Presentation of Multiple Geo-Referenced Videos

Zhang Lingyan

B. Eng. (Hons), Zhejiang University

A THESIS SUBMITTED

FOR THE DEGREE OF MASTER OF SCIENCE

COMPUTER SCIENCE DEPARTMENT
SCHOOL OF COMPUTING

NATIONAL UNIVERSITY OF SINGAPORE

2011

Abstract

Geo-tagging is becoming increasingly common as location information is associated with various data that is collected from a variety of sources such as Global Position System (GPS), compass, etc. In the field of media, images and most recently videos, can be automatically tagged with the geographic position of the camera. Geo-tagged videos can be searched according to the location information which can make the query more specific and precise if the user already knows the place he or she is interested in. In this thesis we consider the challenge of presenting geo-referenced videos and we first review the related work in this area. A number of researchers have focused on on geo-tagged images while few have considered geo-tagged videos. Earlier literature presents the concept of *Field-of-View* (FOV) which we also adopt in our research. In addition, recently the concept of 3D virtual environments has gained increased prominence, with Google Earth being one example. Some of them are so-called mirror-worlds – large-scale environments that are essentially detailed computer-models of our three-dimensional real world. The focus of our work is on utilizing such virtual environments for the presentation of multiple geo-referenced videos. We are proposing an algorithm to compute a reasonable viewing location or *viewpoint* for an observer of multiple videos. Without calculating the viewpoint, it might be difficult to find the best location to watch several videos. Our proposed system automatically presents multiple geo-referenced videos according to an advantageous viewpoint. We performed several experiments to demonstrate the usefulness and feasibility of our algorithm. We conclude the thesis by describing some of the challenges of our research and possible future work.

Acknowledgments

First, my sincere thanks to the guidance of Dr. Roger Zimmermann, my advisor. He carefully taught me a glimpse of the concept of presentation of geo-referenced videos, from time to time to discuss and enlighten me in the right direction, so I benefited a lot.

The completion of this thesis was made possible with the great help of a research fellow of my supervisor Dr. Beomjoo Seo. His significant advice and patient explanations were very useful during the continuation of my research. In addition, graceful thanks go to a previous student of my supervisor, Dr. Sakire Arslan Ay, for her constructive suggestions and academic discussions.

Finally, I would like to thank my parents and my friends for their support.

Contents

Summary

The primary objective of this thesis is to present multiple geo-referenced videos in a useful way within 2D or 3D mirror worlds. As the number of geo-tagged videos is increasing, showing multiple videos within a virtual environment may become an important topic of media research. We conjecture that presenting videos in the context of maps or virtual environments is a more precise and comprehensive way. Our example geo-referenced videos contain longitude, latitude, directional heading, and video timestamp information which can aid in the search of videos that capture a particular region of interest. Our main work focuses on presenting the videos in 3D environments. Therefore, we show the videos with a 3D perspective that may present the scene at a certain angle. Furthermore, to show multiple videos, we propose an algorithm to compute a suitable common viewpoint to observe these videos. To obtain a better viewpoint we provide some guiding rules. Finally, we perform an experiment with our system to examine its feasibility and effectiveness.

We have studied the literature of existing advanced technologies in detail which we leverage for reference. There exist many models that make use of the field-of-view (FOV) concept based on the location and orientation information which we also use in our research. In a virtual world like Second Life, although it is an imaginary environment, the user can watch videos which are correctly warped according to the 3D perspective. Learning from this example, we have adopted video presentation with a 3D perspective in our system. In later sections we will describe the implementation of our system and the design of a prototype of geo-referenced video search engine for both 2D and 3D environments. In our system, we have achieved the querying of geo-referenced videos, and their presentation with Google

Maps and Google Earth. We will show the adopted architecture, the database design and the 2D and 3D implementations. Furthermore, the evaluation of our system is shown, which involves our algorithm for calculating the viewpoint. Combined with a web interface, we can visually show the results and check the effectiveness of our algorithm. Although there are some tradeoffs in our approach, we believe that it is useful. There are many conditions we need to consider when implementing this algorithm. Firstly, if there are more than two videos, calculating the viewpoint is more difficult because maybe two videos are close together or there are more than four videos. Secondly, if two videos are shot in opposite directions we need to decide which video will be in the viewable scene. Thirdly, if the viewpoint is calculated far from the position of the camera, we may need to decide to move closer to the camera. Finally, we also introduce some challenges of our research, show possible future work, and draw conclusions and contributions of our work.

To summarize, our novel system can present multiple geo-referenced videos with a 3D perspective in a corresponding virtual environment. As a basis for our system, we propose an algorithm to show multiple videos. As demonstrated through experiments, the approach produces useful results.

List of Tables

List of Figures

Chapter 1

Introduction

1.1 Motivation

Due to technological advances an increasing number of videos are being collected with certain sensor information from devices such as GPS, digital compasses, and Motes with wireless radios. Additionally, there exist now 2D and 3D virtual environments mirroring our real world. Therefore it is possible to use these sensor meta-data to present the associated videos according to the corresponding viewpoints in these mirror world environments. The captured geographic meta-data have significant potential to aid in the process of indexing and searching geo-referenced video data, especially in location-aware video applications.

If videos are presented in a useful way, users can directly find what they desire through the videos with associated location and orientation information. Furthermore, videos are presented within the 3D virtual environment which contains real world location information (longitude and latitude data). Based on this environment, our presentation approach can match the real worlds videos with the 3D virtual world and give the users an intuitive feel when obtaining the video results.

Technological advances have led to interesting developments in the following three areas:

- Location and direction information can now be affordably collected through GPS and compass sensors. By combining location data with other information, interesting new applications can be developed. Location data also gives rise to a natural organization of information by superimposing it on maps that can be browsed and queried.

- While maps are two-dimensional, three-dimensional mirror worlds have recently appeared. In these networked virtual environments, the real world is "mirrored" with digital models of buildings, trees, roads, etc. Mirror worlds

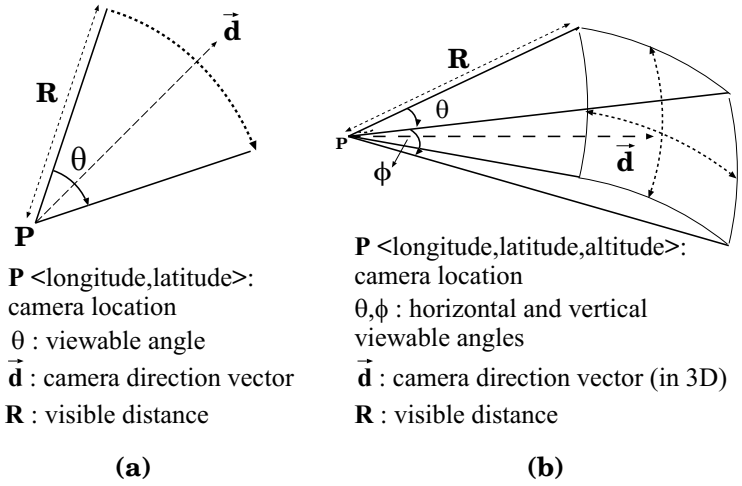


Figure 1.1: Illustration of FOVScene model (a) in 2D and (b) in 3D.

allow a user to explore, for example, a city from the comfort of their home in a very realistic way.

- High quality video camcorders are now quite inexpensive and the amount of user collected video data is growing at an astounding rate. With a large video data set, we can obtain more precise and convincing results.

Our goal with the presented approach is to harness the confluence of the above developments. Specifically, we envision a detailed mirror world that is augmented with (possibly user-collected) videos that are correctly positioned in such a way that they overlay the 2D and 3D structures behind them, hence bringing the mostly static mirror world to life and providing a more dynamic experience to the user who is exploring such a world.

As a basis for our work we leverage a query system called Geo-Referenced Video Search (GRVS) which is a web-based video search engine that allows geo-referenced videos to be searched by specifying geographic regions of interest. To achieve this system, a previous study [3] investigated the representation of a viewable scene of a video frame as a circular sector (i.e., a pie slice shape) using sensor inputs such as the camera location from a GPS device and the camera direction from a digital compass. Figure 1.1 shows the corresponding 2D and 3D field-of-view models. In 2D space, the field-of-view of the camera at time t, $(FOVScene(P, \vec{d}, \theta, R, t))$ forms a pie-slice-shaped area as illustrated in Figure 1.1(a). Figure 1.1(b) shows an example camera *FOVScene* volume in 3D space. For a 3D *FOVScene* representation we would need the altitude of the camera location point and the pitch and roll values to describe the camera heading on the zx and zy planes (i.e., whether camera is di-

rected upwards or downwards). Based on the proposed model, we constructed three video acquisition prototypes (shown in Figures 1.2, 1.3, and 1.4) to capture the relevant meta-data, implemented a database with a real-world video data set captured using our prototype capture systems, and developed a web-based search system to demonstrate the feasibility and applicability of our concept of geo-referenced video search.

Figure 1.2: Early setup for geo-tagged video data collection: laptop computer, OceanServer OS5000-US compass, Canon VIXIA HV30 camera, and Pharos iGPS-500 receiver.

Now we will first discuss the acquisition of geo-referenced videos. Figure 1.2 illustrates the capture application with computer, camera, GPS, and compass separately. When using this early prototype, its operation is very inconvenient. In other words, we need to carry significant equipment to record videos. In contrast, Figures 1.3 and 1.4 show the acquisition applications implemented on mobile phones. We have implemented the software for iPhone and Android. It is obvious that using mobile phones will be more feasible than using the equipment shown in Figure 1.2. Therefore, in our recent work we have been using these phones for video acquisition. With mobile phone applications we can more easily expand our data set and with a larger data set our experimental results will be more convincing.

To test the feasibility of this idea we have collected a number of videos that were augmented with compass and GPS sensor information using the above data acquisition prototype. We then have used Google Maps and Google Earth as a backdrop to overlay the acquired video clips in the correct locations. According to this method, our video results can be presented in an intuitive way, and the users can watch the videos within the mirror world.

Another issue we have considered and implemented in this research is the presentation of multiple videos. As search results contain more and more video clips, our objective is to show multiple videos with a good viewpoint. Then the users can watch several videos at the same time and find the most relevant one more quickly

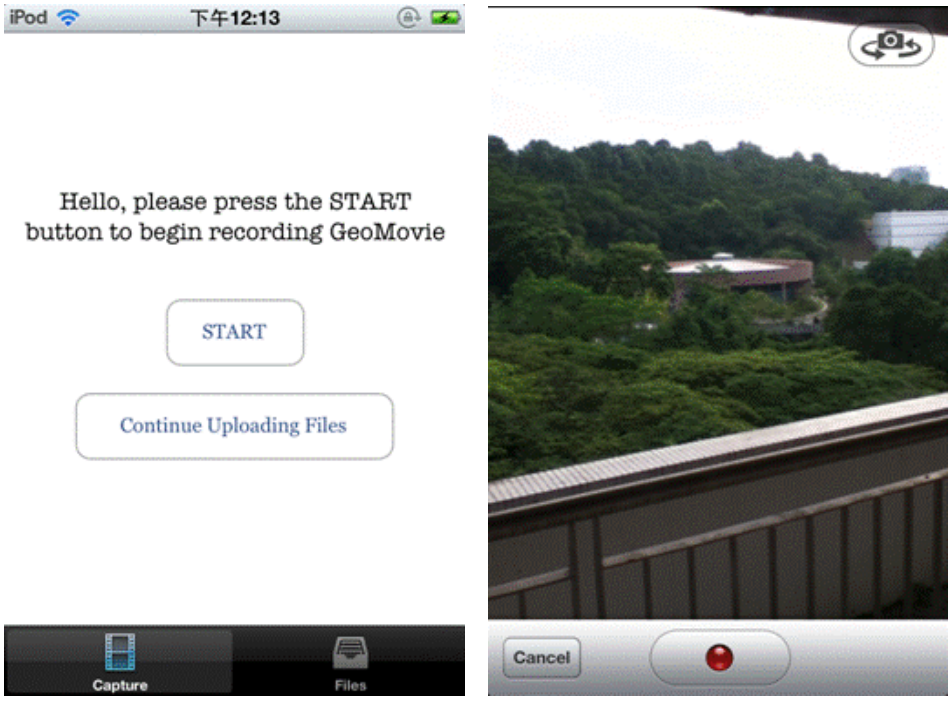


Figure 1.3: Integrated iPhone application for geo-tagged video acquisition.

than otherwise. To achieve this goal, we are proposing an algorithm to compute a common viewpoint. However, there is a tradeoff between obtaining a good viewpoint and providing a smooth trajectory. The trajectory is a path which consists of many viewpoints or camera positions to describe where the view or camera location is. To balance the tradeoff, we have provided a number of rules to solve this problem, and also to improve the efficiency.

1.2 Research Problem Statement

Our research goal is to provide users with an enhanced presentation of multiple geo-referenced videos in a specific region of interest. The term enhanced presentation refers to the display of multiple videos such that each video is rendered on a virtual canvas positioned in a 3D environment to provide an aligned overlay-view with the objects in the background (*e.g.*, buildings). Our conjecture is that such an integrated rendering of videos provides increased situational awareness to users and would be beneficial for a number of applications. Based on this objective we state several research problems that we investigated in our work.

First, we need to determine which environment works best to present the videos. The reason is that using a suitable environment can help users understand the videos more comprehensively. With Google Earth, the 3D virtual models correspond to

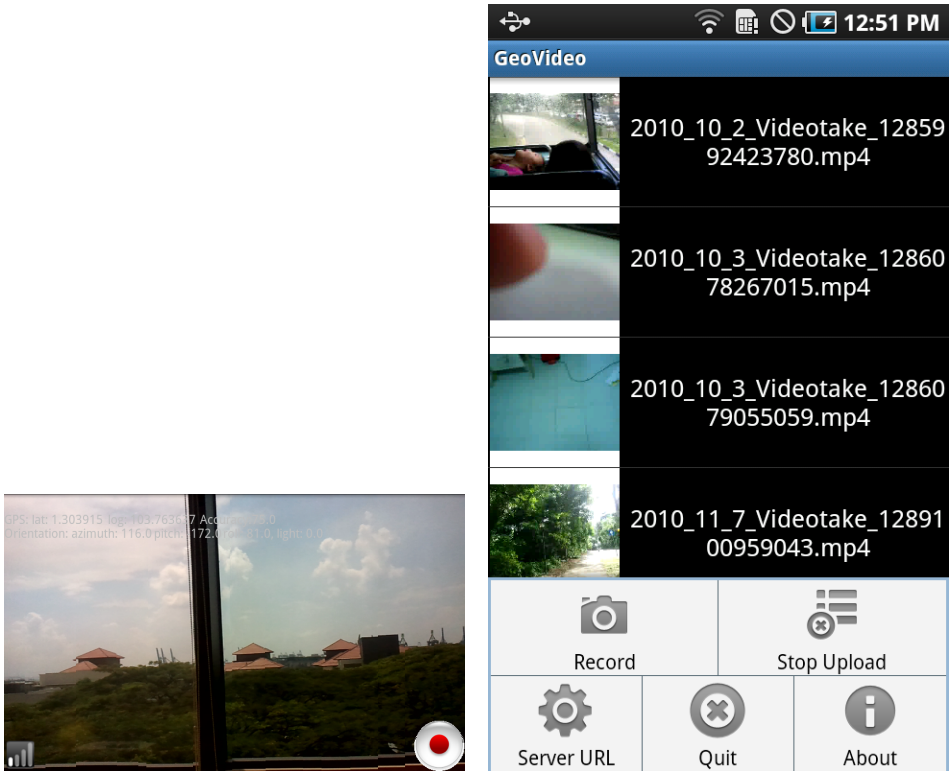


Figure 1.4: Android application for geo-tagged video data acquisition.

objects in the real world (termed a *mirror world*), however, with Second Life the environment is imaginary. Therefore, Google Earth serves as a good choice for our research. Figure 1.5 shows a screen shot of Google Earth, and we can see the virtual 3D buildings in this environment.

Second, it is very important to provide precise visual alignment of the video frames with the 3D virtual world. In such a virtual world, there are many virtual objects corresponding to the real world. By comparing the frames in the videos with such objects we can check the accuracy of our system and our initial data. If a video frame can totally match the objects in the virtual environment, we may say that our system is very precise. However, because of inaccuracies in the initial data from GPS and compass equipment, the matching process is sometimes challenging and the video frames do not match the objects. In such a situation, if the camera location is in the right place (i.e., matching the street, road, etc.), this means that our system is not accurate. We need to check the frames; if these frames are in the range (we will define this later), then we can accept such a result, otherwise, the system may not be good for video presentation.

Third, we need to think about how to reasonably present multiple geo-referenced videos. With an appropriate environment, how to place the videos and how to

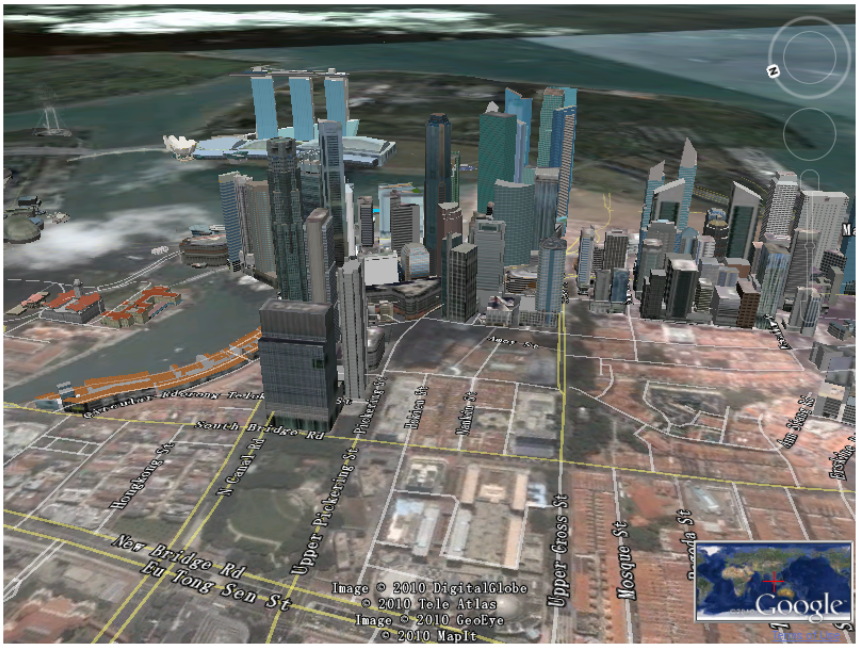
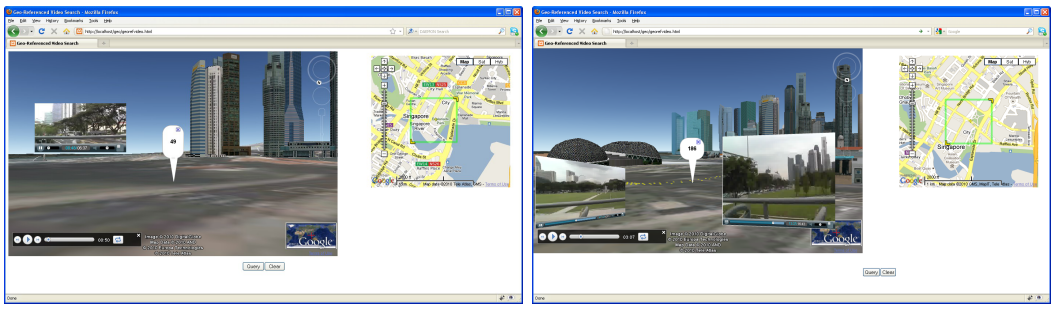


Figure 1.5: Example Google Earth 3D environment of the Marina Bay area in Singapore.

show them are important issues that we need to carefully consider especially for multiple videos. Our conjecture is that showing videos in a 3D environment with a 3D perspective will be better than simply using a flat, non-warped 2D view. We can see the difference between a 2D perspective and a 3D perspective of videos in Figure 1.6. In addition, with the presentation of multiple videos we need to design an algorithm to compute the best viewpoint from which a user can view multiple videos in a suitable way.

Fourth, most of the time the search results contain more than one video. Accordingly, we need to consider how to rank them, and how many videos should be presented at the same time. In addition, as part of these considerations, we also need to consider the network bandwidth. With the presentation of multiple videos in a 3D environment, a possible network bottleneck is a big challenge.

Lastly, given different environments we need to utilize different methods. With 2D environments we can easily present the videos with a flat, non-warped 2D view. Using a video player such as Flowplayer, Adobe Flash Player, etc. we can achieve this. However, with a 3D environment the situation is more complicated. Given videos with a 3D perspective, a normal video player cannot handle these issues. We use the HTML 5 video tag to play videos with a 3D perspective. In addition, the query window should have a 3D shape which means we can query in terms of 3D instead of 2D. More specifically, the 3D FOV model we use is shown in Figure 1.1(b).



(a) (b)

Figure 1.6: The difference between presenting videos in 2D perspective or 3D perspective.

1.3 Thesis Roadmap

The rest of this thesis is organized as follows. Chapter 2 presents a literature survey related to our research. Implementation of our system is described in Chapter 3. In this chapter, we present the detailed technologies we have adopted. Furthermore, in Chapter 4 we describe some experiments and show how our algorithm works. In addition, challenges and future work are outlined in Chapter 5. Finally, Chapter 6 draws conclusions of this thesis.

Chapter 2

Literature Survey

The existing literature on geo-located videos is quite limited. In this chapter we review some early work that has focused on 2D geo-referenced video acquisition, search and presentation. Additionally, we also give a general overview of other relevant research topics. The subsequent parts of this chapter are organized like the following. First, definitions of related concepts are described in Section 2.1 to help explain the content. Second, since our video search engine is based on acquired sensor information (location coordinates, compass data, etc.), in Section 2.2 we review some selected papers of image geo-spatial techniques which utilize different types of sensor information. Third, an effective approach of indexing and retrieving geo-referenced video is necessary for our system. Hence, a brief survey of video retrieval techniques is given in Section 2.3. Fourth, we describe the Field-of-View (FOV) model in Section 2.4. For each video using the FOV model can provide a more accurate position when it is shown on a map. Therefore, some work which exploits the direction (orientation) information in their FOV models are examined in this section. Fifth, how to present video in a 3D environment is another vital problem in our system. In Section 2.5, several approaches which target 3D presentation methods are reviewed. We summarize how these previous techniques have inspired our new algorithm for computing a viewpoint for multiple videos. Finally, we draw conclusions for the literature review in Section 2.6.

2.1 Definition of Related Concepts

To be able to better describe the forthcoming concepts, we first list several definitions of specialized terms.

Document Space: We are only concerned with geographic information of document space which can be broken into two subspaces: *a geographical space* and *a thematic space.*

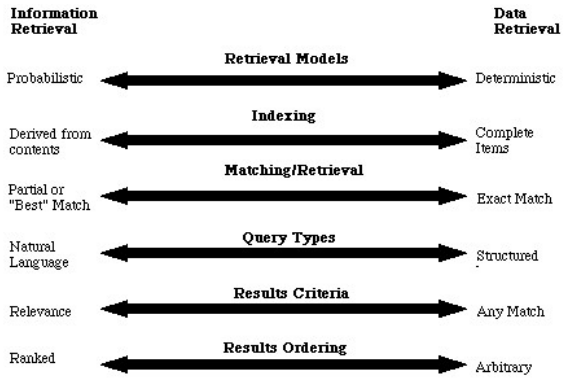


Figure 2.1: Information Retrieval versus Data Retrieval spectrum.

- *Geographical space*: a two-dimensional space representing a geographic coordinate system. Documents can be geometrically represented and applied as footprints in such a space.

- *Thematic space*: a multi-dimensional space where documents are concerned with their theme.

RDF: The Resource Description Framework is a framework for representing Web resources which can be used in a variety of areas. For instance, providing better search engines, describing the content of special web pages or digital libraries, and so on. RDF can denote metadata for inter-communication between applications that exchange information which machines can understand via the web. In addition, RDF metadata is represented by a syntax for encoding and transportation. One choice of syntax is the Extensible Markup Language (XML). Combining RDF and XML can make metadata more understandable. The objective of RDF is to define a mechanism of describing data information without assumptions for a particular domain [35].

GIR: Geographic Information Retrieval can be treated as special case of traditional information retrieval. GIR provides access to geo-referenced information sources which includes all of the core areas of Information Retrieval (IR). In addition, it lays emphasis on spatial and geographic indexing and retrieval.

The concepts of "Information Retrieval" and "Data Retrieval (DR)" related to database management systems (DBMS) are different. A variety of attributes of IR and DR are shown in Figure 2.1. Firstly, in IR, the model of providing access to documents is probabilistic as it is concerned with subjective issues. On the other hand, DR is deterministic with retrieval processes that are certain. In GIR, applications generally adopt both deterministic and probabilistic retrieval. Secondly,

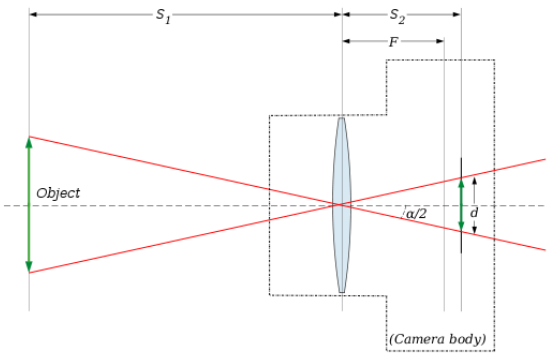


Figure 2.2: Pictorial diagram of angle of view.

indexing for IR is derived from contents while with DR its entirety is the indexing unit. Still the hybrid method is applied for GIR. Thirdly, the matching and retrieval algorithms are based on the retrieval model. In other words, the retrieval algorithms of IR are probabilistic which may include the actual calculation of probabilities. In contrast, the DR algorithms are deterministic which require an exact match of query specification and the contents of a database. Fourthly, the query types of IR and DR are distinct, meaning that IR searches are expressed in natural language that may be ambiguous, while DR queries are expressed in a structured query language which is more precise. Finally, the results for IR are shown in a ranked order while DR query results are arbitrary. As a consequence, Geographic Information Retrieval (GIR) is a combination with a DBMS concerning indexing, retrieving, and searching of geo-referenced information sources [34].

GIS: Geographical Information Systems introduce particular utilities for obtaining, storing, controlling, and showing geo-referenced location data. In a generic sense, GIS are systems that allow users to create queries to match associated geographical information. The most common method of data creation for modern GIS is digitization, where a map is transferred into digital imagery through a computer-aided design (CAD) program [67].

PIRIA: The Program for the Indexing and Research of Images by Affinity is a content-based search engine. Piria is a novel search engine that uses the query-by-example method. When a query is sent to the system, then we can obtain a list of ranked images. The ranking method is not only based on keywords, but also form, color and texture [21]. This technique is described in one of the manuscripts we have reviewed, therefore we introduce this terminology as an illustrative example.

FOV: The field of view (abbreviated FOV) is the (angular or linear or areal) range of the observable world [67]. Different animals have different fields of view which depend on the location of the eyes. Compared with humans who have almost

180-degree forward-facing vision, some birds have a nearly 360-degree field of view. The concept is related with the angle of view, and Figure 2.2 shows the detailed information. A rectilinear lens is in the camera, and S_1 is the distance between the lens and the object. Considering the situation in two dimensions, α is the angle of view, and F is the local length which is attained by setting the lens for infinite focus. According to this figure, we can easily obtain that the "opposite" side of the right triangle is $d/2$, and the "adjacent" side is S_2 (the distance from the lens to the image plane). Therefore, we can obtain Equation 2.1 from basic trigonometry which we can solve for α, and Equation 2.2 is generated. Then the angle of view is given by Equation 2.3, in this equation $f = F$.

$$\tan(\frac{\alpha}{2}) = \frac{d/2}{S_2} \tag{2.1}$$

$$\alpha = 2 \arctan \frac{d}{2S_2} \tag{2.2}$$

$$\alpha = 2 \arctan \frac{d}{2f} \tag{2.3}$$

LIDAR: Light Detection And Ranging is an optical remote sensing system which can collect geometry samples according to measured properties of scattered light to find the range and other characteristics of a distant object. A general method (radar) is to use radio waves to determine distance, but LIDAR adopts laser pulses to compute the distance. The range is computed through the time delay between transmitting a pulse and the detection of the return signal [67].

MBB: The minimum bounding box for a point set in N dimensions is the box which measures the smallest side lengths within which all the points lie. The term "box" stems from its use in the Cartesian coordinate system, and in the 2D case it is also called the minimum bounding rectangle.

IBR: Image Based Rendering depends on a set of two-dimensional (2D) images of a scene to produce a three-dimensional (3D) model to render novel views of this scene with the help of computer graphics and computer vision methods. Typically, IBR is an automatic method to map multiple 2D images to novel 2D images.

FVV: Free View Video allows users to control the viewpoint and generate new views of a dynamic scene from any 3D position.

VBR: Video Based Rendering is an extension of image-based rendering that can handle dynamic scenes [47]. Furthermore, according to Shields [6], VBR can refer to the generation of individual frames by computer. This can be used to produce a fluid video and especially for affecting certain types of applications. For instance, if employing a special filter in a video using a software program, then the video will

be rendered through the computer and each frame will be produced and assembled into a video output.

MVC: Multi-view Video Coding is an amendment to the H.264/MPEG-4 AVC video compression standard to enable the efficient encoding of video sequences from multiple cameras based on a single video stream. In addition, multi-view video contains a large number of inter-view statistical dependencies, therefore the integration of temporal and inter-view prediction is key for MVC [67].

2.2 Geo-Spatial Techniques for Images

There exist several research areas that are concerned with geo-spatial images related with location, time and orientation information. Some research focuses on sharing and browsing of geo-referenced photos, some emphasize hierarchies and clustering of images, and others concentrate on how to present images to users. In the following sections we will perform a detailed literature review.

2.2.1 Image Browsing

We will review several papers related with how to browse images according to location and other relevant information. As examples, many tourists are recording photos of family while traveling, archaeologists take photos of historical relics, and botanists shoot images of plant species. In these situations, the geographic location information is a critical marker when browsing these images. In addition, there are many ways to present location information, such as place names ("San Francisco"), street addresses, zip codes, latitude/longitude coordinates, and so forth. Most of the GIS projects use latitude and longitude coordinates, for example as defined by the WGS84 standard. This is a very concise and accurate way to designate point locations and also a format that can be recognized by certain systems [66]. We will now describe some related projects.

First, Google groups allows the embedding of photos in Google Maps. These photos and videos are both based on their geo-locations, which have to be uploaded manually. GPicSync [53] is a Google project that aims to automatically insert locations into users' photos. Thus, such photos can also be used with any 'geocode aware' application like Google Earth, Flickr, etc. On the other hand, Microsoft groups have introduced the World Wide Media eXchange (WWMX) to browse images on their web site. Toyama *et al.* [66] from Microsoft have presented a system that uses geographic location tags based on WWMX. The WWMX database contains metadata of timestamps and location information, which makes it relatively easy to browse the photos. In addition, acquiring location tags on photos, establishing data structures

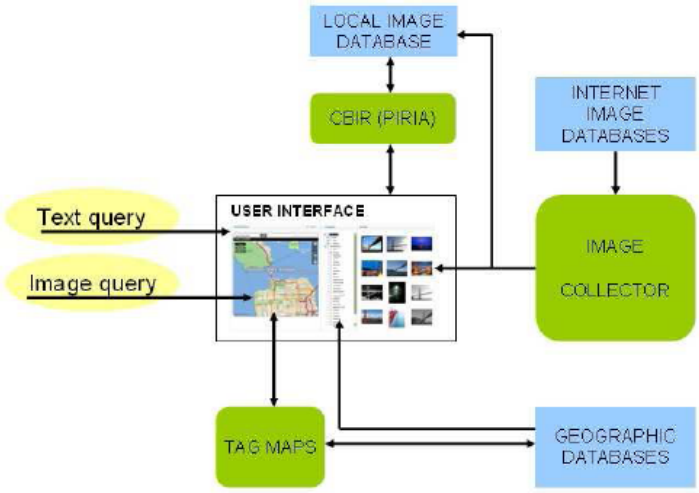


Figure 2.3: Architecture of ThemExplorer.

for images, and implementing UIs for location-tagged photo browsing are the other main contributions of this paper.

Second, another research direction is based on geographical information retrieval. GeoVIBE is a browsing tool which builds on geographical information retrieval (GIR) and textual information (IR) systems [7]. In addition, this system includes two types of browsing strategies: GeoView and VibeView. GeoView enforces a geographical order on the document space with the idea of hypermaps. On the other hand, VibeView presents a similar document space with multiple reference points. The GeoVIBE integrates the two, and users can search information with either geographic clues or conceptual clues. Similarly, Popescu *et al.* [1] presented a suitable and more powerful map-based tool named ThemExplorer which combines a geographical database and a content-based facility. Nowadays, there exist a number of map-based interfaces such as Google Maps, Google Earth, Yahoo Maps, and so on. The authors also evaluated the accuracy of ThemExplorer for browsing geo-referenced images through different dimensions.

Figure 2.3 shows the architecture of ThemExplorer which includes TagMaps, Content-Based Image Retrieval (CBIR), and an image collector. With ThemExplorer, users can ask for geographic names within a certain region, then the system can retrieve the images by querying the database. To search images with CBIR, the system employs PIRIA which is a content-based search engine. This system provides a layering of images according to the geonames in the database. However, there are no usability studies to support the validation of the system.

With a similar idea as ThemExplorer, TagNSearch proposed by Nguyen *et al.* [44]

is also a map-based tool for searching and browsing photographs using related geo-referenced tags. This interface used Flickr [19] as the dataset, and for a given query the imaged will be classified by locations. Each cluster contains a set of geographically nearby images and is associated with a tag cloud that shows an overview of the images' tags in that cluster. According to the combination of clustering and the tag cloud, the search results will be better filtered compared with Flickr alone.

2.2.2 Image Hierarchies and Clustering

In daily life, there exist many ways to manage our images. Martins and Calado [40] presented an approach to perform ranking via Geographic Information Retrieval (GIR) and another paper by Rodden *et al.* [52] shows the result of research on how people manage their personal collections of digital photographs. The findings in this survey paper are obtained from a digital photograph management tool named Shoebox. This tool has many simple functions such as browsing folders, thumbnails and timelines, and also has advanced attributes including content-based image retrieval and speed recognition. According to this paper, it was easier for users to find the digital photos than non-digital ones. However, this strength was dominated by simple features. In other words, the advanced features were not used frequently, and this implies that the tool could probably be improved.

Below we survey work that is concerned with technical image management such as how to automatically organize photos with certain attributes, how to hierarchically organize images, and how to rank photos with certain attributes.

Naaman *et al.* [41] described a system called PhotoCompas to organize digital photos with geographic coordinates. This system generates location and event hierarchies which are created by algorithms that cross time and location to produce an organization. Furthermore, the algorithms can simulate a way of collecting photos and yield a hierarchy with certain comments. The authors were concerned about automatically grouping the photos into events and locations and naming the groups with certain geographical names. The PhotoCompas system is shown in Figure 2.4 which can generate a meaningful organization for photo collections. Furthermore, the system worked well for detecting events, producing location hierarchies and naming. One general problem was described as happening in the naming algorithm. Sometimes, when naming clusters, the suggested cluster name would be difficult to find with respect to the data. For example, with "Northern California" or "East Coast" it seemed feasible to produce results for the former, however, it was much harder for the latter.

Closely related Naaman *et al.*'s research work, Epshtein *et al.* [10] have also shown a framework to organize photos with a scene semantics method. Epshtein *et al.*. use

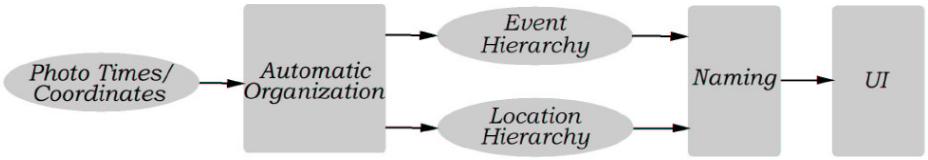


Figure 2.4: PhotoCompas system diagram.

locations and typical view information to display and manage a variety of images. In addition, the authors present the concept of Geo-Relevance and how to compute it with a voting method. Specifically, each point which is visible from the viewpoint of each camera will be voted on. A system is then proposed that obtains typical images from a large number of a geo-positioned set and organizes them hierarchically. For instance, when the system produces a hierarchical organization of a large majority of images of a cathedral it might start from a root node that represents the entire cathedral, and then its children might be a tower, or a gate, and so on. At each level, a child nodes presents more detailed images of the collection. Another approach to organize personal images is proposed by Pigeau *et al.* [49] who apply a statistical method with a related optimization technique based on geographical and temporal image metadata to build and track a hierarchical structure.

Kennedy *et al.* [23] proposed a tool based on context and content to produce diverse and representative image search results with location features and landmarks. In their search task, the authors use location information and other metadata to show a method to extract tags and landmarks. Figure 2.5 shows the system architecture which generates typical views of landmark images. From this figure we identify that at the beginning there exist a majority of tagged images which are then clustering with common views of a landmark. After that, with rank clusters, the system will rank the clusters in terms of their representativeness. Therefore, the highest-ranked images from the highest-ranked clusters are selected and low-ranked ones are discarded. Another related work is by Heuer *et al.* [18] who collected data from Web 2.0 portals and implemented a search engine based on geo-tags. Additionally, the authors explain how to compute the spatial relevance which is the probability that the keyword of a tag is part of the assigned position. Most of the location information are manually inserted, while our system can automatically produce the geo-referenced meta-data and query the related videos according to the location, heading, and video timestamp information.

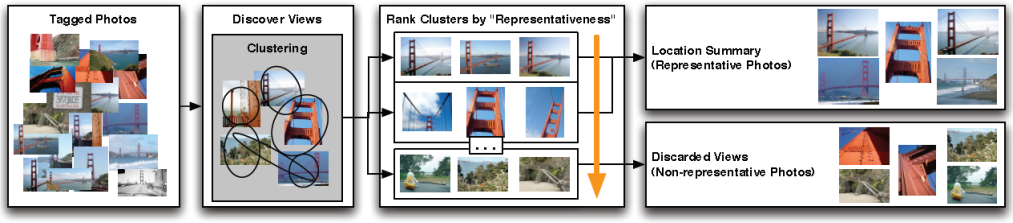


Figure 2.5: System architecture for generating representative summaries of landmark image sets.

Figure 2.6: Estimated camera locations for the Great Wall data set.

2.2.3 Image Presentation

Our research focuses on the presentation of geo-referenced videos. Therefore, the question of how to present the videos is a very important part that we need to consider. In this section we will review two papers which are very related to our work. However, these papers lay emphasis on image presentation instead of videos. No matter whether the presentation is concerned with images, we can draw inspiration from these ideas.

Snavely *et al.* [62] have presented a photo tourism application in three dimensions as shown in Figure 2.6. This method produces estimated camera locations for the Great Wall data set, and the authors introduce a system which automatically computes the viewpoint of each image. We learned the presentation method

of images from this paper, and applied it to our geo-referenced video search with Google Earth. Our video presentation will be adopted for a 3D environment, and also feature a 3D perspective. Under this consideration, we use our own data set and compute the viewpoint to show the videos. In addition, using Google Earth to match the videos with the corresponding 3D virtual buildings in such environment will be more exciting than just presenting images.

Figure 2.7 presents the screen shot of the explorer interface. On the left is an information and search pane. In addition, a thumbnail pane along the bottom controls the sorting of the current thumbnails by date and time and viewing them as a slideshow. Furthermore, there is a map pane in the upper-right corner which displays an overhead view which tracks the user's position and direction. The authors have proposed a reconstruction algorithm, but it has some limitations. When the number of registered cameras grows, the effectiveness of the system will decrease. In addition, without ground control points the algorithm is not guaranteed to generate a metric scene reconstruction. In addition, it is difficult to obtain accurate models.

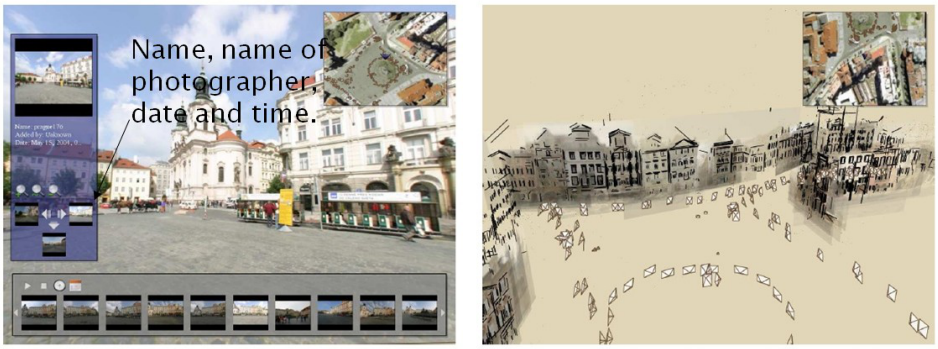


Figure 2.7: Screenshots of the explorer interface. Right: a view looking down on the Prague dataset, rendered in a non-photorealistic style. Left: when the user visits a photo, that photo appears at full-resolution, and information about it appears in a pane on the left.

Another paper related to image presentation was written by Kadobayashi [22] which introduced a new search method based on three dimensional (3D) viewpoints. This paper concentrated on image queries, especially useful for different images with the same object such as in archaeological photographs. In addition, this method can extract the images which contain the same object from different viewpoints.

2.2.4 Summary

In this section we have reviewed related work to geo-tagged image techniques. An overview of the methods in this section is shown in Table 2.1, and it lists the features

of all the techniques.

Technique	*Features*
GPicSync [53]	1. Automatically associates audio or video files in Google Earth and Google Maps. 2. Creates a Google Earth KML file to directly visualize the geocoded photos and track them in Google Earth.
WWMX [66]	1. An end-to-end system that capitalizes on geographic location tags for digital photographs. 2. Location is tied to the semantics of imagery. 3. Browsing by location, whether via maps or by textual place names, is well-understood and intuitive to users.
GeoVIBE [7]	1. Proposes the concept of document space which is divided into a geographical space and a thematic space. 2. Provides a new visual interface to spatial digital libraries.
ThemExplorer [1]	1. Well-structured geographical database to search and retrieve images. 2. Enables searching of images not only with local database, but also on the internet.
TagNSearch [44]	1. The images can be ranked and clustered according to their geo-referenced tag information. 2. With a combination of clustering and the tag cloud, the search results are better sifted.

PhotoCompas [41]	1. Produces hierarchies based on interleaving time and location information. 2. Performes two tasks: one for grouping the photos into distinct events and geographical locations, another for suggesting intuitive geographical names for the resulting groups.
$Geo - Relevance - Hierarchy$ [10]	1. Proposes the notion of Geo-Relevance based on voting approach. 2. Generates a hierarchical organization of the images according to orientation and position information.
$Image - Search - Landmarks$ [23]	1. Combines image analysis, tag data and image metadata to extract meaningful patterns from these loosely-labeled, community-contributed datasets. 2. Generates a summary of the frequently-photographed views by selecting typical views of the landmarks and rejecting outliers.
$Photo - Tourism$ [62]	1. Presentes a novel end-to-end system for taking an unordered set of photos, registering them, and presenting them on a 3D browser. 2. Proposes a reconstruction algorithm and exploration interface on several large sets of photos.

Table 2.1: Summary of features of different techniques for images.

The information for images that are most relevant are geo-tagged properties (longitude, latitude, heading) and also time, the author and other information. We have reviewed several papers, some related with browsing images based on geo-tagged information, some related to creating a hierarchy, clustering and ranking the images to enhance the efficiency of searching images, and some related with the presentation of images in some specific environments. According to the techniques from the literature review, we can leverage some ideas from them. For example, a hierarchy

of images can make our search engine more efficient. In addition, how to present the videos is a key issue in our research. Based on the papers we have reviewed we have learned that it may be a good idea to present videos with a 3D FOV model and in 3D environments. However, the papers in this section have only considered images. Therefore, we extend these techniques to videos in our research.

2.3 Indexing and Retrieving

A geo-referenced information processing system (GIPSY) was proposed by Woodruff *et al.* [68]. In this system, the geo-referenced index terms in plain text are converted to related document indexing and retrieval. Additionally, the words and phrases with geographic data are extracted from text and stored in a database. Therefore, a good algorithm to implement the indexing and retrieval is needed.

In the multimedia domain, the amount of media content is growing which results in an increasing requirement to effectively manage metadata [20]. Theodoridis *et al.* [64] proposed efficient indexing schemes based on spatial and temporal relations. Furthermore, the authors also presented evaluation models associated with these schemes. The indexing methods are bases upon R-tree indexes for spatial data applications, such as GIS, CAD and VLSI design, etc [48]. The authors also applied R-trees to spatio-temporal indexing, and provided hints to the designers so as to how to select the most appropriate scheme based on the authors' requirements. The R-tree is one of the most popular indexing methods for spatial data such as rectangles. There exist a number of papers that investigated this related area, for example, Guttman [17] proposed a dynamic index structure for spatial search which can aid in the design of geo-data applications, and Beckmann *et al.* [5] studied the R*-tree which is an efficient method for indexing points and rectangles.

Despite the importance of the R-tree indexing method in spatial searches, there exist still other directions related to geo-referenced indexing and retrieving methods. Navarrete *et al.* [42] defined an approach for indexing and retrieving geo-referenced video sequences based on their geographic content. The innovation in this paper is that the authors not only utilize the data captured from GPS, but also use data from the compass which produced information to compose a geo-referenced video sequence. The method proposed in this paper is using the thematic geo-referenced information extracted from GIS or spatial databases to segment and index the video sequence. Afterwards the authors focused on retrieving data, namely enabling clients to retrieve elements that satisfy a thematic criteria. For instance, the query request "forests" would return all the fragments of videos containing forests or some related themes. To achieve this, the authors designed a system called VideoGIS [43].

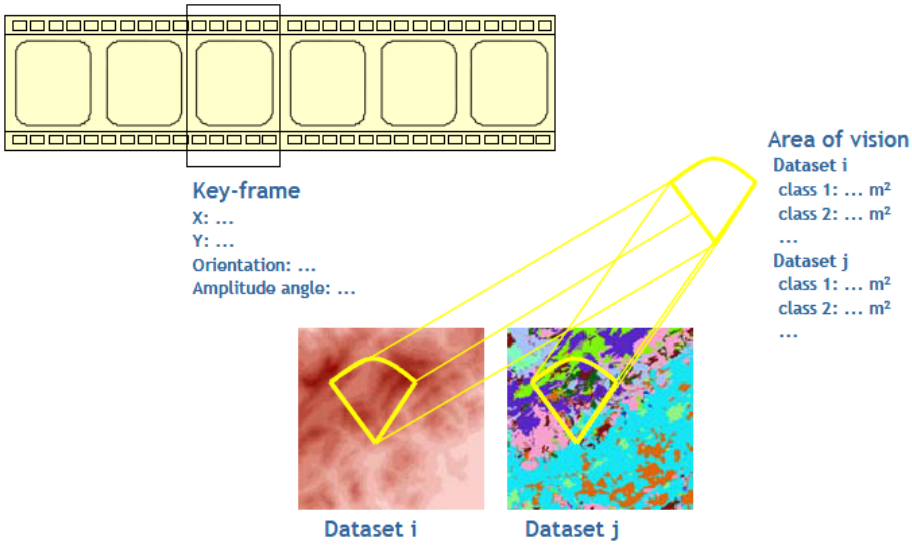


Figure 2.8: Overview of the process of indexing a video segment.

The VideoGIS project combines video and geographic information for the sake of producing hypervideos based on geographic content with navigable context. The video model shown in Figure 2.8 is built on a layering approach which means each indexing theme is associated with a layer or stratum. The video is first segmented based on thematic information and each segment is indexed with corresponding thematic classes. In addition, the contribution of this paper is that the video segments composed of a layer of metadata are stored in an XML file. The metadata includes the indexing theme and the camera properties of a typical frame which enables clients to seek out spatial relations associated with the camera location.

2.4 Field-of-View Models

Geo-referenced medias contain geographic information such as GPS location data, compass heading information, and so forth. To search these kinds of medias we need to build a model to achieve accurate query processing. In this section, we introduce several models. One is related to images and several others are related with geo-referenced videos.

First, Torniai *et al.* [65] have proposed different types of methods to share these meta-data through an RDF description of pictures, location and compass heading information. They also implemented a web-based interface to allow users to check pictures with spatial relationships. Figure 2.9 shows the prototype system for offering methods to share meta-data related to photo collections.

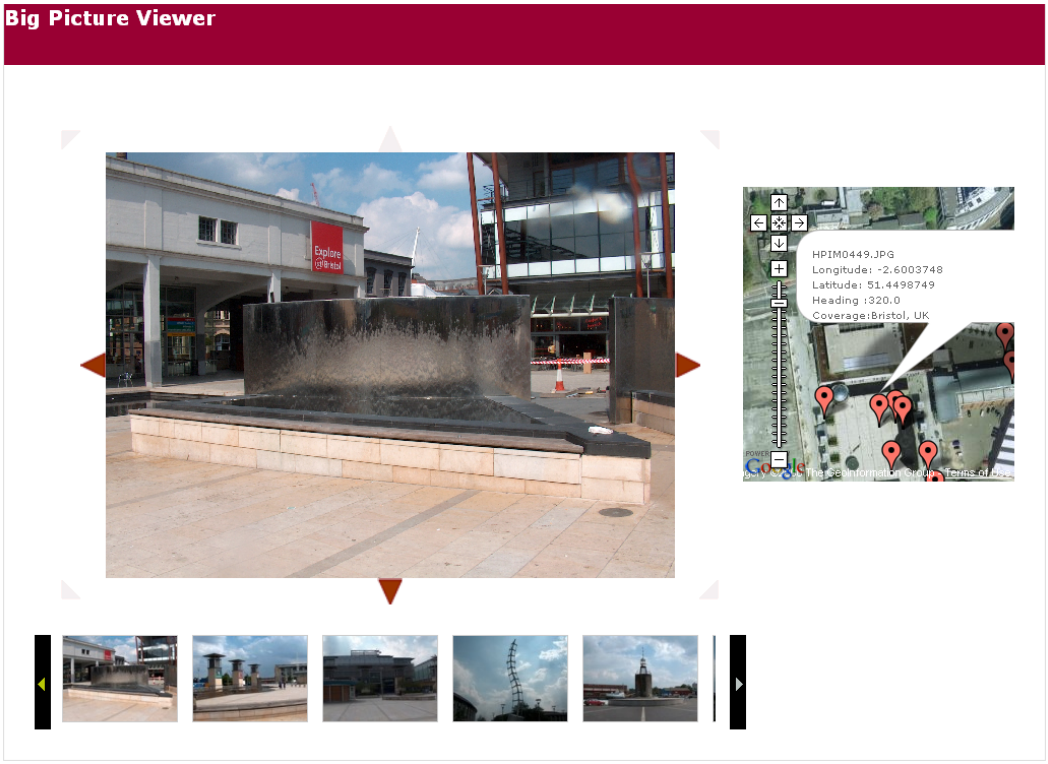


Figure 2.9: Picture browser interface.

In Torniai *et al.*'s work the most important meta-data used is latitude, longitude, and heading information, which is very similar to our research direction. The main difference is between the different media types videos and photos. Furthermore, Torniai *et al.* presented three algorithms for FOV evaluation, spatial relation discovering and picture discovering. Algorithm 1 is most relevant with our research, and to understand this algorithm please consider Figure 2.10. If the direction H_A of $image_a$ is within a range of bearing angle B_A with two points A and B, and H_B of $image_b$ is close to the heading of $image_a$ (H_A), then $image_b$ is in the FOV of $image_a$. In addition, Simon *et al.* [58] also used an FOV cue to leverage this observation with certain objects.

Second, Lewis *et al.* [36] proposed a conceptual model which is suitable for spatial video data sets. The model, called Viewpoint, is a general viewpoint definition as usual, and it uses GIS point and polygon data types. According to Figure 2.11, it is exciting to see such a 3D FOV model which can be used in our future work. On the other hand, Simon *et al.* [59] presented a 2.5 environment model based on visibility and an FOV mofel with a suitable XML-based prototype implementation. Using an XML-based description is notable and desirable, as it can be applied to many applications. Inspired by this idea, we use XML and KML to implement our search

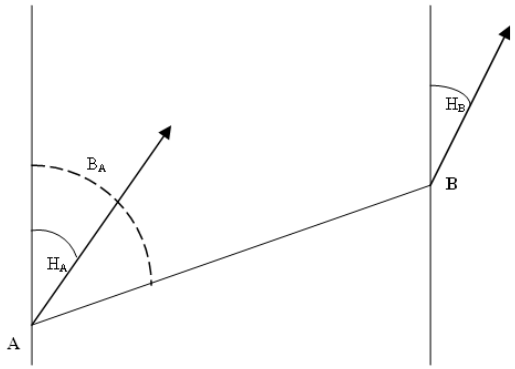


Figure 2.10: Field of view evaluation. If $|H_A$ - $B_A|$ is less than a given threshold then point B is in the field of view of point A. If $|H_A$ - $H_B|$ is less than a given threshold then the pictures taken at A and B have similar heading directions. If both of these conditions are met then $image_b$, taken at point B is in field of view of $image_a$ taken at A.

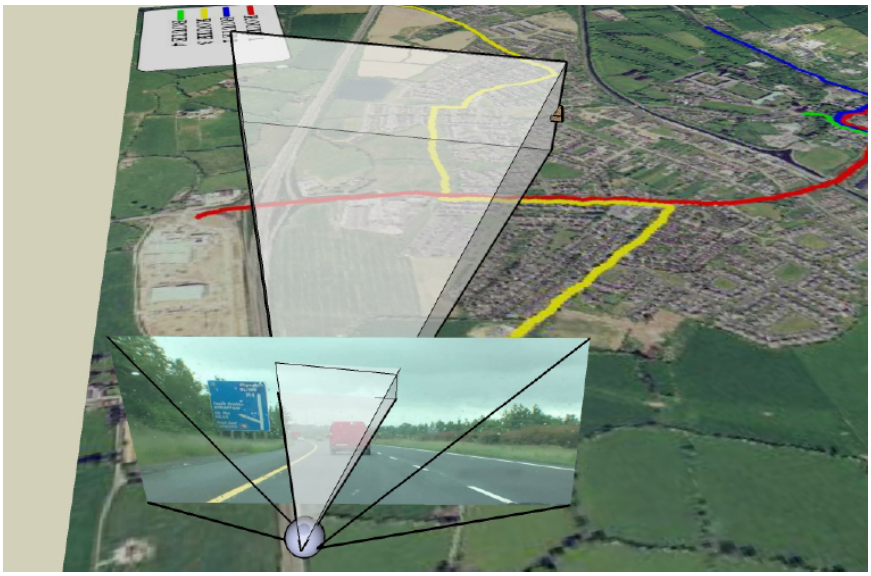


Figure 2.11: Visualization of a Viewpoint in 3D space and how it conceptually relates to a video sequence frame and GPS point. While the image defines a viewing plane that is orthogonal to the Ortho Photo, in spatial terms the polyhedron or more specifically frustum defines the spatial extent. Scales are not preserved.

engine. Besides, our search engine is a web-based system which can be applied to many platforms.

The third FOV model was proposed by Arslan Ay *et al.* [3] and it leverages the GPS and compass information. As can be seen in Figure 2.13, the left figure represents the video search results of a circle scene query. The red rectangle is the

Input: each image pair in the collection $image_a$, $image_b$
Output: distance d(A,B)
`// distance between A and B`

if $d(A,B) < FOV - THRESHOLD$ **then**
  evaluate B_A `// bearing angle between A and B`
  if $|H_A$ - $B_A| < T_{bear}$ AND $|H_A$ - $H_B| < T_{head}$ **then** `// ie B is in field of view of A and ie` $image_a$ `and` $image_b$ `has close camera directions`
    set $fov - relation(image_a, image_b)$
  end
end
else
  do nothing
end

Algorithm 1: Field of view evaluation algorithm.

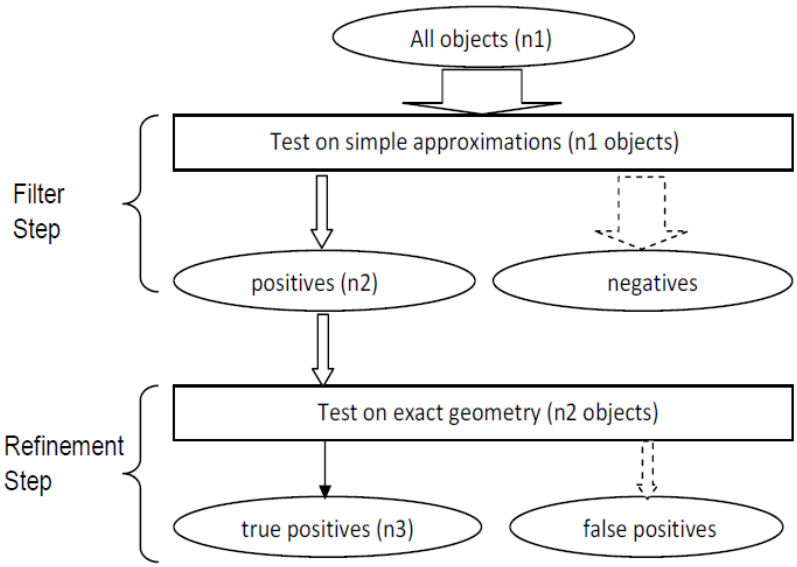


Figure 2.12: Illustration of filter-refinement steps.

query window, the black line is the trajectory for the whole video, the blue circles are based on this trajectory, the red line is the trajectory of results, and the green circles are based on the red trajectory which indicates the video results. On the right side is the corresponding FOV scene query. As shown in (b), the main difference is that the video results are more precise than with the circle scene method. We have implemented our search engine based on this FOV model.

The last model we review was proposed by Kim *et al.* [26] and it proposed a novel

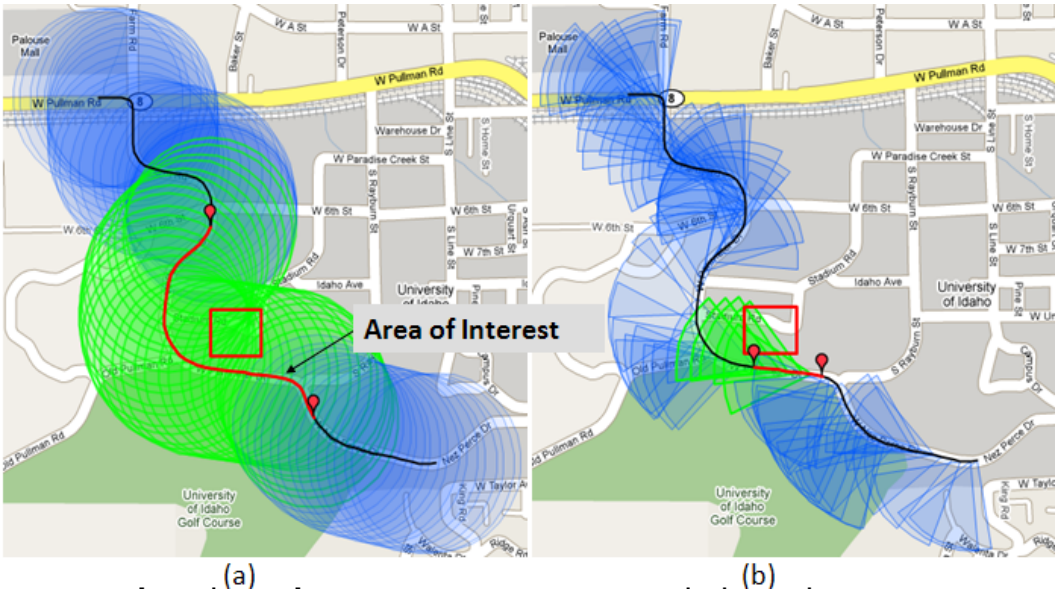


Figure 2.13: The video results of a circle scene query (a) and a FOV scene query.

vector model that is based on the metadata from camera-attached sensors. The field of views (FOV) is comprised of the sensor data which can cover an area of a spatial object and the videos are indexed and searched according to this FOV. The traditional minimum bounding rectangle (MBR) model filters out irrelevant videos from a large number of videos in the first step, and a second step is used to refine the videos with a precise but time-consuming matching function as can be seen in Figure 2.12. Based on this architecture, the authors proposed a novel model to suit the geo-referenced video search which improves the performance of the filter step. In addition, Arslan Ay *et al.* [4] proposed a method to rank the relevant video results because the results of a geo-referenced video query may satisfy the algorithm but may not be visible. The authors presented three ranking algorithms and further showed a histogram-based approach which allows fast computation.

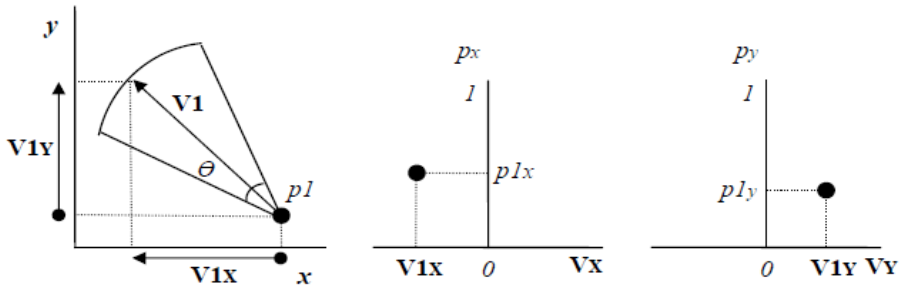


Figure 2.14: FOV representation in different spaces.

In detail, Kim *et al.* introduced a vector model to represent an FOV based on the camera position p and the center vector $\mathbf{V}$. Figure 2.14 shows how the FOV of a video frame forms a circular sector shape in 2D geo-space and the corresponding vector model. The FOV is composed of $< T, p, \theta, \mathbf{V} >$ where T is the real time when the video was captured, p is the camera position, θ is the view angle, and $\boldsymbol{V}$ is the center vector. The authors used space transformation to divide the FOV into two 2D subspaces as shown in Figure 2.14. In other words, $< p_x, p_y, V_X, V_Y >$ is converted to $p_x - V_X$ and $p_y - V_Y$. Based on several experiments the authors concluded that this vector model is more efficient than the MBR model for geospatial video queries.

All the above mentioned FOV models are very similar in the shape that they consider. However, some models focus on finding the related images or videos, while others consider how to make the query more accurate and precise compared with traditional methods, and still others are more concerned with the efficiency of searching. Some models can be adopted in our research such as the FOV model proposed by Arslan Ay *et al.* [3] and Kim *et al.* [26].

2.5 Geo-Location Techniques for Videos

There exist only a few systems combining geo-location information associated with videos. In the following sections we will present the related techniques for videos with geo-referenced information. First, sensor based videos are a premise of our research. The presentation of geo-referenced videos is the most important component and furthermore, for presentation of multiple videos, we need to compute a suitable viewpoint. In addition, when displaying multiple videos we need to consider the network bottleneck. Finally, the 3D environment is also important.

2.5.1 Sensor-Based Videos

In the field of digital sensors embedded in mobile phones and other equipments, there exist some research areas based on sensor information such as establishing traffic light system [2], and a camera sensor network which includes heterogeneous elements compared with homogeneous sensor networks [31]. The main contribution of [31] is the SensEye which is a multi-tier camera sensor network.

Another sensor-based research is described in [38]. The sensor data can aid in searching image and video files in a more precise way. This paper presented a system called SEVA – Sensor Enhanced Video Annotation – which combines relativity, interpolation, and extrapolation techniques together. Using SEVA to generate a tagged stream which can be applied to achieve video search for particular objects. The authors performed some experiments with static and moving objects and a

moving camera.

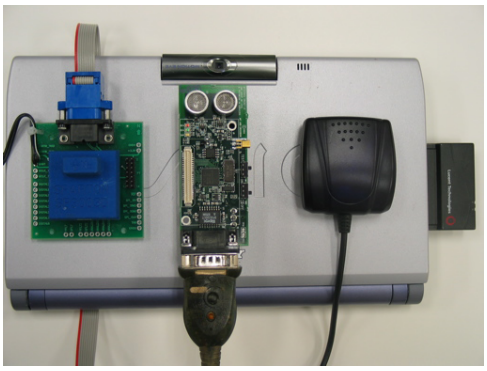


Figure 2.15: SEVA recorder laptop equipped with a camera, a 3D digital compass, a Mote with wireless radio and Cricket receiver, a GPS receiver, and 802.11b wireless.

Figure 2.15 shows the SEVA system with devices such as a 3D digital compass, a mote with wireless radio and a Cricket receiver, a GPS, and a camera. Compared with our acquisition system in Figure 1.2, the main difference is the wireless radio which can detect whether the object is obstructed. We can adopt such equipment in the future to make our search results more accurate.

2.5.2 Presentation of Videos

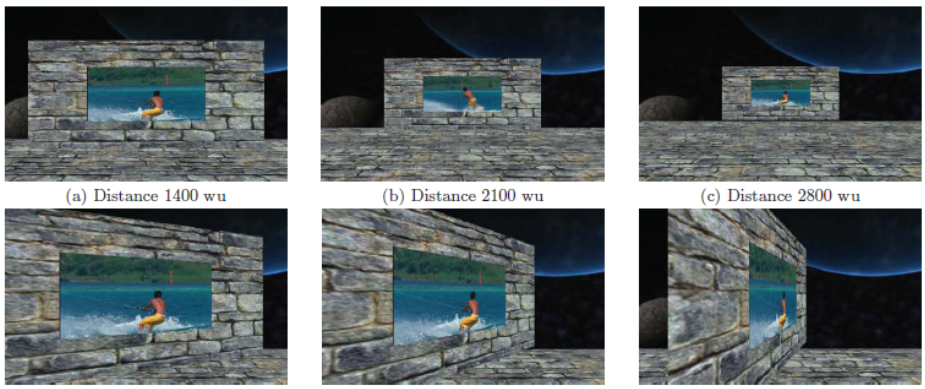


Figure 2.16: Sample screenshots from the prototype.

Nowadays, with the development of mobile devices, many applications have appeared in such environments. According to our research area, we have developed iPhone and Android applications to acquire meta-data with associated videos, and the main reason is that our previous work was implemented based on laptop computers which are not easily used to capture data. For mobile presentations of media,

there exist a number of related methods. For example, Shi *et al.* [56, 55] presented a view-dependent real-time 3D video for mobile devices. The authors emphasize that the 3D video systems need considerable bandwidth and computing resources. Therefore, a video compression technique is proposed. In addition, it is implemented on mobile devices which are very popular nowadays. However, this work is more related with the computer vision research area, and our direction is based on sensor information to present 3D perspective videos independent of the use of mobile devices or computers.

Similarly, Starck *et al.* [63] studied the use of 3D viewpoints which break through the traditional 2D video production. The work focuses on free-viewpoint video for consumers to interact with a scene. The challenge is to use a free-viewpoint to capture real world events and synthesize new content. Furthermore, Starck *et al.* combined image and video-based animations together to produce interactive animations from free-viewpoint video.

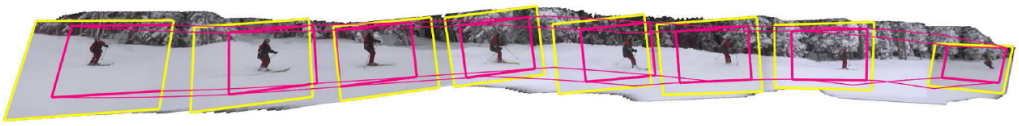


Figure 2.17: Schematic of the Re-cinematography process. Conceptually, an image mosaic is constructed for the video clip and a virtual camera viewing this mosaic is keyframed. Yellow denotes the source camera path, magenta (dark) the keyframed virtual camera.

Another similar method as above [45] shows video streaming in 3D environments and allows the display of several concurrently videos. However, for the same reason as above, the network is a bottleneck, therefore reducing the video quality becomes necessary. For example, when we use a video converter to reduce the video quality, although the frame seems blurry, the size is decreased at a high rate. Another contribution of this paper is shown in Figure 2.16, and it encourages us to develop our method in a 3D environment with a 3D perspective just like shown in the figure. In Second Life, it is easy for avatars to change their view angles; however in other 3D environments such as Google Earth, it is complicated to achieve this. Despite the convenience of the Second Life environment, there are some limitations. For example, it is not simple to do create objects, and the code files are large for implementation. Therefore, we choose Google Earth as our 3D environment. Sometimes, the FOV is established to capture a particular object, and then we can adopt the method proposed by Liu *et al.* [37] to retarget the video. Besides that, Gleicher *et al.* [12] shows another approach to improve camera activity with post-processing. Re-cinematography transforms each frame of a video to suit cinematic

conventions. Figure 2.17 shows the schematic of the Re-cinematography process. The videos are divided into segments with camera motions, and camera paths are keyframed automatically. In this figure yellow denotes the source camera path, and magenta (dark) indicates the keyframed virtual camera.

A method of spatial presentations of geo-referenced data in 3D space is introduced by Koiso *et al.* [30]. This method is an orientation based visualization model which is similar to ours. However, it only focuses on the data not on the video itself. In the authors' model, a spatial object is surrounded in a minimum bounding box (MBB) and based on this, an algorithm is obtained for determining the visualization priority by computing the weight value for each face of the box. Shown in Figure 2.18, there are 26 subdivided boxes adjacent to the surfaces of the MBB which is used to decide the orientation of view.

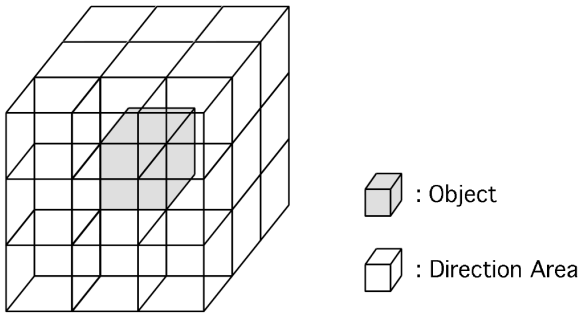


Figure 2.18: Orientation based visualization model using a minimum bounding box, MBB.

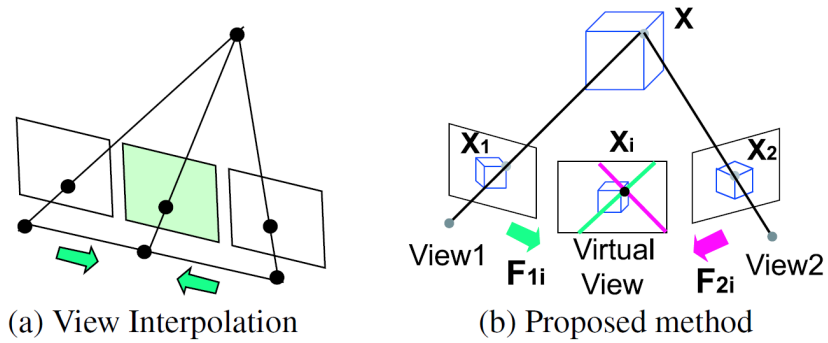
2.5.3 Obtaining Viewpoints of Videos

There exist a number of manuscripts related to obtaining a viewpoint of videos. For example, the Multi-View Synthesis (MVS) approach as described in [9] is scalable with virtual views and can handle scenarios whether the camera inputs decrease or increase. In general, producing virtual view is a significant problem for free viewpoint video systems. In a free viewpoint video, as known from virtual worlds, a user is allowed to freely navigate within real world visual scenes [60]. Therefore, we need to find a way to synthesize viewpoints with multiple view videos. Some research has focused on immersive free view video systems, some concentrate on view morphing, some use epipolar geometry, and some are concern with the coding and rendering of free viewpoint videos.

Kim *et al.* [25] have proposed a immersive free-view video system which can generate 3D video from a random point of view based on outer cameras and an inner omni-directional camera. Due to the property of the inner omni-directional camera,

reconstructing the 3D models can be more elaborate. This is because this type of camera covers a very wide FOV (field of view), and according to the movability, all the sub-cameras can be calibrated in real-time when one of the known markers is detected through one of sub-cameras. The employed techniques are related with computer vision and computer graphics.

Some other approaches are related with view morphing, for example, Zhang *et al.* [73] have proposed a method based on view morphing which resides between two extremes of geometry-based modeling and image-based modeling [57]. Using view morphing can produce an intermediate image plane on two original images, and it is able to predict the scenes. According to the authors, this method requires no setting up of dense cameras. On the other hand, it cannot be used in our research because it will change the original images. Besides the view morphing, there are some techniques such as the epipolar method which is proposed by Kimura *et al.* [28, 27]. The authors have presented a method for composing a viewpoint of multiple view videos for tennis. Typically, the view interpolation composes images from two viewpoints of real images to an intermediate viewpoint which imposes restrictions on the viewpoint position. In this paper, the viewpoint position is free and can be obtained from calculating the center of gravity of the player's region in the videos based on epipolar geometry. Figure 2.19(a) shows the view interpolation which is obtained from relative weights to two reference viewpoints. This method is limited by reference views with the relative weights. In contrast, Figure 2.19(b) presents a method without this limitation. This method divides a tennis scene into dynamic and static regions, and synthesizes a virtual view for every domain. The authors adopt an F-Matrix between the virtual view and the reference view which can map the corresponding points. This is a excellent way to obtain a viewpoint from multiple view videos, however, our method is based on the geo-referenced information which can calculate the viewpoint based on the location and direction information.



(a) View Interpolation (b) Proposed method

Figure 2.19: Transfer of corresponding points.

2.5.4 Video Compression

There exist some topics corresponding to compression and rendering of free viewpoint videos. Under the constraint of network bandwidth, we need to compress the videos or change the video quality based on the condition of the network. Especially, in our research area, we need to show multiple videos at the same time, and this issue becomes more important to consider.

Smolic *et al.* [61] proposed a complete system not only for FOV extraction, representation and rendering, but also for compression and transmission. Each aspect has its own technique, for example, extraction depends on a shape-from-silhouette algorithm, and representation is according to 3D mesh models. The authors' algorithms for view-dependent texture mapping is based on an extension of MPEG-4 (Moving Picture Experts Group) AFX (Animation Framework eXtension). As shown in their results, this complete transmission system is efficient and could be adopted in our system. However, in our work there is no compression issue because we do not handle too many videos. However, this should be considered in the future work.

Another compression named *dynamic point cloud compression* is proposed by Lamboray *et al.* [32, 33]. This coding framework can encode multiple attributes such as depth and color. However, it is an off-line process for encoding. The decoding part allows real-time rendering of the dynamic 3D point cloud. Under the preliminary results, the authors conclude that a 3D video stream can be produced with different bit rates because all data is progressively encoded. The specific codecs are not described in detail and the algorithm has some limitation. For example, the window length for coding needs to be further investigated.

Besides off-line rendering or on-line rendering, Nozick [46] proposed a novel VBR (variable bitrate) method which creates new views from moving cameras. These webcams can be calibrated in real-time based on multiple markers and the method is very efficient as it fully exploits both CPU and GPU. As can be seen, this method has the assumption that the markers must be co-planar. Besides, the markers should be preprocessed which requires efficiency improvements or the use of real-time calibration without markers.

Another method to overcome the network bottleneck is to use layered depth image (LDI) representation [71, 70]. It is necessary to propose an effective framework for compression since the data size of multi-view video is increasing quickly as the number of cameras grows. The authors presented a method to encode multi-view video data with 3D depth information based on layered depth images. Figure 2.20 shows the encoding procedure. First, from the multi-view video with depth information, LDI frames are generated, and residual data is sent to the decoder so as to reconstruct the images. Second, these LDI frames are separated into three com-

ponents: color, depth, and the number of layers (NOL). As shown in the figure, color and depth components need to be preprocessed, and NOL is encoded using the H.264/AVC intra mode. Finally, all the data is encoded with H.264/AVC. This approach is very useful for processing and encoding when the multi-view video data contains depth information.

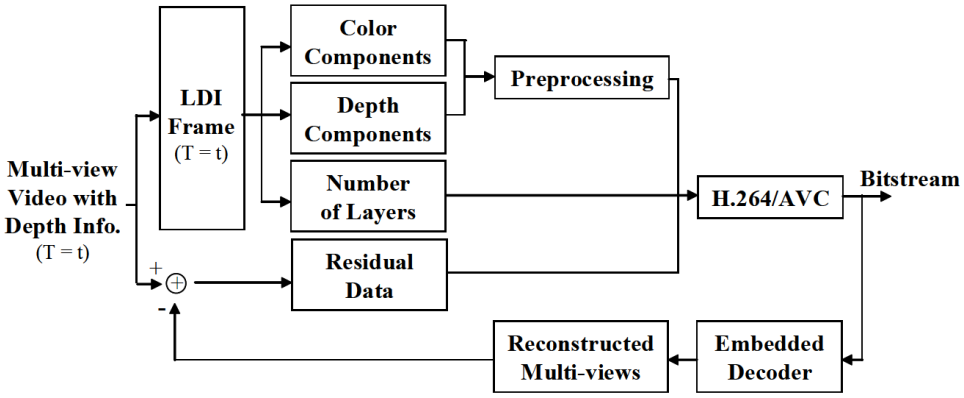


Figure 2.20: H.264 encoder block diagram.

2.5.5 Augmented Environments

To develop sensing and computing techniques, Sebe *et al.* [54] proposed a video surveillance application with augmented virtual environment (AVE). This environment combines dynamic imagery with 3D models to aim at displaying a real-time scene. The paper presented how to detect moving objects, how to track, and how to show 3D elements in the AVE scene. The components of AVE are shown in Figure 2.21.

In Figure 2.21, there are five important components:

- **Imagery acquisition**: The acquisition part is used to capture real-time videos.
- **Geometry model acquisition**: AVE visualization corresponds with the real world, hence, is using geometry model to extract complex building structures. Here the authors utilize an airborne LiDAR sensor system to collect the data.
- **Sensor tracking and calibration**: To preserve accurate registration between geometry model and video information the authors proposed a hybrid-sensor tracking method.

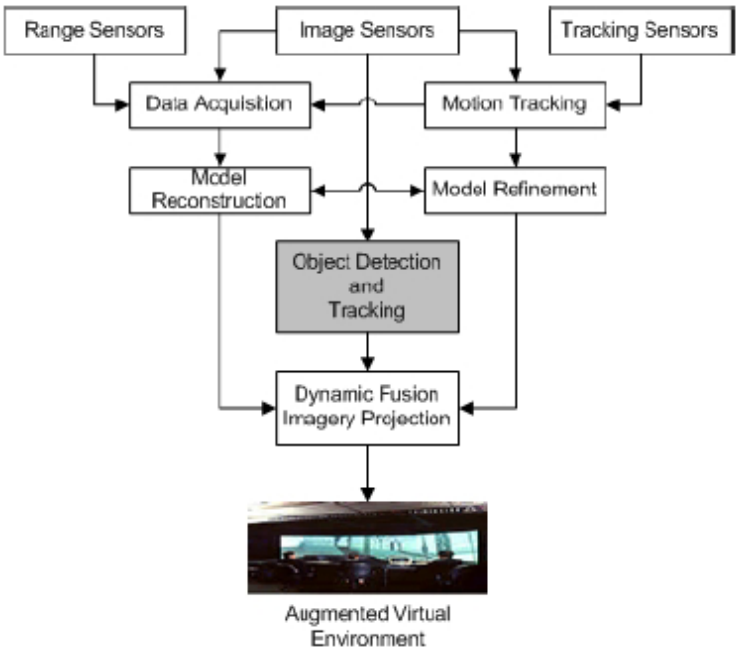


Figure 2.21: Components of the Augmented Virtual Environment (AVE) system with dynamic modeling.

- **Data fusion and video projection**: Real-time rendering of video streams can be projected onto a USC campus model.
- **Object detection and tracking**: Analyzing video imagery for tracking moving objects in the scene.

Kim *et al.* [24] presented methods for augmenting earth maps with dynamic information from videos. The authors proposed different approaches to identify the videos of pedestrians, sports scenes, and cars to augment Aerial Earth Maps (AEMs).

Figure 2.22 shows an overview of the approach and the most important three stages including Observation & Extraction, Registration & Correlation, and Visualization & Synthesis. In the first step, using video data to dynamically augment the AEMs, this should be achieved by extracting information from videos such as geometry information, location information, motion in the environment. The second step is to register the view from the videos to the corresponding view of AEMs. Considering the missing information, this requires designing models from data. The third step produces visualizations from the data, and using behavior simulation, procedural synthesis, view synthesis to synthesize the dynamic information on AEMs.

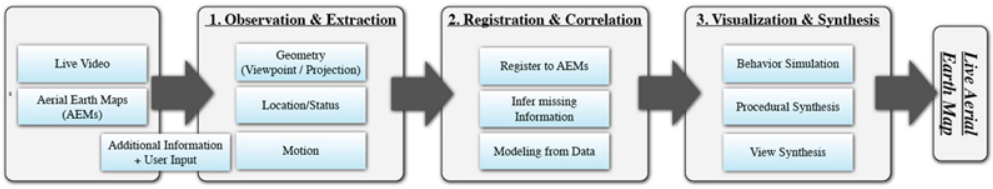


Figure 2.22: Overview of approach to generate ALIVE cities that one can browse and see dynamic and live Aerial Earth Maps, highlighting the three main stages of Observation, Registration, and Simulation.

2.5.6 Summary

In this section, we have described some related work that are concerned with video techniques. Table 2.2 shows the features of the technologies of these related works.

Techniques	*Features*
SEVA [38]	1. Designs a system that records identities and locations of objects along with visual images . 2. Presents detailed experiments to show the accuracy of the system.
TEEVE [56, 55]	1. Presents a view-dependent compression methodology to suit the 3D videos in mobile phone. 2. Introduces reference frame selection algorithms which are designed for 3D video rendering.
Video − Streaming − Virtual − World [45]	1. Performs videos in the 3D environment to test how to position the videos with user perception. 2. Reduces the resolution of videos to overcome the bottlenecks of network bandwidth.
Video − Retargeting − Re − Cinematography [37, 12]	1. Introduces Video Retargeting that adapts video to better suit the target display. 2. Minimizes information loss by balancing the loss of detail.

$Multi-ViewSynthesis(MVS)$ [9]	1. Proposes an approach that can be used in any multi-camera environment and is scalable as virtual views. 2. Gracefully handles scenarios whether the camera inputs decrease or increase.
$ViewInterpolation$ [28, 27]	1. Synthesizes a player-view video from multiple cameras to capture the tennis scene. 2. Proposes an efficient and robust approach to estimate the viewpoint of the player.
$VideoCompression$ [61, 32, 33, 46, 71, 70]	1. Overcomes the bottleneck of network bandwidth. 2. Adapts to the network condition according to change the quality of videos.
$Augmented-Environment$ [54, 24]	1. Proposes a new environment to fuse dynamic information obtained from videos. 2. Analyzes videos with different viewpoints to extract relevant information.

Table 2.2: The features of different techniques for videos.

Some sensor based videos have been collected in geo-spatial databases, and some researchers have studied these databases and presented work related to ours. As we have seen, some videos have location and timecode information, and some even have the range and direction information. Especially for the range information, if we can include this information in our videos, it will be more precise than before because we can check if there are some objects blocked before the objects that we want to identify. In addition, how to present the videos is also an issue we need to focus on. With 3D perspective videos, it is easier to accept when the users change their viewpoint. Furthermore, the viewpoint of multiple videos is another concern in our research. According to the requirements for showing multiple videos, considering the network condition is of importance, therefore, we have reviewed several techniques on video compression.

2.6 Conclusions

The technologies described in this literature survey are shown in a taxonomy in Figure 2.23.

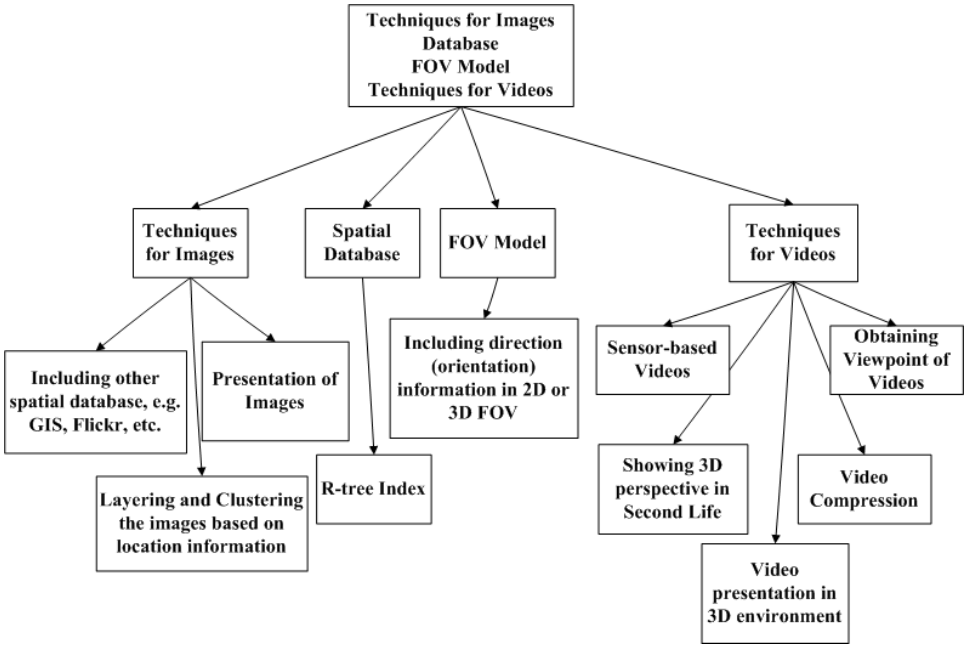


Figure 2.23: A taxonomy of related work technologies.

In the literature survey, we have reviewed many approaches related to our research area. Given these techniques, we can learn from them and adopt some of their ideas in our work. For example, we can apply hierarchical and clustering techniques to our presentation of videos, or rank the videos with certain rules to make the results more precise in our future work. In addition, the field-of-view concept was introduced by many researchers, and some research is based on geo-referenced information such as location (longitude, latitude), heading (orientation), and range data. This sensor information can help us to compose a 3D FOV which can be applied in our research. However, many existing methods neither query the videos in terms of sensor information, nor present 3D perspective videos within a 3D environment. Most of the related work uses geo-referenced metadata to search and cluster images not videos with certain datasets such as from Flickr on 2D maps. Therefore, there exists a considerable opportunity to extend these image techniques to video approaches.

Furthermore, our previous search engine is not efficient and practical because the algorithms, indexing and retrieval models are not mature. Hence we need to improve our technologies which are related with the indexing and modeling parts. Nowadays the metadata is not very large and we can query the videos without considering the

efficiency. However, in the future, we need to support large data collections, and therefore we will consider to adopt the R+tree to be our spatial indexing methods and the vector model proposed by Kim *et al.* [26] to our system.

For video presentations, there exists no relevant work for presenting 3D perspective videos within a 3D environment based on geo-referenced metadata. However, to achieve 3D perspective videos only, Apple Inc. has proposed this concept recently and others also use technologies to achieve this without embedding videos into 3D environments. On the other hand, there exist some research on 3D environments. For instance, showing video streaming in Second Life, and performing 3D video surveillance within Augmented Virtual Environments. All the above work does not provide an existing solution for the problem we are trying to address. Therefore, we propose our own system based on the Google Earth environment and with our own geo-referenced information to achieve effective video search. To present multiple videos, we propose an algorithm based on the geo-referenced sensor meta-data information. There are very few techniques related with our research that calculate the viewpoint based on GPS, compass information. However, there are many methods exploiting the concept of FOV to obtain the viewpoint based videos from multiple cameras. For example, we can use an interpolation technique in the future to make our viewpoint trajectory more smooth. In addition, with video compression technologies we can improve the efficiency of our system. Although we have not implemented this issue yet, this should provide a significant improvement when it is accomplished.

Chapter 3

System Overview

In this chapter we present an overview of our system. Section 3.1 shows the architecture of Geo-Referenced Video Search (GRVS) and also presents a data flowchart to illustrate the processing stages of our system. Section 3.2 presents data collection prototypes which were proposed in our previous work and achieved by Dr. Sakire Arslan Ay, Mr. Guanfeng Wang, and Dr. Beomjoo Seo [3, 72]. According to these data acquisition systems, we can collect the meta-data such as the location, orientation, and video timecode information. Section 3.3 describes the database implementation for our meta-data. The next two sections (Section 3.4 and Section 3.5) describe the most important parts of our system. 2D and 3D search engines are both designed based on a web browser with Google environments.

3.1 Architecture of GRVS

GRVS is a web-based video search engine that allows the searching of videos in specified geographical regions. Figure 3.1 illustrates the overall architecture of our search engine. The numbers (1) through (5) indicate the sequence of interactions between a client and the server. This architecture is suitable for both 2D and 3D search engines. Step (1) is to retrieve map or virtual world information from a Google environment, step (2) sends the query window from the client to the server, (3) performs the query with the database, (4) retrieves the query results, and (5) shows the location (the landmark or trajectory), orientation (the sector direction or 3D orientation) and video clips to the user.

Despite the similarity between both the 2D and 3D architectures, the scenario in the 2D and 3D search engines is somewhat different. In a typical scenario, a user marks a query region on a map, and the search engine retrieves the video segments whose viewable scenes overlap with the user query area. In the 2D geo-referenced video search engine, the server sends back meta-data with an XML file, and the

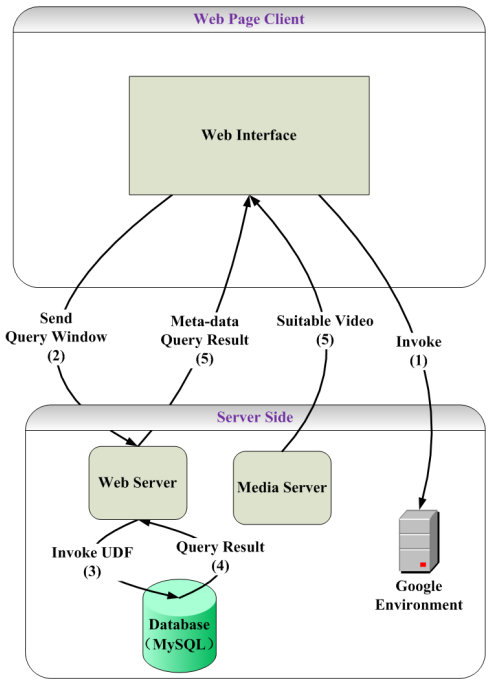


Figure 3.1: Architecture of geo-referenced video search.

client side shows pie slices and trajectories to users. However, in the 3D search engine, the query results are produced into a KML file and an XML file, and the video clips are rendered with a 3D perspective.

In our current implementation the query can be a rectangle or a trajectory. The search engine is comprised of three main components: (1) a database that stores the collected meta-data, (2) a web-based interface that allows the user to specify a query input region and then provides a display of the query results, (3) the interaction between client and server. The meta-data, we require the acquisition of videos that are fused with detailed geo-referenced information. The user interface for the 2D and 3D search engines are different, and more detailed information will be described in Section 3.4 and Section 3.5. In addition, the Google environments as in Figure 3.1 refer to Google Maps and Google Earth.

The described architecture concentrates on the query component. However we still need to describe how to store the meta-data. In our prior work we have proposed a video scene model in which videos are continuously augmented with detailed sensor data such as the current latitude and longitude (gained from GPS) and camera viewing direction (obtained through compass) [3]. The flowchart of GRVS is shown in Figure 3.2. This figure illustrates the input data and the output data streams. The data flow diagram (DFD) is complementary with the architecture figure, and the input data includes meta-data (GPS, compass and video information files), real

videos, and the query window. Additionally, the output data is composed of the query results which show the queried meta-data associated with the videos. Most of the data flow is very obvious except for the combining-data part because the frequency of sensor devices is different. In our setup, $f_{GPS} = 1$ sample/sec, $f_{compass} = 40$ samplels/sec, and $f_{video} = 30$ samples/sec. Therefore we match each GPS entry with the closest video frame timecode and compass direction [3].

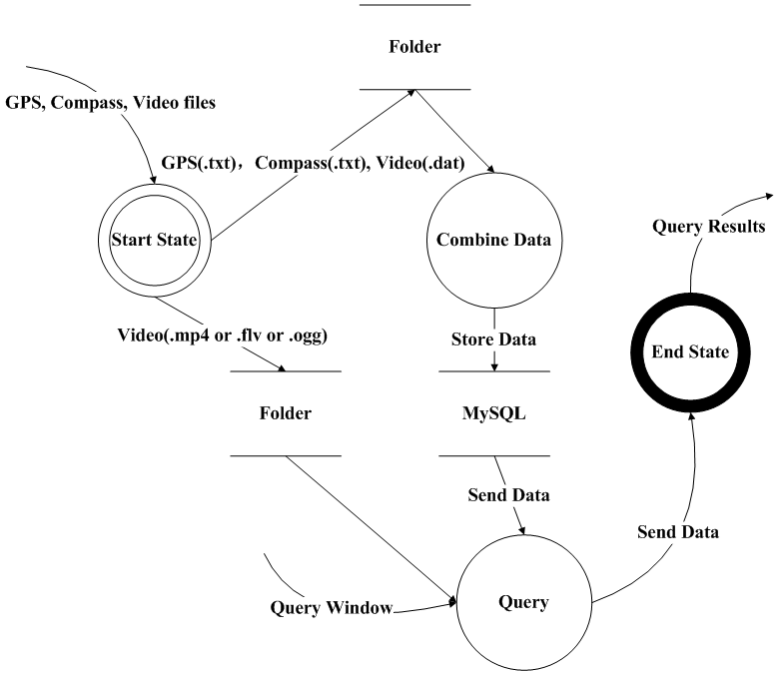


Figure 3.2: Data flow diagram of geo-referenced video search.

We have implemented the engine using the following open source software: XAMPP (Apache, MySQL, PHP and Perl), Wowza Media Server, and Flowplayer [39]. In addition, the languages are C, C++, JavaScript and PHP. Furthermore, the technologies used are Ajax, HTML 5, IFRAME shim [29], Google API, KML, XML and Drawing Graphics with Canvas.

3.2 Data Acquisition

To collect geo-referenced video data we implemented a light-weight acquisition software [3] that can concurrently acquire video, GPS and compass sensor signals while running on a laptop computer. Based on Microsoft Windows and its DirectShow filter architecture, different video formats are potentially supported. As mentioned before, this software is implemented in the previous work. Another prototype is implemented on an Apple iPhone 3GS handset which provides the necessary

built-in GPS receiver and compass functionality. This is developed by Mr. Guanfeng Wang, and similar prototype with Android is developed by Mr. Beomjoo Seo. These two phone based applications are more convenient than the PC based application.

These three prototypes are shown in Figures 1.2, 1.3, and 1.4. According to these applications, we can capture the videos with related meta-data including GPS, compass and video timecode. A Canon VIXIA HV30 camera, an OS5000-US solid state tilt compensated 3-axis digital compass, and a Pharos iGPS-500 GPS receiver constitute Figure 1.2. Particularly, the camera is used to acquire MPEG-2 encoded high-definition video via a FireWire (IEEE 1394) connection. The compass is used to obtain the orientation of the camera, and the GPS receiver is to acquire the camera location. This acquisition software records the geo-references along with the MPEG-2 HD video streams. The system can process MPEG-2 video in real-time (without decoding the stream) and each video frame is associated with its viewable scene information. In our experiments, an FOV was constructed once every second, i.e., one FOV per 30 frames of video. In addition, with iPhone application, to engage and control the built-in GPS receiver and magnetometer, we make use of the Core Location Framework in iPhone OS Core Services Layer. Location data consists of longitude and latitude and we can regard the position of the mobile phone exactly as the position of the camera. For the orientation information, however, we discovered an interesting difference between the true pointing direction and the device heading. Therefore, our iPhone application can fetch the accelerometer data from the UIKit Framework to determine an adjustment and ensure that the data that is recorded represents the camera's direction, even when the phone is held vertically. The user interface of iPhone application is shown in Figure 1.3. Furthermore, the Android prototype is demonstrated in Figure 1.4. With mobile phone applications, it can expand our data set. Based on large data set, our experiment results will be more convinced.

To obtain some experimental data sets, we mounted the recording system setup on a vehicle and captured video along streets. We recorded two sets of video data: (i) one in downtown Singapore and (ii) one in Moscow, Idaho. During video capture, we frequently changed the camera view direction. The acquired data set contains many video clips, ranging from 3 to 21 minutes in duration. At one second interval an FOV was collected, resulting in considerable FOVs in total.

3.3 Database Implementation

The meta-data is stored in the MySQL database. When the user uploads videos with meta-data into the GRVS system, the meta-data is processed automatically

and the viewable scene information is stored in the database. Our design can adapt to a variety of sensor meta-data information as is shown in Table 3.1. The attributes shown in this table can be applied into 2D and 3D FOVs. As shown in Figure 1.1 model (a), heading (the 2D direction), latitude, longitude, R (radius) and viewable angle are composed to 2D FOV. On the other side, model (b) in Figure 1.1 shows 3D FOV, the only difference is the 3D direction which is comprised of heading, roll and tilt. The collected 3D data basically represent the camera direction and location as a vector which describes a 3D field-of-view (FOV).

Once a query is issued, the video search algorithm scans the FOV tables to retrieve the video segments that overlap with the user-specified region of interest. Because of the irregular shapes of FOVs, we implemented several special-purpose MySQL *User Defined Functions* (UDFs) to find the relevant data. A separate UDF is implemented for each query type. Our initial search engine prototype supports two query types: spatial range queries (the query is a rectangular region) and trajectory queries (the query is a trajectory). The system architecture is flexible such that we can enhance the search mechanism and add support for other query types in the future. The video search algorithm is explained extensively in our prior work [3]. One current limitation is that only searches in 2D space are supported. Because of this, the altitude parameter is not implemented. In other words, the search is still performed on the 2D data for 3D search engine, however the results shown in 3D environments are applied with the 3D sensor data.

$filename$	Uploaded video file name
$\langle Plat, Plng \rangle$	<Latitude, longitude> coordinate for camera location (read from GPS)
$altitude$	The altitude of view point (read from GPS)
$theta$	Camera heading relative with the ground (read from compass)
R	Viewable distance
$alpha$	Angular extent for camera field-of-view
$tilt$	Camera pitch relative with the ground (read from compass)
$roll$	Camera roll relative with the ground (read from compass)
$ltime$	Local time for the FOV
$timecode$	Timecode for the FOV in video (extracted from video)

Table 3.1: Schema for 3D field-of-view (FOV) representation.

3.4 2D Search Engine

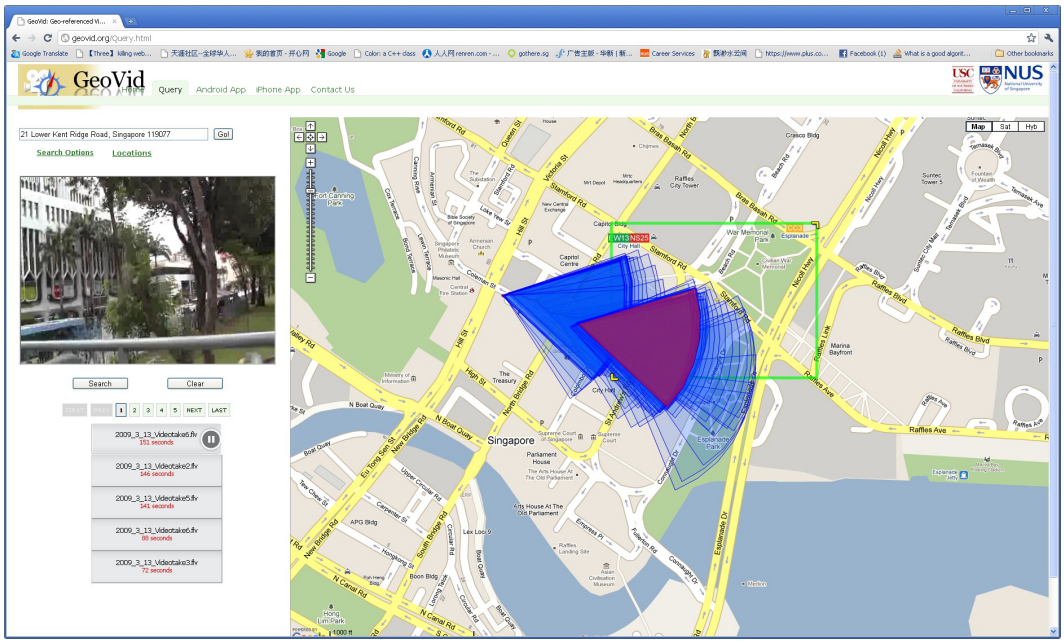


Figure 3.3: Geo-referenced 2D video search engine web interface.

The following sections describe the 2D geo-referenced video search engine in detail. The implemented environment is Google Maps which is a 2D web-based mapping tool (map) to view real world.

3.4.1 Web Interface

The map-based query interface allows users to visually draw the query region. The results of a query contain a list of the overlapping video segments according to the user-specified region of interest. For each returned video segment, we have displayed the corresponding FOVs using Google Maps. To reduce clutter, we draw the FOVs every two seconds. The user can watch the video clips through this 2D environment. Note that the video server precisely streams the video section that is shown in the query region, not the complete video file. During video playback, the FOV whose timecode is closest to the current video frame is highlighted on the map. Each FOV is associated with a video frame timecode, which ensures a tight synchronization between the video playback and the FOV visualization. A sample screen shot of the web-interface is shown in Figure 3.3.

We have implemented the web interface using JavaScript and the Google Maps API [15]. For video playback, we have introduced the Flowplayer [39] which is an open source flash media player. The video files were transcoded into *flv* format or

mp4 format. Note that our search engine implementation is platform independent. We successfully deployed our search engine on both Linux and Windows servers.

Applying Google API to draw query window (in our case is the rectangle) and show pie slices is a good strategy to implement our system. Users can watch videos with corresponding sectors which can identify the region overlapped with the video scene. The technologies of drawing a rectangle are "GLatLng", "GMarker", "addOverlay", "removeOverlay", "setPoint", "savePoint", "GPolyline", and all of them are Google API [15]. Showing pie slices is achieved by function "drawCircle" which is also Google Maps API. Therefore, we do not describe in detail here.

3.4.2 Communication between Client and Server

In our system, exchanging data between the client and server is very important. The main technique in this client-server interaction is coded with Ajax. Ajax techniques are used to send the query window to the backend applications and to retrieve the query results to the frontend applications. With Ajax, web applications can obtain data from the server asynchronously in the background without interfering with the display. At the same time, because of the properties of Ajax, we can establish a dynamic interface for the web page [67]. The communication with the MySQL database and the UDFs is provided via PHP. Furthermore, the meta-data is produced as an XML file shown in Figure 3.4. According to this figure, the meanings of each data are quite clear. The data included in "<show>" forms field of view which also means the pie slice shown in Figure 3.3. Additionally, the data in "<video>" is applied to extract video segments. In 2D search engine, the Flowplayer and Wowza Media Server can achieve this based on the start and end time.

3.4.3 Video Management

The format extracted from our data acquisition software is MPEG-2. However, the size of videos of this format is very large which is unbefitting in real applications. Therefore, it is more reasonable to transform MPEG-2 to FLV format. In addition, the quality of FLV videos is not bad, and the size is much less than before. The tool used to convert the video is *FFmpeg* which is a cross-platform software. Furthermore, FFmpeg is a very fast video and audio converter [11]. If the network condition is good, we will use original format to avoid the complexity of converting large size of videos.

To play *flv* files, the Flowplayer is a perfect player to achieve this [39] which is an open source video player and can be used to embed video streams into our web pages. Furthermore, playing video segments and showing a playlist can be implemented with Flowplayer which makes the interface more friendly. However, without media

```
<?xml version="1.0" encoding="UTF-8"?>
<queryresult>
  <show>
    <FileName>2009_3_13_Videotake5.ogg</FileName>
    <Latitude>1.300775</Latitude>
    <Longitude>103.840773333</Longitude>
    <Heading> 181.8 </Heading>
    <Radius>0.259081</Radius>
    <FOV_Angle>60</FOV_Angle>
  </show>

  <video>
    <VideoFileName>2009_3_13_Videotake5.ogg</VideoFileName>
    <Start>444</Start>
    <End>596</End>
  </video>
<queryresult>
```

Figure 3.4: Sensor meta-data exchanged between client and server. The XML file includes GPS coordinates, compass heading, radius, view angle, and video segment information (start time, duration, and video file name).

server, showing video segments cannot be accomplishable. With this reason, Wowza Media Server was introduced, and this is the only proven high-performance and the first unified media server [69]. The Wowza Media Server allows stream of video content, similar as the Adobe's Flash media server. Using this combination of media server and video player, any segment within a video, specified by starts and end timestamps, can be played. We use this feature to extract the most relevant clips from videos which may potentially be very long and cover a large geographical area.

3.5 3D Search Engine

The following sections systematically present the 3D geo-referenced video search engine, especially for the differences between the 2D and 3D search engine. In 3D search engine, the primary environment is Google Earth instead of Google Maps. However, we still need Google Maps to aid in specifying query region.

3.5.1 Web Interface

Perspective video, i.e., transforming video from a 2D plane into a projected plane in a 3D virtual space in accordance with the user's viewpoint, is one of the major tasks for web-based video overlapping applications. In this domain, there exist

several viable solutions:

- Existing plug-in-based Rich Internet Application (RIA) technologies such as Adobe Flash and Microsoft Silverlight support 3D video rendering capabilities. While available for rapid prototyping, these environments require overlapped web services to provide corresponding RIA-compatible APIs.
- Pixel-level image transformation is also a feasible solution, but it requires significant client-side processing power.
- A Cascaded Style Sheets (CSS) 3D transform has been proposed by Apple Inc., and it is now under development by the W3C CSS level 3 [51]. This method transforms the coordinate space of a video element through a simple change of its transform properties.
- An IFRAME shim can establish a layer on top of the Google Earth web browser plug-in (or other web pages). The IFRAME can aid in the process of rendering videos, and is flexible in any environment. Without this technology, we cannot watch the videos with an appropriate viewpoint.

Considering both practicality and feasibility, we choose the IFRAME shim approach as our main technique to overlay 3D perspective video. Hence, when the viewing direction changes by a certain angle, either the video changes accordingly or the Google Earth background moves to match the 3D orientation. Furthermore, the camera trajectory will also be shown in the 3D worlds. With the presentation of the trajectory, the users will explicitly follow the camera movement associated with the video.

Figure 3.5 shows the video results using our 3D Geo-Referenced Video Search engine. As can be seen, the web browser interface embeds Google Earth and Google Maps on the same page. Superimposed on top of Google Earth are our 3D perspective video segments, while in the lower left bottom is the tour progress bar. The indicator in the progress bar points out the corresponding position within the time interval.

In Google Earth, the number of modeled 3D buildings varies among different cities, but overall the number is steadily increasing. When 3D building structures exist, we can more convincingly overlay the captured video with the virtual world. When viewing these buildings we see whether the scene in a video matches the same position in the virtual world. We can also observe how accurate these 3D buildings have been modeled. One of the challenges in virtual environments is that it may not be very easy to specify the user's region of interest (i.e., the query area). For example, currently Google Earth does not support the specification of a spatial query

Figure 3.5: Geo-referenced 3D video search engine web interface showing multiple videos simultaneously.

rectangle to delineate a search area. For this reason – and because a query area is more naturally expressed in 2D – we use Google Maps to let a user select a query window. The search results are then shown properly placed in Google Earth.

There are a number of techniques applied in our web interface. To embed Google Earth and Google Maps in the same web page, we use Ajax. To implement the interaction between Google Earth and Google Maps interfaces, we introduce the *google.earth namespace*, the *GEView interface*, and the *Maps API GPolygon* [14]. The specific details of each technique are given below.

- The *google.earth namespace* contains global functions to support to the use of the Earth API interfaces. We attach a listener to Google Earth for a specific event, which means that if Google Earth moves, the program will be aware of the movement and simultaneously move Google Maps.

- The *GEView interface* checks the view behavior of the observer camera in Google Earth. There is a function that can return a global view region. It is noteworthy that the returned region may not be very accurate because it will be larger than what is strictly visible.

- *Maps API GPolygon* is an interface to create a polygon in Google Maps. Through this the users directly gain a view of the query region.

3.5.2 Communication between Client and Server

Similar as Section 3.4, 3D search engine also needs communication between client and server. In addition, the main technique is still Ajax. However, the data format or files exchanged between client and server are different. The meta-data of query results is stored in a KML file and an XML file. Firstly, the KML file allows automatically invoke an animated tour through Google Earth. This is a relatively new capability of Google Earth that can automatically traverse the background of the environments. Secondly, the XML file includes view angles, position information, start time and end time of video segments, and video file names.

The XML file format is shown in Figure 3.6, and the only difference is when there is more than one video. According to this figure, on the client side, the users can watch at most four videos at the same time. The video length is different, and the video clips will be automatically displayed. When one video completes, it will be marked as finished and other videos will continue to play until finished.

In addition, a KML file will be produced when the server send the query results to the client. The explanation of the KML is shown in the following:

- The data inside "<gx:Tour>" shows the tour in Google Earth. In other words, the system can automatically tour in the environments through tour tags. Tours are constructed by placing specific elements, in order, into a KML file [50].
- The data within "<LookAt>" tags defines the view of a virtual camera look at the objects. As described in [50], "in Google Earth, the view "flies to" this LookAt viewpoint when the user double-clicks an item in the places panel or double-clicks an icon in the 3D viewer".
- In "<gx:Wait>", the number indicates the wait duration.
- The data encompassed in "<Placemark>" is for trajectory. Through this trajectory, users can easily see the accurate position of the tour.

3.5.3 Video Management

HTML 5 video is the main video technique used in this 3D search engine, and it becomes the new standard way to show videos online. However, it has been impeded by lack of agreement with video format. The video format of HTML 5 is very varying of different web browsers. For example, using *Ogg Theora* format with Mozilla Firefox browser, *H.264* format with Safari browser, and Google Chrome can support both of them [67].

3D Perspective Video

In 3D geo-referenced video search, the presentation of video with a 3D perspective on top of Google Earth is a very complex issue. Besides, it only can be implemented with HTML 5 video. The main technique is *Drawing Graphics with Canvas* of JavaScript. *Canvas.drawImage()* is the most important API which can get the contents of specific HTML 5 elements and use a canvas to put them in. Additionally, handling videos means copying them into a canvas element and manipulating or processing video frames during runtime. However, it consumes much CPU during processing. For instance, a computer with an Intel(R) Core(TM) 2 Quad processor and 4G RAM, this program costs 25 percent of CPU.

Despite the code is not efficient, we still refer it because this is the exclusive example implemented in Windows instead of Mac OS X. To make the program more effective, it follows like: [Video playing]→[Draw Video onto Canvas 1] →[Draw fragments of Canvas 1 onto Canvas 2]. The reason of implementing this is copying pixel data without video tag is very expensive, and so as drawing into a temporary canvas. Therefore, using a final canvas to get from temp canvas is more quickly than using video tag to repeat. Furthermore, *Canvas.drawImage() + matrix* transforms are helpful and efficient compared with *getPixel()* and *setPixel()* [8].

Video Segments

For video segments, same as in the 2D search engine, we show video clips instead of the whole videos respected to the query results. This is because if we query in a certain region, and most of time only the video segments intersect with this region, then we would like to display video segments to increase the efficiency. One solution is to divide the whole video into several video segments. However, for each query, dividing videos increases the complexity. In consideration of this, we find a way to solve this: using seek time, an attribute of HTML 5 video. When the client receives start time and end time of a video segment, the server can stream the video clip. Then we embed the start time into *url* and use the attribute of *url* to split and extract the time. For the end time, we add an listener named "addEventListener" with an attribute "timeupdate" and a function "stop"(the end time). This function can monitor the event, when the time updates, the system react with certain status. A very important attribute is "currentTime", and this can compare with end time. According to these techniques, we can obtain video segments without cutting the videos into pieces. Some other attributes: "pause", "play", "load", and forth are also used in our program [16].

3.5.4 The Algorithm for Presentation of Multiple Videos

We have proposed an algorithm to compute a viewpoint of multiple videos. Most of time, the query results contain multiple videos, then we need to consider how to present these videos.

The viewpoint may show one, two, three, or four videos. More than four videos shown simultaneously is not a good idea, therefore, we only consider the cases with no more than four videos. The reason is that if there are too many videos showing at the same time, it will confuse the users. Based on the user study, four videos shown simultaneously is more than enough. Supposing that there are too many videos, the users will feel difficult to decide which one to watch.

To deal with two videos or more, we have different rules. One video is regarded as a 2D straight line which is composed of the camera location and direction information. Based on this consideration, we can calculate a viewpoint for two videos with the intersection of the two straight lines. However, in fact, it is not enough to just consider with the intersection of two lines because sometimes there is no intersection between two lines, sometimes the intersection is too far away from the camera location, and sometimes the intersection is out of range which means the scene viewed from the intersection is somehow the opposite direction of the original direction.

According to the different situations, we have come up an algorithm to calculate the viewpoint:

Firstly, when the tangent of direction does not exist, which means the angle is either 90 degrees or 270 degrees, there are still many situations we need to consider: one issue is parallel, one is the intersection is too far away from the original camera position, one is out of range, and another one is the quadrant. As shown in Figures 3.8(a) and (b), these are some instances of parallel. Figure 3.8(c) shows the situation for the intersection is far away from two camera positions, and then we move the viewpoint near to the original camera position. Based on this operation, the viewpoint trajectory is smoother, and the video scene can be seen in this reasonable viewpoint. Furthermore, in (d), it is the situation for out of range, our line is radial and therefore, the intersection is not in the range of the radial. In such situation, we just calculate the intersection, and mark the video as indicating the wrong or opposite side of the background for this video. Finally, in (e) and (f), to plus or minus a certain distance depends on which quadrant it is in.

Secondly, when the directions are in the same or opposite for both videos and the angle is neither 90 degrees nor 270 degrees. In the same direction, the only consideration is quadrant. The different situations are shown in Figure 3.9, there is no intersection between the two lines. On the other hand, as the directions are

opposite, there is no intersection either. Figure 3.10 shows different situations when the directions of two videos are opposite. Figure 3.10(a) demonstrates the case when the viewpoint is not in the radial of two videos, and we obtain the viewpoint and present the videos as the opposite of the scene. As shown in Figures 3.10(b) and (c), the scene watched from the viewpoint in (c) is better than in (b). We obtain the midpoint of two videos, and plus or minus a certain distance of this point depends on the direction of videos. There is a tradeoff between the correctness and smooth trajectory. If we just ignore one video which is not good to present, it will dynamically change the number of videos and the trajectory will not be smooth. Our choice is to keep the videos based on the user study.

Finally, besides the above situations, there are many other general cases with intersection of two straight lines. Equation 3.1 is the equation for a straight line, "y" means the parameter of vertical axis, "x" means the parameter of horizontal axis, "dir" means the angle of this line, "tan(dir)" means the gradient of the line, "py" means the value of vertical axis for a specific point, and "px" means the value of horizontal axis for the specific point. With two straight lines, we can calculate the intersection using Equation 3.2 and Equation 3.3. In these equations, "x" means the value for the intersection of horizontal axis (latitude) between two lines, "y" means the value for the intersection of vertical axis (longitude), "py0" and "py1" is the value of vertical axis, "px0" and "px1" is the value of horizontal axis, and "tan(dir0)" and "tan(dir1)" is the gradient for each line. Based on these equations, we can calculate the viewpoint. However, it is not enough because there are many special cases as similar as in Figure 3.8. Figure 3.11 shows the general cases for two videos, (a), (b), and (c) presents the cases when the directions for two videos are in the same quadrant, (d), (e), (f), (j), (k), and (l) presents the cases when the directions for two videos are in the adjacent quadrant, and (g), (h), and (i) presents the cases when the directions for two videos are in the opposite quadrant. We can ignore the parallel case in this condition, and consider the case of intersection is too far away from the original camera position, the case for out of range, and another case for the quadrant. The best situation for multiple videos is shown in Figure 3.12. Based on our algorithm, users can simultaneously watch multiple geo-referenced videos.

$$y = \tan(dir) * x + py - \tan(dir) * px \tag{3.1}$$

$$x = \frac{py0 - py1 - \tan(dir0) * px0 + \tan(dir1) * px1}{\tan(dir1) - \tan(dir0)} \tag{3.2}$$

$$y = \frac{(py0 - \tan(dir0) * px0) * \tan(dir1) - (py1 - \tan(dir1) * px1) * \tan(dir0)}{\tan(dir1) - \tan(dir0)} \tag{3.3}$$

In many cases, we have some rules to present the videos which are under the consideration of correctness, efficiency and effectiveness. Algorithm 2 shows the high level description of our algorithm. The details are described in the above paragraphs and the detailed algorithms are shown in Algorithm 3, Algorithm 4, Algorithm 5, and Algorithm 6. On the whole, if two directions are parallel or too far away from the original camera position, then obtain the viewpoint according to plus or minus a certain distance, if the viewpoint is out of range, then the viewpoint is the intersection or plus or minus a certain distance, and in general cases, calculate the viewpoint according to different quadrant.

Input: Location Information of Two Videos (px0, py0, px1, py1),
Direction Information of Two Videos (dir0, dir1)
Output: Viewpoint

```
1  if dir0=90 OR dir0=270 OR dir1=90 OR dir1=270 then
       // the direction is either 90 degrees or 270 degrees
2      Algorithm 3;
3  end
4  else if dir0=dir1 then
       // the directions of two videos are the same
5      Algorithm 4;
6  end
7  else if abs(dir0-dir1)=180 then
       // the directions of two videos are opposite
8      Algorithm 5;
9  end
10 else
       // General case
11     Algorithm 6;
12 end
```

Algorithm 2: Calculation of the viewpoint of two videos.

Input: Location Information of Two Videos (px0,py0,px1,py1), Direction Information of Two Videos (dir0, dir1)
Output: Viewpoint

```
1  if parallel then
       // the directions of two videos are parallel
2      Plus or minus a certain distance;
3  end
4  if too far away then
       // the intersection is too far away from the original
       camera position
5      Plus or minus a certain distance;
6  end
7  if out of range then
       // the intersection is out of range
8      Keep the intersection, and indicate the video as opposite side of the
       background;
9  end
10 if general then
       // consider the quadrant
11     Keep the intersection or plus or minus a certain distance;
12 end
```

Algorithm 3: Calculation of the viewpoint when the direction is either 90 degrees or 270 degrees.

Input: Location Information of Two Videos (px0,py0,px1,py1), Direction Information of Two Videos (dir0, dir1)
Output: Viewpoint

```
1  Obtain the viewpoint according to different quadrant as in Figure 3.9;
```

Algorithm 4: Calculation of the viewpoint when the direction is the same.

Input: Location Information of Two Videos (px0,py0,px1,py1), Direction Information of Two Videos (dir0, dir1)
Output: Viewpoint

```
1  if out of range then
       // the viewpoint is not in the radial of the videos
2      Plus or minus a certain distance, and indicate the video as opposite
       side of the background;
3  end
4  if general then
       // consider the quadrant
5      Plus or minus a certain distance;
6  end
```

Algorithm 5: Calculation of the viewpoint when the direction is opposite.

```
<?xml version="1.0" encoding="UTF-8"?>
<queryresult>
  <showmultiple4>
    <FileName0>2009_3_13_Videotake2.ogg</FileName0>
    <FileName1>2009_3_13_Videotake3.ogg</FileName1>
    <FileName2>2009_3_13_Videotake5.ogg</FileName2>
    <FileName3>2009_3_13_Videotake6.ogg</FileName3>
    <Latitude>1.29192241667</Latitude>
    <Longitude>103.855525733</Longitude>
    <Latitude0>1.29344266667</Latitude0>
    <Longitude0>103.855377167</Longitude0>
    <Latitude1>1.29093833333</Latitude1>
    <Longitude1>103.8553625</Longitude1>
    <Latitude2>1.29351616667</Latitude2>
    <Longitude2>103.855323333</Longitude2>
    <Latitude3>1.29072866667</Latitude3>
    <Longitude3>103.8555525</Longitude3>
    <Heading>127.0225</Heading>
    <Heading0>171.93</Heading0>
    <Heading1>82.79</Heading1>
    <Heading2>159.09</Heading2>
    <Heading3>94.28</Heading3>
    <Radius>0.259081</Radius>
    <FOV_Angle>60</FOV_Angle>
  <\showmultiple4>

  <video>
    <VideoFileName>2009_3_13_Videotake2.ogg</VideoFileName>
    <Start>322</Start>
    <End>337</End>
  </video>
  <video>
    <VideoFileName>2009_3_13_Videotake3.ogg</VideoFileName>
    <Start>0</Start>
    <End>56</End>
  </video>
  <video>
    <VideoFileName>2009_3_13_Videotake5.ogg</VideoFileName>
    <Start>814</Start>
    <End>893</End>
  </video>
  <video>
    <VideoFileName>2009_3_13_Videotake6.ogg</VideoFileName>
    <Start>258</Start>
    <End>342</End>
  </video>
<queryresult>
```

Figure 3.6: Sensor meta-data exchanged between client and server. The XML file includes GPS coordinates, compass heading, radius, view angle, and video segment information(start time, duration, and video file name) for multiple geo-refernced videos.

```
<?xml version="1.0" encoding="UTF-8"?>
<kml xmlns="http://www.opengis.net/kml/2.2"
xmlns:gx="http://www.google.com/kml/ext/2.2">
  <Document>
    <gx:Tour>
      <name>Show results</name>
        <gx:Playlist>
          <gx:FlyTo>
            <gx:duration>1.0</gx:duration>
            <gx:flyToMode>smooth</gx:flyToMode>
            <LookAt>
              <longitude>103.855525733</longitude>
              <latitude>1.29192241667</latitude>
              <altitude>26.1</altitude>
              <heading>127.0225</heading>
              <tilt>358.1</tilt>
              <roll>355.7</roll>
              <range>400</range>
              <altitudeMode>absolute</altitudeMode>
            </LookAt>
        </gx:FlyTo>
          <gx:AnimatedUpdate>
             <Update>
               <targetHref/>
               <Change>
                 <Placemark targetId="0">
                    <gx:balloonVisibility>1</gx:balloonVisibility>
                 </Placemark>
               </Change>
             </Update>
          </gx:AnimatedUpdate>
          <gx:Wait>
             <gx:duration>14</gx:duration>
          </gx:Wait>
        </gx:Playlist>
    </gx:Tour>
      <Placemark id="0">
          <name>0</name>
          <Point>
           <gx:altitudeMode>absolute</gx:altitudeMode>
           <coordinates>103.855525733,1.29192241667,0</coordinates>
         </Point>
     </Placemark>
  </Document>
</kml>
```

Figure 3.7: Sensor meta-data produced by server, and invoked by client. The KML file includes GPS coordinates, compass heading, waiting time, and trajectory.

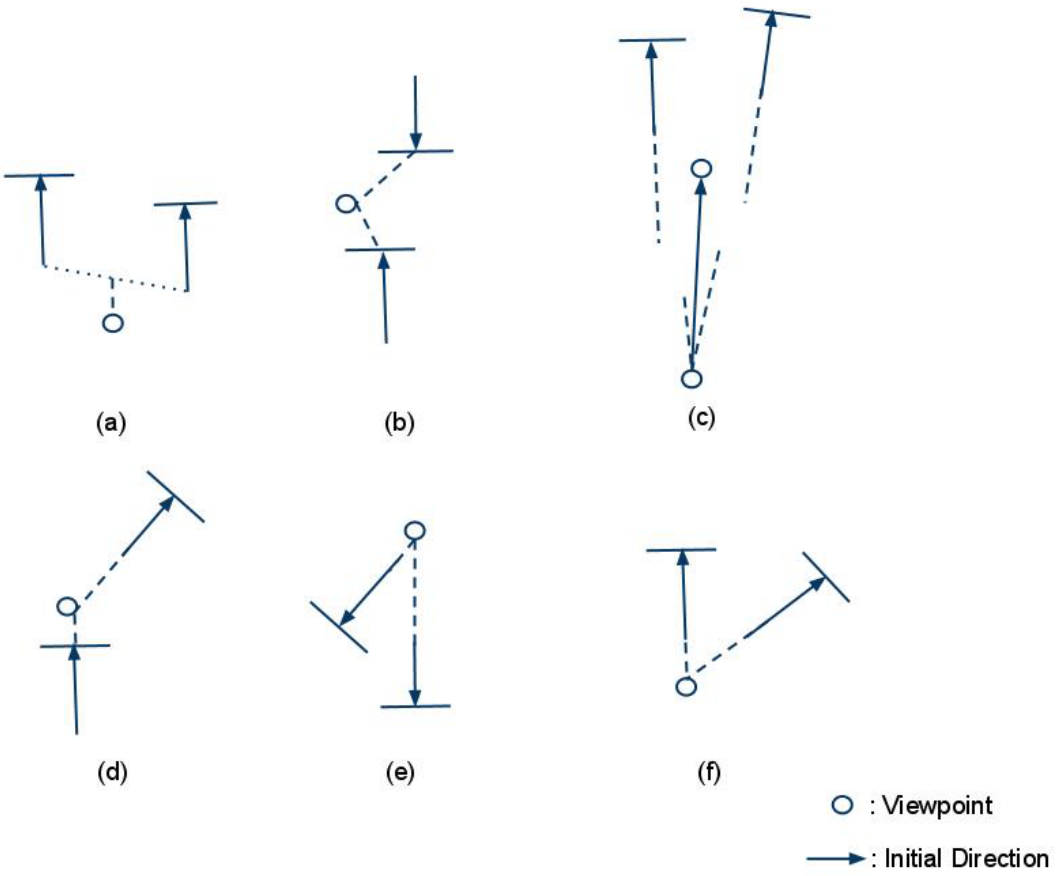


Figure 3.8: Different situations when either of the direction is 90 degrees or 270 degrees.

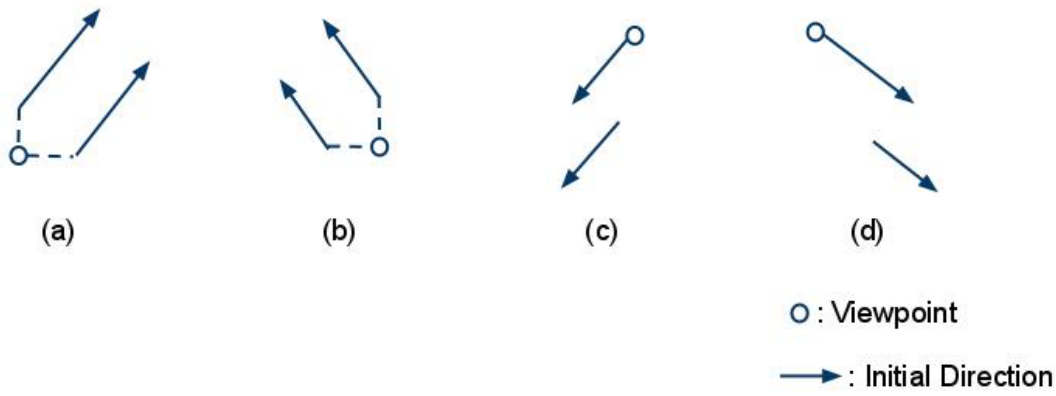


Figure 3.9: Same direction for two videos to compute viewpoint.

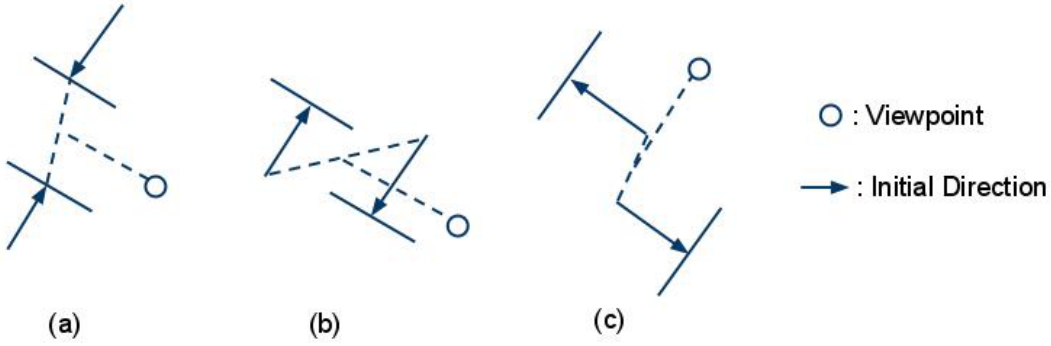


Figure 3.10: Opposite direction for two videos to compute viewpoint.

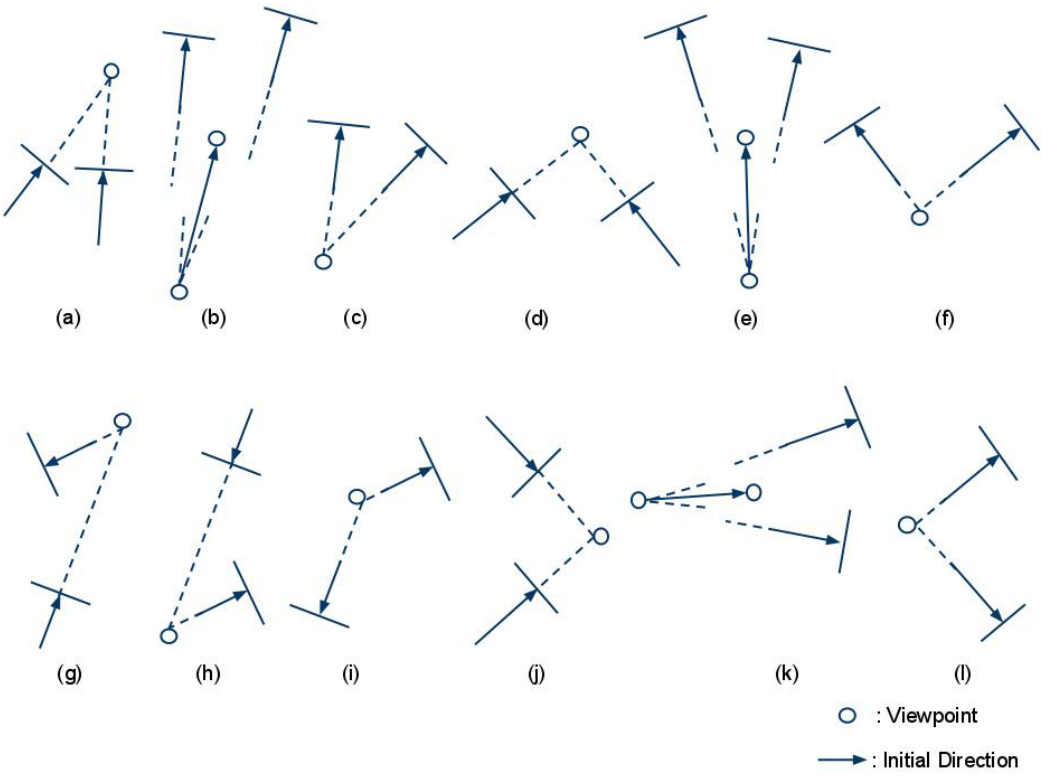


Figure 3.11: General case for two videos to compute viewpoint.

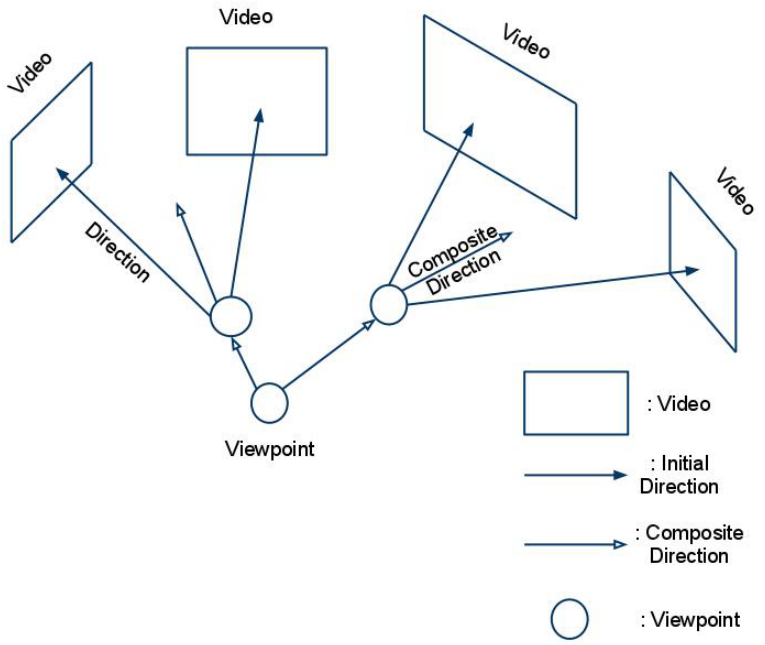


Figure 3.12: Best situation to compute viewpoint of four videos.

Input: Location Information of Two Videos (px0,py0,px1,py1), Direction Information of Two Videos (dir0, dir1)
Output: Viewpoint

```
1 if too far away then
    // the intersection is too far away from the original
    camera position
2   Plus or minus a certain distance;
3 end
4 if out of range then
    // the intersection is out of range
5   Keep the intersection, and indicate the video as opposite side of the
    background;
6 end
7 if general then
    // consider the quadrant
8   Keep the intersection or plus or minus a certain distance;
9 end
```

Algorithm 6: Calculation of the viewpoint in the general case.

Chapter 4

Evaluation

To identify the accuracy of our algorithm, we have established an experiment and tested with different situations. We have implemented this algorithm with C++ in order to easy to debug and visually present the trajectory. However, to figure out the correctness of our system, we move this algorithm back to the server side. In the following parts, we have presented the experiment results, and also shown the discussion and analysis of our algorithm.

4.1 Experiment Design

We have run the system with one, two or three video query results. With running the PHP code, we have recorded the query window (the longitude and latitude information) to manually run our C++ code with Windows Programming. According to different number of videos, the trajectory will be shown on the window which can identify our algorithm. In addition, with the web interface, we have recorded the location (the position of query rectangle) with different video number, and manually move the query window to the specified location. After that, click the "Query" button, and the system will return the query results with the expected number of videos. Then we can obtain the screen shots, and visually compare the results of different number of videos.

There are only a few videos in our system. In other words, the meta-data is not very large in our database. Then it is not difficult to manually test our system with specified number of videos. However, in the future, our system will be mature, and the data will be very large. At that time, we need to consider another way to test our system.

4.2 Experimental Results

Our query results are composed of several videos and FOV information, and we have proposed an algorithm to show multiple geo-referenced videos. The time duration is different for different video clips. To deal with this situation, when one video finish to play, it will stay there until other videos finished. As can be seen in Figure 4.1, Figure 4.2, and Figure 3.5, our multiple videos can be shown simultaneously and with the KML file, we can dynamically present the viewpoint. Using the tour tag can move the background to the suitable position and match the video scene. In these figures, the left side is the Google Earth, on top of it is the balloon which is the marker to indicate the viewpoint, the bar at the left bottom corner is the tour time identifier, and the videos are on the IFRAME shim which seems like the scene with white background. In addition, on the right side, it is the Google Maps. Using this map, we can specify the query window, and the server can send the results back to the web interface.

Figure 4.1: Showing one geo-referenced video.

For simply debugging, we have implemented our algorithm with C++, and in Figure 4.3, we have shown the trajectory of three videos for different cases. In this figure, the black circle is the viewpoint we obtained and the different colors show the trajectory and the direction. Besides, the different colors present the different videos. In (a), it is the trajectory without viewpoint which is used to compare with others. To visualize our results, we have mapped the trajectory to 600*800 window because the location difference is very small to see. Based on our algorithm, we have

Figure 4.2: Showing two geo-referenced videos simultaneously.

presented the viewpoints in (b) which is under consideration of different conditions. In addition, the intersection is too far from the camera positions but we did not deal with it in (c), and in (d), we have not solved the problem when the intersection is out of range. Comparing the different situations, in (c) and (d), the viewpoint is very far from the camera positions. In our algorithm, we have handled this problem which makes our trajectory smoother.

4.3 Discussion and Analysis

According to our experiment results, we can conclude that our algorithm is not so bad to present multiple geo-referenced videos. Figure 4.1, Figure 4.2, and Figure 3.5 show the experiment results, and the scene in the videos and the buildings in the 3D environment is not totally matched. The first reason is our sensor data is not correct enough, and the second reason is that our IFRAME shimmer is fixed. At very beginning, we would like to dynamically change the position of IFRAME shimmer, however, according to the experiments, we have decided to show fixed IFRAME shimmer. Otherwise, if we dynamically change the video position sometimes it will overlap two videos. In addition, this will decrease the efficiency of our system. Besides, to make our system more effective, we estimate the location and heading information for every ten seconds. Nevertheless, the results are amazing to show multiple geo-referenced videos with Google Earth.

There are still some problems in our algorithms. How to deal with more than

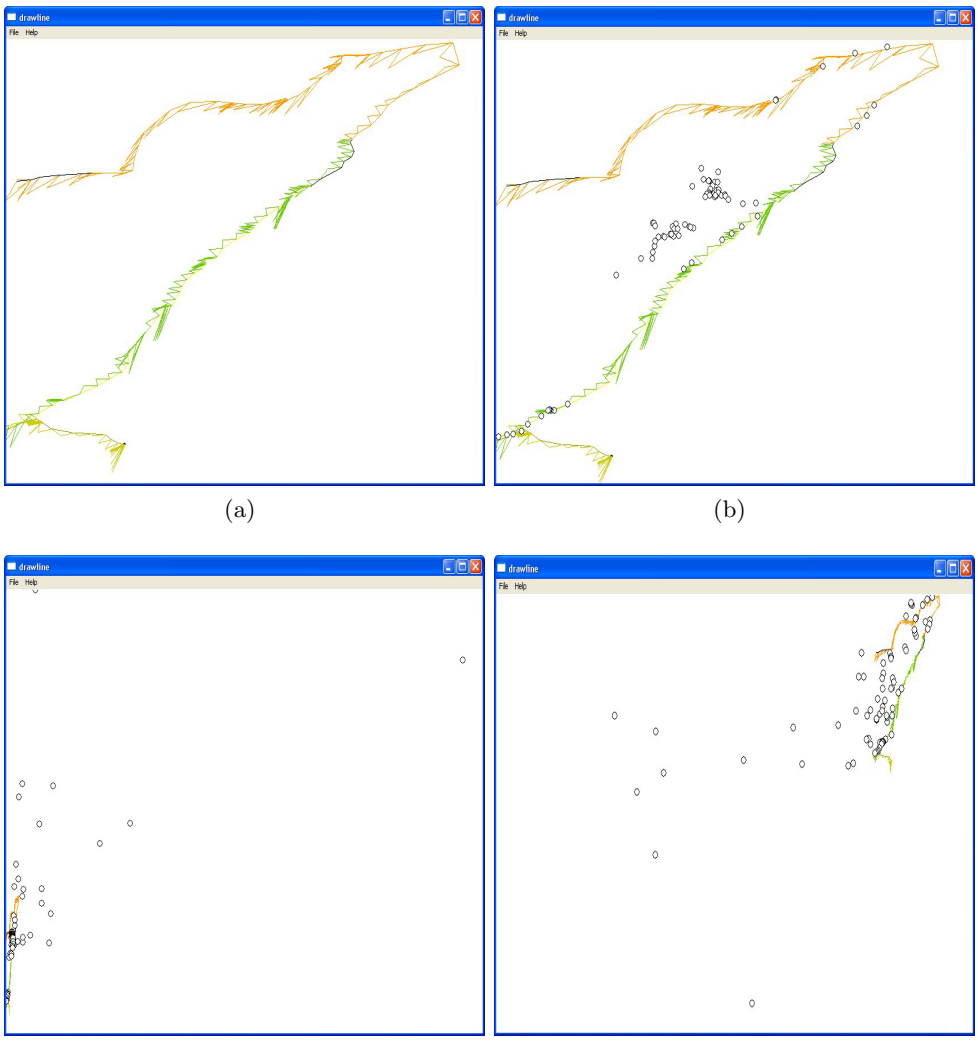


Figure 4.3: The trajectory of three videos for different cases.

four videos, how to match the screen position with the GPS coordinates, and how to indicate the videos who is out of range. Firstly, we have not too many videos in our database, therefore, we just do our experiments with no more than four videos. However, when the videos become more and more, it is necessary to figure out how to deal with more than four videos. This can be achieved by clustering these videos into groups with no more than four videos in one group. Secondly, as in Google Maps, there is an API called "fromLatLngToDivPixel" that can convert the latitude and longitude to screen position [15]. However, in Google Earth, there is no such API, then we just sort the GPS position information and allocate the videos to the corresponding IFRAME shimmer based on the sort results. Finally, we can use different background color or some text to identify the videos which are out of range.

On the whole, our algorithm to compute the viewpoint based on the multiple videos is reasonable and it can be adopted in our system to present multiple geo-referenced videos.

Chapter 5

Challenges and Future Work

5.1 Challenges

Our study presents a novel approach which can automatically locate and display videos in 2D and 3D environments. Some encouraging results are shown in the previous chapter. However, there are still some remaining challenges which need to be addressed in our future work. Below we list a description of five specific issues that we faced in our current research and required further development.

First, the acquired sensor data in our case was not using the same coordinate system as Google Earth or Google Maps. Therefore, the data needs to be converted so that it is compatible with systems such as Google Maps and Google Earth. Our experimental GPS sensor data information is based on a format of degrees, minutes, and seconds. However, the longitude and latitude in Google Earth uses a decimal degree format as represented by the WGS84 coordinate system [13]. The broader issue here is that multiple coordinate systems need to be supported and data needs to be correctly identified and converted to support large-scale video acquisition and applications.

Second, the sensor values in our experiments are essentially noisy sometimes. Hence, the problem of data quality is required further study. For example, a signal interpolation method or an error correction method may be helpful in our work. Moreover, capture video in a good weather condition is also recommended for reaching a high quality. Another issue is the registration accuracy of 3D buildings in Google Earth (or other virtual worlds). Furthermore, the 3D datasets are far from complete and only a few cities have extensive 3D structures in Google Earth. If there is a missed 3D building model in the virtual world, then a visual mismatch will be occurred between video and 3D world in that area. This may ruin a user's navigational experience. However, we assume that in all of our scenarios most 3D buildings are well modeled.

Third, as mentioned earlier, current user interfaces are mostly designed for 2D environment. This makes it difficult for the user to specify a 3D query region using existing interfaces. In our prototype, we use Google Maps to acquire the querying area. In future work, a full 3D query input is necessary in our system. Moreover, playing videos in the 3D scene at the right location is also a challenging work. More specifically, we want to see multiple videos moving with the location identifier. The algorithm of matching the screen position (the iframe shimmer position) with location information (the longitude and latitude information in Google Earth) should be more sophisticated. Otherwise, the videos cannot match the objects in the environment which will provide bad experiences to users.

Fourth, there are many different scenarios for presenting multiple geo-referenced videos. However, not all of these scenarios can be correctly presented due to visual contradiction. There is a tradeoff for each situation. According to user study and application feasibility, some practical rules were settled as described in previous chapter. Additionally, as a future work, the code efficiency and a friendly interface should be considered. For example, if many videos need to be shown simultaneously, then the network bandwidth may be an important issue to be considered. In such case, we may ignore the distant videos and down-sample all playing videos to an appropriate resolution.

Finally, there is a practical challenge of displaying overlapped videos in an environment such as Google Earth. Although some existing image and video editing interfaces were designed for supporting geo-location information, they are still not well-fitted for our work. For example, currently the interface in Google Earth only show *YouTube* videos which are specified by some URL. In our work, we require a more flexible interface which can let user make a selection of IFRAME shim method instead of using Google Earth's API. On the other hand, we use our own media server which can manipulate the source video clips by applying various operations. However, a current limitation is that we cannot perform 3D perspective transformation of the videos. With the technique of IFRAME shim under Mac OS X, we believe we can solve this problem in Google Earth with the latest webKit.

5.2 Future Work

Our study will focus on 3D environment with 3D query method in the future. In present, some follow-up work was under constructing and we list them below: first, the completion and extension of current work; second, a proposed 3D query method to achieve 3D geo-referenced video search; at last, a video quality adjustment technique according to the network condition.

5.2.1 Complete and Extend Previous Work

Though our work can already achieve a good enough performance, we still believe there is room of improvement in the following three areas:

- The meta-data is not very accurate, therefore acquiring more data is very necessary. In addition, the precision of GPS and compass depends on the weather. According to this, we need to acquire data during a clear day. If the data is accurate, then we can figure out whether our code is correct.
- The matching between the position of the presented videos and the position of virtual models in 3D environment is another challenged problem in our work. To achieve this, the screen position of IFRAME Shim with corresponding GPS location information is required to be accurately computed, especially for presenting multiple videos. We have shown some tentative methods in the previous chapter. More sophisticated method was under developing.
- A more functional video presenting system is under designing. We can improve the system to achieve 3D perspective transformation of videos. Apple Inc. has proposed a Cascaded Style Sheets (CSS) 3D transform. CSS is robust enough, but it is not applicable for all the browsers. In addition, this technique is a part of WebKit which allows you to show elements 3D space using CSS [51]. Maybe all the browsers will eventually support this technology, and we can adopt it. Otherwise, we should propose a more efficient algorithm to suit our search engine.

5.2.2 3D Query Method

Our preliminary work is based on 2D search method just like input a 2D query window in Google Maps and present video results with corresponding meta-data on top of Google Earth. This will make user confused because our presentation is in 3D environments. However, the query method is still in 2D. Therefore, we need to bring up a fully 3D query method using our 3D FOV just as Figure 1.1 model (b). Furthermore, we will propose a new algorithm to develop the query, and based on 2D method, we can further extend to 3D.

5.2.3 Adjustment of Video Quality

No matter how many videos shown simultaneously, we need to consider the network's limit. As the growing number of geo-referenced videos, it is very common that multiple videos were retrieved from a single query. Therefore, changing the video quality dynamically according to the network condition is necessary. We also

observed that most users can tolerate low quality videos during multiple video navigating experience. Hence, in the future, we will improve our system by adjusting video quality dynamically based on existing techniques.

Chapter 6

Conclusions

6.1 Summary

We have implemented a web-based video search engine - GRVS to retrieve the georeferenced videos and present these videos in 2D or 3D environments. In addition, we have proposed an algorithm to compute viewpoint for multiple videos. Multiple videos can be presented simultaneously base on our viewpoint computation result. Since the virtual objects in the virtual worlds can be accurately matched to the video scenes, our system can provide pleasable virtual navigation experience to users. In this thesis, Chapter 2 presents a literature survey, and shows some advanced techniques related to our research area. Chapter 3 shows an overview of our system. It provides a detailed description of designation as well as implementation of a georeferenced video search system in 2D and 3D environments. Sequentially, Chapter 4 describes the design and results of the experiments to show the correctness and effectiveness of our algorithm. At the same time, the discussion and analysis of our system are also presented. Next, Chapter 5 shows the future challenges in our research. For each problem, we give a brief explanation for the cause and provide possible solutions. Finally, Chapter 6 draws conclusions and contributions of the thesis.

Using GRVS, users can search for the videos that capture a desired region of interest. Using our novel search technique, highly accurate search results are guaranteed base on our visual scene model. The map-based and earth-based interface enhanced the visual perception and provides the user an intuitive experience of geo-location through video presenting. With 2D search engine, everything works satisfactorily. Our sample website can be accessed at: http://geovid.org/Query.html.

With 3D search engine, we are able to demonstrate automatic placing of videos into three-dimensional coordinate system in Google Earth and the result is very promising. There still remain some challenges to overcome, such as the sensor ac-

curacy of our collected dataset due to weather conditions and other environmental effects. However, most of the data can be fully automatically placed correctly in our experiments. This is crucially important for large-scale datasets processing. Our algorithm of presentation of multiple videos is shown in Chapter 4.

6.2 Contributions

To summarize this thesis, our major contributions were listed in the following:

- Provide a detailed literature review of related work in our research area. Learning from these literatures, we can further extend our system, and make our results more efficient and accurate.
- Describe designation and implementation of 2D/3D geo-referenced video search engines to visually validate our search engine model.
- Propose an algorithm which displays multiple geo-referenced videos in a plausible way.
- Show challenges and future work that we have considered for the follow up work.

Chapter 7

List of Publications

Sakire Arslan Ay, Lingyan Zhang, Seon Ho Kim, He Ma, and Roger Zimmermann. GRVS: A Georeferenced Video Search Engine. In *MM '09: Proceeding of the* 17^{th} *ACM International Conference on Multimedia*, pages 977−978, Beijing, China, 2009.

Lingyan Zhang, Roger Zimmermann, and Guanfeng Wang. Presentation of Geo-Referenced Videos with Google Earth. In 1^{st} *ACM Workshop on Surreal Media and Virtual Cloning* (SMVC), Florence, Italy, 2010. In conjunction with ACM International Conference on Multimedia, 2010.

Bibliography

[1] P. Adrian, M. Pierre-Alain, and K. Ioannis. ThemExplorer: Finding and browsing geo-referenced images. In *CBMI '08: 6th International Workshop on Content-Based Multimedia Indexing*, pages 576 – 583. CBMI, June 2008.

[2] A. Albagul, M. Hrairi, Wahyudi, and M. Hidayathullah. Design and Development of Sensor Based Traffic Light System. *American Journal of Applied Sciences*, March 2006.

[3] S. Arslan Ay, R. Zimmermann, and S. H. Kim. Viewable Scene Modeling for Geospatial Video Search. In *MM '08: 16th ACM International Conference on Multimedia*, pages 309–318, 2008.

[4] S. Arslan Ay, R. Zimmermann, and S. H. Kim. Relevance Ranking in Georeferenced Video Search. *Multimedia Systems Journal*, 16(2), February 2010.

[5] N. Beckmann, H.-P. Kriegel, R. Schneider, and B. Seeger. The R*-Tree: An Efficient and Robust Access Method for Points and Rectangles. In H. Garcia-Molina and H. V. Jagadish, editors, *Proceedings of the 1990 ACM SIGMOD International Conference on Management of Data, Atlantic City, NJ, May 23-25, 1990*, pages 322–331. ACM Press, 1990.

[6] e. C. Brenton Shields. Definition of video rendering, 2009. URL: http://www.ehow.com.

[7] G. Cai. GeoVIBE: A Visual Interface for Geographic Digital Libraries. In *Visual Interfaces to Digital Libraries [JCDL 2002 Workshop]*, pages 171–187, London, UK, 2002. Springer-Verlag.

[8] S. Christmann. Blowing up HTML5 video and mapping it into 3D space, 20 April 2010. URL: http://www.craftymind.com/2010/04/20/blowing-up-html5-video-and-mapping-it-into-3d-space/.

[9] E. Cooke and N. O'Connor. Multiple Image View Synthesis for Free Viewpoint Video Applications, 2005.

[10] B. Epshtein, E. Ofek, Y. Wexler, and P. Zhang. Hierarchical photo organization using geo-relevance. In *GIS '07: Proceedings of the 15th annual ACM international symposium on Advances in geographic information systems*, pages 1–7, New York, NY, USA, 2007. ACM.

[11] FFmpeg Team. FFmpeg Homepage, 15 June 2010. URL: http://ffmpeg.org/.

[12] M. L. Gleicher and F. Liu. Re-cinematography: improving the camera dynamics of casual video. In *MULTIMEDIA '07: Proceedings of the 15th international conference on Multimedia*, pages 27–36, New York, NY, USA, 2007. ACM.

[13] Google Earth Staff. Google Earth User Guide, 2007.

[14] Google Earth Staff. Google Earth API Samples, 6 August 2009. URL: http://earth-api-samples.googlecode.com/svn/trunk/examples/bounds.html.

[15] Google Maps Team. Mashup Mania with Google Maps, January 2009. URL: http://geochalkboard.files.wordpress.com/2009/01/google-maps-pdf-article-v51.pdf.

[16] W. H. A. T. W. Group. HTML5 (including next generation additions still in development), 9 September 2010. URL: http://www.whatwg.org/specs/web-apps/current-work/multipage/video.html.

[17] A. Guttman. R-trees: A Dynamic Index Structure for Spatial Searching. In *INTERNATIONAL CONFERENCE ON MANAGEMENT OF DATA*, pages 47–57. ACM, 1984.

[18] J. T. Heuer and S. Dupke. Towards a Spatial Search Engine Using Geotags. In *GI-Days 2007 - Young Researchers Forum*, pages 199–204, 2007.

[19] Y. Inc. Flickr homepage, 2010. URL:http://www.flickr.com/.

[20] H. Jarvinen. Metadata Management, January 2007.

[21] M. Joint, P.-A. Moellic, P. Hede, and P. Adam. PIRIA: A general tool for indexing, search and retrieval of multimedia content. In *Image processing : algorithms and systems*, pages 116 – 125. SPIE, Bellingham WA, ETATS-UNIS, 2004.

[22] R. Kadobayashi and K. Tanaka. 3D viewpoint-based photo search and information browsing. In *SIGIR '05: Proceedings of the 28th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 621–622, New York, NY, USA, 2005. ACM.

[23] L. S. Kennedy and M. Naaman. Generating diverse and representative image search results for landmarks. In *WWW '08: Proceeding of the 17th international conference on World Wide Web*, pages 297–306, New York, NY, USA, 2008. ACM.

[24] J. L. Kihwan Kim, Sangmin Oh and I. Essa. Augmenting Aerial Earth Maps with Dynamic Information from Videos. In 8^{th} *IEEE International Symposium on Mixed and Augmented Reality, 2009.*, pages 35–38, October 2009.

[25] H. Kim, I. Kitahara, R. Sakamoto, and K. Kogure. An immersive free-viewpoint video system using multiple outer/inner cameras. In *3D Data Processing Visualization and Transmission*, pages 782–789, 2006.

[26] S. H. Kim, S. Arslan Ay, B. Yu, and R. Zimmermann. Vector Model in Support of Versatile Georeferenced Video Search. In 1^{st} *ACM Multimedia Systems Conference*, Scottsdale, Arizona, 22-23 February 2010.

[27] K. Kimura and H. Saito. Player Viewpoint Video Synthesis Using Multiple Cameras. In *Visual Media Production, 2005. CVMP 2005. The 2nd IEE European Conference on*, pages 112–121, 2005.

[28] K. Kimura and H. Saito. Video Synthesis at Tennis Player Viewpoint from Multiple View Videos. In *Proceedings of the 2005 IEEE Conference 2005 on Virtual Reality*, VR '05, pages 281–282, Washington, DC, USA, 2005. IEEE Computer Society.

[29] J. King. How to cover an IE windowed control (Select Box, ActiveX Object, etc.) with a DHTML layer, 21 July 2003. URL: http://www.macridesweb.com/oltest/IframeShim.html.

[30] K. Koiso, T. Matsumoto, and K. Tanaka. Spatial Presentation and Aggregation of Georeferenced Data. In *Proceedings of the Sixth International Conference on Database Systems for Advanced Applications*, DASFAA '99, pages 153–160, Washington, DC, USA, 1999. IEEE Computer Society.

[31] P. Kulkarni, D. Ganesan, P. Shenoy, and Q. Lu. Senseye: a multi-tier camera sensor network. In *MULTIMEDIA '05: Proceedings of the 13th annual ACM international conference on Multimedia*, pages 229–238, New York, NY, USA, 2005. ACM.

[32] E. Lamboray, M. Waschbusch, S. Wurmlin, H. Pfister, and M. Gross. Dynamic Point Cloud Compression for Free Viewpoint Video, 2003.

[33] E. Lamboray, S. Wurmlin, M. Waschbusch, M. Gross, and H. Pfister. Unconstrained free-viewpoint video coding. In *Proceedings of the IEEE International Conference on Image Processing (ICIP) 2004*, pages 24–27, 2004.

[34] R. R. Larson. Geographic Information Retrieval and Spatial Browsing. *GIS and Libraries: Patrons, Maps and Spatial Information*, pages 81–124, April 1996.

[35] O. Lassila and R. R. Swick. Resource Description Framework (RDF) Model and Syntax Specification, 22 February 1999. URL: http://www.w3.org/TR/1999/REC-rdf-syntax-19990222/.

[36] P. Lewis, A. Winstanley, and S. Fotheringham. Position Paper : A conceptual model of Spatial Video moving objects using Viewpoint data structures. In *International Workshop on Moving Objects - from natural to formal language*. Geographical Information Science GIScience, 23 September 2008.

[37] F. Liu and M. Gleicher. Video Retargeting: Automating Pan and Scan. In *MULTIMEDIA '06: Proceedings of the 14th annual ACM international conference on Multimedia*, pages 241–250, New York, NY, USA, 2006. ACM.

[38] X. Liu, M. Corner, and P. Shenoy. SEVA: Sensor-Enhanced Video Annotation. *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMCCAP)*, 5(3):24, 2009.

[39] F. Ltd. Flowplayer - flash video player for the web, 2010. URL: http://flowplayer.org/.

[40] B. Martins and P. Calado. Learning to rank for geographic information retrieval. In *GIR '10: Proceedings of the 6th Workshop on Geographic Information Retrieval*, pages 1–8, New York, NY, USA, 2010. ACM.

[41] M. Naaman, Y. J. Song, A. Paepcke, and H. Garcia-Molina. Automatic Organization for Digital Photographs with Geographic Coordinates. In 4^{th} *ACM/IEEE-CS Joint Conference on Digital Libraries*, pages 53–62, 2004.

[42] T. Navarrete and J. Blat. A Semantic Approach for the Indexing and Retrieval of Geo-referenced Video. In 1^{st} *International Workshop on Semantic-Enhanced Multimedia Presentation Systems (SEMPS)*, 6 December 2006.

[43] T. Navarrete, J. Blat, and D. de Tecnologia. VideoGIS: Segmenting and indexing video based on geographic information. In *5th AGILE Conference on Geographic Information Science*, pages 1–9. Citeseer, 2002.

[44] Q. M. Nguyen, T. N. Kim, D. H.-L. Goh, Y.-L. Theng, E.-P. Lim, A. Sun, C. H. Chang, and K. Chatterjea. TagNSearch: Searching and Navigating Geo-referenced Collections of Photographs. In *ECDL '08: Proceedings of the 12th European conference on Research and Advanced Technology for Digital Libraries*, pages 62–73, Berlin, Heidelberg, 2008. Springer-Verlag.

[45] P. Ni, F. Gaarder, C. Griwodz, and P. Halvorsen. Video Streaming into Virtual Worlds: the Effects of Virtual Screen Distance and Angle on Perceived Quality. In *MM '09: 17^{th} ACM International Conference on Multimedia*, pages 885–888, 2009.

[46] V. Nozick and H. Saito. Real-Time Free Viewpoint from Multiple Moving Cameras. In *Proceedings of the 9th international conference on Advanced concepts for intelligent vision systems*, ACIVS'07, pages 72–83, Berlin, Heidelberg, 2007. Springer-Verlag.

[47] V. Nozick and H. Saito. On-line Free-viewpoint Video: From Single to Multiple View Rendering. In *International Journal of Automation and Computing 5*, pages 257–265, 2008.

[48] J. A. Orenstein. Spatial query processing in an object-oriented database system. In *SIGMOD '86: Proceedings of the 1986 ACM SIGMOD international conference on Management of data*, pages 326–336, New York, NY, USA, 1986. ACM.

[49] A. Pigeau and M. Gelgon. Building and Tracking Hierarchical Geographical & Temporal Partitions for Image Collection Management on Mobile Devices. In 13^{th} *ACM Intl. Conference on Multimedia*, 2005.

[50] C. Reed and Google Earth Staff. KML 2.1 Reference - An OGC Best Practice, 2 May 2007.

[51] C. Reed and Google Earth Staff. CSS 3D Transforms Module Level 3, 20 March 2009. URL: http://www.w3.org/TR/css3-3d-transforms.

[52] K. Rodden and K. R. Wood. How do People Manage their Digital Photographs? In *SIGCHI Conference on Human Factors in Computing Systems*, pages 409–416, 2003.

[53] F. Schnell. GPicSync: Automatically Geocode Pictures from your Camera and a GPS Track Log, 13 April 2009. URL: http://code.google.com/p/gpicsync/.

[54] I. O. Sebe, J. Hu, S. You, and U. Neumann. 3D video surveillance with Augmented Virtual Environments. In *IWVS '03: First ACM SIGMM international workshop on Video surveillance*, pages 107–112, New York, NY, USA, 2003. ACM.

[55] S. Shi, W. J. Jeon, K. Nahrstedt, and R. H. Campbell. Real-time remote rendering of 3D video for mobile devices. In *MM '09: Proceedings of the seventeen ACM international conference on Multimedia*, pages 391–400, New York, NY, USA, 2009. ACM.

[56] S. Shi, K. Nahrstedt, and R. H. Campbell. View-dependent real-time 3d video compression for mobile devices. In *MM '08: Proceeding of the 16th ACM international conference on Multimedia*, pages 781–784, New York, NY, USA, 2008. ACM.

[57] H.-Y. Shum, S. B. Kang, and S.-C. Chan. Survey of image-based representations and compression techniques. *Circuits and Systems for Video Technology, IEEE Transactions on*, 13(11):1020–1037, 2003.

[58] I. Simon and S. M. Seitz. Scene Segmentation Using the Wisdom of Crowds. In *ECCV '08: Proceedings of the 10th European Conference on Computer Vision*, pages 541–553, Berlin, Heidelberg, 2008. Springer-Verlag.

[59] R. Simon and P. Fröhlich. A Mobile Application Framework for the Geospatial Web. In *16^{th} International Conference on World Wide Web (WWW)*, pages 381–390, New York, NY, USA, 2007. ACM.

[60] A. Smolic, K. Mueller, P. Merkle, C. Fehn, P. Kauff, P. Eisert, and T. Wiegand. 3D Video and Free Viewpoint Video - Technologies, Applications and MPEG Standards. *Multimedia and Expo, IEEE International Conference on*, 0:2161–2164, 2006.

[61] A. Smolic, K. Mueller, P. Merkle, T. Rein, M. Kautzner, P. Eisert, and T. Wieg. Free Viewpoint Video Extraction, Representation, Coding and Rendering. In *Proc. IEEE International Conference on Image Processing*, 2004.

[62] N. Snavely, S. M. Seitz, and R. Szeliski. Photo tourism: exploring photo collections in 3D. In *SIGGRAPH '06: ACM SIGGRAPH 2006 Papers*, pages 835–846, New York, NY, USA, 2006. ACM.

[63] J. Starck and A. Hilton. Free-Viewpoint Video for Interactive Character Animation. *Proc. 4th. Symposium on "Intelligent Media Integration for Social Information Infrastructure, Nagoya JAPAN.*, pages 25–30, 2006.

[64] Y. Theodoridis, M. Vazirgiannis, and T. Sellis. Spatio-Temporal Indexing for Large Multimedia Applications. In *icccn*, page 0441. Published by the IEEE Computer Society, 1996.

[65] C. Torniai, S. Battle, and S. Cayzer. Sharing, Discovering and Browsing Photo Collections through RDF geo-metadata. In G. Tummarello, P. Bouquet, and O. Signore, editors, *SWAP*, volume 201 of *CEUR Workshop Proceedings*. CEUR-WS.org, 2006.

[66] K. Toyama, R. Logan, and A. Roseway. Geographic Location Tags on Digital Images. In 11^{th} *ACM Intl. Conference on Multimedia*, pages 156–166, 2003.

[67] Wikipedia. Wiki contents: Ajax, HTML5 video, Field of View, Angle of View, Geographic information system, LIDAR, Mixed Reality, Image-based modeling and rendering and Multiview Video Coding, 2010. URL: http://en.wikipedia.org/wiki/.

[68] A. G. Woodruff and C. Plaunt. GIPSY: Georeferenced Information Processing SYstem. *JASIS*, 45(9):645–655, 1994.

[69] Wowza Media Systems. Wowza Media Systems Homepage, September 2010. URL: http://www.wowzamedia.com/.

[70] S.-U. Yoon, E.-K. Lee, S.-Y. Kim, Y.-S. Ho, K. Yun, S. Cho, and N. Hur. Coding of layered depth images representing multiple viewpoint video. In *Picture Coding Symposium (PCS)*, pages 1–6, 2006.

[71] S.-U. Yoon, E.-K. Lee, S.-Y. Kim, Y.-S. Ho, K. Yun, S. Cho, and N. Hur. Inter-camera Coding of Multi-view Video Using Layered Depth Image Representation. In *Advances in Multimedia Information Processing - PCM 2006, 7th Pacific Rim Conference on Multimedia, Hangzhou, China, November 2-4, 2006, Proceedings*, pages 432–441. Springer, 2006.

[72] L. Zhang, R. Zimmermann, and G. Wang. Presentation of Geo-Referenced Videos with Google Earth. In *MM '10: Proceeding of the 18th ACM International Conference on Multimedia*, New York, NY, USA, 2010. ACM.

[73] Z. Zhang, L. Huo, C. Xia, W. Zeng, and W. Gao. A Virtual View Generation Method for Free-Viewpoint Video System. In *Int. Symp. on Intelligent Signal Processing and Communication Systems (ISPACS 2007)*, pages 361–364, Xiamen, China, 2007.