Keyframe Presentation for Browsing of User-generated Videos on Map Interfaces

Jia Hao, Guanfeng Wang, Beomjoo Seo, Roger Zimmermann
School of Computing, National University of Singapore, Singapore 117417
[haojia,wanggf,seobj,rogerz]@comp.nus.edu.sg

ABSTRACT
To present user-generated videos that relate to geographic areas for easy access and browsing it is often natural to use maps as interfaces. A common approach is to place thumbnail images of video keyframes in appropriate locations. Here we consider the challenge of determining which keyframes to select and where to place them on the map. Our proposed technique leverages sensor-collected meta-data which are automatically acquired as a continuous stream together with the video. Our approach is able to detect interesting regions and objects (hotspots) and their distances from the camera in a fully automated way. Meaningful keyframes are adaptively selected based on the popularity of the hotspots. Our experiments show very promising results and demonstrate excellent utility for the users.

Categories and Subject Descriptors
H.3.5 [Information Storage and Retrieval]: On-line Information Services—Web-based services; I.4.8 [Image Processing and Computer Vision]: Scene Analysis—Sensor Fusion

General Terms
Algorithms, Measurement, Performance

Keywords
Hotspot detection, visible distance, keyframe extraction

1. INTRODUCTION
We consider the problem of presenting user-generated outdoor videos on a map interface for easy browsing and viewing. Creating such videos is now popular and easy due to the technological advances in camera systems. Specifically, smartphones can now record good-quality video and they are carried by millions.

For videos that relate to geographic locations, regions or objects, map-based interfaces are a natural way to present them and make them available for browsing and viewing. Example interfaces are Bing Maps and Google Earth. Two aspects are crucially important for a good thumbnail presentation solution: (1) relevant keyframes must be extracted to show the most salient features of a video and (2) the keyframe thumbnails need to be placed in accurate locations on the map-based interface.

Existing solutions are limited in that they allow users to post videos based on a single GPS location, usually the initial camera position, and a single keyframe is placed there. However, this approach is unsatisfactory under two common conditions: (a) user mobility results in videos taken along some trajectory, so a single thumbnail location is insufficient to describe the video, and (b) the location of the most salient object in the video is often not at the position of the camera, but may in fact be quite a distance away. Consider the example of a user videotaping the pyramids of Giza – he or she would probably need to stand at a considerable distance. However, any user looking for videos of the pyramids would most likely zoom into their actual location.

Here we present our novel and unconventional solution to address the above two challenges. We do not restrict the movement of the camera operator and hence assume that mobile videos may be shot along some trajectory. In the first phase, to obtain the viewable scenes, we continuously collect GPS location and viewing direction information (via a compass sensor) together with the videos. This is easily achievable today as smartphones contain all the necessary sensors for recording videos that are annotated with meta-data. In the second phase we process the sensor meta-data of multiple videos to identify hotspot regions containing important objects or places. The third phase computes a set of visible distances between the camera locations and the hotspots. Finally, in the last step we leverage all the resulting information to extract relevant keyframe thumbnails and place them in their correct location on a map-based interface. Based on our results we believe that our approach provides high utility to the users in a fully automatic process that requires no manual intervention.

Existing work on visible distance estimation relates to the field of adaptively adjusting vehicle speeds [4, 3], which does not have the same goal as our work. The visible distance we aspire to estimate is the distance between the camera location and the key objects in the video frames. To the best of our knowledge, our method is the first to investigate this kind of effective visible distance for video. Most existing keyframe extraction techniques are based on the signal content of video streams [7, 6]. Due to the time-consuming pro-
2. FRAMEWORK DESIGN

Our framework consists of four phases: data collection, hotspot estimation from profiling, visible distance $R$ estimation, and keyframe extraction. In Section 2.1, we present how to collect videos and their sensor measurements. Section 2.2 describes how we generate a popularity map that captures how often an area is pointed at by camera views and how we infer most popular locations. In the rest of this paper we term such a location (or its grid cell representation on a 2D map) as "hotspot". Such hotspots can be a part of an attraction or consist of a more diffuse area that contains no specific physical objects but may be of interest to users (e.g., a beautiful valley). To estimate $R$ accurately, we then associate a camera view with the closest hotspot in Section 2.3. Finally Section 2.4 presents our keyframe extraction algorithm which utilizes all the results of the previous steps.

2.1 Data Collection

A camera positioned at a given point $P$ in geo-space captures a scene whose covered area is referred to as the camera field-of-view (FOV, also called the viewable scene). We adapt the FOV model introduced in our prior work [1], which describes a camera’s viewable scene with four parameters: camera location $P$, camera orientation $α$, viewable angle $θ$ and maximum visible distance $R^{max}$ (see Eqn. 1).

\[
FOV \equiv \langle P, α, θ, R^{max} \rangle
\] (1)

The camera position $P$ consists of the latitude and longitude coordinates read from a positioning device (e.g., GPS) and the camera orientation $α$ is obtained based on the orientation angle provided by a digital compass. $R^{max}$ is the maximum visible distance from $P$ at which a large object within the camera’s field-of-view can be recognized. The angle $θ$ is calculated based on the camera and lens properties for the current zoom level [2]. To acquire sensor-annotated videos we have written two custom recording apps for Android and iOS smartphones. When a mobile device begins to capture video, the GPS and compass sensors are turned on to record the location and orientation of the camera. Our data-acquisition software fetches such sensor values as soon as new values are available. Video data are processed in real-time to extract frame timecodes ($t_f$). All collected meta-data (i.e., location, direction, frame timecode and video ID) are combined as a tuple and uploaded to a server.

Figure 1(a) illustrates the concept of visible distance $R$ estimation based on hotspots. Along the camera trajectory (black line segments), the camera views (blue arrows) tend to point to some areas (red circles) more frequently, and $R$ can be determined as the distance between such popular areas, i.e., hotspots, and the camera locations.

2.2 Hotspot Detection

First, a target space is assumed on a 2D geographical map which is partitioned into equally-spaced square grid cells. For estimating the hotspots in the space, we use a grid counting-based popularity method. Our algorithm maintains a data structure for every map grid cell containing a monotonically increasing counter of interest. Without prior knowledge on the underlying landmarks or attractions, it is incremented whenever its grid cell is visually covered by an FOV. We investigate a line-based coverage model who uses a line vector with length $R^{max}$. Its center is at the GPS location, and its heading is the camera direction – see Figure 1(b). With this model we increase the counters of all the grid cells only along the center line. The rationale is that users tend to focus on objects located at the median plane of the FOV.

Among all the grid cells, we then compute the local maxima of the map and determine them as hotspot if their probability is higher than those of all their neighboring cells and their differences exceed a certain threshold.

2.3 Effective Visible Distance Estimation

Let $H$ be a set of estimated hotspots computed from the previous stage. When new sensor values related to a camera view arrive in the system, they are transformed into a measurement vector $X = \{x_1, \ldots, x_H\}$, where $x_i$ consists of the subtended angle $α$ of a compass value to the hotspot $i \in H$ and the Euclidean distance $d$ to the hotspot. We estimate the effective visible distance $R$ by the closeness – in terms of lower angular disparity – to a hotspot and select the closest one. Eqn. 2 expresses $R$ as the distance to the hotspot which is most likely pointed to by a camera view, by choosing the minimum subtended angle.

\[
R = \min_{x_i} \{d(x_i, a)\}
\] (2)

2.4 Keyframe Extraction

We analyze the visual similarity between frames and select the keyframes with sufficient content change.

2.4.1 Visual Similarity Measurement

We derive an overlap ratio of a projected line between two FOVs to approximate the visual similarity between two frames. To illustrate the analysis, let us consider the diagrams in Figure 2, with a reference FOV located at $A(0,0)$, directing its angle to North along the y-axis, and a target FOV whose location is at $B(x, y)(x \geq 0)$, offset by angle difference $γ$. The angle of view is $2 \times β$ and $R$ is the viewable distance. Our goal is to determine by how much FOV $B$ overlaps with FOV $A$.

In the first step we limit the problem to the case of $0 \leq γ \leq β$ as depicted in Figure 2. We denote the vertical ordinate of the intersection point of FOV $B$’s left ray and y-axis as $l$. For the case of $l \leq R$ (see Figure 2(a)), from $A$’s perspective the overlap ratio (simplified as a line) by $B$ is:

\[
sim(f_A, f_B) = \frac{R \tan β + (R - l) \tan(β - γ)}{2R \tan β}
\] (3)
were detected. The labeled hotspots "A, B, C, D, E, H, for the 71 videos, and the red points are the hotspots that

3.1 Hotspot Detection Results

we only present the detailed results for one video, namely

× size of each cell is about 20 × angular area which is separated into 100 per second. All FOVs of the test videos are within a rect-

the location and orientation sensor information is 1 sample

3.2 Effective Visible Distance Statistics

To evaluate our estimation of the effective visible distance, we manually collected ground truth data. For each video frame represented by the corresponding FOV, we found the most important object in the frame based on human perception, located the object in Google Earth, and then obtained its latitude and longitude. Afterwards, the distance between the camera location and the object was calculated as the ground truth effective visible distance. The frames in transition or containing ambiguous content (even users could not identify the most important object) were discarded. Figure 4 shows the comparison between the ground truth and the estimated effective visible distance for video v8636. Except for some miss estimations due to the incorrect understanding of the hotspots, the estimated distances match very well with the ground truth data. For instance, for the frame at time 40, the estimated distance is 712 m. Correspondingly, we can observe from the actual picture that the focus of the picture is Marina Bay Sands, which is very far away. For the frame at time 120, the focus is the Merlion, which is nearby. Accordingly, the estimated distance is 20 m.

3.3 Keyframe Extraction and Placement Results

The results for the proposed keyframe extraction algorithm are dependent on the threshold T, which determines the final number of keyframes that will be generated. The dependency of the number of keyframes on the threshold T is illustrated in Figure 6. If we set a relatively high threshold value, we might obtain a large number of keyframes with most of them being similar to each other. On the other hand, if we set a relatively small threshold value, the number of keyframes will be less and it is possible that some content between keyframes will be missed. Based on these observations and taking into consideration that mobile phone applications have battery constraints where similar keyframes should be avoided, we selected a threshold of 0.55 as a good compromise for the algorithm.
Figure 4: Comparison between the ground-truth and the estimated visible distance \( R \) for video v8636.

Figure 5: Selected keyframes of video v8636 for the proposed keyframe selection algorithm.

Figure 6: The number of keyframes as a function of the threshold \( T \) using video v8636.

Figure 5 shows the keyframe extraction result of our algorithm which successfully generates a compact keyframe set with adequate coverage for the original video and is computationally fast and efficient.

By applying hotspot detection, visible distance estimation and keyframe extraction, we can utilize a map-based user interface to present user-collected videos. The keyframe placement algorithm has been implemented in a demonstration system [5] which illustrates that when a user navigates along the video trajectory on the map, the corresponding thumbnails for the video keyframes are displayed at the estimated (far or near) locations.

4. CONCLUSIONS

In our study we presented an approach to detect hotspots and their distances from the camera in a fully automated way. We then leveraged that information to obtain an effective keyframe extraction method. The experimental results show that our technique is very effective for presenting user-generated outdoor videos on a map interface for easy browsing and viewing.

Acknowledgments

We would like to acknowledge that this research was carried out at the Centre of Social Media Innovations for Communities (COSMIC), sponsored and supported by the Singapore National Research Foundation and Interactive & Digital Media Program Office, MDA.

5. REFERENCES