Relevance Ranking in Geospatial Video Search

Sakire Arslan Ay
Department of Computer Science
University of Southern California
Los Angeles, CA

Roger Zimmermann
School of Computing
National University of Singapore
Singapore

Seon Ho Kim
Department of Computer Science & Information Technology
University of the District of Columbia
Washington, DC 20008, USA

February 23, 2009

Abstract

Video sensors are becoming ubiquitous and the volume of captured video material is very large. Therefore, tools for searching video databases are indispensable. Current techniques that extract features purely based on the visual signals of a video are struggling to achieve good results. By considering video related meta-information, more relevant and precisely delimited search results can be obtained. In this study we propose a novel approach for querying videos based on the notion that the geographical location of the captured scene in addition to the location of a camera can provide valuable information and may be used as a search criterion in many applications. This study provides an estimation model of the viewable area of a scene for indexing and searching and reports on a prototype implementation. Among our objectives is to stimulate a discussion of these topics in the research community as information fusion of different georeferenced data sources is becoming increasingly important. Initial results illustrate the feasibility of the proposed approach.

1 Introduction

Due to technological advances, an increasing number of video clips are being collected from various devices and stored for a variety of purposes such as surveillance, monitoring, reporting, or entertainment. These acquired video clips contain a tremendous amount of visual and contextual information that makes them unlike any other media type. However, even now, there are no effective ways to index and search video data at the high semantic level preferred by humans. Text annotations of video can be utilized for search, but high-level concepts must often be added by hand and hence this manual task is laborious and cumbersome for large video collections. Content based video retrieval is in its infancy, very challenging and still not always satisfactory.

Some types of video data are naturally tied to geographical locations. For example, video data from traffic monitoring may not have any meaning without its associated position information. Thus, in such applications, one needs a specific location to retrieve the traffic video at that point or in that region. Hence, combining video data with its location coordinates can provide an effective way to index and search videos, especially when a database handles an extensive amount of video data. Note that location information can be collected by various small devices attached to a camera, such as a global positioning system (GPS) sensor (see Figure 1). A preliminary example of this

Figure 1: Experimental hardware and software to acquire georeferenced video.

type of work is Google Earth, which implements such a concept with panoramic images, but only from a high elevation (sky view). A user can find a top view of a point or region given a query point. The current implementation is innovative but has some limitations.

We believe that *georeferenced video search* will play a prominent role in many future applications. However, there are still many open, fundamental research questions in this field. Most videos captured are not panoramic and as a result the *viewing direction* becomes very important. GPS data only identifies object locations and therefore it is imperative to investigate the natural concepts of a viewing direction and a view point. For example, we may be interested to view a building only from a specific angle. The question arises whether a video database search can accommodate such human friendly views. The collection and fusion of multiple sensor streams such as the camera location, field-of-view, direction, etc., can provide a comprehensive model of the *viewable scene*. The objective then is to index the video data based on the human viewable space and therefore to enable the retrieval of more meaningful and recognizable scene results for user queries. Cameras may also be mobile and thus the concept of a camera location is extended to a trajectory. Consequently, finding relevant video segments becomes very challenging. In this study we propose a general methodology to address these and related issues.

One example application that would benefit from our georeferenced video search framework is *geospatial decision making*. The recent rapid increase in the amount of geospatial data available has motivated efforts to integrate multiple geospatial data sets for the purpose of extracting useful information and assisting decision makers. Event extraction – while difficult in the visual signal domain – shows promising results from geospatial information integration and data fusion. Our study provides the following contributions.

- **Automatic annotation of video clips with the camera viewing direction.** While the concept of meta-data annotation has been investigated before, we believe our method is the first to consider the viewing direction.
- **Modeling of the viewable scene.** We propose a viewable scene model that strikes a balance between the complexity of its analytical description and the efficiency with which it can be used for fast searches.
- **Prototype feasibility study.** We have implemented a prototype to demonstrate the feasibility of acquiring, storing, searching, and retrieving video based on our approach.
- **Demonstration of benefits.** From our implementation results we are able to illustrate the benefits of our approach in retrieving the most relevant video segments for a given query.

Before elaborating on our approach in detail, Section **??** contains a brief discussion and survey of related work. We describe the proposed approach for georeferenced video search in Section **??**.

This is followed by a presentation of results based on a real-world data set in Section **??**. Finally, Section **??** discusses open issues and future research directions.

2 Related Work

Associating GPS coordinates with digital media (images and videos) has become an active area of research [20]. In this section, we review the existing work related to search techniques in georeferenced media retrieval and ranking. We will start our survey with methods that specifically consider still images and then move on to videos. We will also briefly describe some prior work in the area of indexing and storage. Lastly, we will mention a few commercial GPS-enabled cameras that produce georeferenced images.

Techniques for Images. There has been significant research on organizing and browsing personal photos according to location and time. Toyama et al. [25] introduced a meta-data powered image search and built a database, also known as World Wide Media eXchange (WWMX), which indexes photographs using location coordinates (latitude/longitude) and time. A number of additional techniques in this direction have been proposed [18, 19]. There are also several commercial web sites [2, 3, 4] that allow the upload and navigation of geo-refenced photos. All these techniques use only the camera geo-coordinates as the reference location in describing images. We instead rely on the field-of-view of the camera to describe the scene. More related to our work, Ephstein et al. [6] proposed to relate images with their view frustum (viewable scene) and used a scene-centric ranking to generate a hierarchical organization of images. Several additional methods are proposed for organizing [21, 13] and browsing [7, 24] images based on camera location, direction and additional meta-data. Although these research work is similar to ours in using the camera field-of-view to describe the viewable scene, their main contribution is on image browsing and grouping of similar images together. [24, 14] use location and other metadata, as well as tags associated with images, and the images visual features to generate representative images within image clusters. Geo-location is often used as a filtering step. [6, 21] solely use location and orientation of camera in retrieving the "typical views" of important object. However their contribution is on segmentation of image scenes and organizing photos based on the image scene similarity. Our work describes a more broad scenario that considers mobile cameras capturing geo-tagged videos and the associated view frustum, which is dynamically changing over time. And our ranking technique do not target any specific application domain, therefore can easily be applied to any specific application.

Techniques for Video. There exist only a few systems that associate videos with their corresponding geo-location. Hwang et al. and Kim et al. propose a mapping between the 3D world and the videos by linking the objects to the video frames in which they appear [11, 15]. However, their work neglects to provide any details on how to use camera location and direction to build links between video frames and world objects. More closely related to our work, Liu et al. [17] presented a sensor enhanced video annotation system (referred to as SEVA) which enables searching videos for the appearance of particular objects. SEVA serves as a good example to show how a sensor rich, controlled environment can support interesting applications, however it does not propose a broadly applicable approach to geo-spatially annotate videos for effective video search. All three studies mentioned above present ideas about how to search georeferenced video collections but do not provide any solutions for analyzing the relevance of search results. To our knowledge, our technique is the first in addressing video ranking based on the "viewable scene" cues. We believe that our approach, when enhanced with an efficient spatio-temporal storage and indexing mechanism, will serve as a general purpose and flexible video search and ranking mechanism that is applicable to any types of video with associated location and direction tags. Consequently it can

be the basis for a tremendous number of multimedia applications.

Beyond georeferenced video ranking, the topic of content based video retrieval and ranking has been studied extensively. The TREC Video Retrieval Evaluation (TRECVID) [22] benchmarking activity has been promoting progress in content-based retrieval of digital video since 2001. Each year, various feature detection methods from dozens of research groups are tested on hundreds of hours of video [23]. Unlike the research activities within the TRECVID benchmark, our focus is solely on high-level descriptions of videos using georeferenced meta-data rather than visual features.

Geospatial Search and Ranking Methods. Although ranking videos based on geospatial properties has not been well studied, there have been several ranking techniques developed for the Geographic Information Retrieval (GIR) systems. Most of these studies compute spatial similarity measures based on the overlap between query region and spatial description of documents using the associated meta-data. Some earlier work [5] studied the basic spatial and temporal relevance calculation methods. More recently, [16], provided a comprehensive summary of geospatial ranking techniques and [8] proposed a global ranking algorithm based on spatial, temporal and thematic parameters. To quantify the relevance of a videos viewable scene to a given query we applied some of the fundamental spatial ranking techniques described in [5] and [8] in our work. Although similar ranking schemes studied before, our work is novel in applying these techniques to rank video data based on viewable scene descriptions.

Indexing and Storage. In our work we propose to use a histogram to accumulate the relevance scores for the camera viewable scenes. Data summarization using histograms is a well-studied research problem in the database community. A comprehensive survey of histogram creation techniques can be found in [12]. In [6] authors use a grid of voting cells to discover the important parts of an image. Their technique use only the spatial attributes to discover the relevant segments of the image scene whereas our ranking methods incorporates both spatial and temporal attributes in calculating relevance.

Commercial products. There exist several GPS-enabled digital cameras which can save the location information with the digital image file as a picture is taken (e.g., Sony GPS-CS1, Ricoh 500SE, Jobo Photo GPS). Very recent models additionally record the current heading (e.g., Ricoh SE-3, Solmeta DP-GPS N2). All current cameras support geotagging for still images only. We believe that, as the use of these cameras increases, more location and direction tagged videos will be produced and there will be a strong need to perform efficient and effective search on those video data.

3 Georeferenced Video Search (or Searching Georeferenced Videos)

In this study our focus is on describing the video content based on the geospatial properties of the region it covers, so that large video collections can be indexed and searched effectively. We refer to this space as the the viewable space of the video scene. In this section, we describe how to quantify, store and query the viewable scene of captured videos. And in Section-4 we introduce several methods to discover the most relevant videos based on the video scene's similarity to the user query.

We model the viewable space of a scene with parameters such as the camera location, the angle of the view, and the camera direction. The camera's viewable scene changes when the camera moves or rotates. This dynamic scene information has to be acquired from sensor-equipped cameras, stored within an appropriate catalog or schema and indexed for efficient querying and retrieval. Our proposed approach consists of four components: 1) modeling of the viewable scene, 2) data acquisition, 3) indexing and querying, and 4) ranking search results. We will now describe first

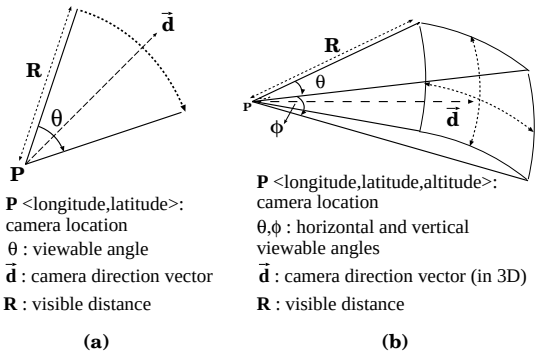


Figure 2: Illustration of *FOVScene* model (a) in 2D (b) in 3D.

three components in turn. Ranking retrieved videos will be discussed extensively in Section-4.

3.1 Georeferenced Video Annotations through Viewable Scene Modeling

We define the scene that a camera captures as the *viewable scene* of the camera. In the field of computer graphics, this area is referred to as *camera field-of-view* (FOV for short). We will use the terms 'viewable scene' and 'field of view (FOV)' interchangeably throughout this manuscript. We describe a camera's viewable scene in 2D space by using the following four parameters, i.e.,

$$FOVScene(P, \vec{d}, \theta, R) \tag{1}$$

where, (1) the camera position P is the $\langle$latitude, longitude$\rangle$ coordinates read from a positioning device (e.g., GPS). (2) the camera direction vector $\overrightarrow{d}$ is obtained based on the orientation angle provided by a digital compass. (3) the camera viewable angle θ describes the angular extent of the scene imaged by the camera [9]. (4) the far visible distance R is the maximum distance at which a large object within the camera's field-of-view can be recognized. The full field-of-view is obtained with the maximum visual angle, which depends on the lens/image sensor combination used in the camera [10]. Smaller image sensors have smaller field-of-view than larger image sensors (when used with the same lens). Alternatively, shorter focal-length lenses have a larger field-of-view than longer focal-length lenses (when used with the same image sensor). The viewable angle θ can be obtained via the following formula (Eqn. 2) [10]:

$$\theta = 2tan^{-1}\frac{y}{2f} \tag{2}$$

where y is the size of the image sensor and f is the focal length of the lens. The relationship between the visible distance R and the viewable angle θ is given in Eqn. 3 [10]. As the camera view is zoomed in or out, the θ and R values will be adjusted accordingly.

$$\theta = 2\arctan\left(\frac{y(R\cos(\frac{\theta}{2}) - f)}{2fR\cos(\frac{\theta}{2})}\right) \tag{3}$$

Our approach in this paper utilizes the set of georeferenced video content captured from a sensor-equipped camera (see Figure 1), which can accurately estimate its current location P, orientation $\vec{d}$ and visual angle θ. 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 2a. Figure 2b 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 directed upwards or downwards). We believe that the extension to 3D is straightforward, especially since we already acquire the altitude level from the GPS device and the pitch and roll values from the compass. For our initial setup we will represent *FOVScene* in 2D space only. We plan to work on the 3D extension in our future work.

3.2 Georeferenced Meta-data Acquisition

A camera's viewable scene changes as it moves or changes its orientation in geo-space. In order to keep track of what the camera sees over time, we need to record the $FOVScene$ descriptions with a certain frequency and produce time stamped meta-data together with time stamped video streams. Our meta-data streams are analogous to sequences of $\langle P, \vec{d}, \theta, R, t \rangle$ quintuples, where t is the time instant at which $FOVScene$ information is recorded. Ideally each camera will store the $FOVScene$ coverage for each individual video frame. However, in large scale applications there may be thousands of moving cameras with different sensing capabilities. We do not make any assumptions about how frequently a camera should record its $FOVScene$ coverage.

Recording georeferenced video streams. Our sensor rich video recording system incorporates three devices: a video camera, a 3D digital compass, and a Global Positioning System (GPS) device. We assume that the optical properties of the camera are known. The digital compass, mounted on the camera, periodically reports the direction in which the camera is pointing. The camera location is read from the GPS device as a ⟨latitude, longitude⟩ pair. Video can be captured with various camera models – we use a high-resolution (HD) camera. Our custom-written recording software receives direction and location updates from the GPS and compass devices as soon as new values are available and records the updates along with the current computer time and coordinated universal (UTC) time. Video data is received from the camera as data packet blocks. Each video data packet is processed in real time to extract frame timecodes and these extracted timecodes are recorded along with the local computer time when the frame was received. Creating a frame level time index for the video stream minimizes the synchronization errors that might occur due to clock skew between the camera clock and the computer clock. In addition, such a temporal video index, whose timing is compatible with other datasets, enables easy and accurate integration with the GPS and compass data.

Calculating viewable angle (θ) and visible distance (R). Assuming that the optical focal length f and the size of the camera image sensor y are known, the camera viewable angle θ can be calculated through Eqn. 2. The default focal length for the camera lens is obtained from the camera specifications. However, when there is a change in the camera zoom level, the focal length f and consequently the viewable angle θ will change. To capture the change in θ, the camera should be equipped with a special unit that will measure the focal length for different zoom levels. Such functionality is not commonly available in today's off-the-shelf digital cameras and camcorders. To simulate the changes in the viewable angle, we have manually recorded the exact video timecodes along with the change in the zoom level. Using the Camera Calibration Toolbox [1] we have measured the f value for five different zoom levels (from the minimal to the maximal zoom level). For all other zoom levels, the focal length f is estimated through interpolation.

The visible distance R can be obtained based on the equation,

$$R = \frac{fh}{y} \tag{4}$$

where f is the lens focal length, y is the image sensor height and h is the height of the target object that will be fully captured within a frame. With regard to the visibility of an object from the current camera position, the size of the object also affects the maximum camera-to-object viewing distance. For large objects (e.g., mountains, high buildings) the visibility distance will be large whereas for small objects of interest the visibility distance will be small. For simplicity in our initial setup we assume R to be the maximum visible distance for a fairly large object. As an example, consider the buildings A and B shown in Fig.-3(a). Both buildings are approximately 8.5m-tall and both are located within the viewable angle of the camera. The distances from the buildings A and B to the camera location are measured as 150m and 300m respectively. The frame

snapshot for the *FOVScene* in Fig.-3(a) is shown in Fig.-3(b). We assume that, with good lighting conditions and no obstructions, an object can be considered visible within a captured frame if it occupies at least 5% of the full image height. For our JVC JY-HD10U camera, focal length is $f = 5.2$mm and the CCD image sensor height is $y = 3.6$mm. Therefore using Eqn.-4 the height of building A is calculated as 8% of the video frame, therefore is considered visible. (Fig.-3) However building B is not visible since it covers only 4% of image frame. Based on the above discussion, the threshold for the far visible distance R for our visible scene model is estimated around 250m. We currently target a mid-range far visible distance of 200-300m. We believe that this range best fits with typical applications that would most benefit from our georeferenced video search (e.g., traffic monitoring, surveillance). Close-up and far-distance will be considered as a part of our future research.

(a) (b)

Figure 3:

Timing and Synchronization. The meta-data entries for compass updates and video frame timecodes have millisecond-granular timing. However GPS location updates are available every second. In order to calculate the camera *FOVscenes*, all three meta-data streams need to be combined and stored as a single stream with an associated common time index. In a sensor-rich system with several attached devices, one challenge is how to synchronize the sensor data read from the attached devices which have different data output rates. Our recording software creates separate data streams for each device, where each meta-data entry is timestamped with the time when the update was received from the device. Later these data streams are combined with a 2-pass algorithm. Such an algorithm processes data in a sliding time window centered at the current time. It will always match the data entries that have the closest timestamps (past or future). In our setup the meta-data output rate for GPS, compass and video are is 1, 40, and 30 samples/sec respectively. Therefore, we match each GPS entry with the temporally closest video frame timecode and compass direction.

For each meta-data entry, in addition to the local time we record the satellite time (in UTC)

that is received along with the GPS location update. The use of the recorded satellite time can be twofold: (1) it enables synchronizing the current computer time with the satellite time (2) it may be used as the time base when executing temporal queries, i.e., by applying the temporal condition of the query to the satellite time. Timestamping the *FOVScene* entries with the satellite time ensures a global temporal consistency among all georeferenced video collections.

Measurement Errors. The accuracy of the *FOVScene* calculation is somewhat dependent on the precision of the location and heading measurements obtained from GPS and compass devices. A typical GPS device is accurate approximately within 10 meters. In our proposed viewable scene model, the area of the region that a typical HD camera captures (*FOVScene*) is on order of tens of thousands of square meters (e.g., at full zoom-out approx. 33,000m^2). Therefore, a difference of 10m is not very significant compared to the size of the viewable scene we consider. Additionally, missing GPS locations – due to various reasons such as a tunnel traversal – can be recovered through estimation such as interpolation. There exists extensive prior work on estimating moving object trajectories in the presence of missing GPS locations. An error in the compass heading may be more significant. Many digital compasses ensure azimuth accuracy of better than 1° (e.g., about 0.6° for the OS5000 digital compass in our system), which will have a minor effect on the viewable scene calculation. However, when mounted on real platforms the accuracy of a digital compass might be affected by local magnetic fields or materials. For our experiments the compass was calibrated within the setup environment to minimize any distortion in compass heading. It is also worth mentioning that multimedia applications often tolerate some minor errors. When a small object is at the edge of viewable scene but is not included in the modeled area, it might not be recognized by a human observer.

3.3 Querying Georeferenced Videos

The next task after collecting georeferenced meta-data is to semantically describe them so that accurate and efficient analysis on the camera viewable scenes is possible. An intuitive way is to store a separate *FOVScene* quintuple including the camera id, video id, frame timecode, camera location, visual angle and camera heading for each video frame. The *FOVScene* coverage of a moving camera over time is analogous to a moving region in the geo-spatial domain, therefore traditional spatio-temporal query types, such as range queries, k nearest neighbor (kNN) queries or spatial joins, can be applied to the *FOVScene* data. In our initial work, we limit our discussion to range queries. The typical task we would like to accomplish is to extract the video segments that capture a given area of interest. As explained in Section 3.2, we can construct the *FOVScene(t,P,*$\vec{d}$*,*θ*,R)* description for every second. Hence, for a given area of interest Q, we can extract the sequence of video frames whose viewable scene overlap with Q. Going from most specific to most general, the query region Q can be a point, a line (e.g., a road), a poly-line (e.g., a trajectory between two points), a circular area (e.g., neighborhood of a point of interest), a rectangular area (e.g., the space delimited with roads) or a polygon area (e.g., the space delimited by certain buildings, roads and other structures). Details of range query processing can be found in our prior work[]. One problem with such a representation on top of a relational model is the computational overhead. In a typical query all frames that belong to the query time interval has to be checked for overlaps. Computational efficiency can be improved by adopting an index structure to store and query *FOVScene* descriptions.

In this study we restrict our example queries to simple spatiotemporal range searches. However, using the camera view direction ($\vec{d}$) in addition to the camera location (P) to describe the camera viewable scene provides a rich information base for answering more complex geospatial queries. For example, if the query asks for the views of an area from a particular angle, more meaningful scene results can be returned to the user. Alternatively, the query result set can be presented

Term	Description
V_k	a video clip k
V_k^F	a video clip k represented by a set of $FOVScenes$
$V_k^F(t_i)$	a polygon shape $FOVScene$ at time t_i, a set of corner points
Q	a polygon query region represented by a set of corner points
$O(V_k^F(t_i), Q)$	overlap region between V_k^F and Q at t_i, a set of corner points
R_{TA}	relevance score with $TotalOveralpArea$
R_D	relevance score with $OverlapDuration$
R_{SA}	relevance score with $SummedAreaofOverlapRegions$
$Grid$	$M \times N$ cells covering the universe
$V_k^G(t_i)$	a $FOVScene$ at time t_i represented by a set of overlap grid cells between $Grid$ and $V_k^F(t_i)$
V_k^G	a video clip k represented by a set of $V_k^G(t_i)$
Q^G	a polygon query region represented by a set of grid cells
$O^G(V_k^G(t_i), Q)$	overlap region between V_k^G and Q at t_i, a set of grid cells
R_{TA}^G	relevance score using grid, extend of R_{TA}
R_D^G	relevance score using grid, extend of R_D
R_{SA}^G	relevance score using grid, extend of R_{SA}

Table 1: Summary of terms

to user as distinct groups of resulting video sections such that videos in each group will capture the query region from a different view point. Some further aspects of a complete system to query georeferenced videos – such as indexing and query optimization – will be explored as part of our future work.

4 Ranking Georeferenced Video Search Results

In video search, when results are returned to user, it is critical to present the most related videos first since human verification (viewing videos) can be very time-consuming. This can be accomplished by creating an order which will rank the videos from the most relevant to the least relevant. Otherwise, although a video clip completely captures the region user is interested in, it may be listed last within query results. It is essential to question the relevance of each video with respect to the user query and to provide an ordering based on estimated relevance.

Analyzing how the *FOVScene (FS)* descriptions of a video overlap with a query region gives clues on calculating its relevance with respect to the given query. A common metric used to measure spatial relevance is the extend of overlap region. The greater the overlap between FS and the query region, the greater the video relevance. Fig.-7 demonstrates two extreme overlap cases, where the first video covers only a small percentage of the query region and the second one covers almost all of it. It is also useful to differentiate between the videos which overlap with the query region for intervals of different length. A video which captures the query region for a longer period will probably include more details about the region of interest and therefore can be more interesting to the user. Note that during the overlap period the amount of overlap at each time instant changes dynamically for each video. Among two videos whose total overlap amounts are comparable, one can cover a small portion of the query region for a long time and rest of the overlap area only for a short time, whereas another video may cover a large portion of the query region for a longer time period. In Fig.-4, although both videos V_{46} and V_{108} have similar overlap amounts, video V_{46} includes more details for the query region since it covers a larger region for a longer period of time.

(a) (b)

Figure 4: Visualization of the overlap between query Q_{207} and videos (a) V_{46} (b) V_{108}

In the following sections, we will explain how we define the overlap between video FS and queries and propose three basic metrics for ranking video search results.

4.1 Preliminaries

Let Q be a polygon shaped query region given by an ordered list of its polygon corners:

$$Q = \{(lon_j, lat_j),\ 1 \leq j \leq m\}$$

where (lon_j, lat_j) is the longitude and latitude coordinates of j^{th} corner point of Q and m is the number of corners in Q. Suppose that a video clip V_k consists of n $FOVScenes$ and t_s and t_e is the start time and end time for video V_k, respectively. The set of FS descriptions for V_k is given by,

$V_k^F = \left\{FOVScene^{V_k}(t_i, P, \vec{d}, \theta, R) \mid 1 \leq i \leq n\right\}$. Similarly, a FS at time t_i can be denoted as $V_k^F(t_i)$.

If Q is viewable by V_k then, the set of FS that capture Q is given by,

$$SceneOverlap(V_k^F, Q) = \left\{V_k^F(t_i) \mid \text{for all } i\ (1 \leq i \leq n) \text{ when } V_k^F(t_i) \text{ overlaps with } Q\right\}$$

The overlap between V_k^F and Q at time t_i, forms a polygon shaped region, as shown in Fig.-5. Let $O(V_k^F(t_i), Q)$ denote the overlapping region between video V_k^F and query Q at time t_i. We define it as an ordered list of corner points that form the overlap polygon. Therefore,

$$\begin{aligned} O(V_k^F(t_i), Q) &= OverlapBoundary(V_k^F(t_i), Q) \\ &= \left\{(lon_j^{t_i}, lat_j^{t_i}),\ 1 \leq j \leq m\right\} \end{aligned} \tag{5}$$

where m is the number of corner points in $O(V_k^F(t_i), Q)$. The function $OverlapBoundary$ returns the overlap polygon which enclose the overlap region. In Fig.-5, these corner points are shown with labels $P1$ through $P9$. Practically, when a pie-shaped FS and polygon shaped query region intersect, the formed overlap region does not always form a polygon. If the arc of FS resides inside Q, part of the overlap region will be enclosed by an arc rather than a line. Handling such irregular shapes is usually unpractical. Therefore we estimate the part of the arc that reside within the query region Q with a series of points on the arc which have 5^o of angular distance between the previous and next point with respect to the camera location point. The implementation of the function $OverlapBoundary$ is given in Alg.-**??**. Note that $OverlapBoundary$ computes the corner points that enclose the overlap polygon where:

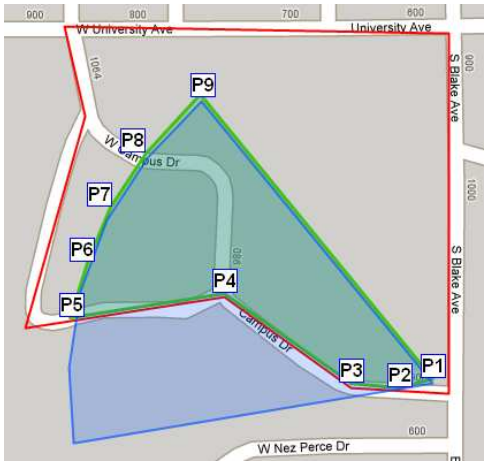


Figure 5: The overlap between a video $FOVScene$ and a polygon query

- a side of the query polygon Q crosses the arc or the sides of the FS
- a corner of the query polygon Q is enclosed within FS
- a corner point of the FS (i.e, camera location point or starting or ending points of the arc) is enclosed within Q
- part of the FS arc is enclosed within Q (the intersecting section of the arc is estimated with a series of points)

4.2 Three Metrics to Describe the Relevance of a Video

We propose three fundamental metrics to describe the relevance (R) of a video V_k with respect to a user query Q based on the followings:

1. *Total Overlap Area (R_{TA}).* Area of the region formed by the intersection of Q and V_k^F. This quantifies how much portion of Q is covered by V_k^F, emphasizing spatial relevance.
2. *Overlap Duration (R_D).* Time duration of overlap between Q and V_k^F in seconds. This quantifies how long V_k^F overlaps with Q, emphasizing temporal relevance.
3. *Summed Area of Overlap Regions (R_{SA}).* Summation of the overlap areas for the intersecting FS during the overlap interval. This balances the spatial and temporal relevance.

4.2.1 Total Overlap Area (R_{TA})

Total overlap area of $O(V_k^F, Q)$ is given by the smallest convex polygon which covers all overlap regions formed between V_k^F and Q. This boundary polygon can be obtained by constructing the convex envelope enclosing all corner points of the overlap regions. Eq.-6 formulates the computation of total overlap coverage. Function *ConvexHull* constructs the convex hull of the polygon corner points, where each point is represented as a (longitude,latitude) pair. Fig.-4 shows examples of the

total overlap coverage between the query Q_{207} and videos V_{46} and V_{108}.

$$\begin{aligned} O\left(V_k^F, Q\right) &= ConvexHull\left(\bigcup_{i=1}^{n}\left\{O\left(V_k^F(t_i), Q\right)\right\}\right) \\ &= ConvexHull\left(\bigcup_{i=1}^{n}\bigcup_{j=1}^{|O(V_k^F(t_i),Q)|}\left\{\left(lon_j^t, lat_j^t\right)\right\}\right) \end{aligned} \tag{6}$$

And the *Relevance using Total Overlap Area* (R_{TA}) is given by the area of the overlap boundary polygon $O(V_k^F, Q)$. (Eq.-7)

$$R_{TA}(V_k^F, Q) = Area\left(O(V_k^F, Q)\right) \tag{7}$$

where function *Area* returns the area of the overlap polygon $O(V_k^F, Q)$. A higher R_{TA} value implies that a video captures a larger portion of the query region Q and therefore its relevance with Q can be higher.

4.2.2 Overlap Duration (R_D)

Relevance using Overlap duration (R_D) is given by the total time in seconds that V_k^F overlaps with query Q. Eq.-8 formulates the computation of R_D.

$$R_D = \sum_{i=1}^{n-1}(t_{i+1} - t_i) \text{ for } i \text{ when } O\left(V_k^F(t_i), Q\right) \neq \varnothing \tag{8}$$

R_D is obtained by summing the overlap time for each FS in V_k^F with Q. We estimate the overlap time for each FS as the difference between two timestamps of sequential FS. If sampling rate for FS is low, (i.e., if the difference between the timestamps of two consecutive FS is large) then R_D might overestimate overlap duration. Although $O(V_k^F(t_i), Q)$ is nonempty, V_k^F might not be overlapping with Q during the whole $[t_{i+1}, t_i]$ interval. When the duration of overlap is long, the video will capture more of the query region and therefore its relevance can be higher.

4.2.3 Summed Area of Overlap Regions (R_{SA})

Total Overlap Area and *Overlap Duration* give the spatial and temporal extend of the overlap respectively. However both relevance metrics express only the properties of overall overlap and do not describe how individual FS overlap with the query region. For example, In Fig.-4, for videos V_{46} and V_{108}, although $R_{TA}(V_{46}^F, Q_{207}) = R_{TA}(V_{108}^F, Q_{207})$ and $R_D(V_{46}^F, Q_{207}) = R_D(V_{108}^F, Q_{207})$, V_{46}^F overlaps around 80% of the query region Q_{207} during the whole overlap interval, whereas V_{108}^F overlaps only 25% of Q_{207} for most of its overlap interval and overlaps 80% of Q_{207} only for the last a few FS. In order to differentiate between such videos, we propose the *Relevance using Summed Overlap Area* (R_{SA}) as the summation of areas of all overlap regions during the overlap interval. Eq.-9 formalizes the computation of R_{SA} for video V_k^F and query Q.

$$R_{SA}\left(V_k^F, Q\right) = \sum_{i=1}^{n} Area\left(O(V_k^F(t_i), Q)\right) \tag{9}$$

where function *Area* returns the area of the overlap polygon $O(V_k^F(t_i), Q)$.

4.3 Ranking Videos Based on Relevance Scores

The proposed metrics describe the most basic relevance criteria that a typical user will be interested in. R_{TA} defines the relevance based on the area of the covered region in query Q whereas R_D define relevance based on the length of the video section that captures Q. R_{SA} includes both area and duration of the overlap in relevance calculation i.e, the larger is the overlap, the bigger the R_{SA} score will be. Similarly, the longer is the overlap duration, the more overlap polygons will be included in the summation.

Since each metric bases its relevance definition on a different criteria, we may not expect to obtain a unique ranking for all three metrics. And without feedback from users it is hard to argue whether one of them is superior to the others. But, we can claim that a certain metric gives the best ranking when the query is specific in describing the properties of videos that the user is looking for. For example, if the user seeks for the videos that give the maximum coverage extend within the query region, the metric R_{TA} will give the most accurate ranking. Based on the query specification either a single metric or a combination of the three can be used to obtain the video ranking. Calculating the weighted sum of several relevance metrics (Eq.-10) is a common technique to obtain an ensemble ranking scheme.

$$Relevance\left(V_k^F, Q\right) = w_1 R_{TA}\left(V_k^F, Q\right) + w_2 R_{SA}\left(V_k^F, Q\right) + w_1 R_D\left(V_k^F, Q\right) \tag{10}$$

To obtain the optimal values for weights w_1, w_2 and w_3 we need a training data set which provides an optimized ranking based on several metrics. However constructing a reliable training data for georeferenced videos is not trivial and requires careful and tedious manual work. There is extensive research on content based classification and ranking of videos using Support Vector Machines (SVM) and other classifiers, which train their classifiers using publicly available evaluation data (for example TRECVID benchmark dataset). There is a need for a similar effort to create public training data for georeferenced videos. In Section-5 we will present results obtained through applying individual metrics to calculate the relevance score of a video. We plan to elaborate on customized multi-level ranking schemes for georeferenced video data as part of our future research work.

4.4 A Histogram Approach for Calculating Relevance Scores

Although we have the exact shape of the overlap region for each individual FS in the previous sections, the computed relevance scores do not tell us much about the distribution of the overlap throughout the query region, i.e, which parts of the query region are more frequently captured in the video and which parts are captured only in a few frames. The distribution of the density of overlap can be meaningful in questioning a video's relevance with respect to a query and in answering user customized queries, therefore should be stored.

We present a histogram based algorithm to extract and store overlap distribution by building an overlap histogram (OH). We first partition the whole geospace into disjoint grid cells such that their union covers the entire universe. Let $Grid = \{c_{i,j} : 1 \leq i \leq M \text{ and } 1 \leq j \leq N\}$ be the set of cells for the $M \times N$ grid covering the universe. Given the FS descriptions V_k^F of video V_k, the set of grid cells that intersect with a particular $V_k^F(t_i)$ can be identified as,

$$\begin{aligned} V_k^G(t_i) =& GridFOVOverlap\left(V_k^F(t_i)\right) \\ =& \left\{c_{m,n} : c_{m,n} \text{ overlaps with } V_k^F(t_i) \text{ and } c_{m,n} \in Grid\right\} \end{aligned} \tag{11}$$

$V_k^G(t_i)$ is the set of overlapping grid cells with $V_k^F(t_i)$ at time t_i, i.e., a grid representation of a FS. Then, V_k^G is a grid representation of V_k^F which is a collection of $V_k^G(t_i)$, $1 \leq i \leq n$. Histogram for V_k^G, denoted as OH_k, consists of grid cells $C_k = \bigcup_{i=1}^{n} V_k^G(t_i)$. Function $GridFOVOverlap$ given in

Alg.-?? determines these overlapping cells. In Alg.-??, we first locate the cell c_{m_p,n_p} which contains the camera location point P for $V_k^F(t_i)$, then greedily search through the neighboring cells towards the direction vector R. Initially we only check for the cells that overlap with the borderline of $V_k^F(t_i)$, then include all other cells enclosed between the border cells. (see Fig.-6)

Figure 6: Grid representation of overlap polygon

For each cell c_j in C_k, *OverlapHist* counts the number of FS samples c_j overlaps with. In other words it calculates the appearance frequency (f_j) of c_j in V_k^G (Eq.-12).

$$f_j = OverlapHist\left(c_j, V_k^G\right) = Count\left(c_j, \left\{V_k^G(t_i) : \text{for all } i,\ 1 \le i \le n\right\}\right) \tag{12}$$

Function *Count* calculates the number of $V_k^G(t_i)$ that cell c_j appears in. Note that *OverlapHist* describes only the spatial overlap between the *Grid* and the video $FOVScenes$. However, in order to calculate the time based relevance scores we also need to create the histogram that summarizes the overlap durations. *OverlapHistTime* constructs a set of time intervals when c_j overlaps with V_k^G. A set I_j holds overlap intervals with cell c_j and V_k^G such as pairs of <starting time, overlap duration>. Then, the histogram for V_k^F, i.e., OH_k, consists of grid cells each attached with a appearance frequency value and a set of overlapping intervals.

Example 1:
Histogram of video clip V_k is constructed as follows:
$OH_k = \{< c_1, f_1, I_1 >, < c_2, f_2, I_2 >, < c_3, f_3, I_3 >\}$
$= \{< (2,3), 3, \{< 2, 10 >, < 20, 5 >\} >, < (3,3), 1, \{< 10, 7 >\} >, < (4,3), 1, \{< 8, 3 >\} >\}$.
This histogram consists of three grid cells c_1, c_2, and c_3 appearing 3, 1, and 1 times in V_k^G, respectively. c_1 appears in two disjoint video segments. One starts at 2 and lasts for 10 seconds. The other starts at 20 and lasts for 5 seconds. c_2 appears once starting at 10 and lasts for 7 seconds. c_3 appears once starting at 8 and lasts for 3 seconds.

Fig.-8 demonstrates an example histogram, where different frequency values within the histogram are visualized with varying color intensities. Note that the greedy algorithm in Alg.-?? enables us to differentiate between the cells that are fully contained and the calls that are partially contained within the $FOVScene$ region. Such a distinction is useful for more accurate estimation of the overlap region. Such an improvement is out of the scope of this paper and will be elaborated as part of our future work.

4.4.1 Running Geospatial Range Queries Using Histogram

Given a polygon shaped query region Q, we first represent Q as a group of grid cells in geospace:

$$Q^G = \{ \text{ all grid cells that overlap with } Q\} \tag{13}$$

We refine the definition of overlap region as a set of overlapping grid cells (O^G) between V_k^G and Q^G. Using the histogram of V_k^G (OH_k), the overlapping grid cell set can be defined as:

$$O^G(V_k^G, Q^G) = \left\{(C_k \text{ of } OH_k) \bigcap Q^G\right\} \tag{14}$$

Note that the grid cells in O^G inherit corresponding frequencies and intervals from OH_k.

Assuming that a query region Q^G consists of four grid cells, $Q^G = \{< 2,2 >, < 2,3 >, < 3,2 >, < 3,3 >\}$ the overlapping cells with the video in Example 1 become: $O^G(V_k^G, Q^G) = \{< (2,3), 3, \{< 2,10 >, < 20,5 >\} >, < (3,3), 1, \{< 10,7 >\} >\}$.

4.4.2 Histogram Based Relevance Scores

Using the grid base overlap region O^G, we redefine the three proposed relevance metrics in Section 4.2.

Total Overlap Cells (R_{TA}^G)

R_{TA}^G is the extend of the overlap region on Q^G, i.e., how many cells in Q^G are overlapping with V_k^G. Thus, R_{TA}^G is simply the cardinality of the overlapping set $O^G(V_k^G, Q^G)$. In the above example, $R_{TA}^G = 2$.

Overlap Duration (R_D^G)

The duration of overlap between a query Q^G and V_k^G can be easily calculated using the interval sets in OH_k: *OverlapHistTime*.

$$R_D^G\left(V_k^G, Q^G\right) = CombineIntervals(OH_k) \tag{15}$$

Function $CombineIntervals$ combines the intervals in the histogram. Note that there may be time gaps when the intervals for some of cells are disjoint. There also are overlap time duration across cells. IN the above example, $R_D^G = 18$ seconds.

Summed Number of Overlapping Cells (R_{SA}^G)

R_{SA}^G is the total number of overlap occurrences between V_k^G and Q^G and therefore is a measure of how many cells in Q^G are covered by video V_k^G and how many times each overlap cell is covered. Since the histogram of a video already holds the appearance frequencies (F) of all overlap cells, it can be defined as follows assuming l cells in the histogram;

$$R_{SA}^G\left(V_k^G, Q^G\right) = \sum_{i=1}^{l} OH_k.f_i \tag{16}$$

As we mentioned in the previous sections, a histogram gives the overlap distribution within the query region with discrete numbers. Knowing the overlap distribution is helpful for interactive video search applications where user might further refine the search criteria and narrow down the search results.

4.5 Run-time Requirements of Ranking

The run-time requirement for computing the overlap region, as described in Section 4.1 and 4.2 is bounded by the number of edges in query polygon Q. The *OverlapBoundary* algorithm can be computed in $O(n)$ time where n is the number of edges in Q. For, the georeferenced video search applications we target, the query region given by the user is usually not a complex polygon, mostly a rectangle shaped region. Therefore, the overlap region for a single $FOVScene$ can be computed in constant time. However, for a realtime georeferenced search system where huge amount of videos are processed, the actual calculation of Alg.-**??** can be very time consuming. However, the grid-based histogram approach can greatly reduce the run-time computing requirement by preprocessing videos, i.e., building histograms of videos. By the fact that video data do not need to be frequently modified in many applications, the histogram approach can provide an appealing practical solution for georeferenced video search. In addition, in georeferenced video search, like many other multimedia search systems it is not critically essential to have very precise calculations. Therefore, estimation of the overlap region up to a certain error rate can be highly acceptable for the application areas we target.

5 Experimental Evaluation

5.1 Data Collection and Methodology

5.1.1 Data Collection

To collect georeferenced video data, we have constructed a prototype system which includes a camera, a 3D compass and a GPS receiver. We used the JVC JY-HD10U camera with a frame size of approximately one megapixel (1280x720 pixels at a data rate of 30 frames per second). It produces MPEG-2 HD video streams at a rate of slightly more than 20 Mb/s and video output is available in real time from the built-in FireWire (IEEE 1394) port. To obtain the orientation of the camera, we employed the OS5000-US Solid State Tilt Compensated 3 Axis Digital Compass, which provides precise tilt compensated headings with roll and pitch data. To acquire the camera location the Pharos iGPS-500 GPS receiver has been used. A program was developed to acquire, process, and record the georeferences along with the MPEG2 HD video streams. The system can process MPEG2 video in real-time (without decoding the stream) and each video frame can be associated with its viewable scene information. In all of our experiments, a FOVScene (FS) was constructed every second, i.e., one FS per 30 frames of video. More details on acquisition and synchronization issues have been provided in Section 3.2 *[Check later depending on previous sections.]*. Although our sensor rich video recording system has been tested mainly with a camera that produces MPEG-2 video output, with little effort it can be configured to support any digital camera producing compressed or uncompressed video streams.

Figure 1 shows the setup for our recording prototype. We have mounted the recording system setup on a pickup truck and captured video outdoors in Moscow, Idaho, traveling at different speeds (max. 25 MPH). During video capture, we frequently changed the camera view directions. The captured video covered a 6 kilometers by 5 kilometers size region quite uniformly. However, for a few popular locations we shot several videos, each viewing the same location from different directions. The total captured data includes 134 video clips, ranging from 60 to 240 seconds in duration. Figure 7 shows a visualization example of camera viewable scenes for two video files on a map. For visual clarity, viewable scene regions are drawn every three seconds. Due to space limitations we cannot include more example visualizations. Further samples can be found at *http://eiger.ddns.comp.nus.edu.sg/geospatialvideo/ex.html*.

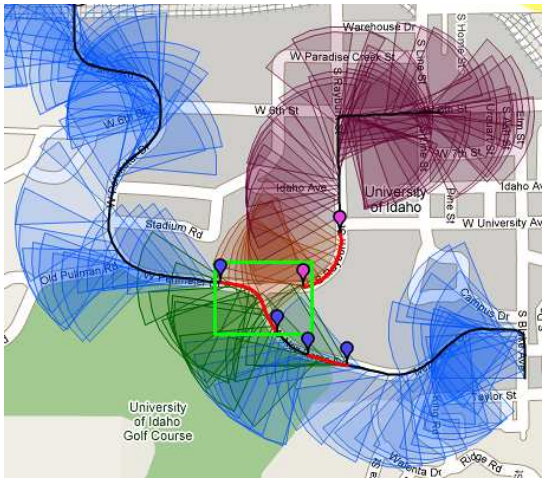


Figure 7: Visualization of viewable scenes on a map.

5.1.2 Methodology

Using the collected 134 video data set that covers 6 kilometer by 5 kilometer area, we generated 250 random range queries with a fixed query range of 300 meter by 300 meter. For each query, we searched the georeferenced video dataset for overlapping videos with the query region (Filter Step). We then calculated the relevance scores based on the three metrics in Section-4.2. The rank lists RL_{TA}, RL_{SA} and RL_D are constructed from the relevance metrics R_{TA}, R_{SA} and R_D respectively. A rank list is a sorted list of video clips in descending order of their relevance scores.

In order to evaluate the accuracy of rankings from our proposed schemes, one needs the "ground truth"' rank order for comparison. Unfortunately, there is no classified publicly available georeferenced video data (similar to TRECVID benchmark evaluation data for still images *[Check later if this is correct]*) that can be the reference for comparison. Therefore we first analyzed and compared the rankings from the proposed schemes each other. Next, we independently conducted experiments to rank the results by human judges. Finally, by comparing the results from human judges and those from the proposed schemes, we evaluated the accuracy of our ranking schemes.

We conducted two set of experiments to evaluate the ranking accuracy of the proposed methods:

1. Experiment 1: We compared the rankings RL_{TA}, RL_{SA} and RL_D with each other over the whole set of 250 queries.

2. Experiment 2: Among the 250 random queries we picked 25 easily recognizable query regions and asked human judges to rate each video file using a four point scale ranging from "3-highly relevant" down to "0-irrelevant". We compared our results to user provided feedback labels over this 25 random queries.

5.1.3 Evaluation Metrics.

Since each ranking scheme interprets the relevance in a different way, it is not expected to obtain a unique result from all schemes. However, we claim that they all should have similar sets of video clips within the top N of their rank lists for some N because of the nature of geospatial query. Having similar results from all three ranking algorithms would show that the resulting videos are most interesting to the user. To compare the accuracy of results, we adopted the *Precision at N (P(N)) metric [Any reference?]*, which is a popular method that reports the fraction of relevant videos ranked in the top N results. We redefine *P(N)* as the fraction of common videos ranked

		MAP at N=1	MAP at N=2	MAP at N=5	MAP at N=10	MAP at N=15	MAP at N=20
Compare All	$\frac{topN(RL_{TA}) \bigcap topN(RL_D) \bigcap topN(RL_{SA})}{N}$	0.60	0.789	0.918	0.993	0.999	1.0
Compare RL_{TA} and RL_{SA}	$\frac{topN(RL_{TA}) \bigcap topN(RL_{SA})}{N}$	0.727	0.839	0.961	0.993	1.0	1.0
Compare RL_{TA} and RL_D	$\frac{topN(RL_{TA}) \bigcap topN(RL_D)}{N}$	0.677	0.842	0.933	0.987	0.999	1.0
Compare RL_{SA} and RL_D	$\frac{topN(RL_{SA}) \bigcap topN(RL_D)}{N}$	0.745	0.885	0.947	0.987	1.0	1.0

Table 2: Comparison of proposed ranking methods: RL_{TA}, RL_{SA} and RL_D

in top N results from more than one rank list. Note that the exact rank of videos within the top N is irrelevant. P(N) only shows the precision of a single query, therefore to measure the average precision over multiple queries, we use *Mean Average Precision (MAP)*, which is the mean of P(N)s for multiple queries. We evaluate the results of Experiment 1 with MAP scores.

For Experiment 2 which includes human judgement, in addition to MAP, a second evaluation metric namely *Discounted Cumulated Gain (DCG)* was used *[Reference]*. DCG systematically combines video rank order and degree of relevance. The discounted cumulative gain vector $\vec{DCG}$ is defined as

$$DCG[i] = \begin{cases} G[1] & \text{if i=1} \\ DCG[i-1] + G[i]/log_e i & \text{otherwise} \end{cases}$$

where $\vec{G}$ is the gain vector which contains the gain values for the ranked videos in order. The gain values correspond to the user assigned relevance labels ranging from 0 to 3. Note that a video with lower relevance label listed at top rank will dramatically increase the DCG sum. But a video with high relevance label listed lower in the rank list will not contribute much to the sum. This is because the lower the position of a relevant video the less valuable it is for the user. The perfect ordering where all highly relevant videos are ranked at the top and less relevant documents are listed lower in the rank list will give the ideal DCG vector. *Normalized-DCG* (NDCG) is the final DCG sum normalized by the DCG of ideal ordering. The higher the NDCG of a given ranking the more accurate it is.

5.2 Comparison of Ranking Accuracy

5.2.1 Comparison of Proposed Ranking Schemes

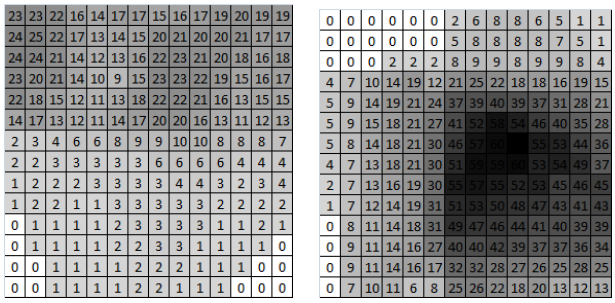
We compare the ranking accuracy of RL_{TA}, RL_{SA} and RL_D using MAP. In Table-2, the first row calculates MAP as the average ratio of the videos that are common to all three rank lists within the top 1, 2, 5, 10 and 20 ranked results for all 250 queries. Second, third and fourth rows give the MAP scores in RL_{TA} and RL_{SA}, RL_{TA} and RL_D, RL_D and RL_{SA} respectively. The results show that the precision increases as N grows and almost full precision is achieved at N=10. Note that we get a very high precision even at N=5. This implies that all three proposed schemes similarly identify the most relevant videos. Table-2 displays RL_{TA}, RL_{SA}, RL_D from a specific query Q_{207}.

The rank differences in RL_{TA}, RL_{SA} and RL_D are mainly due to the different interpretations of relevance. To further investigate the differences and similarities between the rankings, see V_{46} and V_{108} in Table-2. Both videos overlap Q_{207} almost the same amount of time and they both cover almost the whole query region. Thus, R_{TA} and R_D scores for both videos are very close. However R_{SA} for V_{108} is much higher than V_{46}. To investigate the difference, we built the overlap histograms OH_{46} and OH_{108} and extracted the cells that overlap with query Q_{207}. Color highlighted visualizations of $O^G(V^G_{46}, Q^G)$ and $O^G(V^G_{108}, Q^G)$ are shown in Fig.-8. Fig.-8(b) has a higher color

	RL_{TA}	R_{TA} score (km^2)	RL_{SA}	R_{SA} score (km^2)	RL_D	R_D score ($secs$)
1	**46**	0.087	**108**	1.726	**108**	65
2	**108**	0.084	**43**	0.813	**46**	61
3	**43**	0.063	**46**	0.558	**43**	42
4	**107**	0.055	**42**	0.359	**107**	38
5	**42**	0.052	**107**	0.338	**133**	31
6	**131**	0.045	**131**	0.291	**131**	25
7	**132**	0.045	**133**	0.135	**42**	18
8	**133**	0.038	**132**	0.087	**106**	16
9	**109**	0.022	**109**	0.073	**118**	11
10	**118**	0.018	**118**	0.045	**109**	10
11	**47**	0.004	**106**	0.025	**44**	6
12	**106**	0.004	**44**	0.008	**132**	5
13	**44**	0.001	**47**	0.004	**47**	1
14	**65**	0.001	**65**	0.001	**65**	1

Table 3: The ranked video results and relevance scores obtained for Q_{207}

intensity in the middle showing that V_{108} intensively covers the middle part of Q_{207}. This caused by that the middle portion of Q_{207} was far more frequently covered by V_{108} than V_{46}. At this point we can not argue whether one of the ranking methods is better than the others. We believe that each ranking scheme interprets a different aspect of relevance, therefore query results should be customized based on user preferences.



(a) (b)

Figure 8: Color highlighted visualizations for overlap histograms for videos V_{46} and V_{108} [Isn't this switched?]

5.2.2 Comparison with User Feedback

This set of experiments aimed to evaluate the accuracy of our ranking methods by comparing results to user provided relevance feedback. Relevance judgements were made by a student familiar to the region where the videos are captured. We selected 25 query regions from the 250 queries, which are relatively easy to be recognized by human in video. The selected 25 queries returned total 103 videos and each query returned 14 videos on the average. The user manually analyzed all these 103 videos in random order and evaluated the relevance of these videos for each of the 25 queries. The user was asked to rate the relevance based on a four-point scale: *"3 - highly relevant"*, *"2 - relevant"*, *"1 - somehow relevant"* and *"0 - irrelevant"*. Trajectories of camera movements were displayed on a map for all 103 videos. Finally, the user created a rank list per query.

We compare the rankings RL_{TA}, RL_{SA} and RL_D to the user rankings using the metrics DCG and NDCG for the 25 queries. For the comparison, the average of DCG vectors for the rankings

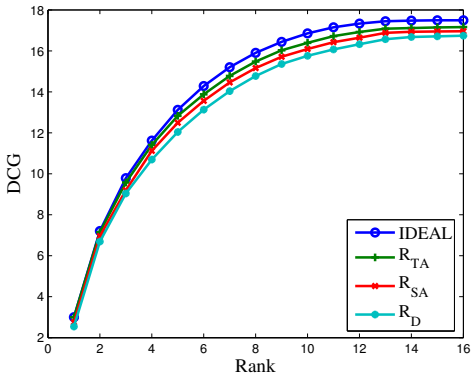


Figure 9: Discounted Cumulated Gain (DCG) curves

were used. Fig.-9 shows the DCG vector curves for the rank lists RL_{TA}, RL_{SA}, RL_D and the ideal curve for ranks 1 to 16. Ideal curve corresponds to the DCG vector based on the user ranking. Clearly, the DCG curves for the proposed schemes have a close match with the ideal DCG curve.

Next, NDCG scores with respect to the ideal curve were calculated. The NDCG scores of RL_{TA}, RL_{SA} and RL_D were 0.975, 0.951 and 0.921, respectively. All scores are close to 1, which implies that all three are highly successful in ranking the most relevant videos at top, similar to human judgement. We observed that rank differences among them mostly occurred in rating less relevant videos. Recall that DCG and NDCG reward relevant videos in the top ranked results more heavily than those ranked lower. High NDCG scores also justify that the proposed ranking methods successfully identify the most relevant videos.

Among the proposed schemes, the highest precision was obtained by RL_{TA} at all levels. This is because human perception for the relevance is more related to what one can actually see clearly (i.e., spatial perception) rather than how long one sees the same thing (i.e., temporal perception). All three ranking schemes describe different properties of video, and importance of a video is highly subjective to users and the criteria users are looking for. Fig.-9 also clearly shows that RL_{TA} in overall has the best accuracy with respect to user ranking among the three.

We are aware that such user judgement is prone to errors. More accurate results can be obtained by performing an intensive user study with far more number of human judges, videos and queries. Such an extensive study is out of the scope of this paper and will be part of our future work.

5.3 Evaluating the Computational Performance

This section evaluates the computational cost of the proposed schemes. In our implementation, three steps account for the computational cost: 1) loading georeferenced data, i.e., FOVScene descriptions, from a hard disk, 2) filter step to exclude videos with no overlap, 3) calculating the area of overlap between FOVScenes (results of the filter step) and query regions, and 4) computing the relevance scores for the three rank lists. For R_{TA} and R_{SA}, step 3 consumes around 60% of the computation time. In step 4, R_{SA} just needs to sum the areas of overlap polygons, however R_{TA} needs to compute the extend of all overlap polygons therefore it takes longer to construct the rank list. R_D is computationally the most efficient since R_D only extracts the time of overlap and skips the overlap area calculation.

Using a 2.33 GHz Intel Core2 Duo PC, we measured the processing time of each scheme to perform the same 250 queries in Section 5.2.1. The test data included 134 videos with a total duration

Step	Calculating RL_{TA}		Calculating RL_{SA}		Calculating RL_D	
	Avg No. of V processed	Avg Time (secs)	Avg No. of V processed	Avg Time (secs)	Avg No. of V processed	Avg Time (secs)
1. Load FOVScene descriptions from disk	134	0.523	134	0.523	134	0.523
2. Filter step	134	0.016	134	0.016	134	0.016
3. Calculate the area of Overlap polygons	8.46	1.176	8.46	1.176	8.46	0.527
4. Calculate the Relevance Scores	8.46	0.367	8.46	0.097	8.46	0.100
Total time (sec)		2.082		1.812		1.166

Table 4: Measured computational time per query

of 175 minutes. A FS was recorded per every second of video so total 10500 FS representations were used in the calculations. Detailed processing time measurements for running the major steps of ranking schemes are summarized in Table-4. Step 1 and 2 were required for all queries and all 134 videos were processed per query. It is important to note that, during query processing, a vast majority of the videos were filtered out through the filter step. As shown in Table-4 for each query, on average 8.46 out of 134 were actually processed in step 3 and 4 since all other videos were excluded through the filter step. The details of the filter step is explained in Section4.3. For a particular query, the average processing time required to construct RL_{TA}, RL_{SA} and RL_D was around 2.082, 1.811 and 1.66 seconds respectively.

The query processing time depends on the number of FOVScenes to be processed which varies query by query. Thus, we next examine how the processing time changes as the number of videos increases. Fig.-10 shows the processing time vs. number of videos for the three rankings. It shows that the processing time linearly grows as a function of videos, i.e., as a function of number of FOVScenes. The small fluctuation was caused from the different number of FOVScenes in a specific video and the number of FOVScenes after the filter steps in a specific query. Thus, we can compute the average time to process a single FOVScene per query. When the processing time per query took 2.082 seconds with 134 videos (total 175 minutes so 10500 FOVScenes) for RL_{TA}, the average processing time per FOVScene per query was 0.198 milliseconds. Similarly, it was 0.172 and 0.110 milliseconds for RL_{SA} and RL_D, respectively. These numbers can provide a good estimation of query processing time with a larger data set. For example, when the size of query range is same but the number of FOVScenes increases to 100,000[1], we can estimate the average query processing time for RL_{TA} as $0.198msec \times 100,000 = 198seconds$.

Section-4.5 briefly discussed the run time requirement of the proposed schemes. In this study we only present the initial findings and do not aim to provide any contributions in efficient retrieval and indexing of the *FOVScene* descriptions. The used methods are not optimized for computational efficiency. However, it is worth to mention that calculating the area of overlap between a pie-shaped *FOVScene* and polygon shaped query is computationally expensive and might not be practical for realtime applications. The histogram based ranking introduced in Section-4.4.2 can move most of the costly computation overhead to offline preprocessing step, leaving the query processing step simpler and faster. Next, we will present the findings on the accuracy and efficiency of histogram based ranking.

[1]There is no direct relation between the number of FOVScenes and the length of video because FOVScene can be sampled with various intervals.

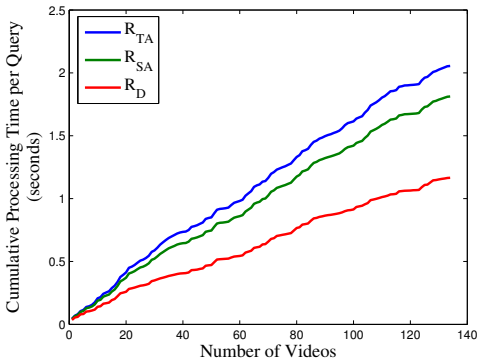
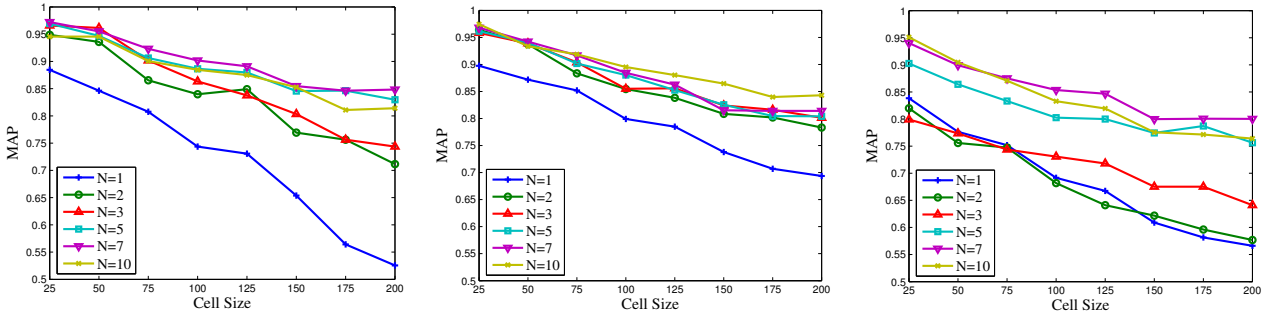


Figure 10: Processing time per query vs Number of videos

5.4 Ranking based on Histogram

We built the overlap histograms for all 134 videos as described in Section-4.4. The same 250 queries were processed using the histograms and the relevance scores were calculated for the returned videos based on the metrics we proposed in Section-4.4.2. Let RL^G_{TA}, RL^G_{SA} and RL^G_D be the rankings obtained from relevance metrics R^G_{TA}, R^G_{SA} and R^G_D respectively.

First, in order to evaluate the accuracy of RL^G_{TA}, RL^G_{SA}, RL^G_D, we compared them to precise rankings RL_{TA}, RL_{SA}, RL_D and measure the precision for various cell sizes. Recall that we use the exact area of overlap polygon for calculating the relevance scores for precise rankings whereas the histogram approximates the overlap polygon with grid cells. Therefore, we use rank lists RL_{TA}, RL_{SA} and RL_D as baseline for comparison. Fig.-11 shows the results using the MAP metric for grid cell sizes varying from 25m by 25m to 200m by 200m. Note that the size of query range was 300m by 300m. MAP Results were averaged across all queries in the test. The results showed that the precision for all three histogram based rankings decreases linearly as the cell size gets larger. It is expected because a larger cell size means a coarser representation of overlapping. However, the degradation of precision was insignificant (especially considering the performance gain explained later) when the cell size becomes small and N becomes large. For example, when the cell size is smaller than 100m by 100m and N is greater than two, MAP becomes greater than 0.9 in Fig.-11(b).



$(a) RL^G_{TA}$ $(b) RL^G_{SA}$ $(c) RL^G_D$

Figure 11: MAP at N for (a) RL^G_{TA}, (b) RL^G_{SA} and (c) RL^G_D for varying cell sizes

Next, we measured the query processing time of the histogram based ranking and compared it with those of the precise rankings. Fig.-12 shows the processing times per query with respect

to the number of videos for both histogram based and R_{TA} when the cell size was [put the cell size used]. Obviously, the histogram ranking demonstrated a high superiority to R_{TA}. This is because most of costly overlap computations are performed while the histogram is being built as a pre-processing step (e.g., when the video is first uploaded to system). The histograms of all videos are constructed just once and all queries can share them. The result is a short query processing time. Foe example, the average query processing time was around just 5% of that of R_{TA} as shown in Fig.-12. Similar results were obtained for other rankings schemes. Again our main goal in this work is not to provide contributions in efficient querying and storage of FOVScenes. Although the histogram based ranking achieves a greatly better performance, there can still be open ways of optimizations such as adopting a well studied index structure.

We already showed that the accuracy of the histogram ranking is highly dependent on the cell size. The smaller the grid cell size the better estimation histogram achieves. However, time to build the histogram increases as cell size gets smaller. We investigated the tradeoff between the precision of ranking and computational cost of building histograms while varying cell sizes. efficiency. CPU processing times to build histograms were recorded as seconds. Fig.-13 shows the change in both precision and CPU time for varying cell sizes for R^G_{TA}. [Is this time to build 134 histograms? or ONe?] As the cell size increases, the precision linearly decreases while CPU time exponentially decreases. When cell size exceeds 75mx75m, the CPU time decreases little while the precision continues to drop steadily. Thus, in our experiments, the cell size between 50mx50m and 75mx75m provides a good tradeoff between the accuracy and build overhead of histograms.

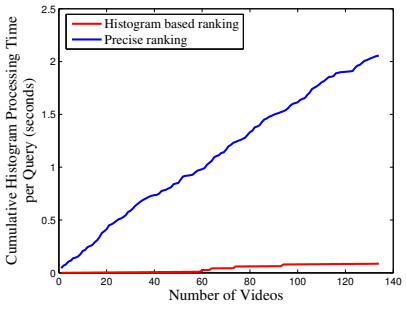


Figure 12: Comparison of precise and histogram based query processing

One important and unique advantage of histogram is to describe both the extend and density of overlap between video *FOVScenes* and the query region. By analyzing the overlap distribution in histogram, it is possible for users to further understand results. Also it can be quite useful in interactive video search where the overlap density through the query region is used to guide the user to further drill down to more specific queries. For example, a visualization of the histogram data similar to Fig.-8 can be provided to the user for the top ranked videos so that user can interactively customize the query and easily access the information he/she is looking for. We plan to elaborate on histogram data analysis as part of our future work.

[Sakire, the following paragraph is not very clear. I know the intention to provide a comparison on exact ranks. But the table does not look good. O notation is confusing. We used O as overlap. Moreover, it is not complete. What if two rank lists have different number of videos?]

Finally, we look at the differences between the rank-orders of videos in histogram based rankings (RL^G_{TA}, RL^G_{SA}, RL^G_D) and original rankings (RL_{TA}, RL_{SA}, RL_D). First, we extract the rank-order of each individual video both in RL^G_{TA} and RL_{TA}, and for each query we compute the mean absolute difference between the rank-orders of all videos in the rank list for that query. We repeate the same

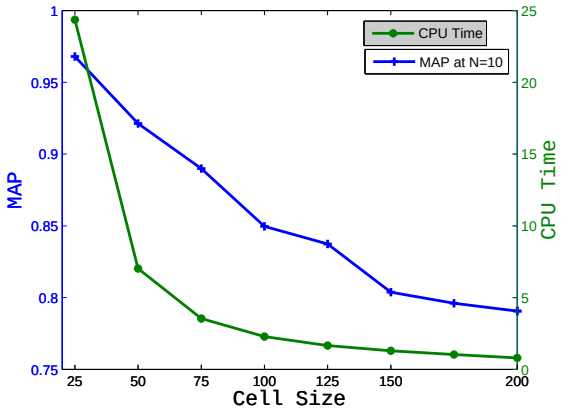


Figure 13: Evaluation of computational time and precision with respect to grid cell size

	Average rank-order difference	Cell Size 25mx25m	Cell Size 50mx50m	Cell Size 100mx100m	Cell Size 150mx150m	Cell Size 200mx200m
RL^G_{TA}	$\frac{1}{\|Q\|}\sum_{q=1}^{\|Q\|} Avg\left(\left\|O(RL_{TA}(q),i) - O(RL^G_{TA}(q),i)\right\|\right)$	0.2243	0.3191	0.7508	0.9662	1.1973
RL^G_{SA}	$\frac{1}{\|Q\|}\sum_{q=1}^{\|Q\|} Avg\left(\left\|O(RL_{SA}(q),i) - O(RL^G_{SA}(q),i)\right\|\right)$	0.2345	0.4069	0.7655	0.9808	1.1116
RL^G_{D}	$\frac{1}{\|Q\|}\sum_{q=1}^{\|Q\|} Avg\left(\left\|O(RL_{D}(q),i) - O(RL^G_{D}(q),i)\right\|\right)$	0.4378	0.7822	1.0366	1.3411	1.5415

$|Q|$: number of queries in dataset, Avg : Average, O: Order function

Table 5: Comparison of rank orders RL^G_{TA}, RL^G_{SA}, RL^G_D with RL_{TA}, RL_{SA}, RL_D

for the rank lists RL^G_{SA} vs. RL_{SA} and RL^G_D vs. RL_D. Results are reported in Table-5. When grid cell size is small the mean order difference is as low as 0.2. Even for large cell sizes the mean order difference is around 1.2 which implies that on average each video is displaced ± 1 position in rank list. Therefore, the histogram rankings not only ensure high precision but also guarantee accurate rank order.

6 Discussion

Talk about other meta-data associated with videos that can be used to improve ranking.

- (i) Average angular distance (or separation) between the overlap region and direction vector as observed from the camera location.
- (ii) Ground Sample Distance (GSD) between the overlap area and camera location. GSD refers to the distance on the ground represented by each pixel in the x and y components, expressed in ground units. For example, if an orthophoto has a 1.0 m GSD, each pixel represents a ground area measuring 1 m x 1 m.

Mention that the histogram based approach gives clues about how heavily does a video show certain parts of the query region. (i.e. query region borders, center or corners). Discuss an interactive interface where user can visually see how the FOVscenes of a video intersects with the query

region and run custom queries.

Our contributions in this paper have been threefold. First, we introduced a methodology for automatic annotation of video clips with a collection of meta-data such as camera location, viewing direction, field-of-view, etc. Such meta-data can provide a comprehensive model to describe the scene a camera captures. We proposed a *viewable scene* model that strikes a balance between the analytical complexity and the practical applicability of the scene description to enable effective and efficient search of videos. Second, we described our implemented prototype which demonstrates the feasibility of acquiring, storing, searching and retrieving meta-data enhanced georeferenced video based on the proposed *viewable scene* model. We collected a sufficiently large set of georeferenced video data using our prototype system. Finally, we demonstrated the benefits of using our approach in accurately retrieving the relevant video segments for a given query. We plan to extend our work in several directions:

(i) In our initial work we used a simple relational database schema to store camera *viewable scenes*. We also mentioned some alternative spatio-temporal structures that can be used to index the area that a camera *viewable scene* covers. However, we argue that current work in spatio-temporal indexing can not fully optimize the search of a dynamically changing *viewable scene*. Therefore, there is a strong need for a better index structure that would specifically target georeferenced annotations of video data.

(ii) In our study we only show examples for simple spatial range queries. However, the proposed *viewable scene* model that includes the camera view direction and camera location provides a rich information base to answer more complex geospatial queries. Similarly, when query results are presented to a user, the resulting video segments can be ranked based on how relevant they are to the query requirements and user interests. In our initial work, we show how the search accuracy can be improved even for simple range queries using our *viewable scene* model. We will elaborate on video ranking in our future work.

(iii) There are several additional factors that influence the effective *viewable scene* in a video, such as occlusions, visibility depth, resolution, etc. The proposed *viewable scene* model has to be extended and improved to account for these factors. Occlusions have been well studied in computer graphics research. We plan to incorporate an existing occlusion determination algorithm into our model.

(iv) To enable video search on a larger scale, a standard format for georeferenced video annotations must be established and issues for enabling automated integration with other providers' data have to be investigated.

7 Conclusion

References

[1] *Camera Calibration Toolbox for Matlab. http://www.vision.caltech.edu/bouguetj/calib_doc/.*

[2] *Flickr. http://www.flickr.com.*

[3] *Geobloggers. http://www.geobloggers.com.*

[4] *Woophy. http://www.woophy.com.*

[5] Kate Beard and Vyjayanti Sharma. Multidimensional ranking for data in digital spatial libraries. *Int. J. on Digital Libraries*, 1(2):153–160, 1997.

[6] Boris Epshtein, Eyal Ofek, Yonatan Wexler, and Pusheng Zhang. Hierarchical Photo Organization Using Geo-Relevance. In 15^{th} *ACM Intl. Symposium on Advances in Geographic Information Systems (GIS)*, pages 1–7, 2007.

[7] Shantanu Gautam, Gabi Sarkis, Edwin Tjandranegara, Evan Zelkowitz, Yung-Hsiang Lu, and Edward J. Delp. Multimedia for Mobile Environment: Image Enhanced Navigation. volume 6073, page 60730F. SPIE, 2006.

[8] Stefan Gobel and Peter Klein. Ranking mechanisms in meta-data information systems for geo-spatial data. In *EOGEO Technical Workshop*, 2002.

[9] Clarence H. Graham, Neil R. Bartlett, John Lott Brown, Yun Hsia, Conrad C. Mueller, and Lorrin A. Riggs. *Vision and Visual Perception*. John Wiley & Sons, Inc., 1965.

[10] Eugene Hecht. *Optics*. Addison-Wesley Publishing Company, 4^{th} edition, August 2001.

[11] Tae-Hyun Hwang, Kyoung-Ho Choi, In-Hak Joo, and Jong-Hun Lee. MPEG-7 Metadata for Video-Based GIS Applications. In *Geoscience and Remote Sensing Symposium*, pages 3641–3643, vol.6, 2003.

[12] Yannis Ioannidis. The history of histograms (abridged. In *Proc. of VLDB Conference*, 2003.

[13] Rieko Kadobayashi and Katsumi Tanaka. 3D Viewpoint-Based Photo Search and Information Browsing. In 28^{th} *Intl. ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 621–622, 2005.

[14] Lyndon S. Kennedy and Mor 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.

[15] Kyong-Ho Kim, Sung-Soo Kim, Sung-Ho Lee, Jong-Hyun Park, and Jong-Hyun Lee. The Interactive Geographic Video. In *Geoscience and Remote Sensing Symposium*, pages 59–61, vol.1, 2003.

[16] Ray R. Larson and Patricia Frontiera. Geographic information retrieval (gir) ranking methods for digital libraries. In *JCDL '04: Proceedings of the 4th ACM/IEEE-CS joint conference on Digital libraries*, pages 415–415, New York, NY, USA, 2004. ACM.

[17] Xiaotao Liu, Mark Corner, and Prashant Shenoy. SEVA: Sensor-Enhanced Video Annotation. In 13^{th} *ACM Intl. Conference on Multimedia*, pages 618–627, 2005.

[18] Mor Naaman, Yee Jiun Song, Andreas Paepcke, and Hector 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.

[19] 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.

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

[21] Ian Simon and Steven M. Seitz. Scene segmentation using the wisdom of crowds. In *Proc. ECCV*, 2008.

[22] Alan F. Smeaton, Paul Over, and Wessel Kraaij. Evaluation campaigns and trecvid. In *MIR '06: Proceedings of the 8th ACM International Workshop on Multimedia Information Retrieval*, pages 321–330, New York, NY, USA, 2006. ACM Press.

[23] Alan F. Smeaton, Paul Over, and Wessel Kraaij. High-Level Feature Detection from Video in TRECVid: a 5-Year Retrospective of Achievements. In Ajay Divakaran, editor, *Multimedia Content Analysis, Theory and Applications*, pages 151–174. Springer Verlag, Berlin, 2009.

[24] Carlo Torniai, Steve Battle, and Steve Cayzer. *Sharing, Discovering and Browsing Geotagged Pictures on the Web*. Springer, 2006.

[25] Kentaro Toyama, Ron Logan, and Asta Roseway. Geographic Location Tags on Digital Images. In 11^{th} *ACM Intl. Conference on Multimedia*, pages 156–166, 2003.