenger can reach B. Moreover, the distance between D and B should not exceed R_t.

The algorithm that I designed to retrieve trips from the graph structure is recursive. Each recursive call correspond to a transfer hop. I refer to the hops as the number of transfers. By definition, the trips within the immediate vicinity of the ambient displays have number of hops equal to 0.

The input of the algorithm are a spatio-temporal bounding box and the maximum number of transfer allowed. The algorithm retrieves all the trips that satisfy the spatio-temporal bounding box conditions, and afterward it perform a recursive call to search for transfers. The algorithm stops when it reaches the maximum number of transfer allowed.

```
SearchRelevantTrips(spatioTemporalBoundingBox, numberOfTransfer)
  A = empty set

  if numberOfTransfer == 0 then
    return A

  S = find stops within the spatial boundaries
  For each stop s in S
    For each edge e in outgoing edges of s
      if edge is within the temporal boundaries then
        A += trip that edge refers to

  if numberOfTransfer == 1 then
    return A

  For each trip t in A
    For each stop s in t stops
      B = compute transfer spatio temporal bounding box
      T = SearchRelevantTrips(B, numberOfTransfer −1)
      A += T
  return A
```

Figure 10: Search algorithm pseudo-code.

6.5 Visualization Implementation

In the previous sections I have described how the architecture has been designed to reduce the work that the client needs to do to visualize the data. In fact, the server takes care of the retrieval of relevant trips from the GTFS dataset. Although this has dramatically reduced the computational time to process data on the client, the web application need to be carefully designed to avoid performance pitfalls.

Figure 11: Visualization layers.

The visualization has been divided into four logically independent layers: the Map Layer, the Transit Layer, the Side Info Bar Layer, and the Animation Timer Layer. Each of these four layers shares the same model, therefore every layer shows information that are coherent with the other layers. The Map Layer renders the map and provides API to convert geographical coordinates to pixel offset within the map container. The Transit Layer is built on top of the map layer and it shows the most important part of the visualization, which is the simulation of the transit network. The Side Info Bar Layer shows information regarding estimated time of arrival of means of transport within the immediate vicinity of the ambient display. Finally, the Animation Timer Layer displays the current time of the animation.

6.5.1 Technologies

As I mentioned in Section 6.1, we have a limited time to update the animation in order to enable smooth animations. During this short timespan we need to update the model of the animation and perform the draw calls. The computational time taken by the former has been greatly reduced introducing a server that preprocess the data, but to obtain smooth animations we also need to minimize the cost of draw calls.

Traditionally web browsers rely entirely on the CPU to render web page content. Although CPU is suitable to render common web pages, if we need to perform computational expensive rendering GPU hardware-acceleration is necessary.

Since the animation should run smoothly, to speedup draw calls is fundamental. For this reason, the Transit Layer and the Map Layer, which are involved in the animation, have been implemented exploiting GPU hardware-acceleration. GPU hardware-acceleration can be accessed in modern browsers through WebGL API (45).

The Map Layer has been rendered using MapBox GL (46), a Javascript library that allows to render vector maps using WebGL. MapBox GL enables smooth map animations, and this feature is fundamental since the map zoom has to be dynamically updated to show larger portion of the map as the vehicles spread on the map. The vector tiles data to render the map is obtained from OpenStreet map (42).

To render the Transit Layer, THREE.js has been used (47). THREE.js a lightweight cross-browser JavaScript library to display animated graphics. It is built on top of WebGL, therefore it takes advantage of GPU hardware-acceleration.

The Side Info Bar Layer need to be updated only once every minute, and the amount of data that it displays is minimal. For these reason, this layer have been implemented using SVG.

The Animation Timer Layer need to reflect the time of the animation, so it is constantly updated. Nevertheless, its updates carry no computational weight, therefore, the layer is implemented using SVG.

6.5.2 Techniques

The idioms that I have adopted to design the core part of the visualization, which is the Transit Layer, can be obtained using simple shapes. In fact, the entire visual design can be obtained using just circles and polylines.

To further improve performances, vehicles and stations are rendered as a unique point cloud, which is a set of vertices that the graphic card renders individually. In this way, every vehicle and station requires only one vertex to be drawn.

Natively the graphic card renders vertices as squares of the configured size. To obtain different 2D shapes I developed a shader that takes in input a template texture representing the shape and it draws the vertex using that texture. This technique is referred to as billboarding (48) and it is often used in combination with the point cloud to efficiently render an high number of shapes (49).

6.6 Optimizations

In this paragraph I briefly discuss some optimizations that can be applied to the system to obtain a substantial speed up and to improve its scalability.

This research work has not been focused on the server side implementation, since the research goal is to design and develop a novel visualization. However, to test the application in a real case, such as in the city of Milan, some optimizations are necessary.

The main performance bottleneck in the system is the search of vehicles in the vicinity of the ambient display. In the current dummy implementation of the system, for each request the algorithm searches through the entire dataset for vehicles that are in the vicinity of the requested position. A more efficient approach to query the dataset is represented by quadtrees (50). Quadtrees are data structures that represent a two-dimensional recursive spatial subdivision. This kind of data structures have been adopted in cartography since early '90s (51) due to their efficiency in two-dimensional space queries. In a quadtree, each internal node contains exactly four children. The search works starting from the root node and recursively filtering out the subregions that does not intersect with the search bounding box. In our domain we can use this approach to improve the efficiency of the trips selection queries by organizing the transport dataset into a quadtree. In this way, when the server receives a request the algorithm searches through the quadtree structure to find the subregion that contains all the relevant trips. More in detail, the search algorithm recursively query the quadtree, filtering out at each step the bounding boxes that do not intersect with the requested bounding box. We can immediately see the terrific advantages of this approach if we consider integration of multiple public transport systems. For example, if the system runs on both Chicago PTS dataset and Milan PTS dataset, and the system receives a request for a position in Milan, then the search algorithm with just one step can immediately exclude the entire Chicago dataset. This approach therefore greatly improves the scalability of the system and ultimately reduces the cost of servers required to run the system.

The other optimization consideration is with client-side performances. The current implementation requires heavy GPU load for each draw call, since the buffer that contains vehicles coordinates is updated every frame of the animation with the new vehicles coordinates computed by the CPU. An interesting solution to explore, in order to avoid to reload the buffer every frame, is the implementation of the entire animation with shaders. More in detail, it would be interesting to study how to pass the animation as a buffer that can be interpreted and manipulated by the shader program. In this way we can take advantage of high parallelism of the GPU and reduce the load on the CPU.

This two optimizations are part of the future work that will be done in order to test the visualization at large scale.

CHAPTER 7

EVALUATION

In this section I describe a case study to demonstrate how the visualization is used to enable wayfinding.

7.1 Case Study

This case study is to demonstrate the usefulness of TransitTrace to find routes. I analyze the case of a student that finds an installation of TransitTrace at the University of Illinois at Chicago.

Assuming that the student wants to go to the Navy Pier, he watches the visualization to seek trips to get there. While the animation plays forward he looks for vehicles that pass by the Navy Pier.

After a while the animation reaches the state represented in Figure 12, and the user sees that the bus 29 is the first feasible connection to the Navy Pier. The user notices also that the bus color is not fully opaque, therefore he knows that he has to find the nearby transport that enables him to transfer on the bus 29. Therefore he waits for the rewinding of the simulation and he traces the bus back. The trail of the bus helps the user to follow back the movements, because he can see a visual preview of the trajectory (Figure 13).

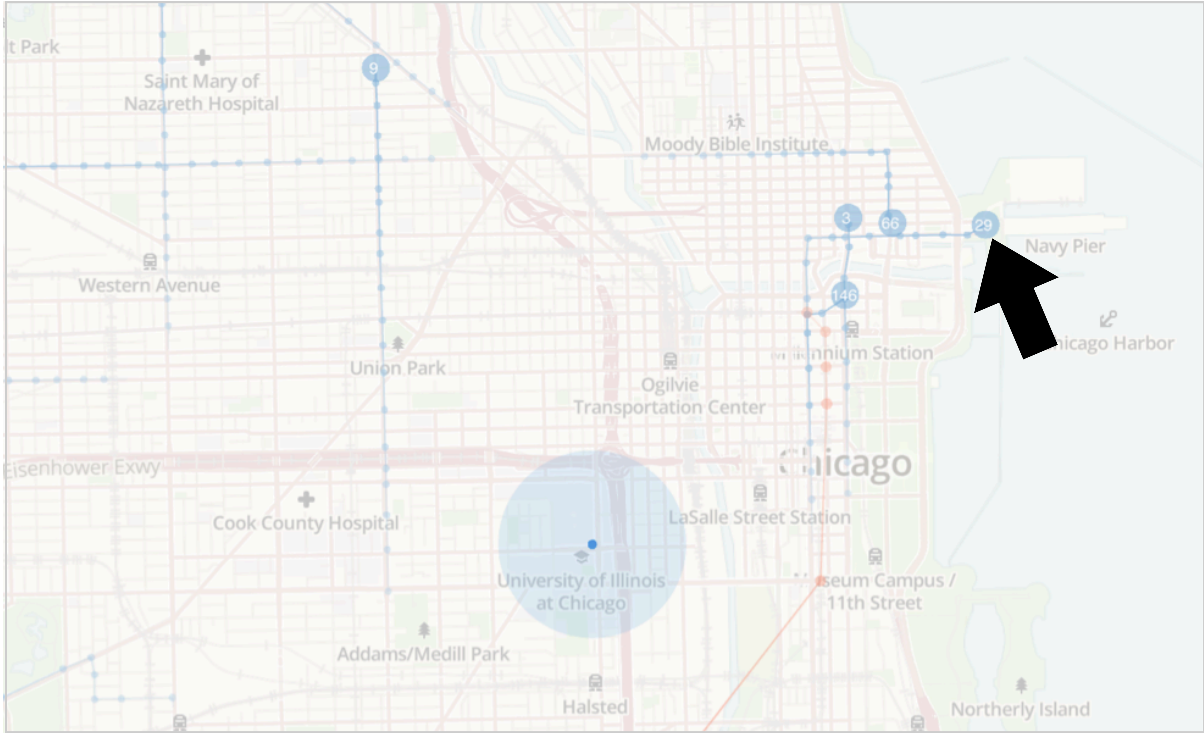


Figure 12: The bus 29 is the first one that passes by the Navy Pier from the user location. The transparency of its marker tell the user that the bus does not origin in the vicinity of the user location.

After a while the bus disappear leaving a transfer marker that guides the user to see what is the connecting transport. As the simulation goes back to the initial state the bus number 12 approaches the transfer marker location as illustrated by Figure 14.

Then, as the bus passes the transfer marker, the transfer marker is removed. This indicates that the bus number 12 is the bus that the user needs to catch in order to transfer to the bus 29, which will take him to the Navy Pier. Finally, the user traces the bus 12 back until it reaches the closest station to the user location (Figure 15). At this point, the user knows that to get

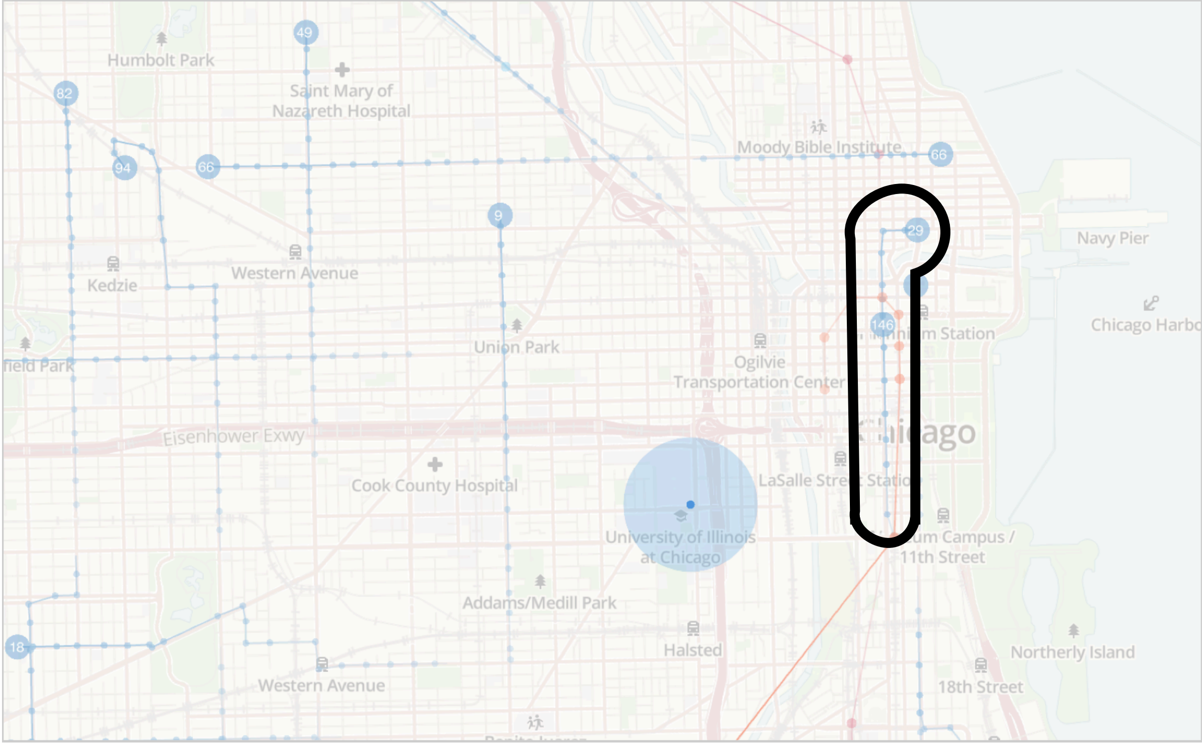


Figure 13: The bus 29 trail helps the user to follow the bus back by revealing in advance the vehicle path.

to the Navy Pier he has to catch the bus number 12 and then transfer to the bus number 29. I emphasize that in the meantime other users could have watched the visualization seeking information to other destinations. I also highlight that the idioms adopted to design the visualization allow the users to easily focus on the transports he needs.

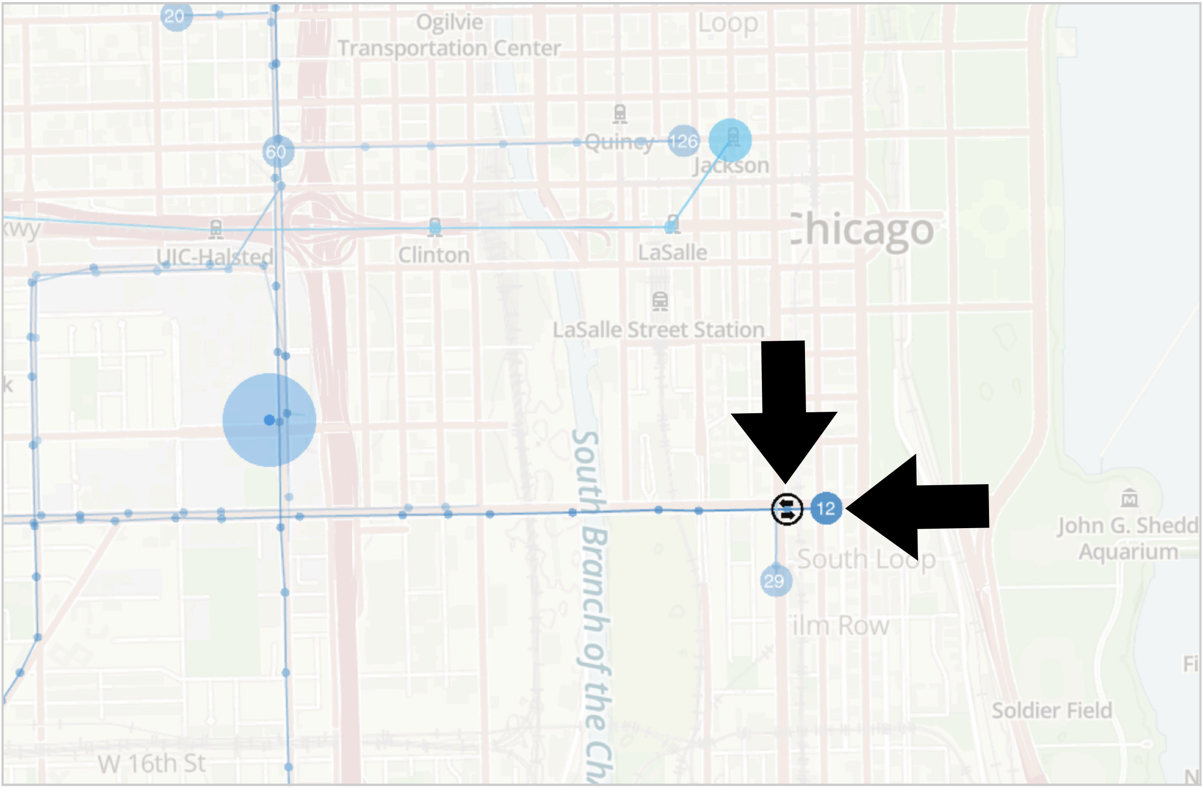


Figure 14: Bus 29 is no longer on the map, and in its place there is a transfer marker that indicates the location where the transfer will occur. In this figure the simulation is playing backward in time to the initial state, and this enable the user to see that the bus number 12 will pass by the transfer location.

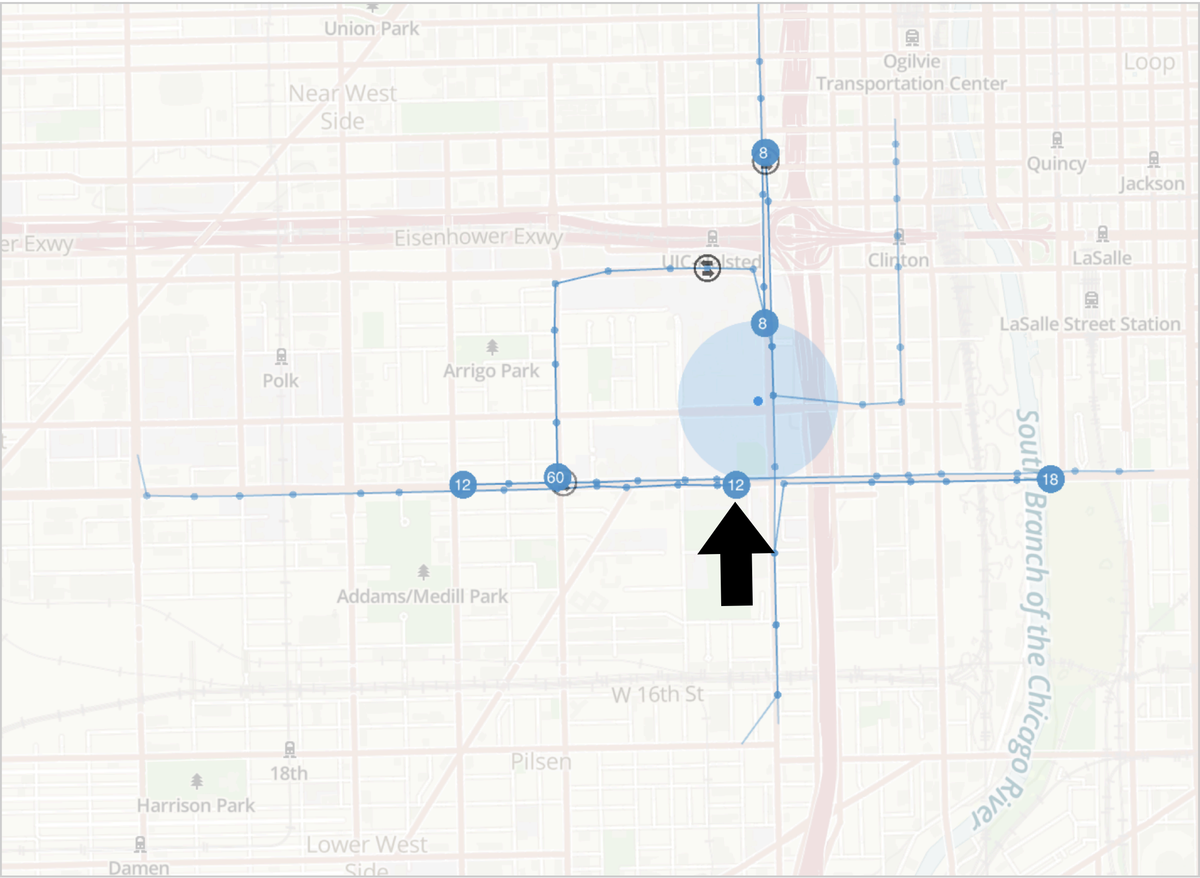


Figure 15: The figure shows at which stations the bus 12 stops in the vicinity of the user location.

CHAPTER 8

DISCUSSION AND FUTURE WORK

This work shows that even without user interaction it is possible to effectively enable real-time wayfinding on a dynamic overview map. The visual design enables exploration of a PTS by multiple users at the same time. In fact, the users can easily focus on the means of transport that are relevant to get to the intended destination.

Another important result is that the user can see many possible routes to her intended destination. The aim is to make the user aware of all the solutions that she has, and let her freely draw her path to the destination. As a matter of fact, this approach enables casual and highly effective exploration of the transit network, without requiring any sort of input from the user. The routes that are displayed, are all available at the time the user is looking at the visualization. Therefore, the user does not waste her time looking at routes that she cannot catch, as it happens with static maps.

The proposed approach presents some limitations that have yet to be addressed.
First of all, stations name are not displayed on the map and this is an important information when it comes to follow a route. For example, it is difficult for the user to understand what stations she is supposed to disembark in order to board on a connecting vehicle.
Often stations are also named after the street name where they are located. For this reason, station names help the user to find stations.

To partially solve this issue, it is possible to take advantage of the zoomed view that we have in the initial state of the animation to show names of stations within the immediate vicinity of the ambient display.

A limitation of the usage of an overview map, is that areas with a high number of vehicles tend to clutter the view. This situations affect the readability of the map, since many vehicles overlap to each other. To overcome this limitation, it is possible to explore map distortion techniques. For example, the map could be distorted based on the density of vehicles.

Another limitation of the usage of an overview map, is that it is difficult to determine specific places (i.e. theater, museums, malls, etc). For example, if the user wants to go to the theater, she has to know the approximate location of the theater on the map. This limitation could be minimized by carefully selecting the landmarks to be displayed on the map. To do that, a measure of saliency for the landmarks should be defined. Moreover it can be interesting to explore the possibility of displaying near-by vehicle landmarks. More in detail, we can temporarily show landmarks within the immediate vicinity of the vehicles, and fade the landmarks out as the vehicles move away. This solution would let the user find out more about what places are accessible from her location.

In its current version, the visualization displays vehicle movements based on interpolation with GTFS feeds. Today more and more transit authority are publishing realtime vehicles position collected through GPS. A future work can improve the visualization by integrating static schedules with realtime updated position. This will give the user a more accurate estimation of the current vehicle positions as well as of the future movements. In fact, realtime data would give information about delays and the actual vehicle speed, that can be used to predict more accurately the future positions of the vehicles.

CHAPTER 9

CONCLUSION

This thesis introduces TransitTrace, a web-based interaction-free visualization that enables navigation of Public Transport Systems on an overview map. The basic idea of the project is to design and develop a large-screen interaction-free visualization showing a summary view of relevant transit information.

I analyzed the Public Transport System domain, and I discussed the main tasks that a visualization tool should support. A brief analysis of the Chicago PTS revealed the challenges of developing a map visualization that enables route planning. In fact, the high complexity of the network tends to result in a highly cluttered visualization.

I presented a visual design that combines trajectory drawing and animation to enable route planning on an overview map. I showed how using trails to partially reveal the vehicle trajectories, it is possible to see reachable destinations and possible connections to other transit lines. Trails are also effective to limit visual clutter.

The initial design of the animation was not effective to enable route planning, since when vehicles reached an intended destination, the information about their origin was lost. I showed how the usage of a reverse animation overcomes this problem. In fact, when the animation plays in reverse, the user can trace back vehicles from the intended destination, ultimately reconstructing the whole route.

Furthermore, I described the complexities of the transit data, and how to deal with them. I described how the system has been designed to efficiently pass this large amount of data through various bottlenecks to an end user's screen. I also described an use case to show how this technique is used to enable wayfinding.

Finally, I highlighted limitations of the technique, and I discussed the possible solutions to be explored as future work.

CITED LITERATURE

1. Woyciechowicz, A. and Shliselberg, R.: Wayfinding in public transportation. Transportation Research Record: Journal of the Transportation Research Board, 1903(1):35–42, 2005.

2. Munzner, T.: Visualization Analysis and Design. CRC Press, 2014.

3. Garland, K.: Mr Beck's Underground Map. Capital Transport, 1994.

4. Google maps. `https://www.google.com/maps`. Accessed: 2015-02-22.

5. Cta train tracker map (beta). `http://www.transitchicago.com/traintrackermap/`. Accessed: 2015-04-11.

6. Trip planner. `http://www.goroo.com/index.htm`. Accessed: 2015-04-11.

7. Do-it-yourself transit info display (beta). `http://www.transitchicago.com/developers/diydisplay.aspx`. Accessed: 2015-02-22.

8. Cta gtfs page. `http://www.transitchicago.com/developers/gtfs.aspx`. Accessed: 2015-03-15.

9. Isochrone maps. `http://en.wikipedia.org/wiki/Isochrone_map`. Accessed: 2015-02-19.

10. Widener, M. J., Farber, S., Neutens, T., and Horner, M.: Spatiotemporal accessibility to supermarkets using public transit: an interaction potential approach in cincinnati, ohio. Journal of Transport Geography, 42:72–83, 2015.

11. O'Sullivan, D., Morrison, A., and Shearer, J.: Using desktop gis for the investigation of accessibility by public transport: an isochrone approach. International Journal of Geographical Information Science, 14(1):85–104, 2000.

12. Zeng, W., Fu, C.-W., Arisona, S. M., Erath, A., and Qu, H.: Visualizing mobility of public transportation system. 2014.

CITED LITERATURE (continued)

13. Clark, J. W.: Time-distance transformations of transportation networks. Geographical Analysis, 9(2):195–205, 1977.

14. Wang, Y.-S. and Chi, M.-T.: Focus+ context metro maps. Visualization and Computer Graphics, IEEE Transactions on, 17(12):2528–2535, 2011.

15. Haunert, J.-H. and Sering, L.: Drawing road networks with focus regions. Visualization and Computer Graphics, IEEE Transactions on, 17(12):2555–2562, 2011.

16. Andrienko, G., Andrienko, N., Bak, P., Keim, D., and Wrobel, S.: Visual analytics of movement. Springer Science & Business Media, 2013.

17. Andrienko, N. and Andrienko, G.: Visual analytics of movement: An overview of methods, tools and procedures. Information Visualization, page 1473871612457601, 2012.

18. Zhong, C., Wang, T., Zeng, W., and Arisona, S. M.: Spatiotemporal visualisation: A survey and outlook. In Digital Urban Modeling and Simulation, pages 299–317. Springer, 2012.

19. Goncalves, T., Afonso, A. P., and Martins, B.: Visual analysis of mobility data. In Mobile Data Management (MDM), 2013 IEEE 14th International Conference on, volume 2, pages 7–10. IEEE, 2013.

20. Nguyen, H. T., Duong, C. K. T., Bui, T. T., and Tran, P. V.: Visualization of spatio-temporal data of bus trips. In Control, Automation and Information Sciences (ICCAIS), 2012 International Conference on, pages 392–397. IEEE, 2012.

21. Bast, H., Brosi, P., and Storandt, S.: Travic: A visualization client for public transit data.

22. Nagel, T. and Groß, B.: Shanghai metro flow-multiple perspectives into a subway system.

23. Janetzko, H., Jäckle, D., Deussen, O., and Keim, D. A.: Visual abstraction of complex motion patterns. In IS&T/SPIE Electronic Imaging, pages 90170J–90170J. International Society for Optics and Photonics, 2013.

24. Tobler, W. R.: Experiments in migration mapping by computer. The American Cartographer, 14(2):155–163, 1987.

CITED LITERATURE (continued)

25. Phan, D., Xiao, L., Yeh, R., and Hanrahan, P.: Flow map layout. In Information Visualization, 2005. INFOVIS 2005. IEEE Symposium on, pages 219–224. IEEE, 2005.

26. Buchin, K., Speckmann, B., and Verbeek, K.: Flow map layout via spiral trees. Visualization and Computer Graphics, IEEE Transactions on, 17(12):2536–2544, 2011.

27. Ware, C.: Information visualization: perception for design. Elsevier, 2012.

28. Lynch, K.: The image of the city, volume 11. MIT press, 1960.

29. Muhlhausen, J.: Wayfinding is not signage. Signs of the Times, 2006.

30. Siegel, A. W. and White, S. H.: The development of spatial representations of large-scale environments. Advances in child development and behavior, 10:9, 1975.

31. Thorndyke, P. W. and Hayes-Roth, B.: Differences in spatial knowledge acquired from maps and navigation. Cognitive psychology, 14(4):560–589, 1982.

32. Colle, H. A. and Reid, G. B.: The room effect: Metric spatial knowledge of local and separated regions. Presence: Teleoperators and virtual environments, 7(2):116–128, 1998.

33. Darken, R. P., Allard, T., and Achille, L. B.: Spatial orientation and wayfinding in large-scale virtual spaces: An introduction. Presence: Teleoperators and Virtual Environments, 7(2):101–107, 1998.

34. Kosslyn, S. M.: Seeing and imagining in the cerebral hemispheres: a computational approach. Psychological review, 94(2):148, 1987.

35. Golledge, R. G.: Wayfinding behavior: Cognitive mapping and other spatial processes. JHU Press, 1999.

36. Raubal, M. and Winter, S.: Enriching wayfinding instructions with local landmarks. Springer, 2002.

37. Elias, B.: Extracting landmarks with data mining methods. In Spatial information theory. Foundations of geographic information science, pages 375–389. Springer, 2003.

CITED LITERATURE (continued)

38. Elias, B. and Brenner, C.: Automatic generation and application of landmarks in navigation data sets. In Developments in spatial data handling, pages 469–480. Springer, 2005.

39. Elias, B. and Sester, M.: Landmarks für routenbeschreibungen. GI-Technologien für Verkehr und Logistik, Hrsg. Jörn Möltgen und Andreas Wytzisk, IfGIprints, 13:375–394, 2002.

40. Gtfs. `https://developers.google.com/transit/gtfs/reference`. Accessed: 2015-03-15.

41. Gtfs-realtime. `https://developers.google.com/transit/gtfs-realtime/`. Accessed: 2015-04-12.

42. Openstreet map. `http://www.openstreetmap.org`. Accessed: 2015-02-19.

43. Thudt, A., Hinrichs, U., and Carpendale, S.: The bohemian bookshelf: supporting serendipitous book discoveries through information visualization. In Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, pages 1461–1470. ACM, 2012.

44. Shneiderman, B.: The eyes have it: A task by data type taxonomy for information visualizations. In Visual Languages, 1996. Proceedings., IEEE Symposium on, pages 336–343. IEEE, 1996.

45. Webgl. `https://www.khronos.org/webgl/`. Accessed: 2015-04-6.

46. mapbox-gl.js. `https://www.mapbox.com/mapbox-gl/`. Accessed: 2015-04-6.

47. Three.js. `http://threejs.org/`. Accessed: 2015-04-6.

48. Rohlf, J. and Helman, J.: Iris performer: a high performance multiprocessing toolkit for real-time 3d graphics. In Proceedings of the 21st annual conference on Computer graphics and interactive techniques, pages 381–394. ACM, 1994.

49. Behrendt, S., Colditz, C., Franzke, O., Kopf, J., and Deussen, O.: Realistic real-time rendering of landscapes using billboard clouds. In Computer Graphics Forum, volume 24, pages 507–516. Wiley Online Library, 2005.

CITED LITERATURE (continued)

50. Finkel, R. A. and Bentley, J. L.: Quad trees a data structure for retrieval on composite keys. Acta informatica, 4(1):1–9, 1974.

51. Shaffer, C. A., Samet, H., and Nelson, R. C.: Quilt: a geographic information system based on quadtrees. International Journal of Geographical Information System, 4(2):103–131, 1990.

VITA

Education

Bachelor of Science Degree in Engineering of Computing Systems
Politecnico di Milano
2010 - 2013

Master of Science Degree in Computer Science
University of Illinois at Chicago, Chicago, IL
2014 - 2015

Master of Science Degree in Engineering of Computing Systems
Politecnico di Milano
2013 - 2015

Master thesis

TransitTrace: An Exploration of Visual Encoding for Effective Route Planning using an Ambient Display
Angus G. Forbes, Professor (for reference, contact: `aforbes@uic.edu`)
Topics: Visualization, Route planning

Publications

BlueSentinel: a first approach using iBeacon for an energy efficient occupancy detection system
Conte, G., De Marchi, M., Nacci, A. A., Rana, V., Sciuto, D.
Publisher: ACM New York, NY, USA

Technological skills

Languages
Objective-C, Javascript, HTML, CSS, C, C++, Java
Technologies
D3, WebGL, iOS, Arduino, iBeacon
Tools
LaTeX, Git, Matlab
Databases
SQL family
OSs
Mac OS, Windows and Linux

VITA (continued)

Working Experience

National Science Foundation / University of Illinois
Research Assistant
Spring 2015
Employed in NSF CREDIT (CybeR-Enabled Demand- Interactive Transit) project.
The project investigates theories and tools to characterize, analyze, operationalize, and evaluate a hybrid system that integrates conventional fixed-route service with demand-interactive service.
Skills and technologies used:

- Javascript
- WebGL (THREE.js)
- Maps (MapBox GL)
- Node.js

inkOfPixel
Co-Founder and Software Engineer
2012-present
Develop Calc 2M, iOS application with over 400,000 users.
Skills and technologies used:

- Objective-C
- Designed a grammar for a custom math engine

Bruroma s.r.l.
Draftsman
Summer 2008
Design machines for medical and confectionary corporates.
Skills and technologies used:

- SolidEdge CAD 3D
- TruTops CAD 2D

BMP
Bricklayer
Summer 2007