Discovering Collective Narratives of Theme Parks from Large Collections of Visitors' Photo Streams

Gunhee Kim*
Seoul National University
Seoul, South Korea, 151-744
gunhee@snu.ac.kr

Leonid Sigal
Disney Research
Pittsburgh PA 15213
lsigal@disneyresearch.com

ABSTRACT

We present an approach for generating pictorial storylines from large collections of online photo streams shared by visitors to theme parks (*e.g.* Disneyland), along with publicly available information such as visitor's maps. The story graph visualizes various events and activities recurring across visitors' photo sets, in the form of hierarchically branching narrative structure associated with attractions and districts in theme parks. We first estimate *story elements* of each photo stream, including the detection of faces and supporting objects, and attraction-based localization. We then create spatio-temporal story graphs via an inference of sparse time-varying directed graphs. Through quantitative evaluation and crowdsourcing-based user studies via Amazon Mechanical Turk, we show that the story graphs serve as a more convenient mid-level data structure to perform photo-based recommendation tasks than other alternatives. We also present storybook-like demo examples regarding exploration, recommendation, and temporal analysis, which may be most beneficial uses of the story graphs to visitors.

Categories and Subject Descriptors

I.4.9 [**Image processing and computer vision**]: Applications

Keywords

Photo storylines; summarization and exploration of multimedia data; user modeling

1. INTRODUCTION

Current technical advances, including widespread availability of mobile photo taking devices and ubiquitous network connectivity, are changing the way we tell our stories. Personal storytelling is becoming more *data-driven* and *vision-oriented*; people can simply capture their memorable moments by a stream of images, and then spontaneously reuse them to deliver their own stories. In addition, with the emergence of social networking, the sharing of such visual stories is becoming effortless. For example, even in a single day, tens of thousands of people visit *Disneyland*, and much of them take and are willing to share large streams of photos that record their experiences with families or friends. Each photo stream tells a slightly different story from its own point of view, but by aggregating them, it is likely that common storylines of *Disneyland* experience emerge. The storylines can summarize a variety of visitors' activity patterns such as popular paths in the theme parks and temporal changes of the flow in and out of the attractions.

In this paper, the term *story* refers to *personal* or *collective narrative*, instead of a *fictional story* as is the case in novels, games, or films. The *personal narrative* describes accounts of specific events that have been personally experienced [9], whereas the *collective narrative* is regarded as a collection of *personal narratives* from different people that share the similar experiences. We consider a photo stream of each visitor as a visual instantiation of his or her personal narrative, and the aggregation of photo streams as that of collective narrative, assuming that photographers take pictures when they encounter scenes or events that they want to remember or tell a story about. We define a photo stream as a set of images that are taken in sequence by a single photographer within a fixed period of time (*e.g.* one day).

Fig.1 summarizes the problem statement. Our goal is to develop an approach for creating and exploring spatio-temporal storylines from large collections of photo streams contributed by visitors to *Disneyland*, along with its public information like visitor's map. We also use meta-data of images such as timestamps and GPS information if available, although they are often noisy and missing (*e.g.* only 19% of photo streams in our dataset has GPS information). Taking advantage of computer vision techniques, we represent the visual contents of images in the form of *story elements* (*e.g.* human faces, supporting objects, and locations), and automatically extract shared key moments and put them together to create a *story graph*, which is a structural summary of branching narratives that visualize various events and/or activities recurring across the input photo sets. To show the usefulness of story graphs, we leverage them to perform photo-based exploration, recommendation, and temporal analysis tasks. For example, once we have story graphs summarizing people's experiences on their *Disneyland* trips, we can recommend pieces of those experiences to new visi-

*This work has been completed when Gunhee Kim was a postdoctoral researcher at Disney Research.

KDD'15, August 10-13, 2015, Sydney, NSW, Australia.

DOI: http://dx.doi.org/10.1145/2783258.2788569.

Figure 1: (a) Given large sets of online photo streams shared by visitors to a theme park (*e.g.* Disneyland), and its public information (*e.g.* visitor's map), we create story graphs as a structural summary visualizing various events or activities recurring across the visitors' photo sets. The story graphs are hierarchically created to enable users to explore the stories between attractions in (b), and sub-stories inside the individual attractions in (c)–(d). In (c), we show the connectivity between story subgraphs at the *Enchanted Tiki Room*, and their details are given in (d). The photos in (c) are the first images of story subgraphs in (d).

tors, suggesting attractions, touring plans, or best ways to experience a given attraction at a particular time of the day.

As a problem domain, we focus on *theme parks*, more specifically *Disneyland*, for the following reasons. First, we can easily obtain abundant visual data, because visiting a theme park is a special leisure activity where much of visitors are willing to take and share the pictures of their experiences[1]. Second, stories play an important role in theme parks for the purpose of user modeling, personalization, and recommendation. Theme parks provide different sets of attractions and entertainment, only parts of which are experienced by individual visitors. Thus a large diversity of possible stories exist according to who visits when. Families with children may prefer to play with the characters their kids like. Such storylines may differ from those of young adults who may enjoy more adventurous rides (*e.g.* roller coasters). Finally, our story extraction and exploration in theme parks can be easily extended to other leisure and entertainment domains, such as city tours [15] and museum tours [30], as long as they have sufficient sets of photo streams. Similar to theme parks, cities also consist of multiple attractions that are more sparsely distributed and events, attendance of which differs by preference, time and visitor types.

Our story graphs can be used as mid-level data structure for a variety of applications, including virtual exploration, path planning, travel recommendations, and temporal mining tasks, among many others. Usually theme park operators, including Disney, do not disclose any statistics on the use of attractions. Thus, our story graphs can indirectly hint the visitors' behavior patterns. Compared to guided routes and lists of events by travel agencies, our visual exploration using story graphs directly builds upon consumer-generated pictures, and thus reflects candid and spontaneous peer costumers' experiences in the original visual forms. It could be more synergetic to integrate our results with the official Apps of theme parks (*e.g.* *My Disney Experience* mobile app of *Disneyland*). Not only for visitors, our story graphs can benefit theme park operators by providing an automatic tool for monitoring the popularities of attractions. We also focus on revealing in-depth branching stories at each attraction, which are not easily mined from other sensor modalities. For example, although a GPS tracker can more accurately localize the visitors' paths, it cannot correctly discover the visitors' activities that occur inside individual attractions.

To conclude, we summarize the contributions of this paper as follows. First, to the best of our knowledge, our work is the first approach for creating spatio-temporal story graphs from large sets of online photo streams, especially in the domain of theme parks where the discovery of underlying stories is of a particular importance for user modeling, personalization and recommendation. Through quantitative evaluation and crowdsourcing-based user studies with Amazon Mechanical Turk, we show that the story graphs serve as a convenient and mid-level data structure to which many applications are supported by simple algorithms applied, including photo-based exploration, recommendation, and temporal analysis. Second, we develop a scalable algorithm for creating semantic story graphs, consisting of two components: story element estimation and graph inference. The story element estimation includes the detection of faces and supporting objects, and attraction-based localization. The graph inference algorithm builds a time-varying sparse story graph directly from estimated story elements of photo streams, while addressing several key challenges of large-scale nature of our problems, including global optimality, easy parallelization, and linear computation time.

2. RELATED WORK

We discuss some representative previous studies from four lines of research that are closely related to our work.

Story extraction from Web data. In recent web mining research, much work has been done to build and visualize storylines from online text corpora such as news articles and scientific papers [1, 5, 22, 26]. The work of [5, 22] addresses the problem of extracting diverse story threads from large document collections. In [1], a probabilistic model is presented to group incoming news articles into temporary but tightly-focused storylines, to identify prevalent topics/keywords and evolving structure of stories. In [26], the NIFTY is developed for real-time tracking of *memes* (*e.g.* short textual phrases that travel and mutate on the Web) from blog posts and news articles. From this line of research, our work differs that we leverage image collections instead of text data, to discover and explore the storylines of user data.

Story extraction from visual data. The story has been used as an important concept to build interactive video

[1]For example, *Disneyland* is reported as the most geo-tagged location in the world on Instagram in 2014, as shown in http://time.com/3618002/most-instagram-places-2014/.

summarization or editing [7, 29], in which the objective is to summarize input video clips in a concise form (*e.g.* static images, shorter videos, or language sentences) while preserving key moments of the videos and allowing users to intuitively explore or edit them. In this category of work, input videos usually contain a small number of specified actors in fixed scenes. In contrast, our main technical challenge is the extraction of shared moments from photo streams that are independently taken by multiple visitors at different time.

The story extraction has been also investigated in the image domain as well. In [19], storyline-based summarization is discussed for a small private photo album. However, it is tested with only a small set of about 200 images from a single user's photo collection. This work is also related to our previous work [12] that leverages a large set of Flickr images to create photo storylines. However, [12] exploits only temporal information between images, whereas we additionally consider spatial information that synergetically interplays with time. Thus, our approach also involves an image localization task. Second, the algorithm of [12] defines the nodes of graphs using low-level image features only, whereas we here leverage high-level semantic story elements, including the detection output of faces and supporting objects.

Data mining for tourism. As massive amounts of travel data are available and the location-based systems proliferate, the data mining techniques begin offering significant benefits on tourism analytics in various ways. Some notable examples include customized tour recommendations in urban areas [6], discovery of international human mobility [25], recommendation of travel packages with consideration of both tourists' interests and travel costs [4], to name a few. Geo-referenced Flickr pictures are also leveraged for this purpose. In [14], representative and diverse travel routes are discovered from the histories of tourists' geotagged photos for landmarks. In [31], a diversity ranking method for such trajectory patterns is proposed. Our main novelty is that we take advantage of computer vision techniques to understand image contents, and thus we can possibly use all the images no matter whether GPS information is available or not. In addition, our idea of story-based summarization and exploration is unique in this line of work.

Exploration of large collections of tourists' images. With recent popularity of online image sharing, there has been a significant amount of effort for intuitively exploring large collections of unstructured tourists' photos. The work of [10] organizes the Flickr images of landmarks by analyzing the associated location information, tags, and contents. One of early pioneering work is *Photo Tourism* [24], which calibrates geo-tagged photos of tourist landmarks in a 3-D space, and enables users to interactively browse them. This work has been extended later in many different directions. In [23], tourists' paths around the landmarks are discovered for better navigation of the 3-D scene. In [21], a semantic navigation is established between regions of 3D landmark models and their corresponding reference texts of Wikipedia. In [15], online geo-tagged images are leveraged to create tours of the world's tourist sites on Google Maps. Compared to this line of research, our work differs in that we aim to build storylines, which illustrate a series of spatio-temporal events or episodes without requiring availability of accurate geometric information as input. Therefore, our system can bypass the time-consuming step of reconstructing 3-D models.

	CA	FL	PA	TK	HK	HI	N/A
# PS	6,402	11,437	1,707	1,849	856	120	5,409
# IM	568.8	1,037.2	197.8	196.3	80.1	19.7	413.6

Table 1: The number of photo streams and images ($\times 10^3$) of the Disneyland dataset according to the park locations. Total numbers of images and photo streams are (2,513,367, 27,780), respectively. (CA: California, FL: Florida, PA: Paris, TK: Tokyo, HK: Hong Kong, HI: Hawaii, N/A: unknown or noisy).

	Districts	Attractions	Dining	Entertainment
(DCA)	8	37	29 (1)	4
(DP)	8	63	22 (1)	8

Table 2: The attraction statistics of *Disney California Adventure* (DCA) and *Disneyland Park* (DP). The numbers in parentheses are about *character dining*.

3. PROBLEM FORMULATION

3.1 The Input Data

The input of our approach is two-fold: a set of visitors' photo streams and side information of the parks.

Photo streams. We download 3,602,727 unique images from Flickr by querying multiple *Disneyland* related keywords. Then, using the timestamp and user information associated with each image, we obtain 27,780 photo streams that contain more than 30 images. The input set of photo streams is denoted by $\mathcal{P} = \{P^1, \cdots, P^N\}$, where N is the number of photo streams. We sort the pictures of each photo stream by timestamps. Next we classify the photo streams according to the park locations where they are recorded: {*California*, *Orlando*, *Tokyo*, *Paris*, *Hong Kong*, *Hawaii*, *N/A*}. The *N/A* label is tagged for the photo streams that are not taken in any Disney parks or whose locations are unknown. We use GPS information when available; in our sets, 5,257 (19%) of photo streams include GPS information. Next we exploit the location keywords (*e.g.* *California*, *Florida*) in the text data (*e.g.* tags and titles) associated with images. Table 1 summarizes the statistics of our Disneyland dataset.

We apply our algorithm to the set of each park separately. To make our discussion easier and more coherent, we henceforth focus on the two parks at *California*: *Disney California Adventure Park* and *Disneyland Park*.

Side information about attractions. Disneyland parks consist of multiple districts, each of which includes a sets of attractions, dining, and other entertainment (*e.g.* parades and stage shows). Table 2 summarizes the statistics obtained from Disney's official maps. For simplicity, we hereafter use the term *attractions* to indicate attractions, dining, and entertainment events without distinction. We denote the set of attractions by $\mathcal{L}$. For dining, we consider a *character dining* as a separate attraction, and all the others as a single *restaurant* attraction. The *character dining* (*e.g.* *Ariel's Grotto*) is a restaurant where visitors can take pictures with characters (*e.g.* *Ariel the little mermaid*). For each attraction, we download at maximum 1,000 top-ranked images from Google and Flickr by querying the attraction name. We use the attraction images as training data for various tasks of story element estimation, such as image localization and supporting object detection, which will be presented in Section 4. We also obtain the GPS

Figure 2: Examples of images and GPS coordinates for four attractions in the *Fantasyland* district of *Disneyland park*. The attractions are identified by Disney's official map, and images are retrieved by Google and Flickr image search. The GPS coordinates are obtained from Google Maps.

coordinates of attractions from Google Maps, except for entertainment events that do not happen in a specific location (*e.g. Mickey's Soundsational Parade*). Fig.2 shows sampled training images and GPS coordinates of four attractions in the *Fantasyland* district. Indeed, each attraction builds on a unique theme, whose visual contents are clearly discriminative with other attractions. Thus, Google and Flickr images found by attraction names yield an excellent repository that captures various canonical views of the attractions.

3.2 The Storylines

The output of our algorithm is two-fold. First, we extract *story elements* from all images of the photo streams using computer vision techniques. Along with two low-level image features denoted by $\mathbf{v}$ (section 4.1), we define four types of *story elements* as story-related, high-level descriptors of images in the context of theme parks: {faces, time, location, supporting objects}, which are represented by four vectors: $\{\mathbf{f}, \mathbf{t}, \mathbf{c}, \mathbf{o}\}$. We will explain each of them in section 4.2–4.4.

Based on the estimated story elements over photo streams, the second output is the story graph $\mathcal{G} = (\mathcal{O}, \mathcal{E})$. The vertices $\mathcal{O}$ correspond to dominant image clusters of the photo streams, and the edge set $\mathcal{E}$ links the vertices that sequentially recur in many photo streams. More rigorous definition will be shown in Section 5.

4. STORY ELEMENT ESTIMATION

We discuss our low-level image description and story element estimation for the input photo streams $\mathcal{P}$.

4.1 Image Features

For low-level image description, we use dense feature extraction with vector quantization, which is one of standard methods in recent computer vision research. We densely extract HSV color SIFT [17] and histogram of oriented edge (HOG) features [3] on a regular grid of each image at steps of 4 and 8 pixels, respectively. We form 300 visual words for each feature type by applying K-means to randomly selected descriptors. Finally, the nearest word is assigned to every node of the grid. As image or region descriptors, we build L_1 normalized spatial pyramid histograms to count the frequency of each visual word at three levels [16]. We define the image descriptor $\mathbf{v}$ by concatenating the two spatial pyramid histograms of color SIFT and HOG features.

4.2 Face and Time Descriptors

Since much of visitors' images contain faces as foregrounds, we define a high-level image descriptor that encodes presence/absence of faces and their spatial layout in the image. For example, we may cluster the images that have similar face sizes and positions (*e.g.* clustering family photos), or bypass the localization step for the images that are fully occupied by faces. For face detection, we use the Fraunhofer engine [20], which returns bounding boxes containing faces with confidence scores. Based on the detection, we compute the following three types of face features. (i) Histogram of face areas: we first make 10 bins in the range of $[0, \sqrt{h \times w}]$, where h and w are image height and width respectively, and count the frequencies of squared areas of detected faces for each bin. (ii) Histogram of face locations: we split the image into 9 evenly split tiles (*i.e.* 3 per dimension), and count the detected face centers in each spatial bin. (iii) Histogram of pairwise face distances: We compute pairwise distances between centroids of all detected faces and make a histogram of 10 bins in the range of [0, diagonal length]. As a final face descriptor vector $\mathbf{f} \in \mathcal{R}^{29}$, we concatenate all the three histogram after L_1 normalization.

The timestamps of images can be trivially obtained from meta-data provided by Flickr. We use month and hour information to define time features because the events in theme parks are seasonally and daily periodic. We define the month feature $\mathbf{t}_m(i) \in \mathcal{R}^{12}$ of image I in which each bin i has a value of Gaussian weighting $\mathbf{t}_m(i) = g(i-m) \propto \exp(-(i-m)^2/\sigma_m)$, where m is the month of image I and $\sigma_m = 1.5$. Its basic idea is that if an image is taken in May, the feature values for nearby months like April and June (*i.e.* $\mathbf{t}_m(4)$ or $\mathbf{t}_m(6)$) are non-zeros as well. The same Gaussian weighting is used for the hour feature with $\sigma_h = 2$. Finally, the two features are L_1-normalized and concatenated into the time descriptor $\mathbf{t}$.

4.3 Localization of Photo Streams

It is important to find out which attraction each image is likely to be taken, because visitors' activities and stories can be modularized according to attractions. Thus, we perform the attraction-based localization, whose objective is to determine the likelihood of each image over attractions. Mathematically, we assign a probability vector $\mathbf{c}_i \in \mathbb{R}^L$ to each image of photo stream $P^n = \{I_1, \ldots, I_{N^n}\}$, where N^n is the number of images in P^n and L is the number of attractions (*i.e.* $L = |\mathcal{L}|$). For notational simplicity, we let $\mathbf{c} = [\mathbf{c}_1, \ldots, \mathbf{c}_{N^n}] \in \mathbb{R}^{L \times N^n}$.

We run localization of each photo stream separately because all photo streams are taken by different users independently of one another. We use a Conditional Random Field (CRF) model to infer the conditional distribution over the attraction labels of each photo stream P^n. The strength of CRF is the flexibility to easily incorporate various pieces of evidence related to localization as energy terms in a single

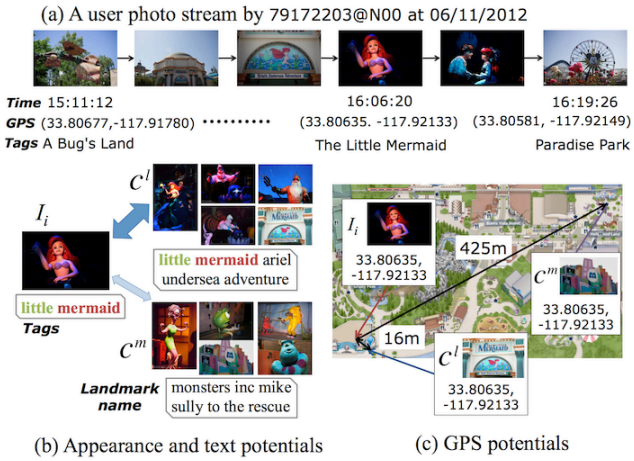

Figure 3: An example of attraction-based localization. (a) An input photo stream with meta-data of timestamps, GPS coordinates, and text tags. (b) The appearance potential of each image is computed by visual similarity with attraction images. The text potential is defined by the tf-idf measure between tags of the image and attraction names. (c) GPS potential is obtained from a normal distribution on the distances between the image and attractions.

unified model. Since each photo stream is a time-ordered sequence of images, it can be modeled by a linear-chain CRF. The conditional probability of the attraction labels $\mathbf{c}$ given a photo stream P^n is defined by

$$\log P(\mathbf{c}|P^n) = \sum_{i=1} \Big(\psi(\mathbf{c}_i, I_i) + \pi(\mathbf{c}_i, I_i) + \lambda(\mathbf{c}_i, I_i)\Big) \quad (1)$$
$$+ \phi(\mathbf{c}_0) \sum_i \phi(\mathbf{c}_i, \mathbf{c}_{i+1}, \Delta_i) - \log Z(P^n)$$

where $Z(P^n)$ is the partition function. In the following, we discuss each term of Eq.(1) in detail.

Appearance potential. The $\psi(c_i^l, I_i)$ represents the likelihood that image I_i is *visually* associated with attraction l. (See Fig.3(b)). Intuitively, $\psi(c_i^l, I_i)$ has a high value if I_i is visually similar to the images of attraction l. Since the training images of attractions are available, the estimation of $\psi(c_i^l, I_i)$ can be accomplished by any image classifier or their combination. In this paper, the potential is computed by summing the normalized scores of two classifiers, a KNN classifier and a linear SVM [28]. For SVM classifiers, we learn multiple classifiers per attraction because the training images of each attraction may contain multiple views; we first split the attraction image set into k different groups using K-means clustering with $k = \min(2.5\sqrt{n}, 200)$, and learn a separate classifier per cluster.

GPS potential. For the 19% of photo streams where GPS information is available, we define the GPS potentials as a normal distribution: $\pi(c_i^l, I_i) = N(x_c^l; x_i, \sigma)$ where x_c^l and x_i are the GPS coordinates of attraction l and image I_i, respectively. (See Fig.3(c)). We set the standard deviation $\sigma = 0.07$ km. That is, if a attraction l is $3\sigma(= 0.21$ km) distant away from I_i, l is very unlikely to be assigned to I_i. This GPS term is ignored for the photo streams without GPS information.

Text potential. We also take advantage of text information when available, as shown in Fig.3(b). We use the tf-idf measure (Term Frequency Inverse Document Frequency).

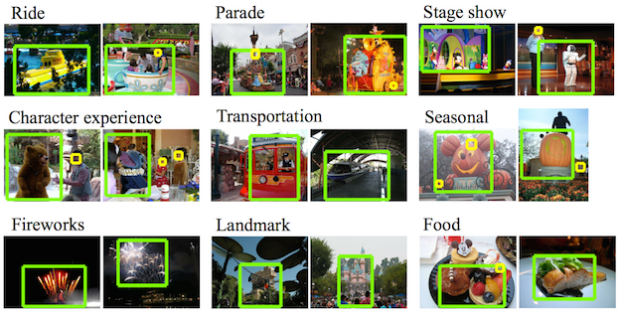

Figure 4: Examples of detected regions of interest encoding supporting objects of different meanings.

Let us denote the text associated with image I_i by D_i, which is a list of words of the title and tags. For each term $q \in D_i$, we compute the tf measure $tf_{q,i}$, which is the number of occurrences of q in D_i, and the idf measure $idf_q = \log(N/df_q)$ where df_q is the number of images containing q, and N is the total number of images. Then, for each attraction l, we generate a word list using its name, denoted by $\mathbf{q}^l$. The potential $\lambda(c_i^l, I_i)$ of image I_i to attraction l is computed by

$$\lambda(c_i^l, I_i) = \sum_{q \in \mathbf{q}^l} tf_{q,i} \times idf_q. \quad (2)$$

Edge potential. The $\phi(c_i^l, c_i^m, \Delta_i)$ is defined by the likelihood that a pair of consecutive images I_i and I_{i+1} are associated with attraction l and m respectively, given Δ_i that denotes the elapsed time between I_i and I_{i+1}. The edge potential depends on both spatial connectivity between attractions and visitors' walking speed in the park. We define the edge potential as follows. We first build a connectivity graph between attractions $\mathcal{G}_c = (\mathcal{L}, \mathcal{E}_c)$ based on the official park map and the GPS coordinates. Each attraction in $\mathcal{L}$ is linked to its k closest attractions, and the edge weight is given by the walking time between a attraction pair, which is computed by dividing the distance by the human walking speed s_w. we set $s_w = 1$km/h. Inspired by a human mobility study [8], we assume that a visitor stays in the same location with a probability α or moves to one of neighbor attractions with $1 - \alpha$. α is sampled from a truncated exponential distribution: $\alpha \propto \exp(-\lambda\Delta_i)$. With probability $1 - \alpha$ of moving to another attraction, we use the Gaussian model for a transition likelihood from attraction l to m:

$$\phi(c_i^l, c_i^m, \Delta_i) = \frac{1}{\sigma\sqrt{2\pi}} \exp(-\Delta_i^2/2\sigma^2) \text{ where } \sigma = s_w\Delta_i. \quad (3)$$

where σ is proportional to Δ_i; with a longer Δ_i, farther attractions can be reached.

Inference. Since our model for the localization is a linear-chain CRF, exact inference is possible. We use the Viterbi algorithm to obtain the most probable attraction assignment of the photo stream $\mathbf{c}^* = [c_1^*, \ldots, c_{n^l}^*]$. We also compute the posterior marginals of location labels $\mathbf{c}$ using the forwards-backwards algorithm.

4.4 Detection of Supporting Objects

We detect the regions of interest (ROI) that may include important supporting objects of images for the following two reasons. First, it can be used as another high-level image descriptor, as shown in Fig.4, because supporting objects illustrate the main theme of the images, such as rides, meals, or interaction with characters. Second, it can be exploited

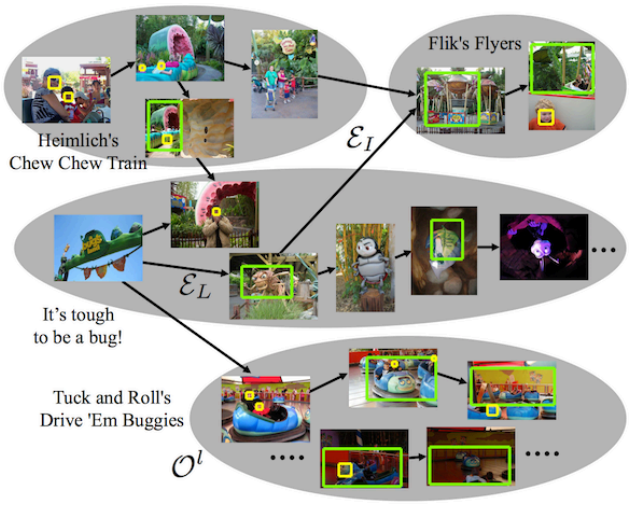

Figure 5: A small piece of story graph between and inside four attractions of *Bug's Land* district of *Disney California Adventure* Park. Each image represents the central image of each vertex.

for user interaction; we allow users to choose ROIs to retrieve the images that share the similar supporting objects, as another control for story exploration. We will introduce examples in Section 7.

For ROI detection, we first learn a set of ROI classifiers of each attraction as follows. We first sample 10 rectangular windows per training image of each attraction using the objectness metric [2], and build a similarity graph between the windows of all training images, using histogram intersection on the color SIFT and HOG features $\mathbf{v}$ of the windows. By applying the diversity ranking and clustering algorithm of [13] to the similarity graph, we discover k representative and discriminative exemplar windows, and cluster windows by associating each window with its closest exemplar. We set $k = \min(2.5\sqrt{n}, 200)$, where n is the size of graph. For each exemplar, we learn KNN and linear SVM classifiers as done for the appearance potential of localization in section 4.3. For a positive training set, we use the exemplar and its 10 nearest neighbors. For a negative training set, we randomly sample 200 windows from the other attractions. As a result, we have k number of classifiers per attraction. We repat this ROI classifier learning for all attractions.

Next we leverage the learned classifiers to detect the ROIs for the images of photo streams. As a pre-processing step, we first sample 100 windows per image using the objectness metric. Our assumption here is that at least some of candidate windows correspond to good potential supporting objects. We then remove the windows if they largely overlap with face detection (*i.e.* if more than 50% of the window area overlaps with any face region), due to redundancy with face detection. Then, if the image is localized as attraction l, we apply a set of learned classifiers for attraction l, and choose one single best window per image as the ROI.

5. STORY GRAPH RECONSTRUCTION

In this section, we discuss how to create a story graph $\mathcal{G} = (\mathcal{O}, \mathcal{E})$ from the results of story element estimation. We build story graphs hierarchically. As shown in Fig.5, the upper-level story graph is defined between attractions, each of which subsumes small story subgraphs that represent sequences of activities inside attraction. That is, in the storyline graph $\mathcal{G} = (\mathcal{O}, \mathcal{E})$, the vertex set can be decomposed into the ones per attraction (*i.e.* $\mathcal{O} = \bigcup_{l \in \mathcal{L}} \mathcal{O}^l$), and the edge set $\mathcal{E}$ consists of two groups: $\mathcal{E} = \mathcal{E}_L \cup \mathcal{E}_I$ where $\mathcal{E}_L$ defines the edges inside each attraction, and $\mathcal{E}_I$ includes edges between attractions.

5.1 Vertices of Story Graphs

Since many input images are redundant, it is inefficient to build a story graph over individual images. Hence, the vertices $\mathcal{O}$ are preferentially defined as *image clusters*. The vertex set $\mathcal{O}^l$ for attraction l is obtained as follows. We first represent each image by concatenating the set of vectors $\{\mathbf{v}, \mathbf{f}, \mathbf{o}, \mathbf{t}\}$, each of which represents the low-level image feature, the face descriptor, the ROI descriptor, and the time descriptor, respectively. We include the ROI descriptor $\mathbf{o}$ to encourage clustering the images that share similar supporting objects. In order to define the vertex set $\mathcal{O}^l$ for each attraction, we first build a similarity matrix between the images localized at attraction l, by applying the histogram intersection to the above descriptors. We then discover k exemplars and clusters using the same diversity ranking and clustering algorithm of [13] with $k = \min(2\sqrt{n}, 200)$. Then, the k clusters constitues the vertex set $\mathcal{O}^l$ for attraction k. We repeat this process for each attraction $l \in \mathcal{L}$. We denote the number of vertices by M (*i.e.* $M = |\mathcal{O}| = \sum_{l \in \mathcal{L}} |\mathcal{O}^l|$).

As a result of clustering for the vertex definition, each image is now associated with a vertex membership vector $\mathbf{x} \in \mathbb{R}^M$, which has only a single nonzero element $x_v = 1$ if the image is a member of vertex $v \in \mathcal{O}$. Therefore, each photo stream $P^n = \{I_1, \ldots, I_{N^n}\}$ can be represented by the membership vectors $P^n = \{\mathbf{x}_1, \ldots, \mathbf{x}_{N^n}\}$.

5.2 Edges of Story Graphs

The edge set $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$, whose adjacency matrix is denoted by $\mathbf{A} \in \mathbb{R}^{M \times M}$, includes directed edges between the vertices that sequentially co-occur across many photo streams. In order for the story graphs to be practical, we enforce the edge set $\mathcal{E}$ to satisfy the following two properties. (i) $\mathcal{E}$ should be *sparse*. If there are too many edges in the graph, the narrative structure is unnecessarily complex. Thus, we retain only a small number of strong story branches per node. (*i.e.* $\mathbf{A}$ has only a few nonzero elements). (ii) $\mathcal{E}$ should be *time-varying*; $\mathcal{E}$ smoothly changes over time in a day $t \in [0, T]$. It means that the preferred transitions between attractions can change over time in the theme parks. For example, in many visitors' photo streams, the images of attraction *Mad Tea Party* are likely to be followed by *dining* images around lunch time, by *fireworks* images at night, or by nearby attraction images like *Pixie Hollow* in other time. Hence, we infer individual $\mathbf{A}^t$ every 30 minutes while changing t from 8AM to 12AM to densely capture such time-varying popular story transitions.

In our previous work [12], we formulate a maximum likelihood estimation for inferring $\{\mathbf{A}^t\}$ given input photo streams $\mathcal{P} = \{P^1, \cdots, P^N\}$, and develop an optimization algorithm that has several appealing properties for large-scale problems, including global optimality, easy parallelization, and linear computation time. We here skip the details of optimization, which can be found in [12]. Instead we denote edge reconstruction procedure by $\{\mathbf{A}^t\}_{t=1}^T = \mathsf{GraphInf}(\mathcal{P}, T)$.

Although we use the similar algorithm for creating edges of graphs with our previous work [12], the story graphs of

Method	Top-1 Attr.	Top-5 Attr.	Top-1 Dist.
(A+T+E+G)	**8.34**%	**20.56**%	**28.03**%
(A+T+E)	5.18%	16.16%	22.87%
(A+T)	4.88%	14.52%	20.12%
(A)	5.02%	15.88%	21.31%
(T)	3.12%	9.17%	17.83%
(VKNN)	5.16%	16.85%	22.12%
(VSVM)	4.63%	15.80%	20.63%
(Rand)	0.93%	4.63%	5.56%

Table 3: Results of image localization. We report top-1/5 attraction accuracies, and top-1 district accuracies. We test our CRF-based approach with different combinations of terms: (T), (A), (G), and (E) indicate appearance, GPS, text, and edge potential.

this work are different from those of [12] in two respects. First, the graphs of [12] are built based on only temporal information between images, whereas here we consider both time and spatial information. Second, the vertices of the graphs of [12] are based on low-level image features only (*e.g.* SIFT and HOG features), whereas we define vertices over *story elements* of high-level semantic image description, including face and object detection. In section 6.2, we empirically compare these two story graphs for the image recommendation tasks.

6. EXPERIMENTS

We evaluate the accuracy of our story element estimation for image localization (Section 6.1). Then, we quantitatively compare the performance of image recommendataion using our story graphs with other candidate methods (Section 6.2).

6.1 Results of Image Localization

We evaluate the performance of our approach for estimating the *locations* of images, which may be considered as the most important story elements. The task here is to find out the attraction that a given image is likely to be taken. We select 108 attractions and restaurants as the classes for localization. We obtain groundtruth by letting human experts to annotate 3,000 images of photo streams. We randomly sample 2,000 images out of them and perform a localization experiment, which is repeated ten times.

For baselines, we implement two vision-based methods, which are solely based on the visual contents of images. We use the training images of each landmark to learn linear-SVM and KNN classifiers, which are denoted by (VSVM) and (VKNN), respectively. In addition, in order to quantize the contribution of terms of our CRF-based localization framework in Eq.(1), we also measure the variation of localization accuracies by changing the combination of the four terms; we use (T), (A), (G), and (E) to indicate appearance, GPS, text, and edge potential, respectively.

Table 3 summarizes the results of our experiments. We also report chance performance (Rand) to show the difficulty of the localization task. Our CRF-based approach outperforms the vision-based methods; the accuracy of our full model (A+T+E+G) is higher than the best vision-based method (VKNN) by 61.6% and 26.7% in terms of the top-1 attraction and those district metric, respectively. We also make three observations from the results. First, the visual content of images is a strong clue for localization. Second, text associated with Flickr images is noisy, and hence provides little to enhance localization accuracy. Third, the edge potential improves the localization accuracy by enforcing temporal constraints between consecutive images.

We note that the task is very difficult, with chance performance of under 1%. In many cases, even human experts feel difficulty in localizing images as content may match multiple locations (*e.g.* Mickey can be observed virtually at any location). We believe overall accuracy can be improved significantly by pre-filtering the images, however, in this experiment, our focus is on assessing the contributions of individual terms toward our CRF localization objective, not absolute performance. Hence, the tendency is important instead of absolute numbers in this experiment.

6.2 Results on Image Prediction

We evaluate the ability of our story graphs to perform photo recommendation, which is one of key practical applications of story graphs. We carry out the following two image sequence prediction tasks: (I) predicting next likely images given a short image sequence, and (II) filling in a missing part of a novel photo stream. The first task simulates the scenario where a visitor takes a small number of pictures during his trip, and the task suggests up the most likely next pictures of attractions by analyzing the photo sets of other users who had similar experience. The second task can show the pictures of alternative paths taken by other visitors. It helps the visitor compare similarity and differences of her trip with others. Fig.6 shows examples of the two prediction tasks.

We use the similar experimental protocol of [12]. We first randomly select 80% of photo streams as a training set and the others as a test set. We reduce each test photo stream into uniformly sampled 100 images, since consecutive images can be often very similar. For task (I), we randomly divide a test photo stream into two disjoint parts. Then, the goal is, given the last 10 images of the first part and next 10 query time points $\mathbf{t}_q = \{t_{q1}, \ldots, t_{q10}\}$, to retrieve the 10 images that are likely to appear at $\mathbf{t}_q$ from the training set. The actual images at $\mathbf{t}_q$ are used as groundtruth. For task (II), we randomly crop out 10 images in the middle of each test photo stream. Then, the goal is to predict the likely images for the missing part given the time points $\mathbf{t}_q$ and five images before and after the missing slots. Since training and test sets are disjoint, each algorithm can only retrieve similar (but not identical) images from training data at best.

As a result of inference of story graphs, we obtain a set of $\{\mathbf{A}^t\}$, which can be regarded as a state transition matrix between vertices of the graph at different time t. We adopt the *state space model* (SSM) to perform all prediction tasks [18]. For example, we can predict next k likely story vertices using the forward algorithm, and infer the best k paths between any two vertices using the top-k Viterbi algorithm.

We compare our approach with the method of [12] and the four baselines used in [12]. The first baseline (Page) is a Page-Rank based image retrieval, which is one of most successful methods to retrieve a small number of canonical images. The second baseline (HMM) is based on the HMM, which has been popularly used for modeling tourists' sequential photo sets. The third (Clust) is a clustering-based summarization on the timeline [11], in which images on the timeline are grouped into 10 clusters using K-means at every 30 minutes. The forth baseline is the method of [12], denoted by (Temp), whose key differences from our approach are (i) use of low-level features only (vs. story elements in our

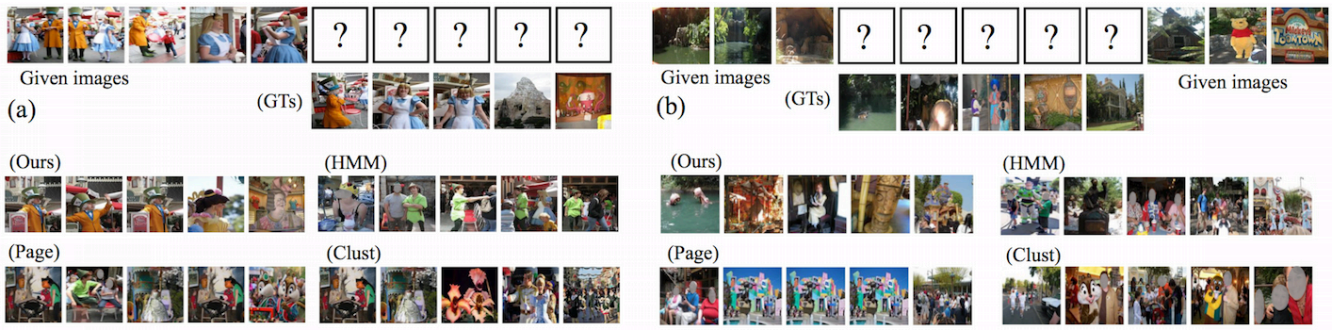

Figure 6: Examples of the two prediction tasks: task (I) (*i.e. predicting likely next images*) in (a), and task (II) (*i.e. filling in missing parts*) in (b). The first row shows a question (*i.e. given* images and five slots to be predicted). In each set, we show hidden groundtruth and predicted images by different algorithms. In AMT preference tests, we show the first row and a pair of predictions by our algorithm and one baseline in a random order. A turker is asked to select the more likely one to occur in the empty slots.

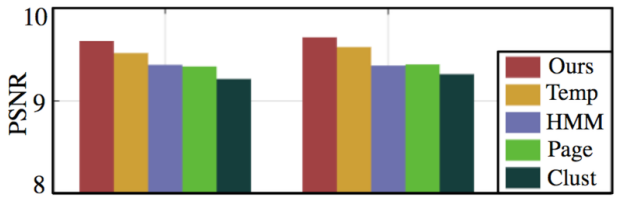

Figure 7: Results of our method and four baselines for the two prediction tasks. The PNSR values are as follows: [(Ours) , (Temp), (HMM), (Page), (Clust)] = [9.65, 9.51, 9.39, 9.37, 9.24] for task (I), and [9.69, 9.58, 9.38, 9.39, 9.29] for task (II).

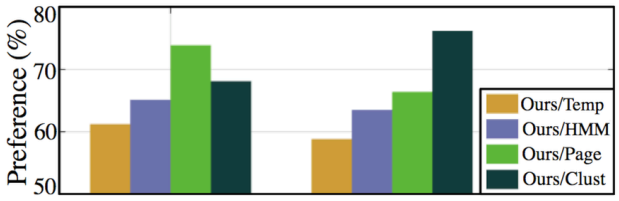

Figure 8: Results of pairwise preference tests via AMT between our method and baselines. The number should be higher than 50% to validate the superiority of our method. The average preferences of our method over (Temp), (HMM), (Page), and (Clust) are [61.2, 65.1, 73.9, 68.1] for task (I), and [58.8, 63.5, 66.4, 76.2] for task (II).

approach) for the node definition, and (ii) temporal information only (vs. spatio-temporal information) for the edge definition. We measure the performance in two different ways: quantitative similarity measures and crowdsourcing-based user studies via Amazon Mechanical Turk (AMT).

Quantitative results. We evaluate the prediction accuracy by measuring the similarity between predicted images and groundtruth images of test photo streams. For a similarity metric, we resize the images into 32×32 tiny images [27] to focus on holistic views instead of minor details, and compute peak signal-to-noise ratio (PSNR) between them. A higher value indicates that the two images are more similar.

Fig.7 shows comparison results between our method and four baselines for task (I) and (II). Our algorithm significantly outperforms all the competitors. That is, predicted images by our method are more similar to groundtruths than those by other baselines. For example, our PSNR performance gains (in dB) over the best baseline (Temp) are 0.131 and 0.106 for the two prediction tasks, respectively. All the numbers can be found in the caption of Fig.7.

User studies via AMT. Actual users' satisfaction is the most important measure of recommendation. Thus, we here evaluate the performance of image recommendation using user studies via Amazon Mechanical Turk (AMT). We use the same experimental setup with the previous tests, except that the number of images to be estimated is reduced to five for easy evaluation by human subjects. We show *given* images and five empty slots as a question (*e.g.* the first row of Fig.6), and then show a pair of image sequences predicted by our algorithm and one of baselines in a random order, while algorithm names are hidden. A turker is asked to choose one of them that is more likely to come at the empty slots. We obtain such pairwise comparison for each test set from at least five different turkers. In the second row of Fig.6, we show the groundtruth (*i.e.* hidden actual images), and predicted images by our algorithm and three baselines.

Fig.8 shows the quantitative results of pairwise AMT preference tests between our method and four baselines. The number indicates the mean percentage of responses that choose our prediction as a more likely one than that of each baseline. Hence, numbers should be higher than 50% to validate the superiority of our algorithm. Our algorithm significantly outperforms all the baselines; for example, our algorithm (Ours) gains 65.1% and 63.5% of votes over the baseline (HMM) in the two tasks.

7. APPLICATIONS

Once we build a story graph as a data-driven summary of visitors' activities in the parks, we can leverage it toward several interesting applications. We present preliminary storybook-like demos regarding exploration, recommendation, and temporal analysis, all of which may be beneficial uses of the story graphs to visitors.

Exploration. The first intuitive application is spatio-temporal exploration of the parks. As shown in Fig.9, we can hierarchically explore the stories between or inside attractions, with a single click of controls. In Fig.9.(a), the map on the left shows the current attraction and next available ones that are popularly visited after the current attraction. A user can proceed to a nearby attraction by clicking

Figure 9: An example of story exploration. (a) The map on the left shows the current attraction and nearby available attractions that are popularly visited from the current location. The right panel illustrates the first images of story subgraphs for the current attraction. The detailed view of each story subgraph is presented in (c). (b) We allows a user to click a region of interest to retrieve its closest neighbors using a simple KNN search. We also show the next most popular transitions from the retrieved images.

one of the blue controls. On the right panel, we show a part of story subgraphs at the current attraction (*i.e.* the *Main Street USA*). Here each photo shows the first image of each story subgraph, whose detailed view is presented in Fig.9.(c). Every story subgraph illustrates a popular coherent story thread, for example, from top to bottom, Fig.9.(c) shows *Walt Disney Statue*, *Sleeping Beauty Castle*, transportation at the main street, character experiences, and parades. Fig.9.(b) shows an example of image retrieval using regions of interest (ROI). By clicking an ROI, we retrieve the closest neighbors using a simple KNN search. For retrieved images, we also show next most popular transitions.

Recommendation. Fig.10 shows an example of predicting the location of a novel image and suggesting next engaging paths. Since we organize images according to attractions in our story graph, we can easily localize the new images using a simple KNN search, as shown in Fig.10.(a). Based on the retrieved images, we suggest several attractions from which the exploration begins. The query image includes *Mickey's Fun Wheel*, which is viewable from most attractions in the *Paradise Pier* district. As done in the exploration example of Fig.9, a visitor is allowed to choose one of nearest attractions for further exploration, or have our system suggest a popular one for them. If a user chooses *California Screamin'*, as shown in Fig.10.(b), we suggest two popular next available attractions, which are *Ariel's Grotto* and *King Triton's Carousel*. On the bottom, we preview the initial segments of story subgraphs of the two attractions.

Temporal analysis. Fig.11 shows an example of temporal analysis that benefits from our story graphs. Every two hours on the timeline, we present exemplar images of two central vertices of the story graph at the *Mickey's Fun wheel*. The ranking scores of vertices at time t can be easily obtained by column-wise summing $\mathbf{A}^t$. As shown in Fig.11, even at a single location, a variety of events happen at different time throughout the day, for example, *Goofy Instant Concert*, *Pixar Play Parade*, *Electrical Parade*, and *World of Color* from 2PM to 10PM. Our story graph can help visitors plan their itinerary beforehand by providing an intuitive way to promptly preview the popular events.

8. CONCLUSION AND DISCUSSION

We presented an approach for creating story graphs from large collections of online photo streams shared by visitors to theme parks. We formulate the story reconstruction as a two-step procedure of story element estimation and inference of sparse time-varying directed graphs. To demonstrate the usefulness of the story graphs, we leverage them to perform photo-based exploration, recommendation, and temporal analysis tasks. Through quantitative evaluation and user studies via AMT, we show that our algorithm outperforms other alternatives for two image prediction tasks.

There are several promising future directions that go beyond the current paper. First, to be more practical for general users, we can leverage guidebooks to infuse semantic meaning for exploration and recommendation. Second, personalization of story graphs is another interesting direction, in order to deliver relevant guidelines to visitors based on the group sizes, people's age and gender, and visiting seasons. Finally, it would be also interesting to implement our technique as a function of Disneyland official mobile apps (*e.g. Disneyland Explorer* and *My Disney Experience*).

9. REFERENCES

[1] A. Ahmed, Q. Ho, J. Eisenstein, E. P. Xing, A. J. Smola, and C. H. Teo. Unified Analysis of Streaming News. In *WWW*, 2011.

[2] B. Alexe, T. Deselaers, and V. Ferrari. What is an Object? In *CVPR*, 2010.

[3] N. Dalal and B. Triggs. Histograms of Oriented Gradients for Human Detection. In *CVPR*, 2005.

[4] Y. Ge, Q. Liu, H. Xiong, A. Tuzhilin, and J. Chen. Cost-aware Travel Tour Recommendation. In *KDD*, 2011.

[5] J. Gillenwater, A. Kulesza, and B. Taskar. Discovering Diverse and Salient Threads in Document Collections. In *EMNLP*, 2012.

[6] A. Gionis, T. Lappas, K. Pelechrinis, and E. Terzi. Customized Tour Recommendations in Urban Areas. In *WSDM*, 2014.

[7] D. B. Goldman, B. Curless, D. Salesin, and S. M. Seitz. Schematic Storyboarding for Video Visualization and Editing. In *SIGGRAPH*, 2006.

Figure 10: An example of predicting the location of a query image and suggesting next engaging story paths. (a) For a query image, we show top 3 most likely attractions. (b) A user selects *California Screamin'* as a next destination from which the story exploration begins. On the bottom, we preview the initial story subgraphs of the two attractions, *Ariel's Grotto* and *King Triton's Carousel*.

Figure 11: Variation of high-ranked vertices in the story graph at *Mickey's Fun Wheel* sampled every two hours. Each image represents the central image of each vertex.

[8] M. C. Gonzalez, C. A. Hidalgo, and A.-L. Barabasi. Understanding Individual Human Mobility Patterns. *Nature*, 453:779–782, 2006.

[9] J. A. Hudson and L. R. Shapiro. From Knowing to Telling: The Development of Children's Scripts, Stories, and Personal Narratives. In A. McCabe and C. Peterson, editors, *Developing Narrative Structure*. LEA, 1991.

[10] 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 *ACM MM*, 2007.

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

[12] G. Kim and E. P. Xing. Reconstructing Storyline Graphs for Image Recommendation from Web Community Photos. In *CVPR*, 2014.

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

[14] T. Kurashima, T. Iwata, G. Irie, and K. Fujimura. Travel Route Recommendation Using Geotags in Photo Sharing Sites. In *CIKM*, 2010.

[15] A. Kushal, B. Self, Y. Furukawa, D. Gallup, C. Hernandez, B. Curless, and S. M. Seitz. Photo Tours. In *3DIMPVT*, 2012.

[16] S. Lazebnik, C. Schmid, and J. Ponce. Beyond Bags of Features: Spatial Pyramid Matching for Recognizing Natural Scene Categories. In *CVPR*, 2006.

[17] D. G. Lowe. Distinctive Image Features from Scale-Invariant Keypoints. *IJCV*, pages 91–110, 2004.

[18] K. P. Murphy. *Dynamic Bayesian Networks: Representation, Inference and Learning*. PhD thesis, University of California, Berkeley, 2002.

[19] P. Obrador, R. de Oliveira, and N. Oliver. Supporting Personal Photo Storytelling for Social Albums. In *ACM MM*, 2010.

[20] T. Ruf, A. Ernst, and C. Küblbeck. Face Detection with the Sophisticated High-speed Object Recognition Engine (SHORE). *Microelectronic Systems*, pages 237–246, 2011.

[21] B. C. Russell, R. Martin-Brualla, D. J. Butler, S. M. Seitz, and L. Zettlemoyer. 3D Wikipedia: Using Online Text to Automatically Label and Navigate Reconstructed Geometry. In *SIGGRAPH Asia*, 2013.

[22] D. Shahaf, C. Guestrin, and E. Horvitz. Metro Maps of Science. In *KDD*, 2012.

[23] N. Snavely, R. G. S. M. Seitz, and R. Szeliski. Finding Paths through the World's Photos. In *SIGGRAPH*, 2008.

[24] N. Snavely, S. M. Seitz, and R. Szeliski. Photo Tourism: Exploring Photo Collections in 3D. In *SIGGRAPH*, 2006.

[25] B. State, I. Weber, and E. Zagheni. Studying Inter-National Mobility through IP Geolocation. In *WSDM*, 2013.

[26] C. Suen, S. Huang, C. Eksombatchai, R. Sosic, and J. Leskovec. NIFTY: A System for Large Scale Information Flow Tracking and Clustering. In *WWW*, 2013.

[27] 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.

[28] A. Vedaldi, V. Gulshan, M. Varma, and A. Zisserman. Multiple Kernels for Object Detection. In *CVPR*, 2009.

[29] M. Wang, R. Hong, X.-T. Yuan, S. Yan, and T.-S. Chua. Movie2Comics: Towards a Lively Video Content Presentation. *IEEE T. Multimedia*, 14(3):858–870, 2012.

[30] J. Xiao and Y. Furukawa. Reconstructing the World's Museums. In *ECCV*, 2012.

[31] Z. Yin, L. Cao, J. Han, J. Luo, and T. Huang. Diversified Trajectory Pattern Ranking in Geo-Tagged Social Media. In *SDM*, 2011.