Travel Video Scene Detection by Search

Wei-Ta Chu, Cheng-Jung Li, and Tsung-Che Lin
National Chung Cheng University, Chiayi, Taiwan
wtchu@cs.ccu.edu.tw, zoneli1987@gmail.com, congjhe@gmail.com

Abstract

We propose an approach to conduct video scene detection especially for travel videos captured by amateur photographers in journeys. The correlation between a travel video and its corresponding text-based travel schedule is discovered. Because scene boundaries are clearly defined in schedules, we segment videos into scenes by checking the discovered cross-media correlation. To make these two modalities comparable, photos related to the visited scenic spots are retrieved from image search engines, by the keywords extracted from text-based schedules. Sequences of keyframes and retrieved photos are represented as visual word histograms, and the problem of correlation determination is then transformed as an approximate sequence matching problem. The experimental results verify the effectiveness of the proposed idea, and show the promising research direction of utilizing cross-media correlation in media analysis.

1. Introduction

Going travel has been one of the most important activities in recent years. People treasure their travel experience, and get used to capture what they see or what they hear in journeys. With the popularity of low-cost and high-efficiency appliances, travelers can capture buildings, landmarks, or events at will, and therefore generate large amounts of digital multimedia data. These massive data obviously give rise to burden of access and browsing.

Among various types of travel media, large volumes of videos captured in journeys especially burden data access, and therefore draw the most challenging research issues. In this article, we focus on segmenting travel videos into semantics-related scenes. Video shots that were captured in the same scenic spot are claimed as in the same video scene. Although scene change detection has widely been studied in news, sports, movie, and TV programs, travel videos have much more severe visual conditions that make conventional scene detection techniques fail. For example, content in the same scenic spot is not always visually similar, which violates the assumption that visually similar shots should be grouped into the same scene. Moreover, travelers who don't specialize in photography may have large hand shake or bad lighting consideration, which cause motion blur or bad exposure for the captured videos.

As the challenges described above, simply analyzing visual content in videos may be insufficient to detect semantics-related scenes. Fortunately, many other data related to this journey would be easily obtained, such as photos captured in the same journey, pre-arranged text-based travel schedule, map, tour guides provided by the tourism bureau. All this information is tightly related to this journey, and therefore information between different modalities are correlated. Chu et al. [1] exploit this idea and conduct travel video scene detection by consulting the cross-media correlation between videos and photos captured in the same journey. They assume that travelers take both digital camcorders and cameras in journeys, and alternately capture travel experience in videos and photos. This assumption facilitates discovering cross-media correlation after videos and photos are transformed into the same representation.

Although the reported results in [1] are satisfactory, the assumption about simultaneously existence of videos and photos corresponding to the same journey is not always true. Motivated by the work in [2], we know that many related information can be retrieved from the internet, and the retrieved results (though they may be noisy) can be used to annotate or manage our own data. In [2], Wang et al. annotate images by discovering text descriptions in retrieved images, which are returned by image search engines based on text queries. With the similar idea, we investigate how to segment travel videos into scenes by discovering correlations between our own videos and the retrieved images, which are searched by the keywords extracted from text-based schedules.

We assume that travelers at least have the captured videos and the pre-arranged text schedule. The travel schedule states the scenic spots to be visited and the temporal order of visiting. The temporal order of scenes captured in videos is the same as scenic spots in the travel schedule. In this work, we first extract name entities of each scenic spot, and search each scenic spot's images from the web by text query. Sequences of keyframes extracted from videos and sequences of images retrieved from the web are then matched to determine their correspondence. After some post-processing, a shot is claimed to be in the scene of "Eiffel tower," for example, if its keyframes correspond to images retrieved from the text query "Eiffel tower."

Contributions of this work are summarized as follows:

- We transform the idea of "annotation by search" [2] into "video scene detection by search." This method explores cross-media correlation to facilitate media management.
- For approximately matching keyframe sequences and image sequences, we introduce an algorithm that is different from similar tasks proposed before. More flexible and practical solutions can be obtained.

The remainder of this paper is organized as follows. Section 2 gives an overview of the proposed system framework. The details of developed components are described in Section 3, including image search and the algorithm for finding correspondence between media. We provide evaluation results in Section 4, followed by the concluding remarks in Section 5.

2. Overview of system framework

Assume that we have a video captured in journeys and the text-based schedule corresponding to this journey. The idea of video scene detection is to explore the correlation between the video and the travel schedule, and then use the scene boundaries defined in the schedule to determine scene boundaries in the video. We transform this problem as a sequence matching problem, with the processes described as follows.

Figure 1 shows the proposed system framework. For the video, we first detect video shots and extract appropriate number of keyframes for each video shot by the global k-means algorithm [3]. Feature points such as scale-invariant feature transform (SIFT) [4] are extracted from each keyframe, and then quantized into visual words [5]. Statistics of visual words are collected to present each keyframe. Finally, the video is transformed into a sequence of keyframes, in the representation of visual word histograms, with the temporal order same as visiting.

For the travel schedule, we first extract name entities of visited scenic spots and then use them to retrieve related images from image search engines, such as Yahoo!, Google, and Flickr. Images related to each scenic spot are sorted in the order of visiting, and are respectively transformed into a sequence of visual word histograms, with the same procedure as that for video keyframes.

With the processes described above, we are able to find the correspondence between two modalities with the same representation. Because not all scenic spots were captured in videos and there are many noises in retrieved images, we conduct approximate sequence matching between them. With the discovered correspondence, keyframes that are matched with images retrieved by the same keyword are claimed to belong to the same video scene.

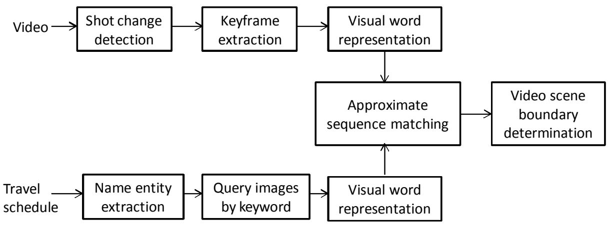

Figure 1. The proposed system framework.

3. Video scene detection

3.1. Video preprocessing

We first find shot boundaries based on color histogram difference between adjacent frames. Each video frame is described by a 16-bin HSV normalized histogram, in which 8 bins are for hue, and 4 bins are for saturation and value, respectively.

To efficiently represent each video shot, we adopt the approach proposed in [6], which automatically determines the most appropriate number of keyframes based on the global k-means algorithm [3]. Global k-means is an incremental deterministic clustering algorithm that iteratively performs k-means clustering while increasing k by one at each step. The clustering process proceeds until clustering results converge. By this algorithm, we overcome the initialization problem of conventional k-means algorithm, and adaptively determine appropriate number of clusters for each shot. Frames in a video shot are clustered into groups, and the frame closest to the centroid of each group is selected as a keyframe.

After extracting keyframes, we would like to filter out keyframes with severe blurred effects, which may damage the matching process later. Edge characteristics based on a wavelet-based method [7] are used to detect occurrence and extent of blur. In addition, illumination information is examined to detect overexposure or underexposure conditions. These processes not only reduce consumption time of determining cross-media correlations, but also eliminate influence of bad-quality images.

Due to uncontrolled environments in journeys, we have to represent data by features that resist to significant visual variations caused by bad photography skills and different settings of various capture devices. In this work, we characterize images by bag of visual words. We apply the difference-of-Gaussian (DoG) detector to detect feature points in keyframes and photos, and use the SIFT

(Scale-Invariant Feature Transform) descriptor to describe each feature point as a 128-dimensional vector [4]. SIFT-based feature vectors are then clustered by a k-means algorithm, and feature points quantized into the same cluster are claimed to belong to the same visual word. For a keyframe, each SIFT-based feature point is categorized as a visual word, and the distribution of visual words in a keyframe is described as a normalized visual word histogram. Therefore, we finally transform the sequence of keyframes into a sequence of normalized visual word histograms.

3.2. Query images by keyword

It's reasonable to assume that travelers have a predefined travel schedule before traveling. The schedule describes where to visit and the order of visiting. Travelers sequentially visit and capture videos, and thus the temporal order of video content is the same as the visited scenic spots. Therefore, the travel video is temporally correlated to the text-based travel schedule.

Because the boundaries between scenic spots in the travel schedule are well defined, we would like to exploit the information to facilitate video scene detection. To find correspondence between these two modalities, we have to transform the text-based schedule into a representation same as the video.

We first extract the name of each scenic spot defined in the schedule, which is then used as a keyword to query related images/photos. In our work, we conduct keyword-based image search in Yahoo! and Google image search engines, and retrieve a few top-ranked images/photos. In addition to the search engines that index images by surrounding text, we also experiment on images retrieved from Flickr, which are indexed by tags provided by users.

Assume that there are V scenic spots to be visited, and the name entities corresponding to these scenic spots are $(k_1 k_2 ... k_V)$, which are temporally sorted, i.e. k_i was visited before k_j if $i < j$. Each entity is used as a keyword to search related photos from image search engines. Similar to video keyframes described above, we extract SIFT feature points from each retrieved photo and then quantize them into visual words. The distribution of visual words in a retrieved photo is described as a normalized visual word histogram. Therefore, we again transform the sequence of retrieved photos into a sequence of normalized visual word histograms. Let's denote the sequence as $X = (\boldsymbol{x}_1 \boldsymbol{x}_2 ... \boldsymbol{x}_m)$, in which $\boldsymbol{x}_i$ denotes the visual word histogram of the ith retrieved photo. There are totally m photos, from the results of retrieving the top-ranked q photos for each keyword, i.e. $m = V \times q$. Two subsequences $S_{k_1} = (\boldsymbol{x}_1 \boldsymbol{x}_2 ... \boldsymbol{x}_q)$ and $S_{k_2} = (\boldsymbol{x}_{q+1} \boldsymbol{x}_{q+2} ... \boldsymbol{x}_{2q})$ correspond to two scenic spots, while the photos in S_{k_1} represent the scenic spot visited before S_{k_2}. Although there is an implicit temporal order between S_{k_1} and S_{k_2} (corresponding to scenic spots k_1 and k_2 in the travel schedule), there is no such relation between photos in the same subsequence, e.g. no special temporal order exists between $\boldsymbol{x}_1$ and $\boldsymbol{x}_q$ in S_{k_1}.

3.3. Maximum-sum segment

Finding correlations between videos and photos retrieved from search engines has been transformed into a sequence matching problem. Generally, the dynamic programming strategy can be used to conduct approximate sequence matching, such as the longest common subsequence problem (LCS). However, photos retrieved by keywords are just "semi-temporally ordered." Although photos related to different keywords are temporally sorted, that related to the same keyword don't follow any specific temporal order. This characteristic destroys the sequential property necessary for the LCS algorithm. In addition, there may be many irrelevant photos in the retrieved data, which makes correlation determination more challenging.

There are two visual word histogram sequences, $X = (\boldsymbol{x}_1 \boldsymbol{x}_2 ... \boldsymbol{x}_m)$ and $Y = (\boldsymbol{y}_1 \boldsymbol{y}_2 ... \boldsymbol{y}_n)$, which respectively corresponds to the retrieved photos and keyframes. The photo sequence X is semi-temporally ordered, i.e. $X = (S_{k_1} S_{k_2} ... S_{k_V})$, where $S_{k_i} = (\boldsymbol{x}_\ell \boldsymbol{x}_{\ell+1} ...)$ consists of photos retrieved from the keyword k_i. The retrieved photos in S_{k_j} are conceptually taken behind that in S_{k_i} if $i < j$, but photos $(\boldsymbol{x}_\ell \boldsymbol{x}_{\ell+1} ...)$ in S_{k_i} are not temporally ordered. With this characteristic, we formulate the correlation determination process as a variation of the maximum-sum segment problem [8]. To find the optimal correspondence between keyframes and a specific photo set S_{k_i}, the goal is to find a segment $Y(p_i, q_i) = (\boldsymbol{y}_{p_i} ... \boldsymbol{y}_{q_i})$ from Y such that the segment $Y(p_i, q_i)$ of the longest length contains similar content as that in S_{k_i}, where $p_i = 1, ..., n-1$, $q_i = 2, ..., n$, and $p_i < q_i$. In addition, the segment $Y(p_i, q_i)$ corresponding to S_{k_i} should be ranked before the segment $Y(p_j, q_j)$ corresponding to S_{k_j} if $i < j$.

To find the segment $Y(p_i, q_i)$ corresponding to the scene $S_{k_i} = (\boldsymbol{x}_\ell \boldsymbol{x}_{\ell+1} ...)$, we first transform the sequence $Y = (\boldsymbol{y}_1 \boldsymbol{y}_2 ... \boldsymbol{y}_n)$ into a real number sequence $Z = (z_1 z_2 ... z_n)$ as follows. Based on the visual word histogram intersection between $\boldsymbol{y}_j$ and $\boldsymbol{x}_l$, denoted by $I(\boldsymbol{y}_j, \boldsymbol{x}_l)$, we first calculate the similarity z'_j between $\boldsymbol{y}_j$ and $\boldsymbol{x}_l$ in S_{k_i}:

$$z'_j = I(\boldsymbol{y}_j, \boldsymbol{x}_{l^*}) \text{ and } l^* = \arg\max_l I(\boldsymbol{y}_j, \boldsymbol{x}_l), \quad (1)$$

where, $l = \ell, \ell+1, ..., \ell + |S_{k_i}| - 1$. The value $|S_{k_i}|$ denotes the number of retrieved photos in this scene.

After mean removing, we obtain

$$z_j = z'_j - \frac{1}{n} \sum_j z'_j. \quad (2)$$

Note that the sequence Z may contain both negative and positive real numbers.

Corresponding to the scene S_{k_i}, we would like to find an interval $[p_i, q_i]$ in Z, $L_i \leq p_i \leq q_i \leq U_i$, such that $Z(p_i, q_i) = (z_{p_i} \ldots z_{q_i})$ is the maximum-sum segment of $Z(L_i, U_i)$, i.e. $\sum_{h=p_i}^{q_i} z_h$ is maximal in all cases in $Z(L_i, U_i)$. The values L_i and U_i respectively denotes the lower and upper bounds for searching the maximum-sum segment, and as a consequence they are used to constrain that the maximum-sum segment corresponding to S_{k_i} should appear before that corresponding to S_{k_j} if $i < j$. To this end, we set the search interval as:

$$L_i = \max(0, n \times \frac{i-2}{V}) \text{ and } U_i = \min(n, n \times \frac{i+2}{V}). \quad (3)$$

The value V is the number of visited scenic spots, i.e. the number of groups of photos retrieved by keywords. Note that the search intervals for successive scenic spots are overlapped. Because travelers may not equally capture content of the same length for different scenic spots, the search interval for each scenic spot is designed to be three times larger than the proportion it corresponds to.

The aforementioned problem can be viewed as a range maximum-sum segment query (RMSQ) problem [8], which is able to be solved by a linear time algorithm. In this work, we apply the algorithm proposed by Chen and Chao [8] to find correspondence between a subsequence in $Y = (\boldsymbol{y}_1 \boldsymbol{y}_2 \ldots \boldsymbol{y}_n)$ and the photos retrieved by a keyword.

Note again that photos in S_{k_i} are not temporally ordered. Therefore, although the keyframes in Y are temporally ordered, we cannot adopt the well-known LCS algorithm to conduct approximate sequence matching. Moreover, the LCS algorithm finds the global optimal matching between two sequences. We cannot control the quality of retrieval, however, and thus many irrelevant photos are in S_{k_i}. Strictly finding the global matching between retrieved photos and keyframes is not reasonable, and the matching result may be disturbed by noises.

3.4. Video scene boundary determination

After determining the correspondence, keyframes in the selected maximum-sum segment are assigned a scene label according to the corresponding photos. Because boundaries of scenic spots have been defined in the travel schedule, we can accordingly estimate scene boundaries in videos. For example, if we find that the scenic spot S_{k_i} corresponds to some keyframes in the representation of visual word histograms $(\boldsymbol{y}_{p_i}, \boldsymbol{y}_{p_i+1}, \ldots, \boldsymbol{y}_{q_i})$, these keyframes are then assigned as in the ith scenic spot.

Note that lengths of max-sum segments corresponding to different scenic spots may be varied. Moreover, because the search intervals for successive scenic spots are overlapped (see Equation (3)), the max-sum segments corresponding to different scenic spots may be overlapped. To handle this problem, we especially examine max-sum segments for any two successive scenic spots. Figure 2 illustrates three possible cases.

Figure 2(a) shows the simplest case, in which two max-sum segments for successive scenic spots are not overlapped. Keyframes $\boldsymbol{y}_{p_i}, \ldots, \boldsymbol{y}_{q_i}$ are assigned as in the ith scenic spot, and keyframes $\boldsymbol{y}_{p_{i+1}}, \ldots, \boldsymbol{y}_{q_{i+1}}$ are assigned as in the $(i+1)$-th scenic spot. For those keyframes in-between q_i and p_{i+1}, the first $(U_i - L_{i+1}) \times \frac{q_i - p_i}{q_{i+1} - p_{i+1}}$ keyframes are assigned as in the ith scene, and the remaining keyframes are assigned to the $(i+1)$-th scene.

If two max-sum segments are overlapped as in Figure 2(b), the keyframes from $\boldsymbol{y}_{p_i}$ to $\boldsymbol{y}_c$ are assigned to the ith scene, where $c = \frac{p_{i+1}+q_i}{2}$. In the case of Figure 2(c), the keyframes from $\boldsymbol{y}_{p_i}$ to $\boldsymbol{y}_{p_{i+1}}$ are assigned to the ith scene, while $\boldsymbol{y}_{p_{i+1}+1}$ to $\boldsymbol{y}_{q_i}$ are assigned to the $(i+1)$-th scene. In the case of Figure 2(d), the keyframes from $\boldsymbol{y}_{p_{i+1}}$ to $\boldsymbol{y}_{q_i}$ are assigned to the ith scene, while $\boldsymbol{y}_{q_i+1}$ to $\boldsymbol{y}_{q_{i+1}}$ are assigned to the $(i+1)$-th scene.

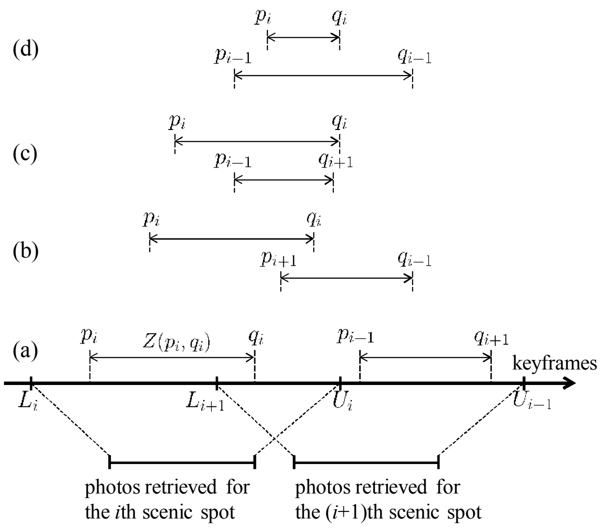

Figure 2. Illustrations of different situations in results of finding max-sum segments.

Table 1. Information of the evaluation dataset.

	# visited scenes	length	# keyframes
Video 1	6	12:58	227
Video 2	4	15:07	153
Video 3	5	08:29	98
Video 4	4	11:03	176
Video 5	3	16:29	136
Video 6	2	05:34	67
Video 7	6	15:18	227

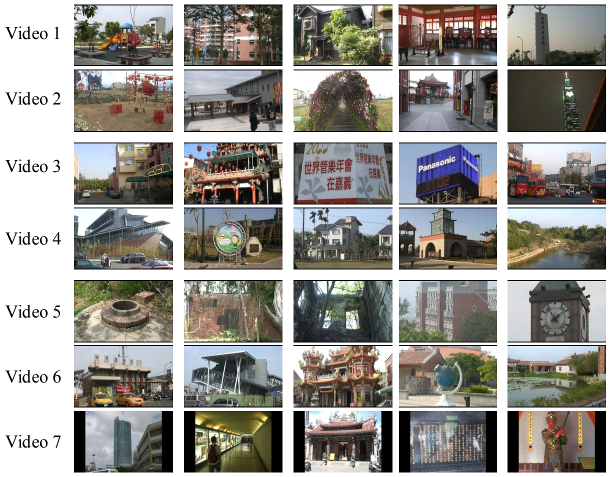

Figure 3. Some snapshots of the evaluated videos.

4. Evaluation

4.1. Evaluation dataset and performance metric

The evaluation dataset includes seven videos captured in different amateur photographers' journeys, and seven text-based travel schedules. Length of each video ranges from five to sixteen minutes, and each video is encoded as in MPEG-1 format with resolution 480×272. Figure 3 shows some snapshots of scenes in the each video. Table 1 shows the information of scenes, keyframes, and length of each travel video. There are totally 30 different visited scenic spots in the evaluation dataset.

According to the travel schedule, we respectively retrieve 18 top-ranked photos from Google and Yahoo! image search engines for each scenic spot. Photos retrieved from two search engines are combined, and there are totally 1011 photos for 30 scenic spots after eliminating some results that cannot be successfully downloaded. For each scenic spot, we also retrieve 36 top-ranked photos from Flickr, which indexes photos by tags provided by users. There are totally 1021 photos retrieved from Flickr, after eliminating some results that cannot be successfully downloaded. Data from "Google and Yahoo!" and "Flickr" are experimented separately to investigate how our proposed method works on photos retrieved by different scenarios. Because resolutions of the retrieved photos are varied, we normalize them into 400×300 for the efficiency of feature extraction and visual word construction.

To evaluate performance of scene detection, we consider overlaps between detected video scenes and ground truths, in terms of purity [9]. Given the ground truth of scenes $S = \{(s_1, \Delta t_1), ..., (s_{Ng}, \Delta t_{Ng})\}$ and the results of scene detection $S^* = \{(s_1^*, \Delta t_1^*), ..., (s_{Nv}^*, \Delta t_{Nv}^*)\}$, a purity value ρ is defined as

$$\rho = \left(\sum_{i=1}^{Ng} \frac{\tau(s_i)}{T} \sum_{j=1}^{Nv} \frac{\tau^2(s_i, s_j^*)}{\tau^2(s_i)} \right) \times \left(\sum_{j=1}^{Nv} \frac{\tau(s_j^*)}{T} \sum_{i=1}^{Ng} \frac{\tau^2(s_i, s_j^*)}{\tau^2(s_j^*)} \right), \quad (4)$$

where $\tau(s_i, s_j^*)$ is the length of overlap between the scene s_i and s_j^*, $\tau(s_i)$ is the length of the scene s_i, and T is the total length of all scenes. In this equation, the first term indicates the fraction of the current evaluated scene, and the second term indicates how much a given scene is split into smaller scenes. The purity value ranges from 0 to 1, and a larger purity value means a better result. In this work, length of a scene, i.e. Δt_i and Δt_i^*, is represented by the number of keyframe.

4.2. Performance comparison and discussion

We conduct three experiments to evaluate the proposed idea:

- Exp 1: Based on the photos retrieved from Google and Yahoo!, using the max-sum segment algorithm to find correspondence and determine scene boundaries accordingly.
- Exp 2: Based on the photos retrieved from Flickr, using the max-sum segment algorithm to find correspondence and determine scene boundaries accordingly.
- Exp 3: According to the number of visited scenic spots, temporally sorted keyframes are equally divided into several groups, and keyframes in the same group are assigned as in the same video scene.

In the first two experiments, we discover correlation between videos and the corresponding travel schedules in terms of temporal and visual characteristics, by the max-sum segment algorithm. In Exp 3, only the temporal order of visited scenic spots is used to define video scene boundaries.

Table 2 shows purity values of video scene detection in three experiments. It's not surprising that performance varies for different datasets. All the following factors may affect detection performance.

Visual quality of travel videos: Features extracted from keyframes with bad visual quality constitute visual word histograms with less reliability, and therefore performance of sequence matching is degraded. In travel videos, motion blur is the main factor of quality degradation. Videos 2 and 4 convey large amount of motion, and generally have worse performance in all three methods.

Popularity of visited scenic spots: If the visited scenic spots are popular, more related photos can be retrieved and ranked first by image search engines. We cannot retrieve enough related photos from Google and Yahoo! for Videos

1, 4, and 5. On the other hand, we can find a few photos that are highly related to the visited scenic spots from Flickr.

Retrieval performance of search engines: Although it's hard to measure retrieval performance of different search engines, accuracy of keyword-based image retrieval directly affect the reliability of correlation determination. Relative to Exp 1, we obtain better performance from Exp 2 in Videos 1, 4, and 5, because more accurate photos can be retrieved from Flickr (due to more accurate tags provided by users). In these cases, we are able to discover more accurate correlation between video keyframes and retrieved photos. On the other hand, retrieval based on user's tags is not always good. For the visited scenic spots corresponding to Videos 3 and 7, more related photos are retrieved from Google and Yahoo! image search, and we can see their superior performance.

User's capturing habits: The naïve approach has the worst performance because no visual correlation is considered in this method. Actually, its performance depends on user's capturing habits. If the traveler equally captures content in every scenic spot, the naïve approach may achieve satisfactory performance. Bad visual quality and less popularity for Video 4, and less correlation between user's photos and retrieved images for Videos 3 and 6, cause that the naïve approach achieves higher purity values than our method.

Overall, the Exp 1 provides the best performance, though the difference between it and Exp 2 is very limited. Although it may be expected that Flickr would provide more accurate search results and therefore derive more accurate correlation, the travel videos captured by amateur photographers may not contain the most popular buildings or landmarks that would be returned as the top results of Flickr.

Name ambiguity would be another problem. The retrieved results of "Arc of Triumph" and "Arch of Triumph" may be different. These effects would be more severe in specific scenic spots that have different nicknames, or in some languages such as Chinese that may indicate the same place by many different names.

Table 2. Performance of video scene detection in terms of purity.

	V1	V2	V3	V4	V5	V6	V7	**Avg.**
Exp 1	0.66	0.52	0.91	0.48	0.80	0.59	0.62	**0.654**
Exp 2	0.77	0.50	0.68	0.61	1	0.59	0.41	**0.651**
Exp 3	0.21	0.62	0.78	0.78	0.49	0.80	0.45	**0.59**

5. Conclusion

We have presented a video scene detection method that focuses on travel videos and specially considers characteristics of information related to journeys. Instead of simply analyzing visual content in videos, we discover temporal and visual correlation between travel videos and their corresponding travel schedules. We search photos related to scenic spots from image search engines, by the name entities of visited scenic spots extracted from the text-based schedules. Correlation between video keyframes and retrieved photos is then determined by the max-sum segment algorithm. Because scene boundaries have been clearly defined in travel schedules, scene boundaries in the keyframe sequence can be determined by checking the discovered cross-media correlation. The experimental results verify the effectiveness of the proposed method. To the best of our knowledge, this work would be one of the first studies to exploit general-purpose image search engines in segmenting user's own videos.

6. Acknowledgement

This work was partially supported by the National Science Council of the Republic of China under grants NSC 98-2221-E-194-056.

References

[1] W.-T. Chu, C.-C. Lin, and J.-Y. Yu. Using cross-media correlation for scene detection in travel videos. In Proc. of ACM International Conference on Image and Video Retrieval, 2009.

[2] X.-J. Wang, L. Zhang, F. Jing, and W.-Y. Ma. Annosearch: image auto-annotation by search. In Proc. of IEEE Computer Society Conference on Computer Vision and Pattern Recognition, vol. 2, pp. 1483-1490, 2006.

[3] A. Likas, N. Vlassis, and J.J. Verbeek. The global k-means clustering algorithm. Pattern Recognition, vol. 36, pp. 451-461, 2003.

[4] D. Lowe. Distinctive image features from scale-invariant keypoints. International Journal of Computer Vision, 60, 2, pp. 91-110, 2004.

[5] J. Sivic and A. Zisserman. Efficient video search for objects in videos. Proceedings of the IEEE, 96, 4, pp. 548-566, 2008.

[6] V.T. Chasanis, A.C. Likas, and N.P. Galatsanos. Scene detection in videos using shot clustering and sequence alignment. IEEE Transactions on Multimedia, vol. 11, no. 1, pp. 89-100, 2009.

[7] H. Tong, M. Li, H.-J. Zhang, and C. Zhang. Blur detection for digital images using wavelet transform. In Proc. of IEEE International Conference on Multimedia & Expo, pp. 17-20, 2004.

[8] K.-Y. Chen and K.-M. Chao. On the range maximum-sum segment query problem. Discrete Applied Mathematics, vol. 155, no. 16, pp. 2043-2052, 2007.

[9] A. Vinciarelli and S. Favre. Broadcast news story segmentation using social network analysis and hidden Markov models. In Proc. of ACM Multimedia, pp. 261-264, 2007.