11-13-2015

Online Moving Object Visualization with Geo-Referenced Data

Guangqiang Zhao
zgq624@gmail.com

DOI: 10.25148/etd.FIDC000741

FLORIDA INTERNATIONAL UNIVERSITY

Miami, Florida

ONLINE MOVING OBJECT VISUALIZATION WITH GEO-REFERENCED DATA

A dissertation submitted in partial fulfillment of the

requirements for the degree of

DOCTOR OF PHILOSOPHY

in

COMPUTER SCIENCE

by

Guangqiang Zhao

2016

To: Interim Dean Ranu Jung
College of Engineering and Computing

This dissertation, written by Guangqiang Zhao, and entitled Online Moving Object Visualization with Geo-Referenced Data, having been approved in respect to style and intellectual content, is referred to you for judgment.

We have read this dissertation and recommend that it be approved.

Tao Li

Masoud Sadjadi

Armando Barreto

Naphtali Rishe, Major Professor

Date of Defense: November 13, 2015

The dissertation of Guangqiang Zhao is approved.

Interim Dean Ranu Jung
College of Engineering and Computing

Andrés G. Gil
Vice President for Research and Economic Development
Dean of the University Graduate School

Florida International University, 2016

DEDICATION

To my beloved parents.

To my dear wife.

ACKNOWLEDGMENTS

I would like to gratefully acknowledge my advising professor Dr. Naphtali Rishe, who gives me the opportunity to do research in the High Performance Database Research Center (HPDRC) and guides me throughout my PhD study.

I am exremely thankful to my committee members, Professor Tao Li, Professor Masoud Sadjadi, and Professor Armando Barreto, for their continuous support and insightful comments.

I would like to say thank you to all my colleagues working in the HPDRC team, for their technical and spiritual support throughout all these years.

Finally, I would like to thank my parents and wife, for their efforts to give me the best education opportunity as well as love.

ABSTRACT OF THE DISSERTATION

ONLINE MOVING OBJECT

VISUALIZATION WITH GEO-REFERENCED DATA

by

Guangqiang Zhao

Florida International University, 2016

Miami, Florida

Professor Naphtali Rishe, Major Professor

As a result of the rapid evolution of smart mobile devices and the wide application of satellite-based positioning devices, the moving object database (MOD) has become a hot research topic in recent years. The moving objects generate a large amount of geo-referenced data in different types, such as videos, audios, images and sensor logs. In order to better analyze and utilize the data, it is useful and necessary to visualize the data on a map.

With the rise of web mapping, visualizing the moving object and geo-referenced data has never been so easy. While displaying the trajectory of a moving object is a mature technology, there is little research on visualizing both the location and data of the moving objects in a synchronized manner.

This dissertation proposes a general moving object visualization model to address the above problem. This model divides the spatial data visualization systems into four categories. Another contribution of this dissertation is to provide a framework, which deals with all these visualization tasks with synchronization control in mind. This platform relies on the TerraFly web mapping system.

To evaluate the universality and effectiveness of the proposed framework, this dissertation presents four visualization systems to deal with a variety of situations and different data types.

TABLE OF CONTENTS

CHAPTER PAGE

LIST OF TABLES

TABLE PAGE

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

1.1 Background and Motivation

In the past decade, the World Wide Web (WWW) and accurate positioning systems, e.g. Global Positioning System (GPS), have greatly influenced the research in the Geographic Information Systems (GISs). Moreover, micro positioning chips can be found in more and more devices such as mobile phones and tablets in the last few years. These evolving technologies have caused a surge in the demand for the Moving Object Database (MOD). The MOD is a spatio-temporal database that manages objects with continuously changing locations over time. The MOD has many applications in areas such as traffic monitoring, fire rescue and police car dispatch, air traffic control, social network analysis, homeland security, and meteorology monitoring, just to name a few.

Data is the fundamental building block in the age of big data. The data collected in the real world usually contains a temporal component, and nowadays at least 60% of the data is geospatially referenced[30]. The data sources include both the consumer and industry products with positioning abilities, such as mobile phones, cameras, camcorders, Unmanned Aerial Vehicles (UAVs) and robots. Besides the location information, they also generate rich data types including photos, audios, videos and numeric values. While some of the data is collected from the fixed equipment, more data is generated from the moving objects now.

With the rapid development of online GIS services, web-based map applications have become popular in everyday life. It is a natural fit to display georeferenced data on a map. Many existing online map services include geotagged photos and videos of certain places[58]. While collaborative and crowdsourcing maps became popular, more users tend to upload and share their georeferenced data online[23].

1.2 Problem Definition

The dissertation mainly focuses on three major fields: (1) moving object database, (2) web mapping and (3) georeferenced data. The following subsections shows each topic in detail.

1.2.1 Moving Object Database

The research of the moving object database focuses on three major directions: modeling, indexing and querying.

Location modeling is the foundation of MOD research. Most existing studies for MOD indexing assume a linear movement model.

Indexes are important for database systems since they can greatly improve the data querying speed. Indexing is even more important for spatial database systems due to high query intensity. There are many existing indexes for the spatial database, and the most famous ones are Grid [68], Quadtree [21], R-tree [29] and their variants.

Unlike traditional databases, in which data remains constant most of the time, MODs are facing queries on a large number of continuously moving objects. Figure 1.1 shows one-hour trajectory data for vessels leaving the Miami port. There are about 200 vessel position updates per second for this application. As such, traditional static indexes are no longer efficient due to huge update costs and poor query performances. New indexes that are capable of dealing with frequent updates are becoming increasingly important as well as necessary for MODs.

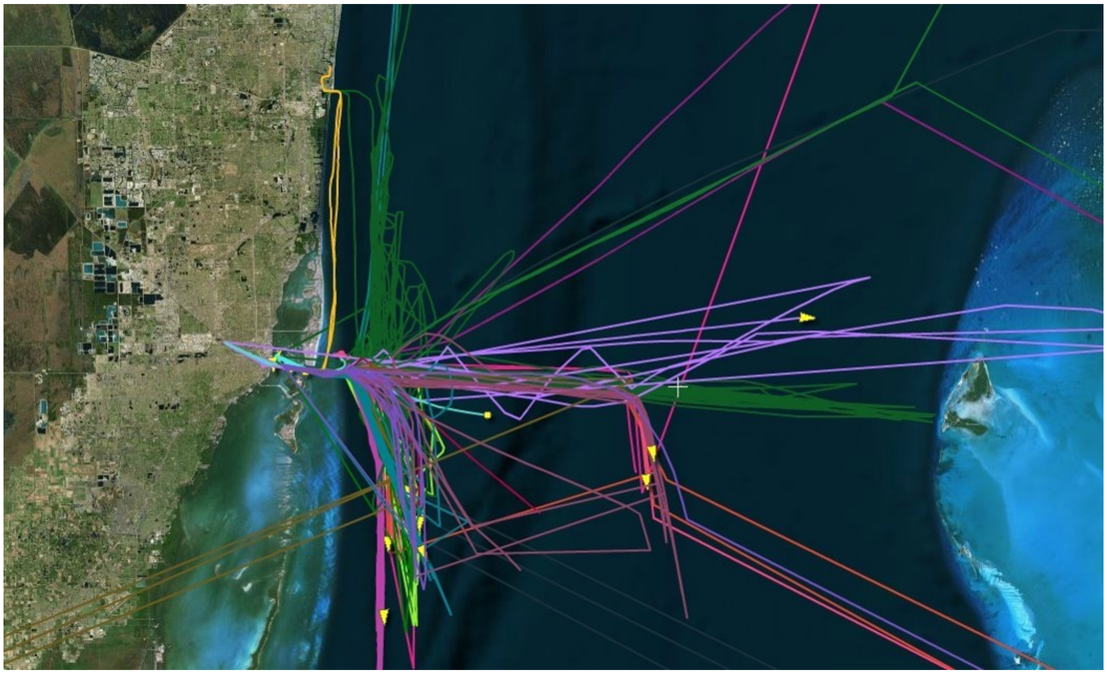


Figure 1.1: Vessel Trajectory of Miami Port

Besides normal database queries, MODs are designed to support some spatio-temporal query types and may be executed on past, current or future time data, e.g. range queries get all the cars inside on a college within 2 hours, k-nearest neighbor (KNN) queries find the 5 closest gas stations from a moving car, or trajectory queries retrieve the path of the helicopter in the last hour [67].

Web mapping is used widely nowadays and users are accustomed to viewing an online map in the web browser. Most of the online map services use the tile system and retrieve information by grids, e.g. Google Map [76] and Bing Map [86]. Efficient querying and smoothly showing moving objects through the internet can be a big challenge. This is another consideration while designing the MOD indexes.

1.2.2 Web Mapping

Unlike the traditional desktop mapping applications, web mapping is based on the client-server structure. A map server is responsible for producing, storing and serving the map data, while the client side application displays the map to the end user.

Many spatial databases are used on the map server such as PostGIS, MySQL, Oracle Spatial and Graph, Microsoft SQL Server and IBM DB2. All of them have implemented the Simple Feature Access[1], which is a storage and access model for the geographical data. The model also defines Well-Known Text (WKT) to describe the vector geometry objects like point, linestring, polygon, etc.

Common protocols for delivering map data involve Web Map Service (WMS)[2], Tile Map Service (TMS)[3], Web Feature Service (WFS)[4] and Web Coverage Service (WCS)[5]. The most famous web mapping server platforms are MapServer[6] and GeoServer[7]. For those who do not want to deploy their own servers, they have the choice to use the public web map services including Google Maps, Bing Maps, MapQuest, OpenStreetMap (OSM)[8] and Here.

[1] http://www.opengeospatial.org/standards/sfa

[2] http://www.opengeospatial.org/standards/wms

[3] http://wiki.osgeo.org/wiki/Tile_Map_Service_Specification

[4] http://www.opengeospatial.org/standards/wfs

[5] http://www.opengeospatial.org/standards/wcs

[6] http://mapserver.org/

[7] http://geoserver.org/

[8] https://www.openstreetmap.org/

While some dedicated client software exists like Google Earth, most client side applications use the web browser to display the map. With the rapid development of Internet technology, especially the JavaScript and HTML5, the functionality of browser-based mapping applications has been able to match up with the desktop software. Most commercial web mapping application program interfaces (APIs) are not free or have some limitations (e.g. Google Maps, Bing Maps, Yahoo! Maps), whereas more opensource applications are coming out, such as the heavyweight OpenLayers and the lightweight Leaflet[89].

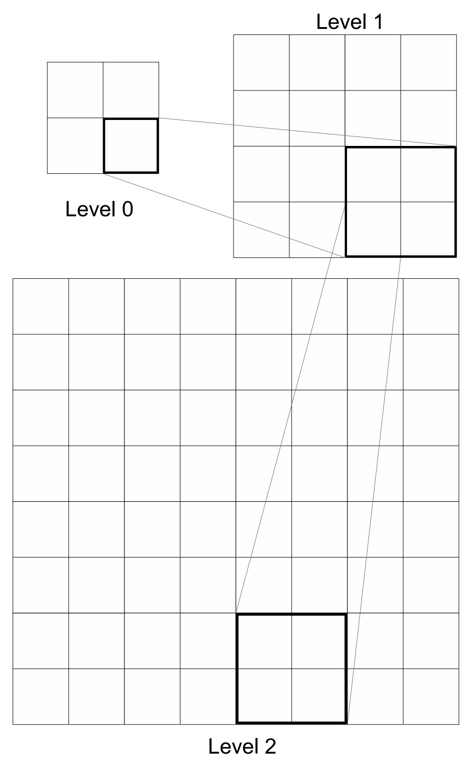


Figure 1.2: TerraFly Level of Detail Tile System

The most popular way for serving map data to the client browser is using the tile system. Map features are pre-selected for different level of detail using styling rules, and then the map of each level is cut into tiles for efficient transferring and display. A sample tile system is shown in Figure 1.2. The tile can be either a pre-rendered image or pre-defined vector data, however the vector tile system is gradually replacing the image tile system [26].

1.2.3 Georeferenced Data

To describe the process of linking an object with a specific location, three words are frequently used: geocoding, georeferencing and geotagging.

Geocoding is the process of finding geographic coordinates using the address or place name. The opposite operation is called reverse geocoding. It is a query performed by the geocoder software based on certain GIS database.

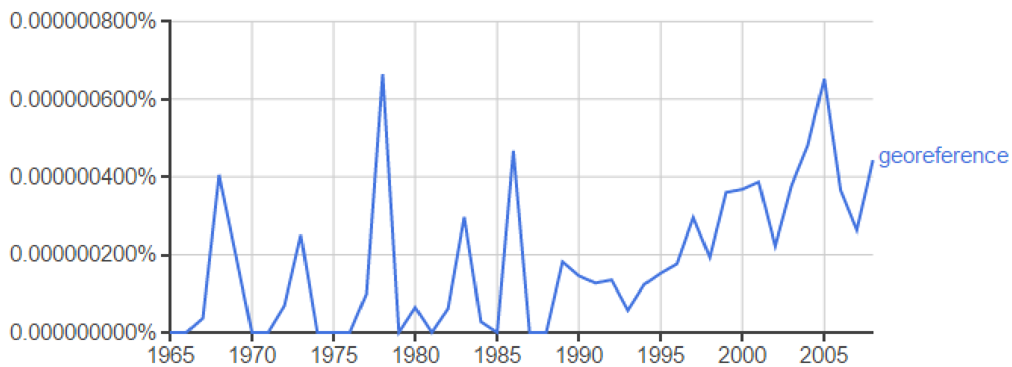


Figure 1.3: Georeference occurred in books.

Source: Google Ngram Viewer

Georeferencing and geotagging have similar definitions. Georeferenced is commonly used in GIS field to define the process to associate an object with a spatial location[5], and it has a long usage history since 1960s, as shown in Figure 1.3.

Geotagging has been more used by general public since 2005 to describe the operation of giving a geotag (geographical location) to a media (usually a photograph or a video), as shown in Figure 1.4.

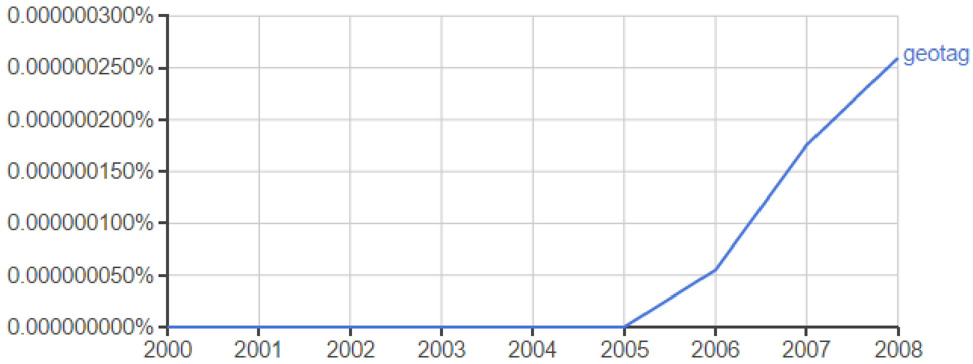


Figure 1.4: Geotag occurred in books.

Source: Google Ngram Viewer

With the popularity of the camera phones, as well as the increasing prevalence of built-in GPS on camera and camcorders, most digital photos and videos taken these days are geotagged automatically. This explains why "geotag" has gained more search interest than "georeference" since 2007, as shown in Figure 1.5.

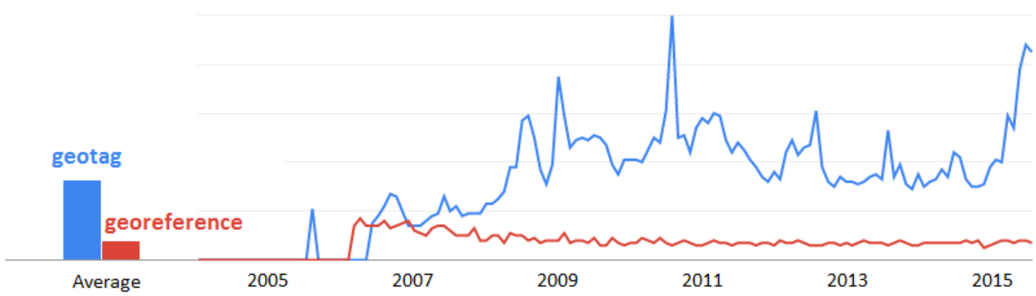


Figure 1.5: Search interest of Georeference and Geotag over time.

Source: Google Trends

While georeference and geotag have much the same meaning for photo, they can be different when referring to videos. Most consumer camcorders only tag one location for each video. In contrast, a dashboard camera (dashcam) can log all the waypoints of a car passing through while recording the video. In order to eliminate ambiguity, the word geotag is used to represent the former case and the word georeference is used to represent the latter case in this dissertation.

1.3 Contributions

The main contributions of the dissertation are shown as follows:

1.3.1 Spatial-temporal Data Visualization Model

This dissertation proposes a model for visualization of the moving object with georeferenced data. Based on this model, the dissertation classifies moving object visualization systems into four categories. This model also reveals the mapping operation between the moving object and the related data.

1.3.2 Abstract Map Layer

Based on the proposed model, the dissertation defines an abstract map layer, which contains the necessary functions and events for most moving object visualization tasks.

1.3.3 Moving Object Visualization Framework

Finally, the dissertation proposes a general framework, which is suitable for visualizing the georeferenced data generated by the moving object.

1.4 Dissertation Outline

The rest of the dissertation is organized as follows: Chapter 2 gives the literature review. Chapter 3 discusses modeling of the moving object and georeferenced data, the abstract map layer and the framework. Chapter 4 proposes a way to accelerate the map matching algorithm. A dashboard camera video share platform is shown in Chapter 5, and an aircraft video visualization system is discussed in Chapter 6. Chapter 7 describes how to design a virtual guidance system using georeferenced data. Chapter 8 provides an example of visualizing big data. Finally, Chapter 9 summarizes the conclusions and discusses future directions.

CHAPTER 2

LITERATURE REVIEW

This chapter reviews the literature in two area: the moving object database and spatial data visualization.

2.1 Moving Object Database

The study of moving object database emerged in the late last century; that is when the civilian GPS became popular. The research of moving object database focuses on three major directions: modeling, indexing and querying.

2.1.1 Modeling

Location modeling is the foundation of the MOD research. Most existing studies for the MOD indexing assume a linear movement model. Wolfson et al. first proposed a moving objects spatio-temporal (MOST) model, which represents the location of the moving object as a dynamic attribute [99].

Whereas most of the studies employ a linear model, some more complex movement models have been proposed. Tao et al. [90] proposed the concept of recursive motion functions to support arbitrary motion patterns. This TPR-tree [83] based model can predict the real-world motion more accurately. Jeung et al. [42] give a hybrid based prediction model, which discovers the model from the recent trajectories of an object.

In order to represent the real world in special conditions, models based on constraints (e.g. road network [92]) have also been proposed.

2.1.2 Indexing

MOD indexing is a well-studied research area. [67] illustrates the evolution of index structures in the past 20 years.

Existing studies can be classified into two categories. The first category focuses on processing queries based on their current positions and future predictions. The second one tries to represent and retrieve historical trajectories. Recent studies have proposed new ways, which can process data at all times instead of having separate indexing structures for past, present or future data [53] [19].

Moving object indexes can also be divided into space partitioning indexes and data partitioning indexes. The former utilizes multidimensional index structures (e.g. TPR-tree [83]), typically the R-tree and its variants. They use the Minimum Bounding Rectangle (MBR) to hold objects that are close to each other in 2-d or 3-d dimensions. The MBR is then organized into a tree to keep minimum overlapping. R-tree based Indexes are less susceptible to data density, but do not perform well in a concurrent environment [83].

The latter approach is to partition the entire space into grids, usually based on the B+-tree or grid index [40] [100]. The objects will be indexed into pre-partitioned space cells. The B+-tree or grid based indexes are widely used in commercial databases, so these kinds of indexes can be easily integrated into an existing DBMS (Database Management System). The disadvantage is that the imbalance of space partitioning will sacrifice the performance. To address this problem, the STRIPES index is proposed which is derived from the quad-tree [71].

While most indexes utilize the unconstraint model (e.g. MOST), some moving object indexes for special environments are proposed in recent years, e.g., road networks [13] or indoor networks. These indexes are more close to real world applications.

2.1.3 Benchmark

Benchmarks have been proven a useful tool to evaluate the performance of DBMSs. There are several benchmarks available for MOD now, e.g. the COST benchmark [41]. Some benchmarks are based on real road network environments, e.g. the BerlinMOD benchmark [15].

2.2 Pathfinding and Map Matching

2.2.1 Pathfinding

Pathfinding is to find the route between two points. The desired route can either be shortest or good enough based on the requirements of different applications. Pathfinding algorithm is one of the fundamental elements of many map matching algorithms.

The most famous and basic one is Dijkstra's algorithm, which finds a shortest path on a weighted graph [14]. Its variant, A* algorithm, is commonly used in real world application. A* uses the heuristic function to continuously estimate the approximate distance between current node and the finish [33]. Although there is no guarantee to find the optimal path, A* is much faster than Dijkstra's algorithm because it examines fewer nodes. An improvement of A* algorithm is using bidirectional search, which will further reduce the search space [38].

Since the Map Matching task only focus on pathfinding in limited local space, we do not introduce algorithms for long-distance navigation here.

2.2.2 Map Matching

Map matching is the procedure of mapping a series of geographic coordinates that recorded using a positioning device to the real path it is on. This kind of algorithms have a long history since the emergence of the satellite navigation system. [75] suggests dividing them into four categories: geometric, topological, probabilistic and others.

Most practical algorithms use the hidden Markov model (HMM) to match track data on road network [66] [27]. This kind of methods treat the recorded coordinates as HMM observations and candidate road segments as HMM states. Then the emission probability is determined by the distance between a observation point and one of its candidate points. The transition probability is how likely the object would travel between candidate points, which is calculated based on the pathfinding algorithm.

2.3 Online Moving Object Visualization

There are two kinds of the visualization targets, including the trajectory and the georeferenced data.

2.3.1 Web Mapping

Web mapping has a long history since the invention of the WWW [65] [25] [47]. [51] argued that one should use web mapping instead of internet mapping when the browser and hyper-text markup language (HTML) are involved. Web maps can be categorized into four types [46], as shown in Table 2.1. The earliest online map is static. It can be as easy as putting the image of a scanned paper map online. The first interactive map "The World Wide Earthquake Locator" was born in 1994 [1].

Table 2.1: Types of web maps.

Static View Only	Dynamic View Only
Static Interactive	Dynamic Interactive

The online interactive maps are built using three different techniques:

- Java applet (introduced in March 1995)
- JavaScript (released in December 1995)
- Flash (published in 1996)

The Java applet and Flash need extra plug-ins to be installed in the browser. Major web mapping providers tend to use JavaScript since it is powerful and natively supported by most browsers [93].

The emergence of many new technologies greatly promoted the development of online maps. For example: HTML5 supports latest multimedia content and provides better map interactivity [91] [59]. Canvas and SVG make vector maps possible [26]. WebGL can generate three-dimensional map [20].

The latest trends of web mapping include open source mapping [89] [62] [7], collaborative mapping (crowdsourcing) [64], mapping mashups [60] [2] and mapping on mobile devices [63].

2.3.2 Mapping Moving Object

Mapping moving object is quite different from drawing static features. Animation is not required even if the map itself is dynamic and interactive [34].

Animation is important in the GIS system. One kind of animation is for depicting the geographic change [31]. For example, [32] visualizes global weather, [52] use animation for sea ice monitoring and [18] use animation for flood simulation. Another kind is for showing the movement of physical entities [9]. These animations can be either offline or near real-time.

This dissertation focuses on showing moving objects on the map. [43] proposes a method to move a marker on the map. [84] further illustrates a system to map "path-enhanced" multimedia. There is also research on showing the tracks on the map [16].

The above methods do not reveal the deep links between the moving objects and their related data. To address this problem, this dissertation finds new ways of showing georeferenced data on map with better user experience. Two systems have been proposed in the author of this dissertation's previous work [103] [104].

CHAPTER 3

MOVING OBJECT VISUALIZATION FRAMEWORK

In this chapter, a general data model is given to describe the moving object along with the georeferenced data. A framework is proposed based on this model.

3.1 Moving Object Visualization Model

3.1.1 Spatial-temporal Data Types

In traditional geo-temporal database, a data item can be represented as a tuple $\langle t, l, d \rangle$, which includes three parameters: time t, location l, and data attributes d. In moving object database, both the location and data attributes can either be fixed or vary over time.

In practical applications, the location and data attributes are stored separately and coupled by time t since they have different storage strategies and update frequencies. There are four combinations in general: $(t, l_f) \sim (t, d_f)$, $(t, l_f) \sim (t, d_v)$, $(t, l_v) \sim (t, d_f)$, and $(t, l_v) \sim (t, d_v)$. This dissertation uses symbol $\sim$ to represent coupling. The subscript f stands for fixed and v represents variable value over time. The visualization environment is set to have two components: the map view and the data view. The location l_f or l_v is visualized inside the map view while the data d_f or d_v is visualized inside the data view.

The data item is archived after every certain time interval t_i. Here, L is defined as archived positions of l_v during a historical time range from t_s to t_e, and f_l is a function to get a historical location based on a given time t:

$$L = f_l(t) \qquad t_s \leq t \leq t_e \tag{3.1}$$

Similarly, D is defined as archived data of d_v during the same historical time range and f_d is a function to get the historical data attributes based on a given time t:

$$D = f_d(t) \qquad t_s \leq t \leq t_e \tag{3.2}$$

Note that the function f_l and f_d can be either continuous or discrete. After mixing all possibilities together, there are four types of combinations, as shown in Table 3.1.

Table 3.1: Geo-temporal data archiving and visualization.

	Live data item	Archived data item	Map view	Data view
(1)	(t, l_f, d_f)	(t, l_f, d_f)	Point	Static Data
(2)	(t, l_f, d_v)	(t, l_f, D)	Point	Dynamic Data
(3)	(t, l_v, d_f)	(t, L, d_f)	Polyline	Static Data
(4)	(t, l_v, d_v)	(t, L, D)	Polyline	Dynamic Data

Type (1) and (2) have fixed locations with static or dynamic data attributes. The geotagging based online photo and video sharing services fall into these two categories. Since the first two types have been well studied and widely used, this dissertation focuses on the last two situations.

Type (3) is purely used for showing the trajectory of a moving object, which is annotated by some static data attributes. The following are some scenarios of type (3):

- The air traffic monitoring system, which shows all routes of the aircrafts.
- The vessel tracking system using Automatic Identification System (AIS).

In type (4), both location and data attributes are the functions of time. In this situation, the georeferenced data is collected by the moving object in a given sampling rate. Here are some typical scenarios of type (4):

- A georeferenced video shot by a car dashboard camera.
- A guidance audio recorded by a guide walking in the park.
- A barometric sensor log file collected by a climber while mountaineering.

3.1.2 Mapping Operations

For better visualization experience, the location and data attributes should be synchronized at any given time t. The first step is converting the discrete function to a continuous function. There are many conversion methods that exist and the selection is data dependent, so this topic will not be carried out in detail. In this article, it is assumed that f_l and f_d are continuous functions, either native or converted.

A common operation for the synchronization task is mapping. For example, given a location l, find out the related data attributes. As shown in equation 3.1, f_l is a bijective function. Because for any $t \in [t_s, t_e]$, there is one and only one paired location. Similarly, f_d is also a bijective function, as shwon in 3.2. Therefore, the mapping functions between location l and data attribute d can be drawn as follows:

$$d = f_d(f_l^{-1}(l)) \qquad d \in D, l \in L \tag{3.3}$$

$$l = f_l(f_d^{-1}(d)) \qquad d \in D, l \in L \tag{3.4}$$

3.2 Abstract Map Tier

Although there are many web-based map APIs that exist in the market, they have similar functions and features. For example, almost all the map APIs have a function to add a marker on the map. In order to avoid putting too much attention into the details of implementation of different APIs, this dissertation proposes an abstract map tier with all necessary functions and features for the moving object visualization tasks.

Figure 3.1 shows the minimal required structure for drawing features on a web map. They are object-oriented and created in top-down order. A map can have multiple layers, and a layer can have any number of features including points, polylines and polygons.

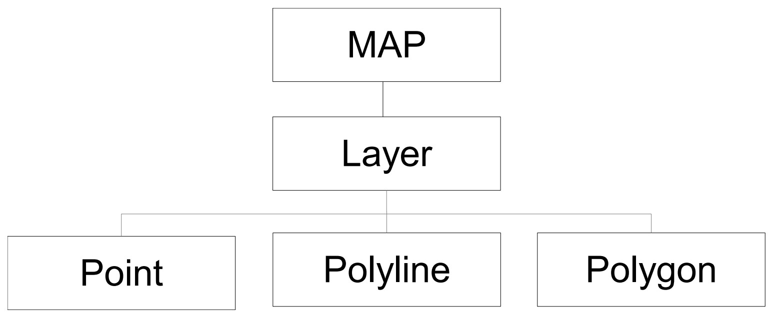


Figure 3.1: Minimal map drawing structure.

The core functions of a web-based map API can be divided into two categories: operations and events. Table 3.2 and 3.3 contain these major functions.

Table 3.2: Online map operations.

Object	Operation	Description
Map	ZoomTo	Zoom the map to a given resolution.
Map	MoveTo	Move the map center to a given coordinate.
Map	AddLayer	Add a layer to the map.
Map	RemoveLayer	Remove a Layer from the map.
Layer	AddPoint	Add a point to the layer.
Layer	AddPolyline	Add a polyline to the layer.
Layer	AddPolygon	Add a polygon to the layer.
Layer	RemoveObject	Remove an object from the layer.
Point/polyline/polygon	MoveTo	Move the position of an object.
Polyline/polygon	Modify	Modify the waypoint of a multi-waypoint object.

Table 3.3: Online map events.

Object	Events	Description
Map	OnZoom	When user zooms the map.
Map	OnMove	When user moves the map.
Map/point/polyline/polygon	OnClick	When user clicks on an object.

Similarly, the data view also has some major operations and events, shown in Table 3.4 and 3.5 separately.

Table 3.4: Data operations.

Operation	Description
WriteData	Add a data item to then end of a data series.
ReadData	Get the value of a data item for a given timestamp.

Table 3.5: Data events.

Operation	Description
OnDataItemClicked	When user clicks on a data item.

Notice that location and data are updated in different frequencies. Most positioning devices refresh every second, but data update frequency can be quite different. For example, a typical online video plays at 30 frames per second while an altitude sensor can refresh at 5Hz. To better synchronize the visualization of map and the data, two different updating strategies are used here:

- Map-driven update - When the map is refreshing faster, the data updating will be triggered in one of the map updating events.
- Data-driven update - When data is refreshing faster, the map will be updated in one of the data updating events.

3.3 Moving Object Visualization Framework

The web-mapping framework is designed using well-known three-tier website architecture based on the client-server architecture model. All the parts are separated into different modules to increase reusability, as shown in Figure 3.2. The client side holds presentation tier. The server side contains the logic tier and data tier.

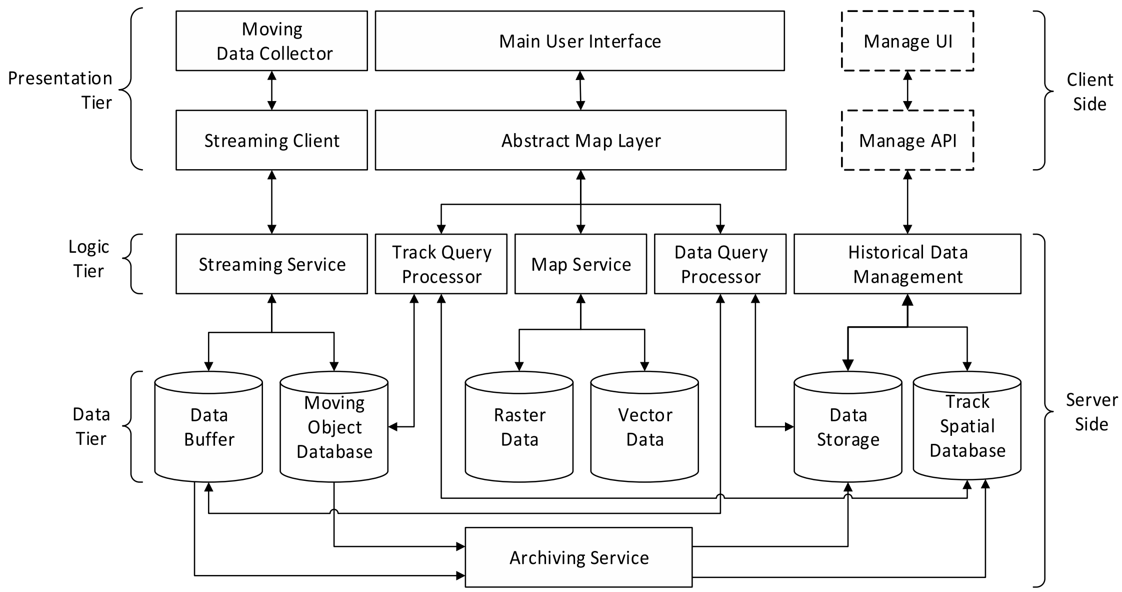


Figure 3.2: Moving object visualization framework.

3.3.1 The Client Side

The main user interface on the client side communicates with the server and shows the map and data to the end user. The abstract map layer is used here to handle the visualization and synchronization of the trajectory and the data.

The moving data collector and the streaming client work as an application on the user's mobile device. The collector gets the business data in real time and calls the streaming client to deliver the static and location data to the server.

The remote management module is optional, since some sites prefer to manage the data locally on the server.

3.3.2 The Server Side

The server side manages three kinds of data: the real-time moving object data, the historical moving object data, and the static base map data. The moving object data can be further divided into two parts: the MOD database is used to store location, and the regular database is used to store other temporal data.

The logic tier contains the business logic to handle the data. The query processor handles general queries from the presentation tier, such as "get all the tracks inside a given polygon" or "return current location of a given object". The query result from the data tier will be delivered to the abstract map layer for further processing.

The data tier executes the CRUD operations (Create, Read, Update and Delete) for the logic tier. There will be at least six dedicated databases to support the whole system. Three kinds of database systems will be used here. The raster and vector data is stored in the spatial database. The real-time and historical location data is stored in the moving object database. The collected business data and other static data of the moving objects is stored in the normal relational database.

3.3.3 More on Archiver

The archiver is a background service that periodically stores the real-time data from the temporary buffer to the historical database. The interval between archiving is based on the data update frequency, the data type and the number of mobile devices. Discrete sensor data is buffered and archived in the same way as location data, so they can be bundled together for archiving. The continuous streaming data is different. For example, video data has its own buffering mechanism that must be executed independently.

3.4 Summary

This chapter proposes a data model with four different visualization types. This dissertation will show an example of type (3) in Chapter 5, and three systems of type (4) in Chapters 4, 6 and 7. The system in Chapter 7 is a real-time application and Chapter 6 introduces a system in a constrained road network.

CHAPTER 4

VEHICLE TRAJECTORY MAP MATCHING FOR VISUALIZATION

4.1 Background and Motivation

Vehicle trajectory map matching is to match the real world GPS tracking data to a given road network model. Map matching widely used in the vehicle navigation system to locate the road segment that a vehicle is driven on. It is also helpful to fix the inaccurate trajectory due to measurement errors.

There are many map matching algorithms exist for different applications. [75] suggests dividing them into four categories: geometric, topological, probabilistic and others. In practice, probabilistic methods have better accuracy and efficiency [66] [54] [27].

Map matching is an important part in the moving object visualization system. Most current algorithms only match the whole trajectory to a road network instead of individual points. It is sufficient for applications on the route level positioning. However, it lacks the ability to precisely recover the vehicle location for each time stamp in the moving object database. Moreover, existing algorithms are not ideal in real world data in terms of execution time and fault tolerance. This article presents a solution to address above problems.

4.2 Problem Definition

Figure 4.1 shows the map matching problem. The green dots are actual GPS tracking points. The black line that connecting them forms a GPS track. One can easily conclude it is on a wrong road segment through common sense. This type of mismatch which comes from GPS error is very common. The green line segment in the

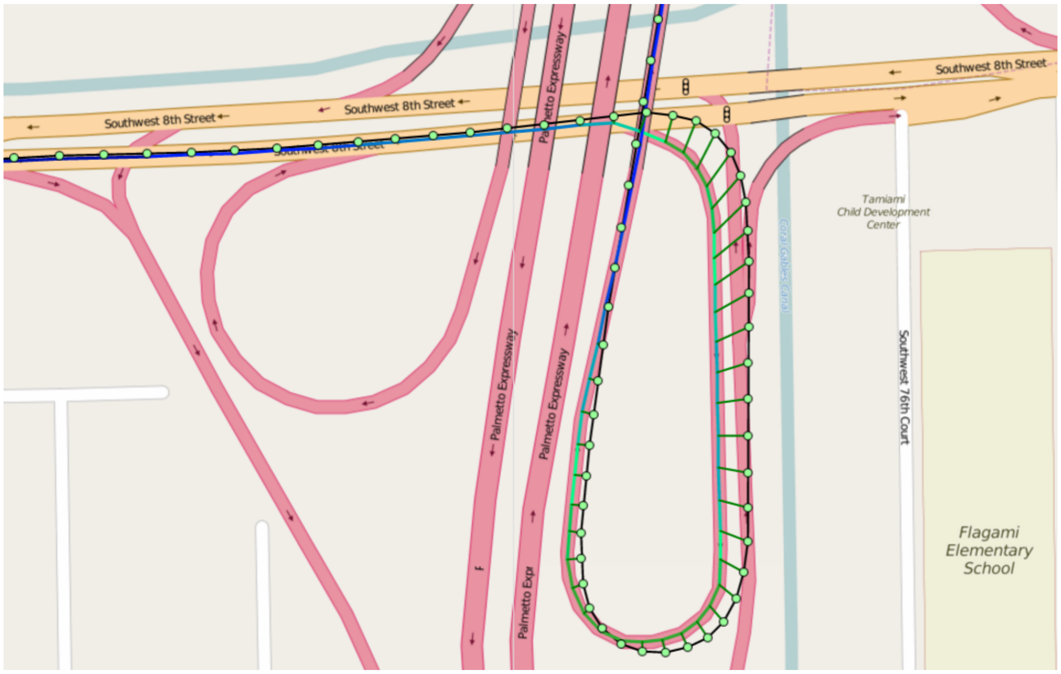


Figure 4.1: Vehicle Trajectory Map Matching

figure shows that the map matching algorithm matches each tracking point to the given road network.

4.2.1 Road Network Model

The road network can be seen as a weighted directed graph $G(V, A)$, where V is a set of vertices (nodes) and A is a set of arrows (arcs). Each road segment is an arc, and the intersection is a node that shared by arcs. The weight of a road is calculated based on its grade and length. The graph uses two directed arcs to present every bidirectional road. So an arc a can be presented as a four-tuple:

$$a = \{start\_node, end\_node, weight, speed\_limit\} \tag{4.1}$$

4.2.2 Vehicle Trajectory

There are many devices have the ability to generate vehicle tracking data. One of the most commonly used device is the dashcam with GPS capability. This kind of dashcam contains a GPS module which continuously record the location of a vehicle at certain interval (normally every second), along with the video. The data format varies from device to device, therefore this article uses abstract metadata that can be parsed from most types of devices.

A GPS trajectory T is a sequence of GPS log entries (waypoints) ordered by timestamp t:

$$T = \{w_t \mid t_0 \leq t \leq t_n\}. \tag{4.2}$$

Each entry can be expressed as a four-tuple:

$$e = \{timestamp, latitude, longitude, speed\} \tag{4.3}$$

4.2.3 Matched Route and Trajectory

A matched route R is the actual route that a vehicle traveled. It is a sequence of connected arcs with correct direction:

$$R = \{a_1, a_2, \ldots, a_m\}. \tag{4.4}$$

Based on the matched route, the matched trajectory MT is a sequence of coordinate points on route R. Each point p_t is the corrected mapping of its corresponding waypoints w_t, as shown in equation. M is the mapping function.

$$MT = \{p_t = M(w_t) \mid t_0 \leq t \leq t_n\}. \tag{4.5}$$

Now the problem of vehicle trajectory map matching is defined as: Given a raw vehicle trajectory T and a road network $G(V, A)$, generate a matched trajectory MT on matched route R.

4.3 Improvement of HMM Map Matching

Many map matching approaches used Hidden Markov Model [37] [73] [66]. [54] can be seen as a variant of HMM. They treat raw GPS points as observations of HMM, and all the candidate way segments as states of HMM. By calculate the probability, it is easy to get the most possible state transition (matched route) based on observation using Viterbi algorithm [94].

Their algorithms have significant advantages over other methods in terms of speed and accuracy. However, it still has room for improvement after studying the mechanism of HMM.

4.3.1 Analysis of HMM in Map Matching

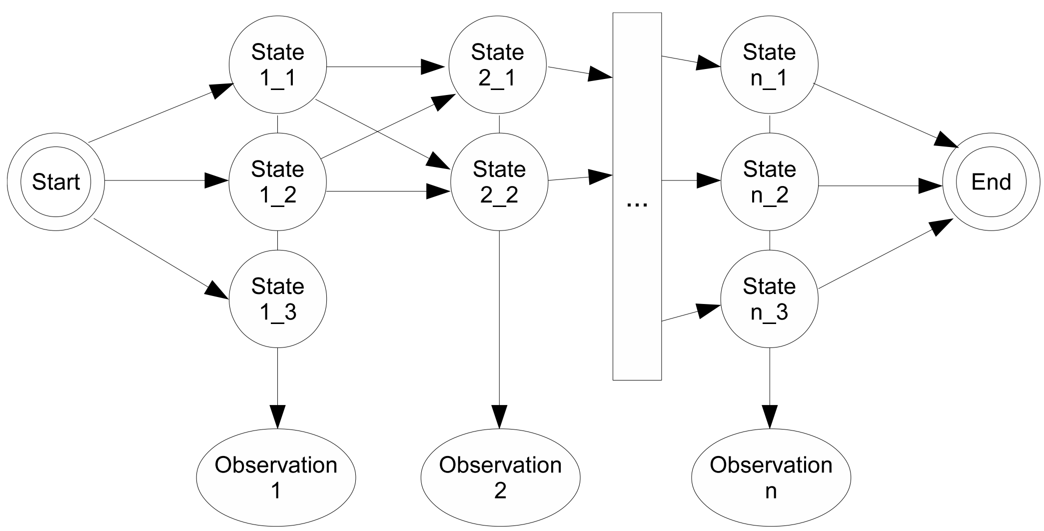


Figure 4.2: HMM in Map Matching

In the context of map matching HMM has some unique properties. First, not all the states are associated with each observation. In contrast, only a small subset of states are related to a specific observation. Second, The state transition directions are consistent with the order of observation sequence. It just like the film tape, each frame is a observation with a subset of states, and the states only move forward.

The standard method to solve HMM is Viterbi algorithm. To use Viterbi for HMM in map matching, this algorithm should be modified a little as shown in Algorithm 1.

Algorithm 1 Viterbi for Map Matching

procedure MM–VITERBI
    for each state $S_{1,i} \in S_1$ **do**
        $S_{1,i}$.Probability=InitialProbability($S_{1,i}$)
    for $i = 2$ to n **do**
        for $j = 1$ to Len(S_i) **do**
            $S_{i,j}$.Probability= $-\infty$
            for $k = 1$ to Len(S_{i-1}) **do**
                CombinedProbability=$S_{i-1,k}$.Probability*
                    TransitionProbability($S_{i-1,k}$, $S_{i,j}$)*
                    ImissionProbability($Observation_i$,$S_{i,j}$)
                if CombinedProbability> $S_{i,j}$.Probability **then**
                    $S_{i,j}$.Probability=Transition($S_{i-1,k}$, $S_{i,j}$)
                    $S_{i,j}$.MostProbableOrigin=$S_{i-1,k}$
    Generate most probable route using MostProbableOrigin information,
    backtracking from state n to state 1.

The complexity of Algorithm 1 is $O(N \times |S|^2)$, where N is the total number of observations and S is the max limit of state quantity in each observation. The map matching program calculates the transition probability using the pathfinding algorithm, which is extremely time consuming. So the best way to speed up the map matching algorithm is to decrease N and S. This article provides a solution to filter out unnecessary points to optimize the running time which is named Filtered HMM (FHMM), as shown in Subsection 4.3.2 and 4.3.3.

4.3.2 Preprocessing Optimization

The raw GPS data normally contains more than necessary points. The dense points not only slow down the algorithm but also bring more error. [66] suggest removing points that within $2\sigma_Z$ of the previous included point, where σ_Z is the standard deviation of GPS measurements. [54] keeps the accuracy above 85% by controlling the sampling interval to 0.5 minutes. However, either distance or time intervals in the above approaches are based on experience, and the parameters may change with different device.

This article uses a new way to effectively clean the data. First, only one point is necessary for a road segment. That means the algorithm can set the interval limit to the average of segment length. Second, road transition happens only at intersection. So a certain angle change can be used as a trigger to mark a key turning point. Third, if a group of GPS points are too far way from their nearest road segments, it normally means they are in the area without road network data (e.g. parking lot) or the data is too inaccurate to use (e.g. device initializing or in the tunnel). Throw this type of data away can increase the accuracy of the matching result.

After preprocessing, the algorithm eliminates 90.4% points while producing the same correct result for a GPS tracking data (1 second sampling rate) shown in Figure 4.3.

4.3.3 Runtime Optimization

Through calculation of HMM, the previous methods using constant radius r to limit the searching area of candidate road segments. They often set r large enough to

Figure 4.3: Cleaned Waypoints As Green Dots

include all the possible situations. This can generate a lot of useless states and waste time on computing their transition probabilities.

By observing it is easy to find:

(1) The road density is not the same from place to place. Road density around highway normally is lower than in the residential area.

(2) It is less likely for a car to change road on highways than on local streets.

So this article uses different initial searching radius r for each type of the road. The initial radius is suitable for most situations. While there are not enough candidates found than the threshold, it will expand automatically and search again. We set r small because the K-Nearest Neighbors(KNN) search costs much less than the transition probability calculation, which is based on expensive pathfinding.

4.4 Trajectory Mapping

The target of map matching is to get the matched route. In order to konw the accurate vehicle location on the route at each timestamp, the GPS points from the trajectory have to be mapped to the route. One straightforward method is finding the nearest projection for each GPS point on the route. However, there are many special situation need to be handled.

First, all the GPS unit has drifting problem while the receiver is at a standstill. The trajectory of a vehicle will move back and forth in a very short range if mapping these drifting points on the route. The trajectory mapping algorithm will combine these points into one point. This point is the statistical center of all the stationary points. Figure 4.4 shows an example of fixing the GPS drifting on red light.

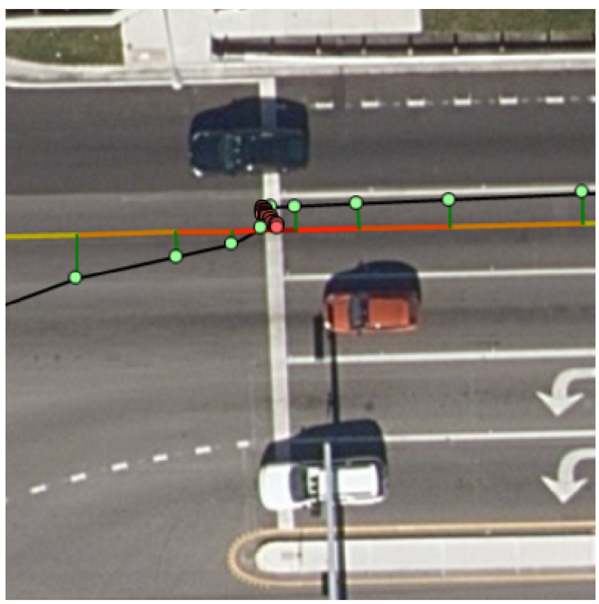


Figure 4.4: GPS Signal Drifting On Red Light

Second, in some place (e.g. urban with high building) where GPS signal is blocked or reflected. The GPS points collected in this kind of places are less accurate than other points and cannot be mapped directly. The solution is continuously monitor the distance between each point and the matched route. When the distance is greater than the threshold, a window is created until next point within threshold is found. Then the points in the window are mapped to the route between the two confident points on each end. The spacing of matched points are scaled from the distance between the unmatched GPS points as shown in Figure 4.5.

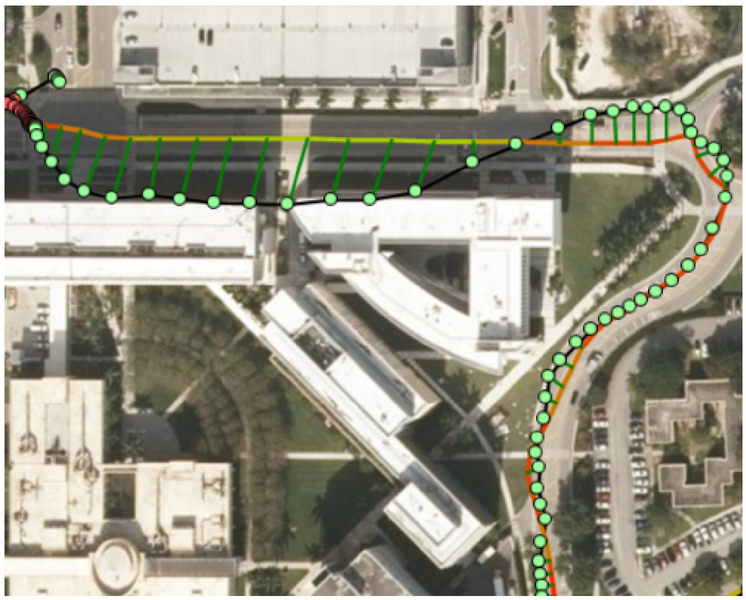
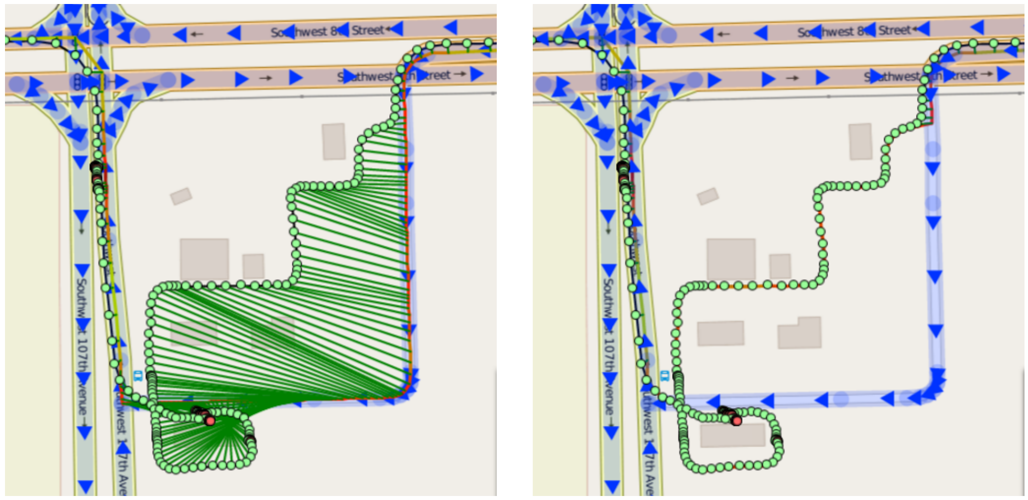


Figure 4.5: GPS Error Fix In Building Area

Third, the road network coverage is limited. A vehicle trajectory can sometimes off-road (e.g. plaza or parking lot). Enforce mapping these points using above method will cause unpredictable results as shown in Figure 4.6a. The better way is skip matching this part of trajectory and leave it as is that shown in Figure 4.6b.

Then the new question becomes how to detect the off-road places and distinguish them from GPS low-signal areas? Since both the off-road and low-signal events have one same behavior that points are far from road, they can all be filtered using above window method. After that more filters can be applied to distinguish them.

From observation it is easy to find low-signal situation has a maximum distance limit. So if any point in the window is beyond this threshold, it is very likely from



(a) (b)

Figure 4.6: (a) Mapping In No Road Area. (b) Skip Mapping For Far Points.

an off-road trajectory. Second observation is that off-road points are more smooth and continuous than low-signal points. So an evaluation function can be used to determine how sparse a trajectory is, and filter out the low-signal trajectory.

4.5 Experiments

The experimental results are compared with original HMM. All the experiments use 318 real trajectory data collected by volunteers using GPS-enabled dashboard cameras. These trajectories are in the range of Miami city. The road network data of Miami comes from Open Street Map, which contains 160,151 arcs and 77,344 nodes as shown in Figure 4.7.

Figure 4.8 shows the the impact of number of raw way points on running time. It is easy to find out the original HMM method is greatly affected by the point quantity. The FHMM method has relatively stable running time that less than one second thanks to the high efficiency filter.

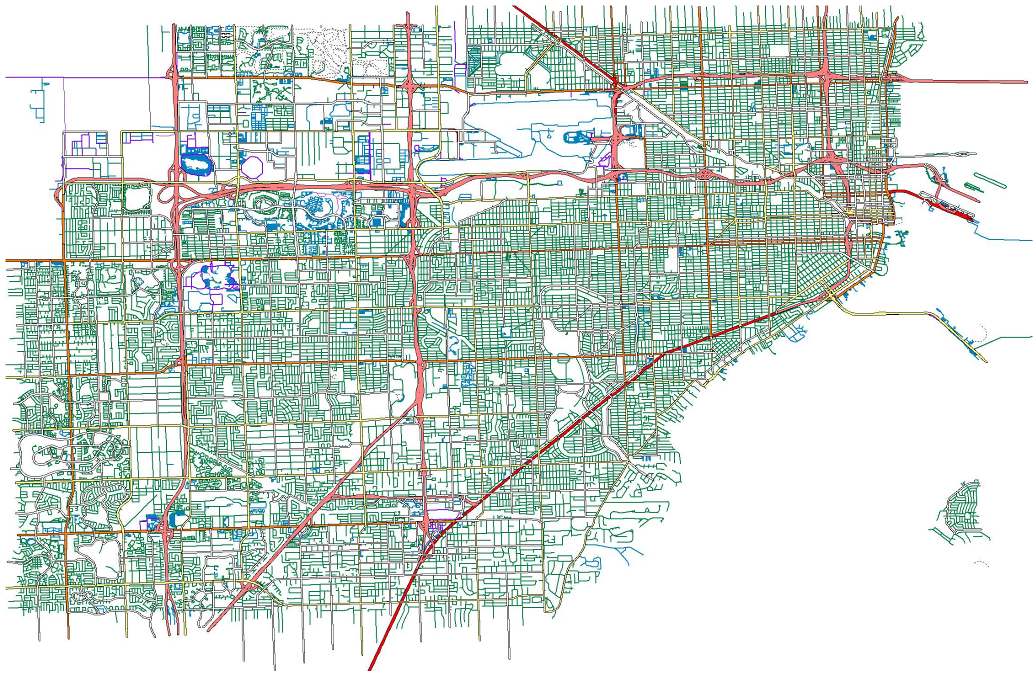


Figure 4.7: Road Network Of Miami

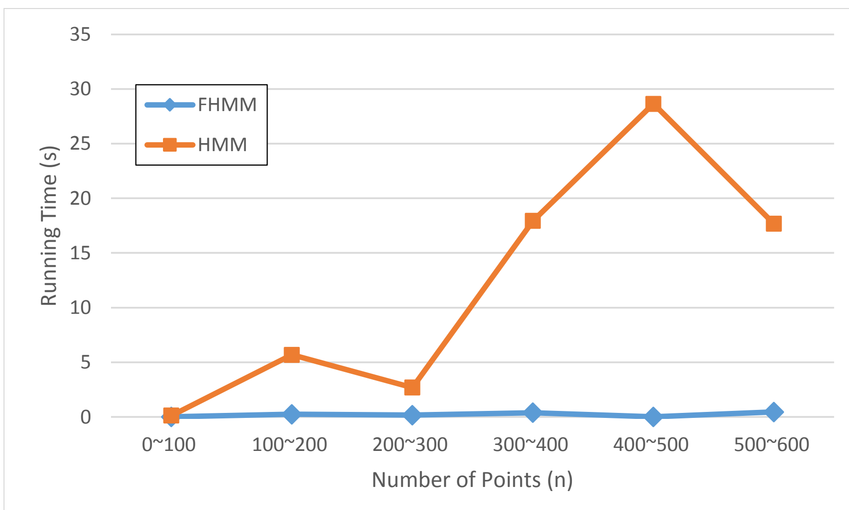


Figure 4.8: Running Time Comparision

Figure 4.9 presents the comparison result of point filter efficiency. The FHMM method can filter more than 90% points in all the tests, while original HMM method is around 60% 70%. Here the filter interval of original HMM is set to 20 meters, because the error rate will increasing dramatically beyond this value.

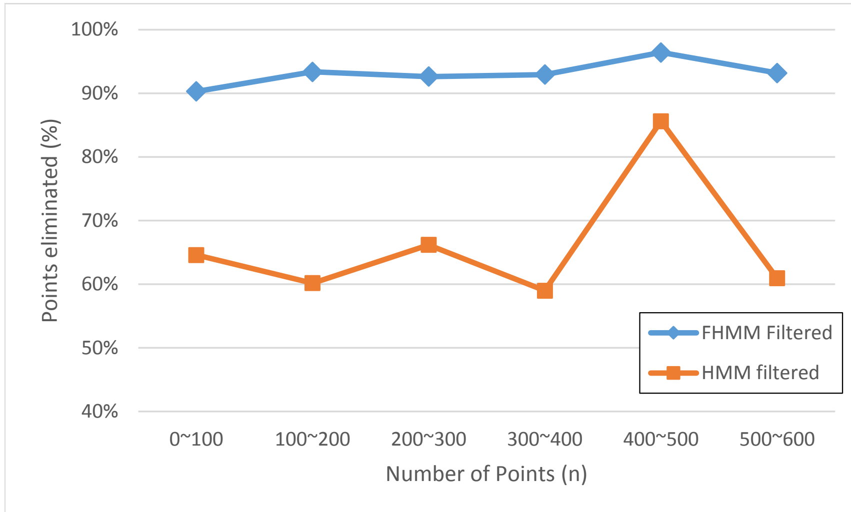


Figure 4.9: Points Elimination Comparision

4.6 Conclusions

This chapter shows an fast and high accuracy map matching method called FHMM that based on the original HMM approach. This method employs preprocessing and runtime optimization to filter out unnecessary way points efficiently. The experiment results show that the running time of FHMM significantly outperforms original HMM approach while still maintain the accuracy.

In future work, more variables will be included in the algorithm since the application is mainly for dashcam. A video-aided map matching algorithm is planned. Gyroscope and accelerometer data can be used as well.

CHAPTER 5

VISUALIZATION OF NETWORK CONSTRAINED VIDEOS

5.1 Introduction

Map have been extremely important trip planning tools for thousands of years. With the rapid development of online GIS services, Web based map applications have become popular in everyday life. Increasingly, drivers tend to review their desired route using online maps before visiting an unknown destination. Some people even engage in a virtual tour instead of physically going to a location. Beyond normal road maps, the use of geotagged multimedia can provide the users an immersive experience. Many existing online map services include geotagged photos and videos of certain places, either uploaded by users or collected by the provider [58].

In recent years, dashboard cameras (dashcam) are widely used and their numbers are increasing rapidly [61]. Many of these videos are geotagged or are accompanied with position logs. Dashboard cameras continuously record videos in the loop mode while users are driving (i.e., the newest footage overwrites the oldest footage), just like security cameras. Considering the fast-growing user group and globalized coverage, if saved and not automatically overwritten, dashcam videos big data can be utilized to benefit our society, such as reporting problems encountered on the road to city managers [61] or mining life patterns [106].

This chapter proposes the City Recorder, an online platform to collect user-uploaded dashcam videos for immersive route preview. Compared with the state of the art, our main contributions include:

- A smart data importing system with the ability to recognize the different brands of dashboard cameras and smartphone applications

- Automatically and intelligently retrieve and merge related videos for specific locations and route previews
- An efficient route selection algorithm to ensure high visibility and minimum video switching
- Synchronous playback of both the video and the route, with cross-interactive capability
- Suggest related videos for a same place but with different time, weather, season and resolution.

The rest of this chapter is organized as follows. In Section 5.2, we introduce the related work. We then describe the architecture of our framework in Section 5.3. A prototype has been implemented and used to validate our system using volunteer collected data, which is discussed in Section 5.6. Finally, we conclude in Section 5.7.

5.2 Related Work

Displaying street-level scenery on a map is not a new idea. Many products on the market can display geotagged multimedia on a map [58][105], most of which are photo-based. However they only reference videos and photos as points on the map. It is difficult for users to completely experience the whole street by these scattered points. Google developed the product named Street-View, based on Google-Maps, which provides panoramic view-points along streets worldwide [4]. Although Street-View covers almost all sceneries along the road, it's still discrete panoramic photos. Users have to press forward again and again to jump from one point to another. Some systems have been proposed to solve this problem by generating smooth videos or photos from panoramas along streets [10][45][72]. These methods depend on the intensity of the panoramas collected and therefore the quality of the videos or images generated from these panoramas can be compromised.

Recently, several platforms and frameworks have been proposed to utilize geo-referenced videos along with the map. These kinds of videos come with spatial and temporal information bound to frames of the video. PLOCAN [81] focuses on combining a Web-based video player and a map together to play dedicated videos with positions shown on the map. Citywatcher [61] lets users annotate dashcam videos and upload them to the City Manager. Chiang et al. [12] proposed a framework to share and search user-uploaded dashcam videos. However, it requires a specific application installed on the smartphone to record the video and does not support video replay by any chosen route.

The limitations of the above methods have motivated us to find new ways of generating street-level videos that provide the best user experience. Our proposed method will not only generate high quality videos for route preview, but also require minimum setup and configuration. In the next section, we will explain in detail how the City Recorder solves these problems.

5.3 System Design

The City Recorder is designed with flexibility and reusability in mind. To modularize various components involved in the process of video collection, route generation and user interaction, we adopted classic three-tier architecture to build this web-GIS system. As shown in Figure 5.1, the presentation tier resides on the client side while the logic tier and data tier run at the server side.

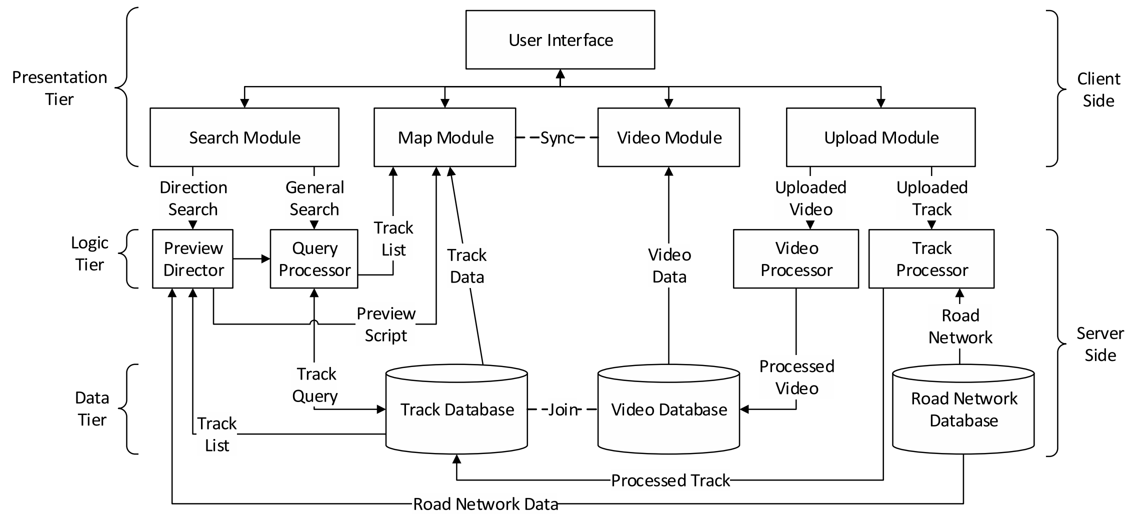


Figure 5.1: City Recorder System Architecture

5.3.1 The Presentation Tier

The presentation tier responds to user requests and communicates with the server. It has an event-driven user interface that relies on four major modules. The map module and video module are core parts of the website to show tracks and videos to users, respectively. Location information in tracks is logged parallel to the video at specific intervals (e.g. one sampling per second). We use TerraFly [78] as our map engine to render tracks as polyline, and the video module utilizes open source jPlayer[1] to play videos that related to the track. The upload module redirects users to the dashcam video uploading page and guides users throughout each step. The search module integrates multiple query functions into one search bar, consists of address search, direction search, video search, etc.

[1] http://jplayer.org/

5.3.2 The Logic Tier

The logic tier contains all the necessary business logic to get the job done. The video processor and track processor are used to transfer user uploaded contents to desired format and store them into a database. More details will be covered in Section 4.

The query processor mainly handles general queries from the presentation tier. A typical query is like "get all the tracks inside the bounding box" or "find all the night time videos recorded last Halloween". The resulting track list from the data tier will be transferred to the map module and displayed to users.

The preview director is the most important part in the route preview system. When a user searches for a route from location A to B, the module will first calculate routing using road network data and then send a query to the query processor: "get all the tracks overlapping the given route". After getting the track list, the preview director will then prepare a preview script containing a video playing sequence for the presentation tier, which brings to the user a smooth preview video. Section 5.5 will explain this algorithm in depth.

5.3.3 The Data Tier

The primary task for this tier is CRUD operations (Create, Read, Update and Delete). Videos and tracks will be stored in the spatial-temporal database to support complicated queries coming from the Logic Tier. The GIS database is optimized using TerraFly automonic resource management technology [57]. Additionally, a road network spatial database is required for routing calculation and map matching of tracks. Our implementation uses open road network data of the OpenStreetMap[2] project. Other data sources like base-map layer data and POI (point of interest)

[2]https://www.openstreetmap.org

data are basic part of the Map Module and are not separately listed in the Figure 7.2.

5.4 Data Collection

5.4.1 Track Data Processor

The track data processor extract tracks from the user uploaded file and stores them in the track database.

Figure 5.2: Dashcam Test Equipment

The first step is converting these tracks to a device-independent format. Since dashboard cameras are still in the development stage, many manufacturers entered this market lately and have developed different standards. Unlike the mature technology of photography meta-data (e.g. the Exif format) to embed geotag information, there is no unified method to store the tracking log within a video file. One approach directly encodes location information into particular frames inside the video file (e.g. MISB Standard 6101[3]). Another popular way is using a separate waypoint log file to accompany the video file (e.g. NMEA 0183[4], GPX, and KML). Some devices add a text overlay on the video to store real-time coordinates. A track parse module inside this processor handles all the track extraction tasks.

The next step is using a map matching algorithm (MMA) to align the track data based on the road network. As a result of limitations of positioning devices and inaccuracy of digital road map, location points inside the track data may be slightly off the road. To determine the road of the driving video, we need to match these points on road segments using MMA module. The matched track will be used by the route preview director as explained in Section 5.

5.4.2 Video Data Processor

The video data processor preprocesses the video to a unified format that is suitable for Web-based players. Video formats differ from one device to another. To ensure smooth video switching while previewing the route, it is desirable to have only one format across all videos. Here we choose mp4 (MPEG-4 Part 14) format because of its broad network compatibility.

Another issue is that most devices support at least 720p video recording, at the 512kbps bit rate, which is about 60MB per minute in the raw format. Considering

[3]http://www.gwg.nga.mil/misb/docs/standards/ST0601.7.pdf

[4]http://en.wikipedia.org/wiki/NMEA_0183

the network bandwidth limitations for most users, compression will be performed to reduce the video size. We utilize a free video processing tool FFmpeg[5] to perform video re-coding and compression tasks.

5.4.3 Stree Data Loader

This chapter uses OpenStreetMap as the street data source. A fast and easy to use street data loader is developed for this task, as shown in Figure 5.3.

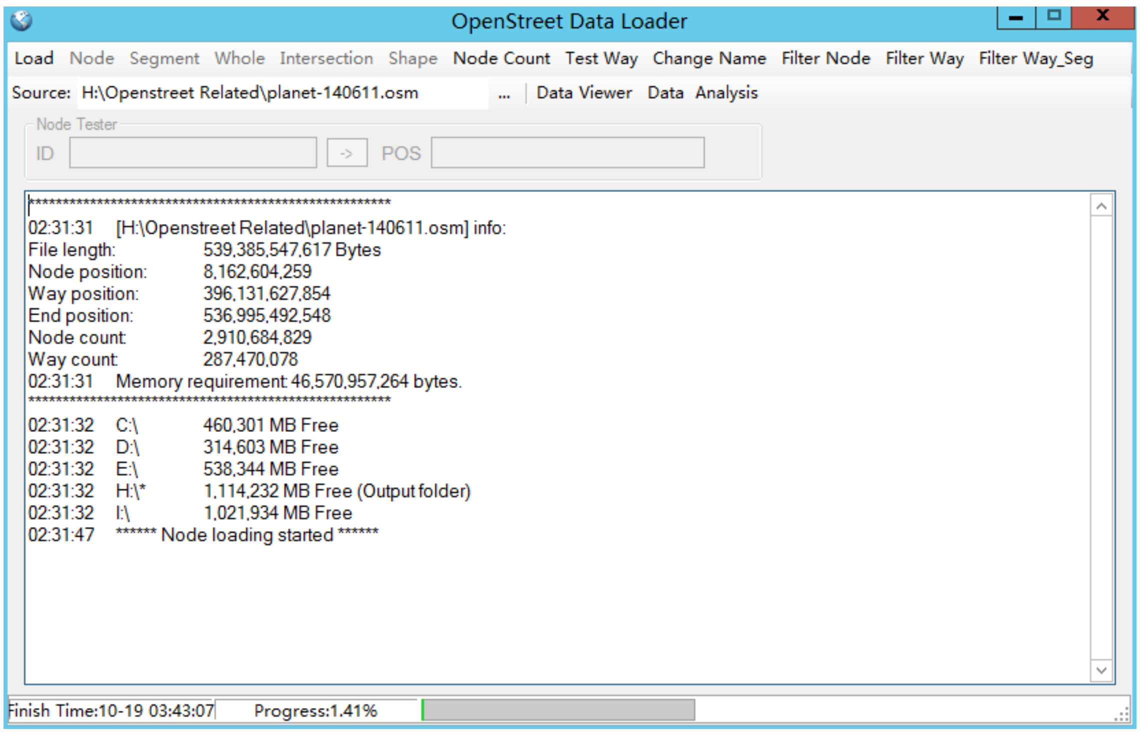


Figure 5.3: OpenStreetMap Loader

5.5 Preview Director

The preview director collects a series of video clips related to the route and arranges them on a timeline to form a preview script. The script will be executed in the

[5]https://www.ffmpeg.org/

presentation tier to show a seamless preview video.

A high quality route preview normally meets the following criteria:

1. Low switching frequency. This is to facilitate users' adaption to the changes caused by the transition from one video clip to another, since videos differ in many ways, like the time of day, the season of the year, the weather conditions, the device type and the lane of driving.

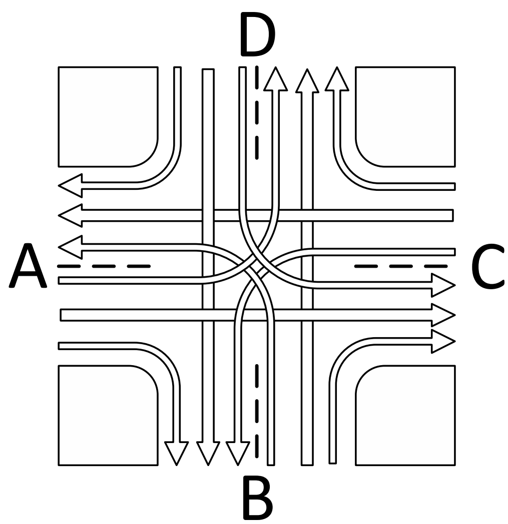


Figure 5.4: Intersection

2. Natural transition between videos, even at road intersections. Video clips must have the same directions to generate the preview route. In addition, the ones which have the same turning direction with the preview route are preferred. Figure 5.4 illustrates the most common 4-way intersection that contains 12 possible driving directions. For a route from A to D, we have three available video clips in Figure 5.5. It is easy to observe that the solution in Figure

   [h!]

   5.6 is smoother than in Figure 5.7 although it has one more transition.

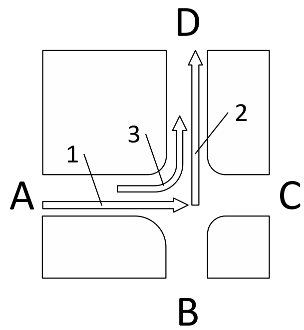


Figure 5.5: Available Videos

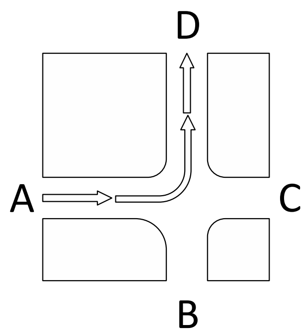


Figure 5.6: Smooth Turn

3. Using high quality videos first when we have multiple candidates. Factors affecting the video choice include the time of day, the weather conditions, resolution, bit rate, etc. Normally, day-time videos are better than night-time, and 1024p videos have more details than 720p.

5.5.1 Query Related Videos

First we get the preview route from the routing service OSRM[6], which calculates the directions between locations given by users using the road network database. Here we define *related videos* as the ones at least partially overlapping the preview

[6]http://project-osrm.org/

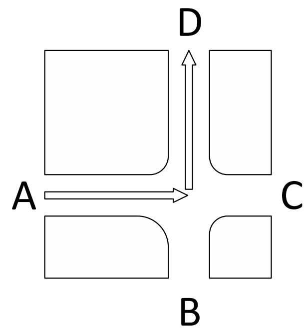


Figure 5.7: Sharp Turn.

route and having same direction with it. Then we cut non-overlapping parts off the videos and generate a list of related video clips. Ideally these clips will cover the whole preview route.

5.5.2 Video Selection Algorithm

Our task is to select the minimum required videos with high quality to cover the preview route. It can be reduced to the well-known weighted interval coverage problem [6]. Consider a preview route from an origin o to a destination d as an interval $I = [o, d]$ and a set S of intervals $I_i = [s_i, e_i], i = 1, \ldots, n$ (I_i is the ith related video which overlaps I from s_i to e_i). Here we assume that S covers I. Each video clip has a weight w_i calculated by adding up flaw factors (a lower weight number represents a better video quality). A path $S' = J_1, \ldots, J_k$ from o to d is a subset of S such that J_i and J_{i+1} overlap (or, at least, connected) for every $i \in 1, \ldots, k-1$. The length l of S' is defined as $l = \sum_{i=1}^{k} w_i$. Our task is finding the subset S' which has the minimum length l for given S.

Using this method we can convert the available video tracks in Figure 5.5 into an interval model as shown in Figure 5.8. If we view intervals as nodes and use edges to indicate that two intervals are overlapped, the problem can also be converted into

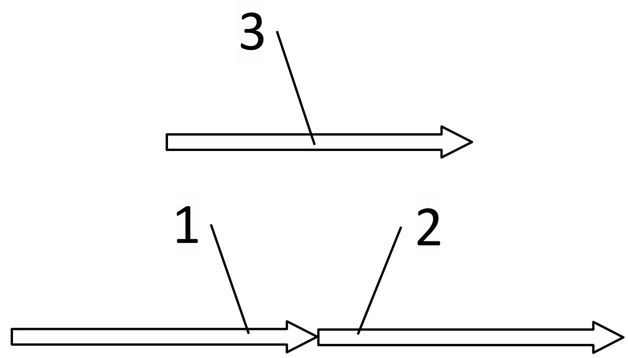


Figure 5.8: Interval Modal

a graph model as presented in Figure 5.9. Note the interval nodes are weighted. Here we use a dotted line because the transition from Interval 1 to 2 is too sharp (we define a direction change of more than 45 degrees as *sharp*). We can give these dotted lines high weights compared to other lines with weight 1. This problem can be solved using a variant of Dijkstra's algorithm [14], by calculating the shortest distance using weight of nodes and edges together.

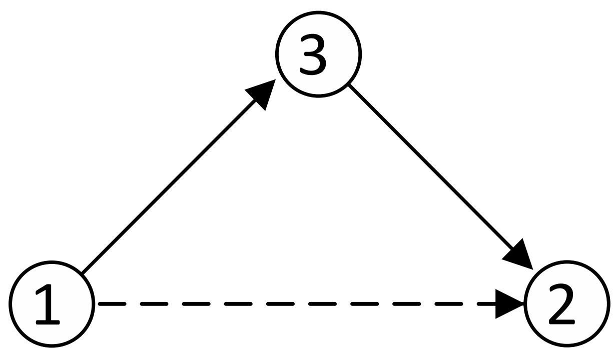


Figure 5.9: Graph Modal

In reality, we cannot guarantee full video coverage over a city. In the situation that no video is available for a section of a route, we use alternative data sources described in [72].

5.5.3 Generating Scripts

After executing the above mentioned algorithm we get a sequence of video clips. The present step prepares a script for the online player, which contains multiple modes of action. An action has the form [*video_id, start_time,end_time*], indicating which videos should be played at what time. One thing we have to consider is choosing the correct cutting position for the video clips. If two videos are connected with no overlaps then no cutting is necessary. We just play the next one after the previous one ends. If they have an overlapping part, we choose a point with the minimum direction changes as the transition point.

5.6 Use Case Study

In this section we will demonstrate the effectiveness of our system and show how it is used to help people through a series of use cases.

5.6.1 General View

The left panel of the main page, as shown in Figure 5.10, is a map with dashcam video tracks rendered as polylines in different colors. A list of these videos with more information can be found at top right. When mousing over either the list or the track polyline, the corresponding preview frame picture of the video will show at the bottom right corner. If clicking instead of mousing over, it will enter the player mode as shown in Figure 5.11. In this mode, the position on the map is synchronized with video playback progress. The video will jump to the specific time when a track is clicked, and vice versa. A button switches the map between the street level detail

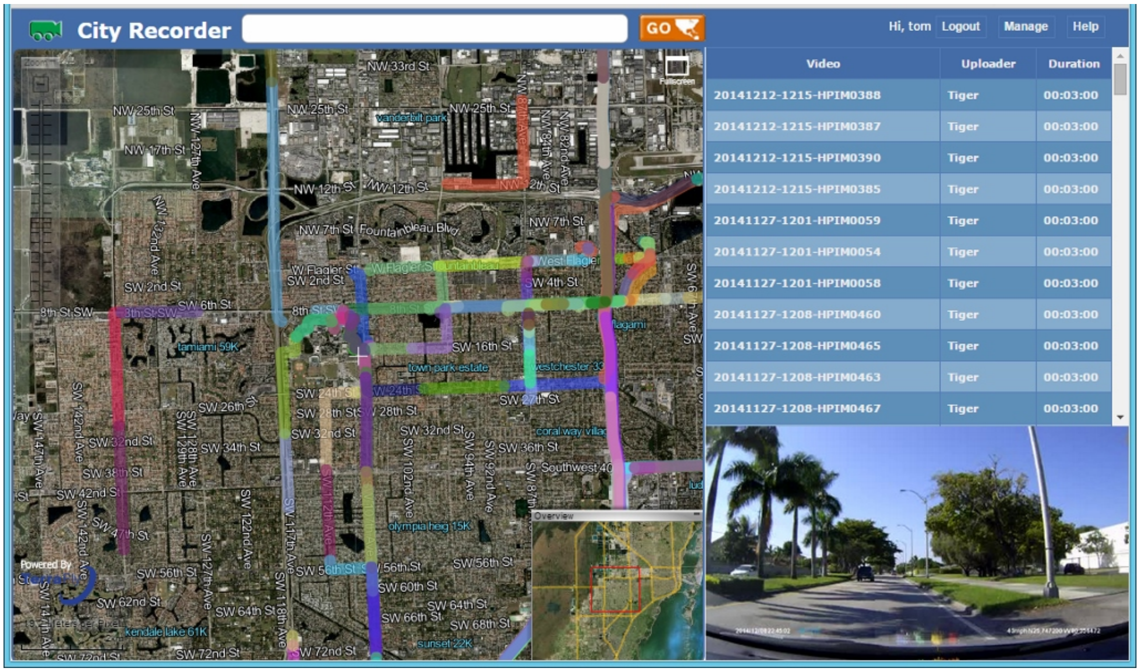


Figure 5.10: Main Page

and a zoomed-out global overview. Velocity and altitude are displayed in virtual instruments along with other information at the top right corner.

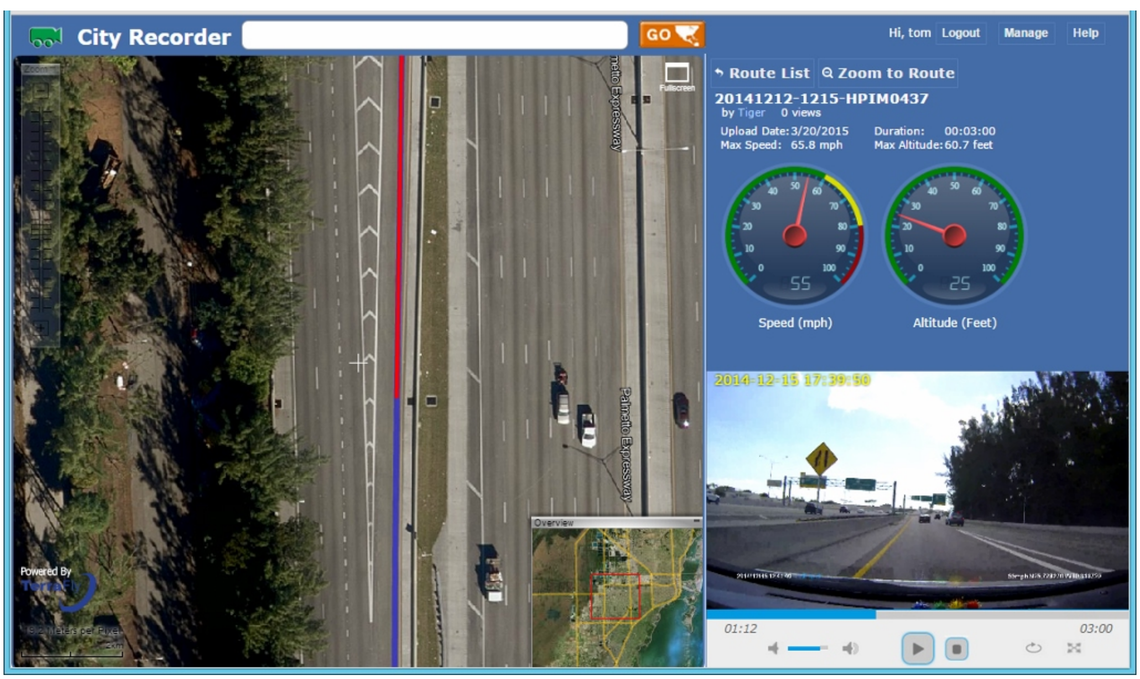


Figure 5.11: Geo-Video Player

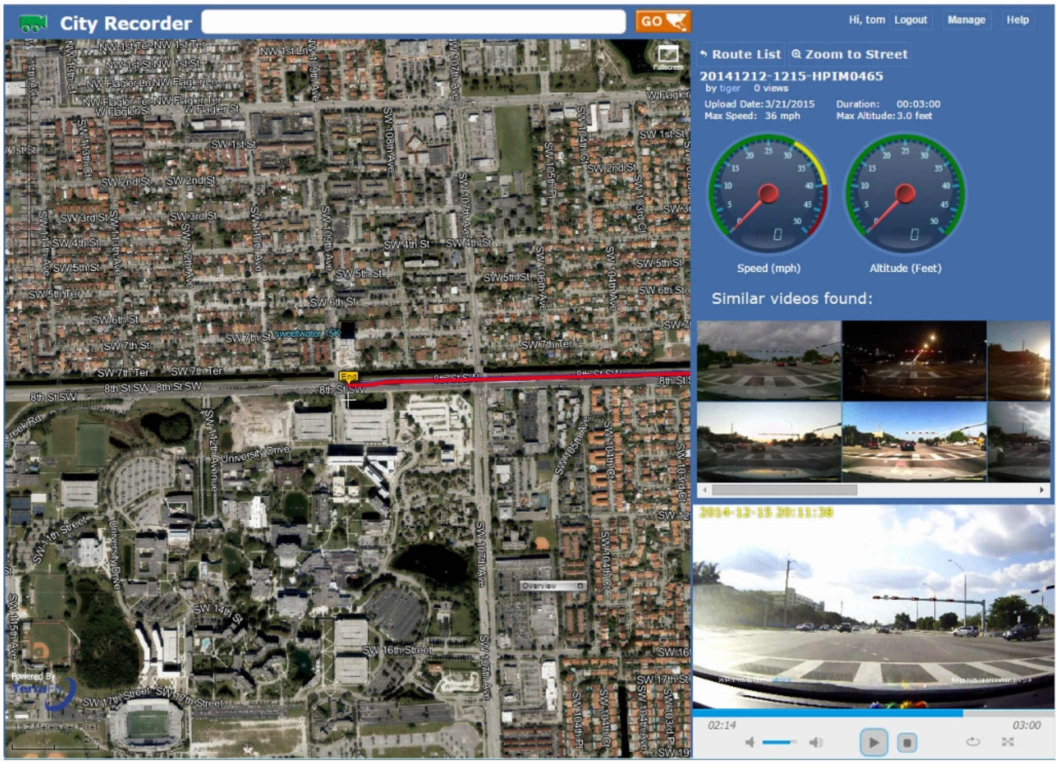


Figure 5.12: Video Suggestion

If other videos are found at the same location and have same directions while in the player mode, the application will push notifications to users as shown in Figure 5.12. This allow users to see a place under different time of day, season of year, weather conditions, etc. It will greatly improve the user experience.

5.6.2 Route Preview

When a user searches for a route, the application enters into the route preview mode as shown in Figure 5.13. It is an online driving simulator, which give users a realistic riding experience. The simulator connects videos generated from Section 5.5 together and plays them seamlessly. The whole user interface is designed to mimic the car interior including instruments, onboard video player, and navigation system.

Figure 5.13: Driving Simulator

5.6.3 Data Management

Users can upload their georeferenced videos to our system using a guided interface as shown in Figure 5.14. Then the data will be processed at the server side and the estimated time is displayed to users as shown in Figure 5.15. Users can choose between waiting online or getting a notification email. Users have the option to change privacy settings of videos as seen in Figure 5.16. They can also share a video with others using the system generated URLs, and the video will not available to public without this URL.

5.7 Discussion and Future Work

This study presented the City Recorder, which is not only a platform for dashboard camera video sharing, but also an online video touring service. City Recorder is a public service that uses crowdsourcing. That is, it relies on contribution from

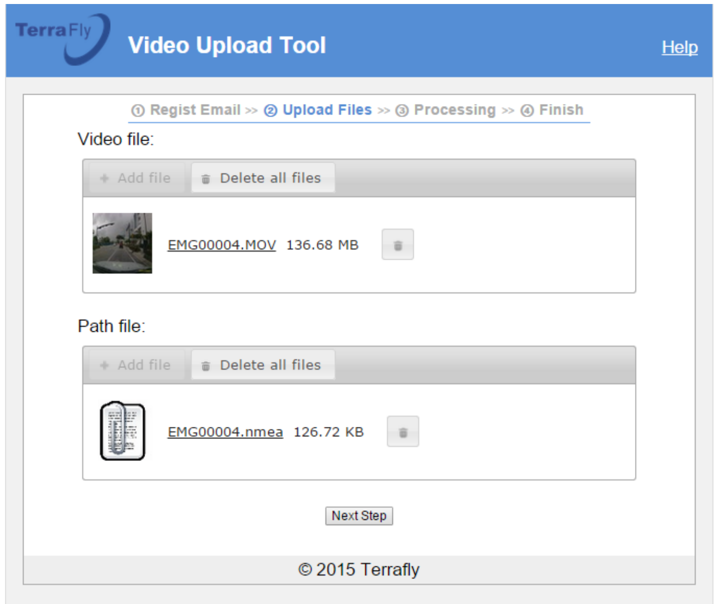


Figure 5.14: Data Uploading

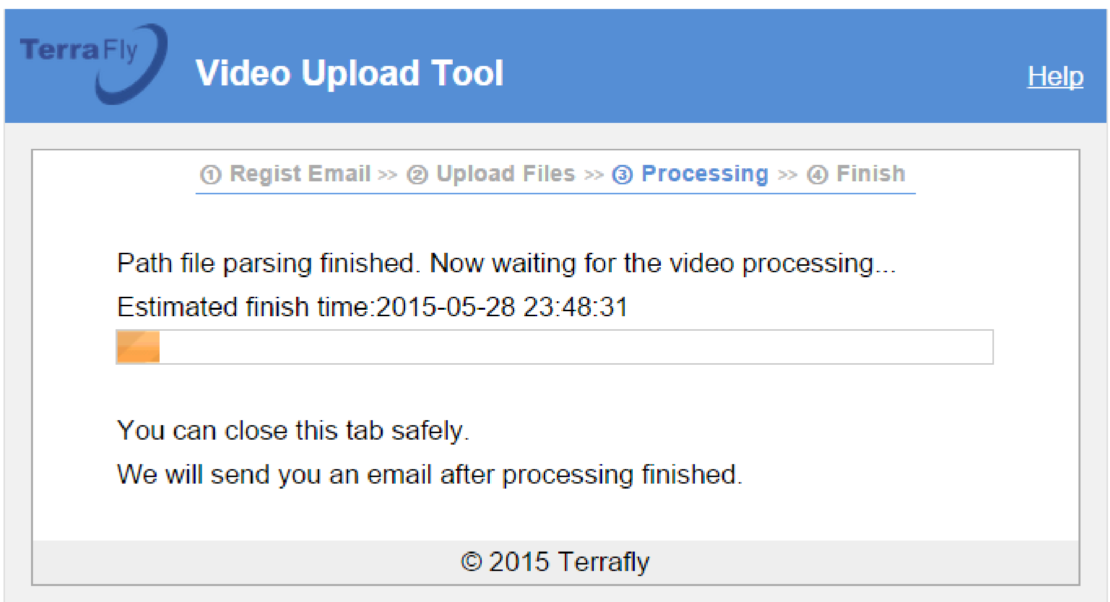


Figure 5.15: Data Processing

and cooperation of voluntary users. The more uploading that occurs, the better video coverage. Crowdsourcing has been found to be advantageous and successful in numerous geolocation-related domains in recent year. There are, however, some challenges with this approach. It is difficult to ensure the quality of the preview in

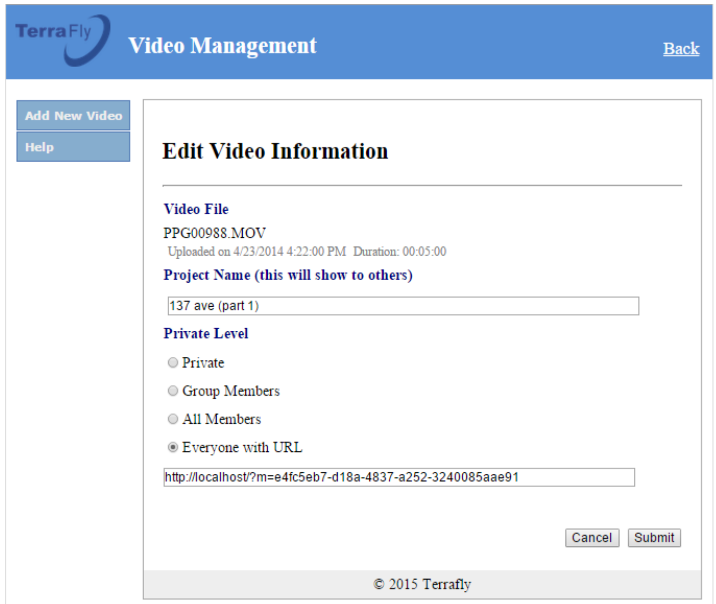


Figure 5.16: Data Management

low coverage areas, even after patching with alternative solutions.

Another concern is privacy protection. Users now can set the sharing levels to public or private, and they can cut off unwanted parts while uploading. In the future, we will further utilize some mature technologies like automatically blurring faces and license plates [24] to prevent leakage of sensitive information. One interesting research approach shows how to make the objects immediately in front of cars transparent after analyzing a set of videos recorded at same place [11]. It can also be used in our system to improve the video quality.

Route previewing is only one of the beneficial usages of this Big Data. Much useful information can be discovered by mining the data, such as road hazard report, traffic congestion area detection, finding carpool partners, and climate research. We are planning to add a data mining interface for this platform to perform further analyses.

CHAPTER 6

VISUALIZATION OF LIVE GEO-REFERENCED VIDEOS

6.1 Introduction

A photograph is usually taken from a location. An action video, which can be seen as a series of photographs from different locations, usually reflects an object moving along a trajectory. In recent years, thanks to the price drop and miniaturization of Global Positioning System (GPS) chips, GPS-enabled video recording devices have proliferated. One remarkable example is the now widely used dashboard cameras (dashcam), which continuously record videos as well as GPS tracks while the users are driving [61]. Other examples include smartphones, unmanned aerial vehicles (UAVs), and sports (a.k.a. action) camcorders. Some devices with networking capabilities can even streaming real-time video remotely. Videos taken by such devices comprise Big Data. However, the playback software modules come from a variety of providers and they are often incompatible with each other. Integration is need to facilitate route preview, virtual sightseeing, crime monitoring, smart city applications [61] and data mining [106].

With the rapid development of online geographic information system (GIS) services, Web-based map applications have entered the everyday life. Many existing online map services include geotagged photos and videos, either uploaded by users or collected by the provider [58]. This chapter proposes the City Recorder, a framework to integrate georeferenced videos from moving objects for immersive city observation through multidimensional perspective. Compared with the state of the art, our main contributions include:

- A smart data importing system able to recognize a wide variety of video and GPS track forms from devices of wide range of brands
- Supports real-time video streaming along with historical video archiving
- Geotagging service for video recording devices without GPS capabilities
- When possible, retrieve and display data from other sensors (e.g. accelerometer, gyroscope, and barometer)
- Automatically and intelligently classify related videos of specific locations by time of day, weather conditions, etc.
- Synchronous playback of both the video and the route, with cross-interactive capability

The rest of the chapter is organized as follows. In Section 6.2, we introduce the related work. We then describe the architecture of our framework in Section 6.3. A prototype has been implemented and used to validate our system using volunteer collected data, which is discussed in Section 6.6. Finally, we conclude in Section 6.7.

6.2 Related Work

Displaying multimedia containing geographic information on a map is not a new idea. Many products on the market can display geotagged multimedia on a map [58][105], most of which are photo-based. And for videos, they only reference them as points (markers on the map). It is difficult for users to understand how the videos were recorded by these scattered markers. Google developed a product named StreetView, based on GoogleMaps, which provides panoramic view-points along streets worldwide [4]. Although StreetView covers almost all sceneries along the road, it's still discrete photos. Some systems have been proposed to solve this problem by generating smooth videos or photos from panoramas along streets [10][45][72]. These methods depend on the intensity of the panoramas collected and therefore the quality of the videos or images generated from these panoramas can be compromised.

Recently, several platforms and frameworks have been proposed to utilize georeferenced videos along with the map. These kinds of videos come with spatial and temporal information bound to frames of the video. PLOCAN [81] focuses on combining a Web-based video player and a map together to play dedicated videos with positions shown on the map. Citywatcher [61] lets users annotate dashcam videos and upload them to the City Manager. Chiang et al. [12] proposed a framework to share and search user-uploaded dashcam videos. However, it requires a specific application installed on the smartphone to record the video. Furthermore, there is no existing platform that integrates all kinds of georeferenced videos. The limitations of the above methods have motivated us to find new ways of showing georeferenced videos that provide the best user experience. Our proposed platform is based on our previous work named City Recorder [104]. We will introduce the architecture of Moving Object Mapper in the next section.

6.3 System Design

The Moving Object Mapper is an online GIS platform based on classic three-tier server-client. As shown in Figure 6.1, the presentation tier resides on the client side while the logic tier and data tier run at the server side.

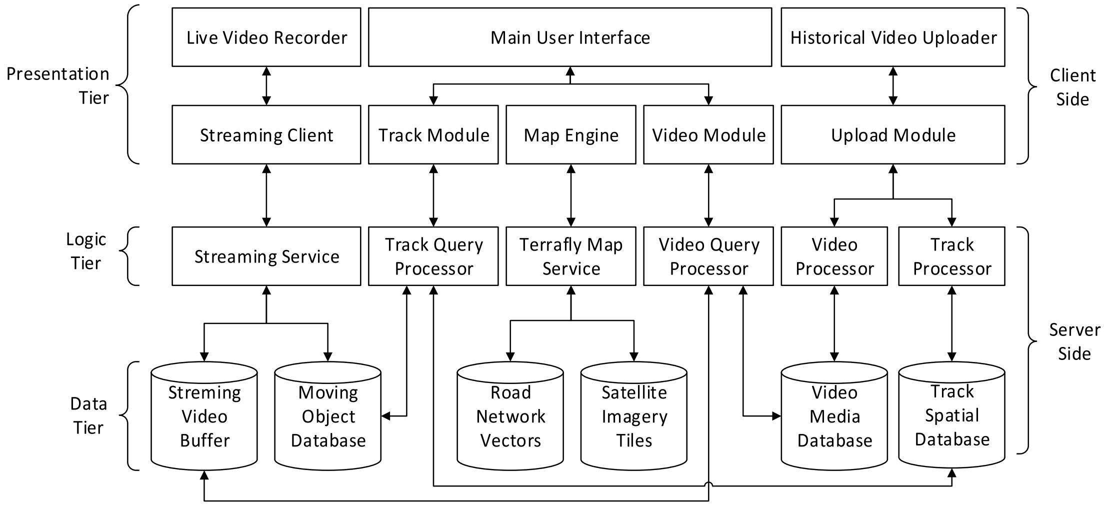


Figure 6.1: System Architecture

6.3.1 The Presentation Tier

The presentation tier collects data from users and respond with query results. The event-driven main user interface relies on three major modules. The track module and video module are core parts of the website to show tracks and videos to users, respectively. We use TerraFly [78] as our map engine to support the track visualization, and the video module utilizes the open-source jPlayer[1]. The historical uploader redirects users to the georeferenced video uploading page and guides users through-out each step. Real-time videos will be transferred to the server by the streaming client.

6.3.2 The Logic Tier

The logic tier processes data between presentation tier and data tier. The historical data will be transferred to desired format and stored into the database, which covered in Section 4. And in Section 5 we will show how to deal with data streaming.

[1]http://jplayer.org/

The track and video query processor mainly handles general queries from the main user interface. Section 6 will explain their visualization in depth.

6.3.3 The Data Tier

We use various types of databases to adapt different data in the data tier. The videos are stored in multi-media database. The tracks, road network vectors and satellite imagery tiles are stored in separate spatial databases. The recording locations of live videos are constantly updated into the MOD (Moving Object Database).

6.4 Historical Data Processing

Most GPS-enabled devices will store track files along with recorded videos in some kinds of storage media. Here the historical data specifically refers to the offline data that users uploaded manually.

6.4.1 Track Data Processor

The track data processor extract tracks from the user-uploaded file and stores them into the track database. Since there are a variety of devices from different manufacturers on the market, the first step is converting these tracks to a device-independent format. The most common track file formats include: NMEA 0183[2], GPX, and KML. A track parse module inside this processor handles all the track extraction tasks.

For videos that recorded along streets (e.g. dashcam videos), we can align the track data based on the road network using a map matching algorithm (MMA).

[2] http://en.wikipedia.org/wiki/NMEA_0183

Regarding to the devices without GPS capabilities, there is also a geotagging tool for users to manually mark the track on the map.

6.4.2 Video Data Processor

Same as tracks, we want a unified format for the videos which is suitable for Web-based players. The mp4 (MPEG-4 Part 14) is chosen considering its broad network compatibility. Another task is compression in order to reduce the video size. We utilize a free video processing tool FFmpeg[3] to perform video re-coding and compression tasks.

Then we utilize Support Vector Machines (SVM) to learn and classify the videos by different characteristics (e.g. weather, road condition, and time of day).

6.5 Live Video Streaming

We provide a mobile application for live video streaming purpose. After users correctly registered on the server, the application will notify the server whenever it is online. The video will be transferred to the server wirelessly along with current location from the GPS sensor.

The location of the moving object is updated in the MOD. Then the buffered video file on server is ready for responding to streaming requests from the users. The videos and tracks will be archived into the historical database in a certain time interval.

[3] https://www.ffmpeg.org/

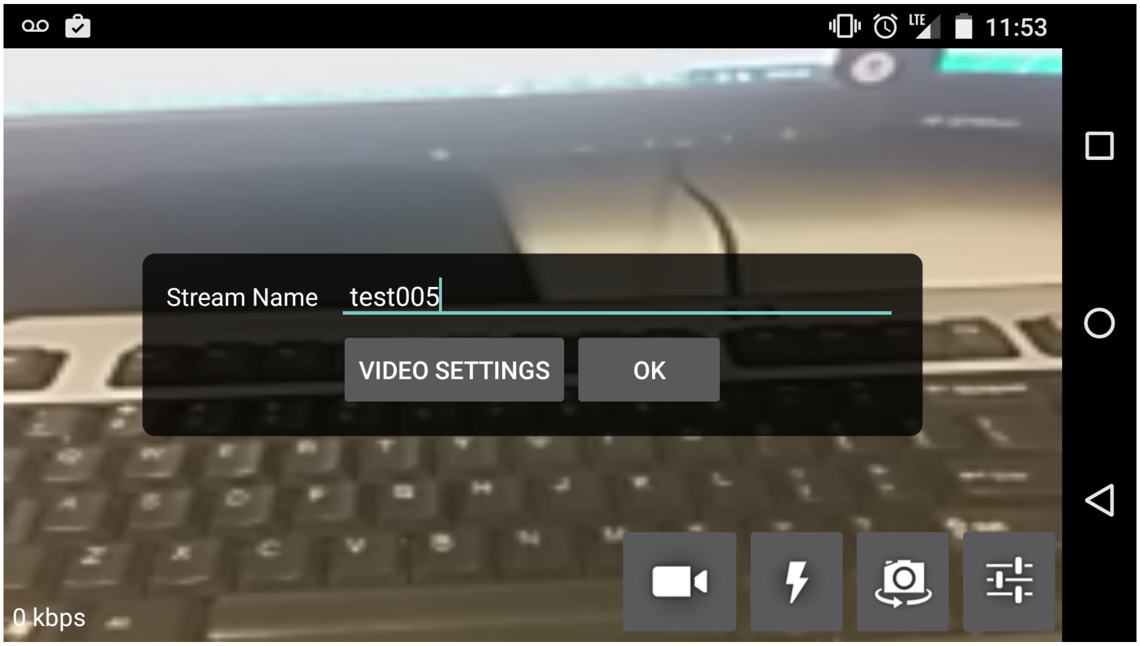


Figure 6.2: Live Video Streaming App

Figure 6.3: Live Video Streaming User Interface

6.6 Use Case Study

In this section we will demonstrate the effectiveness of our system and show how it is used in real life through a series of use cases.

6.6.1 Street Network Based Videos

Figure 6.4: Street Tracks

Most street network based videos come from dashboard cameras. Figure 6.4 shows that a car is turning right. The line graph below the video indicates speed drop. It also can be switched to altitude graph as needed. Figure 6.5 is a map with tracks rendered as polylines in different colors. We did a route query between two points. After clicking the simulator button, it will open the driving simulator shown in Section 6.6.3.

6.6.2 Free Moving Objects

Free moving objects (e.g. pedestrians, boats, and aircrafts) usually have unpredictable tracks. Figure 3 shows a live streaming video from a balloon. The polygon represents the field of vision projected on the ground.

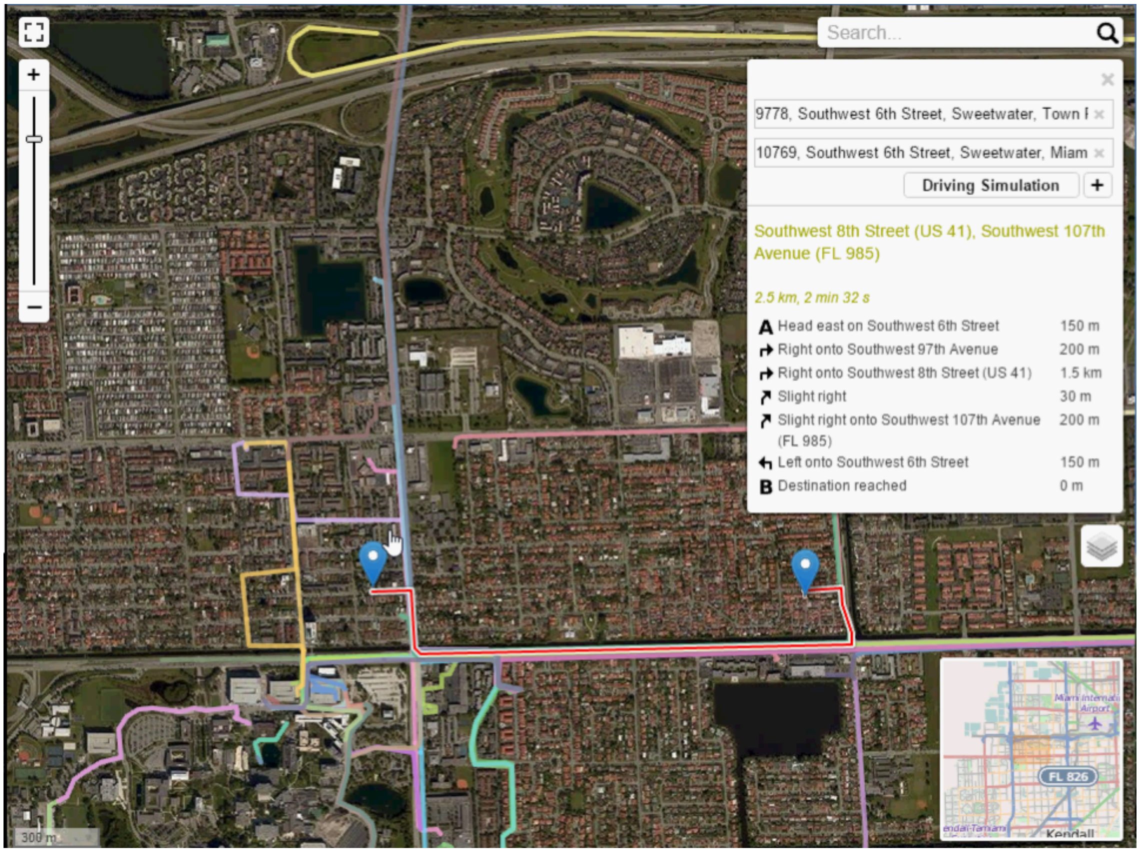


Figure 6.5: Route Query

6.6.3 Moving Video Simulator

The simulator is designed for providing the user an immersive experience. We have designed several different themes to simulate various life situations. Figure 6.7 simulate a car using the dashcam video. If other videos are found at the same location and have same directions, the application will notify users (See the blue Night Mode button). This allows users to preview a place under different time of day, season of year, weather conditions, etc. It will greatly improve the user experience. Figure 6.8 shows an aircraft simulator using the video from a hexacopter drone.

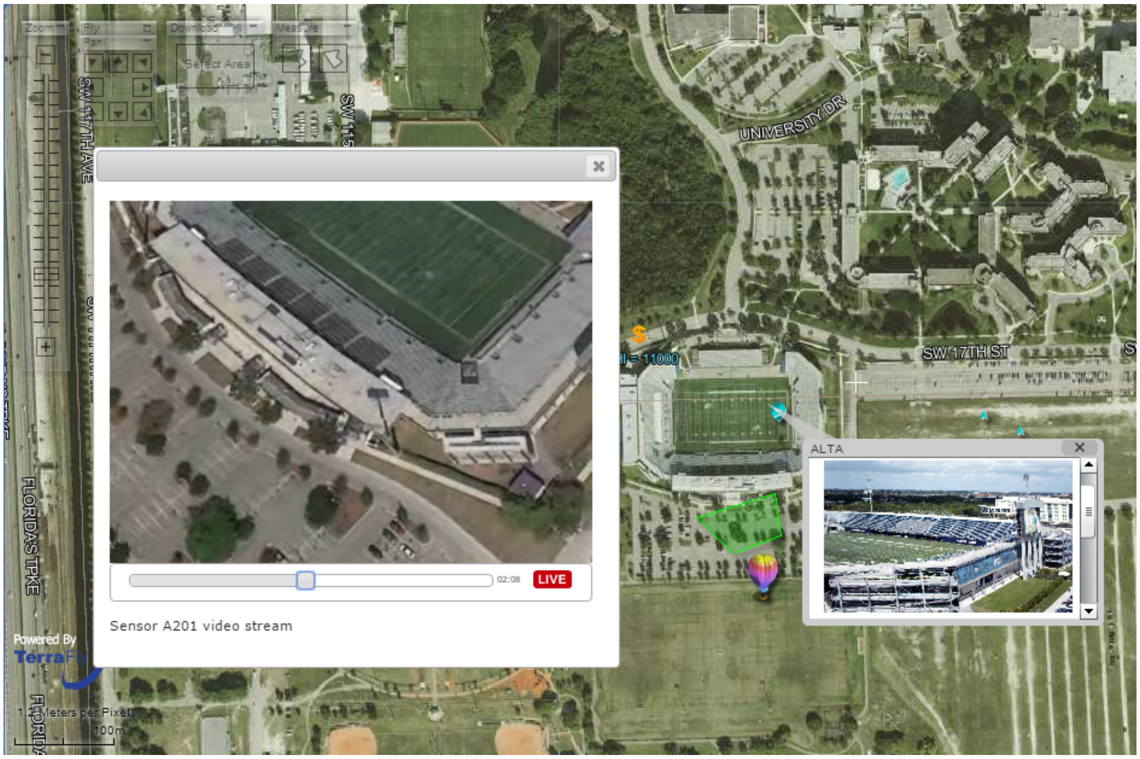


Figure 6.6: Live Video from UAV.

6.6.4 Data Management

Users can upload their georeferenced videos to our system using a guided interface as shown in Figure 6.9. Then the data will be processed at the server side and the estimated time is displayed to users as shown in Figure 6.10. Users can choose between waiting online or getting a notification email. Users have the option to change privacy settings of videos as seen in Figure 6.11. They can also share a video with others using the system generated URLs, and the video will not available to public without this URL.

Figure 6.7: Car Simulator

6.7 Discussion and Future Work

This study presented the Moving Object Mapper for georeferenced video sharing. City Recorder is a public service that relies on contribution from and cooperation of voluntary users. This mode is proven to be successful in numerous geolocation-related domains in recent years (e.g. OpenStreetMap). However, it is difficult to ensure the quality of the videos or detect inappropriate contents. Another concern is privacy protection. We are working on utilizing some technologies like automatically blurring faces and license plates [24] to prevent publication of sensitive information. As we are entering an era of Virtual Reality (VR), we are also planning to deliver 3D contents via the Web.

Figure 6.8: Aircraft Simulator

Figure 6.9: Data Uploading

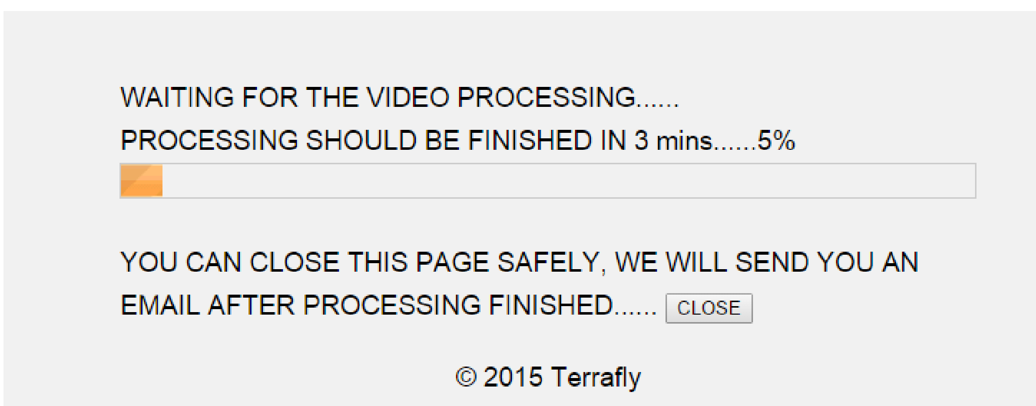


Figure 6.10: Data Processing

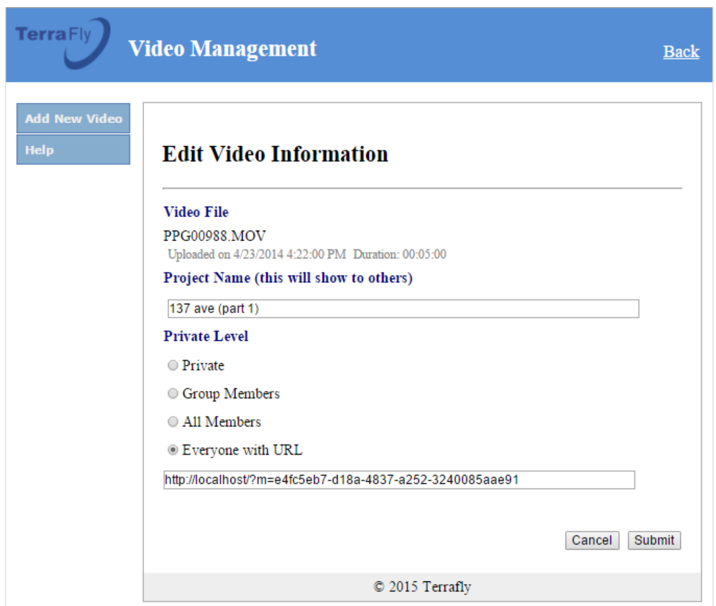


Figure 6.11: Data Management

CHAPTER 7

VISUALIZATION OF MOVING OBJECT WITH AUDIO: AUTOPILOT

7.1 Introduction

Autopilot is a core product of TerraFly that provides users the route plan and preview service [79]. Everyone may be familiar with one place while feeling unfamiliar to another place. Autopilot provides a platform for everyone to share their geography knowledge by collaborating with each other. With Autopilot, a volunteer editor can create a route with a bunch of waypoints and upload several audio guidance file. After the editing, everything is saved to server and ready to be published as a dedicated URL. The end user who opens the URL will see the map flying through the route while the guidance audio is played at the same time. This system is an example of type (4) visualization defined in the Chapter 3 for audio data type.

7.1.1 Motivations

The previous system was designed for advanced users who are familiar with coordinate system and audio transcoding. However, it is rather difficult for the average users. And it is inconvenient for ourdoor use since the tool only supports the desktop browser.

Second, the users was only able to describe a place using voice. It is desirable to give more expression powers to editors like the ability to add the images and text descriptions.

Third, the route and the audio is not synchronized very well. If the map delays at the browser of an end user, the audio will be ahead of time. In addition, the user cannot fast forward or backward to a specific location.

7.1.2 Contributions

Motivated by the problems addressed above, this chapter introduces the next version of Autopilot. Compared to the old system, the main difference is as follows:

- A new mobile phone application is developed to provide one-stop recording, editing and uploading.
- The ability to support multiple format including audio, photo and text.
- The framework has been changed based on Chapter 3 that give the user more playback control.

7.2 Background

7.2.1 Online Map for Guidance Support

Online guidance map service has a long history. [87] developed a system on hand held devices to get information based on location. [69] proposed a web-based guiding tool using Google Map. Nevertheless, most early systems only have basic image displaying and text description functions.

[36] proposed a multimedia mapping system that using mashup technology. The risk of blending multiple technologies together is that it may cause confusion and chaos. In addition, it lacks of high-level interaction and cooperation between different modules.

Because of the rapid development of smart devices, most current solutions are based on smart phones [44] [95] [70]. While they are useful for outdoor navigation and guidance, they usually lack the desktop tool to preview and plan the route.

7.2.2 TerraFly

TerraFly is a powerful yet easy to use online map platform [79]. It is fully customizable with multi-layered data visualization capabilities. TerraFly is not only a web mapping API but also a complete solution to support visualization and querying of geospatial data [80] [55] [101]. The whole system is backed by 40TB of the basemap data other user-specific data that served on TerraFly's server farm, which is optimized using autonomic resource management technology [56].

Besides normal web mapping functionalities and elements as shown in Figure 7.1, TerraFly is specialized in visualizing dynamic and moving data thanks to its animation functions. The map can fly-over through a series of waypoints, or show the object moving smoothly along a track. It can also visualize complicated scientific data using GeoCloud system [102] [96].

7.2.3 HTML5

HTML5 is a online standards that finalized and published on 28 October 2014 by the World Wide Web Consortium (W3C) [35]. As the name implies it is the fifth version of HyperText Markup Language(HTML) which is used for creating websites. The core advantage of this version over its predecessor is the ability to support latest multimedia and graphic technology, which is a good news for web mapping applications [74].

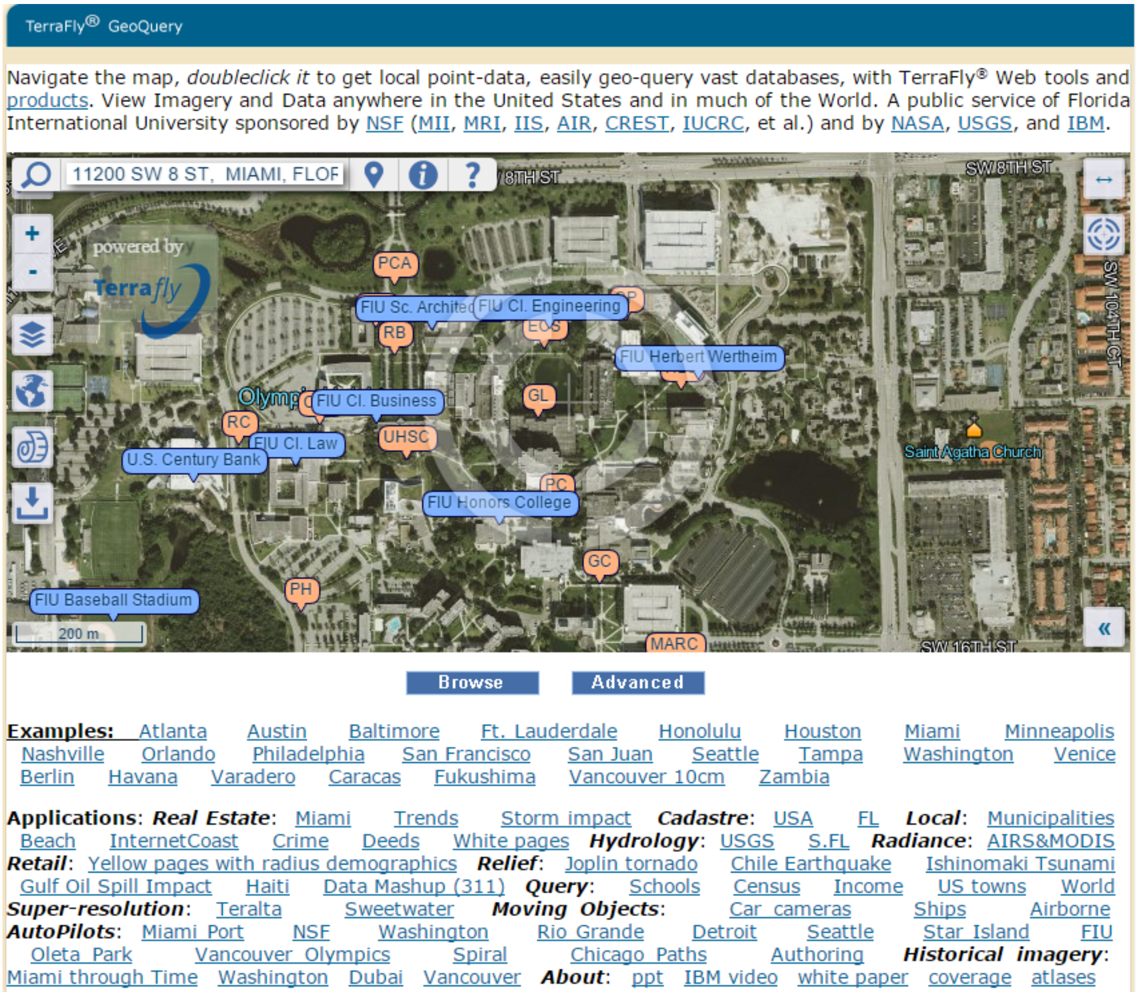


Figure 7.1: TerraFly map overview.

7.3 System Architecture

The system uses the well-known three-tier website model as shown in Figure 7.2. Every rectangular box is an independent module for enhanced reusability. This framework design is based on the proposed model in Chapter 3.

There are two versions of user interfaces on the client side: the desktop version and the mobile application version. They have similar viewer interfaces because they do the same job, except that the mobile version is optimized specially for mobile devices. However, the editor interfaces are different. The desktop version can only

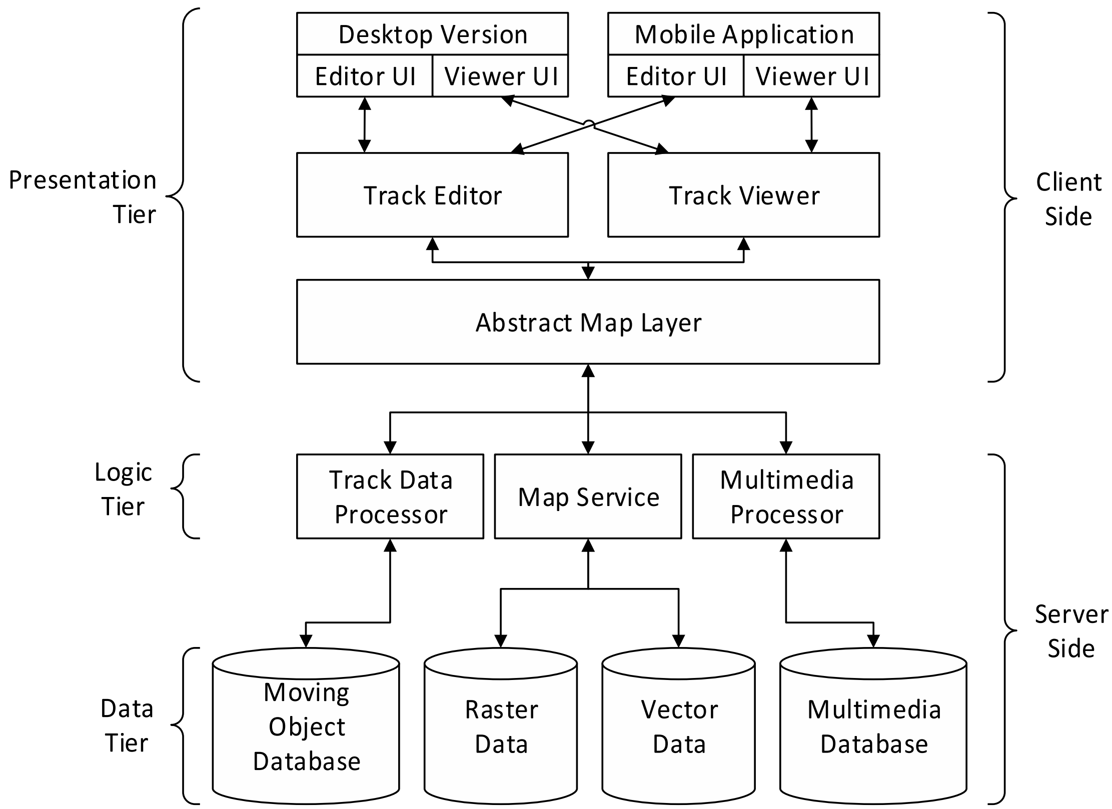


Figure 7.2: Autopilot System Architecture

add waypoints manually by clicking the map. In contrast, the mobile version can automatically add viapoints using the satellite positioning sensor data.

The server side has four databases. The TerraFly Map Service manages raster and Vector database. Track data processor stores the waypoints into historical moving object database. The multimedia processor handles photos and audios. After transcoding and compression, the audio or photo will be sent to multimedia database.

7.4 Use Case Study

This section presents the major functions of this system using some real-life scenarios.

7.4.1 The Editor

The user uses route editor to create or modify the guidance route. The user can also add supplemental material such as text descriptions, photos and audios. The map loads and displays the related multimedia materials at the proper time and location, which is predefined by the user.

Figure 7.3 shows the desktop designer. On the left side is the map window. Control window is on the top right corner, which is above the waypoint list. When the user clicks on the maps, a new waypoint will be added into the map and the list as well. The user can further define the waypoint in detail using the controls, including position, speed, and zoom operation. If necessary, the user can upload a photo or an audio, and then bound it to a waypoint.

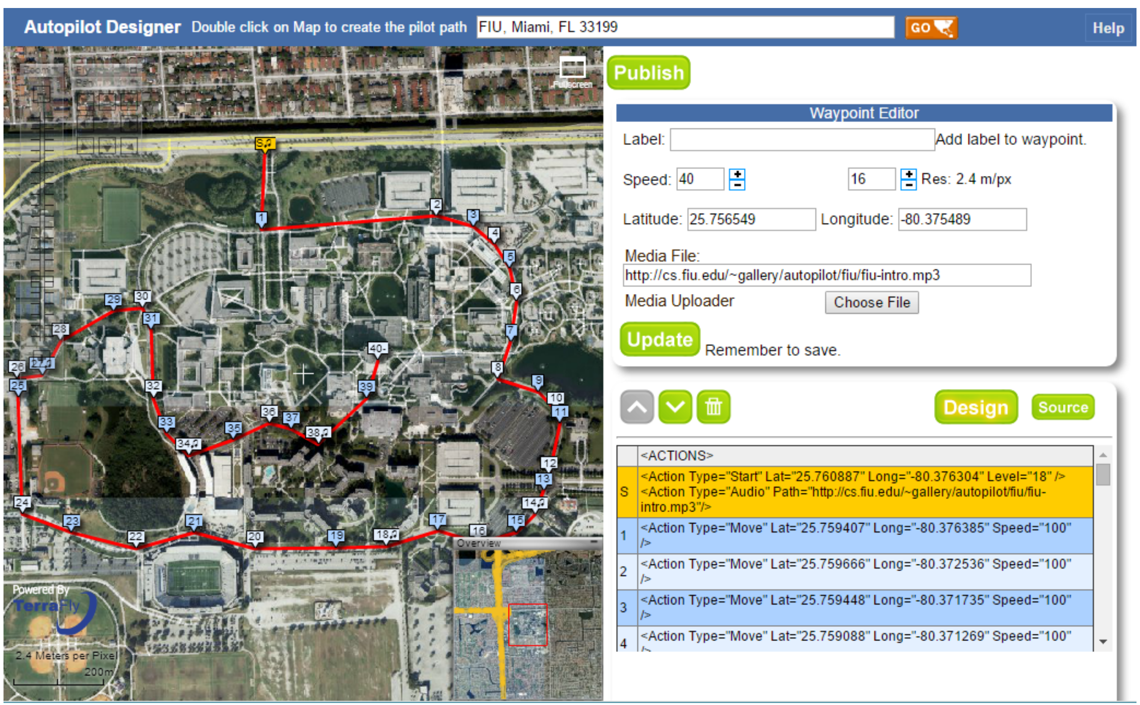


Figure 7.3: Autopilot route desktop editing tool.

Figure 7.4 shows the mobile version editor application. Different from the desktop version, the waypoints will be added to the map automatically using the mobile device's built-in GPS. The user can also using the camera and voice recorder of the device to add photo and audio in real time. This gives a great convenience to the users outdoor.

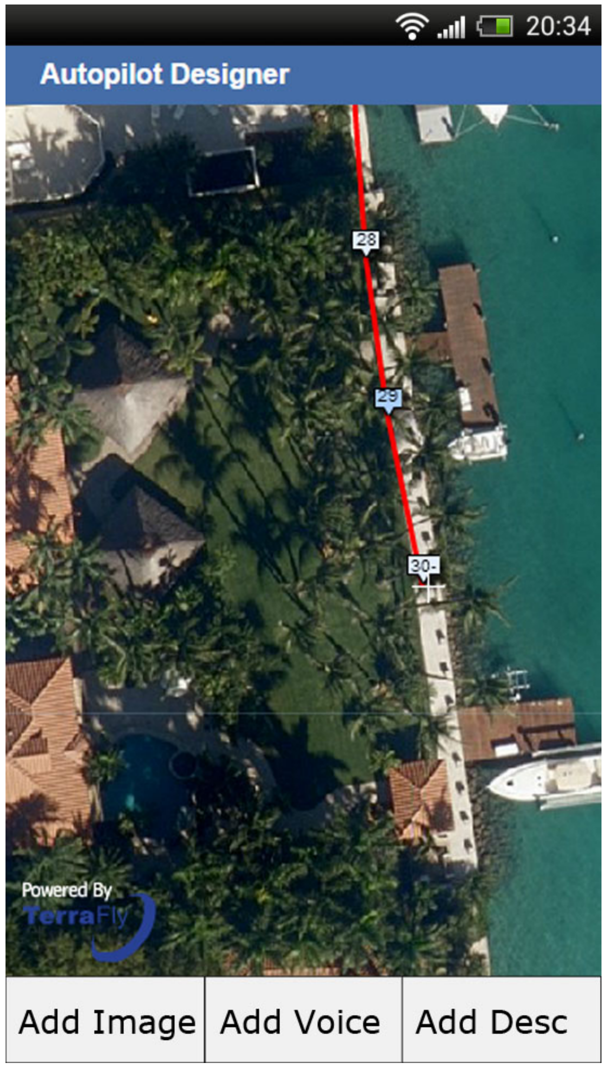


Figure 7.4: Autopilot route mobile editing tool.

After the editing is finished, the user can click the publish button to easily get a sharable URL of the online guidance, as shown in Figure 7.5. The tool will also generate the HTML code for the user who wants to embed the page into their own website. It is useful for making a mashup application quickly.

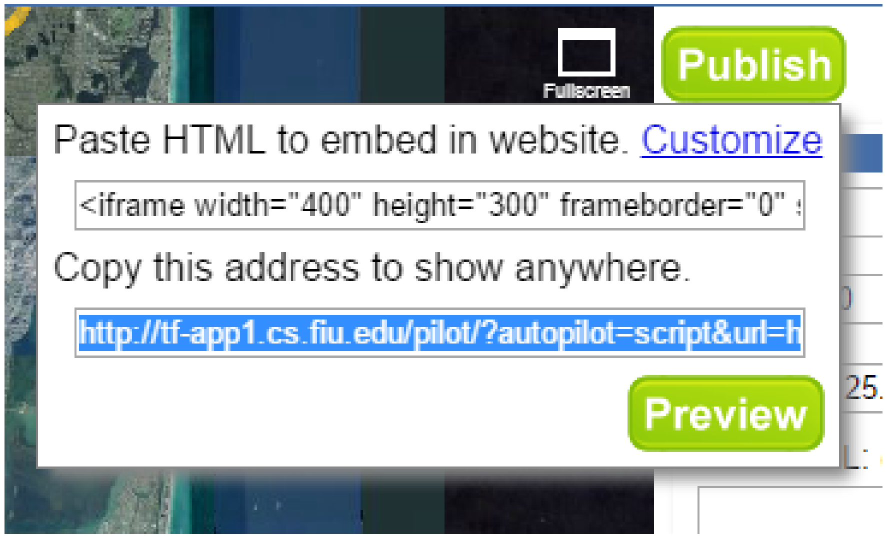


Figure 7.5: Autopilot share function.

7.4.2 The Viewer

The route viewer shows the guidance route to the end users aided by descriptions, images and audios as shown in Figure 7.6. An overview window is shown in the bottom right corder, which tells the current progress.

The audio plays at the right time that defined by the editor. The image also pops up at specific location for better understanding the local scene. The user can stop or pause the guidance at any time.

A thoughtful re-edit function is provide for the creator or viewer who wants to modify the background script, as easy as clicking the button right to the stop button.

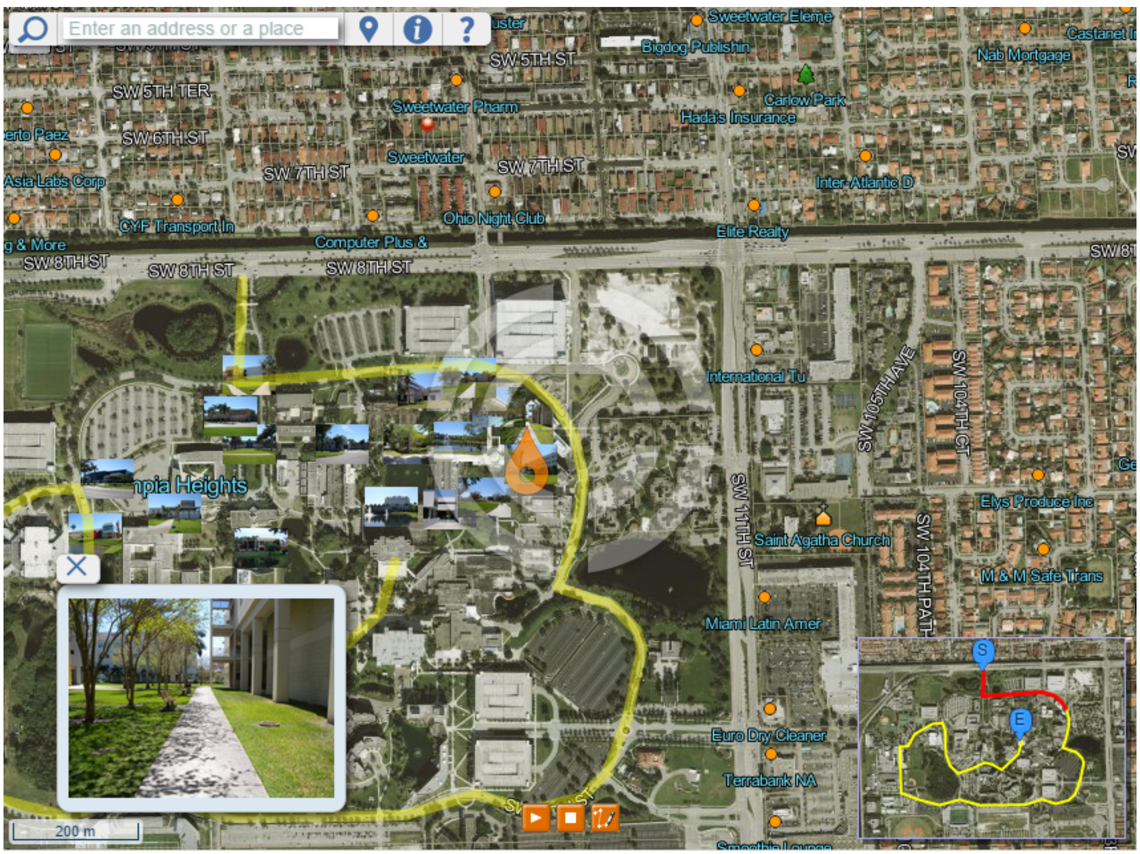


Figure 7.6: Autopilot route guidance interface.

CHAPTER 8

BIG DATA VISUALIZATION: AIS VISUALIZER

8.1 Introduction

Before 2011, GPS is the only satellite positioning system worldwide. Many new systems appear in recent years including GLONASS from Russia, Galileo from Europe union, Beidou from China, and IRNSS from India. The positioning systems are widely used in our daily life now, such as air traffic control, fire rescue, and wildlife tracking. To manage the positions of these moving objects, there are great demands for high-performance databases. However, traditional databases are not capable of handling the ultra-high update frequency of the moving objects.

The Automatic Identification System (AIS) is a ship tracking system, which send out the location of a ship automatically. This system also can prevent the collisions between vessels. It is required to be installed on vessels over 300GT and all passenger ships by the International Maritime Organization (IMO)'s International Convention for the Safety of Life at Sea [50].

8.2 Motivation

Consider there are at least 104304 ships are registered in IMO [39], the AIS data update can be very frequent. On the test server used in this dissertation, the database receives about 200 update requests per second. Moreover, all the historical data together is very huge for a normal database. It is a big challenge to manage and visualize the AIS data.

8.3 Background

There are many different ways to visualization AIS data, for different analytic purpose. To analysis vessel presence during a time span, density methods are always popular. [98] [17] [85] draw continuous density layer, while [3] and [22] represent density in discrete grids.

To visualize position-related attributes, they usually represent trajectory into segments and using attribute value to change the appearance of these segments. The change can be represented by coloring, shading or symbols [28] [48] [88] [97]. [49] uses HTML5 to visualize the vessel.

8.4 System Architecture

The architecture of the system is shown in Figure 8.1. The well-developed three tier website structure is adopted.

8.4.1 Client

The designing of client side is pretty easy, using powerful TerraFly web-mapping API. The user interface contains the map and query module. The queries are generated by the query module and sent to the server. The returned data is shown on the map, such as historical trajectories and live positions of vessels. The abstract map layer handle all the request between the UI and server.

8.4.2 Server

The AIS data service listens to the UDP (User Datagram Protocol) port continuously. The received data will be parsed and stored into live MOD temporarily.

The archiving service backs up all the live data into the historical MOD in a pre-defined interval. The data processor gets the vessel attributes and photos from ship information database. The terraFly map service is run on the raster and vector databases.

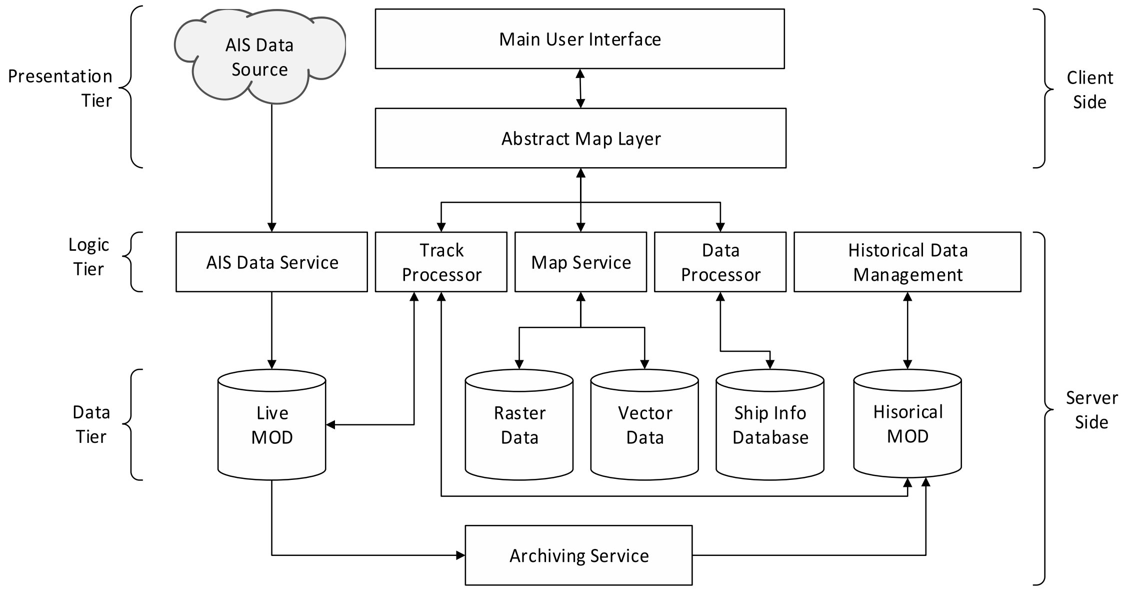


Figure 8.1: AIS system architecture

8.5 Data Handling

AIS system is data centric. The data handling work flow is shown in Figure 8.2.

8.5.1 Data Retrieval

The data receiver get the raw data stream through the Internet. The received data will be parsed and indexed into the MOD.

The AIS data is transfered to the receiver using the Time Division Multiple Access(TDMA) technology. The format of the data is the NMEA 0183/2000[1]. These

[1] https://www.nmea.org/content/nmea_standards/nmea_standards.asp

data have the introducer “!AIVDM” or “!AIVDO” that represents packets from other ships and current ship, respectively [77].

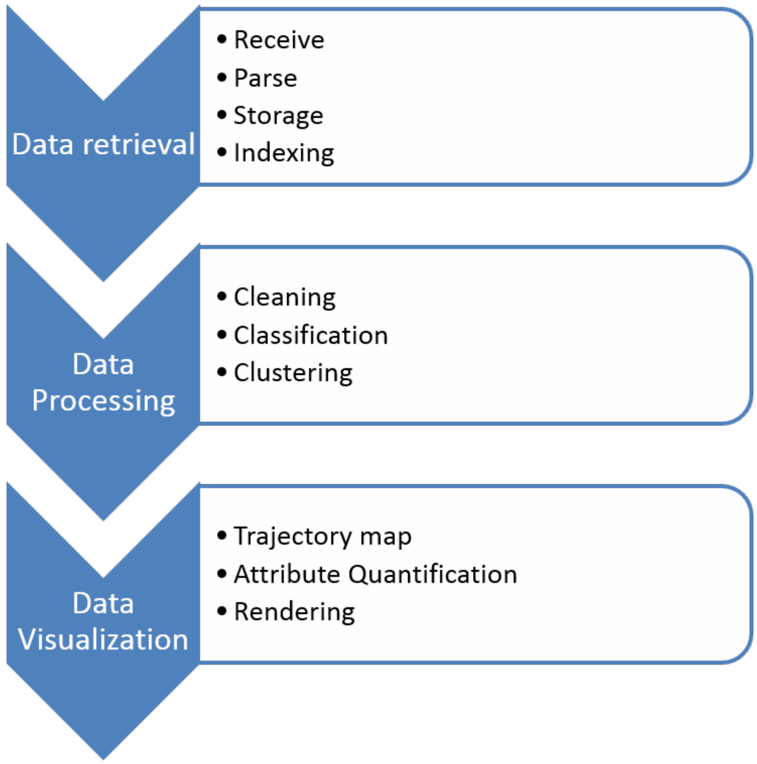


Figure 8.2: AIS system work flow

8.5.2 Data Processing

After the data retrieval stage, all the data needs to be cleaned. Sometimes the vessel generates dirty data when the GPS starts up. In addition, the redundant data is received occasionally because of transmission problems.

The historical data is later classified and clustered using the TerraFly GeoCloud service [102].

8.5.3 Data Visualization

The data is very huge considering the large quantities of vessels. Therefore, it is unrealistic to display live trajectory at global view. To better visualize the vessel movement, this system visualizes the data using different strategies based on resolution levels. At higher level, the vessels will be rendered as a density map.

8.6 Use Case Study

This dissertation uses the AIS Hub [8] as the data feed pool. The data is freely exchanged between data providers all over the world.

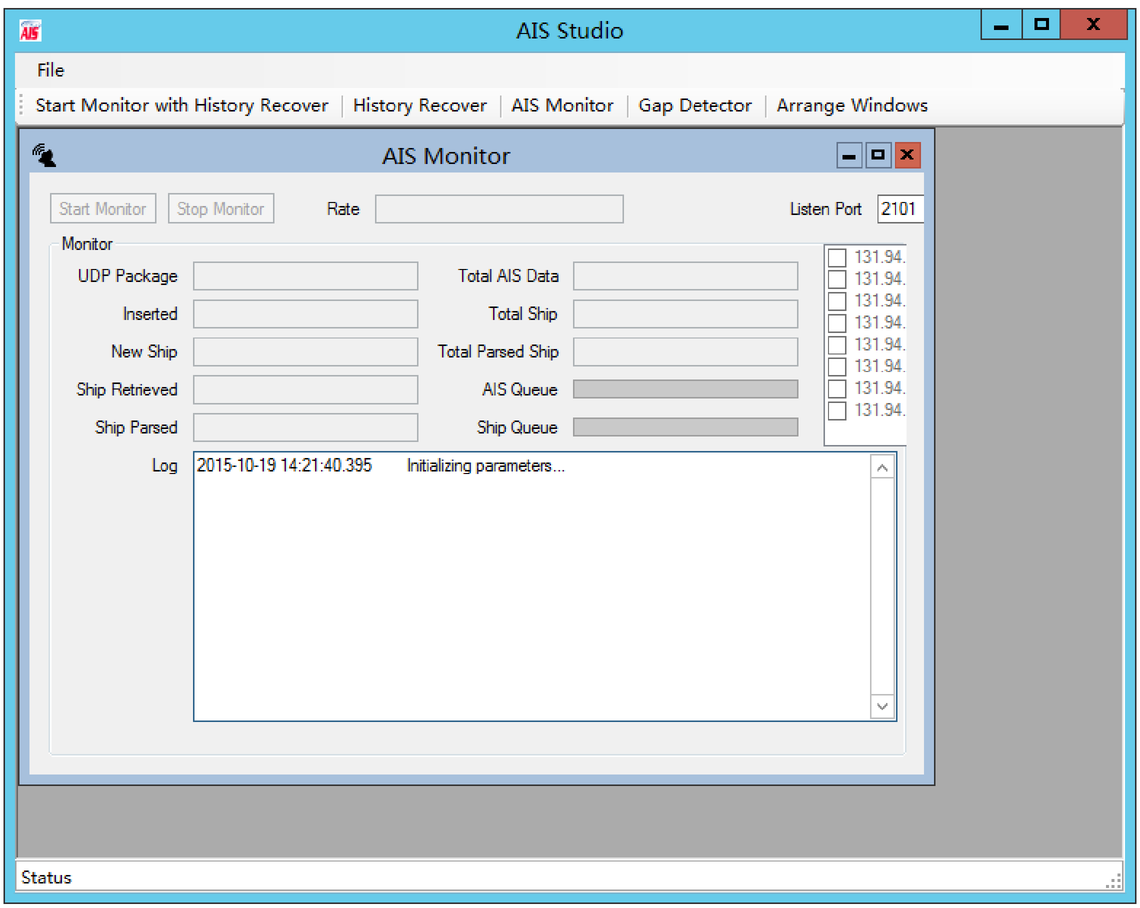


Figure 8.3: AIS data loader.

8.6.1 Data Loader

The data loader tool is shown in Figure 8.3. The data received from the UDP port will be parsed on the fly and stored into the database. The system hardware utilizes the solid-state drives (SSDs) to increase the data throughput. The data is also distributed on different virtual servers for better practical performance and stability. These virtual machines are automatically managed using the v-TerraFly technology [56].

8.6.2 Global Overview

The global overview UI is shown in Figure 8.4. The density map of vessels is calculated on the server side every five minutes and pre-rendered as transparent PNG (Portable Network Graphics) images. Benefited from this idea, the map rendering performance at the client side is very fast.

Figure 8.4: AIS global hotspot view.

8.6.3 Local Overview

The local live and historical data visualization UI is shown in Figure 8.5. The yellow arrow shows a vessel's location and course. The historical trajectory of selected time lapse is shown as polyline right behind the arrow. These trajectories are rendered in varies colors to represent the different types of vessel, such as tankers, bulk carriers, container ships and passenger ships.

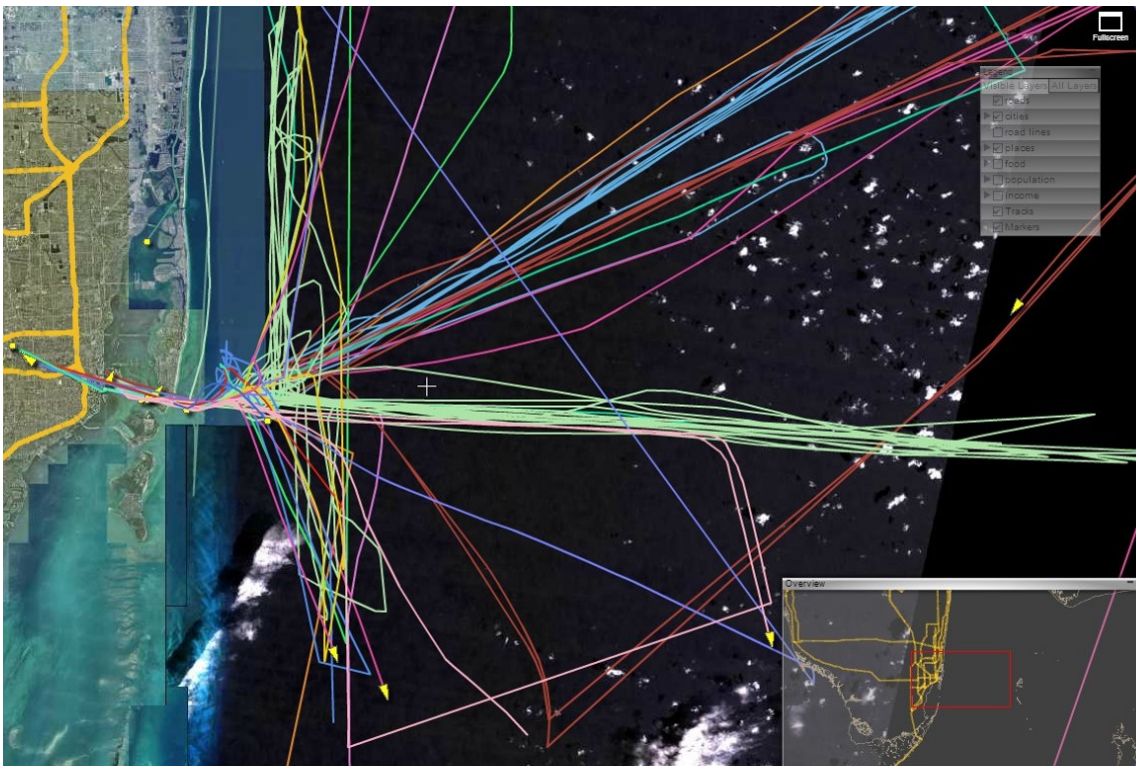


Figure 8.5: AIS local trajectory view.

CHAPTER 9

CONCLUSION

9.1 Summary

In recent years we have witnessed the vigorous development online map service. As the smart mobile devices are essential in people's every day lives [82], the collaborative mapping and crowdsourcing become the new direction of online map. Most of the data generated by these devices is georeferenced. After a decade of research, the moving object database becomes practical for support these georeferenced data. However, little research has been conducted on visualization of these data along with their trajectories.

This dissertation proposed a data model for describing the georeferenced data and geolocations of moving objects. Four different visualization tasks are addressed with some sample scenarios. The model also reveals the relationship and interactions between the location and data attributes.

An abstract map layer is given for visualization task based on the above model.

A general moving object visualization framework is proposed for quick design and realize real life georeferenced data visualization tasks.

Four systems have been developed based on the framework covering many common aspects during moving object visualization such as:

- Data acquisition time: live and historical
- Georeferenced data type: image, audio, video and numerical data
- Moving object type: pedestrian, automobile, ship and aircraft

9.2 Future Work

One direction of the future work is to extend this platform to mobile devices. Although currently the web-based map supports the browser in most mobile devices, it did not use the full potential of these devices. The mobile devices have more ways to interact with (e.g.: touch gestures). Further more, a dedicated application can have better graphics performance than the web browser.

BIBLIOGRAPHY

[1] Jhummarwala Abdul, MB Potdar, and Prashant Chauhan. Parallel and distributed gis for processing geo-data: An overview. *International Journal of Computer Applications*, 106(16), 2014.

[2] Saeed Aghaee and Cesare Pautasso. Mashup development with html5. In *Proceedings of the 3rd and 4th International Workshop on Web APIs and Services Mashups*, page 10. ACM, 2010.

[3] Gennady Andrienko and Natalia Andrienko. A general framework for using aggregation in visual exploration of movement data. *The Cartographic Journal*, 47(1):22–40, 2010.

[4] Dragomir Anguelov, Carole Dulong, Daniel Filip, Christian Frueh, Stéphane Lafon, Richard Lyon, Abhijit Ogale, Luc Vincent, and Josh Weaver. Google street view: Capturing the world at street level. *Computer*, (6):32–38, 2010.

[5] Stan Aronoff. Geographic information systems: a management perspective. 1989.

[6] Mikhail J. Atallah, Danny Z. Chen, and DT Lee. An optimal algorithm for shortest paths on weighted interval and circular-arc graphs, with applications. *Algorithmica*, 14(5):429–441, 1995.

[7] Daniel Caldeweyher, Jinglan Zhang, and Binh Pham. Opencis - open source gis-based web community information system. *International Journal of Geographical Information Science*, 20(8):885–898, 2006.

[8] Fulya T Cetin, Burcu Yilmaz, Yildiray Kabak, Ju-Hwan Lee, Cengiz Erbas, Erdem Akagunduz, and Sang-Jae Lee. Increasing maritime situational awareness with interoperating distributed information sources. Technical report, DTIC Document, 2013.

[9] Noppadol Chadil, Apirak Russameesawang, and Phongsak Keeratiwintakorn. Real-time tracking management system using gps, gprs and google earth. In *Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology, 2008. ECTI-CON 2008. 5th International Conference on*, volume 1, pages 393–396. IEEE, 2008.

[10] Billy Chen, Boris Neubert, Eyal Ofek, Oliver Deussen, and Michael F Cohen. Integrated videos and maps for driving directions. In *Proceedings of the 22nd*

annual ACM symposium on User interface software and technology, pages 223–232. ACM, 2009.

[11] Shao-Chi Chen, Hsin-Yi Chen, Yi-Ling Chen, Hsin-Mu Tsai, and Bing-Yu Chen. Making in-front-of cars transparent: Sharing first-person-views via dashcam. In *Computer Graphics Forum*, volume 33, pages 289–297. Wiley Online Library, 2014.

[12] Chuan-Yen Chiang, Shyan-Ming Yuan, Shian-Bo Yang, Guo-Heng Luo, and Yen-Lin Chen. Vehicle driving video sharing and search framework based on gps data. In *Genetic and Evolutionary Computing*, pages 389–397. Springer, 2014.

[13] Victor Teixeira De Almeida and Ralf Hartmut Güting. Indexing the trajectories of moving objects in networks*. *GeoInformatica*, 9(1):33–60, 2005.

[14] Edsger W Dijkstra. A note on two problems in connexion with graphs. *Numerische mathematik*, 1(1):269–271, 1959.

[15] Christian Düntgen, Thomas Behr, and Ralf Hartmut Güting. Berlinmod: a benchmark for moving object databases. *The VLDB Journal*, 18(6):1335–1368, 2009.

[16] Ross G Dwyer, Charles Brooking, Wilfred Brimblecombe, Hamish A Campbell, Jane Hunter, Matthew Watts, and Craig E Franklin. An open web-based system for the analysis and sharing and analysis of animal tracking data. *Animal Biotelemetry*, 3(1):1–11, 2015.

[17] Jason A Dykes and David M Mountain. Seeking structure in records of spatio-temporal behaviour: visualization issues, efforts and applications. *Computational Statistics & Data Analysis*, 43(4):581–603, 2003.

[18] T Iype Eldho, AT Kulkarni, J Mohanty, EP Rao, and BK Mohan. A web gis based simulation tool for coastal urban flood prediction. 2014.

[19] Ying Fang, Jiaheng Cao, Yuwei Peng, and Liwei Wang. Indexing the past, present and future positions of moving objects on fixed networks. In *Computer Science and Software Engineering, 2008 International Conference on*, volume 4, pages 524–527. IEEE, 2008.

[20] Lei Feng, Chaoliang Wang, Chuanrong Li, and Ziyang Li. A research for 3d webgis based on webgl. In *Computer Science and Network Technology (ICC-*

SNT), 2011 International Conference on, volume 1, pages 348–351. IEEE, 2011.

[21] Raphael A. Finkel and Jon Louis Bentley. Quad trees a data structure for retrieval on composite keys. *Acta informatica*, 4(1):1–9, 1974.

[22] Pip Forer and Otto Huisman. Space, time and sequencing: Substitution at the physical/virtual interface. In *Information, Place, and Cyberspace*, pages 73–90. Springer, 2000.

[23] Steffen Fritz, Ian McCallum, Christian Schill, Christoph Perger, Roland Grillmayer, Frédéric Achard, Florian Kraxner, and Michael Obersteiner. Geo-wiki. org: The use of crowdsourcing to improve global land cover. *Remote Sensing*, 1(3):345–354, 2009.

[24] Andrea Frome, German Cheung, Ahmad Abdulkader, Marco Zennaro, Bo Wu, Alessandro Bissacco, Hartwig Adam, Hartmut Neven, and Luc Vincent. Large-scale privacy protection in google street view. In *Computer Vision, 2009 IEEE 12th International Conference on*, pages 2373–2380. IEEE, 2009.

[25] Pinde Fu and Jiulin Sun. *Web GIS: principles and applications.* Esri Press, 2010.

[26] Julien Gaffuri. Toward web mapping with vector data. In *Geographic Information Science*, pages 87–101. Springer, 2012.

[27] CY Goh, J Dauwels, N Mitrovic, MT Asif, A Oran, and P Jaillet. Online map-matching based on hidden markov model for real-time traffic sensing applications. In *Intelligent Transportation Systems (ITSC), 2012 15th International IEEE Conference on*, pages 776–781. IEEE, 2012.

[28] Hanqi Guo, Zuchao Wang, Bowen Yu, Huijing Zhao, and Xiaoru Yuan. Tripvista: Triple perspective visual trajectory analytics and its application on microscopic traffic data at a road intersection. In *Pacific Visualization Symposium (PacificVis), 2011 IEEE*, pages 163–170. IEEE, 2011.

[29] Antonin Guttman. *R-trees: a dynamic index structure for spatial searching*, volume 14. ACM, 1984.

[30] Stefan Hahmann and Dirk Burghardt. How much information is geospatially referenced? networks and cognition. *International Journal of Geographical Information Science*, 27(6):1171–1189, 2013.

[31] Mark Harrower and Sara Fabrikant. The role of map animation for geographic visualization. *Geographic visualization*, pages 49–65, 2008.

[32] Mark Harrower, Alan MacEachren, and Amy L Griffin. Developing a geographic visualization tool to support earth science learning. *Cartography and Geographic Information Science*, 27(4):279–293, 2000.

[33] Peter E Hart, Nils J Nilsson, and Bertram Raphael. A formal basis for the heuristic determination of minimum cost paths. *Systems Science and Cybernetics, IEEE Transactions on*, 4(2):100–107, 1968.

[34] Tomislav Hengl, Pierre Roudier, Dylan Beaudette, and Edzer Pebesma. plotkml: scientific visualization of spatio-temporal data. *Journal of Statistical Software*, 63(5), 2015.

[35] I Hickson, R Berjon, S Faulkner, T Leithead, E Doyle Navara, and E Oonnor. S. pfeiffer," html5", w3c recommendation rec-html5-20141028, october 2014.

[36] Shunfu Hu. Multimedia mapping on the internet using commercial apis. In *Online Maps with APIs and WebServices*, pages 61–71. Springer, 2012.

[37] Britta Hummel. Map matching for vehicle guidance. *Dynamic and Mobile GIS: Investigating Space and Time. CRC Press: Florida*, 2006.

[38] Takahiro Ikeda, Min-Yao Hsu, Hiroshi Imai, Shigeki Nishimura, Hiroshi Shimoura, Takeo Hashimoto, Kenji Tenmoku, and Kunihiko Mitoh. A fast algorithm for finding better routes by ai search techniques. In *Vehicle Navigation and Information Systems Conference, 1994. Proceedings., 1994*, pages 291–296. IEEE, 1994.

[39] Craig Jallal. How many ships are there in the world? https://shippingresearch.wordpress.com/2012/07/31/how-many-ships-are-there-in-the-world/, JULY 2012. Accessed: 10/19/2015.

[40] Christian S Jensen, Dan Lin, and Beng Chin Ooi. Query and update efficient b+-tree based indexing of moving objects. In *Proceedings of the Thirtieth international conference on Very large data bases-Volume 30*, pages 768–779. VLDB Endowment, 2004.

[41] Christian S Jensen, Dalia Tiešytė, and Nerius Tradišauskas. The cost benchmarkomparison and evaluation of spatio-temporal indexes. In *Database Systems for Advanced Applications*, pages 125–140. Springer, 2006.

[42] Hoyoung Jeung, Qing Liu, Heng Tao Shen, and Xiaofang Zhou. A hybrid prediction model for moving objects. In *Data Engineering, 2008. ICDE 2008. IEEE 24th International Conference on*, pages 70–79. IEEE, 2008.

[43] Anton V Jouline, William E Reed, Ryan Massie, Gary Chevsky, and Michiel Frishert. Map interface with a movable marker, April 5 2011. US Patent 7,920,963.

[44] Michael Kenteris, Damianos Gavalas, and Daphne Economou. Mytilene e-guide: a multiplatform mobile application tourist guide exemplar. *Multimedia Tools and Applications*, 54(2):241–262, 2011.

[45] Johannes Kopf, Billy Chen, Richard Szeliski, and Michael Cohen. Street slide: browsing street level imagery. In *ACM Transactions on Graphics (TOG)*, volume 29, page 96. ACM, 2010.

[46] Menno-Jan Kraak. Settings and needs for web cartography. *Web cartography: Developments and prospects*, pages 1–7, 2001.

[47] Menno-Jan Kraak. The role of the map in a web-gis environment. *Journal of Geographical Systems*, 6(2):83–93, 2004.

[48] Menno-Jan Kraak and Otto Huisman. 17 beyond exploratory visualization of space–time paths. *Geographic data mining and knowledge discovery*, page 431, 2009.

[49] Seo-Jeong Lee and Jae-Wook Lee. A development of ais vessel monitoring system on online map using html5. *Journal of Korean navigation and port research*, 35(6):463–467, 2011.

[50] MJ Lewandowski and DJ Pietraszewski. Automatic identification system; a general discussion of development, application, and implementation. *USCG R&D Center, junij*, 2000.

[51] Songnian Li. Web mapping/gis services and applications. *Advances in photogrammetry, remote sensing and spatial information sciences*, pages 335–353, 2008.

[52] Songnian Li, Chenfeng Xiong, and Ziqiang Ou. A web gis for sea ice information and an ice service archive. *Transactions in GIS*, 15(2):189–211, 2011.

[53] Dan Lin, Christian S Jensen, Beng Chin Ooi, and Simonas Šaltenis. Efficient indexing of the historical, present, and future positions of moving objects. In *Proceedings of the 6th international conference on Mobile data management*, pages 59–66. ACM, 2005.

[54] Yin Lou, Chengyang Zhang, Yu Zheng, Xing Xie, Wei Wang, and Yan Huang. Map-matching for low-sampling-rate gps trajectories. In *Proceedings of the 17th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*, pages 352–361. ACM, 2009.

[55] Yun Lu, Mingjin Zhang, Shonda Witherspoon, Yelena Yesha, and Naphtali Rishe. Sksopen: Efficient indexing, querying, and visualization of geo-spatial big data. In *Machine Learning and Applications (ICMLA), 2013 12th International Conference on*, volume 2, pages 495–500. IEEE, 2013.

[56] Yun Lu, Ming Zhao, Lixi Wang, and Naphtali Rishe. v-terrafly: large scale distributed spatial data visualization with autonomic resource management. *Journal of Big Data*, 1(1):1–19, 2014.

[57] Yun Lu, Ming Zhao, Guangqiang Zhao, Lixi Wang, and Naphtali Rishe. Massive gis database system with autonomic resource management. In *Machine Learning and Applications (ICMLA), 2013 12th International Conference on*, volume 2, pages 451–456. IEEE, 2013.

[58] Jiebo Luo, Dhiraj Joshi, Jie Yu, and Andrew Gallagher. Geotagging in multimedia and computer vision-a survey. *Multimedia Tools and Applications*, 51(1):187–211, 2011.

[59] Bo Mao, Zhiang Wu, and Jie Cao. A framework for online spatio-temporal data visualization based on html5. In *Proceedings of the International Conference on Photogrammetry, Remote Sensing and Spatial Information Sciences,(XXII ISPRS Congress, Melbourne, Australia, 25 August–01 September, 2012)*, pages 123–127, 2012.

[60] Alan Lowe McConchie. *Mapping mashups: Participation, collaboration and critique on the World Wide Web.* PhD thesis, UNIVERSITY OF BRITISH COLUMBIA (Vancouver, 2008.

[61] Alexey Medvedev, Arkady Zaslavsky, Vladimir Grudinin, and Sergey Khoruzhnikov. Citywatcher: Annotating and searching video data streams for smart cities applications. In *Internet of Things, Smart Spaces, and Next Generation Networks and Systems*, pages 144–155. Springer, 2014.

[62] Tyler Mitchell. *Web mapping illustrated: using open source GIS toolkits.* " O'Reilly Media, Inc.", 2005.

[63] Antoni Moore. Web cartography–map design for interactive and mobile devices. *Journal of Spatial Science*, 60(1):195–196, 2015.

[64] Pascal Neis, Dennis Zielstra, and Alexander Zipf. The street network evolution of crowdsourced maps: Openstreetmap in germany 2007–2011. *Future Internet*, 4(1):1–21, 2011.

[65] Andreas Neumann. Web mapping and web cartography. In *Encyclopedia of GIS*, pages 1261–1269. Springer, 2008.

[66] Paul Newson and John Krumm. Hidden markov map matching through noise and sparseness. In *Proceedings of the 17th ACM SIGSPATIAL international conference on advances in geographic information systems*, pages 336–343. ACM, 2009.

[67] Long-Van Nguyen-Dinh, Walid G Aref, and Mohamed Mokbel. Spatio-temporal access methods: Part 2 (2003-2010). 2010.

[68] Jürg Nievergelt, Hans Hinterberger, and Kenneth C Sevcik. *The grid file: An adaptable, symmetric multi-key file structure.* Springer, 1981.

[69] Bing Pan, John C Crotts, and Brian Muller. Developing web-based tourist information tools using google map. *Information and Communication Technologies in Tourism 2007*, pages 503–512, 2007.

[70] Rohit R Pardeshi, Sandip N Patil, Sharmeen M Mukadam, and Shivani Sthapak. Mobile travel guide-a smart way to travel. *IJECCE*, 4(2):482–484, 2013.

[71] Jignesh M Patel, Yun Chen, and V Prasad Chakka. Stripes: an efficient index for predicted trajectories. In *Proceedings of the 2004 ACM SIGMOD international conference on Management of data*, pages 635–646. ACM, 2004.

[72] Chi Peng, Bing-Yu Chen, and Chi-Hung Tsai. Integrated google maps and smooth street view videos for route planning. In *Computer Symposium (ICS), 2010 International*, pages 319–324. IEEE, 2010.

[73] Oliver Pink and Britta Hummel. A statistical approach to map matching using road network geometry, topology and vehicular motion constraints. In *In-*

telligent Transportation Systems, 2008. ITSC 2008. 11th International IEEE Conference on, pages 862–867. IEEE, 2008.

[74] Zhiyong Qi. Geovisualization using html5: a case study to improve animations of historical geographic data. *Student thesis series INES*, 2013.

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

[76] Jens Eilstrup Rasmussen, Lars Eilstrup Rasmussen, Bret Steven Taylor, James Christopher Norris, Stephen Ma, Andrew Robert Kirmse, Noel Phillip Gordon, and Seth Michael Laforge. Digital mapping system, January 2 2007. US Patent 7,158,878.

[77] Eric S Raymond. Aivdm/aivdo protocol decoding. *GPSD documentation Version*, 1, 2009.

[78] Naphtali Rishe, Shu-Ching Chen, Nagarajan Prabakar, Mark Allen Weiss, Wei Sun, Andriy Selivonenko, and D Davis-Chu. Terrafly: A high-performance web-based digital library system for spatial data access. In *ICDE Demo Sessions*, pages 17–19, 2001.

[79] Naphtali Rishe, Borko Furht, Malek Adjouadi, Armando Barreto, Evgenia Cheremisina, Debra Davis, Ouri Wolfson, Nabil Adam, Yelena Yesha, and Yaacov Yesha. Geospatial data management with terrafly. In *Handbook of Data Intensive Computing*, pages 637–665. Springer, 2011.

[80] Naphtali Rishe, Christine Lisetti, and Debra Davis. Terrafly and affective computing agents for epidemiology: Web-based visualization and analysis for public health and safety. *Epidemiology*, page 1, 2011.

[81] Jorge Rodríguez, Alexis Quesada-Arencibia, David Horat, and Eduardo Quevedo. Web georeferenced video player with super-resolution screenshot feature. In *Computer Aided Systems Theory-EUROCAST 2013*, pages 87–92. Springer, 2013.

[82] Salesforce.com. 2014 mobile behavior report. [Online] Available: https://www.exacttarget.com/sites/exacttarget/files/deliverables/etmc-2014mobilebehaviorreport.pdf, 2014.

[83] Simonas Šaltenis, Christian S Jensen, Scott T Leutenegger, and Mario A Lopez. *Indexing the positions of continuously moving objects*, volume 29. ACM, 2000.

[84] Ramin Samadani and Michael Harville. Systems and methods of viewing, modifying, and interacting with "path-enhanced" multimedia, June 14 2005. US Patent 6,906,643.

[85] Roeland Scheepens, Christophe Hurter, Huub Van de Wetering, and Jarke Van Wijk. Visualization, selection, and analysis of traffic flows. 2015.

[86] Joe Schwartz. Bing maps tile system. Microsoft Developer network Available: http://msdn.microsoft.com/en-us/library/bb259689.aspx, 2009. Accessed: 2015-10-12.

[87] Todd Simcock, Stephen Peter Hillenbrand, and Bruce H Thomas. Developing a location based tourist guide application. In *Proceedings of the Australasian information security workshop conference on ACSW frontiers 2003-Volume 21*, pages 177–183. Australian Computer Society, Inc., 2003.

[88] David Spretke, Peter Bak, Halldor Janetzko, Bart Kranstauber, Florian Mansmann, and Sarah Davidson. Exploration through enrichment: a visual analytics approach for animal movement. In *Proceedings of the 19th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*, pages 421–424. ACM, 2011.

[89] Stefan Steiniger and Andrew JS Hunter. The 2012 free and open source gis software map–a guide to facilitate research, development, and adoption. *Computers, Environment and Urban Systems*, 39:136–150, 2013.

[90] Yufei Tao, Christos Faloutsos, Dimitris Papadias, and Bin Liu. Prediction and indexing of moving objects with unknown motion patterns. In *Proceedings of the 2004 ACM SIGMOD international conference on Management of data*, pages 611–622. ACM, 2004.

[91] Mats Taraldsvik. Exploring the future: is html5 the solution for gis applications on the world wide web? *Norvegian University of science and technology, Department of civil and transport technology*, 2011.

[92] Michalis Vazirgiannis and Ouri Wolfson. A spatiotemporal model and language for moving objects on road networks. In *Advances in Spatial and Temporal Databases*, pages 20–35. Springer, 2001.

[93] Luc Vincent. Taking online maps down to street level. *Computer*, 40(12):118–120, 2007.

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

[95] Dan Wang and Zheng Xiang. The new landscape of travel: A comprehensive analysis of smartphone apps. *Information and communication technologies in tourism*, 2012:308–319, 2012.

[96] Huibo Wang, Yun Lu, Yudong Guang, Erik Edrosa, Mingjin Zhang, Raul Camarca, Yelena Yesha, Tajana Lucic, and Naphtali Rishe. Epidemiological data analysis in terrafly geo-spatial cloud. In *Machine Learning and Applications (ICMLA), 2013 12th International Conference on*, volume 2, pages 485–490. IEEE, 2013.

[97] Colin Ware, Roland Arsenault, Matthew Plumlee, and David Wiley. Visualizing the underwater behavior of humpback whales. *IEEE Computer Graphics and Applications*, (4):14–18, 2006.

[98] Niels Willems, Huub Van De Wetering, and Jarke J Van Wijk. Visualization of vessel movements. In *Computer Graphics Forum*, volume 28, pages 959–966. Wiley Online Library, 2009.

[99] Ouri Wolfson, Benwei Xu, Sam Chamberlain, and Liqin Jiang. Moving objects databases: Issues and solutions. In *Scientific and Statistical Database Management, 1998. Proceedings. Tenth International Conference on*, pages 111–122. IEEE, 1998.

[100] Man Lung Yiu, Yufei Tao, and Nikos Mamoulis. The b dual-tree: indexing moving objects by space filling curves in the dual space. *The VLDB Journal*, 17(3):379–400, 2008.

[101] Chunqiu Zeng, Hongtai Li, Huibo Wang, Yudong Guang, Chang Liu, Tao Li, Mingjin Zhang, Shu-Ching Chen, and Naphtali Rishe. Optimizing online spatial data analysis with sequential query patterns. In *Information Reuse and Integration (IRI), 2014 IEEE 15th International Conference on*, pages 253–260. IEEE, 2014.

[102] Mingjin Zhang, Huibo Wang, Yun Lu, Tao Li, Yudong Guang, Chang Liu, Erik Edrosa, Hongtai Li, and Naphtali Rishe. Terrafly geocloud: An online

spatial data analysis and visualization system. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 6(3):34, 2015.

[103] Guangqiang Zhao, Mingjin Zhang, Tao Li, Shu-ching Chen, and Naphtali Rishe. City recorder: Virtual city tour using geo-referenced videos. In *2015 IEEE International Conference on Information Reuse and Integration (IRI)*, 2015.

[104] Guangqiang Zhao, Mingjin Zhang, Tao Li, Shu-Ching Chen, Ouri Wolfson, and Naphtali Rishe. Moving video mapper and city recorder with geo-referenced videos. *The Sixteenth International Conference on Web Information System Engineering (WISE), Miami, FL*, 2015.

[105] Yan-Tao Zheng, Zheng-Jun Zha, and Tat-Seng Chua. Research and applications on georeferenced multimedia: a survey. *Multimedia Tools and Applications*, 51(1):77–98, 2011.

[106] Yu Zheng, Longhao Wang, Ruochi Zhang, Xing Xie, and Wei-Ying Ma. Geolife: Managing and understanding your past life over maps. In *Mobile Data Management, 2008. MDM'08. 9th International Conference on*, pages 211–212. IEEE, 2008.

VITA

GUANGQIANG ZHAO

1983	Born, Taiyuan, China
2006	B.S., Software Engineering Beihang University Beijing, China
2007–2015	Graduate Research Assistant Florida International University Miami, FL, USA
2015	Doctoral Candidate in Computer Science Florida International University Miami, FL, USA

PUBLICATIONS AND PRESENTATIONS

Guangqiang Zhao, Mingjin Zhang, Tao Li, Shu-ching Chen, and Naphtali Rishe. "City recorder: Virtual city tour using geo-referenced videos." In *2015 IEEE International Conference on Information Reuse and Integration (IRI)*, 2015.

Guangqiang Zhao, Mingjin Zhang, Tao Li, Shu-Ching Chen, Ouri Wolfson, and Naphtali Rishe. "Moving video mapper and city recorder with geo-referenced videos." In *The Sixteenth International Conference on Web Information System Engineering (WISE)*, 2015.

Lu, Yun, Ming Zhao, Guangqiang Zhao, Lixi Wang, and Naphtali Rishe. "Massive GIS Database System with Autonomic Resource Management." In *Machine Learning and Applications (ICMLA), 2013 12th International Conference on*, vol. 2, pp. 451-456. IEEE, 2013.