To appear, Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2011.

Clues from the Beaten Path: Location Estimation with Bursty Sequences of Tourist Photos

Chao-Yeh Chen and Kristen Grauman
University of Texas at Austin
chaoyeh@mail.utexas.edu, grauman@cs.utexas.edu

Abstract

Image-based location estimation methods typically recognize every photo independently, and their resulting reliance on strong visual feature matches makes them most suited for distinctive landmark scenes. We observe that when touring a city, people tend to follow common travel patterns—for example, a stroll down Wall Street might be followed by a ferry ride, then a visit to the Statue of Liberty. We propose an approach that learns these trends directly from online image data, and then leverages them within a Hidden Markov Model to robustly estimate locations for novel sequences of tourist photos. We further devise a set-to-set matching-based likelihood that treats each "burst" of photos from the same camera as a single observation, thereby better accommodating images that may not contain particularly distinctive scenes. Our experiments with two large datasets of major tourist cities clearly demonstrate the approach's advantages over methods that recognize each photo individually, as well as a simpler HMM baseline that lacks the proposed burst-based observation model.

1. Introduction

People often look at their pictures and think about where they were taken. Tourists frequently post their collections online to share stories of their travels with friends and family. Today, it is a cumbersome manual process to organize vacation photos and make them easily searchable: users must tag photos with relevant keywords and manually split batches into meaningful albums. When available, geo-reference data from GPS sensors can help automate some aspects of this organization; however, GPS falls short for images taken indoors, and a mere 3% of the billions of existing consumer photos online actually have a GPS record [1]. Even with precise positioning, world coordinates alone are insufficient to determine the site tags most meaningful to a person, which can vary substantially in spatial scope depending on the content (e.g., photos within the Roman ruins, vs. photos at the Mona Lisa).

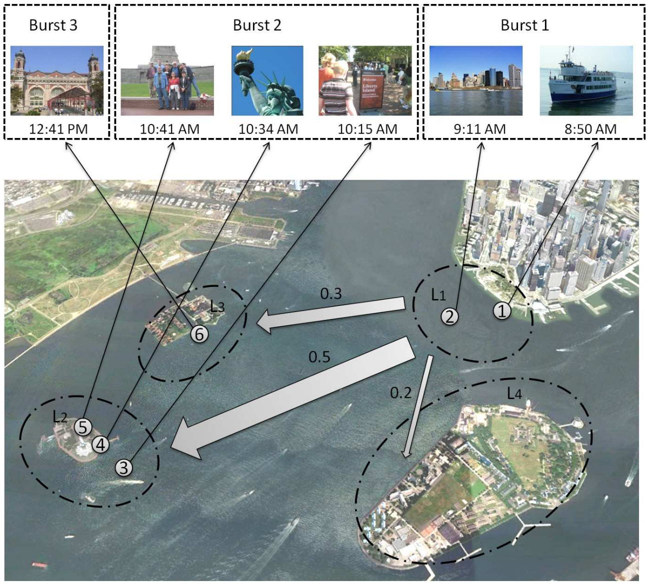

Figure 1. We propose to exploit the travel patterns among tourists within a city to improve location recognition for new sequences of photos. Our HMM-based model treats each temporal cluster ("burst") of photos from the same camera as a single observation, and computes a set-to-set matching likelihood function to determine visual agreement with each geospatial location. Both the learned transition probabilities between locations and this grouping into bursts yield more accurate location estimates, even when faced with non-distinct snapshots. For example, the model benefits from knowing that people travel from L_1 to L_2 more often than L_3 or L_4, and can accurately label all the photos within Burst 2 even though only one (the Statue of Liberty) may match well with some labeled instance.

Thus, there is a clear need for image-based location recognition algorithms that can automatically assign geographic and keyword meta-data to photos based on their visual content. The wide availability of online consumer photo collections in recent years has spurred much research in the field [14, 12, 21, 18, 15, 16, 4, 13, 6, 8, 7, 9]. In particular, approaches based on matching sparse local invariant features can reliably identify distinctive "landmark" buildings or monuments [14, 12, 21, 18, 15], while methods using image-level descriptors together with classifiers offer a coarser localization into probable geographic regions [6, 4, 9, 22, 7].

Despite impressive progress, there are several limitations to previous methods. First, techniques that rely on precise local feature matches and strong geometric verification are restricted in practice to recognizing distinctive landmarks/facades, which account for only a fraction of the photos tourists actually take. Second, methods that use global image feature matches combined with nearest neighbor matching make strong assumptions about the density (completeness) of the labeled database available, and are generally validated with error measures that may be too coarse for some users' goals (e.g., a single location label has a granularity of 200-400 km [6, 7]). Finally, almost all existing techniques attempt to recognize a single image at a time, disregarding the context in which it was taken.

We propose an approach for location estimation that addresses these limitations. Rather than consider each snapshot in isolation, we will estimate locations across the time-stamped *sequences* of photos within a user's collection. What does the sequence reveal that each photo alone does not? Potentially, two very useful pieces of information: 1) People generally take photos in "bursts" surrounding some site or event of interest occurring within a single location (e.g., one snaps a flurry of photos outside the Pantheon quickly followed by a flurry within it), which means we have a powerful label smoothness constraint from the timestamps themselves, and 2) Tourists often visit certain sites within a city in a similar order, which means we have useful transition priors between the locations in a sequence. The common transitions may stem not only from the proximity of prime attractions, but also external factors like walking tours recommended by guidebooks, or the routes and schedules of public transportation. For example, in New York, a stroll down Wall Street might be followed by a ferry ride, then a visit to the Statue of Liberty or Ellis Island museum (see Figure 1).

Thus, our key new idea is to learn the "beaten paths" that people tend to traverse, and then use those patterns when predicting the location labels for new sequences of photos. During training, we use geo-tagged data from the Web to discover the set of locations of interest within a city, as well as the statistics of transitions between any pair of locations. This data is used to construct a Hidden Markov Model (HMM). Given a novel test sequence, we first automatically segment it into "bursts" of photos based on temporal (timestamp) similarity. We then treat each burst as an observation, and define a likelihood function based on set-to-set matching between that burst's photos and those within the training set for each location. Compared to a naive single-image observation, this likelihood is more robust to test photos unlike any in the training set, and can latch on to any informative matches within a burst that suggest the true location (e.g., see Burst 2 in Fig. 1, in which the Statue of Liberty is distinctive, but the shots of people posing would not be). This is an important advantage of the system, since it means the GPS-labeled training data need not cover all aspects of the entire city to be viable.

While a few previous methods incorporate some form of temporal context, unlike our approach they are intended for video inputs and learn transitions between well-defined areas (rooms) within a building [19, 20], are interested in coarse world region labels (each location = 400 km) [7], or use a short fixed-size temporal window as context and do not model the site-to-site travel trends of tourists [9].

We validate our approach with two large datasets of major tourist cities downloaded from Flickr. The results clearly demonstrate its advantages over traditional methods that recognize each photo individually, as well as a naive HMM baseline that lacks the proposed burst-based observation model. The system's performance suggests exciting possibilities not only for auto-tagging of consumer photos, but also for tour recommendation applications or visualization of flow of travelers in urban settings.

2. Related Work

Many location recognition algorithms use repeatable scale-invariant feature detectors combined with multi-view spatial verification methods to match scenes with distinctive appearance (e.g., [14, 12, 21, 15]). To cope with the massive amount of local features that must be indexed, such methods often require novel retrieval strategies, including hierarchical vocabularies and informative feature selection [12, 15]. Access to many users' online collections together with such effective local feature matching methods have led to compelling new ways to browse community photo collections [18, 17], and to discover iconic sites [16, 13, 8, 4]. In particular, the authors of [17] show how to discover viewpoint clusters around a landmark photographed by many people, so as to automatically compute navigation controls for image-based rendering when later interactively perusing the photos in 3D.

Recent work shows that with databases containing millions of geo-tagged images, one can employ simpler global image descriptors (e.g., Gist, bag-of-words histograms) and still retrieve relevant scenes for a novel photo [6, 9, 7, 22]. While some are tested with landmark-scale metrics [9, 22], others aim to predict locations spanning hundreds of kilometers [6, 7]. While an intriguing use of "big data", such measures are likely too coarse to be useful for auto-tagging applications. (For example, with a geo-tagged database of 6 million images, nearest neighbors labels 16% of a test set correctly, and only within 200 km of the true location [6].)

For either type of approach, an important challenge is that many images simply don't capture distinctive things. In realistic situations, many traveling photos do not contain visually localizable landmarks—such as group photos, vehicles, or a snapshot in a McDonald's. Thus far, most techniques sidestep the issue by manually removing such instances during dataset creation. Some attempt to prune them automatically by building classifiers [13] or manually-defined tag rules for the image crawler (i.e., exclude images tagged with ‘wedding’ [6, 7]), or else bolster the features with textual tags [9]. Instead, we explore how estimating locations for sequences—using both bursts and within-city travel patterns—can overcome this common failure mode in a streamlined way. As a result, the approach can correctly tag more diverse photos, without requiring an expansion to the underlying labeled dataset.

While timestamps have long been used to organize photos into clusters or “temporal events” [10, 3, 2], much less work considers how temporal cues might improve location estimation itself. Perhaps most related to our work, the authors of [7] develop an HMM-model parameterized by time *intervals* to predict locations for photo sequences taken along transcontinental trips. Their work also exploits human travel patterns, but at a much coarser scale: the world is binned into 3,186 400 km^2 bins, and transitions and test-time predictions are made among only these locations. Whereas that approach leverages physical travel constraints and common flight patterns (i.e., one cannot be in Madrid one hour and Brazil the next), our method learns patterns in how tourists visit sites within a popular city, and predicts labels at the granularity of interest for auto-tagging (i.e., it will tag an image as ‘Colosseum’ rather than simply ‘Italy’). A further distinction is our proposed set-to-set observation likelihood, which we show outperforms an image-based likelihood as used in [7].

The authors of [9] consider location recognition as a multi-class recognition task, and the five images before and after the test image serve as temporal context within a structured SVM model. This strategy is likely to have similar label smoothing effects as our method’s initial burst grouping stage, but does not leverage statistics of travel patterns. Further, because that approach targets cross-city recognition, the smoothing likely occurs at a higher granularity, i.e., to keep a sequence of predictions within the same city. Outside of the tourist photo-tagging domain, previous work with wearable cameras has shown that the temporal context is useful for video-based room recognition [19, 20]; in contrast to our widely variable consumer photos, such data has the advantage of both dense sampling in time and restricted well-defined locations.

3. Approach

We present the proposed algorithm divided into its training and testing stages. During training, we use geo-tags on labeled images to quantize a city into its locations of interest, record the priors and transition probabilities between these locations using the training sequences, and extract visual features from each image. During testing, we are given a novel photo sequence, divide it into a series of burst observations using the timestamps, extract visual features, and then estimate the location for each burst via inference on the HMM. This section explains these steps in detail.

3.1. Training Stage

The training images for a given city originate from online photo collections, and each has a timestamp, GPS geo-tag, and user ID code. In our experiments, we download about 75K-100K training images per city, from over 1000 photographers each. Note that the user IDs allow us to extract sequences of photos from the same photographer.

Discovering a City’s Locations. Rather than define the true locations of interest with a fixed grid—which could artificially divide important sites—we use a data-driven approach to discover the regions visited by tourists. Specifically, we apply mean shift clustering to the GPS coordinates of the training images. Then, each location that emerges is a hidden state in our model. Figure 2 depicts the locations for our datasets. At test time, we will estimate to which of the discovered locations the novel images belong.

Visual Feature Extraction. For every image, we extract three visual features: Gist, a color histogram, and a bag of visual words. Gist captures the global scene layout and texture [11], while the color histogram characterizes certain scene regions well (e.g., green plants in a park, colorful lights downtown). The bag-of-words descriptor summarizes the frequency with which prototypical local SIFT patches occur; it captures the appearance of component objects, without the spatial rigidity of Gist. For images taken in an identical location, this descriptor will typically provide a good match. Note, however, that we forgo the geometric verification on local features typically done by pure landmark-matching systems (e.g., [12]). While it would certainly refine matching results for distinctive buildings or monuments, we also care about inexact matches for non-distinctive scenes (e.g., a view of the bay, or a bus stop), in order to get a distribution over possible locations that can be exploited well during HMM inference.

Location Summarization. Since the data consists of users’ uploaded images, certain popular locations contain many more images than others. The higher density of images has potential to both help and hurt performance. On the one hand, more examples means more coverage, or less chance to miss a corresponding scene at test time. On the other hand, more examples usually also means more “noisy” non-distinct images (portraits, pictures of food) that can bias the observation likelihood if the locations are imbalanced. For example, if 5% of the training images contain a car, then the most popular locations will likely contain quite a few images of cars. At test time, any image containing a car could have a strong match to them, even though it may not truly be characteristic of the location.

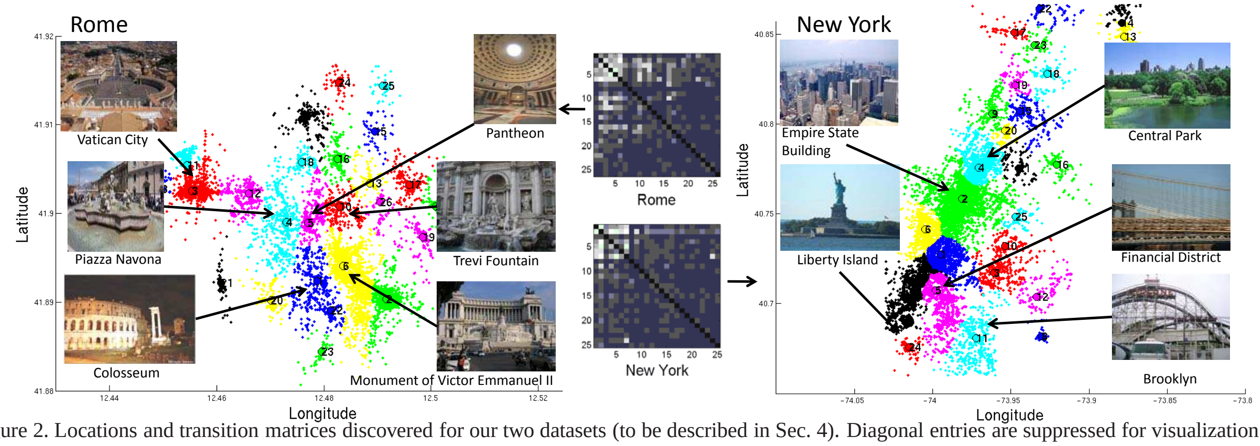

Figure 2. Locations and transition matrices discovered for our two datasets (to be described in Sec. 4). Diagonal entries are suppressed for visualization.

Thus, we consider an optional location summarization procedure to focus the training images used to compute our set-based likelihood function (which is defined in Section 3.2). The idea is to automatically select the most important aspects of the location with minimal redundancy. We apply the efficient spherical k-centroids algorithm [5] to each location's training images. These centroids then serve as the representative instances to which we attempt to match novel photos. However, we use all training images when computing transition and prior probabilities. We show results both with and without summarization below.

Learning the Hidden Markov Model. We represent the learned tourist travel patterns with a Hidden Markov Model. An HMM is defined by three components: the initial state priors, the state transition probabilities, and the observation likelihood. We define the first two here and defer the likelihood to our description of the testing stage below.

The location prior is derived from the distributions of images in the training set. Suppose we have N locations defined via mean shift for the current city. Let N_i denote the number of images taken within the i-th location. The prior for the location state at any time t is then simply:

$$P(L_t = i) = \frac{N_i + \lambda_L}{\sum N_i + \lambda_L}, 1 \leq i \leq N, \quad (1)$$

where λ_L is a regularization constant to make it possible to begin in locations that may not have been present in the training set.

The transition probabilities reveal the pattern of typical human movement between locations in the city. The transition probability between two photo bursts $t-1$ and t is:

$$P(L_{t-1} = i | L_t = j) = \frac{N_{ij} + \lambda_t}{\sum N_{ij} + \lambda_t}, \quad 1 \leq i, j \leq N, \quad (2)$$

where N_{ij} is the number of transitions from location i to j among the training sequences, and λ_t is a regularization constant to avoid treating any transition as impossible.

3.2. Testing Stage

Given a novel sequence of a single tourist's timestamped images, we divide it into a series of bursts, and then estimate their locations by running inference on the HMM.

Grouping Photos into Bursts. A burst is meant to capture a small event during traveling, such as a visit to some landmark, a dinner in a restaurant, entering a museum, or taking a ferry. When inferring the labels for a novel sequence, we will assume that all photos in a single burst have the same location label. The ideal method should make the bursts large enough to substantially benefit from the label smoothing effects, but small enough to ensure only a single location is covered.

We use mean shift on the timestamps to compute the bursts, which is fairly flexible with respect to the frequency with which people take photos. We have also explored alternatives such as an adaptive bandwidth mean shift and grouping using both temporal and visual cues, but we found each variant to produce similar final results, and thus choose the timestamp method for its simplicity.

Location Estimation via HMM Inference. Let $S = [B_1, \ldots, B_T]$ denote the series of T bursts in a novel test sequence, where each $B_t = \{I_{t_1}, \ldots, I_{t_G}\}$ is a group of photos in an individual burst, for $t = 1, \ldots, T$. For example, the whole sequence S might span several days of touring, while each burst B_t would typically consist of photos taken within $\sim$30 minutes. For convenience, below we will simply use G to denote $|B_t|$, the cardinality of the t-th burst, though it varies per burst.

Our goal is to estimate the most likely series of locations: $\{L_1^*, \ldots, L_T^*\}$, where each L_t^* denotes the location label attributed to each image in the t-th burst. To estimate these labels, we need to define the observation likelihood distribution, $P(I_{t_1}, \ldots, I_{t_G} | L_t = i)$, for $i = 1, \ldots, N$. Our definition must reflect the fact that some images within a burst may not have a strong match available in the training set, even for those taken at the same true location (e.g., imagine

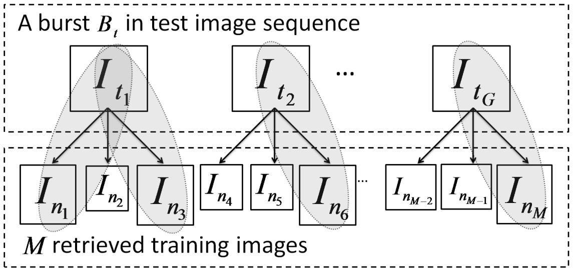

Figure 3. Illustration of Eqns. 3 and 4. Given a burst B_t that contains G images, say we retrieve $K = 3$ neighbors for each test image, giving $3 \times G$ retrieved training images. Among them, image 1, 3, 6, and M are from L_1, which means that, $M_1 = \{I_{n_1}, I_{n_3}, I_{n_6}, I_{n_M}\}$. Thus, the numerator in Equation 4 is affected by the four $D(I_{t_*}, I_m)$ pairs circled in the figure.

a close-up facial portrait taken while waiting in line at the Statue of Liberty). Thus, we treat the photos within a burst as a single observation, and define a set-to-set likelihood model to robustly measure the visual similarity between the burst and any given location.

For each image in the burst, we gather its K most similar images across the entire training set using the visual features defined in Section 3.1 and a Euclidean distance weighted per feature type. (If we are using the optional summarization stage, these neighbors are chosen among only the representative training image exemplars.) Note that while retrieving these nearest neighbor images, we ignore which state each neighbor comes from. This yields a set of $M = K \times G$ neighbor training images $\{I_{n_1}, I_{n_2}, \ldots, I_{n_M}\}$.

Next we divide the M images according to their true locations, yielding one set per location, $M_1, M_2, \ldots, M_N$, some of which may be empty. Now we define the probability that the location is i given burst t:

$$P(L_t = i | I_{t_1}, \ldots, I_{t_G}) \propto \left(\sum_{m \in M_i} \omega(I_m) \right) + \lambda_c, \qquad (3)$$

where M_i denotes the set of retrieved images coming from location i, and λ_c is a regularization constant to ensure that each location has nonzero probability. For every retrieved image I_m in M_i, its contribution to the location likelihood above is given by:

$$\omega(I_m) = \frac{\exp(-\gamma D(I_{t_*}, I_m))}{\sum_{l=1}^{M} \exp(-\gamma D(I_{t_*}, I_{n_l}))}, \qquad (4)$$

where I_{t_*} denotes the burst's image that was nearest to I_m when doing the initial neighbor retrieval, and γ is a standard scaling parameter. The distance $D(I_{t_*}, I_m)$ defines the visual feature similarity between the two images, and is the same weighted Euclidean distance over the Gist, color, and bag-of-word descriptions used to retrieve the neighbors. The denominator of Eqn. 4 normalizes according to the extent to which I_{t_*} is similar to retrieved images in all locations. Please see Figure 3 for an illustration.

Finally, we can use the above to compute the image burst B_t's likelihood via Bayes Rule:

$$\begin{aligned} P(B_t | L_t = i) &= \frac{P(L_t = i | I_{t_1}, \ldots, I_{t_G}) P(I_{t_1}, \ldots, I_{t_G})}{P(L_t = i)}, \\ &\propto \frac{(\sum_{m \in M_i} \omega(I_m)) + \lambda_c}{N_i + \lambda_L}, \end{aligned} \qquad (5)$$

where $P(I_{t_1}, \ldots, I_{t_G})$ is constant. Bayes Rule is also used in this reverse fashion in [7], although in that model the likelihood is computed for a single image.

Having already defined the location prior and location transition probabilities during the training stage, we can use this burst likelihood to predict novel sequences of locations. We use the Forward Backward algorithm, and estimate the locations based on the hidden state with the maximum marginal at each burst.

A simpler alternative HMM would consider *each image* in the sequence as a individual observation, and estimate the likelihood term according to the nearest neighbor for that image [7] or using some parametric distribution [20]. The advantage of our burst-based likelihood over such an "image-based" HMM is that strong matches for some portion of the burst can influence the probability, while the noisy or non-distinct views are discounted. Furthermore, images that do not provide good cues for determining location—yet *do* appear in most locations (e.g., human, car)—will not strongly influence the probabilities due to the normalization in Eqn. 4. We directly validate the impact of the burst design in the experiments.

Note that our approach is intended to auto-tag sites within a given city; in our experiments we treat the city itself as given (i.e., we know a batch of snapshots are taken from New York). This is a reasonable assumption, given that people typically put photos in an album or folder with at least this specificity very easily. However, one could potentially automate the city label as well based on identifying a match for even one distinctive image, and thereby automatically choose which model to apply.

4. Experimental Results

Our experiments demonstrate the approach with real user-supplied photos downloaded from the Web. We make direct comparisons with four key baselines, and analyze the impact of various components.

Data Collection. Existing datasets lack some aspects necessary to test our method: several collections [6, 9, 7] are broader than city-scale, whereas we are interested in within-city location tagging; others are limited to streetside images taken from a vehicle rather than a tourist [15]. Therefore, we collect two new datasets from Flickr, maintaining all user IDs, timestamps, and ground truth GPS.

We consider two major tourist cities, New York and Rome. We obtain the data by querying Flickr for images

Dataset	Rome	New York
# Train\Test Images	32942\22660	28950\28250
# Train\Test Users	604\470	665\877
Avg # photos per test seq	52 (std 119)	37 (std 71)
Avg time period of test seq	3.77 days	3.33 days

Table 1. Properties of the two datasets.

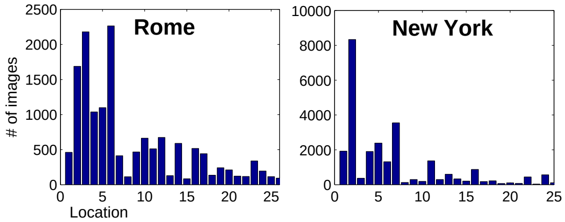

Figure 4. Number of images per location for the two datasets.

tagged with "New York City" or "NYC" or "Rome" within the past two years. Images with geo-tags outside of the urban areas are removed[1], and locations discovered in the training set with fewer than 100 images are discarded. We link images taken by the same user within 30 days to form traveling sequences. To ensure no overlap in the train and test sets, we take images from 2009 to form the training set, and those from 2010 to form the test set. See Table 1 for more stats on the collection.[2]

Image Features and Distance. For each image, we extract a 960-dimensional GIST descriptor, 35-bin color histogram (15 bins for hue and saturation, 5 bins for the lightness), and a DoG+SIFT bag-of-words histogram (1500 words for Rome and 1000 for New York, which we choose based on the size of dataset). While even larger vocabularies are often used for specific landmark matching, ours are coarser in order to allow matches with less distinctive scenes. We normalize each descriptor type by the average nearest feature distance, and then combine them when computing inter-image distances with a weighted Euclidean metric. Specifically, we use a weight ratio of $1 : 2 : 7$ for Gist:color:SIFT, respectively, based on our intuition of the relative strength of each feature, and brief manual inspection of a few initial image neighbors.

Parameters. The HMM has several regularization parameters necessary to maintain nonzero probabilities for each location (see Sec. 3.1). We set $\lambda_L = 200$, based on the median location cluster's diameter, and $\lambda_t = 1500$, based on the median of diagonal entries in the transition matrix. We set $\lambda_c = 10^{-4}$, and $\gamma = 2.0$, based on the distribution of inter-image distances. We use $K = 10$ neighbors for the burst-based likelihood. The bandwidths for mean shift clustering of the geo-tags and images for bursts are set to 0.17/0.6 (Rome/NewYork) mi. and 1.2 hour, respectively, based on the scale of the cities and typical human behavior. We did not optimize parameter values for performance; initial tests indicated that the parameters affect our method and the baselines similarly.

	NN	Img- HMM	Int- HMM	Burst Only	Burst-HMM (Ours)
Avg/seq	0.1502	0.1608	0.1728	0.1764	**0.2036**
Overall	0.1592	0.1660	0.1771	0.2617	**0.2782**

(a) Rome dataset

	NN	Img- HMM	Int- HMM	Burst Only	Burst-HMM (Ours)
Avg/seq	0.2323	0.2124	0.2330	0.2099	**0.3021**
Overall	0.2302	0.2070	0.2304	0.2055	**0.3143**

(b) New York City dataset

Figure 5. Location estimation accuracy on Rome (a) and New York (b).

City Location Definitions. Mean shift on the training data discovers 26 locations of interest for Rome, and 25 for New York (see Figures 2 and 4). The average location size is 0.2 mi^2 in Rome, and 3 mi^2 in New York. The ground truth location for each test image is determined by the training image it is nearest to in geo-coordinates.

Baseline Definitions. To verify the advantages of our approach, we compare it to four baselines: (1) **Nearest Neighbor (NN)** classification using the image features (as in [6]), (2) an image-to-image HMM we refer to as **Img-HMM**, where each image is treated as an observation, but transitions and priors are the same as in our model, (3) an interval-based HMM **Int-HMM**, which is similar to Img-HMM but uses time intervals to define transitions as in [7], and (4) a **Burst Only** baseline, which uses the same bursts as computed for our method, but lacks the travel transitions and priors (i.e., this baseline uses the burst-likelihood alone to classify a burst at a time). We use the same visual features for all methods to ensure the fairest comparison.

The NN baseline uses no temporal information, whereas the Burst Only only uses the timestamps to cluster the images, but no transition priors. The Img-HMM is equivalent to our system when each burst is restricted to be a single photo. An Int-HMM is used in [7], and NNs are used in [6]. These are the most important baselines to show the impact of our algorithm design, since they cover both alternate choices one could make in an HMM design for this problem, as well as state-of-the-art methods for location recognition.

Location Estimation Accuracy. Figure 5 compares the different methods' performance. We report both the average rate of correct predictions across the test sequences, as well as the overall correct rate across all test images. Our Burst-HMM achieves the best accuracy under both metrics, by a substantial margin in most cases. Our method's improvements over the Burst Only baseline show the clear advantage of modeling the "beaten path" hints, while our gains over the Img-HMM and Int-HMM show that the burst-based likelihood has the intended robustness.

The Int-HMM outperforms the Img-HMM. This difference is slight, though, likely because the physical con-

[1] NY lng -74.03 - -73.86, lat 40.66-40.87; Rome lng 12-13, lat 40-42.
[2] Our data is available at http://vision.cs.utexas.edu/projects/.

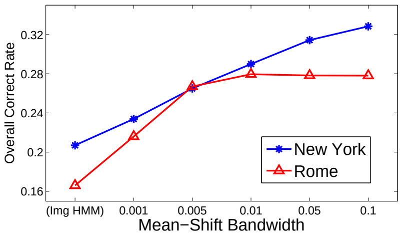

Figure 6. The Img-HMM is a special case of our algorithm where the bandwidth is fixed at 0 (i.e., single image bursts).

straints conveyed by the time intervals are weaker within a city than they are across continents (as was the focus in [7]); that is, while a person cannot move 300 miles in 30 minutes, he may reach many locations in a city within a similar amount of time.

The NN baseline fares much better on the New York data than on Rome; upon inspection, we found that this is due to the highly imbalanced distribution of images per location within the New York dataset (e.g., 32% comes from Times Square). The NN approach matches many test images to some image belonging to those popular/well-covered locations, which happens to be an advantage in this case. For Rome, where locations are more balanced in coverage, we see how NN suffers from the lack of temporal context relative to all other methods. In general, adding a geometric verification stage to the feature-based matching could boost results for distinctive landmarks. However, good matching approaches yield $\sim$0.6 AP when *all* queries contain a recognizable landmark (e.g., [12]), while in our test data fewer than half of the images actually do.

Figure 8 shows an example test sequence from either dataset, along with our predictions and those of the Img-HMM baseline. The result illustrates the expected tradeoffs of the two approaches.

Impact of Burst Density. On average, each burst in the test set has 7.7 images for Rome, and 6.1 for New York. Since our method assigns all images within a burst with the same label, the precision of these automatically computed bursts will influence our final results. Using a fixed bandwidth for mean shift, the percentage of images within a burst that are from the same location is 79% and 91% for either dataset. Thus, we do give up a small number of correct labels from the start; however, as seen in the Img-HMM and Burst-HMM comparisons, this loss is outweighed by the accompanying stronger likelihood function. One might be able to generalize our approach to use a "soft" assignment into bursts, but we leave this as future work.

As the size of the bursts decreases, our model approaches the Img-HMM model. We illustrate this in Figure 6, plotting our test performance as a function of the mean-shift bandwidth. Indeed, we see that accuracy converges to the Img-HMM baseline once the bandwidth is 0, i.e., each burst is a single image (leftmost points). We also see that our accuracy is fairly insensitive to a range of values for this parameter. This indicates the range of typical photo-taking frequencies among the hundreds of Flickr users in our data is well-handled by the density-based mean shift clustering.

	NN		Img-HMM		Burst-HMM (Ours)	
Summ?	No	Yes	No	Yes	No	Yes
Avg/seq	0.2323	0.1502	0.2124	0.2674	0.3021	**0.3108**
Overall	0.2302	0.1592	0.2070	0.2797	0.3143	**0.3475**

Table 2. Effect of summarization for the imbalanced New York dataset.

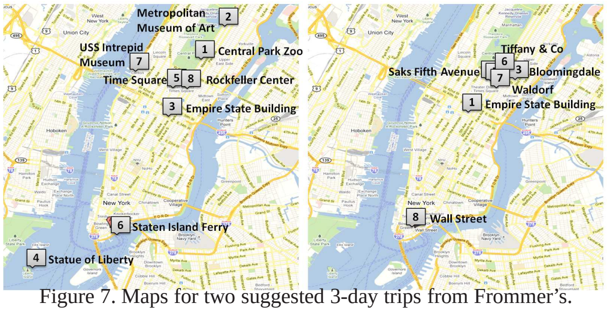

Figure 7. Maps for two suggested 3-day trips from Frommer's.

	Rand. Walk	Rand. Walk(TS)	Guidebook
Route Prob.	$6.3 \cdot 10^{-12}$	$4.2 \cdot 10^{-11}$	$2.0 \cdot 10^{-4}$

Table 3. Probability of recommended vs. random routes under our model.

Impact of Location Summarization. Since an extreme location imbalance bias is detrimental to the HMM methods, we further explored an optional location *summarization* stage (see Sec. 3.1). Table 2 shows the results for the New York data. The representative images selected per location do indeed resolve this issue, improving results for both our method and the Img-HMM baseline, and decreasing accuracy for NN, since it is no longer favorably biased to predict the most popular locations. Summarization had no significant impact on the Rome data, since the distribution of location images is more balanced (see Fig. 4).

Discovering Travel Guides' Beaten Paths. Travel guides often suggest popular itineraries for exploring a city. Does anyone actually use them? We can find out with our model! If the data is representative and our model is working well, we expect the suggested travel patterns to be assigned high probability by our Burst-HMM. Thus, we obtain 7 itineraries for spending "3 days in NYC" from 7 popular travel guides with postings online (e.g., Frommers). Then, we compare their path probability to that of a random walk. We consider two forms of random walks: one with a random initial state sampled from the prior, and one specifically starting from the popular Times Square (TS) location. We generate 10,000 random paths for each baseline.

Table 3 shows the results. The mean of the 7 guidebook itineraries is indeed higher than the random walks, and all individual guidebook paths are significantly more probable. Figure 7 depicts two of the guidebook itineraries; both are much more probable than a random walk of sites in NYC, even if starting in the popular Times Square.

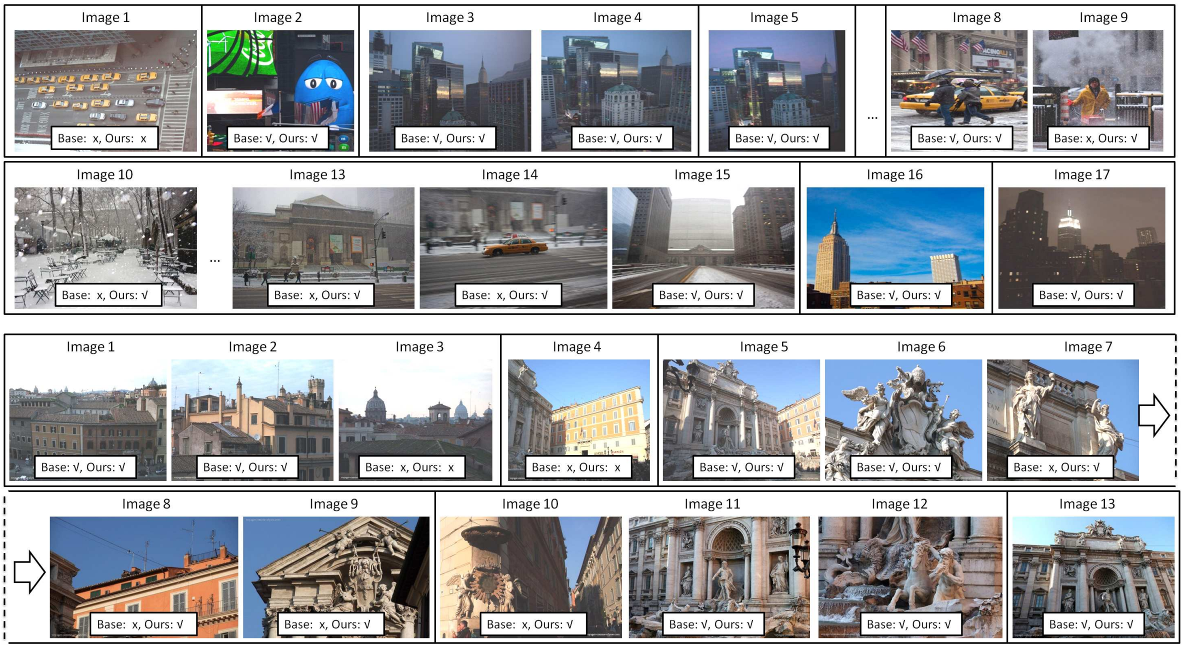

Figure 8. Example results comparing predictions by our Burst-HMM ("Ours") and the Img-HMM baseline ("Base"). Images in the same cell are from the same burst. A check means correct prediction, an 'x' means incorrect. **Top, NY:** Images with distinct features (such as Images 2-5, and 16-17) are predicted correctly by both methods. While the baseline fails for less distinctive scenes (e.g., Image 8-14), our method estimates them correctly, likely by exploiting both informative matches to another view within the burst (e.g., the landmark building in Image 8 or 13), as well as the transitions from burst to burst. Our method can also fail if a burst consists of only non-distinctive images (Image 1). **Bottom, Rome:** Similarly, we correctly locate the 2nd burst due to the strong hints in Images 5 and 6 for the famous location, while the baseline fails on Image 8 due to the lack of temporal constraints and distinctive features.

5. Conclusions

We presented a novel approach that learns and exploits the behavior of tourist photographers. We are the first to show how travel patterns over several days can strengthen within-city recognition, and to explore how photo burst events serve as a powerful labeling constraint. Our results show that even with relatively limited labeled data, the Burst-HMM is more accurate than traditional NN matching and two image-based HMM methods. Our findings also hint at novel future applications beyond auto-tagging, such as collaborative filtering of vacation itineraries.

Acknowledgements This research is supported in part by DARPA CSSG N10AP20018 and the Luce Foundation.

References

[1] *http://techcrunch.com/2009/04/07/who-has-the-most-photos-of-them-all-hint-it-is-not-facebook/*.

[2] L. Cao, J. Luo, H. Kautz, and T. S. Huang. Annotating collections of photos using hierarchical event and scene models. In *CVPR*, 2008.

[3] M. Cooper, J. Foote, A. Girgensohn, and L. Wilcox. Temporal event clustering for digital photo collections. In *AVM MM*, 2003.

[4] M. Cristani, A. Perina, U. Castellani, and V. Murino. Geo-located image analysis using latent representations. In *CVPR*, 2008.

[5] I. S. Dhillon and D. S. Modha. Concept decompositions for large sparse text data using clustering. In *Machine Learning*, 2001.

[6] J. Hays and A. A. Efros. im2gps: estimating geographic information from a single image. In *CVPR*, 2008.

[7] E. Kalogerakis, O. Vesselova, J. Hays, A. Efros, and A. Hertzmann. Image sequence geolocation with human travel priors. In *ICCV*, 2009.

[8] X. Li, C. Wu, C. Zach, S. Lazebnik, and J.-M. Frahm. Modeling and recognition of landmark image collections using iconic scene graphs. In *ECCV*, 2008.

[9] Y. Li, D. J. Crandall, and D. P. Huttenlocher. Landmark classification in large-scale image collections. In *ICCV*, 2009.

[10] A. C. Loui. Automatic image event segmentation and quality screening for albuming applications. In *ICME*, 2000.

[11] A. Oliva and A. Torralba. Building the gist of a scene: the role of global image features in recognition. In *Prg in Brain Rsrch*, 2006.

[12] J. Philbin, O. Chum, M. Isard, J. Sivic, and A. Zisserman. Object retrieval with large vocabularies and fast spatial matching. In *CVPR*, 2007.

[13] T. Quack, B. Leibe, and L. V. Gool. World-scale mining of objects & events from community photo collections. In *CIVR*, 2008.

[14] F. Schaffalitzky and A. Zisserman. Multi-view matching for unordered image sets, how do I organize my holiday snaps? In *ECCV*, 2002.

[15] G. Schindler, M. Brown, and R. Szeliski. City-scale location recognition. In *CVPR*, 2007.

[16] I. Simon, N. Snavely, and S. M. Seitz. Scene summarization for online image collections. In *ICCV*, 2007.

[17] N. Snavely, R. Garg, S. M. Seitz, and R. Szeliski. Finding paths through the world's photos. In *SIGGRAPH*, 2008.

[18] N. Snavely, S. M. Seitz, and R. Szeliski. Photo tourism: Exploring photo collections in 3d. In *SIGGRAPH*, 2006.

[19] T. Starner, B. Schiele, and A. Pentland. Visual contextual awareness in wearable computing. In *Intl Symp on Wearable Comp*, 1998.

[20] A. Torralba, K. Murphy, W. Freeman, and M. Rubin. Context-based vision system for place & object recognition. In *ICCV*, 2003.

[21] W. Zhang and J. Kosecka. Image based localization in urban environments. In *3DPVT*, 2006.

[22] Y. Zheng, M. Zhao, Y. Song, H. Adam, U. Buddemeier, A. Bissacco, F. Brucher, T. Chua, and H. Neven. Tour the world: building a web-scale landmark recognition engine. In *CVPR*, 2009.