Jointly Summarizing Large Collections of Web Images and Videos for Storyline Reconstruction

Gunhee Kim
Disney Research Pittsburgh
gunhee@cs.cmu.edu

Leonid Sigal
Disney Research Pittsburgh
lsigal@disneyresearch.com

Eric P. Xing
Carnegie Mellon University
epxing@cs.cmu.edu

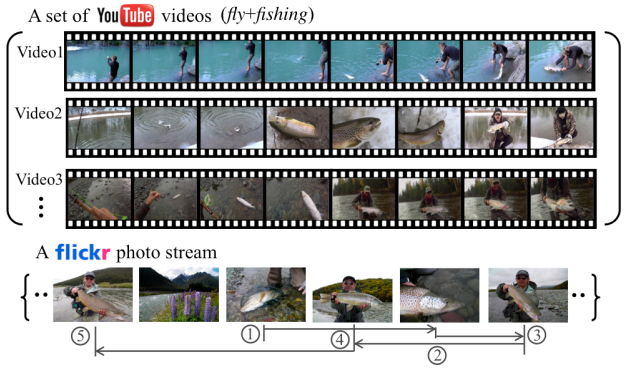
Abstract

In this paper, we address the problem of jointly summarizing large-scale Flickr images and YouTube user videos. Starting from the intuition that the characteristics of the two media are different yet complementary, we develop a fast and easily-parallelizable approach for creating not only high-quality video summary but also a novel structural summary of online images as storyline graphs, which can illustrate various events or activities associated with the topic in a form of a branching network. In our approach, the video summarization is achieved by diversity ranking on the similarity graphs between images and video frames. The reconstruction of storyline graphs is formulated as the inference of sparse time-varying directed graphs from a set of photo streams with assistance of videos. For evaluation, we create the datasets of 20 outdoor recreational activities, consisting of 2.7M of Flickr images and 16K of YouTube user videos. Due to the large-scale nature of our problems, we evaluate our algorithm via crowdsourcing using Amazon Mechanical Turk. In our experiments, we demonstrate that the proposed joint summarization approach outperforms other important baselines and our own methods using videos or images only.

1. Introduction

The recent explosive growth of online multimedia data has posed a new set of challenges in computer vision research. One of such infamous difficulties is that much of the data accessible to users are neither refined nor structured for later use, and subsequently has led to the *information overload* problem; users are often overwhelmed by the flood of unstructured pictures and videos, and in danger of getting lost in the data. Therefore, it is increasingly important to automatically summarize a large set of multimedia data in an efficient yet comprehensive way.

In this paper, we address the problem of jointly summarizing large-scale online images (*e.g.* Flickr) and videos (*e.g.* YouTube), especially in terms of *storylines*. Handling both still images and videos is becoming more needed, due to the recent convergence between cameras and camcorders. For example, any smartphone users can seamlessly record their memorable moments via both pictures and videos by freely switching between them with a single tap.



(a) A set of videos help discover the sequential structure of a photo stream

(b) A set of images help detect representative keyframes of a user videos

Figure 1. Intuition on the benefits of jointly summarizing Flickr images and YouTube videos with examples of *fly+fishing*. (a) Although images in a photo stream are taken consecutively, the underlying sequential structure between images is missing, which can be discovered by help of a crowd of videos. (b) Typical user videos contain much of noisy and redundant information, which can be removed using similarity votes cast by a large set of images that are taken more carefully from canonical viewpoints.

More importantly, jointly summarizing images and videos is *mutually-rewarding* for the summarization purpose, because their characteristics as recording media are different yet complementary (See Fig.1). The strength of images over videos lies in that images are more carefully taken so that they capture the subjects from canonical view-

points in a more semantically meaningful way. However, still images are fragmentally recorded, and thus the sequential structure is often missing even between consecutive images in a single photo stream. On the other hand, videos are *motion pictures*, which convey temporal smoothness between frames. However, one major issue of videos is that they contain much of noisy or redundant information with often poor quality. Therefore, as shown in Fig.1, we take advantage of sets of images to get rid of such noisy, redundant, or semantically meaningless parts of videos. In the reverse direction, we leverage sets of videos to glue fragmented images into coherent and smooth threads of storylines.

We first collect large sets of photo streams from Flickr and user videos from YouTube for a topic of interest (*e.g.* *fly+fishing*). We summarize each video with a small set of keyframes using similarity votes cast by the images from the most similar photo streams. Subsequently, leveraging the continuity information between the selected keyframes of videos, we discover the underlying sequential structure between images in each photo stream, and summarize the sets of photo streams in the form of *storyline graphs*. We represent the storylines as directed graphs in which the vertices correspond to dominant image clusters, and the edges connect the vertices that sequentially recur in many photo streams and videos. The summarization as storyline graphs is advantageous especially for the topics that consist of a sequence of activities or events repeated across the photo and video sets, such as recreational activities, holidays, and sports events. Moreover, the storyline graphs can characterize various branching narrative structure associated with the topic, which help users understand the underlying big picture surrounding the topic (*e.g.* a variety of activities that people usually enjoy during *fly+fishing*).

In our approach, the video summarization is achieved by diversity ranking on the similarity graphs between images and video frames (section 3). The reconstruction of storyline graphs is formulated as the inference of sparse time-varying directed graphs from a set of directed trees created from photo streams (section 4). As a result, our method provides several appealing properties, especially for large-scale problems, such as optimality guarantee, linear complexity, and easy parallelization.

For evaluation, we create the datasets of 20 outdoor recreational activities, which consist of about 2.7M images from 35K photo streams from Flickr and 16K user videos from YouTube. Due to the large-scale nature of our problems, we evaluate our algorithms via crowdsourcing using Amazon Mechanical Turk (section 5). In our experiments, we quantitatively show that the proposed joint summarization approach outperforms other baselines and our methods using videos or images only, for the tasks of video summarization and storyline reconstruction.

1.1. Previous work

Due to volume of literature on the subject, here we discuss a representative selection of works, from three notable lines of research, most closely related to our work.

Story-based image summarization: One of most common ways to summarize large-scale image databases is the *image retrieval*, with a small number of the most representative images (*e.g.* Google and Bing image search engines). Recent important threads of image summarization work in computer vision research are as follows. First, there have been many studies to summarize human's visual concepts of general categories with iconic images [6, 18]. Another important line of work is to group and organize unstructured community photos of popular landmarks in a spatially browsable way [20]. Finally, the work of [10] is related to our work in that it leverages large-scale Flickr images, and its research objective is motivated by the photo storyline reconstruction. However, [10] is a preliminary research that solely focuses on the alignment and segmentation of photo streams; no storyline reconstruction is explored.

Story-based video summarization: The story-based video summary has been actively studied in the context of sports [8] and news [16]. However, in this category of work, the videos of interest usually contain a small number of specified actors in fixed scenes with synchronized voices and captions, all of which are not available in unstructured user images and videos on the Web. The work of [9] may be one of the closest ones to our work, because images are used as a prior to create summaries of user-generated videos on eBay sites. The key difference of our work is that we complete a loop between jointly summarizing images and videos in a mutually-rewarding way. Also, our storyline summaries can support multiple branching structures unlike simple keyframe summaries of [9]. Recently, the summarization of ecocentric videos [13, 14] has emerged as an interesting topic, in which compact story-based summaries are produced from user-centric daylife videos. The objective of our work differs in that we are interested in the summarization for the collections of online images and videos that are independently taken by multiple anonymous users, instead of a single user's hours-long videos.

Computer vision leveraging both images and videos: Recently, it is gaining popularity to achieve challenging computer vision problems by leveraging both images and videos. Sets of new powerful algorithms have been developed by pursuing synergic interplay between the two complementary domains of information, especially in the areas of adapting object detectors between images and videos [17, 22], human activity recognition [4], and event detection [5]. However, the storyline reconstruction extracted from both images and videos still remains as a novel and largely under-addressed problem.

1.2. Summary of Contributions

We summarize the contributions of this work as follows.

(1) We propose an approach to jointly summarize large sets of online images and user videos in a mutually-rewarding way. Our method creates not only high-quality video summary but also a novel structural summary of online images as *storyline graphs*, which can visualize various events or activities associated with the topic in a form of branching networks. To the best of our knowledge, our work is the first attempt so far to leverage both online images and videos for reconstructing the storyline graphs.

(2) We develop an approach for video summarization and storyline reconstruction, which can address several key challenges of large-scale nature of our problems, including optimality guarantee, linear complexity, and easy parallelization. With experiments on large-scale Flickr and YouTube datasets and crowdsourcing based evaluations using Amazon Mechanical Turk, we show the superiority of our approach over other candidate methods for both summarization tasks.

2. Problem Setting

Input: The input to our algorithm is a set of photo streams $\mathcal{P} = \{P^1, \cdots, P^L\}$ and a set of videos $\mathcal{V} = \{V^1, \cdots, V^N\}$, for a topic class of interest. L and N indicate the number of input photo streams and videos, respectively. Each photo stream, denoted by $P^l = \{p_1^l, \cdots, p_{L^l}^l\}$, is a set of photos taken in sequence by a single photographer within a fixed period of time $[0, T]$ (*i.e.* single day in this paper). We assume that each image p_i^l is associated with a timestamp t_i^l, and each photo stream is sorted by the timestamp. We uniformly sample videos into a set of frames, every 0.5 sec, which is denoted by $V^n = \{v_1^n, \cdots, v_{N^n}^n\}$. As a notation convention, we use superscripts to denote photo streams/videos and subscripts to denote images/frames.

Output: The output of our algorithm is two-fold. The first output is the summary S^n of every video $V^n \in \mathcal{V}$ (*i.e.* $S^n \subset V^n$). We pursue keyframe-based summarization (*e.g.* [9]), in which we choose α^n most representative but discriminative keyframes out of all frames $V^n = \{v_1^n, \cdots, v_{N^n}^n\}$. In our algorithm, α^n is automatically chosen according to the contents of video V^n. The second output is the storyline graphs $\mathcal{G} = (\mathcal{O}, \mathcal{E})$. Conceptually, the vertices $\mathcal{O}$ correspond to dominant image clusters across the dataset, and the edge set $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$ connects the vertices that sequentially recur in many photo streams and videos. More rigorous mathematical definition will be given in Section 4.

Image Description and Similarity Measure: In order to capture various visual information, we apply three different features extraction methods to images and frames of videos. We densely extract HSV color SIFT and histogram of oriented edge (HOG) feature on a regular grid of each image/frame at steps of 4 and 8 pixels, respectively. We also obtain the *Tiny* image feature [23], which is RGB values of a 32×32 resized tiny image. Then, we build an L_1-normalized three-level spatial pyramid histogram for each feature type, and concatenate them into a singe vector denoted by $\mathbf{v}$. For similarity measure σ, we use the histogram intersection. The three descriptors are equally weighted.

K-NN graphs between photo streams and videos: Due to the extreme diversity of the Web images and videos even associated with the same keyword, we build K-nearest graphs between $\mathcal{P}$ and $\mathcal{V}$ so that only sufficiently similar photo streams and videos help summarize one another.

For each photo stream $P^l \in \mathcal{P}$, we find K_P-nearest videos using the similarity calculated by Naive-Bayes Nearest-Neighbor method [2] as follows. Given a photo streams P^l and a video V^n, for each image $p \in P^l$, we obtain the first nearest neighbor in V^n denoted by $\text{NN}(p)$. The similarity from P^l to V^n is computed by $\sum_{p \in P^l} \|\sigma(p, \text{NN}(p))\|^2$. We let $\mathcal{N}(P^l)$ be the K_P-nearest videos to P^l. Likewise, we run the same procedure to obtain K_V-nearest photo streams $\mathcal{N}(V^n)$ for each video V^n.

3. Video Summarization

In this section, we discuss how to summarize each video $V^i \in \mathcal{V}$ by leveraging a large set of images. For summarization of a video V^i, we first build a similarity graph between frames in V^i and images of photo streams $\mathcal{N}(V^i)$, as shown in Fig.1.(b). We first model consecutive frames of V^i as k-th order Markov chain. Next, each image in $\mathcal{N}(V^i)$ casts similarity votes by connecting with its k_P-nearest frames with the weight of feature similarity. Since most of shared images online are carefully taken by photographers who try to express his/her experiences and intents to be as clear as possible, even simple similarity voting by a crowd of images can discover semantically meaningful keyframes, which will be demonstrated in the experiments (Section 5).

Once we build the graph $\mathcal{G}_V^i = (\mathcal{U}^i, \mathcal{E}^i)$ where the node set $\mathcal{U}^i = V^i \cup \mathcal{N}(V^i)$, we select α^i keyframes as a summary of V^i using the diversity ranking algorithm proposed in [11], which is formulated as a temperature maximization with α^i number of heat sources. Intuitively, the sources should be located in the nodes that are densely connected to other nodes with high edge weights. At the same time, the sources should be sufficiently distant from one another because nearby nodes to the sources already earn high temperatures. We let $\mathbf{G}^i$ be the adjacency matrix of $\mathcal{G}_V^i$. In order to model the heat dissipation, a ground node g is connected to all nodes with a constant dissipation conductance z (*i.e.* appending an $|\mathbf{G}^i| \times 1$ column $\mathbf{z}$ to the end of $\mathbf{G}^i$). The optimization of α^i keyframe selection can be expressed

by the equation below:

$$\max \sum_{x \in \mathcal{U}^i} u(x) \tag{1}$$

$$\text{s.t. } u(x) = \frac{1}{d_x} \sum_{(x,y) \in \mathcal{E}^i} \mathbf{G}(y,x) u(x) \text{ for } d_x = \sum_{(x,y) \in \mathcal{E}^i} \mathbf{G}(y,x)$$

$$u(g) = 0, \;\; u(s) = 1 \text{ for } s \in S^i \subset V^i, |S^i| \leq \alpha^i,$$

where $u(x)$ is the temperature at x and d_x is the degree of x. The first constraint describes the temperature of each node observes the diffusion law. The second constraint tells the temperature of ground and heat sources are 0 and 1, respectively. S^i is the set of α^i selected keyframes. In [11], the objective of Eq.(1) is proved to be submodular, and thus we can compute a constant factor approximate solution by a simple greedy algorithm, which starts with an empty S^i and iteratively adds the image s that maximizes the marginal temperature gain, $\Delta U = U(S^i \cup \{s\}) - U(S^i)$, where $U(S^i) = \sum_{x \in \mathcal{U}^i} u(x)$ when sources are located in S^i. In our approach, we keep increasing α^i until the marginal temperature gain ΔU is below the threshold.

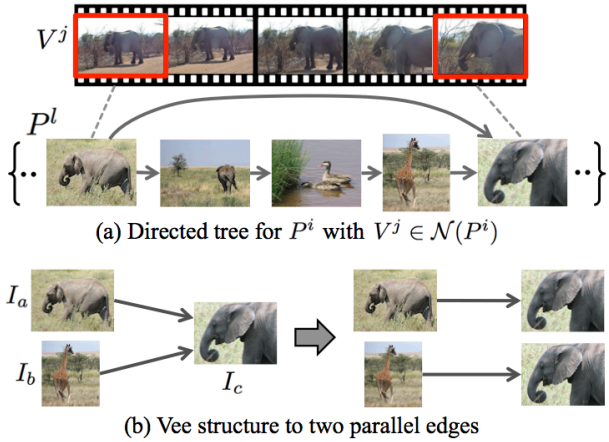
4. Photo Storyline Reconstruction

In this section, we discuss the reconstruction of a storyline graph $\mathcal{G} = (\mathcal{O}, \mathcal{E})$ from a set of photo streams $\mathcal{P}$ with assistance of the video set $\mathcal{V}$.

4.1. Definition of Storyline Graphs

Definition of Vertices: Since the image sets are large and ever-growing, and much of images are highly overlapped, it is inefficient to build a storyline graph over individual images. Hence, the vertices $\mathcal{O}$ are preferentially defined as *image clusters*. For each descriptor type j, we construct D_j visual clusters ($D_j = 600$) by applying the K-means to randomly sampled descriptors. That is, we can obtain J different views of storyline graphs for a given dataset ($J = 3$ in our case), and each image can be represented as J vectors $\mathbf{x}^{(j)} \in \mathbb{R}^{D_j}$ with only one nonzero value indicating its cluster membership (*i.e.* identically as a single vector $\mathbf{x} \in \mathbb{R}^D$ by concatenating all J vectors). In addition, we can extend the model by allowing soft assignment in which an image is associated with c multiple clusters with weights.

Definition of Edges: In our approach, we let the edge set $\mathcal{E}$ satisfy the following two properties [12, 21]. (i) $\mathcal{E}$ should be *sparse*. The *sparsity* is encouraged in order to avoid an unnecessarily complex narrative structure; instead we retain only a small number of strong story branches per node. (ii) $\mathcal{E}$ should be *time-varying*. That is, $\mathcal{E}$ smoothly changes over time in $t \in [0, T]$, because the popular transitions between images vary over time. For example, in the *snowboarding* photo streams, the *skiing* images may be followed by *lunch* images around noon but by *sunset* images in the evening.



(a) Directed tree for P^i with $V^j \in \mathcal{N}(P^i)$

(b) Vee structure to two parallel edges

Figure 2. We build the directed tree $\mathcal{T}^l$ for a photo stream P^l with its nearest videos $\mathcal{N}(P^l)$. (a) First, the images in P^l are represented by k-th order Markov chain. Then, additional links are connected based on one-to-one correspondences between keyframes of V^j with images in P^l. (b) Since the vee structure is an impractical artifact, it is replaced by two parallel edges.

Based on the two requirements, we obtain a set of time-specific $\{\mathbf{A}^t\}$ for $t \in [0, T]$, where $\mathbf{A}^t$ is the adjacency matrix of $\mathcal{E}^t$. Although we can compute $\mathbf{A}^t$ at any point t, in practice, we uniformly split $[0, T]$ into τ time points (*e.g.* every 30 minutes), at which $\mathbf{A}^t$ is estimated. In addition, we penalize nonzero elements of each $\mathbf{A}^t$ for sparsity.

4.2. Modeling of Storyline Graphs

In this section, we formulate a maximum likelihood estimation for inferring the storyline graph.

Our first step is to represent each photo stream P^l as a directed tree $\mathcal{T}^l$ based on the sequential relevance. Our underlying idea is that the consecutive images in a photo stream are loosely sequential, but their links are not as strong as those between the frames of a video. As shown in a toy example of Fig.2, we first represent a photo stream P^l as a k-th order Markov chain. Next, for each neighbor video $V^j \in \mathcal{N}(P^l)$, we choose α keyframes using the algorithm in section 3. (*i.e.* $\alpha \propto$ (length of a video)). We find one-to-one bipartite matching between the selected frames and the images in P^l using Hungarian algorithm. Then, we additionally connect any pairs of images in P^l that are linked by consecutive frames in V^j. We assign edge weights by similarity between images. Finally, as shown in Fig.2.(b), we replace any *vee* structure with two parallel edges after copying image I_c because it is an impractical artifact. That is, the vee structure occurs only because both $I_a \rightarrow I_c$ and $I_b \rightarrow I_c$ are observed in P^l or bypaths via $\mathcal{N}(P^l)$, not because I_c appears only after both I_a and I_b occur.

In the current formulation, videos are used only for discovering the edges of storyline graphs, and do not contribute to the definition of vertices. This is due to that we here limit the storyline graphs as a structural summary of online im-

ages. However, it is straightforward to include video frames for the node construction without modifying the algorithm.

We derive our model from the likelihood $f(\mathcal{P})$ of an observed set of photo streams $\mathcal{P} = \{P^1, \cdots, P^L\}$. Note that each image p_i^l in photo stream P^l is associated with $\mathbf{x}_i^l$ and timestamp t_i^l. The likelihood $f(\mathcal{P})$ is defined as follows.

$$f(\mathcal{P}) = \prod_{l=1}^{L} f(P^l), \text{ where } f(P^l) = \prod_{\mathbf{x}_i^l \in P^l} f(\mathbf{x}_i^l, t_i^l | \mathbf{x}_{p(i)}^l, t_{p(i)}^l) \quad (2)$$

where $\mathbf{x}_{p(i)}^l$ and $t_{p(i)}^l$ denote the descriptor vector and timestamp of the parent of $\mathbf{x}_i^l$ in the directed tree $\mathcal{T}^l$. Note that without the vee structure, each image has only one parent. For the transition model $f(\mathbf{x}_i^l, t_i^l | \mathbf{x}_{p(i)}^l, t_{p(i)}^l)$, we use the *linear dynamics model*, as one of the simplest transition models for dynamic Bayesian networks (DBN)

$$\mathbf{x}_i^l = \mathbf{A}_e \mathbf{x}_{p(i)}^l + \boldsymbol{\epsilon}, \quad \text{where } \boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \sigma^2 \mathbf{I}) \quad (3)$$

where $\boldsymbol{\epsilon}$ is a vector of Gaussian noise with zero mean and variance σ^2. In order to model temporal information between $t_{p(i)}^l$ and t_i^l as well, we use the *exponential* rate function that has been widely used to represent temporal dynamics of diffusion networks [19]: the (x, y) element a_{xy} of $\mathbf{A}_e$ has the form of $\alpha_{xy} \exp(-\alpha_{xy}\Delta)$ where $\Delta = |t_i^l - t_{p(i)}^l|$ and α_{xy} is the transmission rate from codeword x to y. Note that $\alpha_{xy} \geq 0$. As $\alpha_{xy} \to 0$, the consecutive occurrence from codeword x to y is very unlikely. By letting $\mathbf{A} = \{\alpha_{xy} \exp(\alpha_{xy})\}_{D \times D}$, we have $\mathbf{A}_e = g_i \mathbf{A}$ with $g_i = \exp(-\Delta)$, which is computed for each training data.

For better scalability, we impose a practically reasonable assumption on the transition model: *Each codeword of* $\mathbf{x}_i^l$ *is conditionally independent of another given* $\mathbf{x}_{p(i)}^l$. That is, the transition likelihood factors over individual codewords: $f(\mathbf{x}_i^l, t_i^l | \mathbf{x}_{p(i)}^l, t_{p(i)}^l) = \prod_{d=1}^{D} f(x_{i,d}^l, t_i^l | \mathbf{x}_{p(i)}^l, t_{p(i)}^l)$. Consequently, from Eq.(3), we can express the transition likelihood as Gaussian distribution: $f(x_{i,d}^l, t_i^l | \mathbf{x}_{p(i)}^l, t_{p(i)}^l) = \mathcal{N}(x_{i,d}^l; g_i \mathbf{A}_{d*} \mathbf{x}_{p(i)}^l, \sigma^2)$, where $\mathbf{A}_{d*}$ denotes the d-th row of the matrix $\mathbf{A}$. Finally, the log-likelihood $\log f(\mathcal{P})$ in Eq.(2) can be written

$$\log f(\mathcal{P}) = -\sum_{l=1}^{L} \sum_{i \in P^l} \sum_{d=1}^{D} f(x_{i,d}^l) \quad \text{where} \quad (4)$$

$$f(x_{i,d}^l) = \left(\frac{N^l}{2} \log(2\pi\sigma^2) + \frac{1}{2\sigma^2} (x_{i,d}^l - g_i \mathbf{A}_{d*} \mathbf{x}_{p(i)}^l)^2 \right)$$

4.3. Optimization

Our optimization problem is to discover nonzero elements of $\mathbf{A}^t$ for any $t \in [0, T]$, by maximizing the log-likelihood of Eq.(4). For statistical tractability and scalability, we take advantage of the constraints and the assumption described in previous section.

First, one difficulty during optimization is that for a fixed t, the estimator may suffer from high variance due to the scarcity of training data (*i.e.* images occurring at time t). In order to overcome this, we take advantage of the constraint that $\mathbf{A}^t$ *varies smoothly across time*; thus, we can estimate $\mathbf{A}^t$ by re-weighting the observation data near t accordingly. Second, thanks to the conditional independence assumption per codeword dimension, we can reduce the inference of $\mathbf{A}^t$ to a *neighborhood selection*-style optimization [15], which enables to estimate the graph by *independently* solving a set of atomic weighted lasso problem for each dimension of codewords d while guaranteeing asymptotic consistency. Hence, the optimization becomes trivially parallelizable per dimension. Such property is of particular importance in our problem possibly using millions of images. Finally, we encourage a sparse solution by penalizing nonzero elements of $\mathbf{A}^t$. As a result, we estimate $\mathbf{A}^t$ by iteratively solving the following optimization D times:

$$\widehat{\mathbf{A}}_{d*}^t = \operatorname{argmin} \sum_{l=1}^{L} \sum_{i \in P^l} w^t(i) (x_{i,d}^l - g_i \mathbf{A}_{d*}^t \mathbf{x}_{p(i)}^l)^2 + \lambda \|\mathbf{A}_{d*}^t\| \quad (5)$$

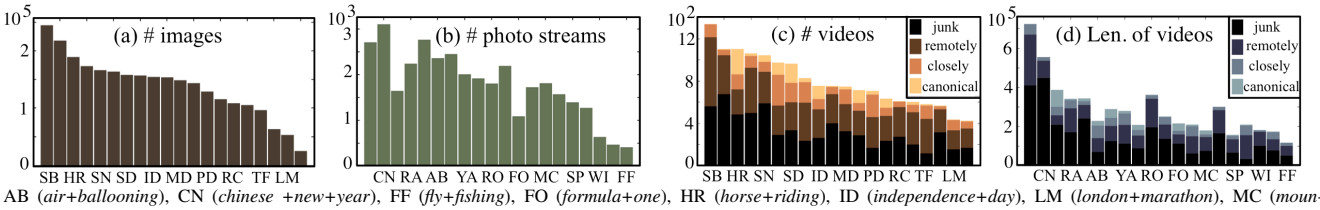
where $w^t(i)$ is the weighting of an observation of image p_i^l in photo stream l at time t. That is, when the timestamp of p_i^l (*i.e.* t_i^l) is close to t, $w^t(i)$ is large so that the observation contributes more on the graph estimation at t. Naturally, we can define $w^t(i) = \frac{\kappa_h(t - t_i^l)}{\sum_{l=1}^{L} \sum_{i=2}^{N_l} \kappa_h(t - t_i^l)}$ where $\kappa_h(u)$ is Gaussian RBF kernel with a kernel bandwidth h (*i.e.* $\kappa_h(u) = \exp(-u^2/2h^2)/\sqrt{2\pi}h$).

In Eq.(5), we include ℓ_1-regularization where λ is a parameter that controls the sparsity of $\widehat{\mathbf{A}}_{d*}^t$. It not only avoids overfitting but also is practically useful because the story branches at each node are simple enough to be easily understood, with only a small number of strong story links. Consequently, our graph inference reduces to iteratively solving a standard weighted ℓ_1-regularized least square problem, whose global optimum solution can be attained by highly scalable techniques such as coordinate descent [7]. In summary, the graph inference can be performed in a linear time with respect to all parameters, including the number of images and the number of codewords D. We present the more details of the algorithm including the pseudocode in the supplementary material.

As of now, we perform the *structure learning* to discover the topology of the storyline graph (*i.e.* nonzero elements of $\{\mathbf{A}^t\}$). We can run the *parameter learning* (*i.e.* estimating actual associated weights) while fixing the topology of the graph. Since the structure of each graph is known and all photo streams are independent of one another, we can trivially solve for MLE of $\widehat{\mathbf{A}}^t$, which is similar to that of the transition matrix of k-th Markovian chains.

5. Experiments

We evaluate the proposed approach from two technical perspectives: video summarization in section 5.1 and image summarization as storylines in section 5.2.



AB (*air+ballooning*), CN (*chinese +new+year*), FF (*fly+fishing*), FO (*formula+one*), HR (*horse+riding*), ID (*independence+day*), LM (*london+marathon*), MC (*mountain+camping*), MD (*memorial+day*), PD (*st+patrick+ day*), RA (*rafting*), RC (*rock+climbing*), RO (*rowing*), SB (*surfing+beach*), SD (*scuba+diving*), SN (*snowboarding*), SP (*safari+park*), TF (*tour+de+france*), WI (*wimbledon*), YA (*yacht*).

Figure 3. The Flickr/YouTube datasets of 20 outdoor recreational classes. (a)–(b) The number of images and photo streams of Flickr dataset: (2,769,504, 35,545). (c)–(d) The number and total length of YouTube videos: (15,912, 1,586.8 hours).

Flickr/YouTube dataset: Fig.3.(a)–(b) summarize our Flickr dataset of 20 outdoor recreational activity classes that consists of about 2.7M images from 35K photo streams. Some classes are re-used from the datasets of [10], and the others are newly downloaded using the same crawling method, in which the topic names are used as search keywords and all queried photo streams of more than 30 images are downloaded without any filtering.

Fig.3.(c)–(d) show the statistics of our YouTube datasets with about 16K user videos. We query the same topic keywords using YouTube built-in search engines, and download only the Creative Commons licensed videos. In addition, since YouTube user videos are extremely noisy, we manually rate them into one of four categories: *canonical*, *closely/remotely related*, and *junk*. These labels are not used by the algorithms but for the groundtruth labeling only.

5.1. Results on Video Summarization

Tasks: Due to the large-scale nature of our problems, we obtain groundtruth labels via crowdsourcing using Amazon Mechanical Turk (AMT), inspired by [9]. For each topic class, we randomly sample 100 test videos that are rated as *canonical* or *closely-related*. Then, we uniformly sample 50 frames from each test video, and ask a turker to select 5$\sim$10 images that must be included if he wants to make a storyline summary. We obtain such summarization of each test video from at least five different turkers for the validity of the groundtruth. We run our algorithm and baselines to select a small number of keyframes as a summary of each test video. The performance of algorithms is measured by comparing between the groundtruth labels and selected keyframes. We compute the similarity-based average precision (AP) proposed in [9]. We defer the detail of how to compute the average precision to the supplementary.

Baselines: We select four baselines based on the recent studies of video summarization [9, 13, 14]. The (`Unif`) samples α keyframes uniformly from each test video. The (`KMean`) and the (`Spect`) are the two popular clustering methods, K-means and spectral clustering, respectively. They first create α clusters and select the images closest to the cluster centers. The (`RankT`) is one of state-of-the-art keyframe extraction methods using the rank-tracing technique [1]. The (`OursV`) is our discriminative ranking method described in section 3 without involving similarity votes by images. It is compared with our fully-geared algorithm (`OursIV`) in order to justify the usefulness of joint summarization between images and videos.

Results: Fig.4 reports the average precisions of our algorithms and baselines across 20 activity classes. Our algorithm significantly outperforms all the baselines in most classes. (*e.g.* The average AP of the (`OursIV`) is **0.0893**, which is notably higher than 0.0808 of the best baseline (`KMean`) by 9.5%. The performance of the (`KMean`) and the (`Spect`) highly depends on the number of clusters α. We change α from 5 to 25, and report the best results.

Fig.5 compares video summarization results produced by different methods. The (`Unif`) cannot correctly handle different lengths of subshots in a single video (*i.e.* redundant images can be selected from long subshots while none from interesting short ones). One practical shortcoming of the (`KMean`) and the (`Spect`) is that it is hard to know the best α beforehand even though the performance highly depends on α. Overall, all algorithms except the (`OursIV`) suffer from the limitations of using low-level features only. For example, as shown in Fig.5.(a), the (`OursV`) and the (`KMean`) detect meaningless completely-gray *sky* frames in 3rd and 5th column, respectively. Such frames with no semantic meaning occur frequently in user videos, whereas very few in the image sets. Therefore, even though (`OursIV`) uses the same low-level features, it can easily suppress such unimportant information thanks to the similarity votes produced by images that photographers take more carefully with sufficient semantic intent and value[1].

5.2. Results on Photo Storyline Summarization

Task: The quantitative evaluation on the storyline reconstruction is inherently difficult because there is no groundtruth available. Moreover, it is painfully overwhelming for a human labeler to evaluate the storylines summarized from large sets of images. (*e.g.* Given multiple storyline graphs with hundreds of nodes created from millions of images, a human labeler may feel hopelessly devastated to judge which one is better). In order to overcome such inherent difficulty of the storyline evaluation, we design the

[1] Unfortunately, such semantic significance is not fully evaluated by the AP metric of Fig.4, which is solely based on low-level feature differences.

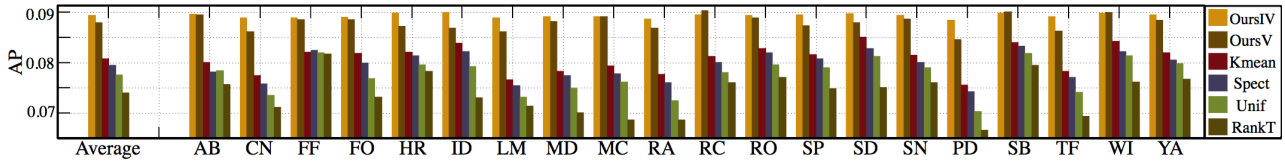


Figure 4. Comparison of average precisions (APs) between our methods (`OursIV`) and (`OursV`) and the baselines (`Unif`), (`KMean`), (`Spect`), and (`RankT`). The acronyms of activities are referred to Fig.3. The leftmost bar set shows the average APs for all classes. (`OursIV`): **0.0893**, (`OursV`): 0.0880, (`Unif`): 0.0776, (`KMean`): 0.0808 (`Spect`): 0.0795, and (`RankT`): 0.0740.

Figure 5. Qualitative comparison of video summarization results. From top to bottom, we show AMT groundtruth and the same number of selected keyframes by our algorithms (with and without similarity voting by images), and two baselines (`KMean`) and (`Unif`).

following evaluation task via crowdsourcing.

We first run our algorithms and baselines to generate storyline graphs from the dataset of each class. We then sample 100 canonical images on the timeline as test instances $\mathcal{I}_Q$. Based on the reconstructed storyline, each algorithm can retrieve one image that is most likely to come next after each test image $I_q \in \mathcal{I}_Q$. (*i.e.* We first identify which node corresponds to the I_q, and follow the most strongly connected edge to the next likely node, from which the central image is retrieved). For evaluation, a turker is shown the test image I_q, and then a pair of images predicted by our algorithm and one of baselines, and asked to choose the one that is more likely to follow I_q than the other. We design the AMT task as a pairwise comparison instead of a multiple-choice question (*i.e.* selecting the best one among the outputs of all algorithms). We obtain such pairwise comparison for each of $\mathcal{I}_Q$ from at least three different turkers. In summary, the underlying idea of our evaluation is that we let a crowd of labelers, each of whom evaluates only a basic unit (*i.e. an important edge* of the storyline), instead of the assessment of the whole storyline, which is practically impossible.

Baselines: We compare three baselines with our approach. The first (`Page`) is a Page-Rank based image retrieval that simply selects the top-ranked image around the timestamp of I_q. It is compared to show that the sequential summary as storylines can be more useful than the traditional retrieval method. The (`HMM`) is an HMM based method that has been popularly applied for modeling tourists' sequential photo sets [3]. This comparison can tell the importance of our branching structure over the linear storyline of the (`HMM`). The (`Clust`) is a simple clustering-based summarization on the timeline [10], in which images are distributed on the timeline of 24 hours, and grouped into 10 clusters at every 30 minutes. We also compare with the our algorithm using images only, denoted by (`OursI`), in order to quantify the improvement by joint summarization with videos. We present more details of application of our algorithm and baselines in supplementary material.

Results: Fig.6 show the results of pairwise preference tests obtained via AMT between our algorithm and each baseline. The number indicates the mean percentage of responses that choose our prediction as a more likely one to come next after each I_q than that of the baseline. That is, the number should be higher than at least 50% to validate the superiority of our algorithm. Even considering a certain degree of unavoidable noisiness of AMT labels, our output is significantly preferred by AMT annotators. For example, our algorithm (`OursVI`) gains 75.9% of votes, far outdistancing the best baseline (`HMM`). Importantly, more than two thirds of responses (*i.e.* 67.9%) prefer the results of the (`OursVI`) over those of the (`OursI`), which indeed supports our argument that a crowd of videos help improve the quality of the storylines from users' point of view.

Fig.7 illustrates another interesting qualitative comparison between our method and baselines. Given a pair of images that are distant in a novel photo stream (*i.e.* images with red boundaries in Fig.7.(a)), each algorithm predicts 10 images that are likely to occur between them using its own storyline graph (*i.e.* each algorithm finds out the *best* path between the two images). As shown in Fig.7.(a), our algorithm (in the second row) can retrieve the images that are very similar to the hidden groundtruths (in the first row). Using the iterative Viterbi algorithm, the (`HMM`) retrieves reasonably good but highly redundant images, which

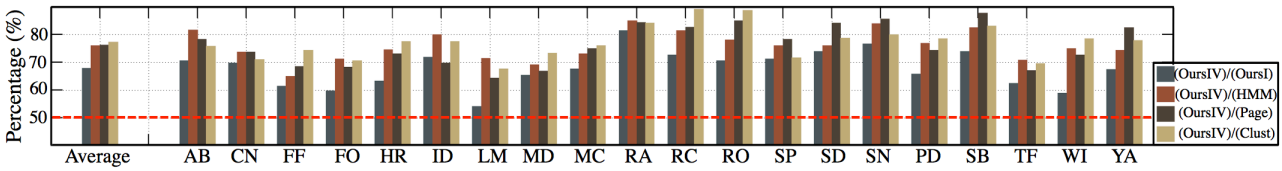


Figure 6. The results of pairwise preference tests between our method (OursIV) and each baseline via Amazon Mechanical Turk. The numbers indicates the percentage of responses that our prediction is more likely to occur next after I_q than that of the baseline. At least the number should be higher than 50% (shown in red dotted line) to validate the superiority of our algorithm. The leftmost bar set shows the average preference of our (OursIV) for all 20 classes: [**67.9**, **75.9**, **76.1**, **77.1**] over (OursV), (HMM) , (Page) , and (Clust).

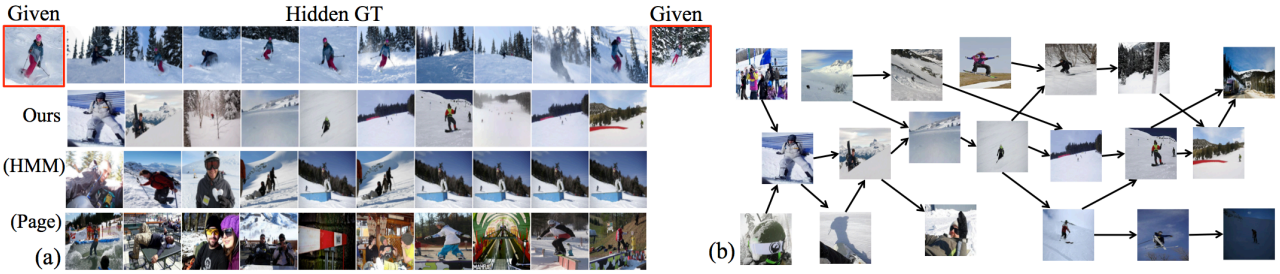


Figure 7. Examples of an qualitative comparison between our method and baselines. (a) Given a pair of distant images in a photo stream (*i.e.* the ones with red boundaries), each algorithm predicts the best path between them, and samples 10 images. (b) A downsized version of our storyline graph that is used for the prediction of (a).

are in part due to its inability to represent various branching structures. The (Page) retrieves top-ranked images (*i.e.* representative and high-quality images) at each query time point. However, it has no use of the sequential structure, and thus there is no connected story between retrieved images. Fig.7.(b) shows a downsized version of our storyline graph that is used for creating the result of Fig.7.(a). Although we can freely choose the temporal granularity to zoom in or out the storylines, we here show only a small part of them for better visibility. We present more illustration examples of storyline graphs in the supplementary.

6. Conclusion

In this paper, we proposed a scalable approach to jointly summarize large-scale Flickr images and YouTube videos, and created a novel structural summary as storyline graphs visualizing a variety of underlying narrative branches of topics. We validated the superior performance of our approach via the evaluation using Amazon Mechanical Turk.

References

[1] W. Abd-Almageed. Online, Simultaneous Shot Boundary Detection and Key Frame Extraction for Sports Videos Using Rank Tracing. In *ICIP*, 2008. 6

[2] O. Boiman, E. Shechtman, and M. Irani. In Defense of Nearest-Neighbor Based Image Classification. In *CVPR*, 2008. 3

[3] C.-Y. Chen and K. Grauman. Clues from the Beaten Path: Location Estimation with Bursty Sequences of Tourist Photos. In *CVPR*, 2011. 7

[4] C. Y. Chen and K. Grauman. Watching Unlabeled Video Helps Learn New Human Actions from Very Few Labeled Snapshots. In *CVPR*, 2013. 2

[5] L. Chen, L. Duan, and D. Xu. Event Recognition in Videos by Learning from Heterogeneous Web Sources. In *CVPR*, 2013. 2

[6] J. Deng, W. Dong, R. Socher, L. J. Li, K. Li, and L. Fei-Fei. ImageNet: A Large-Scale Hierarchical Image Database. In *CVPR*, 2009. 2

[7] W. J. Fu. Penalized Regressions: The Bridge Versus the Lasso. *J. Computational Graphical Statistics*, 7:397–416, 1998. 5

[8] A. Gupta, P. Srinivasan, J. Shi, and L. S. Davis. Understanding Videos, Constructing Plots: Learning a Visually Grounded Storyline Model from Annotated Videos. In *ICCV*, 2009. 2

[9] A. Khosla, R. Hamid, C. J. Lin, and N. Sundaresan. Large-Scale Video Summarization Using Web-Image Priors. In *CVPR*, 2013. 2, 3, 6

[10] G. Kim and E. P. Xing. Jointly Aligning and Segmenting Multiple Web Photo Streams for the Inference of Collective Photo Storylines. In *CVPR*, 2013. 2, 6, 7

[11] G. Kim, E. P. Xing, L. Fei-Fei, and T. Kanade. Distributed Cosegmentation via Submodular Optimization on Anisotropic Diffusion. In *ICCV*, 2011. 3, 4

[12] M. Kolar, L. Song, A. Ahmed, and E. P. Xing. Estimating Time-Varying Networks. *Ann. Appl. Stat.*, 4(1):94–123, 2010. 4

[13] Y. J. Lee, J. Ghosh, and K. Grauman. Discovering Important People and Objects for Egocentric Video Summarization. In *CVPR*, 2012. 2, 6

[14] Z. Lu and K. Grauman. Story-Driven Summarization for Egocentric Video. In *CVPR*, 2013. 2, 6

[15] N. Meinshausen and P. Bühlmann. High-Dimensional Graphs and Variable Selection with the Lasso. *Ann. Statist.*, 34(3):1436–1462, 2006. 5

[16] H. Misra, F. Hopfgartner, A. Goyal, P. Punitha, and J. M. Jose. TV News Story Segmentation Based on Semantic Coherence and Content Similarity. In *MMM*, 2010. 2

[17] A. Prest, C. Leistner, J. Civera, C. Schmid, and V. Ferrari. Learning Object Class Detectors from Weakly Annotated Video. In *CVPR*, 2012. 2

[18] R. Raguram and S. Lazebnik. Computing Iconic Summaries of General Visual Concepts. In *CVPR Workshop Internet Vision*, 2008. 2

[19] M. G. Rodriguez, D. Balduzzi, and B. Schölkopf. Uncovering the Temporal Dynamics of Diffusion Networks. In *ICML*, 2011. 5

[20] I. Simon, N. Snavely, and S. M. Seitz. Scene Summarization for Online Image Collections. In *ICCV*, 2007. 2

[21] L. Song, M. Kolar, and E. Xing. Time-Varying Dynamic Bayesian Networks. In *NIPS*, 2009. 4

[22] K. Tang, V. Ramanathan, L. Fei-Fei, and D. Koller. Shifting Weights: Adapting Object Detectors from Image to Video. In *NIPS*, 2012. 2

[23] A. Torralba, R. Fergus, and W. T. Freeman. 80 Million Tiny Images: A Large Data Set for Nonparametric Object and Scene Recognition. *IEEE PAMI*, 30:1958–1970, 2008. 3