# Somebody Helps Me: Travel Video Scene Detection Using Web-based Context

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

## Abstract

We conduct video scene detection with the aids of web-based context, especially for travel videos captured by amateur photographers in journeys. Correlations between personal videos and predefined travel schedules, which are used to retrieve related data from general-purpose image/video search engines, are discovered. Because scene boundaries are clearly defined in travel schedules, we segment videos into scenes by checking the discovered cross-media correlation. To make different modalities comparable, keyframes extracted from videos and images retrieved from web are represented by visual word histograms, and the problem of correlation determination is then transformed as an approximate sequence matching problem. We prioritize different visual words according to statistics of retrieved data, and evaluate similarity between images based on the weighting scheme. To systematically determine scene boundaries after finding cross-media correlation, we introduce an energy minimization framework to jointly consider visual, temporal, and context information. Experimental results verify the effectiveness of the proposed idea, and show that it’s promising to utilize cross-media correlation and web-based context in media analysis.

**Keyword:** video scene detection; web-based context; approximate sequence matching; maximum-sum segment; energy minimization

## 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 generate large amounts of digital multimedia data. Massive data, therefore, draw urgent demands for efficient access and management functions.

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 [15], sports, movie, and TV programs [13][14], 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 are 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 in captured videos.

Because of the challenges described above, simply analyzing visual content in videos may be insufficient to detect semantics-related scenes. Fortunately, context information such as photos captured in the same journey and pre-arranged text-based travel schedules, which are is tightly related to this journey, can provide insights to facilitate cross-media analysis and management. In [1], we proposed this idea and conduct travel video scene detection by consulting cross-media correlations between videos and photos captured in the same journey. We assumed that travelers take both digital camcorders and cameras in journeys, and alternately capture travel experience in videos and photos. Cross-media correlation between them is discovered after they are transformed into the same representation.

The assumption of simultaneous existence of videos and photos corresponding to the same journey is not always true. Nowadays, from the web we may be able to find any information related to a specific query, which may be shared by somebody we don’t know. Based on the ideas in [1] and [2], we can retrieve data that are related to the visited scenic spots, find cross-media correlation between web-based context and our own travel videos, and then segment our own videos into semantics-related scenes. In our previous work [10], we assumed that travelers only have the captured videos and a pre-arranged text schedule, which 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 that in the travel schedule. Name entities of visited scenic spots are extracted from the schedule, and are used to search related images from web-based image search engines. A sequence of keyframes extracted from the user’s travel video and a sequence 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.” Thanks to somebodies who share their images relevant to our visited scenic spots, the developed system gains extra leverage from the largest database (the web) to conduct video scene detection [10].

Although we have verified the idea of utilizing web-based context to analyze our personal data, data retrieved from the web are very noisy, and many factors influence the detection performance. We sum up related issues as follows, which were originally described in the discussion section of [10], and propose new techniques to address them as contributions of the current article.

1) *Visual quality of travel videos:* Features extracted from keyframes with bad visual quality constitute visual word histograms of less reliability, and therefore performance of sequence matching is degraded.
2) *Popularity of visited scenic spots:* If the visited scenic spots are not popular, few related photos can be retrieved from the top-ranked results of image search engines.
3) *Retrieval performance of search engines:* Although it's hard to measure retrieval performance of different search engines, accuracy of keyword-based image/video retrieval directly affect the reliability of correlation determination.

In this work, we propose an adaptive weighting scheme to emphasize discriminative features and thus define a more reliable distance metric. This technique addresses the issues 1 and 3 above. For issue 1, we further enhance the preprocess step by a state-of-the-art blur detection module, so that more blurred keyframes extracted from the travel videos can be discarded. Especially for issue 2, we investigate how volume of web-based data affects the final performance. Moreover, we further include videos retrieved from Youtube, and investigate how different types of web-based context affect the proposed system. At the last but not the least, to make scene boundary determination process more systematic, we model it as a binary labeling problem and find the optimal solution by a graph cut algorithm [12]. This formulation eliminates the heuristic rules defined in [10].

The remainder of this paper is organized as follows. Section 2 gives literature survey on video scene detection. An overview of the proposed system framework is described in Section 3. Section 4 provides details of the developed components, including preprocessing, the adaptive weighting scheme, the algorithm for finding correspondence between media, and the algorithm for determining scene boundaries. We provide evaluation results in Section 5, followed by the concluding remarks in Section 6.

## 2. Related Works

To make literature survey focused, we start surveying from home video, which is a superset of travel video. The difference between general home videos and travel videos is described in the end of Section 2.1. Section 2.2 provides surveys on video scene detection.

### 2.1. Home Video Analysis

Because there is no benchmark and evaluation metric for home video analysis, studies in this field are diverse and rise from different perspectives. Although there is no conventional rules in capturing home videos, Gatica-Perez et al. [16] cluster video shots based on visual similarity, duration and temporal adjacency, and accordingly find hierarchical structure of videos. On the basis of motion information, Pan and Ngo [17] decompose videos into snippets, which are then used to index home videos. For the purpose of automatic editing, temporal structure and music information are extracted, and subsets of video shots are selected to generate highlights [18] or MTV-style summaries [19]. Recently, Peng et al. [20] take media aesthetics and editing theory into account, and develop a new human-computer interface to facilitate home video skimming. In [21], a system called Hyper-Hitchcock is developed to semi-automatically edit videos and equip hyperlink properties. From the perspective of intention analysis, [22] and [23] model user intention for video repurposing and browsing.

While there is rich literature considering motion and visual characteristics in home videos, fewer studies have been proposed to handle a specific sub-category of home videos, and elaborately exploit related domain knowledge. The work by Cheng et al. [24] provides an example on this direction, in which they take knowledge of wedding customs and develop segmentation and event recognition modules. While wedding videos convey one of the treasured moments in our lives, amounts of such data are largely less than that captured in journeys. Although a few studies were proposed on travel videos, most of them, unfortunately, leave out unique and useful characteristics in such media.

Different from other home videos, travel videos have special characteristics that may be conducive or cumbersome to practical technique development: 1) According to a pre-arranged travel schedule, travelers visit scenic spots and capture photos/videos sequentially. 2) Content captured at the same scenic spot would have significantly different appearances, which destroys conventional methods for image clustering or video scene detection. 3) Scenic spots are visited sequentially, and various media are taken alternately or simultaneously in the same temporal order. Different media may thus be correlated. With the above characteristics, we design a system that specially analyzes travel videos.

### 2.2. Video Scene Detection

For video scene detection, Yeung and Yeo [14] propose a classical work called scene transition graph to describe relationships between video shots, and achieve scene detection by analyzing links in the graph. For movies, Hanjalic et al. [25] investigate context between video shots based on keyframes represented by DC images, and determine boundaries of logical story units such as dialogue and action scenes. Sundaram and Chang [26] take film-making rules and psychology of audition into account to build a computational scene model, which mimics characteristics of human's short-term and long-term memory. Rasheed and Shah [13] develop a two-pass algorithm based on motion, shot length, and color properties, to find semantics-related scenes in movies and TV shows. More recently, Chasanis et al. [6] estimate appropriate number of keyframes for each video shot based on a spectral clustering approach, and then determine scene boundaries by sequence alignment techniques. Due to significance of video scene detection, integrated framework such as [28] and systematic evaluation method such as [30] have been proposed for years. In addition, the TRECVID benchmark [29] also issues the "story segmentation" task, while it focuses only on news and TV programs.

### 2.3. With the Leverage of Other Context

It's a consensus that simultaneously analyzing and fusing multiple modalities achieves more promising performance. However, few studies have been conducted to investigate relationships between media that are seemingly irrelevant at first, but actually are subtly related after careful consideration. Takeuchi and Sugimoto [31] propose an interesting home video summarization system that infers user's preference from his auxiliary photo collections, rather than from the video itself. Users may be interested in capturing the same things in the same style, which inspires the user-adaptive cross-media summarization system. Wang et al. [2] search semantically and visually similar images from the web and mine annotations from them to annotate our own images. This work further inspires us to view the web as the richest database. More recently, Vallet et al. [32] exploit external resources to facilitate identification of query semantics, and thus improve video retrieval performance. Wang et al. [33] utilize diverse set of data with different properties to boost video classification accuracy. Classifiers trained from different data sources are elaborately fused to categorize wild web videos. In our work, thanks to the subtle correlations between web-based context and our personal data, a system is developed to automatically analyze our personal data. The major difference between our work and [2] is that their image database is actually well defined, but we would suffer from many noises retrieved from the web.

## 3. System Framework

Assume that we have a video captured in a journey and a text-based schedule corresponding to this journey. According to name entities stored in the schedule, we search related images or videos from the web, which may be shared by someone else who visited the scenic spots as in our travel videos. Correlations between the retrieved data and our travel videos are then discovered to determine scene boundaries in our travel video. Determining correlation between different modalities is formulated as a sequence matching problem, while one sequence is transformed from web-based context and another sequence is transformed from our own travel video.

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]. Keyframes with degraded visual quality due to motion blur or over/under exposure are filtered out by a quality assessment module. Feature points such as scale-invariant feature transform (SIFT) [4] are extracted from each keyframe, and then quantized into visual words [5] to capture concepts in visual appearance. Statistics of visual words are collected as a visual word histogram to represent each keyframe. Finally, the video is transformed into a sequence of visual word histograms, with the temporal order same as visiting.

For the travel schedule, name entities of visited scenic spots in the text-based schedule are first extracted. Related images and videos shared on the web are then retrieved by a query-by-keyword scheme from image search engines, such as Yahoo!, Google, and Flickr, and from video search engines, such as Youtube and viemo. Images or keyframes extracted from retrieved videos are temporally sorted in the order of visiting, and are respectively transformed into a sequence of visual word histograms, with the same procedure as that for travel video keyframes.

With the processes described above, we are able to determine correspondence between media with the same representation. Note that there are many noises in our travel videos, and a portion of retrieved data is not truly related to visited scenic spots. Therefore, we propose a distance metric with an adaptive weighting scheme to evaluate similarity between our own data and that from the web. The maximum-sum segment algorithm [8] is then applied to conduct approximate sequence matching for two sequences. With the discovered correspondence, keyframes that are matched with data retrieved by the same name entity are claimed to belong to the same video scene.

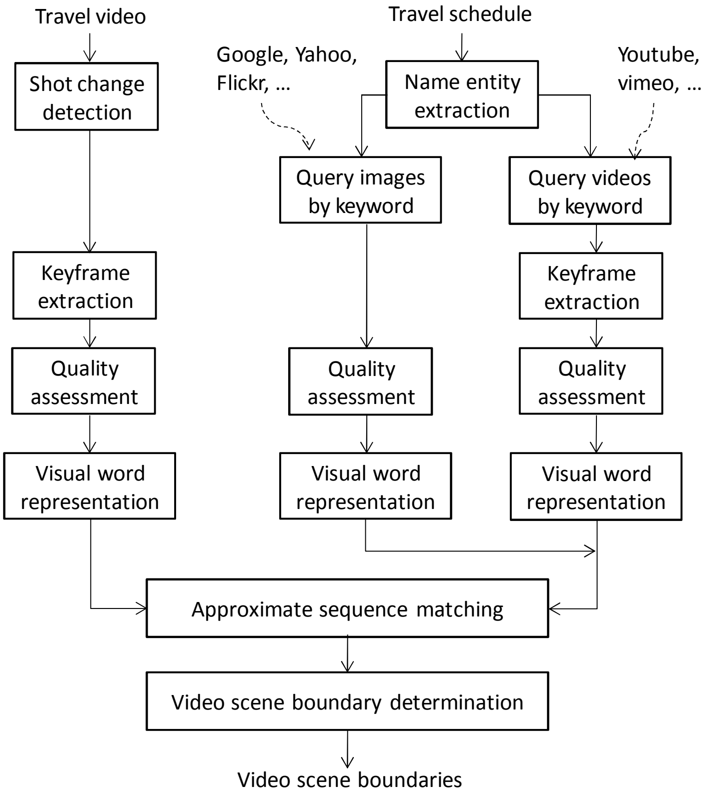


Figure 1. The proposed system framework.

## 4. Video Scene Detection

### 4.1 Video Preprocessing

We first find shot boundaries based on color histogram difference between adjacent frames. Each video frame is described by a normalized HSV color 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 blur or keyframes with insipid content, which may damage the matching process later. Figures 2(a) and 2(b) show blurred keyframes, in which high frequency information is lost often due to handshaking. In this work, we adopt CPBD (cumulative probability of blur detection) [7] to estimate extent of blur. Each keyframe is first divided into

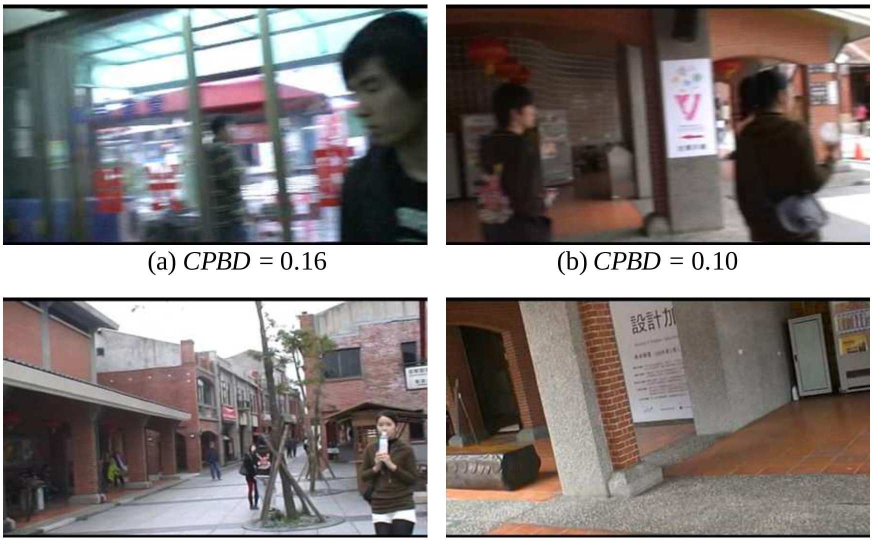
$64 \times 64$ blocks, and edge information is extracted from each block. For an edge block, the probability of blur detection is modeled as

$$P_{blur} = 1 - \exp\left(-\left|\frac{w(e_i)}{w_{JNB}(e_i)}\right|^{\beta}\right), \tag{1}$$

where $w(e_i)$ is the width of the edge $e_i$ in this block, and $w_{JNB}(e_i)$ is the "just noticeable blur" (JNB) edge width that depends on local contrast and is measured by psychological experiments. This metric evaluates the cumulative probability of image blocks that have blur extent lower than "just noticeable blur":

$$CPBD = P(P_{blur} \leq P_{JNB}) = \sum_{P_{blur}=0}^{P_{blur}=P_{JNB}} P(P_{blur}). \tag{2}$$

If many blocks in the keyframe are blurred, many of them have blur extent larger than JNB, and thus the corresponding *CPBD* value is smaller. For example, the *CPBD* values are smaller for Figures 2(a) and 2(b). Without camera and object motion, Figure 2(d) has the largest *CPBD* value. Although there is no camera motion in Figure 2(c), parts of blocks with object motion are detected as blur, and thus the *CPBD* value is in-between. In this work, we filter out keyframes with *CPBD* values lower than 0.2. Details of parameter settings please refer to [7].



(a) *CPBD* = 0.16

(b) *CPBD* = 0.10

(c) *CPBD* = 0.53

(d) *CPBD* = 0.73

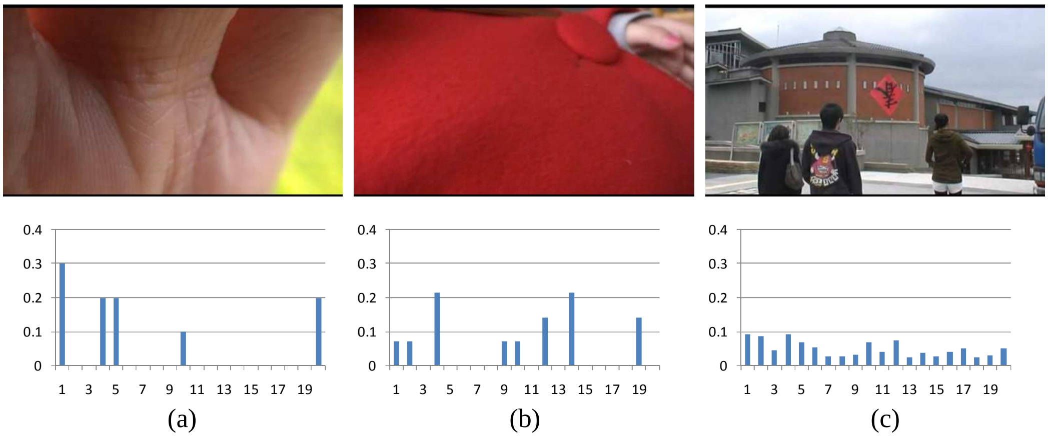
Figure 2. Examples of blur detection for keyframes.

After filtering out blurred keyframes, 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, in which "images" generally denote keyframes extracted from our own data or photos retrieved from the web. 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. We finally transform the sequence of keyframes into a sequence of normalized visual word histograms.

In addition to filter out blurred keyframe, we would like to discard the ones that are “meaningless.” Figures 3(a) and 3(b) shows two meaningless keyframes. Figure 3(a) was captured because the photographer wanted to hold the camera tightly in walking, and his hand accidentally occluded the lens. Recall that we focus on travel videos captured by amateur photographers, and such photography errors are not rare. Figure 3(b) shows another extremely bad case in which the view was aslant occupied by someone else’s clothing. The bottom row of Figure 3 shows the corresponding visual word histogram of the images in the top row. Because large portion of feature points in Figures 3(a) and 3(b) present the same content and are quantized into the same visual words, only some specific bins of the histogram have large values. To detect keyframes largely occupied by nonsense content, we evaluate each keyframe’s entropy based on its corresponding visual word histogram:

$$H_i = -\sum_{j=1}^{N} p_j \log p_j, \tag{3}$$

where $p_j$ denotes value of the $j$th bin of the visual word histogram, and $N$ is the number of visual words used to represent images. Keyframes that have entropy values lower than a threshold are discarded. To sum up, blur detection and insipid image detection not only reduces consumption time of determining cross-media correlations in the following sections, but also eliminates influence of bad-quality images.



(a) (b) (c)

Figure 3. Examples of insipid keyframe detection. Top row: original keyframes; bottom row: the corresponding visual word histograms.

## 4.2 Query Web-based Context 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. The travel video is therefore temporally correlated to the text-based travel schedule. Because 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 modalities, we have to transform the text-based schedule into a representation same as the video.

Information shared on the web serves as the largest database in the world and provides clues for transforming text-based schedules into visual appearance. We imagine that somebodies visited the same places as us, and they captured and shared data on the web. If we are able to retrieve these data and find correlations between theirs and ours, we may segment our travel videos with the help from somebodies we don't know. To implement this idea, we first extract name entities of scenic spots defined in the schedule, which are then used as keywords to query related images/videos on the web. Two types of web-based resources are investigated: images retrieved by general-purpose image search engines such as Yahoo!, Google, and Flickr, and videos retrieved by the largest video sharing platform, i.e. Youtube. Note that image search in Yahoo! and Google is different from that in Flickr. The former indexes images by surrounding text, while Flickr indexes images by tags especially provided by users. We include varied web-based contexts to investigate how resources presented in different modalities and retrieved from different platforms affect the proposed idea.

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 data from the web. Although name entity extraction has been studied for years, detailed name entity extraction techniques are beyond the scope of this paper. In this work, we assume that text-based schedules are well defined, and names of scenic spots are extracted manually, without losing novelty of the proposed ideas.

For the images searched from Yahoo!, Google, and Flickr, and keyframes of videos searched from Youtube (by the same method described in Section 4.1), we describe them as normalized visual word histograms as well. We again transform the sequence of retrieved data 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 $i$th retrieved photo/keyframe. For each keyword, we retrieve the top-ranked $q$ photos from an image search engine, or extract $q$ keyframes from the top-ranked videos from Youtube, 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, and the images in $S_{k_1}$ represent the scenic spot visited before $S_{k_2}$. In the case that $S_{k_i}$ represents photos retrieved from search engines, 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 images in the same subsequence, e.g. no special temporal order exists between $\boldsymbol{x}_1$ and $\boldsymbol{x}_q$ in $S_{k_1}$. In the case that $S_{k_i}$ represents keyframes extracted from retrieved videos, the temporal order from $\boldsymbol{x}_1$ to $\boldsymbol{x}_q$ is not necessarily the same as the visiting order in our own travel videos. Therefore, the web-context sequence $X = (\boldsymbol{x}_1\boldsymbol{x}_2...\boldsymbol{x}_m)$ is just “semi-temporally ordered.”

### 4.3 Distance Metric

We now have 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 web-based context and keyframes extracted from our travel video. Before correspondence determination, we need to define a distance metric for measuring a pair of images $\boldsymbol{x}_i$ and $\boldsymbol{y}_j$. In our previous work [10], histogram intersection is used to calculate similarity between two images, i.e.

$$S(i,j) = \sum_{k=1}^{N} \min(h_i[k], h_j[k]), \tag{4}$$

where $h_i[k]$ denotes value of the $k$th bin of the visual word histogram of $\boldsymbol{x}_i$, and $N$ is the number of visual words to represent an image. However, this method equally treats visual words in similarity measurement. From the perspective of document analysis, we know that some words play more important roles in presenting main concepts of a document. By analogizing an image as a document constituted by visual words, we argue that different visual words should be appropriately weighted so that similarity between images can be well described.

Conceptually, images retrieved based on the same keyword should present content directly related to the scenic spot, if the image or video search engines have perfect retrieval performance. However, none of the current search engines has perfect performance, and the amount of related data on the web depends on popularity of this scenic spot. The retrieved data, therefore, often consist of noise. Figure 4 shows search results from Google, based on the keyword “Ta-Lin train station”. Only the images with bold borders are directly related to this scenic spot.

The set of images retrieved from search engines is not perfect. However, it doesn’t mean that the retrieved set is not trustworthy at all. Previous research found that, though there are noises in the retrieved results, most of the top 10~20 images returned by Google are relevant to queries, when queries are general terms like *building, tiger*, and *sea* [27]. Statistics from the top retrieved results are thus meaningful. Based on the trust that these famous search engines would put their best efforts on accuracy of top-ranked retrieved results, we investigate characteristics of the top retrieved results and devise a weighting scheme. In contrast to [27], which removes irrelevant images with the helps from contemporary text-based search engines, we try to prioritize different visual words and develop a better distance metric to reduce the influence brought by noisy data.

The degree of discrimination of a visual word depends on two factors:

- For the images that contain a specific visual word, this visual word is more important for them if its average occurrence frequency is higher.
- A visual word is more discriminative if it occasionally presents in some images' visual word histograms. The visual word may be absent in noisy images, and appears in truly related images. On the other hand, if a visual word appears in all retrieved images, it provides less information for distinguishing truth data from noisy data.

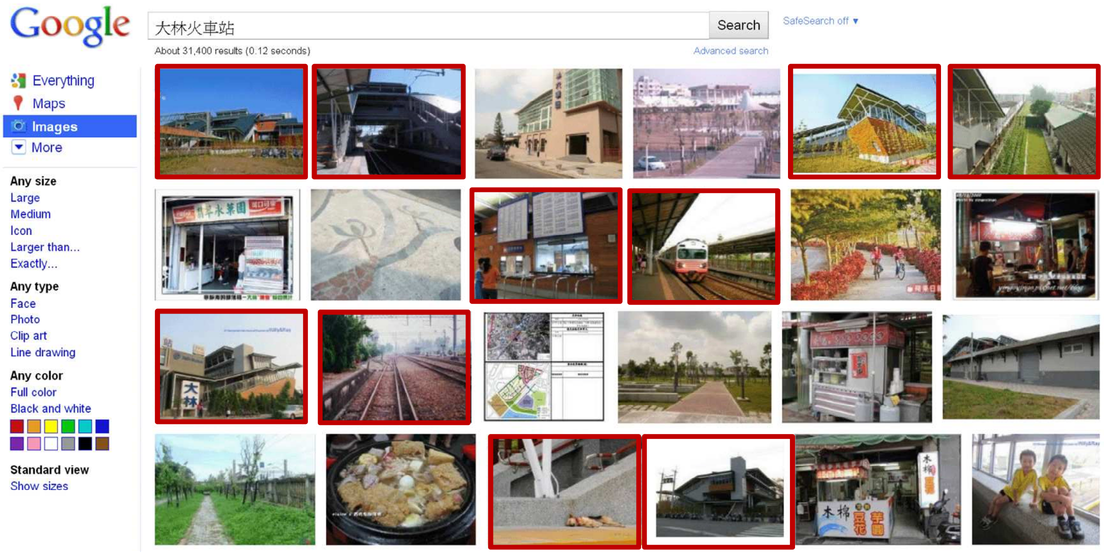
By combining two factors described above, we determine the weight of the $k$th visual word from the retrieved data corresponding to a scenic spot by

$$w_k = \frac{\sum_{j=1}^{q} h_j[k]}{|Z_k| + \epsilon} \times \left(1 - \frac{|Z_k|}{q+1}\right), \quad (5)$$

where $Z_k$ denotes the set of images that contain the $k$th visual word, $|Z_k|$ denotes the number of such images, and $q$ is the number of images retrieved for a specific scenic spot. The parameter $\epsilon$ is set as a small value to avoid zero denominator. The first term in eqn. (5) denotes the average count of the $k$th visual word in $Z_k$. More frequently the $k$th visual word appears in $Z_k$, larger this term is. The second term denotes degree of discrimination of this visual word. If all retrieved images contain the $k$th visual word, $|Z_k| = q$. This means that if a visual word appears in more retrieved images, it is less useful to discriminate different images. Note that the design of eqn. (5) follows the idea of term frequency multiplying inverse document frequency, which is widely used to detect important words in text documents.

Because characteristics of data retrieved by different keywords may be different, weights of visual words for different scenic spots are calculated adaptively. Finally, the similarity between the retrieved image $\boldsymbol{x}_i$ and the keyframe $\boldsymbol{y}_j$ extracted from our travel video is calculated as

$$I(\boldsymbol{x}_i, \boldsymbol{y}_j) = \sum_{k=1}^{N} w_k \times \min(h_i[k], h_j[k]). \quad (6)$$



大林火車站

About 31,400 results (0.12 seconds)

Figure 4. Image search results of “Ta-Lin train station.” Only the images with bold borders are directly related to this place.

### 4.4 Maximum-Sum Segment

Finding correlations between our travel videos and images retrieved from the web has been transformed into a sequence matching problem. Generally, the dynamic programming strategy can be used to conduct approximate sequence matching, such as finding the longest common subsequence (LCS) [11]. However, images retrieved by keywords are just “semi-temporally ordered.” Although images related to different keywords are temporally sorted, that related to the same keyword don’t necessarily follow the same rule. This characteristic destroys the sequential property necessary for the LCS algorithm. In addition, there may be many irrelevant images in the retrieved data, which makes correlation determination more challenging (see Figure 4).

Let’s consider 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 images and keyframes. The sequence $X$ is semi-temporally ordered, i.e. $X=(S_{k_1}S_{k_2}...S_{k_V})$, where $S_{k_i}=(\boldsymbol{x}_l\boldsymbol{x}_{l+1}...)$ consists of images retrieved from the keyword $k_i$. 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 image set $S_{k_i}$, we have 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 $i<j$.

The conventional maximum-sum segment problem is given by a nonempty sequence of real numbers, and the goal is to find the contiguous subsequence that has the largest total sum. To find the segment $Y(p_i, q_i)$ corresponding to the scene $S_{k_i} = (\boldsymbol{x}_l \boldsymbol{x}_{l+1} \cdots)$, we first transform the sequence $Y = (\boldsymbol{y}_1 \boldsymbol{y}_2 \cdots \boldsymbol{y}_n)$ into a real number sequence $Z = (z_1 z_2 \ldots z_n)$ as follows. Based on the weighted visual word histogram intersection (eqn. (6)) between $\boldsymbol{y}_j$ and $\boldsymbol{x}_a$, denoted by $I(\boldsymbol{y}_j, \boldsymbol{x}_a)$, we first calculate the similarity $z'_j$ between $\boldsymbol{y}_j$ and $\boldsymbol{x}_a$ in $S_{k_i}$:

$$z'_j = I(\boldsymbol{y}_j, \boldsymbol{x}_{a^*}) \text{ and } a^* = \arg\max_a I(\boldsymbol{y}_j, \boldsymbol{x}_a), \tag{7}$$

where, $a = l, l+1, \ldots, l + |S_{k_i}| - 1$. The value $|S_{k_i}|$ denotes the number of retrieved images for this scene. To obtain a sequence of real numbers, in which positive numbers denote similarity between a keyframe and an image is higher than the average level, and negative numbers denote similarity between them is under the average level, we remove mean of the similarity sequence, i.e.

$$z_j = z'_j - \tfrac{1}{n} \textstyle\sum_j z'_j. \tag{8}$$

Recall that scenic spots were visited sequentially and the content in travel video should present the visited places in the same order. It's not likely that the keyframes at the beginning of the travel video (e.g. $\boldsymbol{y}_1$, $\boldsymbol{y}_2$, and $\boldsymbol{y}_3$) match with the images retrieved by the keyword representing the last visited scenic spot (e.g. $\boldsymbol{x}_m$, $\boldsymbol{x}_{m-1}$, and $\boldsymbol{x}_{m-2}$). By considering the temporal characteristic, we don't search for the maximum-sum segment from the whole sequence, but instead search from an interval in which image matches reasonably exist.

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. $\sum_{a=p_i}^{q_i} z_a$ 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:

$$L_i = \max(0, n \times \tfrac{i-2}{V}) \text{ and } U_i = \min(n, n \times \tfrac{i+2}{V}). \tag{9}$$

The value $V$ is the number of visited scenic spots. 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.

Given the search interval $Z(L_i, U_i) = (z_{L_i}, z_{L_i+1}, \ldots, z_{U_i})$, the maximum-sum segment is determined by the algorithm shown in Figure 5. Let $C[j]$ denote the cumulative sum of $Z(L_i, U_i)$, defined by $C[j] = \sum_{L_i \leq a \leq j} z_a$ for $L_i \leq a \leq U_i$ and $j \leq U_i$. In the cumulation process, if the cumulative sum is positive at $j-1$ but negative at $j$, we find a candidate segment from $\ell$ to $j$. For this candidate segment, we store the largest cumulative sum in $\mathcal{M}[k]$, and store the left boundary $\mathcal{L}[k]$ and the right boundary $\mathcal{U}[k]$ of a subsegment that causes $\mathcal{M}[k]$. After finding all candidate segments, we take the subsegment that causes the overall maximum sum.

In each iteration in the loop, there is one addition for accumulation, and $j$ comparisons are needed to simultaneously find the index that causes the maximum sum and the value of maximum sum. Suppose the length of $Z(L_i, U_i)$ is $N$, the value of $j$ is at most $N$ , and the total complexity of the loop is $O(N \times (1+j)) \approx O(N^2)$ in the worst case. Finding the subsegment that causes the overall maximum sum takes $O(K)$ if $K$ candidates are found. Note that $K \ll N$. Overall, if $V$ scenic spots were visited, finding all maximum-sum segments for the whole video takes $O(V \times (N^2 + K)) \approx O(VN^2)$, where $V$ is much smaller than $N$. In this paper we just introduce the idea of using this algorithm to find optimal correspondence, without considering elaborate data structure or other special design to reduce computational complexity. Elegant algorithm design for related issues please refer to [8].

```
Input: A sequence of real numbers $Z(L_i, U_i) = (z_{L_i} z_{L_i+1} \cdots z_{U_i})$
Output: Indices of the maximum-sum segment $(p_i, q_j)$ in $Z(L_i, U_i)$
$\ell = L_i$;
$C(L_i - 1) = 0$;
for $j \leftarrow L_i$ to $U_i$ do
    $C(j) \leftarrow z_j + \max(C(j-1), 0)$; //Cumulate real numbers
    if $C(j) < 0$ then
        $\mathcal{U}[k] \leftarrow \arg\max_{\ell \le a \le j} C(a)$; //Find the idx that causes the maximum sum
        $\mathcal{L}[k] \leftarrow \ell$;
        $\mathcal{M}[k] \leftarrow \max_{\ell \le a \le j} C(a)$;      //Store the maximum sum
        $C(j) \leftarrow 0$;                            //Reset the cumulation process
        $\ell \leftarrow j + 1$;
        $k \leftarrow k + 1$
    end if
end for
$k^* = \arg\max_k \mathcal{M}[k]$;
$p_i = \mathcal{L}[k^*]$;
$q_i = \mathcal{U}[k^*]$;
```

Figure 5. The maximum-sum segment algorithm.

### 4.5 Video Scene Boundary Determination

After determining correspondence, keyframes in the selected maximum-sum segment are assigned a scene label according to the corresponding images. For example, if we find that keyframes $\boldsymbol{y}_{p_i}, \boldsymbol{y}_{p_i+1}, \ldots, \boldsymbol{y}_{q_i}$ correspond to data retrieved based on the name entity of $S_{k_i}$, these keyframes are assigned as in the $i$th scenic spot.

Note that lengths of maximum-sum segments corresponding to different scenic spots may be varied. Moreover, because the search intervals for successive scenic spots are overlapped (see eqn. (9)) and noisy images retrieved by different keywords may be similar, the maximum-sum segments corresponding to different scenic spots may be overlapped. To handle this problem, we especially examine maximum-sum segments for any two successive scenic spots. Figure 6 illustrates four possible cases.

- The case in Figure 6(a)

Figure 6(a) shows the simplest case, in which two maximum-sum segments for successive scenic spots are not overlapped. Keyframes $\boldsymbol{y}_{p_i}, \ldots, \boldsymbol{y}_{q_i}$ are assigned as in the $i$th 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 $p_{i+1}$, we have to determine each keyframe in either the $i$th scene or the $(i+1)$th scene. In our previous work [10], we simply determine by linear interpolation based on the ratio of numbers of keyframes in the $i$th maximum-sum segment to that in the $(i+1)$th maximum-sum segment, which is a blind process without consideration of visual similarity and locality characteristics.

We can view this as a binary label problem and solve it by an optimization formulation [36][37]. Gu et al. [36] jointly consider local temporal continuity and global distribution of time and content to conduct scene detection. Wang et al. [37] consider both content coherence and temporally context by formulating this task as a chain segmentation problem. In this work we develop an energy minimization formulation similar to [36], but further consider the influence of web-based context. This method eliminates the limitation that parametric models have limited performance to describe various complex scenes [37].

Now the goal is to find a labeling that assigns each keyframe $\boldsymbol{y}_k$, $q_i \leq k \leq p_{i+1}$, a label $f_k$, where the labeling is considered to be consistent with the observed data and should conform to smoothness of neighborhood. The formal energy minimization problem is considered as

$$E(f) = \sum_{q_i<k<p_{i+1}} D_k(f_k) + \sum_{\{k,k+1\}} V_{k,k+1}(f_k, f_{k+1}), \qquad (10)$$

where $D_k(f_k)$ denotes the cost of assigning $\boldsymbol{y}_k$ the label $f_k$, $f_k = \{0,1\}$. The label $f_k = 0$ means that $\boldsymbol{y}_k$ is labeled as in the $i$th scene, and $f_k = 1$ means that $\boldsymbol{y}_k$ is labeled as in the $(i+1)$th scene. The value $D_k(f_k)$ is called *data cost*, which will be defined later. The set $\{k, k+1\}$ denotes pairs of keyframes that are temporally adjacent to each other, e.g. $\boldsymbol{y}_k$ and $\boldsymbol{y}_{k+1}$. The value $V_{k,k+1}(f_k; f_{k+1})$ denotes the penalty of assigning adjacent keyframes $\boldsymbol{y}_k$ and $\boldsymbol{y}_{k+1}$ the labels $f_k$ and $f_{k+1}$, respectively, and is called *smoothness cost*. Although finding the optimal solution of this formulation is NP-hard, fast approximate algorithms have been developed [12].

The idea of defining the data cost is that if a keyframe $\boldsymbol{y}_k$ is more similar to the ones in the $i$th scene, the data cost $D_k(f_k = 0)$ is smaller, and contrarily $D_k(f_k = 1)$ is larger. Two sets of data are considered to define the data cost of labeling $\boldsymbol{y}_k$: intra-domain cost and inter-domain cost. The intra-domain cost is derived from the distance between $\boldsymbol{y}_k$ and other keyframes that have been assigned, i.e. $\boldsymbol{y}_{p_i}$, …, $\boldsymbol{y}_{q_i}$ and $\boldsymbol{y}_{p_{i+1}}$, …, $\boldsymbol{y}_{q_{i+1}}$. The inter-domain cost is derived from the distance between $\boldsymbol{y}_k$ and images respectively retrieved by the keywords representing the $i$th scene and the $(i+1)$th scene. The retrieved images are respectively denoted by $\boldsymbol{x}_{i,1}$, …, $\boldsymbol{x}_{i,n_i}$ and $\boldsymbol{x}_{i+1,1}$, …, $\boldsymbol{x}_{i+1,n_i}$, assuming that $n_i$ images are retrieved for both scenes. By considering intra- and inter-domain factors, the overall data cost $D_k^{(a)}(f_k = 0)$ for Figure 6(a) is defined as

$$D_k^{(a)}(f_k = 0) = \alpha D_k^r(f_k = 0) + (1-\alpha) D_k^t(f_k = 0). \tag{11}$$

The intra-domain cost $D_k^r(f_k = 0)$ is

$$D_k^r(f_k = 0) = 1 - \tfrac{1}{q_i - p_i + 1} \textstyle\sum_{p_i \le j \le q_i} I(\boldsymbol{y}_j, \boldsymbol{y}_k), \tag{12}$$

where $I(\boldsymbol{y}_i, \boldsymbol{y}_k)$ is the weighted histogram intersection defined in eqn. (6). The data cost of assigning $\boldsymbol{y}_k$ as in the $i$th scene is inversely proportional to the average similarity between it and the keyframes in $i$th scene. The intra-domain cost $D_k^r(f_k = 1)$ of assigning $\boldsymbol{y}_k$ as in the $(i+1)$th scene is defined similarly.

The inter-domain cost $D_k^t(f_k = 0)$ is defined as

$$D_k^t(f_k = 0) = 1 - \tfrac{1}{n_i} \textstyle\sum_{1 \le j \le n_i} I(\boldsymbol{x}_{i,j}, \boldsymbol{y}_k). \tag{13}$$

The parameter $\alpha$ controls the relative importance for intra- and inter-domain costs. Because the keyframe $\boldsymbol{y}_k$ is captured by the same setting as other keyframes, we impose higher confidence on intra-domain cost, and thus set $\alpha = 0.7$ in this work.

For smoothness cost, more similar $\boldsymbol{y}_k$ and $\boldsymbol{y}_{k+1}$ are, higher penalty is imposed if they are assigned different labels. Therefore, this factor is defined as

$$V_{k,k+1}(f_k, f_{k+1}) = \beta (f_k - f_{k+1})^2 I(\boldsymbol{y}_k, \boldsymbol{y}_{k+1}), \tag{14}$$

where $\beta$ is a parameter controlling the weight of weighted histogram intersection between two adjacent keyframes, and is set as 0.01 in this work.

In travel videos, a temporal characteristic can also be considered in the optimization process. The keyframe that is closer to the $i$th scene tends to be assigned to the $i$th scene. Therefore, this factor, called *temporal cost*, is considered by

$$T_k^{(a)}(f_k = 0) = \tfrac{k - q_i + 1}{p_{i+1} - q_i + 1} \text{ and } T_k^{(a)}(f_k = 1) = \tfrac{p_{i+1} - k + 1}{p_{i+1} - q_i + 1}. \tag{15}$$

Finally, in our work the energy term that should be minimized is modified as

$$E^{(a)}(f) = \textstyle\sum_{q_i < k < p_{i+1}} D_k^{(a)}(f_k) + \sum_{q_i < k < p_{i+1}} T_k^{(a)}(f_k) + \sum_{\{k,k+1\}} V_{k,k+1}(f_k, f_{k+1}) \quad . \tag{16}$$

The computational complexity of calculating these three costs is analyzed as follows. If the number of images in the search region ($p_i$ to $q_i$, or $p_{i+1}$ to $q_{i+1}$) is $N_1$, and the number of bins in visual word histograms is $N_2$, the complexity for calculating the intra-domain cost is $O(2\times N_1\times(2N_2))$, where $2N_2$ accounts for the comparison and weighting multiplication in eqn. (6). If the number of retrieved images is $N_3$, the complexity for calculating the inter-domain cost is $O(N_3\times(2N_2))$. For the smoothness cost, if the number of undetermined images is $K$, the corresponding complexity is $O(K\times(2N_2))$. For the temporal cost, the corresponding complexity is simply $O(2K)$. Overall, the computational complexity of calculating energy for an undetermined keyframe is dominated by the number of visual words ($N_2$).

- Other cases

If two maximum-sum segments are overlapped as in Figure 6(b), the keyframes from $\boldsymbol{y}_{p_{i+1}}$ to $\boldsymbol{y}_{q_i}$ are reexamined, and the energy term is defined as

$$E^{(b)}(f)=\sum\nolimits_{p_{i+1}\le k\le q_i}D_k^{(b)}(f_k)+\sum\nolimits_{p_{i+1}\le k\le q_i}T_k^{(b)}(f_k)+\sum\nolimits_{\{k,k+1\}}V_{k,k+1}(f_k,f_{k+1}) \quad , \tag{17}$$

where the boundary conditions of data cost and temporal cost should be appropriately modified as follows. The overall data cost $D_k^{(b)}(f_k=0)$ for Figure 6(b) is defined as

$$D_k^{(b)}(f_k=0)=\alpha D_k^r(f_k=0)+(1-\alpha)D_k^t(f_k=0), \tag{18}$$

where

$$D_k^r(f_k=0)=1-\frac{1}{p_{i+1}-p_i+1}\sum\nolimits_{p_i\le j\le p_{i+1}}I(\boldsymbol{y}_j,\boldsymbol{y}_k), \tag{19}$$

$$D_k^t(f_k=0)=1-\frac{1}{n_i}\sum\nolimits_{1\le j\le n_i}I(\boldsymbol{x}_{i,j},\boldsymbol{y}_k). \tag{20}$$

The overall data cost $D_k^{(b)}(f_k=1)$ for assigning $\boldsymbol{y}_k$ as in the $(i+1)$th scene is defined similarly.

The temporal cost for Figure 6(b) is defined as

$$T_k^{(b)}(f_k=0)=\frac{k-p_{i+1}+1}{q_i-p_{i+1}+1} \text{ and } T_k^{(b)}(f_k=1)=\frac{q_i-k+1}{q_i-p_{i+1}+1}. \tag{21}$$

In the cases of Figure 6(c) and Figure 6(d), the temporal cost and smoothness cost are not well defined, and only the inter-domain data cost can be used. For Figure 6(c), the keyframes to be reexamined are from $\boldsymbol{y}_{p_{i+1}}$ to $\boldsymbol{y}_{q_i}$, and the energy term to be minimized is

$$E^{(c)}(f)=\sum\nolimits_{p_{i+1}\le k\le q_i}D_k^t(f_k). \tag{22}$$

For Figure 6(d), the keyframes to be reexamined are from $\boldsymbol{y}_{p_i}$ to $\boldsymbol{y}_{q_{i+1}}$, and the energy term to be minimized is

$$E^{(d)}(f)=\sum\nolimits_{p_i\le k\le q_{i+1}}D_k^t(f_k). \tag{23}$$

These two troublesome cases are caused by significant amounts of noisy images in the retrieved data or user's travel videos.

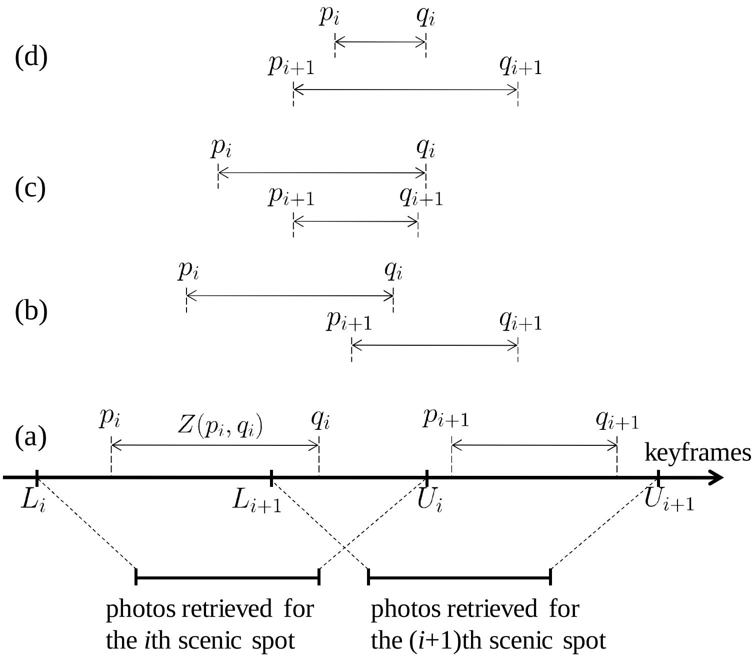


Figure 6. Illustrations of different situations in results of maximum-sum segment determination.

## 5. Evaluation

### 5.1. 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 $480 \times 272$. Figure 7 shows some snapshots of scenes in each video. Table 1 shows information of scenes, keyframes, and length of each travel video. There are totally 30 different visited scenic spots in the evaluation dataset.

According to travel schedules, we respectively retrieve 40 top-ranked photos from Google, Yahoo!, and Flickr image search engines for each scenic spot. Data from three sources are experimented separately to investigate how the proposed method works on photos retrieved by different search scenarios. Because resolutions of the retrieved photos are varied, we normalize them into $400 \times 300$ for efficiency of feature extraction and visual word construction.

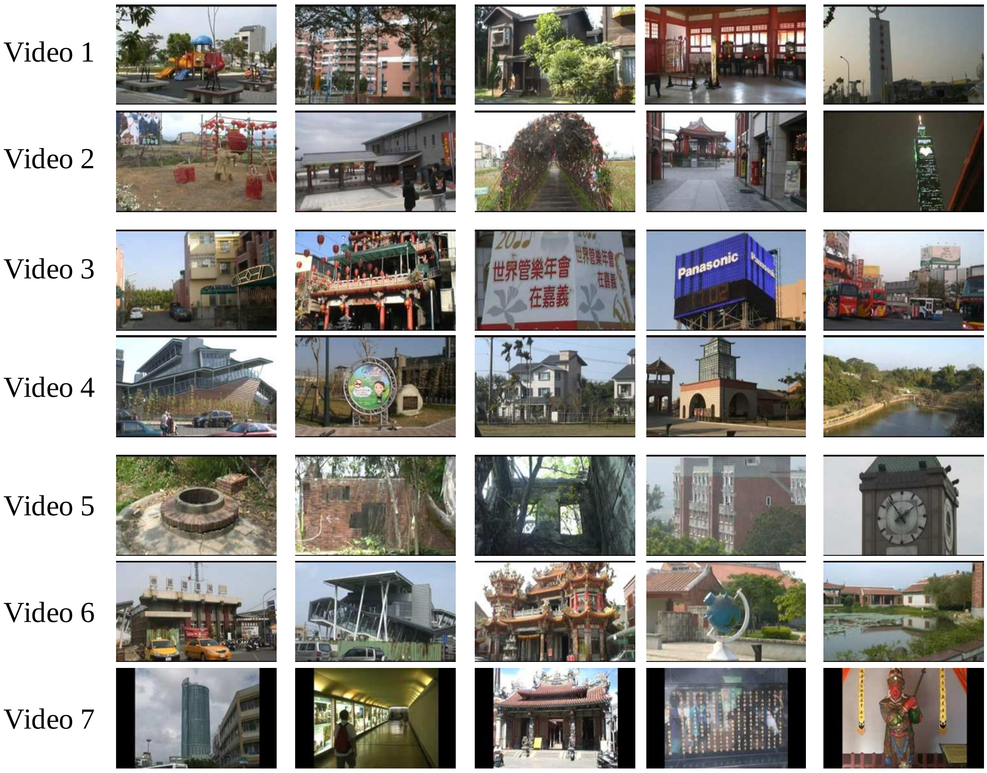


Figure 7. Some snapshots of the evaluated videos.

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 |

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 $\rho$ is defined as

$$\rho = \left( \sum_{i=1}^{Ng} \frac{\tau(s_i)}{T} \sum_{j=1}^{Nv} \frac{\tau^2(s_i, s_j^*)}{\tau^2(s_i)} \right) \times \left( \sum_{j=1}^{Nv} \frac{\tau(s_j^*)}{T} \sum_{i=1}^{Ng} \frac{\tau^2(s_i, s_j^*)}{\tau^2(s_j^*)} \right), \quad (24)$$

where $\tau(s_i, s_j^*)$ is the length of overlap between $s_i$ 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, for example $\tau(s_i)$, is represented by the number of shots in this scene.

## 5.2. Performance Comparison

### 5.2.1 Performance with Weighting Scheme

We first evaluate whether the proposed weighting scheme better examines similarity between noisy data sets. In Table 2, the numbers in the rows with “W-” denote the purity values of scene detection with the weighted distance metric. The best performance for each dataset is emphasized with bold face. We see that except for the first dataset, the best performance can be obtained by using the weighting scheme. It’s not surprising that varied performance can be achieved for different datasets due to significant content variations and characteristics of visited scenic spots. The last row shows the improvement from the non-weighting scheme to the weighting scheme in the best case. We see significant improvement for datasets 2, 3, 4, 5, and 7, and averagely we obtain 20% improvement for the whole evaluation data. The weighting scheme fails for the first dataset. Because part of visited scenes is only known by the natives and is not popular, it’s hard to accurately retrieve related images from search engines and derive appropriate weightings for different visual words.

An interesting observation is that we mostly obtain the best scene detection performance by consulting images retrieved from Google or Yahoo, which may inspire us a way to evaluate a general-purpose image search engine. On the other hand, from the Flickr-retrieved images we obtain the worst performance. This may due to that we only retrieve related images based on tag search rather than full text. Because of human subjectivity or noisy data, images with the same tag may not be visually coherent [34], which makes visual analysis futile.

Table 2. Performance of scene detection with weighted vs. nonweighted visual word histogram intersection.

| | data1 | data2 | data3 | data4 | data5 | data6 | data7 | Avg. |
|---|---|---|---|---|---|---|---|---|
| Google | **0.77** | 0.56 | 0.49 | **0.73** | 0.49 | **0.86** | 0.41 | 0.62 |
| W-Google | 0.6 | **0.98** | 0.6 | 0.53 | 0.62 | **0.86** | 0.47 | **0.67** |
| Yahoo | 0.57 | 0.51 | 0.54 | 0.56 | 0.67 | 0.58 | 0.57 | 0.57 |
| W-Yahoo | 0.5 | 0.94 | **0.68** | 0.6 | **0.78** | 0.51 | **0.71** | **0.67** |
| Flickr | 0.43 | 0.49 | 0.36 | 0.55 | 0.6 | 0.58 | 0.55 | 0.51 |
| W-Flickr | 0.48 | 0.56 | 0.43 | **0.73** | 0.49 | 0.58 | 0.5 | 0.54 |
| improvement | -0.28 | 0.75 | 0.26 | 0.33 | 0.16 | 0 | 0.25 | 0.21 |

### 5.2.2 Performance with Blur Detection

By adopting the method in [7], we examine blur extent of keyframes extracted from our videos and images retrieved from three search engines, and thus filter out blurred images. Table 3 shows that with blurred image filtering, performances of scene detection in different settings improve from 2% to 5%. With the results in Table 2 and Table 3, all experiments in the following context are conducted with the proposed weighting scheme and appropriate blur filtering.

Table 3. Performance of scene detection with and without blur detection.

| | Google | Yahoo | Flickr |
|---|---|---|---|
| w/o blur detection | 0.67 | 0.67 | 0.54 |
| w. blur detection | 0.70 | 0.70 | 0.59 |

### 5.2.3 Performance of Different Boundary Determination Methods

Boundaries between scenes in the previous two subsections are determined by the energy minimization framework described in Section 4.5. Here we provide detailed performance comparison to verify superiority of the newly-proposed method. Three methods are compared: the naïve method, the method in [10], and the method newly proposed in this article. In the naïve method, suppose that *k* scenic spots were visited, we sort keyframes according to their temporal order and equally divided them into *k* groups. Keyframes in the same group are assigned as in the same video scene. Note that in any of the three methods, keyframes extracted from the same shot may be assigned to different scenes. To eliminate this unreasonable assignment, we determine the scene label of keyframes extracted from the same shot by majority voting. Therefore, keyframes in the same shot has the same scene label, which implies that each shot has a corresponding scene label. Therefore, the final results of boundary determination are shots with associated scene labels, and the purity values are calculated at shot level.

Figure 8 shows average purity values for three methods. The naïve approach has the worst performance because no visual correlation is considered. 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. The newly-proposed method has significant improvement over our previous work [10]. Over 0.1 purity improvement can be obtained if we discover correlation based on data retrieved by Google or Yahoo, while about 0.05 purity improvement is obtained for the Flickr case. We believe that this work surpasses the previous one [10] from the following perspectives:

- Keyframes of ill visual quality are filtered out by an advanced method [7] to avoid their bad influence on sequence matching. (Section 4.1)
- Keyframes of nonsense content are filtered out by considering visual content entropy to avoid their bad influence on sequence matching. (Section 4.1)
- Visual words are prioritized by an adaptive weighting scheme so that this new distance metric more appropriately captures similarity between images. (Section 4.3)
- Scene boundaries are refined by an energy minimization method, which jointly considers content coherence, temporal continuity, and web-based context. (Section 4.5)

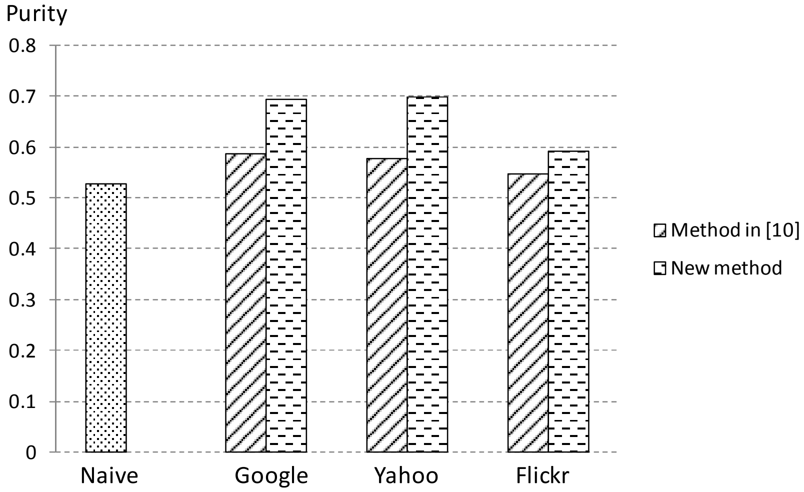


Figure 8. Average purity values of three methods: the naïve method, the method in [10], and the newly-proposed method.

### 5.2.4 Performance with Web-based Video Context

We have verified that web-based context provides implicit but useful clues for segmenting our own videos. In this subsection, we further experiment whether the idea can be applied when other web-based contexts are considered. To this end, we search three top-ranked web-based videos from Youtube, by the name entities extracted from the travel schedules as well. Techniques of keyframe extraction, blur filtering, and visual word histogram representation are applied to the retrieved videos.

Table 4 compare purity values obtained with different contexts. We surprisingly see that scene detection referring to keyframes from Youtube videos has great performance. Comparing Youtube keyframes with images retrieved from image search engines, entropy of keyframes in the same scenic spot is slightly smaller than that of images. On the other hand, keyframes in different scenic spots often significantly differs in visual appearance. Although search results of Youtube are not very satisfactory in terms of human's subjectivity, characteristics of low intra-variation and high inter-variation benefits segmenting video into scenes.

Table 4. Performance of scene detection with contexts from different sources.

| | Google | Yahoo | Flickr | Youtube |
|---|---|---|---|---|
| Purity | 0.70 | 0.70 | 0.59 | 0.72 |

### 5.2.5 Performance of Methods Using Different Features

To verify superiority of utilizing web-context and travel video characteristics, we compare our work with a clustering method that considers similarity between keyframes by solely visual content, or by both visual and temporal information. The affinity propagation (AP) algorithm [35] is adopted to cluster keyframes into a targeted number of groups, which is set according to the corresponding text schedule. The AP algorithm takes similarity between pairs of keyframes as input, randomly chooses an initial subset of keyframes as examplars, and iteratively exchanges messages between examplars and other data points until convergence. Two types of messages are considered: responsibility and availability. The responsibility message $r(m,n)$ indicates how well point $n$ serves as the examplar for point $m$. The availability $a(m,n)$ indicates how well point $m$ chooses point $n$ as its examplar. Jointly considering these two messages indicates how likely points $m$ and $n$ should be clustered together.

The visual similarity $I_v(\boldsymbol{k}_i,\boldsymbol{k}_j)$ between keyframes $\boldsymbol{k}_i$ and $\boldsymbol{k}_j$ is defined as the weighted visual histogram intersection (eqn. (6)). The temporal similarity between them is defined as $I_t(\boldsymbol{k}_i,\boldsymbol{k}_j)=\exp(-|j-i|)$, where $i$ and $j$ are in the unit of frame numbers. They are linearly combined to jointly consider visual and temporal information: $I_o(\boldsymbol{k}_i,\boldsymbol{k}_j)=\gamma I_v(\boldsymbol{k}_i,\boldsymbol{k}_j)+(1-\gamma)I_t(\boldsymbol{k}_i,\boldsymbol{k}_j)$, where the parameter $\gamma$ is set as 0.9 empirically, indicating that visual similarity is more important in defining similarity.

After clustering by the AP algorithm, a post-processing is applied to ensure that keyframes extracted from the same shot are in the same cluster, and temporally adjacent keyframes should be in the same cluster. For example, if the text schedule indicates that scenic spots were visited in the order of A to D, but scene labels of a sequence of keyframes are AAACDBBBCCDD, the fourth and the fifth keyframes should be reassigned as A or B according to their similarity to scenes A and B, maintaining correct temporal order in the meanwhile.

Table 5 shows performance comparison between the affinity approach with visual similarity (V), with both visual and temporal similarity (V+T), and the best cases of our approach. We see that for all datasets our approach achieves much better performance. By comparing AP(V) with AP(V+T), we found that temporal information provides little improvement over the one only using visual information. On the other hand, our approach that jointly considers visual, temporal, and web-context-based similarity has great performance improvement. As we described in Section 1, it's not necessary that visually similar shots belong to the same scene, especially for travel videos.

Table 5. Comparison of purity values based on different approaches.

| | data1 | data2 | data3 | data4 | data5 | data6 | data7 | Avg. |
|---|---|---|---|---|---|---|---|---|
| AP(V) | 0.46 | 0.62 | 0.43 | 0.62 | 0.49 | 0.54 | 0.47 | 0.52 |
| AP(V+T) | 0.43 | 0.90 | 0.42 | 0.69 | 0.53 | 0.54 | 0.35 | 0.55 |
| Our | 0.77 | 0.98 | 0.68 | 0.73 | 0.78 | 0.86 | 0.71 | 0.79 |

## 6. Conclusion

We have presented a novel video scene detection method that exploits web-based context to segment personal video collections, especially for travel videos captured in journeys. In addition to analyze personal videos, we retrieve web-based context by keyword search on general-purpose image and video search engines, and then discover temporal and visual correlation between them to facilitate scene detection. Keyframes extracted from videos and the retrieved images are represented by visual word histograms. A weighting scheme is designed to adaptively prioritize different visual words, with which similarity between images can be characterized well. Correlation between media is then determined by an approximate sequence matching algorithm, i.e. the maximum-sum segment algorithm. With the cross-media correlation, an energy minimization framework is introduced to determine scene boundaries in keyframe sequences. Experimental results verify effectiveness of the proposed method and superiority over previous works. We discuss performance variations derived from different types of web-based context. In the future, this idea may be extended to conduct various kinds of multimedia content analysis, such as face clustering or identification with the aids of social context, or personalized media management with the help of web-based context.

## Reference

[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] N.D. Narvekar and L. J. Karam. A no-reference perceptual image sharpness metric based on a cumulative probability of blur detection. In Proc. of IEEE International Workshop on Quality of Multimedia Experience, pp. 87-91, 2009.

[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.

[10] W.-T. Chu, C.-J. Li, and T.-C. Lin. Travel video scene detection by search. In Proc. of Pacific Symposium on Image and Video Technology, 2010.

[11] T.S. Cormen, C.E. Leiserson, R.L. Rivest, and C. Stein. Introduction to Algorithms. The MIT Press, 2001.

[12] Y. Boykov, O. Veksler, and R. Zabih. Fast approximation energy minimization via graph cuts. IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 23, no. 11, pp. 1222-1239, 2001.

[13] Z. Rasheed and M. Shah. Scene detection in Hollywood movies and tv shows. In Proc. of IEEE Computer Society Conference on Computer Vision and Pattern Recognition, vol. 2, pp. 343-348, 2003.

[14] M. Yeung and B.-L. Yeo. Segmentation of video by clustering and graph analysis. Computer Vision and Image Understanding, vol. 71, no. 1, pp. 94-109, 1998.

[15] X. Zhu, L. Wu, X. Xue, X Lu, and J. Fan. Automatic scene detection in news program by integrating visual feature and rules. In Proc. of IEEE Pacific Rim Conference on Multimedia, pp. 843-848, 2001.

[16] D. Gatica-Perez, A. Loui, and M.-T. Sun. Finding structure in home videos by probabilistic hierarchical clustering. IEEE Transactions on Circuits and Systems for Video Technology, vol. 13, no. 6, pp. 539-548, 2003.

[17] Z. Pan and C.-W. Ngo. Structuring home video by snippet detection and pattern parsing. In Proc. of ACM International Workshop on Multimedia Information Retrieval, pp. 69-76, 2004.

[18] X.-S. Hua, L. Lu, and H.-J. Zhang. Optimization-based automated home video editing system. IEEE Transactions on Circuits and Systems for Video Technology, vol. 14, no. 5, pp. 572-583, 2004.

[19] S.-H. Lee, S.-Z. Wang, and C.C.J. Kuo. Tempo-based MTV-style home video authoring. In Proc. of IEEE International Workshop on Multimedia Signal Processing, 2005.

[20] W.-T. Peng, Y.-H. Chiang, W.-T. Chu, W.-J. Huang, W.-L. Chang, P.-C. Huang, and Y.-P. Hung. Aesthetics-based automatic home video skimming system. In LNCS 4903, pp. 186-197, 2008.

[21] F. Shipman, A. Girgensohn, and L. Wilcox. Authoring, viewing, and generating hypervideo: an overview of Hyper-Hitchcock. ACM Transactions on Multimedia Computing, Communications, and Applications, vol. 5, no. 2, Article no. 15, 2008.

[22] R.S.V. Achanta, W.-Q. Yan, and M.S. Kankanhalli. Modeling intent for home video repurposing. IEEE Multimedia, vol. 13, no. 1, pp. 46-55, 2006.

[23] T. Mei and X.-S. Hua. Intention-based home video browsing. In Proc. of ACM Multimedia, pp. 221-222, 2005.

[24] W.-H. Cheng, Y.-Y. Chung, Y.-Z. Lin, C.-C. Hsieh, S.-Y. Fang, B.-Y. Chen, and J.-L. Wu. Semantic analysis for automatic event recognition and segmentation of wedding ceremony videos, vol. 18, no. 11, pp. 1639-1650, 2008.

[25] A. Hanjalic, R.L. Lagendijk, and J. Biemond. Automated highlevel movie segmentation for advanced video retrieval system. IEEE Transactions on Circuits and System for Video Technology, vol. 9, no. 4, pp. 580-588, 1999.

[26] H. Sundaram and S.-F. Chang. Determining computable scenes in films and their structures using audio-visual memory models. In Proc. of ACM Multimedia, pp. 95-104, 2000.

[27] Y. Liu, D. Zhang, and G. Lu. SIEVE – search images effectively through visual elimination. In Proc. of International Conference on Multimedia Content Analysis and Mining, pp. 381-390, 2007.

[28] J. Wang and T.-S. Chua. A framework for video scene boundary detection. In Proc. of ACM Multimedia, pp. 243-246, 2002.

[29] A.F. Smeaton, P. Over, and W. Kraaij. Evaluation Campaigns and TRECVid, In Proc. of ACM International Workshop on Multimedia Information Retrieval, pp. 321-330, 2006.

[30] J. Vendrig and M. Worring. Systematic evaluation of logical story unit segmentation. IEEE Transactions on Multimedia, vol. 4, no. 4, pp. 492-499, 2002.

[31] Y. Takeuchi and M. Sugimoto. User-adaptive home video summarization using personal photo libraries. In Proc. of ACM International Conference on Image and Video Retrieval, pp. 472-479, 2007.

[32] D. Vallet, I. Cantador, and J.M. Jose. Exploiting external knowledge to improve video retrieval. In Proc. of ACM International Conference on Multimedia Information Retrieval, pp. 101-110, 2010.

[33] Z. Wang, M. Zhao, Y. Song, S. Kumar, and B. Li. YouTubeCat: Learning to categorize wild web videos. In Proc. IEEE Conference on Computer Vision and Pattern Recognition, pp. 879-886, 2010.

[34] L. Kennedy, M. Naaman, S. Ahern, R. Nair, and T. Rattenbury. How Flickr helps us make sense of the world: context and content in community-contributed media collections. In Proc. of ACM Multimedia, pp. 631-640, 2007.

[35] B.J. Frey and D. Dueck. Clustering by passing messages between data points. Science, vol. 315, no. 5814, pp. 972–976, 2007.

[36] Z. Gu, T. Mei, X.-S. Hua, X. Wu, and S. Li. EMS: Energy minimization based video scene segmentation, pp. 520-523, 2007.

[37] J. Wang, X. Tian, L. Yang, Z.-J. Zha, and X.-S. Hua. Optimized video scene segmentation. In Proc. of IEEE International Conference on Multimedia & Expo, pp. 301-304, 2008.