GRVS: A Georeferenced Video Search Engine

Sakire Arslan Ay†, Lingyan Zhang‡, Seon Ho Kim§, Ma He‡, Roger Zimmermann‡
†Computer Science Department, University of Southern California, Los Angeles, CA 90089
‡School of Computing, National University of Singapore, Singapore 117417
§Department of CS & IT, University of the District of Columbia, Washington, DC 20008
arslan@usc.edu, skim@udc.edu, {lingyan,mahe,rogerz}@comp.nus.edu.sg

ABSTRACT
An increasing number of recorded videos are being tagged with geographic properties of the camera scenes. This metadata is of significant use for storing, indexing and searching large collections of videos. By considering video related meta-information, more relevant and precisely delimited search results can be returned. Our system implementation demonstrates a prototype of a georeferenced video search engine (GRVS) that utilizes an estimation model of a camera’s viewable scene for efficient video search. For video acquisition, our system provides an automated annotation software that captures videos and their respective field of views (FOV). The acquisition software allows community-driven data contributions to the search engine.

Categories and Subject Descriptors
H.2.4 [Database Management]: Systems—Query processing; H.2.4 [Database Management]: Systems—Multimedia databases

General Terms
Algorithms, Measurement, Performance

Keywords
Video search, georeferencing, meta-data fusion, GPS

1. INTRODUCTION
Sensors attached to cameras, such as GPS and digital compass devices, allow users to collect geographic properties of camera scenes simultaneously while video is recorded. The captured geographic meta-data have significant potential to aid in the process of indexing and searching of georeferenced video data, especially in location aware video applications.

Our 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. Based on the proposed model, we constructed a camera prototype to capture the relevant meta-data (Figure 1), implemented a database with a real-world video data set captured using our prototype camera, and developed a web-based search system to demonstrate the feasibility and applicability of our concept of georeferenced video search.

Figure 1: Setup for data collection: laptop computer; OceanServer OS5000-US compass; Canon VIXIA HV30 camera; Pharos iGPS-500 receiver.

2. SEARCH ENGINE
GRVS is a web-based video search engine that allows georeferenced videos to be searched by specifying geographic regions of interest. 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 our current implementation the query can be a rectangle or a trajectory. Note that GRVS is quite different from Google Street View, which renders panoramic – but static – still images referenced to map locations. Furthermore, the data for GRVS is provided by the community with the use of low-cost capture equipment.

The search engine is comprised of three main components: (1) a database that stores the collected meta-data, (2) a media server that manages the videos, and (3) a web-based interface that allows the user to specify a query input region and then provides a display of the query results. We implemented the engine using the following open source software: the LAMP stack (i.e., Linux, Apache, MySQL, PHP), the Wowza Media Server, and the Flowplayer [1].

2.1 Meta-data Collection
To collect georeferenced video data we implemented a light-weight acquisition software that can concurrently acquire video, GPS and compass sensor signals while running on a laptop computer. Based on MS Windows and its DirectShow filter architecture, different video formats are potentially supported. In our experimental system (shown in Figure 1), a Canon VIXIA HV30 camera is used to acquire MPEG-2 encoded high-definition video via a FireWire
Table 1: Schema for FOV representation.

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Filename</td>
<td>Uploaded video file</td>
</tr>
<tr>
<td>FOVId</td>
<td>ID of the FOV (field-of-view)</td>
</tr>
<tr>
<td>&lt;&lt;Lat, Long&gt;</td>
<td>Coordinate for camera location (read from GPS)</td>
</tr>
<tr>
<td>theta</td>
<td>Camera view direction (read from compass)</td>
</tr>
<tr>
<td>R</td>
<td>Viewable distance</td>
</tr>
<tr>
<td>alpha</td>
<td>Angular extent for camera field-of-view</td>
</tr>
<tr>
<td>time</td>
<td>Local time for the FOV</td>
</tr>
<tr>
<td>timecode</td>
<td>Timecode for the FOV in the video (extracted from video)</td>
</tr>
</tbody>
</table>

(IIEEE 1394) connection. To obtain the orientation of the camera, we employ the OS5000-US solid state tilt compensated 3-axis digital compass and the Pharos iGPS-500 GPS receiver is used to acquire the camera location. The acquisition software records the georeferences 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.

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 34 video clips, ranging from 3 to 21 minutes in duration. At one second intervals an FOV was collected, resulting in 17,982 FOVs in total.

2.2 Database Implementation

When the user uploads videos into the GRVS system, video meta-data is processed automatically and viewable scene information is stored in a MySQL database. Each individual FOV is represented as a tuple based on the schema given in Table 1.

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].

2.3 Web User Interface

A map-based query interface allows users to draw the query region visually. The result of a query contains a list of the overlapping video segments. For each returned video segment, we display the corresponding FOVs on the map. To reduce clutter we draw the FOVs every 2 seconds. The user can browse through the resulting video segments and interactively play videos. Note that the video server streams precisely 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 2.

![Figure 2: Georeferenced Video Search Engine Web-Interface.](http://eiger.ddns.comp.nus.edu.sg/geo/Query.html)

We implemented the web interface using JavaScript and the Google Maps API [2]. Ajax techniques were used to send the query window to the MySQL database and to retrieve the query results. With Ajax, web applications can obtain data from the server asynchronously in the background without interfering with the display. The communication with the MySQL database and the UDFs was provided via PHP. For video playback we used the Flowplayer [1], an open source flash media player. The video files were transcoded into H.264 format. Note that with these technologies our search engine implementation is platform independent. We have successfully deployed the GRVS system on both Linux and Windows servers.

3. CONCLUSIONS

We have implemented a web-based video search engine – GRVS – to query a database of georeferenced videos. Using GRVS, users can search for the videos that capture a particular region of interest. Our novel search technique, which is based on our viewable scene model [3], ensures highly accurate search results. The map-based interface enhanced with visual features provides the user with a clear understanding of the geo-location seen in the video. We will demonstrate our system using real videos captured in Singapore and displayed on top of Google Maps. We will further show example queries and explain the features of the search engine. We also plan to demonstrate our camera prototype and illustrate how to capture georeferenced videos. The demonstration system is available online at http://eiger.ddns.comp.nus.edu.sg/geo/Query.html.

4. REFERENCES