2015 IEEE 16th International Conference on Information Reuse and Integration

City Recorder: Virtual City Tour Using Geo-Referenced Videos

Guangqiang Zhao, Mingjin Zhang, Tao Li, Shu-ching Chen and Naphtali Rishe
School of Computing and Information Sciences,
Florida International University, Miami, Florida, 33199, USA
Email: `{gzhao002,zhangm,taoli,chens,rishen}@cs.fiu.edu`

***Abstract*—Dashboard cameras are increasingly used these days worldwide to record driving videos. These devices generate a vast amount of geo-referenced videos. However, most of this valuable data is lost due to loop recording. We present the City Recorder, a platform to provide street level video tours based on user-uploaded driving videos. We demonstrate here how to get a smooth route previewing experience using the best available videos. We show several use cases in a real urban environment.**

***Keywords*-Digital City; Route Preview; Geo-Referenced Video; Online Map; Dashboard Camera**

I. 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 geo-tagged multimedia can provide the users an immersive experience. Many existing online map services include geo-tagged photos and videos of certain places, either uploaded by users or collected by the provider [1].

In recent years, dashboard cameras (dashcam) are widely used and their numbers are increasing rapidly [2]. Many of these videos are geo-tagged 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 [2] or mining life patterns [3].

This paper proposes the City Recorder, a framework 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 able to recognize the different brands of dashboard cameras and support various other data sources like 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.
- Integrate various kinds of privacy protection techniques into one system

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

II. RELATED WORK

Displaying street-level scenery on a map is not a new idea. Many products on the market can display geo-tagged multimedia on a map [1][4], 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 [5]. 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 [6][7][8]. 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 [9] focuses on combining a Web-based video player and a map together to play dedicated videos with positions shown on the map. Citywatcher [2] lets users annotate dashcam videos

978-1-4673-6656-4/15 $31.00 © 2015 IEEE
DOI 10.1109/IRI.2015.53

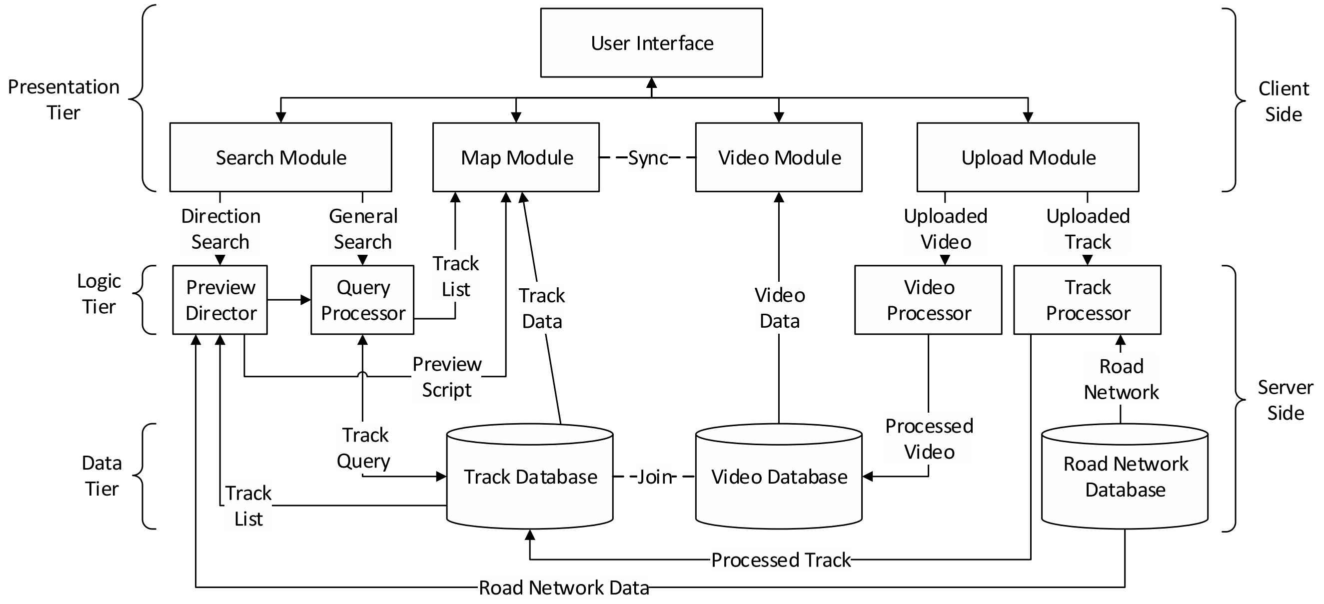


Figure 1: City Recorder System Architecture

and upload them to the City Manager. Chiang et al. [10] 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.

III. 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 1, the presentation tier resides on the client side while the logic tier and data tier run at the server side.

A. 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 [11] 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.

B. 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 V will explain this algorithm in depth.

---

[1] http://jplayer.org/

C. 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. 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) data are basic part of the Map Module and are not separately listed in the Figure 1.

IV. DATA COLLECTION

A. Track Data Processor

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

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 geo-tag 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.

B. 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 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.

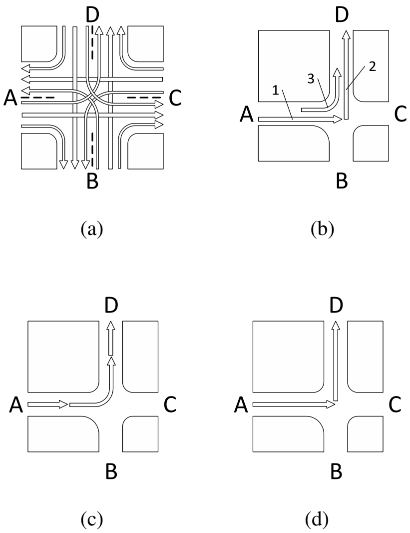


Figure 2: (a) Intersection. (b) Available Videos. (c) Smooth Turn. (d) Sharp Turn.

V. 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 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.
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 2a 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 2b. It is easy to observe that the solution in Figure 2c is smoother than in Figure 2d although it has one more transition.

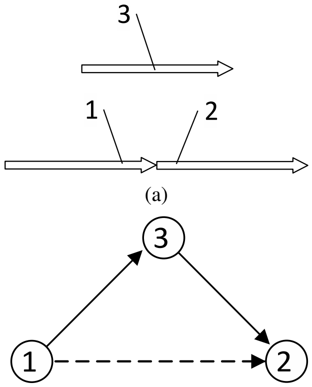
---

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

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

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

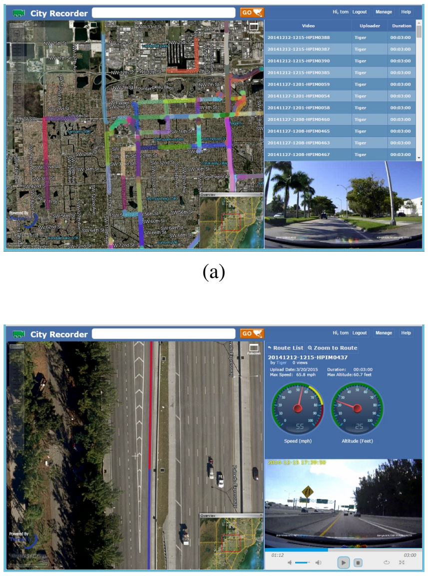
[5] https://www.ffmpeg.org/



(a)

(b)

Figure 3: (a) Interval Modal. (b) Graph Modal.



Video	Uploader	Duration
20141212-1215-HPIM0388	Tiger	00:03:00
20141212-1215-HPIM0387	Tiger	00:03:00
20141212-1215-HPIM0390	Tiger	00:03:00
20141212-1215-HPIM0385	Tiger	00:03:00
20141127-1201-HPIM0059	Tiger	00:03:00
20141127-1201-HPIM0054	Tiger	00:03:00
20141127-1201-HPIM0058	Tiger	00:03:00
20141127-1208-HPIM0460	Tiger	00:03:00
20141127-1208-HPIM0465	Tiger	00:03:00
20141127-1208-HPIM0463	Tiger	00:03:00
20141127-1208-HPIM0467	Tiger	00:03:00

(a)

20141212-1215-HPIM0437
by Tiger 0 views
Upload Date: 3/20/2015
Duration: 00:03:00
Max Speed: 65.8 mph
Max Altitude: 60.7 feet

2014-12-15 17:39:50

(b)

Figure 4: (a) Main Page. (b) Geo-Video Player.

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.

A. *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 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.

B. *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 [12]. 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 2b into an interval model as shown in Figure 3a. If we view intervals as nodes and use edges to indicate that two intervals are overlapped, the problem can also be converted into a graph model as presented in Figure 3b. 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 [13], by calculating the shortest distance using weight of nodes and edges together.

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 [8].

C. *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.

VI. 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.

---

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

20141212-1215-HPIM0465
by tiger 0 views
Upload Date: 3/21/2015 Duration: 00:03:00
Max Speed: 36 mph Max Altitude: 3.0 feet

Similar videos found:

2014-12-15 20:11:38

Figure 5: Video Suggestion

A. General View

The left panel of the main page, as shown in Figure 4a, 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 4b. 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 and a zoomed-out global overview. Velocity and altitude are displayed in virtual instruments along with other information at the top right corner.

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. 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.

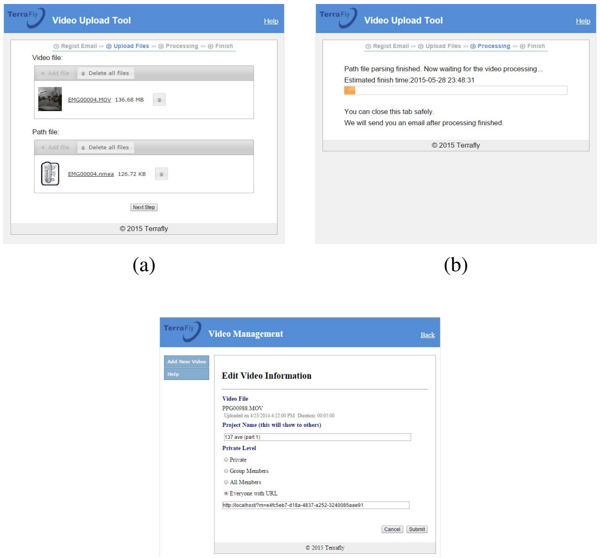
B. Route Preview

When a user searches for a route, the application enters into the route preview mode as shown in Figure 6. It is an online driving simulator, which give users a realistic riding experience. The simulator connects videos generated from Section V together and plays them seamlessly. The whole user interface is designed to mimic the car interior including instruments, onboard video player, and navigation system.

C. Data Management

Users can upload their geo-referenced videos to our system using a guided interface as shown in Figure 7a. Then the data will be processed at the server side and the estimated time is displayed to users as shown in Figure 7b. 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 7c. 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: Driving Simulator



Video file:
EMG00004.MOV 136.68 MB

Path file:
EMG00004.nmea 126.72 KB

(a)

Path file parsing finished. Now waiting for the video processing...
Estimated finish time:2015-05-28 23:48:31

You can close this tab safely.
We will send you an email after processing finished.

(b)

Edit Video Information

Video File
PPG00988.MOV
Uploaded on 4/23/2014 4:22:00 PM Duration: 00:05:00
Project Name (this will show to others)
137 ave (part 1)
Private Level
○ Private
○ Group Members
○ All Members
● Everyone with URL
http://localhost/?m=e4fc5eb7-d18a-4837-a252-3240085aae91

(c)

Figure 7: (a) Data Uploading. (b) Data Processing. (c) Data Management.

VII. 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 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 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 [14] 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 [15]. 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.

VIII. ACKNOWLEDGEMENT

This material is based in part upon work supported by the National Science Foundation under Grant Nos. I/UCRC IIP-1338922, AIR IIP-1237818, SBIR IIP-1330943, III-Large IIS-1213026, MRI CNS-1429345, MRI CNS-0821345, MRI CNS-1126619, CREST HRD-0833093, I/UCRC IIP-0829576, MRI CNS-0959985, RAPID CNS-1507611, and U.S. DOT Grant ARI73. Includes material licensed by TerraFly (http://terrafly.com) and the NSF CAKE Center (http://cake.fiu.edu).

REFERENCES

[1] J. Luo, D. Joshi, J. Yu, and A. Gallagher, "Geotagging in multimedia and computer vision-a survey," *Multimedia Tools and Applications*, vol. 51, no. 1, pp. 187–211, 2011.

[2] A. Medvedev, A. Zaslavsky, V. Grudinin, and S. Khoruzhnikov, "Citywatcher: Annotating and searching video data streams for smart cities applications," in *Internet of Things, Smart Spaces, and Next Generation Networks and Systems*. Springer, 2014, pp. 144–155.

[3] Y. Zheng, L. Wang, R. Zhang, X. Xie, and W.-Y. Ma, "Geolife: Managing and understanding your past life over maps," in *Mobile Data Management, 2008. MDM'08. 9th International Conference on*. IEEE, 2008, pp. 211–212.

[4] Y.-T. Zheng, Z.-J. Zha, and T.-S. Chua, "Research and applications on georeferenced multimedia: a survey," *Multimedia Tools and Applications*, vol. 51, no. 1, pp. 77–98, 2011.

[5] D. Anguelov, C. Dulong, D. Filip, C. Frueh, S. Lafon, R. Lyon, A. Ogale, L. Vincent, and J. Weaver, "Google street view: Capturing the world at street level," *Computer*, no. 6, pp. 32–38, 2010.

[6] B. Chen, B. Neubert, E. Ofek, O. Deussen, and M. F. Cohen, "Integrated videos and maps for driving directions," in *Proceedings of the 22nd annual ACM symposium on User interface software and technology*. ACM, 2009, pp. 223–232.

[7] J. Kopf, B. Chen, R. Szeliski, and M. Cohen, "Street slide: browsing street level imagery," in *ACM Transactions on Graphics (TOG)*, vol. 29, no. 4. ACM, 2010, p. 96.

[8] C. Peng, B.-Y. Chen, and C.-H. Tsai, "Integrated google maps and smooth street view videos for route planning," in *Computer Symposium (ICS), 2010 International*. IEEE, 2010, pp. 319–324.

[9] J. Rodríguez, A. Quesada-Arencibia, D. Horat, and E. Quevedo, "Web georeferenced video player with super-resolution screenshot feature," in *Computer Aided Systems Theory-EUROCAST 2013*. Springer, 2013, pp. 87–92.

[10] C.-Y. Chiang, S.-M. Yuan, S.-B. Yang, G.-H. Luo, and Y.-L. Chen, "Vehicle driving video sharing and search framework based on gps data," in *Genetic and Evolutionary Computing*. Springer, 2014, pp. 389–397.

[11] N. Rishe, S.-C. Chen, N. Prabakar, M. A. Weiss, W. Sun, A. Selivonenko, and D. Davis-Chu, "Terrafly: A high-performance web-based digital library system for spatial data access." in *ICDE Demo Sessions*, 2001, pp. 17–19.

[12] M. J. Atallah, D. Z. Chen, and D. Lee, "An optimal algorithm for shortest paths on weighted interval and circular-arc graphs, with applications," *Algorithmica*, vol. 14, no. 5, pp. 429–441, 1995.

[13] E. W. Dijkstra, "A note on two problems in connexion with graphs," *Numerische mathematik*, vol. 1, no. 1, pp. 269–271, 1959.

[14] A. Frome, G. Cheung, A. Abdulkader, M. Zennaro, B. Wu, A. Bissacco, H. Adam, H. Neven, and L. Vincent, "Large-scale privacy protection in google street view," in *Computer Vision, 2009 IEEE 12th International Conference on*. IEEE, 2009, pp. 2373–2380.

[15] S.-C. Chen, H.-Y. Chen, Y.-L. Chen, H.-M. Tsai, and B.-Y. Chen, "Making in-front-of cars transparent: Sharing first-person-views via dashcam," in *Computer Graphics Forum*, vol. 33, no. 7. Wiley Online Library, 2014, pp. 289–297.