Noname manuscript No.
(will be inserted by the editor)

Modeling and Recognition of Landmark Image Collections Using Iconic Scene Graphs

Rahul Raguram · Changchang Wu · Jan-Michael Frahm · Svetlana Lazebnik

Received: date / Accepted: date

Abstract This article presents an approach for modeling landmarks based on large-scale, heavily contaminated image collections gathered from the Internet. Our system efficiently combines 2D appearance and 3D geometric constraints to extract scene summaries and construct 3D models. In the first stage of processing, images are clustered based on low-dimensional global appearance descriptors, and the clusters are refined using 3D geometric constraints. Each valid cluster is represented by a single iconic view, and the geometric relationships between iconic views are captured by an *iconic scene graph*. Using structure from motion techniques, the system then registers the iconic images to efficiently produce 3D models of the different aspects of the landmark. To improve coverage of the scene, these 3D models are subsequently extended using additional, non-iconic views. We also demonstrate the use of iconic images for recognition and browsing. Our experimental results demonstrate the ability to process datasets containing up to 46,000 images in less than 20 hours, using a single commodity PC equipped with a graphics card. This is a significant advance towards Internet-scale operation.

1 Introduction

Today, more than ever before, it is evident that "to collect photography is to collect the world" [Sontag, 1977]. More and more of the Earth's cities and sights are photographed each day from a variety of digital cameras, viewing positions and angles, weather and illumination conditions; more and more of these photos get tagged by users and uploaded to photo-sharing websites. For example, on Flickr.com, locations form the single most popular category of user-supplied tags [Sigurbjörnsson and van Zwol, 2008]. With the growth of community-contributed collections of place-specific imagery, there comes a growing need for algorithms that can distill their content into representations suitable for summarization, visualization, and browsing.

In this article, we consider collections of Flickr images associated with a landmark keyword such as "Statue of Liberty," with often noisy annotations and metadata. Our goal is to efficiently identify all photos that actually represent the landmark of interest, and to organize these photos to reveal the spatial and semantic structure of the landmark. Any system that aims to meet this goal must address several challenges inherent in the nature of the data:

- **Contamination**: When dealing with community-contributed landmark photo collections, it has been observed that keywords and tags are accurate only approximately 50% of the time [Kennedy et al., 2006]. Since we obtain our input using keyword searches, a large fraction of the input images comprises of "noise," or images that are unrelated to the concept of interest.
- **Diversity:** The issue of contamination aside, even "valid" depictions of landmarks have a remarkable degree of diversity. Landmarks may have multiple aspects (sometimes geographically dispersed), they may be photographed at different times of day and in different weather conditions, to say nothing of non-photorealistic depictions and cultural references (Figure 1).
- **Scale**: The typical collection of photos annotated with a landmark-specific phrase has tens to hundreds of thousands of images. For example, there

Dept. of Computer Science, University of North Carolina, Chapel Hill, NC 27599-3175
E-mail: {rraguram,ccwu,jmf,lazebnik}@cs.unc.edu

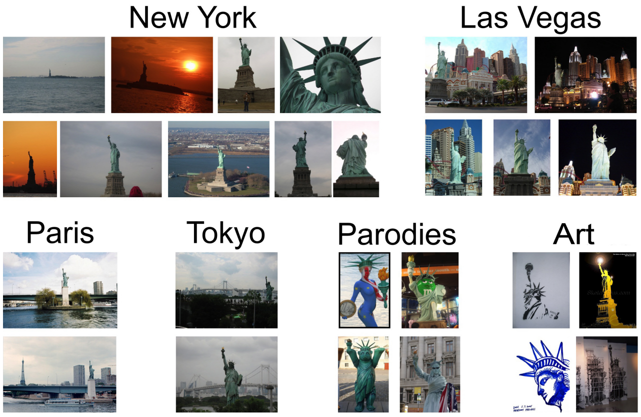


Fig. 1 The diversity of photographs depicting "Statue of Liberty." There are copies of the statue in New York, Las Vegas, Tokyo, and Paris. The appearance of the images can vary significantly based on time of day and weather conditions. Further complicating the picture are parodies (e.g., people dressed as the statue) and non-photorealistic representations. The approach presented in this article relies on rigid 3D constraints, so it is not applicable to the latter two types of depictions.

> are over 140,000 images on Flickr associated with the keyword "Statue of Liberty." If we wanted to process such collections using a traditional structure from motion (SfM) pipeline, we would have to take every pair of images and try to establish a two-view relation between them. The running time of such an approach would be at least quadratic in the number of input images. Clearly, such brute-force matching is not scalable; we need smarter and more efficient ways of organizing the images.

Fortunately, landmark photo collections also possess helpful characteristics that can actually make large-scale modeling easier. The main such characteristic is redundancy: people tend to take pictures from certain viewpoints and to frame their compositions in consistent ways, giving rise to many large groups of very similar-looking photos. Our system is based on the observation that such groups can be discovered using 2D appearance-based constraints that are considerably more efficient than full-blown SfM constraints, and that the iconic views representing these groups form a complete and concise summary of the scene, so that most of the subsequent computation can be restricted to the iconic views without much loss of content.

Figure 2 gives an overview of our system and Algorithm 1 shows a more detailed summary of the modeling steps. Our system begins by clustering all input images based on 2D appearance descriptors, and then it progressively refines these clusters with geometric constraints to select *iconic images* that represent dominant aspects of the scene. These images and the pairwise geometric relationships between them define an *iconic scene graph*. In the next step, this graph is used for efficient reconstruction of a 3D skeleton model, which is subsequently extended using additional relevant images. Given a new test image, we can register it into the model in order to answer the question of whether the landmark is present in the test image. In addition, as a natural consequence of the structure of our approach, the image collection can be cleanly organized into a hierarchy for browsing.

Since our method efficiently filters out unrelated images using 2D appearance-based constraints, which are computationally cheap, and applies more computationally demanding geometric constraints to much smaller subsets of "promising" images, it is scalable to large photo collections. Unlike approaches based purely on SfM, e.g., [Agarwal et al., 2009], it does not require a massively parallel cluster of hundreds of computing cores and can process datasets consisting of tens of thousands of images within hours on a single commodity PC.

The rest of this article is organized as follows. Section 2 places our research in the context of other related work on landmark modeling. In Section 3 we introduce the steps of our implemented system. Section 4 presents experimental results on three datasets: the Notre Dame cathedral in Paris, Statue of Liberty, and Piazza San Marco in Venice. Finally, Section 5 closes the presentation with a discussion of limitations and directions for future work.

An earlier version of this work was originally presented in [Li et al., 2008]. For the present article, the system has been completely re-implemented to include much faster GPU-based feature extraction and geometric verification, an improved image registration algorithm leading to higher precision and recall, and a new incremental reconstruction strategy delivering larger and more complete models. Videos of computed 3D models, along with complete browsing summaries, can be found on the project website.[1]

2 Previous Work

Our system offers a comprehensive solution to the problems of dataset collection, 3D reconstruction, scene summarization, browsing and recognition for landmark images. In this section, we discuss related recent work in these areas.

At a high level, one of the goals of our work can be described as follows: starting with the heavily contaminated output of an Internet image search query, we want to extract a high-precision subset of images

[1] http://www.cs.unc.edu/PhotoCollectionReconstruction

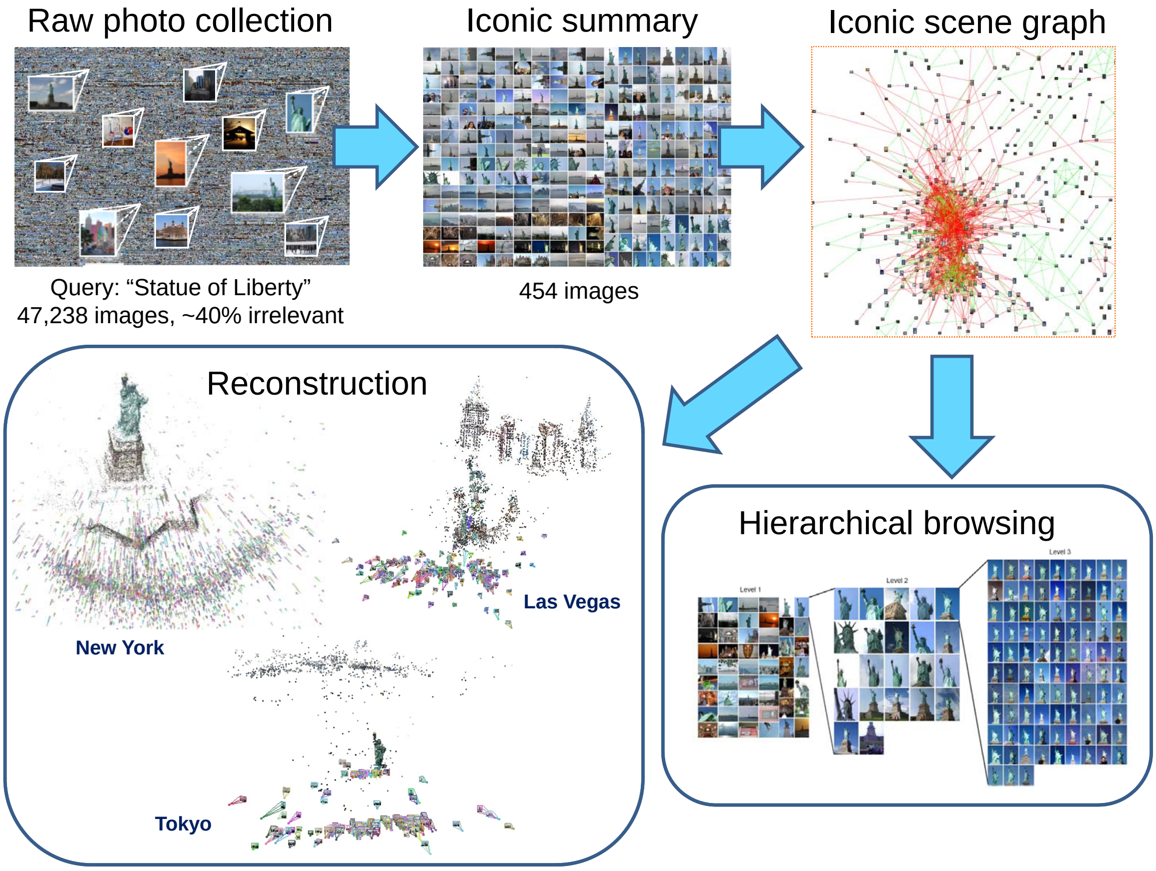


Fig. 2 Overview of our system. The input to the system is a raw, contaminated photo collection, which is reduced to a collection of representative iconic images by 2D appearance-based clustering followed by geometric verification. The geometric relationships between iconic views are captured by the iconic scene graph. The structure of the iconic scene graph is used to automatically generate 3D point cloud models, as well as to impose a hierarchical organization on the images for browsing. Videos of the models, along with a browsing interface, can be found at `www.cs.unc.edu/PhotoCollectionReconstruction`.

that are actually relevant to the query. Several existing approaches consider this problem of *dataset collection* for generic visual categories not characterized by rigid 3D structure[Fergus et al., 2004, Berg and Forsyth, 2006, Li et al., 2007, Schroff et al., 2007, Collins et al., 2008]. These approaches use statistical models to combine different kinds of 2D image features (texture, color, keypoints), as well as text and tags. However, for our specific application of landmark modeling, such statistical models do not provide strong enough geometric constraints. Philbin and Zisserman [2008], Zheng et al. [2009] have presented dataset collection and object discovery methods specifically adapted to landmarks. These methods use indexing based on keypoints followed by loose geometric verification using 2D affine transformations or spatial coherence filters. Unlike them, our method includes an initial stage in which images are clustered using *global* image features, giving us a bigger gain in efficiency and an improved ability to group similar viewpoints. Another difference between our method and [Philbin and Zisserman, 2008, Zheng et al., 2009] is that we perform geometric verification by applying full 3D SfM constraints instead of loose 2D spatial constraints.

To discover all the images belonging to the landmark, we first try to find a set of *iconic views*, corresponding to especially popular and salient aspects. Recently, a number of papers have proposed a very general notion of *canonical* or *iconic images* as good representative images for arbitrary visual categories [Berg and Berg, 2009, Jing and Baluja, 2008, Raguram and Lazebnik, 2008]. These approaches try to find iconic images essentially by 2D image clustering, with some possible help from additional features such as text. Berg and Forsyth [2007], Kennedy and Naaman [2008] have used similar 2D cues to select representative views of landmarks without taking into account the full 3D constraints associated with landmark scenes.

For rigid 3D object instances, canonical view selection has been studied both in psychology [Palmer et al., 1981, Blanz et al., 1999] and in computer vision [Denton et al., 2004, Hall and Owen, 2005, Weinshall et al., 1994]. Palmer et al. [1981] propose several criteria to determine whether a view is "canonical", one of which is particularly interesting for large image collections: *When taking a photo, which view do you choose?* As observed by Simon et al. [2007], community photo collections provide a likelihood distribution over the viewpoints from which people prefer to take photographs. In this context, canonical view selection can be thought of as identifying prominent clusters or modes of this distribution. Simon et al. [2007] find these modes by clustering images based on the output of local feature matching and epipolar geometry verification between every pair of images in the dataset – steps that are necessary for producing a full 3D reconstruction. While this solution is effective, it is computationally expensive, and it treats scene summarization as a by-product of 3D reconstruction. By contrast, we regard summarization as an image organization step that precedes and facilitates 3D reconstruction.

The first approach for organizing unordered image collections was proposed by Schaffalitzky and Zisserman [2002]. Sparse 3D reconstruction of landmarks from Internet photo collections was first addressed by the *Photo Tourism* system [Snavely et al., 2006, 2008b], which achieves high-quality reconstruction results with the help of exhaustive pairwise image matching and global bundle adjustment of the model after inserting each new view. Unfortunately, this process does not scale to large datasets, and it is particularly inefficient for heavily contaminated collections, most of whose images cannot be registered to each other. The Photo Tourism framework is more suited to the case where a user submits a predominantly "clean" set of photographs for 3D reconstruction and visualization. This is precisely the mode of input adopted by the Microsoft Photosynth software,[2] which is based on Photo Tourism.

After the appearance of Photo Tourism, several researchers have developed more efficient SfM methods that exploit the redundancy in community photo collections. In particular, many landmark image collections consist of a small number of "hot spots" from which photos are often taken. Ni et al. [2007] have proposed a technique for out-of-core bundle adjustment that locally optimizes the "hot spots" and then connects the local solutions into a global one. In this paper, we follow a similar strategy of computing separate 3D reconstructions on connected sub-components of the scene, thus avoiding the need for frequent large-scale bundle adjustment. Snavely et al. [2008a] find *skeletal sets* of images from the collection whose reconstruction provides a good approximation to a reconstruction involving all the images. However, computing the skeletal set still requires as an initial step the exhaustive verification of all two-view relationships in the dataset. Similarly to Snavely et al. [2008a], we find a small subset of the collection that captures all the important scene aspects. But unlike Snavely et al. [2008a], we do not need to compute all the pairwise image relationships in the dataset; instead, we rely on 2D appearance similarity as a rough approximation of the "true" multi-view relationship, and reduce the number of possible pairwise relationships to consider through an initial clustering stage. As a result, our technique is capable of handling datasets that are an order of magnitude larger than those in [Snavely et al., 2008a], at a fraction of the running time. Finally, while we do not assume that our photo collections contain geolocated images, it should be noted that when available, this information can be leveraged to improve efficiency. For instance, an initial rough clustering could be performed using only geographic location data, and the obtained cluster centers could then be used to seed the image-based clustering process. Additional techniques to combine image-based techniques with information obtained from geotags can be found in [Quack et al., 2008, Crandall et al., 2009].

In another related recent work, Agarwal et al. [2009] present a distributed system for reconstructing very large-scale image collections. This system uses the core algorithms from [Snavely et al., 2008b,a], implemented and optimized to harness the massive parallelism of multi-core clusters. To speed up the detection of geometrically related images, Agarwal et al. [2009] use feature-based indexing in conjunction with approximate nearest neighbor search [Arya et al., 1998]. They also use query expansion [Chum et al., 2007] to extend the initial set of pairwise relationships. Using a compute cluster with up to 500 cores, the system of Agarwal et al. [2009] is capable of reconstructing city-scale image collections containing 150,000 images in the span of a day. These collections are larger than ours, but the cloud computing solution is expensive: it costs around \$10,000 to rent a cluster of 1000 nodes for a day.[3] By contrast, our system runs on a single commodity PC and uses a combination of efficient algorithms and low-cost graphics hardware to achieve fast performance. Specifically, our system currently processes up to 46,000 images in approximately 20 hours using a PC with an

[2] http://photosynth.net

[3] http://aws.amazon.com/ec2

Intel core2 duo processor with 3GHz and 2.5GB RAM as well as an NVidia GTX 280 graphics card.

Finally, unlike [Agarwal et al., 2009, Ni et al., 2007, Snavely et al., 2008a,b], our approach is concerned not only with reconstruction, but also with recognition. We pose landmark recognition as a binary problem – given a query image, find out whether it contains an instance of the landmark of interest – and solve it by attempting to retrieve iconic images similar to the test query. To accomplish this task, we use methods common to other state-of-the-art retrieval techniques, including indexing based on local image features and geometric verification [Chum et al., 2007, Philbin et al., 2008]. Of course, alternative formulations of the landmark recognition problem are also possible. For example, Li et al. [2009] perform multi-class landmark recognition using a more statistical approach based on a support vector machine classifier. At present, we have not incorporated a discriminative statistical model into our recognition approach. However, we expect that classifiers trained on automatically extracted sets of iconic images corresponding to many different landmarks would produce very satisfactory results.

3 The Approach

In this section, we present a description of the components of our landmark modeling system. Algorithm 1 gives a high-level summary of these components, and Figure 2 illustrates the operation of the system.

3.1 Initial Clustering

The first step of our system is to identify a small set of *iconic views* to summarize the scene content. Similarly to Simon et al. [2007], we define iconic views as representatives of dense clusters of similar viewpoints. However, while Simon et al. [2007] define similarity of any two views in terms of the number of 3D features they have in common, we adopt a more perceptual criterion. Namely, if there are many images in the dataset that share a very similar viewpoint in 3D, then a number of them will have a very similar image appearance in 2D, and they can be grouped efficiently using a low-dimensional global description of their pixel patterns.

The global descriptor we use is *gist* [Oliva and Torralba, 2001], which was found to be effective for grouping images by perceptual similarity and retrieving structurally similar scenes [Hays and Efros, 2007, Douze et al., 2009]. We generate a gist descriptor for each image in the dataset by computing oriented edge responses at three scales (with 8, 8 and 4 orientations, respectively), aggregated to a 4×4 spatial resolution. In addition, we augment this gist descriptor with color information, consisting of a subsampled image, at 4×4 spatial resolution. We thus obtain a 368-dimensional vector as a representation of each image in the dataset. We implemented gist extraction as a series of convolutions on the GPU[4], achieving computation rates of 170 Hz (see Table 2 for detailed timings).

Algorithm 1 System Overview

1. Initial clustering (Section 3.1): Run k-means clustering on global *gist* descriptors to partition the image collection into clusters corresponding to approximately similar viewpoints and scene conditions.

2. Geometric verification and iconic image selection (Section 3.2): Perform robust pairwise epipolar geometry estimation between a few top images in each cluster. Reject all clusters that do not have enough geometrically consistent images. For each remaining cluster, select an *iconic image* as the image that gathers the most inliers to the other top images, and discard all cluster images inconsistent with the iconic.

3. Re-clustering and registration (Section 3.3): Perform clustering and geometric verification on the images discarded during Step 2. This enables the discovery of smaller iconic clusters. After identifying additional iconic images, make a final pass over the discarded images and attempt to register them to any of the iconics.

4. Computing the iconic scene graph (Section 3.4): Register each pair of iconic images to each other and create a graph whose nodes correspond to iconic images, edges correspond to valid two-view transformations between iconics, and edge weights are given by the number of feature inliers to the respective transformations. This graph will be used to guide the subsequent 3D reconstruction process. Use tag information to reject isolated nodes of the iconic scene graph that are likely to be semantically unrelated to the landmark.

5. 3D reconstruction (Section 3.5): Efficiently reconstruct sparse 3D models from the set of images registered to the iconic representation. The reconstruction proceeds in an incremental fashion, by first building multiple 3D sub-models from the iconics, merging them whenever possible, and finally growing all models by incorporating additional non-iconic images.

In order to identify typical views, we cluster the gist descriptors of all our input images using the k-means algorithm. In this initial stage, it is acceptable to produce an over-clustering of the scene, since in subsequent stages, we will be able to restore links between clusters that have sufficient viewpoint similarity. For this reason, we set the number of clusters k to be fairly high ($k = 1200$ in the experiments, although the outcome is not very dependent on the exact value used). In all of our experiments, the resulting clusters capture the popular viewpoints quite well. In particular, the largest gist clusters tend to be quite clean (Figure 3). If we rank the gist clusters in decreasing order of size, we can see that the top few clusters have a remarkably high precision (Figure 7, Stage 1 curve).

[4] code in preparation for release

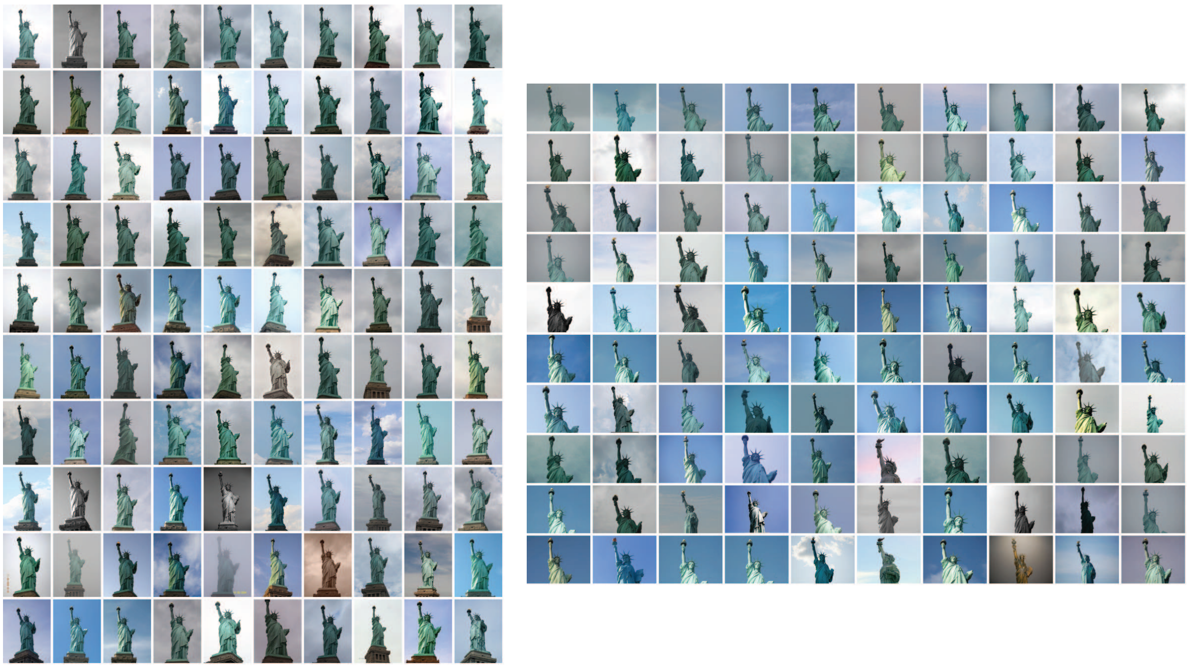


Fig. 3 Snapshots of two gist clusters for the Statue of Liberty dataset. For each cluster, the figure shows the hundred images closest to the cluster mean. Even without enforcing geometric consistency, these clusters display a remarkable degree of structural similarity.

3.2 Geometric Verification and Iconic Image Selection

Of course, clustering based on low-dimensional global descriptors has its drawbacks. For one, the gist descriptor is sensitive to image variation such as clutter (for example, people in front of the camera), lighting conditions, and camera zoom. These factors can cause images with similar viewpoints to fall into different clusters. But also, images that are geometrically or semantically unrelated may end up having very similar gist descriptors and fall into the same cluster. Examples of two clusters with inconsistent images are shown in Figure 4. Since we are specifically interested in recovering scenes with a static 3D structure, we need to enforce strong geometric constraints to filter out structurally inconsistent images from clusters. Thus, the second step of our system consists of applying a geometric verification procedure within each cluster.

The goal of geometric verification is to identify clusters that have at least n images that are consistent in both appearance as well as geometry (in our current implementation, $n = 3$). To this end, we start by selecting an initial subset of n representative images from each cluster by taking the images whose gist descriptors are closest to the cluster mean. Next, we attempt to estimate the two-view geometry of every pair in this subset. Inconsistent images within this subset are identified and replaced by the next closest image to the cluster mean, until a subset of n valid images is found, or all cluster images are exhausted. To test whether a pair of images is consistent, we attempt to estimate a two-view relationship, i.e., epipolar geometry or a homography. A valid epipolar geometry implies that a fundamental matrix exists for freely moving cameras capturing a non-planar scene. A valid homography indicates planar scene structure or rotating cameras.

The standard first step in the robust fitting of a two-view relationship is establishing putative matches between keypoints extracted from both images. We extract SIFT keypoints [Lowe, 2004] using an efficient in-house GPU implementation, SiftGPU[5], which is capable of processing 1024×768 images at speeds of 16Hz on an Nvidia GTX 280. Feature extraction is performed at a resolution that is suitable for the geometric verification task. Empirically, we have observed that SIFT features extracted at the resolution of 1024×768 produce registration results that are comparable those achieved at the original resolution. Putative feature matching is also performed on the GPU. Finding all pairwise distances between SIFT descriptors in the two images reduces to multiplication of large and dense descriptor matrices. Thus, our routine consists of a call to dense matrix multiplication in the CUBLAS library[6] with subsequent instructions to apply the distance ratio test [Lowe, 2004] and to report the established cor-

[5] Available online: http://cs.unc.edu/∼ccwu/siftgpu/

[6] http://developer.download.nvidia.com/compute/cuda/1_0/CUBLAS_Library_1.0.pdf

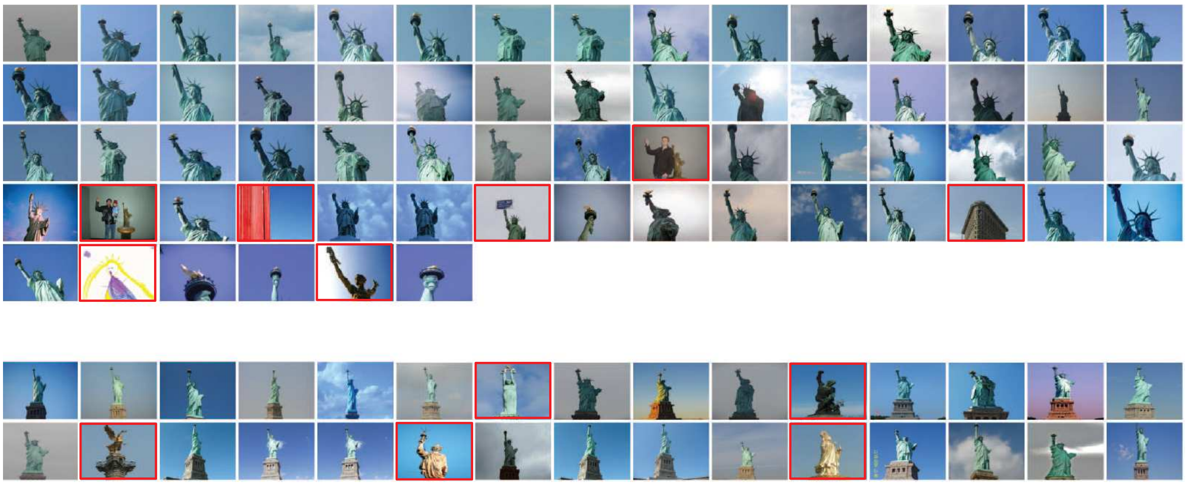


Fig. 4 Snapshots of two clusters containing images inconsistent with the dominant 3D structure. By enforcing two-view geometric constraints, these images (outlined in red) are filtered out.

respondences. To increase the ratio of correct putative matches, we retain only those correspondences that constitute a mutual best match in both the forward and reverse directions.

Once putative matches have been established, we estimate the two-view relationship between the images by applying ARRSAC [Raguram et al., 2008], which is a robust estimation framework capable of real-time performance. We couple this estimation algorithm with QDEGSAC [Frahm and Pollefeys, 2006], which is a robust model selection procedure that accounts for different types of camera motion and scene degeneracies, returning either an estimate for a fundamental matrix or a homography depending on the scene structure and camera motion. If the estimated relation is supported by less than m inliers (m=18 in the implementation), the images are deemed inconsistent.

The image that gathers the largest total number of inliers to the other $n-1$ representatives from its cluster is declared the *iconic image* of that cluster. The inlier score of each iconic can be used as a measure of the quality of each cluster. Precision/recall curves in Figure 7 (Stage 2a) demonstrate that inlier number of the iconic does a better job than cluster size in separating the "good" clusters from the "bad" ones. Note, however, that ranking of clusters based on inlier number of the iconic does not penalize clusters that have a few geometrically consistent images but are otherwise filled with garbage. Once the iconic images for every cluster are selected, we perform geometric verification for every remaining image by matching it to the iconic of its cluster and rejecting it if it has fewer than m inliers. As shown in Figure 7 (Stage 2b), this individual verification improves precision considerably.

As can be seen from Table 2 (Stage 2 column), geometric verification takes just under an hour on the Statue of Liberty dataset, and just under half an hour on the two other datasets. It is important to note that efficiency gains in this step come not only from limiting the number of pairwise geometric verifications, but also from targeting the verifications towards the right image pairs. After all, robust estimation of two-view geometry tends to be fast for images that are geometrically related and therefore have a high inlier ratio, while for unrelated images, the absence of a geometric relation can only be determined by carrying out the maximum number of RANSAC iterations. Since images in the same gist cluster are more likely to be geometrically related, the average number of ARRSAC iterations for within-cluster verifications is comparably low.

3.3 Re-clustering and Registration

While the geometric verification stage raises the precision of the registered images, it also lowers the overall recall by rejecting relevant images that didn't happen to be geometrically consistent with the chosen iconic of their clusters. Such images often come from less common aspects of the landmark that did not manage to get their own cluster initially. To recover such aspects, we pool together all images that were discarded in the previous step, and apply a second round of clustering and verification. As in Section 3.2, we select a single iconic representative per each new valid cluster. As shown in Table 2, this contributes a substantial number of additional iconics to the representation.

After augmenting the initial set of iconics, we perform a final "cleanup" attempting to match each leftover image to the discovered scene structure. In order to efficiently do this, we retrieve, for each leftover image, the k nearest iconics in terms of gist descriptor distance (with k=10 in our current implementation), attempt to register the image to each of those iconics, and assign it to the cluster of the iconic with which it gets the most inliers (provided, of course, that the number of inliers exceeds our minimum threshold).

As seen in Figure 7 (Stage 3), re-clustering and registration increases recall of relevant images from 33% to 50% on the Statue of Liberty, from 46% to 66% on the San Marco dataset, and from 38% to 60% on Notre Dame. In terms of computation time, re-clustering and registration takes about three times as long as the initial geometric verification (Table 2, Stage 3 column). The bulk of this time is spent attempting to register all leftover images to all iconics, since, not surprisingly, the inlier ratios of such images tend to be relatively low. Even with the additional computational expense, the overall geometric verification portion of our algorithm compares quite favorably to that of the fastest system to date [Agarwal et al., 2009], which uses massive parallelism and feature-based indexing to speed up putative matching. On the Statue of Liberty dataset, our system performs both stages of clustering and verification on about 46,000 images in approximately seven hours on one core (Table 2, Totals column). For the analogous processing stages, Agarwal et al. [2009] report a running time of five hours on 352 cores (in other words, 250 times more core hours than our system) for a dataset of only 1.3 times the size.

3.4 Constructing the Iconic Scene Graph

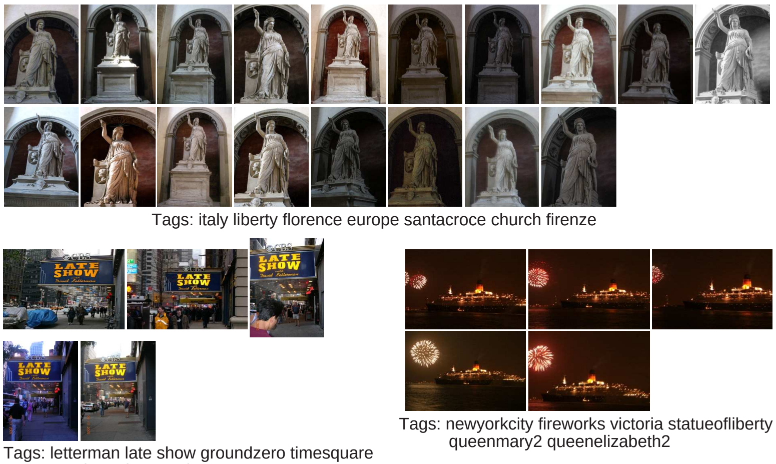
The next step of our system is to build an *iconic scene graph* to capture the full geometric relationships between the iconic views and to guide the subsequent 3D reconstruction process. To do this, we perform feature matching and geometric verification between each pair of iconic images. Note that in our experiments, the number of iconics is orders of magnitude smaller than the total dataset size (several hundred iconics vs. tens of thousands initial images), so exhaustive pairwise verification of iconics is fast. Feature matching is carried out using the techniques described in Section 3.2, but the geometric verification procedure is different. For verifying the geometric consistency of clusters, we sought to estimate a fundamental matrix or a homography. But now, as a prelude to the upcoming SfM stage, we seek to obtain a two-view metric reconstruction.

Pairwise metric 3D reconstructions can be obtained by the five-point relative pose estimation algorithm [Nistér, 2004] and triangulating 3D points based on 2D feature matches. This algorithm requires estimates of internal calibration parameters for each of the cameras. To get these estimates, we make the zero skew assumption and initialize the principal point to be in the center of each image; for the focal length, we either read the EXIF data or use the camera specs for a common viewing angle. In practice, this initialization tends to be within the calibration error threshold of 10% tolerated by the five-point algorithm [Nistér, 2004], and in the latter stages of reconstruction, global bundle adjustment refines the calibration parameters.[7] Note that the inlier ratios of putative matches between pairs of iconic images tend to be very high and consequently, the five-point algorithm requires very few RANSAC iterations. For instance, in the Statue of Liberty dataset, of the image pairs that contain at least 20 putative matches, 80% of the pairs have an inlier ratio larger than 50%.

After estimating the two-view pose for every pair of iconic images, we construct the *iconic scene graph*, where nodes are iconic images, and the weight of the edge connecting two iconics is defined to be the number of inliers to their estimated pose. Iconic pairs with too few inliers (less than m) are given zero edge weight and are thus disconnected in the graph. For all of our datasets, the iconic scene graphs have multiple connected components corresponding to non-overlapping viewpoints, day vs. night views, or even geographically separated instances of the landmark (e.g., copies of the Statue of Liberty in different cities).

In general, lacking GPS coordinates or higher-level knowledge, we do not have enough information to determine whether a given connected component is semantically related to the landmark. However, we have noticed that single-node connected components are very likely to be semantically irrelevant. In many cases, they correspond to groups of near-duplicate images taken by a single user and incorrectly tagged (see Figure 5). To prune out such clusters, we use a rudimentary but effective filter based on image tags. First, we create a "reference list" of tags that are considered to be semantically relevant to the landmark by taking the tags from all iconic images that have at least two connections to other iconics (empirically, these are almost certain to contain the landmark). To have a more complete list, we also incorporate tags from the top ten cluster images registered to these iconics. The tags in the list are ranked in decreasing order of frequency. Next, isolated iconic images

[7] Note that our initial focal length estimate can be wrong for cameras with interchangeable lenses. The error can be particularly large for very long focal lengths, resulting in camera center estimates that are displaced towards the scene points. For example, for a zoomed-in view of Statue of Liberty's head, the estimated camera center may be pushed off the ground towards the head. Some of this effect is visible in Figure 11.



Tags: italy liberty florence europe santacroce church firenze

Tags: letterman late show groundzero timesquare goodmorningamerica

Tags: newyorkcity fireworks victoria statueofliberty queenmary2 queenelizabeth2

Fig. 5 Sample clusters from the Statue of Liberty dataset that were discarded by tag filtering. Each of these clusters is internally geometrically consistent, but does not have connections to any other clusters. By performing a simple tag-based filtering procedure, these spurious iconics can be identified and discarded. Note that in downloading the images, we used Flickr's full text search option, so that "Statue of Liberty" does not actually show up as a tag on every image in the dataset.

are scored based on the median rank of their tags in the reference list. Tags that do not occur in the list at all are assigned an arbitrary high number. Clusters with a high median rank are considered to be unrelated to the landmark and removed from the dataset. As shown by the Stage 4 curves in Figure 7, this further improves precision over the previous appearance- and geometry-based filtering stages.

3.5 Reconstruction

This section describes our novel incremental approach to reconstructing point cloud 3D models from the set of iconic images. The algorithm starts by building multiple 3D sub-models covering the iconic scene graph, then it looks for common 3D features to merge different sub-models, and finally, it grows the resulting models by registering into them as many additional non-iconic views as possible. The sequence of these steps is shown in Algorithm 2 and discussed in detail below.

To initialize the process of incremental 3D reconstruction, we pick the pair of iconic images whose two-view reconstruction (computed as described in Section 3.4) has the highest inlier number and delivers a sufficiently low reconstruction uncertainty, as computed by the criterion of Beder and Steffen [2006]. Next, we iteratively register additional cameras to this model. At each iteration, we propagate correspondences from the reconstructed 3D points to the iconics not yet in the model that see these points. Then we take the iconic that has the highest number of correct 2D-3D correspondences to the current sub-model, register it to the sub-model, and triangulate new 3D points from 2D-2D matches between the iconic and other images already in the model. After each iteration, the 3D sub-model and camera parameters are optimized by an in-house implementation of fast non-linear sparse bundle adjustment[8]. If no further iconics have enough 2D-3D inliers to the current sub-model, the process starts afresh by picking the next best pair of iconics not yet registered to any sub-model. Thus, by iterating over the pool of unregistered iconic images, multiple 3D sub-models are reconstructed.

The above process may produce multiple sub-models that contain overlapping 3D structure and even share some of the same images, but that were not reconstructed together because neither one of the models has a single iconic with a sufficient number of 2D-3D matches to another model. Instead, such models may be linked by a larger number of images having relatively few correspondences each. To account for this case, every time we finish constructing a sub-model, we collect all 3D point matches between it and each of the models already reconstructed, and merge it with a previous model provided a sufficient number of such matches exist (≥ 25, in our experiments). The merging step uses ARRSAC to robustly estimate a similarity transformation based on the identified 3D matches.

Even after the initial merging, we may end up with several separate sub-models coming from the same connected component of the iconic scene graph. This happens when none of the connections between iconic images in different sub-models are sufficient for direct reg-

[8] Available online: http://cs.unc.edu/∼cmzach/oss/SSBA-1.0.zip

Algorithm 2 Incremental Reconstruction

```
Extract connected components of the iconic scene graph and their spanning trees.
for each connected component do

  [Incrementally build and merge sub-models.]
  while suitable initial iconic pairs exist do
    Create 3D sub-model from initial pair.
    while there exist unregistered iconic images with enough 2D-3D matches to sub-model do
      Extend the sub-model to the image with the most 2D-3D matches.
      Perform bundle adjustment to refine the sub-model.
    end while
    Check for 3D matches between current sub-model and other sub-models in the same component and merge sub-models if there are sufficient 3D matches.
  end while

  [Attempt to discover non-iconic links between models.]
  for each pair of sub-models connected in the spanning tree do
    Search for additional non-iconic images to provide common 3D points between the sub-models (see text).
    If enough common 3D points are found, merge the sub-models.
  end for

  [Grow models by registering non-iconic images.]
  for each 3D model do
    while there exist unregistered non-iconic images with enough 2D-3D matches to the model do
      Expand model by adding non-iconic image with highest number of 2D-3D correspondences.
      If more than 25 images added to model, perform bundle adjustment.
    end while
  end for

end for
```

istration. To successfully merge such models, we need to search for additional *non-iconic* images to provide the missing links. To identify the most promising merging locations, we consider the maximal spanning tree (MST) of the iconic scene graph. In this representation, each sub-model reconstructs a subset of images in a MST. We consider neighbouring pairs of source and target iconics that belong to two different models and are connected by an edge of the MST, and we search for non-iconic images that can be registered to both of them. The search is conducted in the iconic clusters of the source and the target, as well as in the clusters of other iconics connected to the source in the MST (Figure 6). To maintain efficiency, we stop the search after finding five images with a sufficient number (≥ 20) of correspondences *both* to the source and the target. The triplet correspondences are then registered into the 3D sub-models of the source and the target, providing common 3D points for merging. We apply an analogous linking process to attempt to register iconic images that could not be placed in any 3D sub-model.

At this point, most of the models having common structure are typically merged together, and in their totality, the models cover most of the scene content present in the iconic images. In the last stage of the reconstruction algorithm, we try to make the models as complete as possible by incorporating non-iconic images from clusters of the registered iconics. This process takes advantage of feature matches between the non-iconic images and their respective iconics that were established during the earlier geometric verification stage (Section 3.2). The 2D matches between the image and its iconic determine 2D-3D correspondences between the image and the 3D model into which the iconic is registered, and ARRSAC is used to determine the camera pose. Since the model structure at this point tends to be fairly stable, we carry out a full bundle adjustment after adding every 25 images. Detailed results of our 3D reconstruction algorithm are shown in Figures 11-16, and timings are presented in Table 3.

4 Experimental Results

We have tested our system on three large landmark image datasets: the Statue of Liberty (47,238 images), Piazza San Marco in Venice (45,322 images), and the Notre Dame cathedral in Paris (11,900 images). Each of these datasets presents different challenges for our system: for example, the relative lack of texture on the Statue of Liberty poses a problem for SIFT-based matching, while the often cluttered San Marco square poses a challenge for gist clustering.

4.1 Data Collection

The datasets used for evaluation were automatically downloaded from Flickr.com using keyword searches. We randomly split each dataset into a “modeling” part, which forms the input to the system described in Section 3, and a much smaller independent “testing” part, which will be used to evaluate recognition performance in Section 4.4. Because the modeling datasets contain tens of thousands of images, we have chosen to label only a small randomly selected fraction of them. Note that the ground-truth labels are not used by the modeling algorithm itself; they are needed only to measure recall and precision for the different stages described in Sections 3.1-3.4. The smaller test sets are completely labeled. Our labeling is very basic, merely recording whether the landmark is present in the image or not, without evaluating the quality or geometric fidelity of

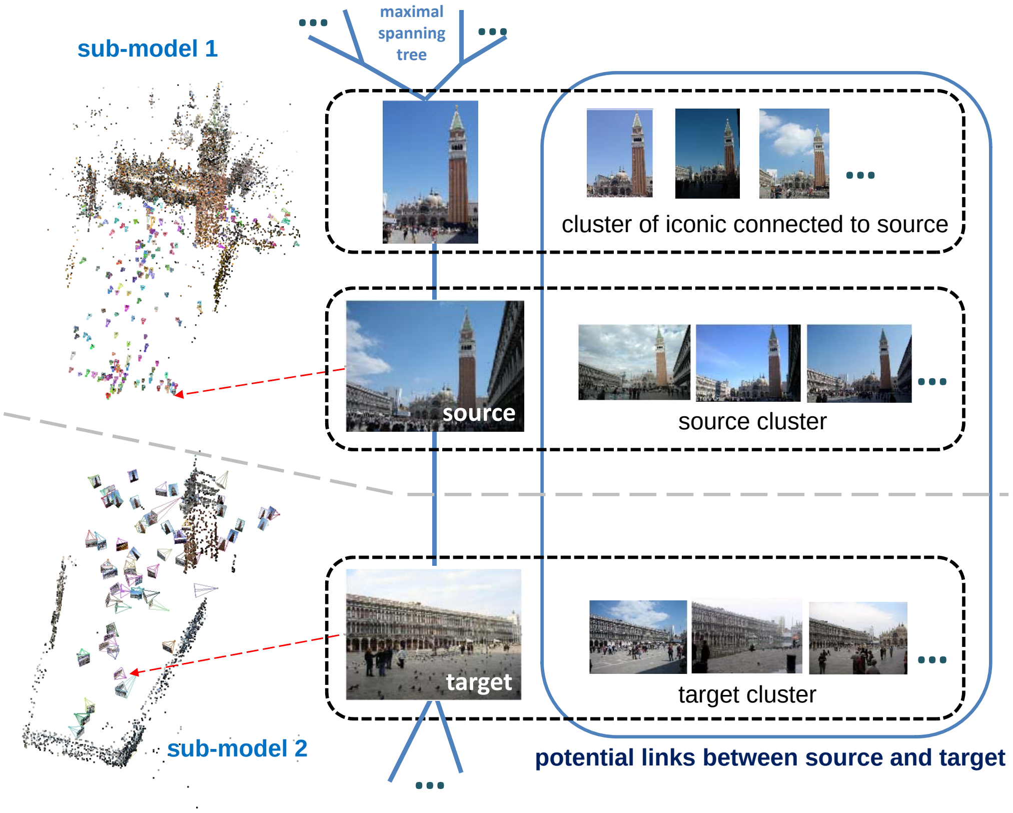


Fig. 6 3D sub-model merging: The target and source iconics are not registered in the same 3D sub-model due to a lack of common 3D points. In this case, the source iconic is registered to a sub-model encompassing mainly the front of the San Marco square, and the target is registered to a sub-model of the back of the square. To merge the two sub-models, we need to find additional non-iconic images matching both the source and the target. We search for such images in the clusters of the source and target iconic, as well as in the clusters of other iconics connected to the source. The found matching images are then used to establish common 3D points to register the two 3D sub-models.

	Modeling			Testing	
Dataset	Unlabeled	Pos.	Neg.	Pos.	Neg.
Statue of Liberty	43,845	1,383	918	631	461
San Marco	39,003	2,094	3,131	384	710
Notre Dame	9,776	562	518	546	498

Table 1 Summary statistics of the datasets used in this paper. The columns list the numbers of labeled and unlabeled images for the modeling and testing phases. Links to all the Flickr images from the datasets can be downloaded from our project website.

a given view. In particular, artistic depictions of landmarks are labeled as positive, even though they cannot be registered to our iconic representation using SfM constraints. Table 1 gives a breakdown of the numbers of labeled and unlabeled images in each of our datasets. The proportions of negative images (40% to 60%) give a good idea of the initial amount of contamination.

4.2 Modeling Results

Figure 7 shows a quantitative evaluation of the performance of each of the successive modeling stages of our approach, corresponding to stages 1-4 in Algorithm 1. Performance is measured in terms of *recall* (i.e., out of all the "positive" landmark images in the dataset, how many are incorporated into the iconic representation at the given stage) and *precision* (out of all the images currently incorporated, what proportion are "positive" landmark images). Stage 1 in Figure 7 corresponds to ranking images based on the size of their gist clusters. Precision starts off very high for the few largest clusters, but drops off rapidly for the smaller clusters. The geometric verification step improves the precision due to the removal of inconsistent clusters (Stage 2a), as does registering all images to the iconic of their gist cluster (Stage 2b). However, geometric verification decreases recall due to rejecting positive images not consistent with the iconic of their cluster. The reclustering and registration stage allows us to incorporate such images into additional iconic clusters, leading to improved recall (Stage 3). Finally, the tag filtering stage results in the removal of geometrically consistent, but semantically irrelevant clusters, leading to an additional increase in precision (Stage 4). Thus, every step of our modeling framework is well justified in terms of increasing either the precision or the recall of the iconic representation. In the end, we get over 90% precision and 47-64% recall on all datasets. The imperfect precision is due to images being registered to semantically irrelevant iconics, as discussed next, while the recall rates reflect the proportion of "unregistrable" positive images, as discussed further in Section 4.4.

Figures 8, 9 and 10 show the sets of iconic images (iconic summaries) generated by our system from each of the three datasets. For the most part, these summaries are very clean and complete, with just a few irrelevant or ambiguous iconics. For example, the summary for the Statue of Liberty dataset includes a few iconics corresponding to views of lower Manhattan, Ellis Island, and an M&M statue that parodies the Statue of Liberty. The iconic summary of San Marco contains a few views of Castillo de San Marcos in Florida, while the summary for Notre Dame contains some views of Notre Dame cathedrals in Montreal and Indiana that did not get removed by our tag-based filtering step (Section 3.4).

Table 2 shows running times for Stages 1-3 of the modeling pipeline (Stage 4, corresponding to pairwise matching of iconic images, is included in the reconstruction timings of Table 3). All the processing was done on a single commodity PC with an Intel core2 duo processor with 3GHz, 2.5GB RAM and an NVidia 280GTX graphics card. Total modeling times are about 2.5 hours for the Notre Dame dataset, and just under seven hours for the Statue of Liberty and San Marco datasets. The table also lists the number of iconics present at the end of each respective stage, along with the total number of images that the system was able to register to the iconics.

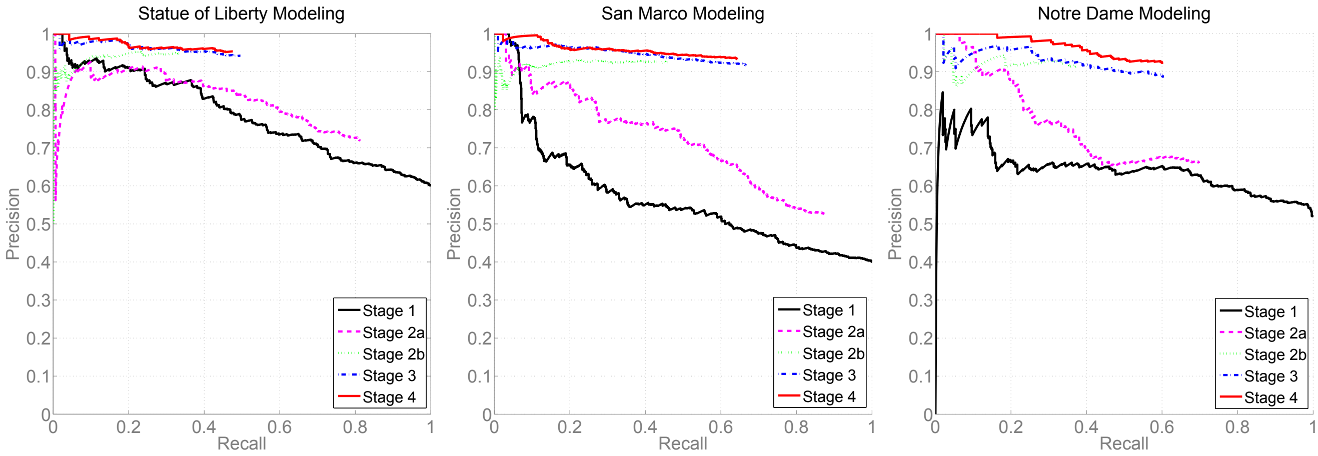
4.3 Reconstruction Results

Figure 11 shows 3D models reconstructed from the Statue of Liberty dataset. The largest model (Figure 11 (a)) incorporates front and side views of the Statue of Liberty in New York. We obtain a separate model for the back view of the Statue (Figure 11 (b)). The front and back models are not merged because of a lack of connecting intermediate views in the dataset, with the lack of texture on the statue posing an additional challenge. Figure 12 shows additional models obtained from this dataset, including the interior of the Ellis Island National Monument, and copies of the statue in Las Vegas and Tokyo. For the latter two, we obtain separate models for day and night views of the same scene. The merging of the day and night models fails because the drastic illumination change makes SIFT feature matching unreliable.

Figure 13 (a) shows the biggest reconstructed model for the San Marco dataset. Unlike the earlier version of our system [Li et al., 2008], the current implementation is able to obtain a single, complete model of the entire square. The model is merged from three initially separate sub-models: a sub-model encompassing the front of the square and the cathedral, and day and night submodels of the sides and back of the square. Given that the feature matches between the day and night components are fewer and less reliable than matches within components, the walls of the square from the merged models do not align perfectly, as illustrated in Figure 13 (b). Figure 14 shows two additional San Marco models: one of the interior of the cathedral, and another one of the side of the cathedral as seen from the courtyard of the Doges' Palace.

Figure 15 (a) shows the biggest Notre Dame model, which incorporates 1,300 views of the cathedral facade. Figure 15 (b,c) shows two additional models, for the back and side of the cathedral. These two models are not merged together because the parts of the cathedral structure visible from the two vantage points are essentially orthogonal to each other. The back and side models are also not merged with the front, which is less surprising because there are almost no photographs that can simultaneously see the facade and any other part of the cathedral structure. Finally, Figure 16 shows a couple of models constructed from different parts of the cathedral's interior.

Table 3 lists timings for the 3D reconstruction stage, along with the numbers of reconstructed images for each of the three datasets. For example, for the Statue of Liberty, our reconstruction process registers 9,934 images in about 13 hours. This running time is comparable to that of Agarwal et al. [2009], who register about 12,000 images from their Dubrovnik dataset in 16.5 hours on a single core (note that reconstruction is much harder to parallelize than pairwise image matching and geometric verification). Combining the totals from Tables 2 and 3, the overall modeling and reconstruction times are about 20 hours for Statue of Liberty, 14 hours for San Marco, and 9 hours for Notre Dame.



	Final recall	Final precision
Stage 1	1.0	0.60
Stage 2a	0.81	0.72
Stage 2b	0.33	0.95
Stage 3	0.50	0.94
Stage 4	0.47	0.95

	Final recall	Final precision
Stage 1	1.0	0.40
Stage 2a	0.87	0.53
Stage 2b	0.46	0.92
Stage 3	0.66	0.92
Stage 4	0.64	0.94

	Final recall	Final precision
Stage 1	1.0	0.52
Stage 2a	0.70	0.66
Stage 2b	0.38	0.91
Stage 3	0.60	0.89
Stage 4	0.60	0.92

Fig. 7 Top: Recall/precision curves for modeling. Bottom: summary of final (rightmost) precision and recall values for each curve. The different stages correspond to stages 1-4 in Algorithm 1. **Stage 1:** Clustering using gist and ranking each image by the size of its gist cluster (Section 3.1). **Stage 2a:** Geometric verification of iconics and ranking each image by the inlier number of its iconic (Section 3.2). The recall is lower because inconsistent clusters are rejected. **Stage 2b:** Registering each image to its iconic and ranking the image by the number of inliers of the two-view transformation to the iconic (Section 3.2). Unlike in the first two stages, images are no longer arranged by cluster, but ranked individually by this score. The recall is lower because images with not enough inliers to estimate a two-view transformation are rejected. **Stage 3**: Images discarded in the previous stages are subject to a second round of re-clustering and geometric verification (Section 3.3). This results in an increase in recall due to the discovery of additional iconic clusters. **Stage 4:** Tag information is used to discard semantically unrelated clusters (Section 3.4). Note the increase in precision due to the removal of spurious iconics.

	Feature extraction		**Stage 1**	**Stage 2**		**Stage 3**		**Totals**		
	Gist	**SIFT**	**Gist clustering**	**Geometric verification**		**Re-clustering and registration**				
Dataset	Timing hrs:min	Timing hrs:min	Timing hrs:min	Timing hrs:min	Initial iconics	Timing hrs:min	Additional iconics	Timing hrs:min	Total iconics	Images registered
Liberty	0:04	2:12	0:21	0:56	260	3:21	212	6:53	454	13,888
San Marco	0:04	3:18	0:19	0:24	270	2:47	195	6:52	417	12,253
Notre Dame	0:01	0:57	0:03	0:22	211	1:02	81	2:25	249	3,058

Table 2 Summary statistics for the modeling stage of the processing pipeline, wherein the raw image collection is reduced to a set of representative iconic images. The table lists the time taken for each stage of processing, along with the total number of iconic images present at each step. The summary column lists the final number of iconics, along with the total number of images that could be registered to these iconics. Note that the final number of iconics reflects the results following the tag-filtering step, where some iconics are rejected from each dataset. Specifically, this step rejects 20 iconics from the Statue of Liberty dataset, 47 from the San Marco dataset, and 43 from the Notre Dame dataset. The timing for this step is on the order of a few seconds, and is thus omitted from the table.

4.4 Recognition Results

This section considers the problem of landmark recognition, which we formulate as follows: given an image that was not in the initial collection, attempt to register it into our iconic representation and report success or failure. This can be useful in order to perform online updating of the 3D models or simply to answer the question of whether the image contains the landmark of interest. In our formulation, landmark recognition is conceptually very close to the task of registering leftover images in Stage 3 of our modeling pipeline (Section 3.3). In this section, we take a closer look at this task with a few new goals in mind. First, a separate evaluation of recognition performance allows us to test important implementation choices that we have not yet examined, such as the use of keypoint-based indexing

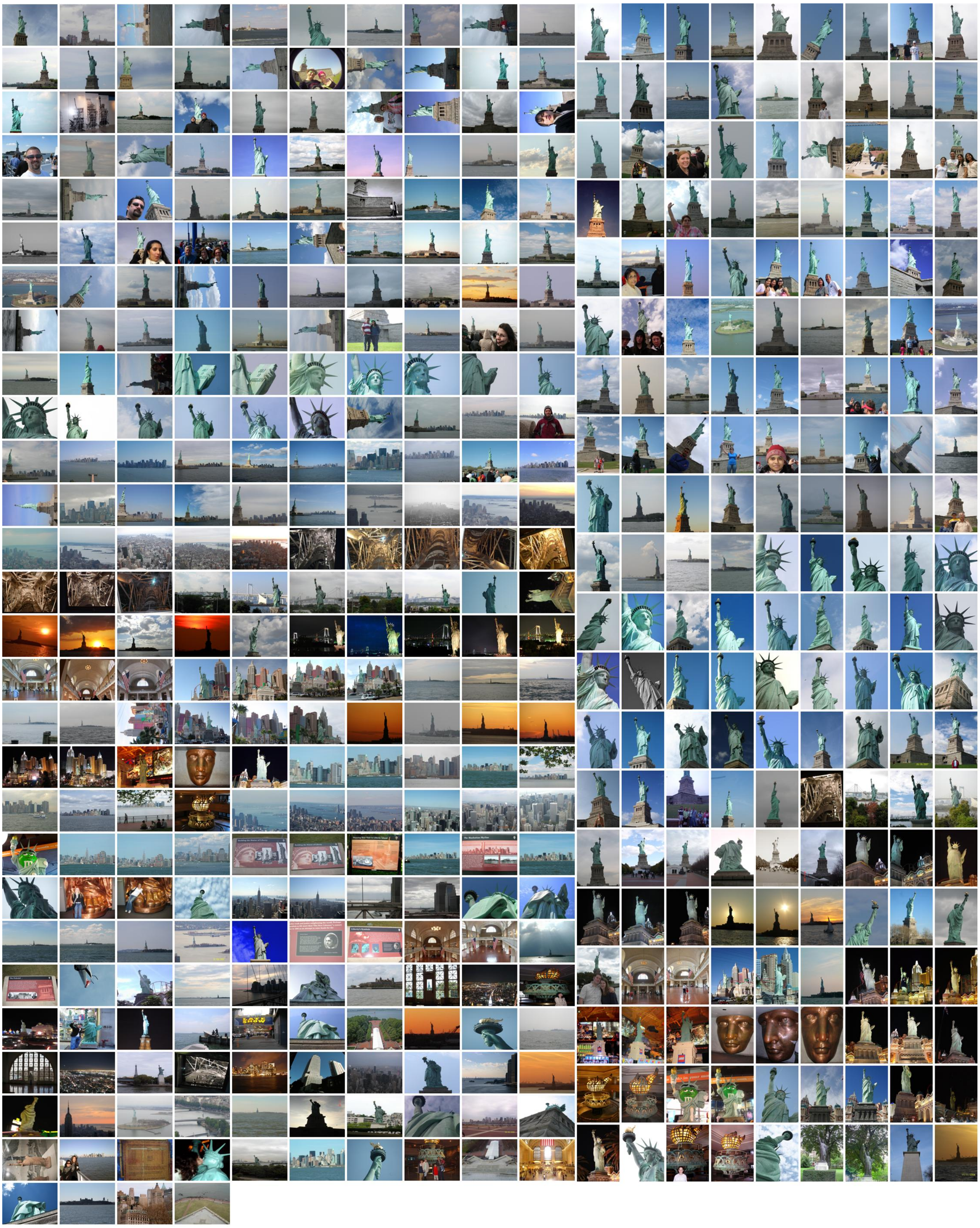


Fig. 8 Iconic summary of the Statue of Liberty dataset, containing 454 iconic images. Iconic images depict copies of the Statue in New York, Las Vegas, and Tokyo. There are also a few technically spurious, but related iconics corresponding to views of lower Manhattan, Ellis Island, and an M&M statue that parodies the Statue of Liberty.

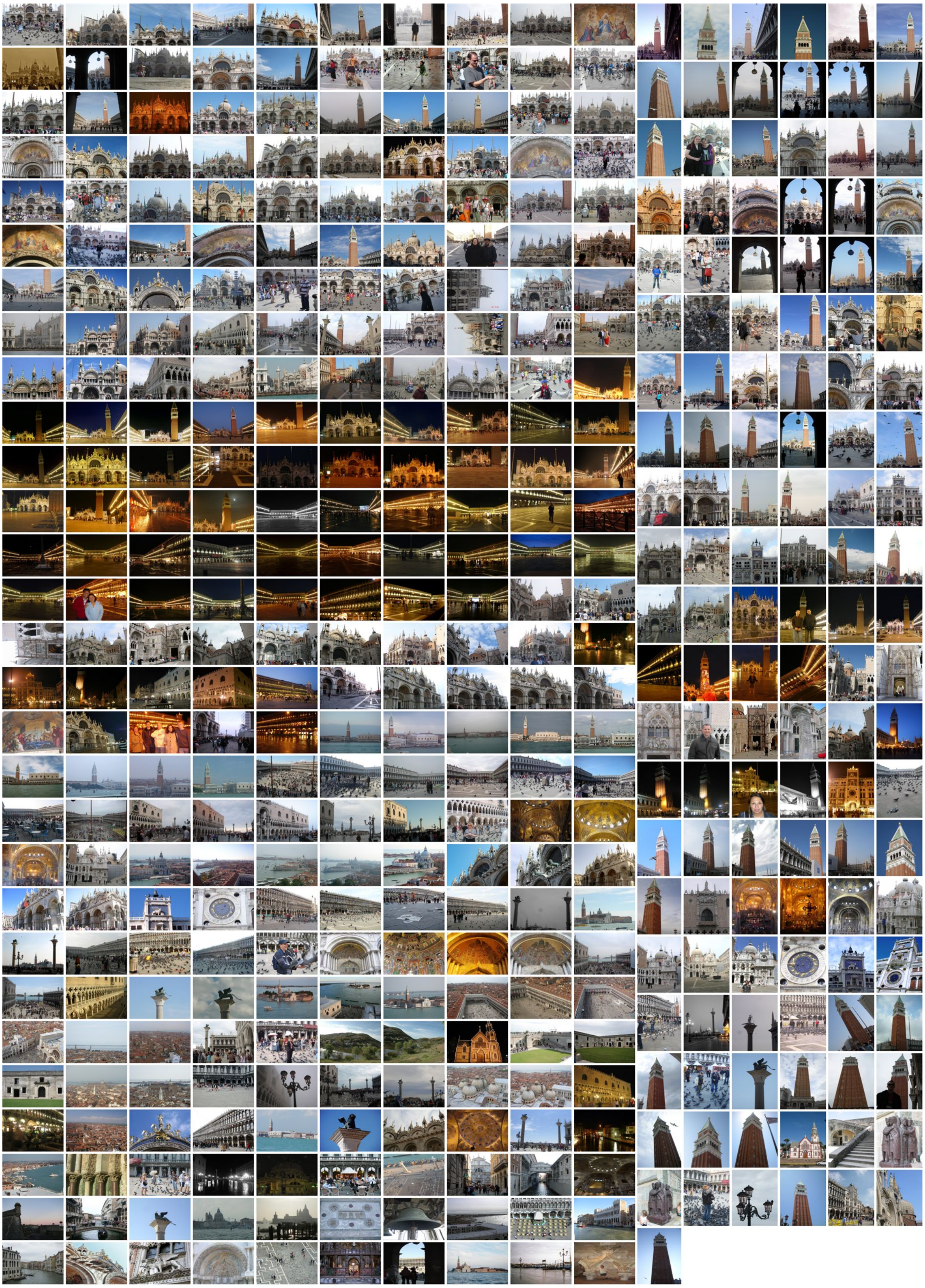


Fig. 9 Iconic summary of the San Marco dataset, containing 417 iconic images. This summary retains a few spurious iconics that did not get rejected by our tag filtering step, including views of Castillo de San Marcos in Florida (see also Figure 20-B).

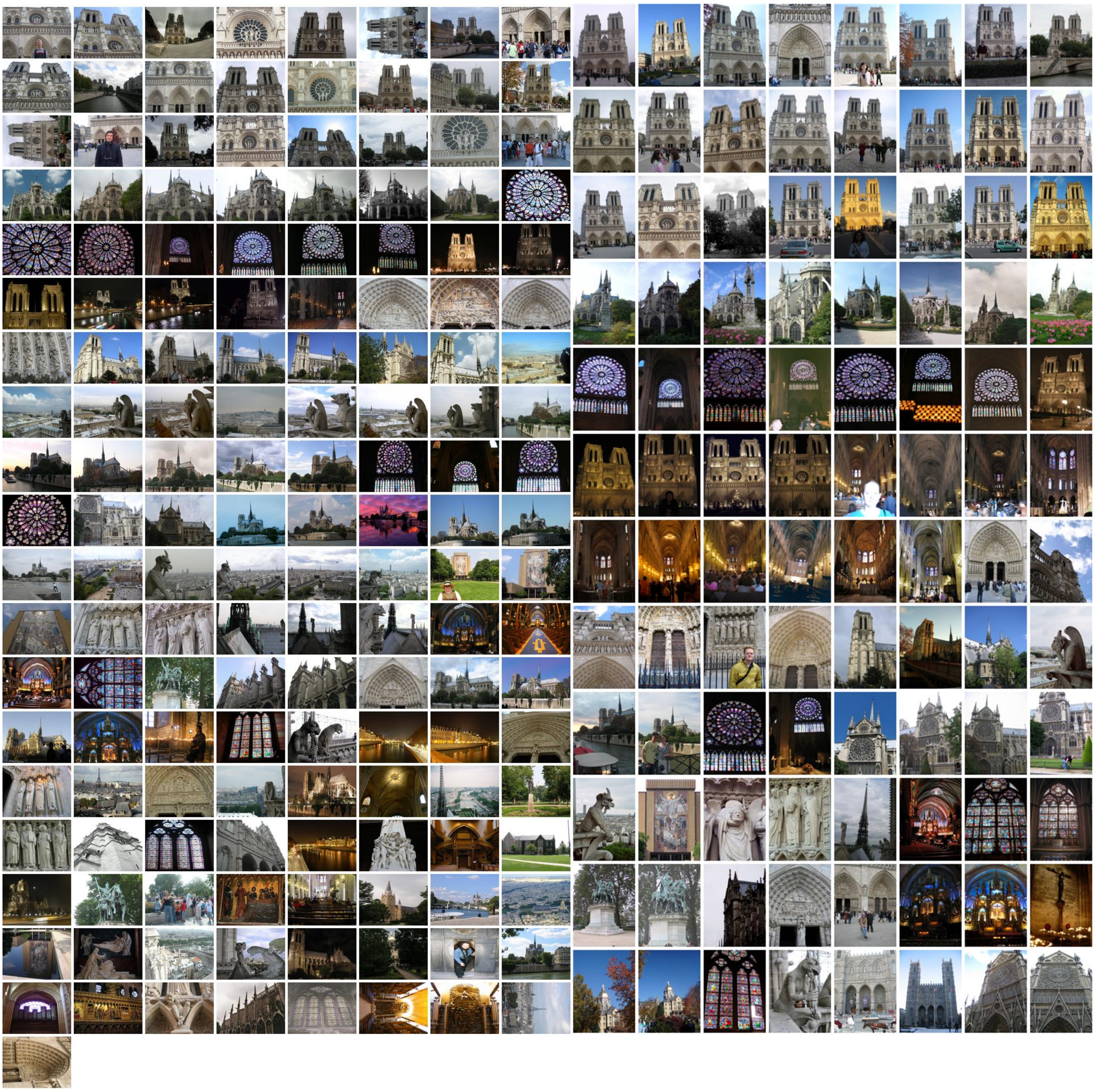
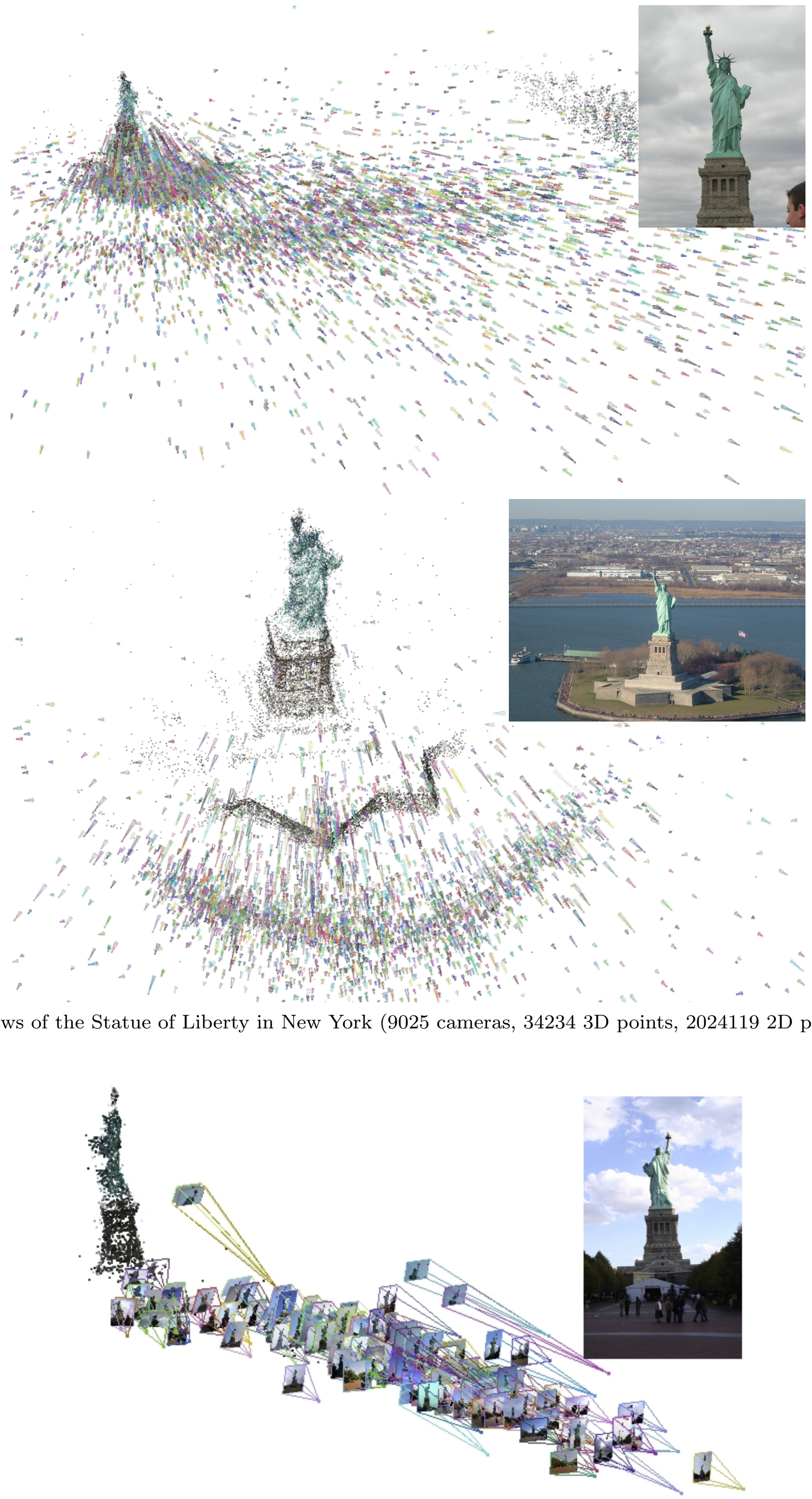


Fig. 10 Iconic summary of the Notre Dame dataset, containing 249 iconic images. There are a few unrelated iconics showing Notre Dame cathedrals in Indiana and Montreal (see also Figure 20-A).

instead of gist descriptors for image matching. Second, it allows us to quantify the extent to which our iconic summaries are representative of all landmark images marked as positive by human observers. Third, by examining individual examples of successful and unsuccessful recognition (Figures 18-20), we can get a better idea of the factors that limit the recall and precision of our system.

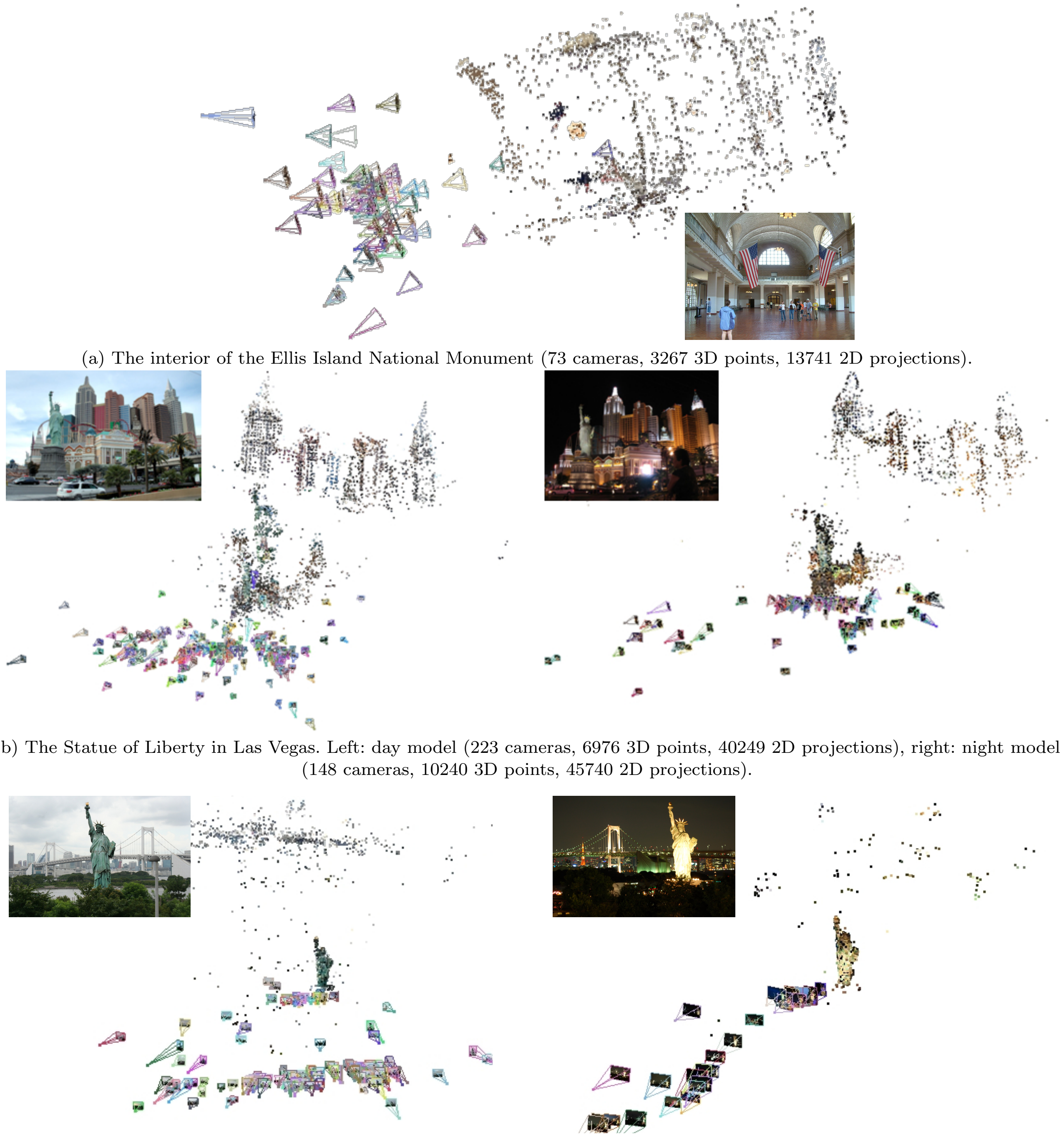
We perform recognition by treating the test image as a retrieval query against the database of iconic images belonging to the landmark of interest. Specifically, we retrieve one or more iconic images that obtain the highest matching score with the test image (according to a given retrieval scheme) and make a yes/no decision by setting a threshold on the retrieval score. We evaluate performance quantitatively by plotting a recall/precision curve of the test images ordered from



(a) Two views of the Statue of Liberty in New York (9025 cameras, 34234 3D points, 2024119 2D projections).

(b) The back side of the Statue of Liberty (171 cameras, 3131 3D points, 43557 2D projections).

Fig. 11 Selected final models of the Statue of Liberty dataset. In this and the following model figures, the inset images give representative views of the 3D structure covered by the model.



(a) The interior of the Ellis Island National Monument (73 cameras, 3267 3D points, 13741 2D projections).

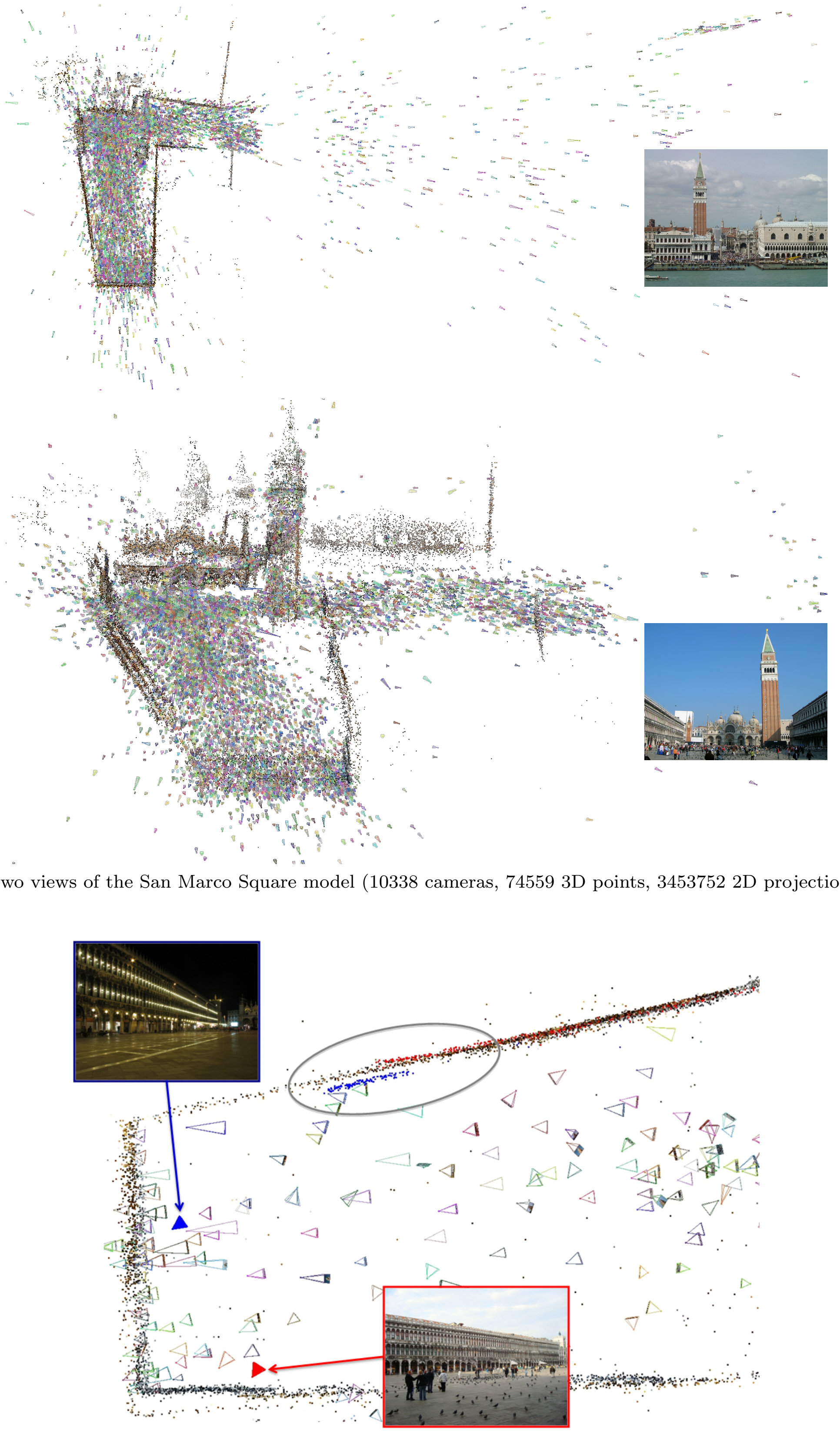
(b) The Statue of Liberty in Las Vegas. Left: day model (223 cameras, 6976 3D points, 40249 2D projections), right: night model (148 cameras, 10240 3D points, 45740 2D projections).

(c) The Statue of Liberty in Tokyo. Left: day model (146 cameras, 3221 3D points, 28223 2D projections), right: night model (27 cameras, 1658 3D points, 7949 2D projections).

Fig. 12 Additional models constructed from the Statue of Liberty dataset.

highest to lowest score. Figure 17 shows the results for several retrieval strategies, which are as follows:

1. Retrieve images using tag information. This strategy is meant mainly as a baseline to demonstrate the discriminative power of tags alone, and as such, it employs a very simple scoring scheme. Given a test image with associated tags, we retrieve the single iconic image that contains the largest fraction of these tags. It can be seen from Figure 17 that by itself, this scheme is quite unreliable.
2. Compare the test image to the iconics using either gist descriptors or a bag-of-features representation using the vocabulary tree indexing scheme [Nister and Stewenius, 2006]. In either case, the retrieval



(a) Two views of the San Marco Square model (10338 cameras, 74559 3D points, 3453752 2D projections).

(b) Overhead view of the square highlighting misalignment artifacts between day and night sub-models. The two sample views come from different models and give rise to inconsistent structure due to a lack of direct feature matches.

Fig. 13 Biggest reconstructed 3D model for the San Marco dataset.

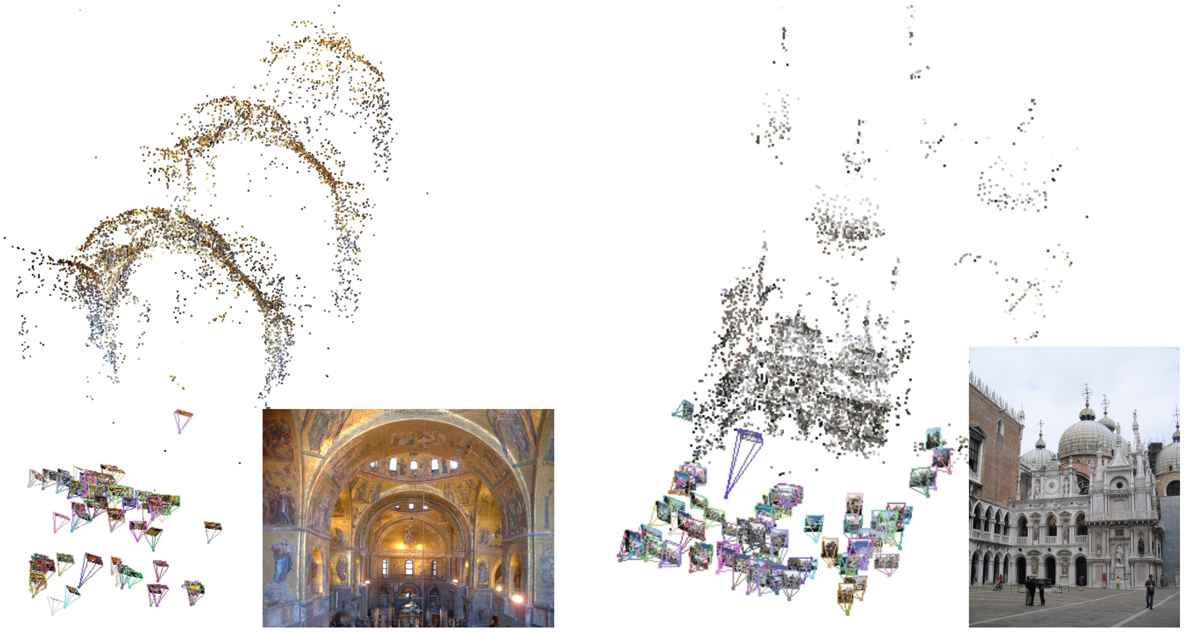


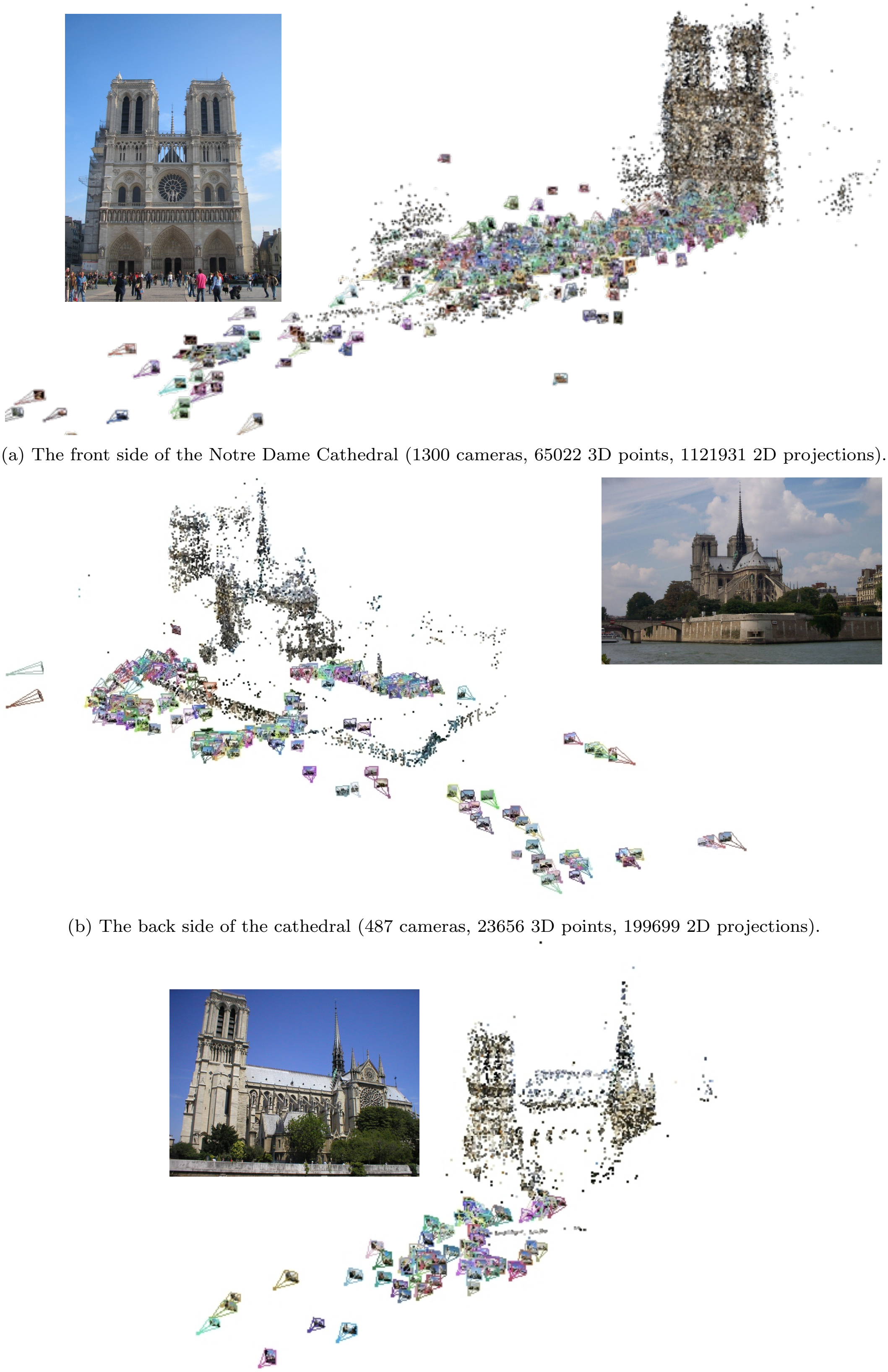
Fig. 14 Additional models reconstructed from the San Marco Square dataset. Left: the nave of the church (81 cameras, 8585 3D points, 45164 2D projections). Right: the side of the church (90 cameras, 6439 3D points, 46947 2D projections).

Dataset	**Construction of initial sub-models**		**Link discovery and merging of sub-models**		**Expansion of models with non-iconic images**		**Total**
	Timing hrs:min	Reconstructed iconics	Timing hrs:min	Reconstructed images	Timing hrs:min	Reconstructed images	Timing hrs:min
Liberty	0:32	309	2:51	434	9:34	9,934	12:57
San Marco	0:28	317	0:39	424	6:07	10,429	7:14
Notre Dame	0:10	115	0:21	186	5:46	2,371	6:17

Table 3 Summary statistics of the steps of our 3D reconstruction (refer back to Section 3.5). Note that the first stage, construction of sub-models, includes the time for finding metric reconstructions between pairs of iconics and constructing the iconic scene graph. It can be seen that significantly large datasets can be processed on the order of a few hours.

score is the similarity (or inverse distance) between the query and the single best-matching iconic. The performance of gist descriptors is shown in Figure 17 (a), and the performance of the vocabulary tree is shown in (b). For San Marco and Notre Dame, gist and vocabulary tree have roughly similar performance. However, for the Statue of Liberty, the performance of the vocabulary tree is almost disastrous – even worse than that of the tag-based baseline. This is due to the relative lack of texture in many Statue of Liberty images, which gives too few local feature matches for the vocabulary tree to work reliably. The strong performance of global features as compared to local ones validates our implementation choice of relying so extensively on gist descriptors, given that local features are often preferred in image search applications [Douze et al., 2009]. Our results seem to indicate that, provided that we query against a set of sufficiently diverse and representative views, global features will work well for retrieval.

3. Retrieve the top k candidate iconics using either gist or vocabulary tree and perform geometric verification with each candidate as described in Section 3.2. In this case, the retrieval score is the number of inliers to a two-view transformation (homography or fundamental matrix) between the test image and the best-matching iconic. It can be seen that for both kinds of descriptors, geometric verification significantly improves accuracy, as does retrieving more candidate iconics for verification (we show results for $k = 1, 5$, and 20). A high inlier score is a strong indication of the presence of the landmark, whereas a very low inlier score is inconclusive. The colored circles on the curves labeled "GIST+kNN" or "VocTree+kNN" correspond to an inlier threshold of 18, which represents the point up to which a classification can be made with reasonable confidence. Note that this is the same threshold that



(a) The front side of the Notre Dame Cathedral (1300 cameras, 65022 3D points, 1121931 2D projections).

(b) The back side of the cathedral (487 cameras, 23656 3D points, 199699 2D projections).

(c) The right side of the cathedral (94 cameras, 5414 3D points, 31171 2D projections).

Fig. 15 The three largest final models of the Notre Dame dataset.

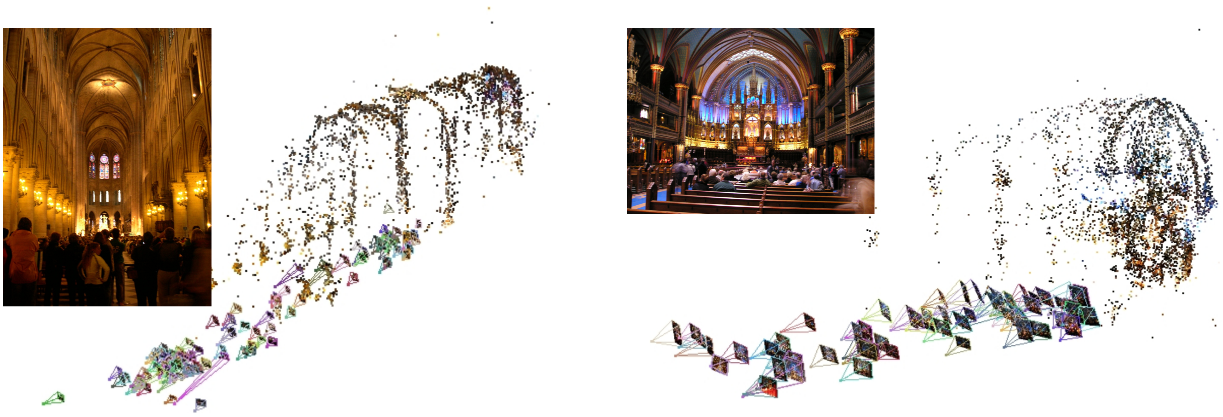


Fig. 16 Two models for different parts of the Notre Dame cathedral interior. The model on the left consists of 118 cameras, 5167 3d points and 28398 2D projections. The model on the right consists of 65 cameras, 6458 3D points and 2D 30609 projections.

is used for geometric verification of images against iconics during modeling (Section 3.2). We can see that the best recall rates reached on the test sets for this threshold are all in the 60% range, which is comparable to the recall rates of images registered into the iconic representation during modeling (Figure 7).

Figure 18 shows successful recognition examples for the three datasets. It can be seen that our system is able to return a correct match in the presence of occlusions (18-B). In addition, in the case of almost identical landmarks occurring in different locations, such as the Statue of Liberty in Tokyo (18-A), we are able match the test image to the correct instance of the landmark. Correct matches are also returned for some atypical views, such as photographs taken from inside of the Notre Dame Cathedral (18-C).

Figure 19 shows some typical false negatives where a test image containing the landmark does not gather enough inliers to any of the iconics. For instance, in the case where the landmark occupies only a very small area of the image (19-A), neither gist descriptors nor feature-based geometric verification provide strong evidence in favor of the image. Artistic depictions of the landmark (19-B) fail geometric verification, while significantly atypical views (19-C) may not have matching iconics. Based on the recall rates for modeling and testing presented in Figures 7 and 17, it appears that roughly 40% of all images labeled as positive by human observers fall into the above “unregistrable” categories.

As a consequence of the strong geometric constraints enforced in our system, false positives (or images that hurt precision) are significantly less frequent. Two example cases are shown in Figure 20, where the error arises because the set of iconic images itself contains false positives. For example, the iconics for the Notre Dame dataset include images of the Notre Dame Basilica in Montreal (20-A), and the iconics for the San Marco dataset include images of the Castillo de San Marcos in Florida (20-B).

4.5 Browsing

As a final application of the proposed iconic scene representation, we describe how to hierarchically organize landmark images for browsing.

Iconic scene graphs tend to contain clusters of iconics that have strong geometric connections among them, corresponding to dominant aspects of the landmark. We identify these components by partitioning the graph using normalized cuts (N-cuts) [Shi and Malik, 2000]. The N-cuts algorithm requires the desired number of components to be specified as input. We have found that specifying 40 to 50 components produces acceptable results for all our datasets. Note that in our earlier work [Li et al., 2008], N-cuts was also used to initialize sub-models during reconstruction. Since then, we have found that hard initial partitioning is not as conducive to model merging as the incremental scheme of Section 3.5; however, N-cuts still produce very good results for the application of browsing.

The components of the iconic scene graph form the top level of the browsing hierarchy. The second level of the hierarchy consists of iconic images grouped by component. The user can click on the representative iconic of each component (which we select to be the iconic with the largest gist cluster) to “expand” the component and see all the iconic images that belong to it. The third level consists of all remaining non-iconic images in the dataset grouped by the iconic of their gist cluster. During interactive browsing, each iconic image can be expanded to show all the images from its cluster, which will all tend to be very similar in appearance to the iconic. Figure 21 gives a snapshot of this three-level

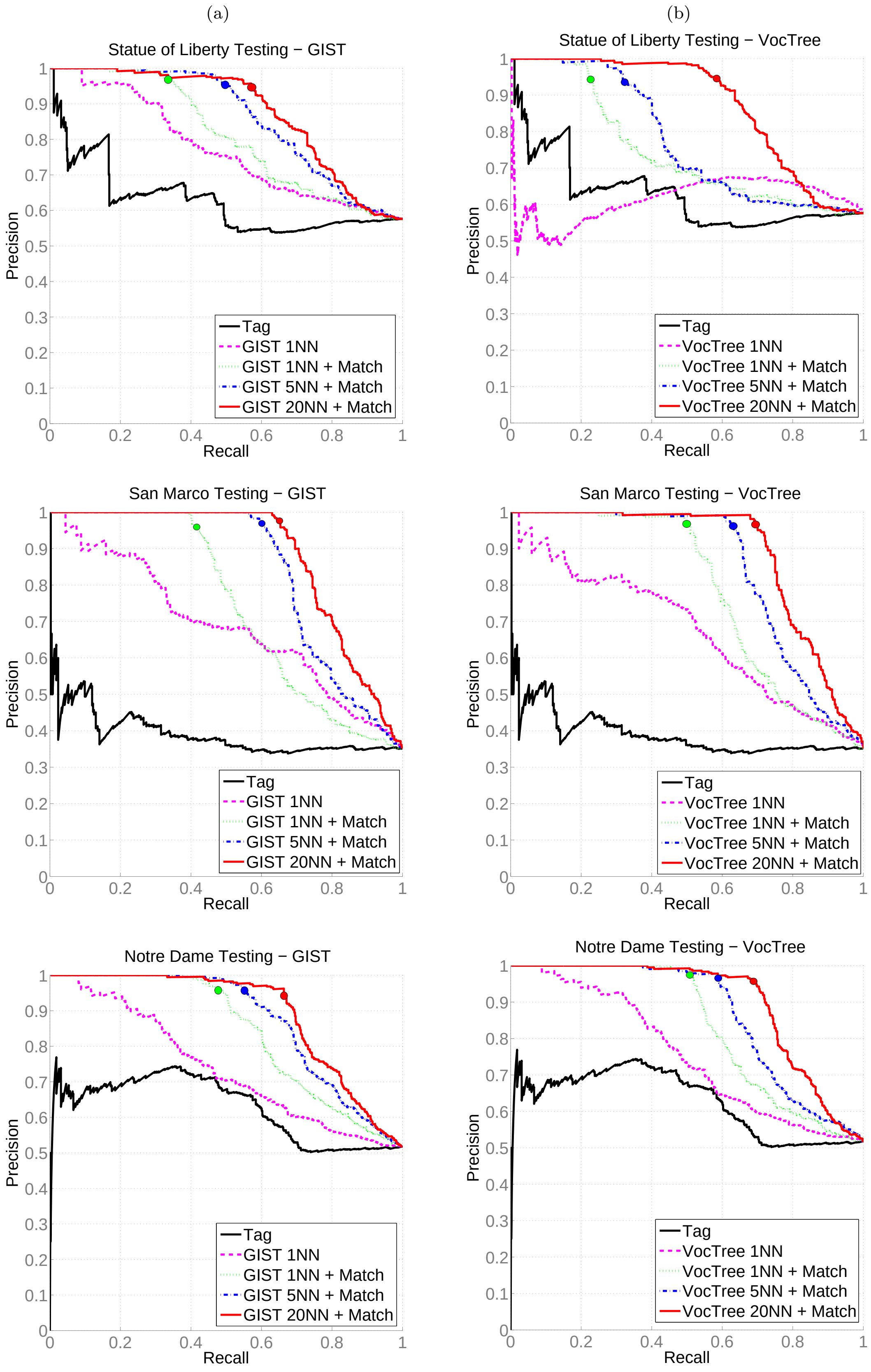


Fig. 17 Recall/precision curves for testing. The different retrieval strategies are as follows. **GIST 1NN** (resp. **VocTree 1NN**): retrieval of the single nearest iconic using the gist descriptor (resp. vocabulary tree); **GIST kNN+Match** (resp. **VocTree kNN+Match**): retrieval of k nearest exemplars using gist (resp. vocabulary tree) followed by geometric verification; **Tag:** tag-based ranking. The colored circles on the curves correspond to an inlier threshold of 18.

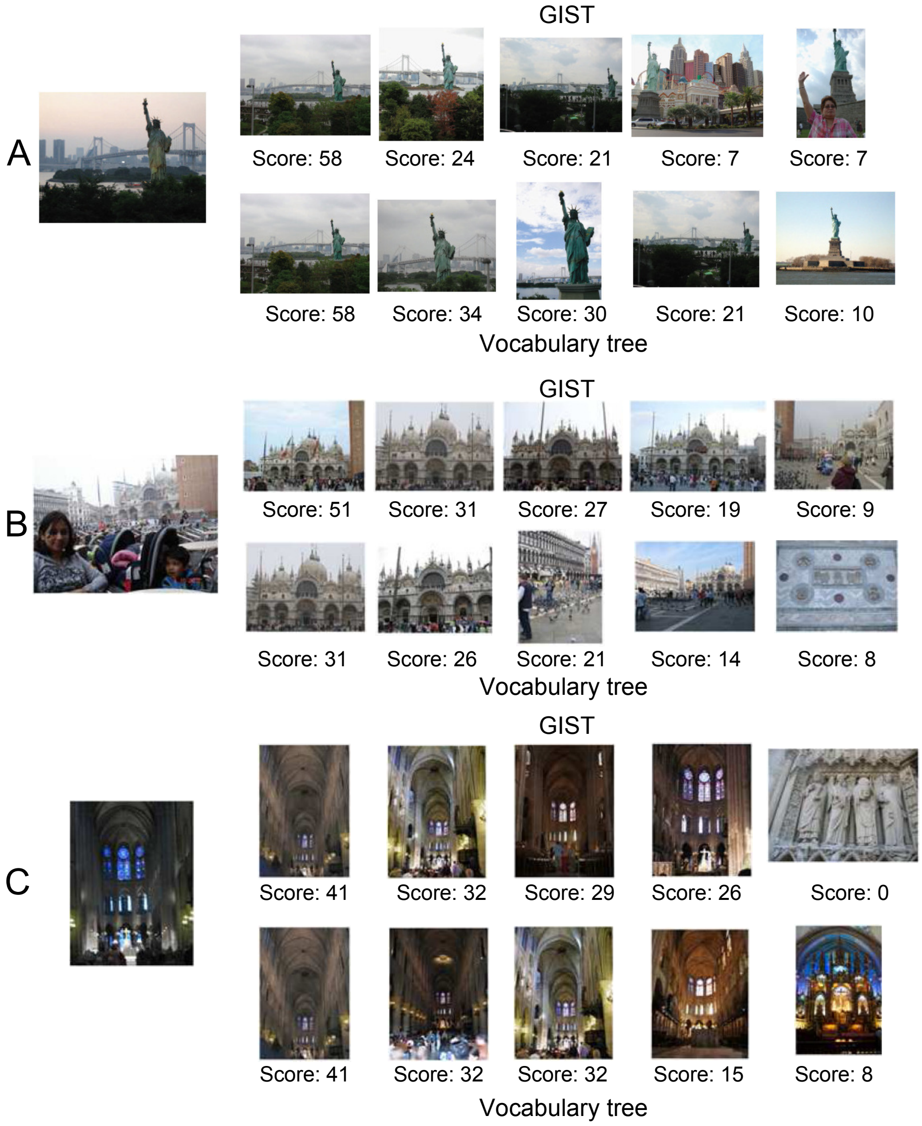


Fig. 18 An illustration of one successful case for image retrieval for each dataset. In each of the examples A-C, the query image is on the left. On the right, the top row shows the top five iconics closest to the query in terms of gist distance, re-ranked by the number of inliers to an estimated two-view transformation. As discussed in the text, inlier threshold of 18 corresponds to reliable registration. The bottom row shows analogous results for the closest five iconics according to the vocabulary tree score.

organization for the Statue of Liberty dataset. All three datasets can be browsed interactively on our website[9].

5 Conclusion

In this article, we have presented a scalable, unified solution to the problems of dataset collection, scene summarization, browsing, 3D reconstruction, and recognition for landmark image collections gathered from the Internet. By efficiently combining 2D appearance cues with 3D geometric constraints, we are able to robustly deal with the significant amount of clutter present in community-contributed photo collections. Our implemented system can process up to fifty thousand images on a single commodity PC in roughly a day. While there remains much scope for further optimization, this already represents an order of magnitude improvement over existing techniques that do not make use of cloud computing.

[9] www.cs.unc.edu/PhotoCollectionReconstruction

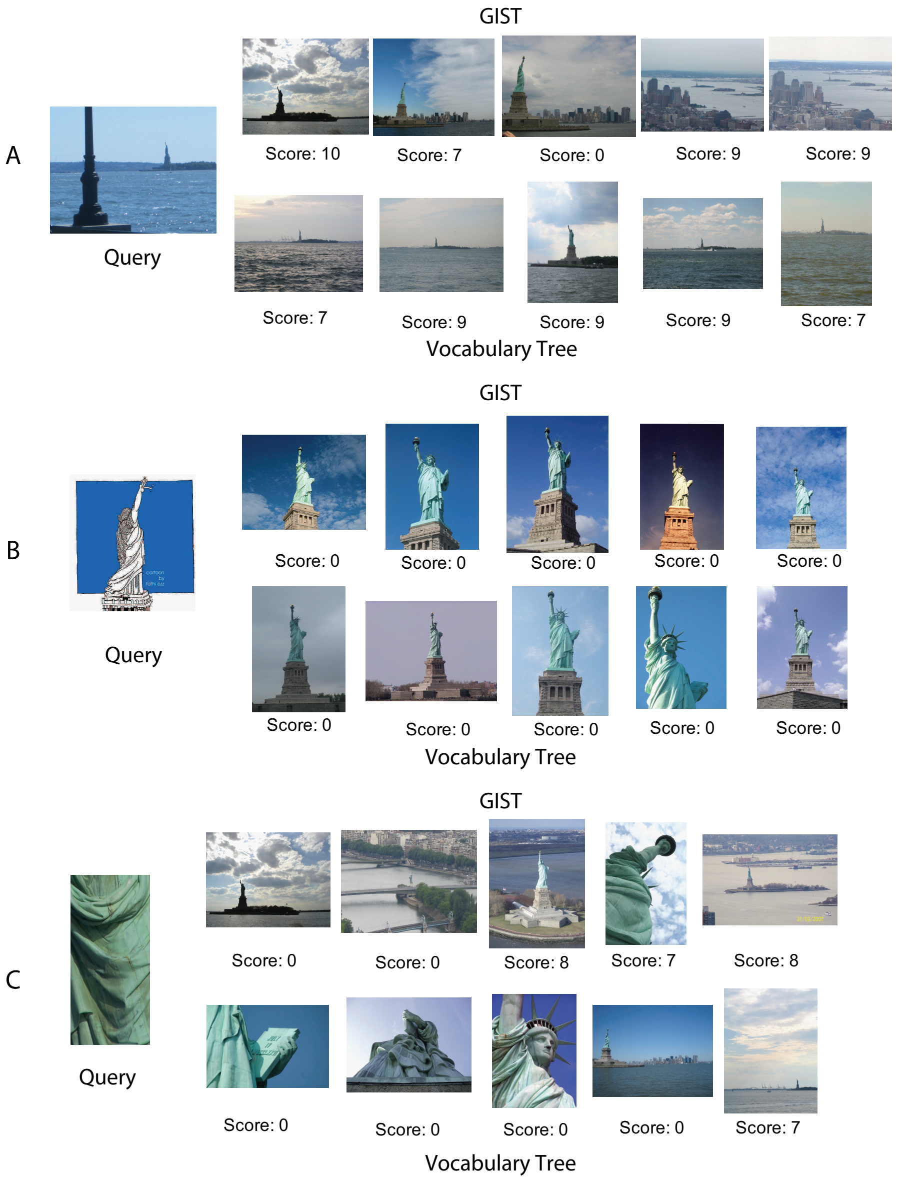


Fig. 19 Typical false negatives, or test images that could not be reliably registered to any of the iconics. The layout of the figure is the same as that of Figure 18.

A number of interesting research challenges remain open. In order to further scale our system, we need to be able to perform the initial gist clustering step in memory for much larger datasets. To this end, we are currently exploring techniques for compressing gist descriptors to short binary strings whose Hamming distances approximate Euclidean distances in the original feature space [Torralba et al., 2008].

Currently, we assume that the iconic scene graph is small enough (a few hundred iconic images), so that it can be computed by exhaustive pairwise matching of iconics and traversed exhaustively during SfM. Scaling to much larger graphs will require feature-based indexing of iconic images, as well as graph simplification techniques similar to those of Snavely et al. [2008a]. It may also necessitate the development of efficient out-of-core bundle adjustment techniques similar to [Ni et al., 2007] that use the connectivity of the iconic scene graph.

One of the limitations of our current system actually stems from its greatest source of efficiency, namely, its reliance on iconic images. By definition, iconic images correspond to “popular” viewpoints from which many people take photos of a landmark. While this helps in drastically reducing the redundancy that is present in community photo collections, it can pose difficulties when merging two 3D sub-models, where non-iconic

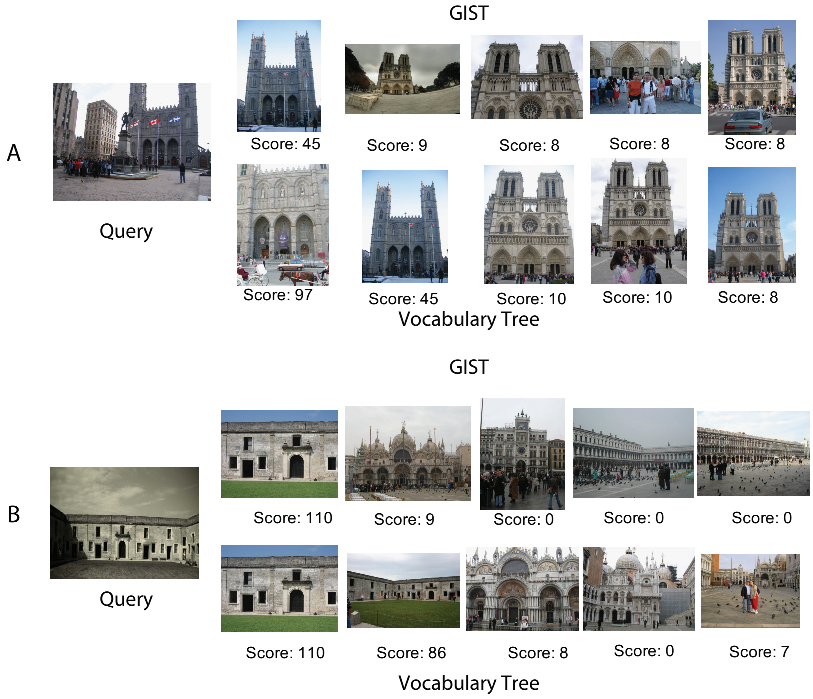


Fig. 20 An illustration of the less common case of false positives in landmark recognition. In both examples, the error arises due to the presence of an iconic image that represents a different, though similarly named landmark. The top matching iconic in A corresponds to the Notre Dame Basilica in Montreal, while the matching iconic in B is of the Castillo de San Marcos in Florida.

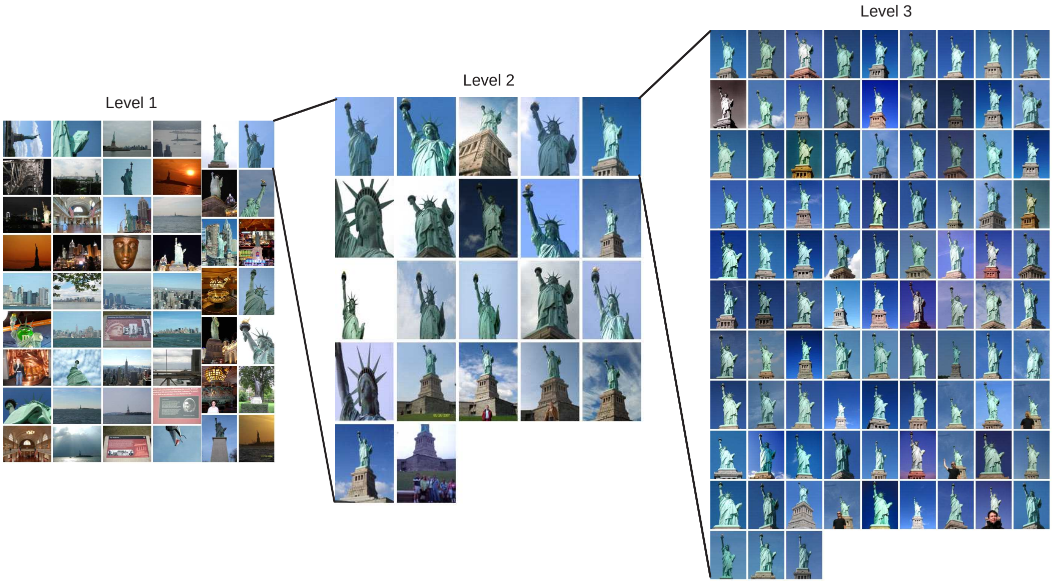


Fig. 21 Hierarchical organization of the dataset for browsing. Level 1: components of the iconic scene graph. Level 2: Each component can be expanded to show all the iconic images associated with it. Level 3: each iconic can be expanded to show the images associated with its gist cluster. Our three datasets may be browsed online at www.cs.unc.edu/PhotoCollectionReconstruction.

views may be required to provide the intermediate connections. While the link discovery method described in Section 3.5 is able to recover some missing links, it is not always successful. In the future, we plan to work on improved link discovery algorithms that use more sophisticated image retrieval techniques such as query expansion to find rare connecting views.

Illumination changes pose another major challenge for modeling and recognition. In fact, as discussed in Section 4, one of the biggest causes of failure for 3D model merging is a difference in the lighting between the two components (i.e., day vs. night). Methods for illumination modeling like those of Haber et al. [2009] may help in addressing this problem.

Acknowledgments

This research was supported in part by DARPA ASSIST program, NSF grants IIS-0916829, IIS-0845629, and CNS-0751187, and other funding from the U.S. government. Svetlana Lazebnik was supported by the Microsoft Research Faculty Fellowship. We would also like to thank our collaborators Marc Pollefeys, Xiaowei Li, Christopher Zach, and Tim Johnson.

References

Sameer Agarwal, Noah Snavely, Ian Simon, Steven M. Seitz, and Richard Szeliski. Building Rome in a day. In *ICCV*, 2009.

S. Arya, D. Mount, N. Netanyahu, R. Silverman, and A. Wu. An optimal algorithm for approximate nearest neighbor searching fixed dimensions. In *JACM*, volume 45, pages 891–923, 1998.

C. Beder and R. Steffen. Determining an initial image pair for fixing the scale of a 3d reconstruction from an image sequence. In *Proc. DAGM*, pages 657–666, 2006.

T. L. Berg and D.A. Forsyth. Automatic ranking of iconic images. Technical report, University of California, Berkeley, 2007.

Tamara L. Berg and Alexander C. Berg. Finding iconic images. In *The 2nd Internet Vision Workshop at IEEE Conference on Computer Vision and Pattern Recognition*, 2009.

T.L. Berg and D.A. Forsyth. Animals on the web. In *CVPR*, 2006.

V. Blanz, M. Tarr, and H. Bulthoff. What object attributes determine canonical views? *Perception*, 28 (5):575–600, 1999.

O. Chum, J. Philbin, J. Sivic, M. Isard, and A. Zisserman. Total recall: Automatic query expansion with a generative feature model for object retrieval. In *ICCV*, 2007.

B. Collins, J. Deng, L. Kai, and L. Fei-Fei. Towards scalable dataset construction: An active learning approach. In *ECCV*, 2008.

David J. Crandall, Lars Backstrom, Daniel Huttenlocher, and Jon Kleinberg. Mapping the world's photos. In *Proceedings of the 18th international conference on World wide web*, WWW '09, pages 761–770, New York, NY, USA, 2009. ACM.

T. Denton, M. Demirci, J. Abrahamson, A. Shokoufandeh, and S. Dickinson. Selecting canonical views for view-based 3-d object recognition. In *ICPR*, pages 273–276, 2004.

Matthijs Douze, Hervé Jégou, Harsimrat Sandhawalia, Laurent Amsaleg, and Cordelia Schmid. Evaluation of gist descriptors for web-scale image search. In *International Conference on Image and Video Retrieval*, 2009.

Rob Fergus, Pietro Perona, and Andrew Zisserman. A visual category filter for Google images. In *ECCV*, 2004.

J.-M. Frahm and M. Pollefeys. RANSAC for (quasi-) degenerate data (QDEGSAC). In *CVPR*, volume 1, pages 453–460, 2006.

Tom Haber, Christian Fuchs, Philippe Bekaert, Hans-Peter Seidel, Michael Goesele, and Hendrik P.A. Lensch. Relighting objects from image collections. In *Proceedings of CVPR*, 2009.

P. Hall and M. Owen. Simple canonical views. In *BMVC*, pages 839–848, 2005.

James Hays and Alexei A. Efros. Scene completion using millions of photographs. In *SIGGRAPH*, 2007.

Yushi Jing and Shumeet Baluja. Visualrank: Applying PageRank to large-scale image search. *PAMI*, 30: 1877–1890, 2008.

L. Kennedy, S.-F. Chang, and I. Kozintsev. To search or to label?: Predicting the performance of search-based automatic image classifiers. In *ACM Multimedia Information Retrieval Workshop (MIR 2006)*, 2006.

Lyndon Kennedy and Mor Naaman. Generating diverse and representative image search results for landmarks. In *Proceedings of the Seventeenth International World Wide Web Conference (WWW 2008)*, 2008.

Li-Jia Li, Gang Wang, and Li Fei-Fei. Optimol: automatic object picture collection via incremental model learning. In *CVPR*, 2007.

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.

Yunpeng Li, David J. Crandall, and Daniel P. Huttenlocher. Landmark classification in large-scale image collections. In *ICCV*, 2009.

D. Lowe. Distinctive image features from scale-invariant keypoints. *IJCV*, 60(2):91–110, 2004.

Kai Ni, Drew Steedly, and Frank Dellaert. Out-of-core bundle adjustment for large-scale 3d reconstruction. In *ICCV*, 2007.

D. Nistér. An efficient solution to the five-point relative pose problem. *PAMI*, 26(6):756–770, 2004.

D. Nister and H. Stewenius. Scalable recognition with a vocabulary tree. In *CVPR*, 2006.

Aude Oliva and Antonio Torralba. Modeling the shape of the scene: a holistic representation of the spatial envelope. *IJCV*, 42(3):145–175, 2001.

S. Palmer, E. Rosch, and P. Chase. Canonical perspective and the perception of objects. *Attention and Performance*, IX:135–151, 1981.

J. Philbin and A. Zisserman. Object mining using a matching graph on very large image collections. In *Proceedings of the Indian Conference on Computer Vision, Graphics and Image Processing*, 2008.

J. Philbin, O. Chum, M. Isard, J. Sivic, and A. Zisserman. Lost in quantization: Improving particular object retrieval in large scale image databases. In *CVPR*, 2008.

Till Quack, Bastian Leibe, and Luc Van Gool. World-scale mining of objects and events from community photo collections. In *Proceedings of the 2008 international conference on Content-based image and video retrieval*, CIVR '08, pages 47–56, New York, NY, USA, 2008. ACM.

Rahul Raguram and Svetlana Lazebnik. Computing iconic summaries of general visual concepts. In *Workshop on Internet Vision CVPR*, 2008.

Rahul Raguram, Jan-Michael Frahm, and Marc Pollefeys. A comparative analysis of RANSAC techniques leading to adaptive real-time random sample consensus. In *ECCV*, 2008.

Frederik Schaffalitzky and Andrew Zisserman. Multi-view matching for unordered image sets, or "how do i organize my holiday snaps?". In *ECCV '02: Proceedings of the 7th European Conference on Computer Vision-Part I*, pages 414–431, 2002.

F. Schroff, A. Criminisi, and A. Zisserman. Harvesting image databases from the web. In *ICCV*, 2007.

J. Shi and J. Malik. Normalized cuts and image segmentation. *PAMI*, 22(8):888–905, 2000.

B. Sigurbjörnsson and R. van Zwol. Flickr tag recommendation based on collective knowledge. In *WWW*, 2008.

Ian Simon, Noah Snavely, and Steven M. Seitz. Scene summarization for online image collections. In *ICCV*, 2007.

N. Snavely, S. M. Seitz, and R. Szeliski. Skeletal sets for efficient structure from motion. In *CVPR*, 2008a.

N. Snavely, S. M. Seitz, and R. Szeliski. Modeling the world from Internet photo collections. *International Journal of Computer Vision*, 80(2):189–210, November 2008b.

Noah Snavely, Steven M. Seitz, and Richard Szeliski. Photo tourism: Exploring photo collections in 3d. In *SIGGRAPH*, pages 835–846, 2006.

Susan Sontag. *On Photography.* Penguin, 1977.

A. Torralba, R. Fergus, and Y. Weiss. Small codes and large databases for recognition. In *CVPR*, 2008.

D. Weinshall, M. Werman, and Y. Gdalyahu. Canonical views, or the stability and likelihood of images of 3d objects. In *Image Understanding Workshop*, 1994.

Yan-Tao Zheng, Ming Zhao, Yang Song, H. Adam, U. Buddemeier, A. Bissacco, F. Brucher, Tat-Seng Chua, and H. Neven. Tour the world: Building a web-scale landmark recognition engine. In *CVPR*, 2009.