Scene Understanding Using Internet Photo Collections

Ian Simon

A dissertation submitted in partial fulfillment
of the requirements for the degree of

Doctor of Philosophy

University of Washington

2010

Program Authorized to Offer Degree: Computer Science and Engineering

University of Washington
Graduate School

This is to certify that I have examined this copy of a doctoral dissertation by

Ian Simon

and have found that it is complete and satisfactory in all respects,
and that any and all revisions required by the final
examining committee have been made.

Chair of the Supervisory Committee:

Steven M. Seitz

Reading Committee:

Steven M. Seitz

Richard Szeliski

Brian Curless

Date:

Signature

Date

University of Washington

Abstract

Scene Understanding Using Internet Photo Collections

Ian Simon

Chair of the Supervisory Committee:
Professor Steven M. Seitz
Department of Computer Science and Engineering

With billions of photos now online, much computer vision research has been devoted to using Internet photo collections for tasks such as visualization, learning object category models, and 3D scene reconstruction. While most of this recent work leverages the sheer quantity of online images, I present several approaches for using the distribution of images (and associated metadata) to extract structured information about 3D scenes. In essence, I use online photo collections as a proxy for human perception in aggregate, treating each photo as a statement about the world and not just a source of visual data.

I present three examples of information extraction leveraging the distribution of online photos from the photo-sharing site Flickr. First, I demonstrate the selection of canonical views of objects and scenes via a greedy image clustering algorithm. Second, I show how scenes can be decomposed into individual objects by using a cue based on the field-of-view of large numbers of images. Finally, I extract scene-scale human movement patterns from the distribution of photo sequences. Based on these projects, I demonstrate applications to scene summarization, browsing, image/object tagging, and visualization.

What objects and views do people find interesting? What is an object? How do people move around while exploring a scene? How do people frame their photos? In this thesis, I suggest a new way to answer such questions about the world, and our perception of it, using Internet photo collections.

TABLE OF CONTENTS

Page

LIST OF FIGURES

Figure Number Page

LIST OF TABLES

Table Number Page

ACKNOWLEDGMENTS

Most of all, I thank Steve Seitz for his guidance, encouragement, and patience during our time together. His superhuman ability to tolerate my moodiness and stubbornness would serve him well were he to abandon his professorship for a career in social work. I also thank Rick Szeliski and Brian Curless for reading drafts of this thesis and providing useful feedback. I thank Dan Weld and David McDonald for serving on my defense committee.

I thank Robert Pless for pointing my life in the general direction of computer vision. I thank Noah Snavely, for without his research this thesis (and many others) could not exist. I thank Sameer Agarwal, Yasu Furukawa, Pravin Bhat, and Eli Shechtman for helpful research-related discussions. I thank Dan Morris and Sumit Basu for providing me a fun (research) diversion from this work. I thank Axé Capoeira Seattle, Ivan Salaverry MMA, and Rewind for providing me fun (non-research) diversions from this work.

I thank Alex Jaffe and Kayur Patel, on whom I took out my frustrations while writing my thesis, and Nodira Khoussainova, who provided a monetary incentive to finish writing.

Finally, I thank Lindsay Michimoto, for keeping me on track during my time in graduate school, or at least making me aware of the track's existence.

DEDICATION

to Liz and my parents, whose claim that their love and support is not contingent on my completion of this thesis shall not be tested

Chapter 1

INTRODUCTION

Can we learn anything about the world from the photographs people choose to take? A person taking a photo must make several decisions: where to stand, which direction to point the camera, when to capture the photo, etc. These decisions are affected by the photographer's perception of the scene being photographed. What if we could extract some aspects of this perception from the decision to take the photo? From a single photograph, it is difficult to infer too much, since we only know about a single action made by one photographer. However, from a collection of thousands of images of a scene like the Pantheon, or even all of Rome, we can start to look for patterns of photo-taking behavior. The huge quantity of image data now online opens up new possibilities for extracting useful information by analyzing the distribution of online photos. What objects and views do people find interesting? What is an object? How do people move around while exploring a scene? Some of these questions are difficult or costly to address in other ways. In this thesis, I will describe several projects aimed at answering these questions using a combination of computer vision and statistical techniques to process large online photo collections.

In the first project, **scene summarization** (Chapter 3), I automatically derive, from thousands of images downloaded from photo-sharing site Flickr [2], a short visual summary of a scene or city that captures the key sites of interest. In an interactive setting, a user can see *canonical views* of each site of interest, and browse photos that are similar to each canonical view. When user-provided textual tag data is available, we show how it can be used to augment scene summaries by analyzing the tag statistics. At a high level, scene summarization works by computing modes in the distribution of photos that tourists take. This is done by visually matching the images in the collection, then clustering them based on visual similarity. I introduce a greedy variant of k-means to perform the clustering, as well as an orthogonality constraint that penalizes overlap between clusters. Figure 1.1 shows an example summary of Rome computed by my system.

In the second project, **scene segmentation** (Chapter 4), I extract interesting objects pictured in

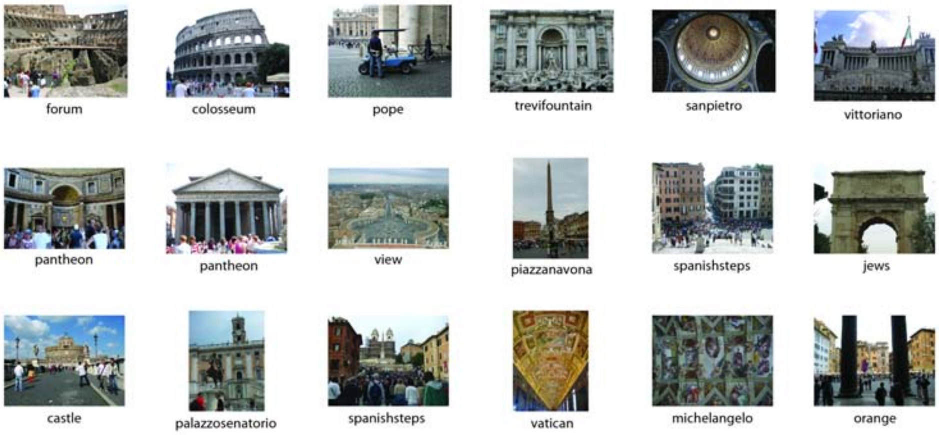


Figure 1.1: A summary of Rome computed from 20,000 Flickr images by my scene summarization algorithm.

a large photo collection by using a *field-of-view* cue — people tend to take photos of "objects" as opposed to arbitrary regions of a scene, and these objects are usually framed within the field of view of the photo. This means that points in the scene that frequently appear together in images are likely to be part of the same object. To perform the segmentation of a scene into objects, I first create a sparse 3D reconstruction of the scene using a modern structure-from-motion algorithm [118], then fit a latent topic model to the set of images and scene points. I also demonstrate the use of this analysis to display the relative importances of objects, automatically compute textual labels for these objects from noisy Flickr tags (which are attached to images, not image regions), and browse a scene using an interactive map.

In the third project, **tourist flow** (Chapter 5), I mine photos on Flickr to derive human motion patterns at the world's tourist sites. These photos are usually timestamped and the location at which each photo was taken can be computed using structure-from-motion. Even though a single photographer typically provides only a few photos, often separated by minutes, by aggregating data over hundreds of photographers it is possible to get a more detailed picture of how people explore such sites, in the form of flow fields and other representations and visualizations. Figure 1.2 shows examples of the kind of data that is produced by tourist flow, for the case of the interior of St. Peter's

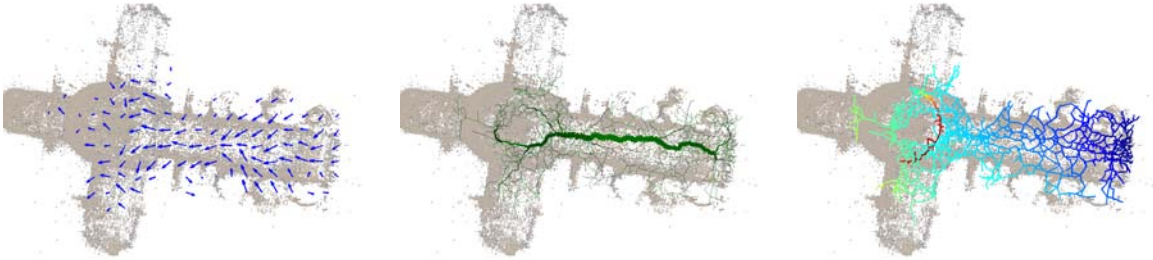


Figure 1.2: Three visualizations of tourist flow at St. Peter's Basilica. **(left)** A flow field showing estimated instantaneous velocity vectors for the population of photographers, capturing how people walk from the entrance (on the right) towards the altar (left center). **(center)** A plot showing popular paths (thick = more popular). Green edges have stronger directionality (people traverse that edge predominantly in one direction). **(right)** Rank plot, showing the order in which parts of the scene are visited, in order of violet (first), blue, green, yellow, orange, and red (last). The red ring corresponds to views from the top of the dome, which visitors tend to visit last.

Basilica in Rome.

1.1 Growth of online photo collections

The Internet has become, among other things, a repository of photos. Recent estimates put the number of online photos in the tens of billions [109], and this is only including the largest image warehouses: ImageShack, Facebook, PhotoBucket, Flickr, Multiply, and Picasa. And while the photos on some of these sites, like Facebook, are mostly private and accessible only to friends and family of the photographer, others, like Flickr, encourage photo sharing, resulting in huge numbers of photos that are viewable by anyone. Of the over 4.5 billion images currently on Flickr, around 50% are publicly accessible. For the first time in history, anyone with an Internet connection has access to billions of photos.

Taken together, these photos cover almost the entire world (Figure 1.3). A search for "Rome" on Flickr yields over 2 million photos. And since these photos only include photos labeled "Rome", and only around 30% of images on Flickr are labeled, there are likely to be many more photos of Rome that a simple text search is unable to discover. There are also 3.5 million photos of Tokyo,

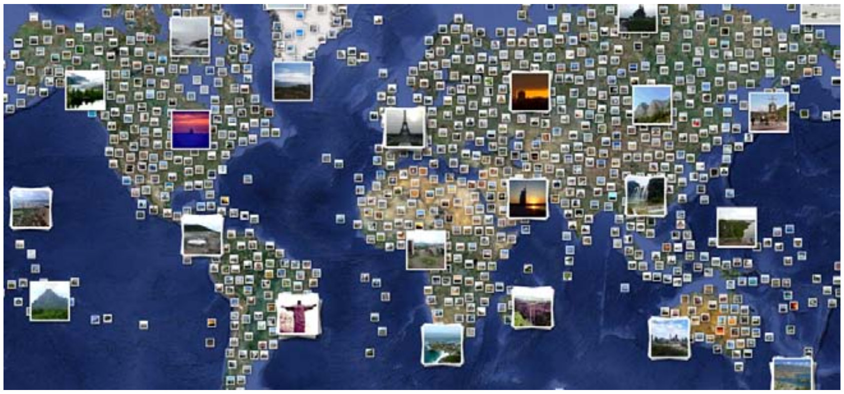


Figure 1.3: A map, from Google [4], showing the coverage of photos from its Panoramio service. Note that essentially all of the land area of Earth is covered. The actual distribution of photos is not this uniform.

3 million of Seattle, 980 thousand of Disneyland, 950 thousand of Dublin, 830 thousand of Second Life, 820 thousand of Buenos Aires, 150 thousand of various Starbucks, 120 thousand of Antarctica, 66 thousand of Brunei, 55 thousand of Easter Island, 16 thousand of Pyongyang, North Korea, 1600 of Yakutsk, Russia, and 360 of Shickshinny, Pennsylvania. If we also consider images located elsewhere on the net, these numbers get even bigger — a Google image search [3] for “Rome” yields 20 million results, ten times as many as Flickr.

In addition, the number of photos online continues to grow rapidly. Around 3 million photos are uploaded to Flickr daily (see Figure 1.4 for a graph of Flickr’s growth history). While Facebook’s photo upload rate is about 10 times faster, most of the photos are private. The types of photos uploaded to Facebook are also likely different, with more photos of people, as the site is focused on social networking rather than photo sharing. Therefore Flickr has, thus far, been the image repository of choice for the academic research community.

And though the research community and technology companies are still figuring out what to do with all this data, some impressive efforts have already taken place, enabling previously impossible experiences like browsing remote locations in 3D via photos [118], automatically estimating the location at which a photo was taken using only the photo itself [55], and automatically filling in missing image regions with data from similar images [56].

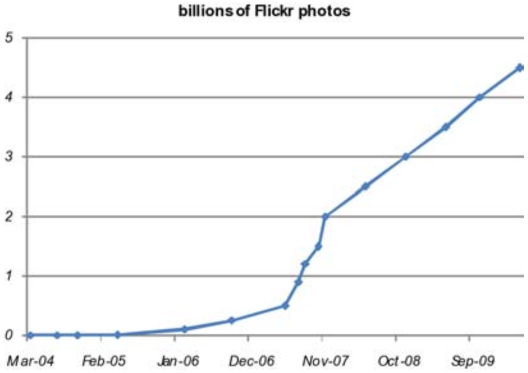
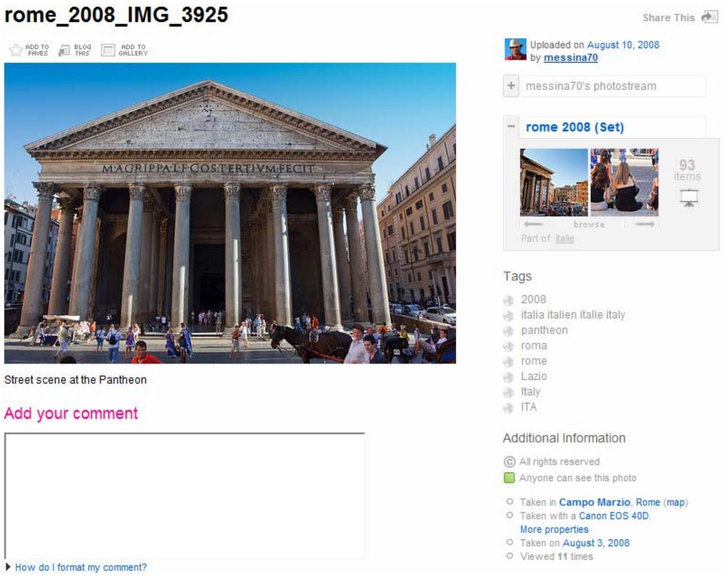


Figure 1.4: A graph showing the growth of Flickr over time. Flickr assigns each image a unique ID which roughly corresponds to the order in which images are uploaded to the site — i.e. the image with ID 2000000000 is approximately the 2 billionth image uploaded. Since each image is also timestamped, it is straightforward to query the size of Flickr at any date, or the date at which Flickr was a particular size.

The explosion of visual content available also brings with it an explosion of metadata. The information content of photos uploaded to Flickr and other photo sharing sites is greater than what is contained in the pixels alone, as users frequently *tag* their images with descriptive terms, and images often contain other associated data, including the time and approximate location at which the photo was taken. Figure 1.5 shows a photo containing much of this information. This photo is of the front façade of the Pantheon in Rome. It is also tagged with the label “pantheon”, indicating that the content of the photo has something to do with the Pantheon. It has several additional tags, including “Italy” and “2008” which, while correct, correspond to concepts much broader than what’s contained in the image pixels. The photo also has more structured metadata such as capture time (August 3, 2008) and location (Campo Marzio, Rome).

1.2 Data-driven modeling of human vision

From a computer vision perspective, the huge amount of online imagery is a wonderful new resource. Researchers now have access to orders of magnitude more data, and have used this data to create better solutions to classical problems, such as recognition, and to create entirely new tech-



rome_2008_IMG_3925

Street scene at the Pantheon

Uploaded on August 10, 2008
by messina70

messina70's photostream

rome 2008 (Set)

93 items

Part of: italie

Tags

2008
italia italien italie italy
pantheon
roma
rome
Lazio
Italy
ITA

Additional Information

All rights reserved
Anyone can see this photo

Taken in Campo Marzio, Rome (map)
Taken with a Canon EOS 40D.
More properties
Taken on August 3, 2008
Viewed 11 times

Figure 1.5: A photo of the Pantheon as displayed by Flickr. Some of the associated metadata, including tags, location, camera model, and time, are shown on the right.

niques as mentioned above. However, in addition to simply having *more* data, we now have access to a fundamentally different kind of data: real-world images from real people. Traditional imagery used for training computer vision algorithms was obtained in a lab, under controlled conditions. We now have many samples from the distribution of photos that people actually take.

Taking a step back, we can imagine a world in which photographers randomly position themselves, randomly aim, and shoot. The photos we would see in such a world are very much unlike the ones we see in reality. Random photographs are exceedingly unlikely to be framed in a way that respects the organization of the world in the human mind, and we should expect to see partial objects as well as objects viewed from angles that make perception and identification difficult [96]. In reality, we know that even untrained photographers do not take photographs in this manner, and that the act of picture-taking is a declaration of some kind about the content of the photo, and thus about the underlying world.

The field of computer vision can roughly be split into two disciplines[1]. The first involves obtain-

[1] Of course, there are many possible taxonomies of computer vision.

ing a physical representation of the world from 2D projections of light bouncing off objects. In these types of problems, there is generally a single “correct” representation, even if it is underdetermined from the input provided. Problems in this category include structure-from-motion, multiview stereo, motion capture, photometric stereo, camera pose estimation, and other such geometric problems.

The other discipline of computer vision is more closely tied to the human visual system. Problems in this discipline include object category recognition, image and video annotation, saliency estimation, and summarization. Here, an answer is correct if it agrees with human observers. This brings up an interesting issue, as human observers don’t necessarily agree with each other in all cases. One way to establish “ground truth” in these cases is to poll many observers — a correct answer should then agree with many of the observers. For the problem of segmentation, the Berkeley segmentation dataset [84] attempts to establish such ground truth.

Most of the success achieved by computer vision systems to this day are of the first type — reconstructing the physical world, matching rigid objects, etc. The second type, closer to classic problems in artificial intelligence, has proven difficult to solve. Answering questions based on visual appearances seems to require reverse engineering the human visual system, or at least greatly increasing our understanding of it. And while recent advances in these areas have been made, general computer vision systems which are able to segment and recognize a wide variety of objects under natural conditions do not currently exist.

However, with the rise of the Internet, we now have greatly increased access to human artifacts, such as hyperlinked text documents and images with various metadata, not to mention humans themselves. This suggests a new approach to the second type of vision problem, in which, directly or indirectly, we have actual humans answer the questions we want answered.

Taking this approach directly involves either paying humans (Mechanical Turk), tricking them (ESP Game) [2], or asking them to perform computer vision tasks willingly (LabelMe). The indirect approach appears promising in that we can take advantage of large quantities of already-existing data on Internet photo-sharing sites. One drawback to this approach is that we are more limited in the information we have access to: mostly just the photos themselves, with some noisy text labels and other unvalidated metadata.

[2] Whether or not this involves actual deception, performing computer vision tasks is an unintended byproduct of the game from the perspective of the player.

1.3 Using computer vision to understand image collections

How can we use large numbers of photographs to answer vision questions that seem to require human perception or understanding? Let's use segmentation as an example. At first thought, an image couldn't possibly tell us anything about its own segmentation, unless we use algorithms that attempt to reproduce human output via low-level and high-level image operations (this is what is usually done). Suppose, however, we also know something about the distribution of images that people take. Perhaps objects are more likely to appear in certain spatial configurations than others. In this case, we have a reasonable prior on the segmentations we're likely to see before looking at even a single image pixel.

On the surface, this idea bears some similarity to the idea of the *statistics of natural images*, where a distribution over naturally occurring images is used as a prior for various problems, including denoising [102], deblurring [39], and superresolution [123]. However, the particular form of this prior is extremely low-level — it's a nonlinear penalty on the gradient magnitudes in the image. A related class of low-level priors is also used by Torralba and Oliva [124] for scene and object categorization. Priors based on the distribution of photos that people actually take could be called "statistics of human photos". And instead of low-level gradients, this prior is concerned with higher-level constructs like objects. In general, if we know something about the mental model of the world humans tend to have, and we know how this model impacts the photographs they take, we can use a large set of such photographs to answer questions about the world.

Even without using computer vision techniques to process the image data itself, it's possible to extract information from Internet photos. For instance, Rattenbury et al. [101] detect and label interesting locations and events using geotagged Flickr photos, ignoring the photo content. However, this approach is dependent on unreliable user-provided metadata, such as text tags, geotags, and timestamps. In addition, this approach can tell us nothing about new unannotated photos. Using computer vision techniques to link up objects between images lets us overcome both of these problems. While in theory images could also be doctored and therefore unreliable, this would be quite time-consuming, not to mention probably useless. In addition, our computer vision matching techniques are robust enough to handle many things gracefully, such as color-adjusted images, visually identical objects, stitched panoramas, or even warped images.

1.4 Applications

What can we gain from applying unsupervised learning techniques and computer vision to large Internet image databases? One simple reason is to gain a better understanding of the world and the kind of photos people take. What is the most popular view of the Statue of Liberty? Are there even popular views, or do people take photos from more or less random viewpoints? Do people frame photos with important objects near the center, or do they obey the rule-of-thirds [40]? What paths do people take as they traverse the Roman Forum? Apart from any potential technological impact, these questions serve to broaden and deepen human knowledge.

Of course, answering such questions does enable new things. One important area in which extracting information from 3D scenes is helpful is visualization. Inferring a set of canonical views for a scene can be applied directly to visualization – for instance, an image search for “rome” can return a set of canonical photos instead of ordering them by date or the PageRank of the page on which the photo appears. These canonical views can also be used as “shortcuts” in a 3D scene browser like Photo Tourism. Other visualizations can be created which are based on objects in the scene, and this requires some process for segmenting the scene into objects. And information about human paths through a scene can be visualized directly, used to guide virtual tours in a 3D browser, or simply to sort a set of summary views into the order in which they might be seen.

Another benefit is that we are able to construct representations that are more easily searched and indexed. Around half of the photos on Flickr are untagged — these images are currently unfindable. An untagged photo of the Pantheon cannot be found by a person searching for photos of the Pantheon via the word “pantheon”, since images are not currently indexed by content. In addition, many actual tags are unrelated to the image content, or related in a way that is not useful to anyone other than the photographer. By using computer vision to process large numbers of images and applying unsupervised learning techniques to the visually matched images, we are able to identify good labels that refer to the image content.

These techniques can also aid in landmark recognition. In the simplest case, it is possible to identify a small set of canonical images on which image-based recognition can then be applied [77], but more complex techniques that recognize landmarks based on sequences of photos have also been successful [78].

1.5 Contributions

In this thesis, I demonstrate new automatic techniques for the following:

1. Choosing a set of canonical views to summarize a scene by applying clustering techniques on a large collection of images from Internet photo-sharing sites. Unlike earlier techniques that used only geotags and textual tags, we use the visual content of the photos to perform the summarization.

2. Segmenting a 3D scene (and hence images of that scene) into individual objects by using latent models from text-based document processing, in tandem with a *field-of-view* cue, which expresses a prior that photos are likely to be framed around important objects.

3. Extracting labels for scenes and objects from noisy user-provided tag data assigned to individual photos.

4. Discovering human movement patterns for traversing many different tourist sites, using photo timestamps and structure-from-motion to reconstruct precise photographer position.

In addition, I explore the idea of using photos from online collections as samples drawn from a distribution of what humans find photo-worthy, rather than as a source of low-level visual data. And while previous work in this space had been restricted to using only the metadata associated with photos, I show that using visual matching and reconstruction techniques from computer vision, applied to the actual image content, allows us to overcome many of the issues with using metadata alone.

1.6 Terminology

Throughout this thesis, I use the term *photo* interchangeably with *image*, both of which refer to an ordinary 2D image. I also use the term *view* to refer to an image, but with more emphasis on the camera parameters (location, orientation, focal length) that gave rise to the image. I often use the term *camera* to refer to an instance of such parameters, rather than the physical device itself — the cameras associated with two images may in fact be the same physical device. I define a *collection* as a set of photos, and I consider two photos to *overlap* if at least one scene point (defined below) is visible in both photos. Two photos are *connected* if they are linked by a sequence of overlapping photos. I define a *scene* in a collection as the site viewed by a set of connected photos. Note that in the limit, the entire world consists of a single scene. However, the current state of photo coverage, with dense pockets of connected photos separated by unphotographed areas, is such that connectedness is a reasonable proxy for scenehood.

A *scene point* is a 3D point in a scene visible to multiple cameras, and a *reconstruction* is a set of cameras and scene points, as well as the incidence matrix specifying which scene points are visible in which cameras. A *feature* refers to an interest point in a 2D image as computed by an algorithm such as SIFT [80], and a *feature track* is a chain of matching features across multiple images, thought to correspond to a single scene point. A *user* or *photographer* refers to the human who took some subset of the photos in the collection.

Chapter 2
RELATED WORK

My research for this thesis draws on prior work in a number of different areas. Recovering aspects of the human perceptual model of the world from photos draws on work done in the psychology community on canonical views and object grouping, as well as perception of photographs. In the computer vision and graphics communities, much recent research has been done that leverages Internet photo collections to solve new and existing problems. Previous work on summarizing large photo collections came from outside of computer vision (and mostly ignored the photo contents), but more recent work has discovered that modern visual matching techniques can be of great assistance. Other areas of relevance include large-scale image matching and 3D reconstruction, which are necessary steps in much of my work, and latent variable modeling of images and annotations. In this section, I discuss related work in the above areas in the context of my own work on Internet photo collections.

2.1 Perception of views and objects

Concepts from perception, such as objects and canonical views, help structure our understanding of the world, and also affect our interaction with it. Of particular interest to this thesis, perceptual concepts have an effect on the photos that humans take. This is useful in that it gives us some hope of recovering instances of these concepts from a large collection of photographs of the world. Doing this in a principled manner requires some study of perception itself.

2.1.1 Canonical views

A *canonical view* is a view that captures the essence of a geometric object. Such views have been studied for over twenty years in the psychology community. Defining precisely what makes a view "canonical" is still a topic of debate. In their seminal work, Palmer et al. [96] proposed four different criteria, paraphrased as follows:

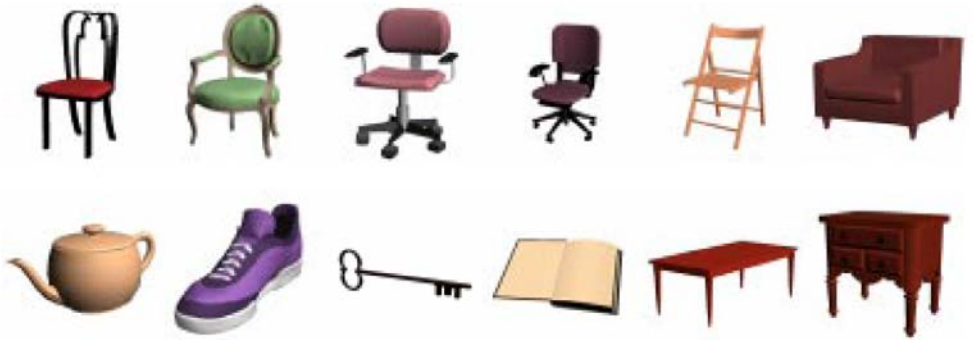


Figure 2.1: Experimentally-determined canonical views of everyday objects from Blanz et al. [13]

Criterion 1: Given a set of photos, which view do you like the best?

Criterion 2: When taking a photo, which view do you choose?

Criterion 3: From which view is the object easiest to recognize?

Criterion 4: When imagining the object, which view do you see?

In a series of experiments designed to compare these criteria, Palmer et al. found significant correlation between all four tasks, concluding that human observers choose the same types of views regardless of the task. Subsequent studies provide further support for many of these conclusions, although experiments by Blanz et al. [13] provide conflicting conclusions for the fourth criterion, finding that people tend to imagine objects in plan views, but prefer looking at off-axis views. (Example canonical views from Blanz et al. are shown in Figure 2.1.) For the purposes of this thesis, the first three tasks are most relevant, and these perceptual experiments suggest that recurring views in multi-user photo collections (criterion 2) are also likely to be visually appealing (criterion 1) and facilitate recognition (criterion 3), the latter verified in experiments by Edelman and Bülthoff [34, 17].

There is also existing work on computing canonical views in the computer vision literature. Note that the criteria of Palmer et al. are not applicable for computer vision tasks as they are defined in terms of a human observer. Hence, canonical views work in the computer vision community has sought to identify principles and algorithms based on the geometry of an object or a set of photos. For example, Freeman [43] and Weinshall et al. [128] quantify the likelihood of a view by analyzing the range of viewing conditions that produce similar views, using knowledge of object geometry. The underlying assumption is that *accidental* views are less likely to be canonical. An accidental view of an object is one for which any small perturbation results in a structurally different appearance — new faces and edges become visible. For example, imagine viewing a pencil by looking at the point head-on.

More closely related to this thesis are methods that take as input a set of photographs. In particular, Denton et al. [29] use a semidefinite programming relaxation to select canonical views from a larger set of views, choosing views which are as similar as possible to the non-canonical views while being dissimilar to each other. Hall and Owen [54] define canonical views as images with *low* likelihood, while being orthogonal to each other. They compute a 10- to 20-dimensional eigenmodel of a set of images (represented as vectors of grayscale values), then iteratively extract the least likely view that is nearly orthogonal to all previously selected views.

In this thesis, I take a fundamentally different approach to computing canonical views that is more directly related to the original principles in Palmer et al. Instead of attempting to infer such views from the geometry or from a set of uniformly sampled views, I use photo sharing websites to sample the distribution of views from which people choose to take photographs. Hence, I rely on a population of photographers to provide a likelihood distribution over camera viewpoints (as in criterion 2), and the task reduces to computing clusters and peaks of this distribution.

A second fundamental difference between this thesis and prior work on canonical views is my focus on *large scale scenes* rather than individual objects. While many objects can be represented effectively with a single canonical view, the same is not true of scenes. (Consider a church, for example, which has both an interior and an exterior, and may require several images to capture the interesting aspects.) And while most prior work on canonical views considered only a limited range of viewpoints (e.g., views on a hemisphere), images from photo sharing websites tend to have a broad sampling of positions, orientations, and focal lengths, sampling a 7-dimensional viewing space.

2.1.2 Images and objects

What is an object? This question is surprisingly difficult to answer. Early *gestalt* psychologists enumerated a set of perceptual grouping principles that are used to join a set of primitive elements (points, lines, etc.) into a single entity [129]. These principles include proximity, appearance similarity, common fate (similarity of movement), and several others. Unfortunately, the gestalt principles are extremely difficult to apply in practice for several reasons. One is that each principle is only clearly demonstrable when all others are equal — no rules specify how to join primitives when multiple principles are in conflict. Another is that "primitives" are not well-defined in the first place, as the dots and lines used in many gestalt demonstrations are themselves formed by the input from many sensory units. More recent work on perceptual grouping has established new grouping principles such as common region [91] (enclosure by a common boundary) and connectedness [95], as well as performed human experiments in order to test the power of various conflicting grouping cues [73, 93]. Still, there is no dominant end-to-end framework for perceptual grouping, though some recent work has taken steps toward modeling combinations of gestalt principles probabilistically [72]. A more thorough exposition of gestalt grouping principles appears in Palmer's textbook [92].

Image segmentation

The *segmentation* problem in computer vision is the computational analogue of the perceptual grouping problem. Usually, the primitive elements are pixels, and an image segmentation is a partition of the set of pixels into regions corresponding to objects. In some cases, the segmentation is hierarchical — pixels are grouped into more basic objects or parts, which are grouped into larger objects, and so on. An enormous amount of work has addressed the segmentation problem, such that it would be impossible to cover thoroughly in this thesis, but a few relevant milestones are worth mentioning. One early approach to segmentation was based on edge detection. Marr and Hildreth [83] defined edges as zero-crossings of the second derivative of image intensity, which also produces well-defined regions. However, this edge detector is usually outperformed by more advanced detectors [19] that do not have the property of partitioning the image into regions. Shi and Malik [115] proposed a segmentation technique known as *normalized cut* based on regions instead of edges, in which spectral clustering is performed on the pixel neighborhood graph. More general clustering algorithms such as k-means [79] and mean shift [23] are also used for image segmentation. However, none of these approaches give satisfactory results in all cases, and fully automatic image segmentation remains an unsolved problem in computer vision.

I again take a fundamentally different approach to segmentation that exploits large photo collections and relies on a simple cue — primitives that appear together in many photographs are likely to be grouped together. One way to think about this is as a new grouping cue, similar to the gestalt cues. However, instead of being a low-level perceptual cue, this cue takes advantage of the grouping cues that humans actually use, via their photographs.

Photo composition

Many conventions exist in photography regarding the framing of photographs. For instance, it is often recommended that a photo be framed according to the *rule-of-thirds* — major boundaries in the photograph, such as the horizon, should be positioned one-third of the way into the frame, and prominent objects should be positioned at the intersection of the horizontal and vertical one-third line [40]. Other conventions include having distances and sizes of objects conform to the *golden ratio* $\frac{1+\sqrt{5}}{2}$, and avoiding accidental views and alignments between objects [36].

Recent work by Palmer et al. [94], however, suggests that some of these conventions are perceptually unfounded. In a series of experiments, they found that people preferred front-facing objects to be situated in the center of the image, not at the one-third mark. For side-facing objects, people preferred the object to be facing the center, possibly as a result of the center bias and a conjecture that the "perceptual extent" of the object, which might include the space in front of it, differs from its physical extent. (This concept is also known as "lead room".) These biases were present when people evaluated a set of computer-generated test images, as well as when they positioned the objects themselves. Further experiments confirmed these biases in both the horizontal and vertical dimensions, as well as in actual photographs. The aesthetic value of the golden ratio had been tested previously on many occasions, with results that are at best inconclusive [53]. Markowsky [81] also disputes the supposed presence of the golden ratio in many works of art and architecture.

Konkle and Oliva [71] have found that in addition to preferred views (Section 2.1.1), objects also have preferred sizes relative to a fixed frame. This *canonical size* is proportional to the logarithm of the actual size of the object. In a series of experiments, they found that people preferred photos with objects at the canonical size, and also drew and imagined the object at this size, similar to the canonical view results of Palmer et al [96]. Other recent work explores different composition biases in photos. Gallagher and Chen [46] discover several properties of the distribution of photos of groups of people, and use these to aid in recognition tasks. For instance, males are likely to stand in the back row, and toward the edges of a group photo, while females cluster in the center, and this can be used as a prior for image-based gender classification. Hoiem et al. [60] exploit the likelihood of objects being present at different image locations to aid in object detection tasks, demonstrating the benefit of this type of prior for several object types in outdoor street scenes. While this doesn't use photo framing per se, as the prior depends more on real-world constraints (e.g. cars are usually on the ground), Hoiem's framework could also make use of object location distributions that arise from deliberate photo compositions.

2.2 Internet photo collections in computer vision

Recently, Internet photo collections have become popular in the computer vision and graphics communities. These photo collections are interesting for several reasons. One reason is that because of their sheer size and the real-world conditions under which the photos were taken, existing algorithms must be made to handle orders of magnitude more images under an extremely wide range of camera parameters, lighting, and so on. As part of their Photo Tourism work [118], Snavely et al. built a system for matching hundreds of unorganized real-world images using David Lowe's SIFT interest point descriptor [80], then creating a 3D reconstruction using an incremental structure-from-motion algorithm, alternating between adding new cameras to the reconstruction and running a global bundle adjustment step [10]. Further work by Snavely et al. [119] enabled the 3D reconstruction step to handle thousands of images, by first identifying and reconstructing a smaller *skeletal set* of images that approximates the coverage of the entire set while ensuring robustness of the reconstruction. At the same time, advances in large-scale image matching by Nister and Stewenius [88], Philbin et al. [97], and Chum et al. [21] enabled image search and retrieval systems (for exact object/landmark matches) capable of supporting databases containing millions of images. More recently, Agarwal et al. [8] built a matching and reconstruction pipeline capable of handling hundreds of thousands of images of entire cities. It is the existence of such systems that makes this thesis possible — previously, it would have been impossible to collect statistics on camera poses at the scale required. Other work by Goesele et al. [50] and Furukawa et al. [45] also builds on the efforts of Snavely et al. by addressing the multiview stereo problem for Internet photo collections, creating dense 3D models of objects and scenes.

Another popular use of online photo collections is as image data for computer graphics applications. Lalonde et al. [75] created a user-assisted system for inserting new objects into existing photos. Instead of manipulating the color, lighting, resolution, etc. of the new object to match the existing photo, they search a huge collection of photos to find objects that already match, greatly simplifying the user input needed. Similarly, Hays and Efros [56] fill holes in existing photos by finding visually and semantically plausible regions from a large photo collection. Kaneva et al. [68] create imaginary virtual tours by searching a large photo collection for image sequences which are plausibly (but not actually) related by pans and zooms.

A third use of Internet photo collections is as training data for various recognition tasks. Fergus et al. [37] used Google's image search [3] to learn object category models, using a spatial latent topic model to deal with the fact that an object may appear anywhere in the image, or not at all. Since then, a great deal of work [116, 110, 76, 12] has taken a similar approach, using semisupervised or unsupervised learning to extract models for recognizing object categories, faces, and so on. The manner in which I use online photo collections is conceptually different from all of the above. Previous work aimed to take advantage of the newfound quantity of online photos, or their variation. My work depends on the fact that these photos are taken by actual humans, and can be thought of as *votes* for some perceptual interpretation of the world.

2.3 Automatic image annotation

Part of this thesis involves automatically selecting text labels for images and objects. Much previous work exists on this topic, with the goal of answering two general questions:

1. Given an image, can we automatically select text labels to associate with the image? Can we go even further and associate them with particular objects in the image?

2. Given a text query, can we find a set of images containing the concept indicated by the query?

Both questions have obvious applications to image organization, browsing, and retrieval. Almost all recent approaches to answering these two questions involve representing an image and its accompanying annotation as a set of “concepts” from which both the image and its annotation are generated. This is an extension of ideas from text-only retrieval, where text documents are modeled as a bag-of-words with each word being drawn from one or more concepts [27, 58, 15].

The prototypical work on image auto-annotation is by Barnard and Forsyth, who extended hierarchical co-occurrence models for text [59] to apply to image regions and associated caption words. Given a collection of captioned images, they learn a generative model that can be applied to both auto-annotation and image search. This model consists of a hierarchical concept tree, with more general concepts (like “sky”) appearing near the root of the tree, and more specific concepts (like “tiger”) appearing near the leaves. Each image is modeled as belonging to a single cluster, which corresponds to a path from the root of the concept tree to a leaf. (Since the cluster variable is not observed, each image has some probability of belonging to each cluster.) An image is represented as a set of blobs and words, where blobs are image regions as in Blobworld [20] or normalized cuts [115], and words are associated text labels. Each blob and each word in an image is generated by one of the concept nodes on the path corresponding to the image’s cluster. The distribution of blobs and words given each concept is learned from a collection of captioned images using the EM algorithm [28]. To label a new image, inference is performed in the generative model (given the blobs only) to compute a distribution over words, and the image can be labeled with the highest probability words.

Many variations on this basic idea appear in the literature. Blei and Jordan [14] improve upon the previous model by making it more LDA-like [15]. In the previous model, the entire image is assigned (softly) to a cluster, which performs poorly when annotating images with objects that appear together infrequently. Blei and Jordan also explicitly link the regions and words by modeling each word as being generated by one of the blobs in the associated image. Barnard et al. [11] discuss many variations on their previous model, including a simple language translation model that first quantizes the image regions (first appearing in Duygulu et al. [33]).

Some work has been done on auto-annotation methods that don't use the image data at all. Naaman et al. [86] describe LOCALE, a system for image auto-annotation and retrieval using only tags and geotags, where the images are submitted and tagged by a community of users. Using geographic data instead of visual data restricts this approach to labels associated with a fixed location, but at the same time transforms the problem from a domain in which representing a concept is extremely difficult (image data) to one in which it is straightforward (2D points).

Most of the previous work on auto-annotation suffers from a few aspects that makes it less useful for annotating Flickr images of 3D scenes. First, previous methods are not designed to work in the case of noisy annotations. Second, previous methods rely on pre-segmentation using low-level features. Finally, previous methods attempt to solve the very difficult problem of object category recognition in general images, whereas I am interested in segmenting and labeling object instances from static 3D scenes, a setting in which object recognition is better understood.

My approach for combining visual and textual data is perhaps simpler than the previous approaches, as I am applying auto-annotation as a post-processing step — Flickr tags are too noisy to aid in the segmentation process. In addition, I am not attempting to learn an object category model to apply to unseen images, and handling new images of an already-annotated scene is a simpler matter of image matching and registration. Processing Flickr tags to automatically annotate images also has an advantage in that tagged photos are available in much larger quantities than verified expert-labeled images.

Chapter 3

SCENE SUMMARIZATION FOR ONLINE IMAGE COLLECTIONS

How can the visual splendor of a city like Rome be conveyed in a few images? While a good guidebook can provide a lot of information and context to plan your trip, guidebooks tend to be far less efficient at conveying what you should expect to see. This chapter addresses the problem of automatically selecting images that best summarize a scene by analyzing tourist photos for a large population of people.

If a site is visually interesting, it's almost certain that there are several photos of it on the Internet, uploaded by people who have visited that site in the past. Hence, the collection of photos on the Internet comprises an extremely rich and increasingly comprehensive visual record of the world's interesting and important sites. However, the unorganized nature of this collection makes finding relevant photos very difficult. For example, a search for "rome" on the photo-sharing site Flickr [2] returns a few million thumbnails, listed page-by-page in one of several sort orders, including date taken and an "interestingness" score whose formula has not been made public. Figure 3.1 shows one million of these images.

The objective in this work is to automatically derive, from photos downloaded from Internet sharing sites, a one page visual summary of a scene or city that captures the key sites of interest. In an interactive setting, a user can see "canonical views" or *exemplar* photos of each site of interest, and browse photos on the Internet that correspond to each canonical view. When textual user "tag" data is available, I show how it can be used to augment scene summaries by analyzing the tag statistics.

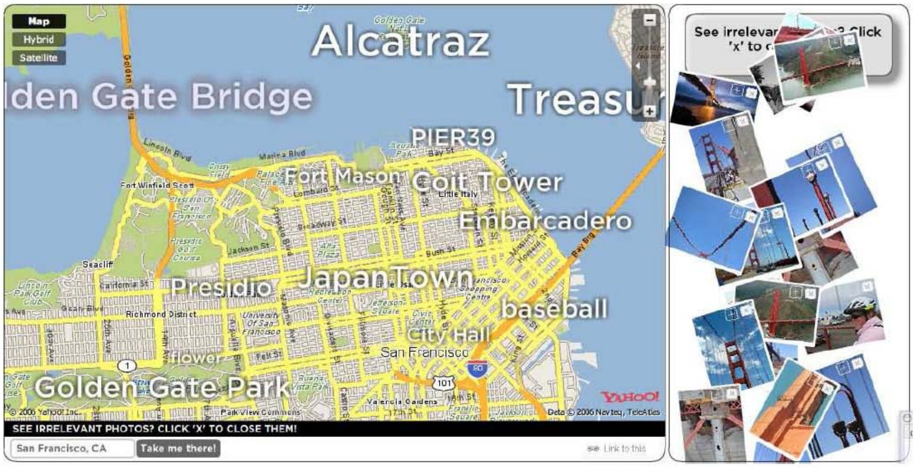
My approach to scene summarization involves three problems. The first is to partition the image set into groups of images, each corresponding to a different representative view of that scene. The second is to identify a canonical view to represent each group. The third is to compute textual tag information that best represents each view. Computing a city summary further requires identifying all of the distinct sites of interest in that city.

Figure 3.1: One million images of Rome downloaded from Flickr.

At a technical level, my approach works by applying clustering techniques to partition the image set into groups of related images, based on SIFT feature co-occurrences. The clustering is performed using a greedy method that outperforms state-of-the-art approaches for this application. Canonical views are found by using a likelihood measure, also defined based on feature co-occurrences. Descriptive textual tags are computed using probabilistic reasoning on histograms of image-tag co-occurrences. Due to the large amount of noise in user tags, obtaining high quality tags turns out to be a challenging problem, on which I show promising results.

3.1 Related work

As mentioned in Chapter 1, Internet photo collections are enormous and continue to grow. Summarization is one approach for dealing with such collections of images. Generally, a summary of an image collection can be thought of as a small, easily-digestible document (possibly interactive) that conveys the content of the entire collection. A summary may be in the form of an interactive map, a small set of representative images, or even a set of links to relevant Wikipedia pages. Summarization techniques are not necessarily able to incorporate new images dynamically, and typically do not deal explicitly with image regions (in fact, many do not even look at the pixels of the image).



SEE IRRELEVANT PHOTOS? CLICK 'X' TO CLOSE THEM!

Figure 3.2: A screenshot of the map-based World Explorer tag browser from Ahern et al. [9]

In addition, summarization has usually been applied to shared collections of images of static 3D scenes (tourist sites, for instance).

An early research project with the explicit goal of summarization was in 2006, by Jaffe et al. [62]. This was an extension of earlier work by Naaman et al. [87] intending to organize a single user's personal photo collection into discrete events at a hierarchy of locations, rather than summarization per se. Jaffe et al. summarize a multi-user geotagged photo collection from Flickr by performing hierarchical clustering on the geographic location (latitude and longitude) only, then automatically selecting a single tag to associate with each cluster. Using the tagged clusters, they create an interactive map where each tag is associated with a particular location and scale. This interactive tag map (shown in Figure 3.2) is described in more detail by Ahern et al. [9].

Other work attempts to organize image collections based only on text metadata. While ignoring the image content sounds like a poor strategy, such methods work surprisingly well in a number of instances, most notably the original Google Images algorithm [3]. Metadata-based work in the research community includes Clough et al. [22], who construct a hierarchy of images using only textual data from a captioned set of images and the concept of subsumption. A tag t_i subsumes another tag t_j if the set of images tagged with t_i is a superset of the set of images tagged with t_j. Schmitz [108] uses a similar approach but relies on Flickr tags, which are typically noisier and less informative than the captions. None of the above approaches take advantage of the visual information in the images to fill in for bad or missing metadata.

My work on summarization for this thesis uses large photo collections (with up to 250,000 photos) of 3D scenes without using geotags, instead performing clustering based on visual feature matching. Following my initial work in this area, a great deal of vision-based summarization work has taken place. After identifying tags that refer to places (like "Logan Airport") and events (like "World Cup") using the technique of Rattenbury et al. [101], Kennedy et al. [70] use vision-based clustering to create summaries from sets of images with place tags. Jing et al. [64] select a single canonical view from the results of a Google image search. Jing and Baluja [63] extend this technique to select a set of images, using a PageRank-like [90] criterion. Quack et al. [100] describe a technique for summarizing even larger collections of geotagged photos on Flickr, using both geotags and visual features. They use the geotags to avoid the more expensive step of visual feature matching for pairs of photos not lying in the same latitude-longitude grid cell. Both my work and theirs attempts to automatically annotate image clusters by selecting accurate tags from the pool of user-submitted Flickr tags, but they go further and use these tags to link image clusters to Wikipedia articles. Crandall et al. [25] similarly use geotagged photos to avoid expensive matching, and also relate summarization to the problem of visual landmark recognition. Li et al. [77] also explore the relationship between landmark summarization and recognition, and propose an iconic scene graph representation that is useful for summarization, recognition, and 3D reconstruction tasks. They also perform clustering on the GIST global image descriptors [89] to simplify the image matching task.

The Photo Tourism system of Snavely et al. [118] allows a user to browse a set of photos taken at a single site. Using a structure-from-motion algorithm to compute a sparse 3D reconstruction, along with the camera location and orientation for each photo, they provide an interactive 3D environment for exploring the scene by transitioning between photographs. While not a summary per se, this explorable reconstruction provides a more natural way to browse a photo collection than sifting through hundreds or thousands of thumbnails. One interesting component of this system is a technique by Szeliski [122] for estimating the vertical direction (which also allows auto-correction of rotated images), by essentially treating the horizontal direction in each photo as a vote for a family of ground plane orientations, taking advantage of a simple prior — photographers may adjust the camera's pitch, but they typically won't rotate it around the viewing direction.

Some existing work deals with the problem of laying out a set of images in an aesthetically pleasing way. Rother et al. [103] summarize a set of images with a "digital tapestry". They synthesize a large output image from a set of input images, stitching together salient and spatially compatible blocks from the input image set. Wang et al. [127] create a "picture collage", a 2D spatial arrangement of the images in the input set chosen to maximize the visibility of salient regions. In both of these works, the set of images to appear has already been chosen, and the visual layout is to be determined. I ignore issues of layout and focus on selecting the set of images to appear in the summary. Once selected, these images could be arranged in any desired configuration.

3.2 Existence of canonical views

For photo clustering to work effectively, the distribution of photos should be non-uniform, and well-represented by a set of modes. Is this requirement satisfied for the photos that humans take at most scenes? By examining the distribution of photos of a scene, it is possible to get a sense for how the scene is typically photographed. There are multiple possible ways to visualize this distribution: drawing the cameras on an overhead map, showing an unsorted list of thumbnails, or running our clustering algorithm and displaying the resulting exemplars. These have advantages and disadvantages — drawing the cameras on a map makes it easy to see where people are taking photos from, but less easy to see what these photos are of, while showing only thumbnails makes the relationships between them unclear.

Figure 3.3 shows a graph-based visualization of several scenes, created using GraphViz [5]. This visualization is a 2D physics-based embedding, where a set of masses (one per photo) are connected pairwise by springs (with strength corresponding to overlap between photos), and the system is allowed to converge to a stable state. In the visualization, a black dot corresponds to a photo, while overlapping photos are connected by a green edge (though there are so many edges that individual edges are mostly indistinguishable). Some thumbnails are also displayed at their embedded locations.

Figure 3.3 (a) and (b) show Flickr photo collections of the Pantheon and the Statue of Liberty, while (c) shows a lab-created data set from a multiview stereo benchmark [112]. And while the lab-

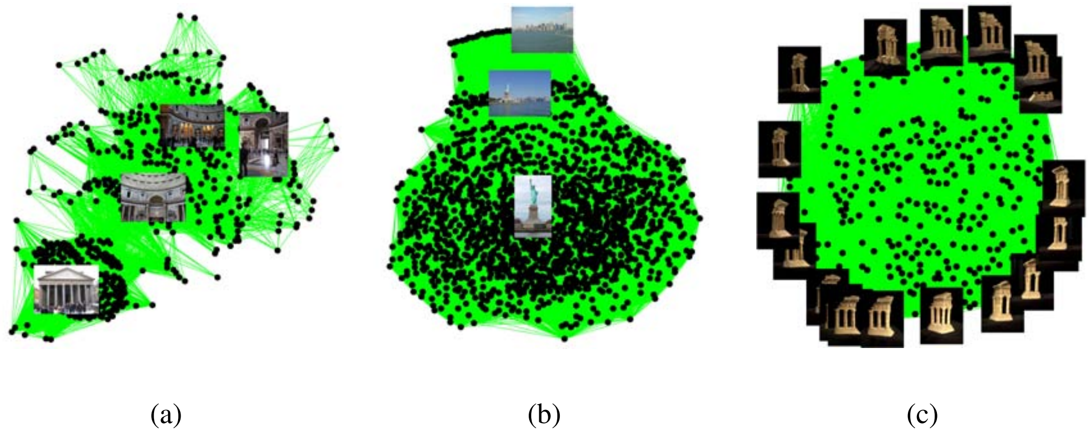


Figure 3.3: Scene graphs for (a) the Pantheon, (b) the Statue of Liberty, and (c) a structured set of images of a 3D model. The structured images were taken from uniformly-spaced camera viewpoints, which results in an image similarity graph which is more-or-less uniform. The Pantheon graph, in contrast, consists of multiple loosely-connected clusters, while the Statue of Liberty graph consists predominantly of a single large cluster.

created data appears uniformly distributed, the online photo collections show noticeable clustering into dominant views, indicating that there is potentially useful information in the distribution of photos humans take. Recovering canonical views and clusters from this distribution is therefore reasonably likely to succeed.

3.3 Problem statement

In its most basic form, a *summary* is a set of photos that represents the most interesting visual content of a scene. The purpose of a summary is to quickly give a viewer an accurate impression of what a particular scene looks like. In Section 3.6, I augment summaries to handle photo collections containing multiple scenes.

The goal, then, given a set of photos $\mathcal{V}$ of a single scene S, is to compute a summary $\mathcal{C} \subseteq \mathcal{V}$ such that most of the interesting visual content in $\mathcal{V}$ is represented in $\mathcal{C}$.

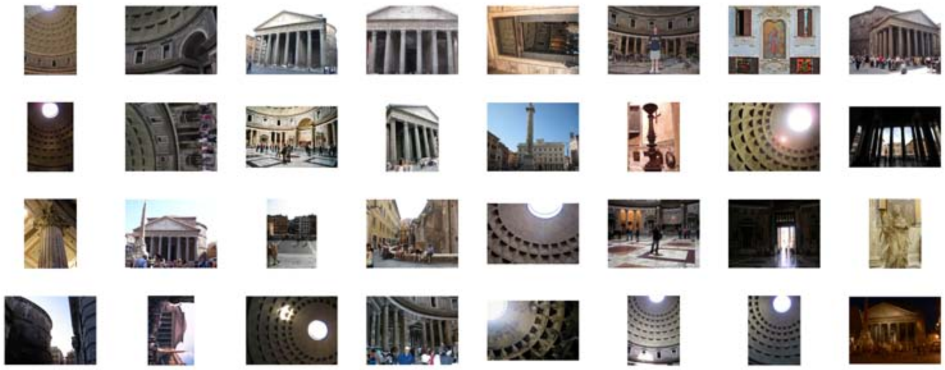


Figure 3.4: A random set of 32 images of the Pantheon. My algorithm takes an unsorted image set like this one, but containing thousands of images, and selects a set of canonical views to serve as a summary.

3.4 Scene summarization algorithm

Given a set of views $\mathcal{V}$ of scene S (see Figure 3.4), I wish to compute a summary $\mathcal{C} \subseteq \mathcal{V}$ that represents the most interesting visual content in $\mathcal{V}$. Before discussing the algorithm, I describe our representation of views and scenes:

Scene S is represented as a set of visual features $f_1, f_2, \ldots, f_{|S|}$. Each visual feature corresponds to exactly one point in the 3D environment. (However, it is possible that due to large differences in lighting or viewing direction, the same 3D point corresponds to multiple features.) A view $V \in \mathcal{V}$ is represented as the subset of S corresponding to the features which are visible in the view. Therefore, the set of photos $\mathcal{V}$ can be represented by an $|S|$-by-$|\mathcal{V}|$ Boolean matrix. This type of term-document matrix is often used as input for systems dealing with text documents [**?**], and more recently images [99]. Note that in many previous cases, each entry (i, j) in the term document matrix is a tally (how many times term/feature i appears in document/image j). In our case, since a feature corresponds to an actual 3D point, it can only be present or absent.

The feature-image incidence matrix is computed by the visual matching procedure described in Appendix A. At a high level, this procedure involves computing interest points in all images, identifying pairs of images with some visual overlap, finding a geometrically consistent set of feature matches for each such image pair, and then joining feature matches into tracks that span multiple images. Each track then corresponds to a single 3D point.

3.4.1 Selecting the summary views

There are a number of possible criteria for choosing views to include in the summary, some of which are:

likelihood - An image should be included if it is similar to many other images in the input set.

coverage - The summary should cover as many visual features of the scene as possible.

orthogonality - Two images should not both be included if they are similar to each other.

I focus mainly on likelihood, as we are interested in harnessing the consensus of users of photo sharing sites for selecting canonical views. However, I also explore the use of the other criteria.

Image likelihood

The most popular criteria in previous work on canonical views are likelihood ([43], [128]) and orthogonality ([29], [54]). However, in previous work, the likelihood of an image referred to the range of viewing parameters that produces similar views. We, on the other hand, have a set of images distributed according to the viewpoint preferences of human photographers. Our likelihoods are measured on this distribution and not inferred solely from geometry (or using a uniform distribution over viewing directions). We define the *similarity* between two views as:

$$\mathrm{sim}(V_i, V_j) = \frac{|V_i \cap V_j|}{\sqrt{|V_i||V_j|}} \tag{3.1}$$

Equation (3.1) measures the cosine of the angle between the normalized feature incidence vectors for the two images. If both views have the same number of features, this is simply the fraction of features that are shared. If the two views do not share any features, the similarity is zero. It is also possible to include features with no matches (which are ordinarily discarded in the creation of the feature-image incidence matrix). This has the effect of reducing the similarity between pairs of images in which one or both images have many unmatched features, which are sometimes caused by foreground clutter. The effect tends to be far too strong, however, and penalizes images with textured regions, so I currently ignore these features.

Another modification is to use tf-idf weighting [106]. In this scheme, each feature occurrence is weighted by its "term frequency" (number of times it appears in an image, which is always 1 in this case) and "inverse document frequency" (logarithm of the total number of images divided by the number of images containing the feature) before normalization. Using inverse document frequency increases the importance of features that appear infrequently. I use tf-idf weighting to compute most of the results in this chapter, but find that it has little to no effect on the output summaries.

In a slight abuse of notation, we will use V to refer to the set of features in a view as well as the normalized Boolean feature incidence vector or tf-idf vector. So:

$$\mathrm{sim}(V_i, V_j) \quad = \quad V_i \cdot V_j \tag{3.2}$$

A simple definition of *likelihood* is then:

$$\mathrm{lik}(V) = \sum_{V_i \in \mathcal{V}} (V_i \cdot V) \tag{3.3}$$

This definiton of likelihood is closely related to the log likelihood of the set of images $\mathcal{V}$ being drawn from a von Mises-Fisher distribution (a spherical analogue of a Gaussian) with the normalized feature incidence vector for V as mean parameter μ:

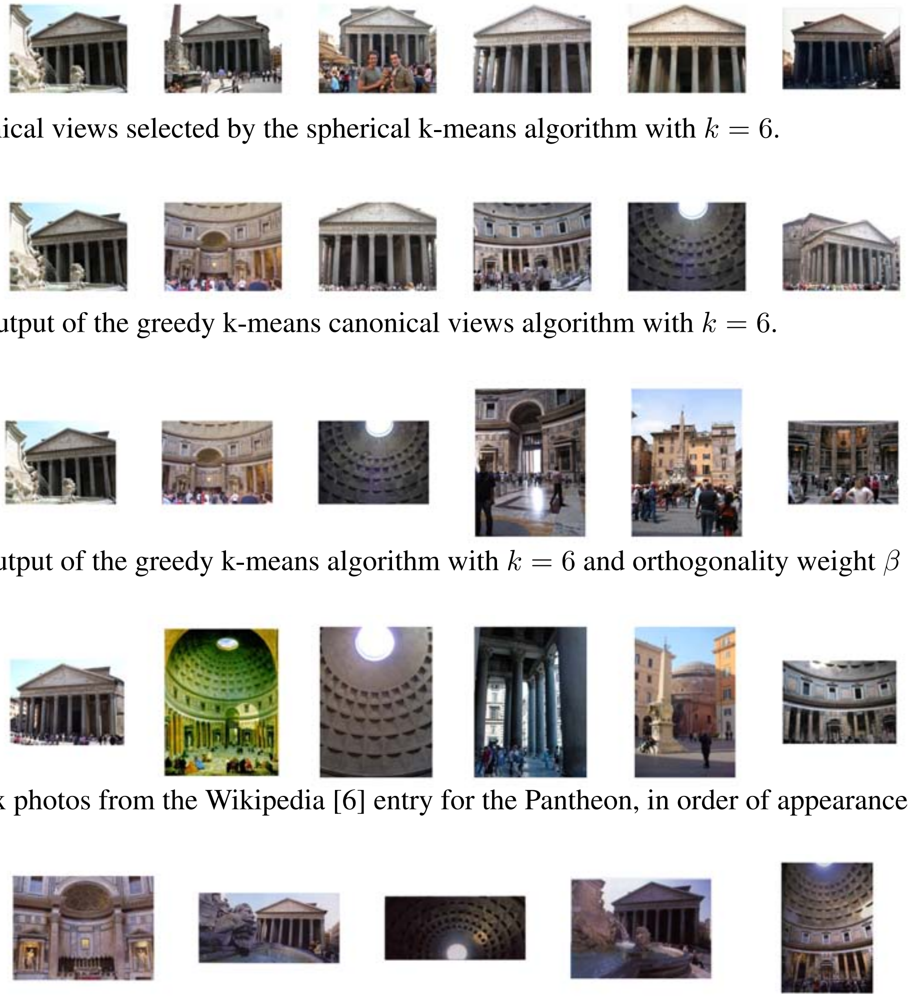
$$p(X|\mu, h) \quad = \quad \prod_{x \in X} f(h) e^{h(x \cdot \mu)} \tag{3.4}$$

$$\log p(X|\mu, h) \quad = \quad h \sum_{x \in X} (x \cdot \mu) + \log f(h) \tag{3.5}$$

where h is the nonegative concentration parameter and $f(h)$ is the normalizing constant chosen so that $p(x|\mu, h)$ integrates to one. Note that h only specifies a linear transformation on the sum of similarities, and can often be ignored.

Clustering objective for canonical views

Because our goal is to represent the target image set $\mathcal{V}$, we include a quality term for each view $V_i \in \mathcal{V}$ expressing the similarity between V_i and its closest canonical view $C_{c(i)}$ in $\mathcal{C}$, where c contains the mapping of views to canonical views. Also, we want to penalize solutions with too



(a) Canonical views selected by the spherical k-means algorithm with $k = 6$.

(b) The output of the greedy k-means canonical views algorithm with $k = 6$.

(c) The output of the greedy k-means algorithm with $k = 6$ and orthogonality weight $\beta = 100$.

(d) All six photos from the Wikipedia [6] entry for the Pantheon, in order of appearance.

(e) Left to right: one Pantheon photo from the Berlitz [111] and Lonely Planet [61] guidebooks, and three from Fodor's [42]. These are the only images of the Pantheon in the three guidebooks.

Figure 3.5: Comparison of several summaries of the Pantheon. Summary (a) illustrates the failure of the spherical k-means algorithm to find meaningful clusters. Summaries (b) and (c) are typical results, and also demonstrate the effect of the explicit orthogonality constraint. Hand-created summaries (d) and (e) are included for comparison. Note that my computed summary views are quite similar to those in Wikipedia and the guidebooks. When we produce larger summaries, we often select interesting views which are left out of Wikipedia and typical guidebooks.

many canonical views, as our summaries are meant to be readable quickly, so we include a cost term α for each canonical view. Our algorithm attempts to maximize the following quality function:

$$Q(\mathcal{C}) \quad = \quad \sum_{V_i \in \mathcal{V}} \left(V_i \cdot C_{c(i)}\right) - \alpha|\mathcal{C}| \tag{3.6}$$

The summation term is closely related to the log likelihood of the set of views $\mathcal{V}$ being drawn from a mixture of von Mises-Fisher distributions with equal mixture weights and common concentration parameter h. The $-\alpha|\mathcal{C}|$ term can be thought of as enforcing a geometric prior on the number of canonical views. Alternatively, we can fix the number of canonical views and maximize only the similarity portion of the objective.

This objective function implicitly encourages the canonical views to be orthogonal, as each view V need only be explained by one canonical view. In cases where orthogonality is more important, I add an extra term to the objective function:

$$Q(\mathcal{C}) = \sum_{V_i \in \mathcal{V}} \left(V_i \cdot C_{c(i)}\right) - \alpha|\mathcal{C}| - \beta \sum_{C_i \in \mathcal{C}} \sum_{C_{j>i} \in \mathcal{C}} \left(C_i \cdot C_j\right)$$

This explicitly penalizes pairs of canonical views for being too similar (see Figure 3.5(c)).

Without the $-\alpha|\mathcal{C}|$ term, the function could be optimized by a simple modification of the spherical k-means algorithm ([30]) in which the means are restricted to views in the data set. However, even in the simplified case where $|\mathcal{C}|$ is known, the spherical k-means algorithm performs poorly when the dimension is large and many pairs of views have zero overlap. To illustrate why this is so, imagine a scene with two objects A and B. Now suppose 60% of the views contain only object A, 30% of the views contain only object B, and 10% of the views contain both objects. If we initialize two canonical views randomly, the most likely configuration is two views of object A only, from which the desired solution is unreachable. This situation is not far-fetched from reality, as Schlieder and Matyas [107] have shown that online tourist photos appear to follow a power-law distribution, with a characteristic long tail. In general, the spherical k-means algorithm is extremely sensitive to the initial configuration (see Figure 3.5(a)). We avoid this problem by using the following greedy algorithm, beginning with $\mathcal{C} = \emptyset$:

1. For each view $V \in \mathcal{V} \setminus \mathcal{C}$, compute
   $Q_V = Q(\mathcal{C} \cup \{V\}) - Q(\mathcal{C})$.

2. Find the view V^* for which Q_{V^*} is maximal.

3. If $Q_{V^*} > 0$, add V^* to $\mathcal{C}$ and repeat from step 1. Otherwise, stop.

At each iteration, we choose the view that will cause the largest increase in the quality function and add it to the set of canonical views, as long as this increase is at least α. If not, we stop. Cornuejols et al. [24] proved that this greedy algorithm always yields a solution that has quality at least $\frac{e-1}{e}$ times the optimal solution, where e is the base of the natural logarithm. I find that the greedy algorithm (Figure 3.5b and 3.5c)) also performs much better in practice than the standard spherical k-means algorithm (Figure 3.5a), which has an arbitrarily bad approximation ratio. This algorithm also frees us from having to choose the number of canonical views in advance, though we do need to specify α. When using explicit orthogonality penalties, the proof of approximation bound no longer applies, though we find the algorithm still works well in practice. We have also experimented with running the standard spherical k-means algorithm initialized with the means chosen by the greedy algorithm. For most of our data sets, this changes very few of the means, and changes them to nearly identical views.

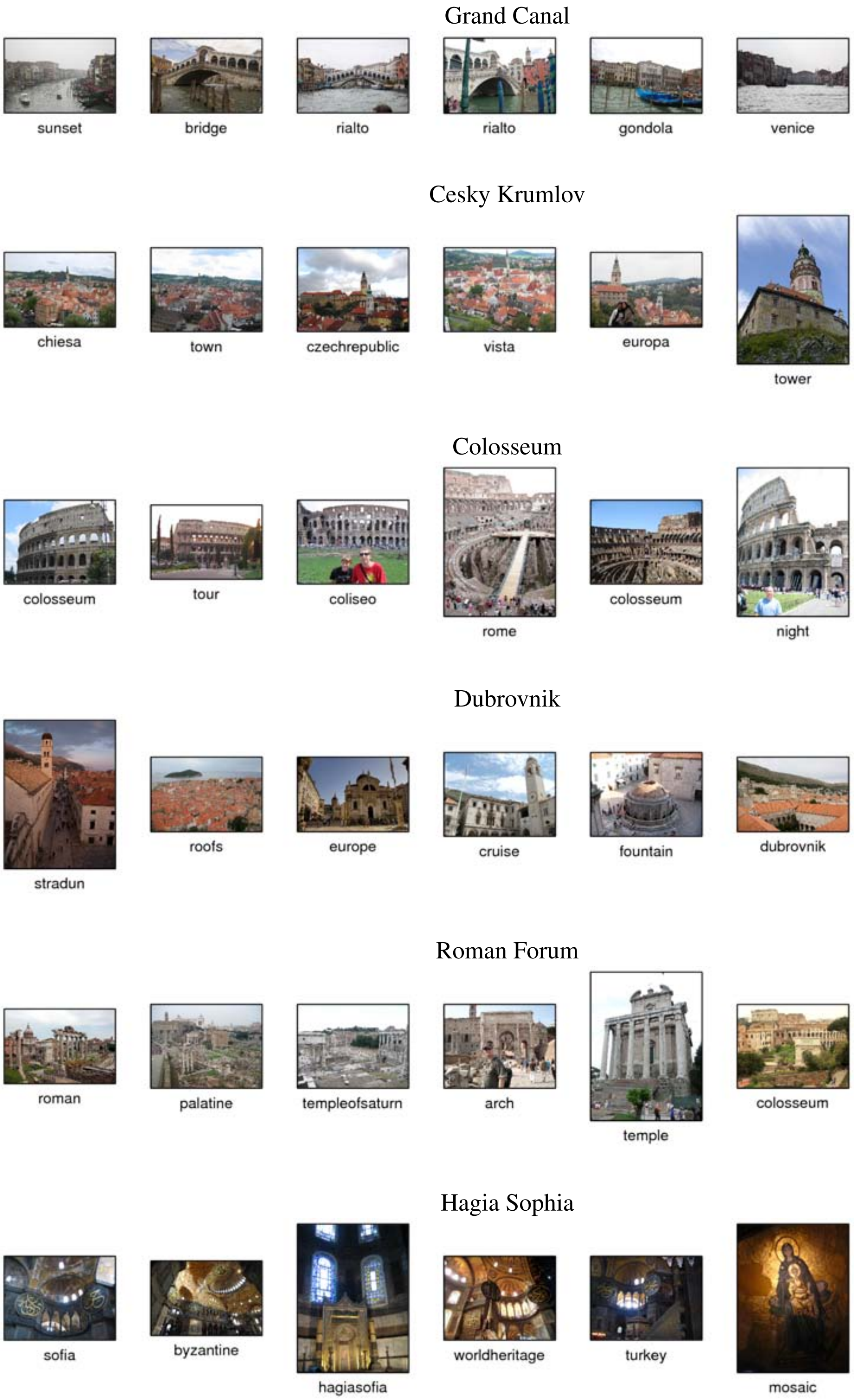
Figures 3.6, 3.7, and 3.8 show summaries computed by my greedy k-means algorithm for all scenes, with tags automatically selected by the approach described in Section 3.6.2.

3.5 Evaluation

In this section I describe qualitative and quantitative approaches for evaluating scene summarization algorithms.

3.5.1 Comparison of summarization objectives

Other exemplar-finding algorithms have been proposed for summarization. One well-known approach is VisualRank [63], an analogue to Google's PageRank [90] algorithm for ranking web search results. The idea is that instead of connections based on hyperlinks, connections between images are based on visual similarity between the two images. Then, the VisualRank of an image is some estimate of its centrality in the underlying collection of images. While centrality may be a reasonable criterion for ranking web search results, it is less reasonable as the sole criterion for constructing a summary set of images. Image similarities, unlike the web hyperlinks used by PageRank,



Grand Canal

Cesky Krumlov

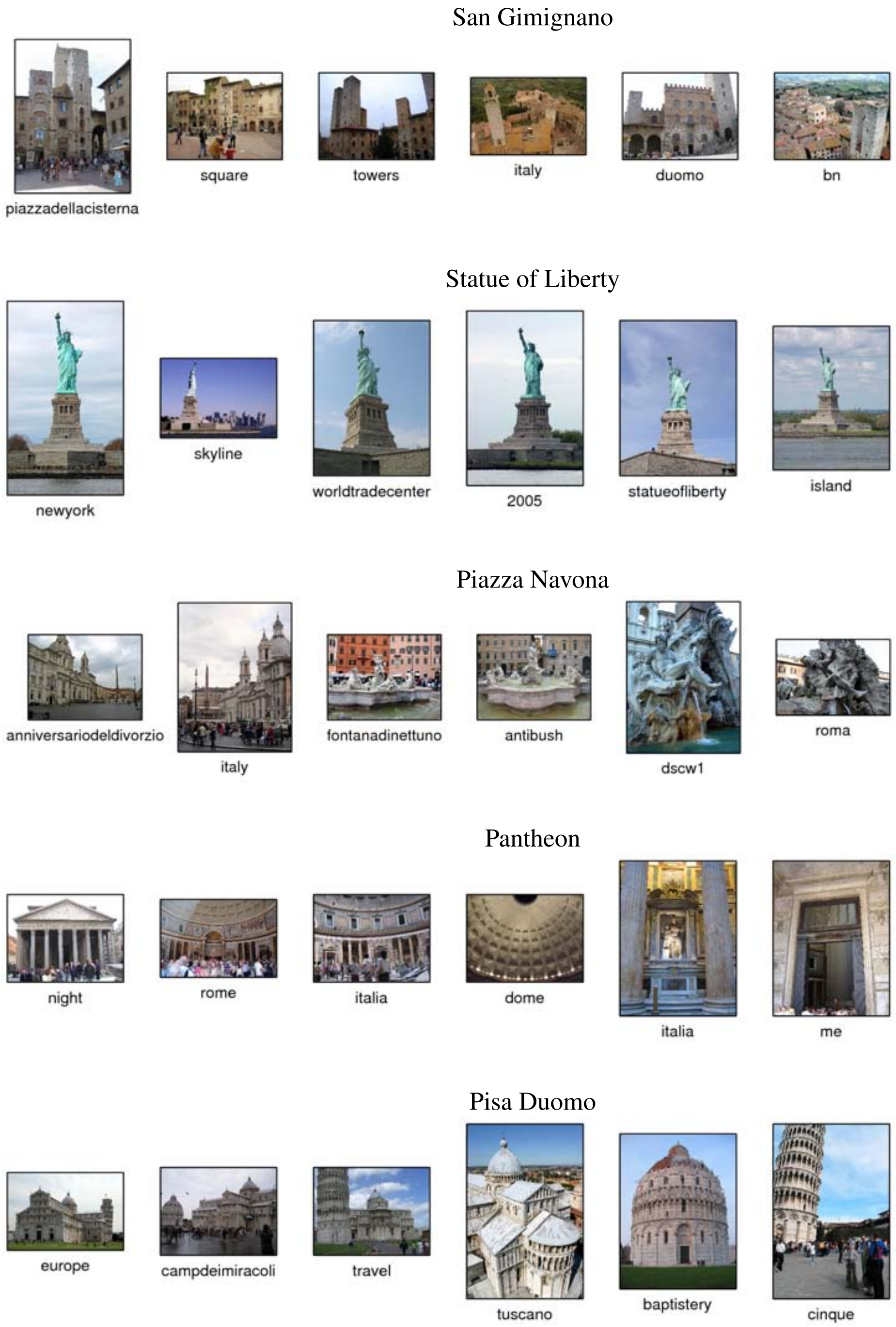
Colosseum

Dubrovnik

Roman Forum

Hagia Sophia

Figure 3.6: Six-image summaries of several scenes, with orthogonality penalty $\beta = 100$.



San Gimignano

Statue of Liberty

Piazza Navona

Pantheon

Pisa Duomo

Figure 3.7: Six-image summaries of several scenes, with orthogonality penalty $\beta = 100$.

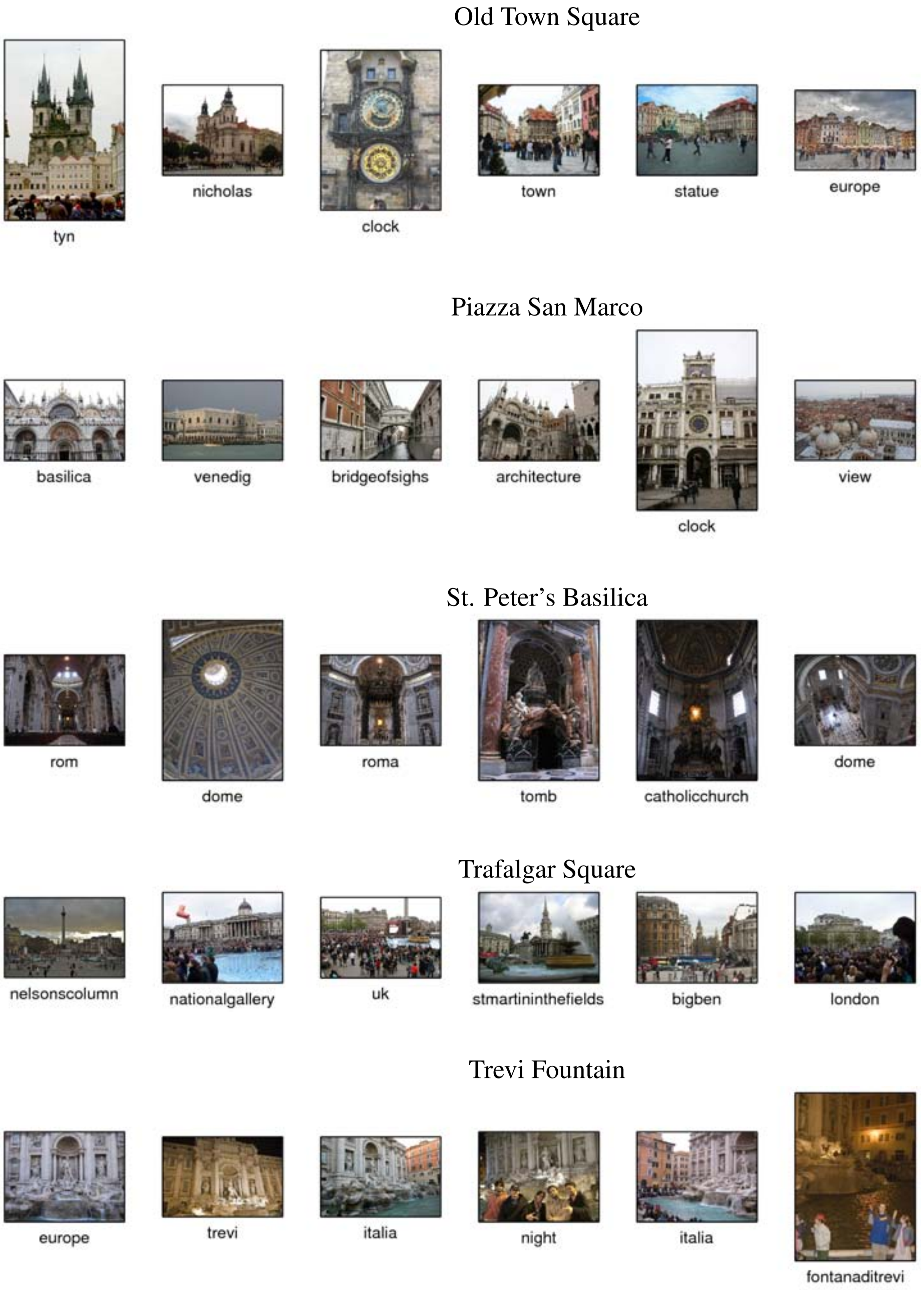


Figure 3.8: Six-image summaries of several scenes, with orthogonality penalty $\beta = 100$.

are symmetric and near-transitive — if image A is similar to B and B is similar to C, then it is likely that A is also similar to C. This means that if two images are visually similar, they are likely to have similar VisualRank, and the set of highest-ranking images is likely to consist of near-duplicates.

Another summarization algorithm that could be used is an exemplar-based version of mean shift [23], where the means must be points in the data set. Mean shift works by initializing a set of windows centered on each of the data points, and repeatedly shifting each window to the mean of the points it contains until convergence. The resulting fixed points are then chosen as exemplars. This algorithm has a few drawbacks. First, since it is not optimizing anything, it's unclear how other desired criteria, such as the orthogonality constraint, could be incorporated. Second, in practice, the mean shift algorithm often returns summaries containing several dozen images for small scenes even at the maximum window size, where any two views with nonzero overlap are considered neighbors.

Figure 3.10 shows a summary of Old Town Square in Prague computed by VisualRank, mean shift, a greedy covering algorithm that tries to cover as much of the *scene* as possible, and my greedy k-means algorithm that attempts to explain as many *images* as possible. As expected, the VisualRank algorithm produces a summary set containing only images of the Tyn church, the most-photographed object in the scene. The scene coverage algorithm tends to choose images with lots of "stuff", whether or not these are commonly-photographed views. (This may also be a reasonable approach to summarization.) Mean shift is unable to produce a summary of fewer than 11 images for this scene, and the summary set contains some redundancy.

3.5.2 Comparison of algorithms for summarization objective

I use the greedy k-means algorithm to optimize the similarity objective in Section 3.4.1, as it is conceptually simple and has a guaranteed approximation ratio. However, other algorithms exist that attempt to maximize the same objective. One simple algorithm is ordinary k-means with the means restricted to input data points, which performs poorly, as discussed above. Another algorithm that has received some attention is *affinity propagation* by Frey and Dueck [44]. Affinity propagation uses a message-passing scheme to simultaneously compute exemplars and clusters. While this algorithm provides no optimality guarantees, it has been shown to perform well in practice for a number of problems [32, 31, 49]. However, for clustering problems that arise from similarity graphs on

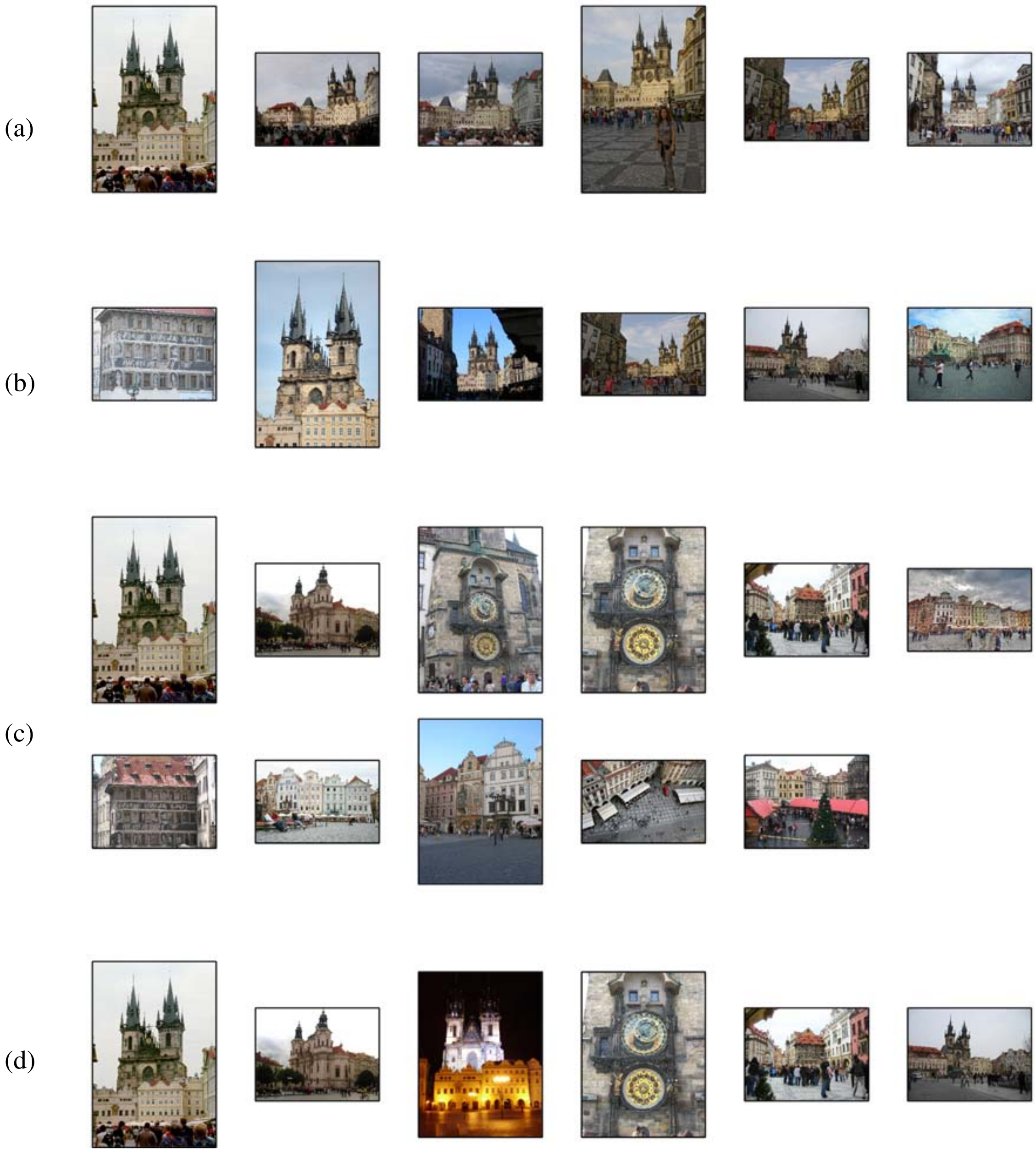


Figure 3.9: Scene summaries of Prague's Old Town Square computed by (a) VisualRank, (b) a greedy scene covering algorithm, (c) mean shift, and (d) greedy k-means. Note that summary (c) contains 11 images.

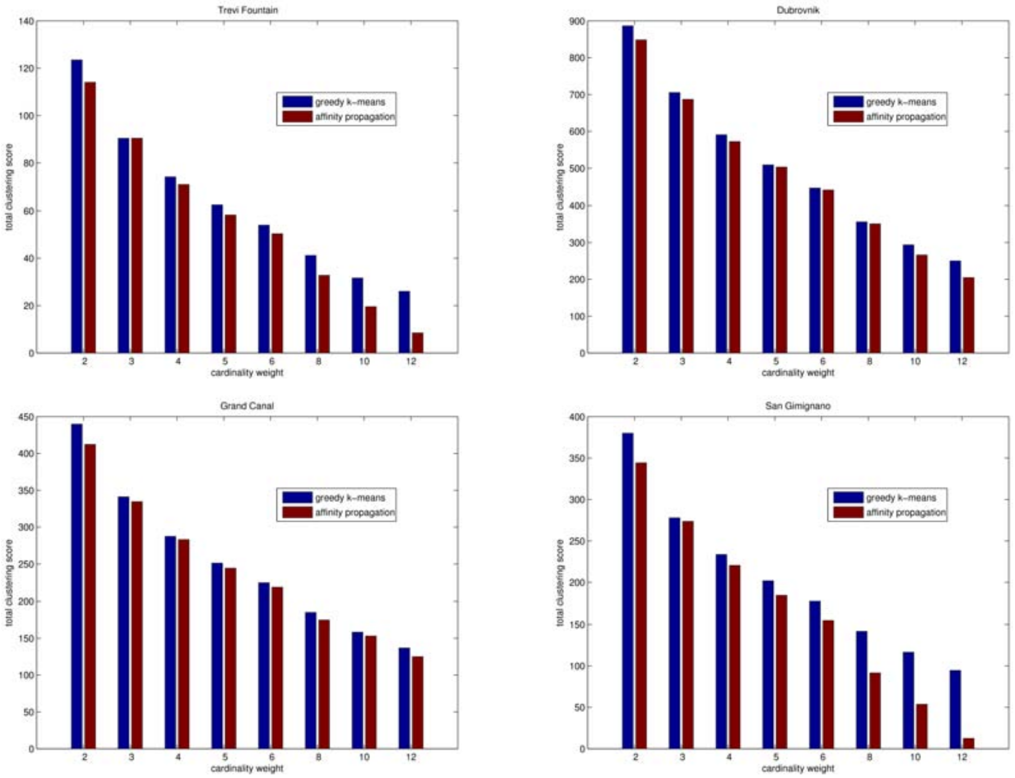


Figure 3.10: Greedy k-means and affinity propagation objective function values for several scenes, and multiple values of the cardinality weight α for each scene. Greedy k-means nearly always outperforms affinity propagation, with a greater difference for larger values of α (fewer clusters).

Flickr photo collections, greedy k-means outperforms affinity propagation the vast majority of the time.

Figure 3.10 shows the objective function values achieved by greedy k-means and affinity propagation for a few scenes and several values of the cardinality weight α, which indirectly controls the number of canonical views. Table 3.1 shows the median ratio across several different values of the cardinality penalty α of the objective achieved by greedy k-means to the objective achieved by affinity propagation. In all cases greedy k-means outperformed affinity propagation, though not by a huge factor. However, the added simplicity of greedy k-means further justifies its use on these problems.[1]

[1] One potential drawback to greedy k-means is its sequential nature — each exemplar depends on the previous exem-

	median ratio		median ratio
Grand Canal	1.04	Cesky Krumlov	1.09
Colosseum	1.15	Dubrovnik	1.04
Roman Forum	1.07	Hagia Sophia	1.07
San Gimignano	1.10	Statue of Liberty	1.09
Piazza Navona	1.05	Pantheon	1.14
Pisa Duomo	1.09	Old Town Square	1.06
Piazza San Marco	-	St. Peter's Basilica	1.06
Trafalgar Square	1.18	Trevi Fountain	1.07

Table 3.1: Median improvement (over multiple values of the cardinality penalty α) in objective function value achieved by my greedy k-means algorithm over affinity propagation for all scenes except Piazza San Marco, on which affinity propagation ran out of memory. Greedy k-means outperformed affinity propagation for all scenes, is guaranteed to achieve a constant factor of the optimal objective value, and is conceptually much simpler than affinity propagation.

3.6 Image browsing application

In this section, I describe an image browsing application that can be constructed on top of this basic scene summarization method, extending it to photo collections containing many scenes, and incorporating user-specified tag data into the summaries.

The photo sharing website Flickr [2] contains over 2 million photos of the city of Rome, spanning such sites as the Colosseum (over 100,000 photos), St. Peter's Basilica (over 70,000 photos), and the Trevi Fountain (over 60,000 photos). The photos are organized by user-specified tags, by photographer, and, in some cases, by timestamps and geotags. A system for summarizing and browsing photos using only this data is handicapped in several ways, as illustrated in Figure 3.12:

- Some photos are missing relevant tags. A photo of the Colosseum may be tagged with "rome" but not "colosseum". Thus, a summary of the Colosseum could not include this photo, and it

plars selected. The message-passing approach taken by affinity propagation may be more easily parallelizable.

would not be browseable under the “colosseum” tag.

- Some photos have tags that are misleading. A user might tag a set of photos with “vatican” even if some of the photos were not taken in the Vatican. A summary of the Vatican may then include these photos incorrectly.
- Some photos have tags that are uninformative. A user might tag a set of photos with “vacation2005”, which may be useful for the user’s own photo organization, but useless for creating a summary of Rome that is of value to multiple users.
- Tags are essentially useless for summarizing or browsing a scene where most of the photos depict the same object. In the case of the Trevi Fountain, a flat index of over 60,000 photos is too large to browse, and the variation among the photos is not reflected in the tags.

I present a prototypical browsing application using our scene summaries that resolves all of these issues. The application can function in the complete absence of tags, but when tags are provided, our system can extract tags which are likely to be correct and use them to enhance the browser.

3.6.1 Organizing the photos for browsing

For a single scene, our set of canonical views $\mathcal{C}$, along with the mapping from each image in $\mathcal{V}$ to its most similar canonical view, can serve as a simple two-level hierarchy for image browsing. The top level of the hierarchy contains the canonical views, and beneath each canonical view C is the set of images $V \in \mathcal{V}$ such that C is the most similar canonical view to V.

For larger image collections that span multiple scenes, we add another level to the top of the hierarchy. We construct the three-level hierarchy in two steps:

1. Find the connected components of the image collection; each connected component comprises a *scene*, as defined in Section 3.3.

2. For each scene, use the greedy algorithm from Section 3.4.1 to compute the canonical views.

A browseable summary of the city of Rome can be found on our project web page, located at `http://grail.cs.washington.edu/projects/canonview/`. Screenshots of this summary can be seen in Figure 3.11. Summaries computed from even larger photo collections can be found at `http://grail.cs.washington.edu/rome/`. These city-level summaries of Rome (150,000 images), Venice (250,000 images), and Dubrovnik (57,845 images) are computed using the matching and reconstruction system of Agarwal et al. [8] and my summarization algorithm.

Note that image connectivity does not necessarily correspond with the semantic concept of a scene, and connected components is prone to oversegmentation and undersegmentation. One possible solution to the undersegmentation problem is to use the soft clustering approach of Philbin and Zisserman [98]. In the next section, I introduce a different technique for getting around scene connectivity issues, by exploiting tag data.

3.6.2 Incorporating tag data

At either the scene or cluster level, I enhance the summaries by displaying one or more tags for each canonical view. As the user-specified tags associated with each image may be unreliable (see Figure 3.12), I look at all images in the scene or cluster to choose the tags to display. Two main difficulties arise in selecting appropriate tags:

1. The most popular tags in the cluster may be associated with a broader concept than the cluster itself. For example, the most popular tags for a cluster containing images of the Pantheon may be “italy” and “rome”.

2. The occurrence of a tag may be highly correlated with the cluster because of the behavior of a few users. Tags like “anniversary2005” or “jason” could be strongly associated with a cluster, but do not help describe the scene.

I define a function $\mathrm{score}(c, t)$ that measures how well tag t describes cluster c. A first approach might be to choose tags with large values of $P(t|c)$. However, this falls into the first trap above, and assigns tags that are correct, but not very discriminative, like “rome” or “italy”, to most clusters. One might also consider choosing tags with large values of $P(c|t)$. This avoids the first problem,

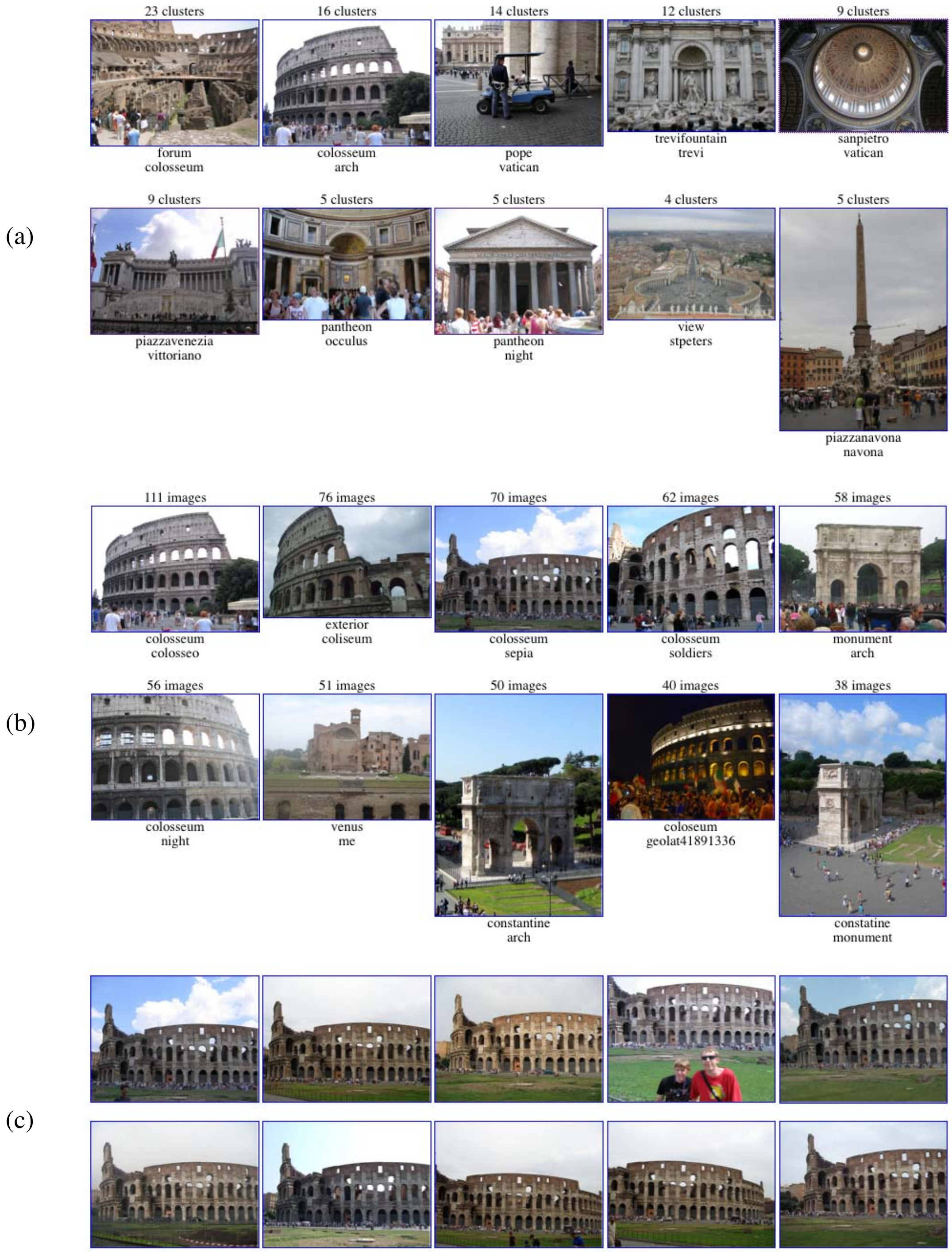


Figure 3.11: **(a)** A top-level summary of 20,000 images of Rome. **(b)** A scene-level summary of the Colosseum exterior, accessed via the second image of the top-level summary. **(c)** One of the clusters of Colosseum images, associated with the third canonical view of the scene-level summary.

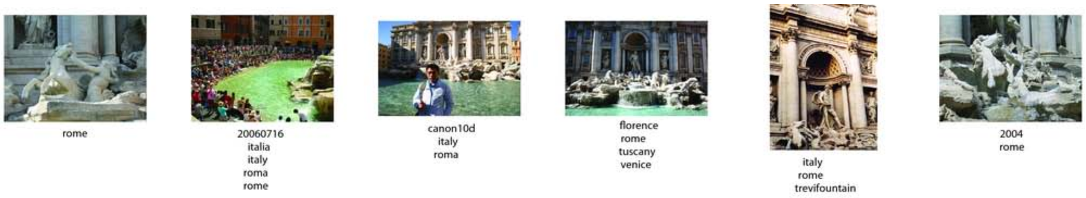


Figure 3.12: Six randomly selected images of the Trevi Fountain, and tags given to each image by Flickr [2] users. Note that none of these tags are useful for browsing images of the Trevi Fountain.

but ends up choosing useless tags that happen to be discriminative, like “anniversary2005”, for most clusters. To resolve both issues, I compute the score as a pointwise divergence between the joint probability $P(c,t)$ and the probability of co-occurrence if the cluster and tag are independent:

$$\mathrm{score}(c,t) = P(c,t)\log\frac{P(c,t)}{P(c)P(t)} \tag{3.7}$$

For all probabilities, we treat each image as a sample and count the number of co-occurrences. We therefore define:

$$P(c,t) = \frac{\left|\{V\in\mathcal{V}\ ,\ c(V)=c, t\in T(V)\}\right|}{|\mathcal{V}|} \tag{3.8}$$

$$P(c) = \frac{\left|\{V\in\mathcal{V} \mid c(V)=c\}\right|}{|\mathcal{V}|} \tag{3.9}$$

$$P(t) = \frac{\left|\{V\in\mathcal{V} \mid t\in T(V)\}\right|}{|\mathcal{V}|} \tag{3.10}$$

where $c(V)$ is the cluster associated with view V, and $T(V)$ is the set of tags associated with V. Note that strictly speaking, P represents frequencies instead of probabilities, as we are only measuring the former. I also remove all tags which have been given by fewer than 3 users, as these are likely to be noise. I use the above score function at both the scene and cluster level. However, for many clusters, accurate tags do not exist or are rare enough to be indistinguishable from user-specific tags.

Using tags for browsing can avoid problems associated with the connected components segmentation. For example, in Figure 1.1, the Pantheon is split among multiple segments, since connecting images are missing. However, in our browseable index (Figure 3.11), we also allow a user to view

	top 1	top 3	best		top 1	top 3	best
Grand Canal	4	4	4	Cesky Krumlov	1	1	1
Colosseum	0	0	0	Dubrovnik	3	4	5
Roman Forum	3	4	4	Hagia Sophia	1	1	2
San Gimignano	4	5	5	Statue of Liberty	1	1	1
Piazza Navona	1	3	4	Pantheon	1	2	4
Pisa Duomo	1	1	2	Old Town Square	4	4	5
Piazza San Marco	4	5	6	St. Peter's Basilica	2	3	4
Trafalgar Square	4	4	5	Trevi Fountain	0	0	0

Table 3.2: Number of accurately-tagged clusters for each scene, out of the top 6 clusters computed by my summarization algorithm. For each scene, my tagging algorithm selected its top 3 tags for each cluster, in order. This table lists the number of clusters for which the top tag was correct, the number of clusters for which one of the top 3 tags was correct, and the number of clusters that are taggable. For example, different views of the entire Trevi Fountain are not really taggable in any meaningful way.

the set of clusters associated with a given tag. Under the “pantheon” tag, clusters from both segments appear in the index. Note that this is not the same as ordinary browsing by tags, for example on Flickr [2], as in our index many of the images browseable under the tag “pantheon” were not given the tag by any Flickr user.

Tag evaluation

To evaluate the automatic tag selection algorithm, I computed the top 3 tags for the top 6 clusters in each scene, and counted the number of clusters for which (a) the top tag, (b) any of the top 3 tags, and (c) any possible tag could be said to accurately describe the cluster (but not the entire scene). The results are shown in Table 3.2. For most scenes, where the clusters are even taggable, my algorithm selects good tags for most of the clusters. An example of a scene with untaggable clusters is the Statue of Liberty — the clusters just consist of different views of the statue. While this evaluation is subjective, I have also included the summaries and tags themselves (Figures 3.6, 3.7, and 3.8), so the reader may perform his/her own evaluation.

3.7 Discussion

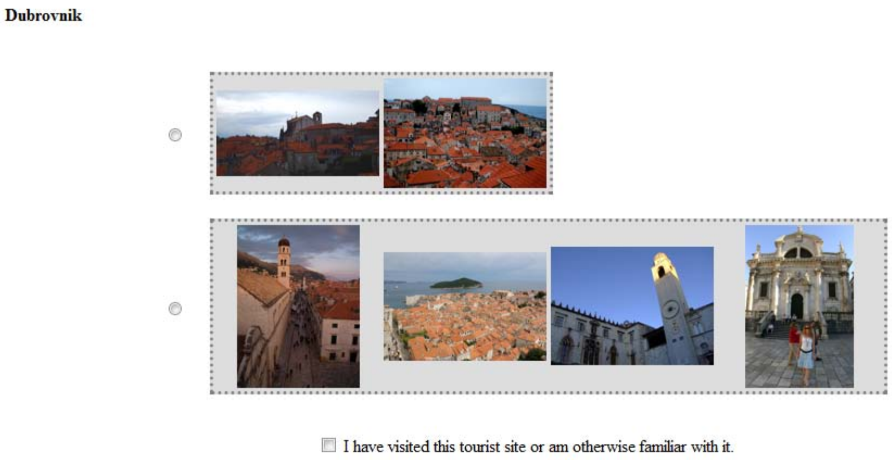
In this chapter, I defined the problem of scene summarization, and provided an algorithm that solves this problem on large image sets. When textual tags are associated with each image, they can be used to enhance the summaries, in spite of the large amount of noise in the tags. I also demonstrated an image browsing application that uses this summarization approach, and show how scene summaries can serve as portals into an interactive 3D browser (see the project web page at `http://grail.cs.washington.edu/projects/canonview/`).

The computed summaries and image browser allow a user to quickly navigate a huge collection of images in a way that was previously impossible, and has the capability to greatly enhance the experience of browsing photo collections on Flickr [2] and other photo sharing sites.

3.7.1 What do people look for in a summary?

One interesting question that remains to be answered fully is what criteria humans use for judging or creating scene summaries. Kennedy and Naaman [69] performed an experiment where they had subjects evaluate scene summaries for representativeness (few irrelevant photos), uniqueness (little overlap among photos), comprehensiveness (no important views missing), and overall satisfaction. The summaries to be evaluated were generated by several clustering methods using combinations of tags, visual features, and location. Using visual features was found to be a significant improvement over tags and location, especially in terms of representativeness and overall satisfaction. However, this improvement may merely reflect the fact that visual matching is useful for pruning irrelevant photos (since they won't match anything).

In an attempt to discover some of the summarization criteria used by humans, I set up a similar experiment using workers from Amazon Mechanical Turk [1]. Each worker was shown two summaries of the same scene, for three different scenes, and had to choose the preferred summary for each scene. I also asked each worker to indicate whether or not he/she was familiar with each scene. The summaries shown were of two types: some were automatically computed using greedy k-means



Dubrovnik

○

○

□ I have visited this tourist site or am otherwise familiar with it.

Figure 3.13: A screenshot of one of the summary comparisons shown to Mechanical Turk users.

with varying numbers of clusters, and others consisted of the photos from a single Flickr user's photostream. In total, there were approximately 30 summaries per scene for 15 different scenes, and each pair of summaries for the same scene was compared 4.6 times, for 30000 total comparisons.

I gave the following instructions:

> You will be presented with two sets of photos of several tourist sites. For each tourist site, choose the set of photos you would prefer to include in a guidebook entry about the site. An ideal set of photos should show the important views of the site with as little repetition as possible. You may also consider other criteria such as image quality or relevance.

A screenshot of one of the summary comparisons can be seen in Figure 3.13.

The results of this experiment were mostly inconclusive. 72.8% of the time, the longer summary was chosen. Since each pair of summaries was evaluated by multiple workers, it's possible to compute the consistency of the evaluations. 76.5% of the time, the summary preferred by a particular worker was the summary chosen most often when the exact same pair was compared. This is an upper bound on how well any feature of the summaries could predict summary preferences, meaning that simply choosing the longer summary will achieve close to optimal accuracy. All workers agreed on 2127 out of 6514 evaluation pairs (32.7%). For these pairs with unanimous agreement, the longer summary was preferred 95.1% of the time.

For summaries that were the same length, the summary preferred by a particular worker was the more popular of the two summaries 70.0% of the time, indicating that there was at least some agreement beyond length. However, neither similarity to the collection nor orthogonality was able to predict the correct classification with much accuracy, achieving 52.9% and 55.0% accuracy, respectively. Simple measures of image quality were also not good predictors of the preferred summary.

The worker claimed to be familiar with the scene only 18.8% of the time. The most recognized scene was the Statue of Liberty (26.8%) and the least recognized was Dubrovnik (14.9%). Claiming to recognize a scene required checking a checkbox, so these low familiarity rates can at least partly be attributed to laziness. Overall, the outcome of this experiment illustrates the difficulty of eliciting scene summary preferences from humans in a meaningful way. An interesting future experiment might be to ask users to construct summaries themselves from a collection of images of manageable size.

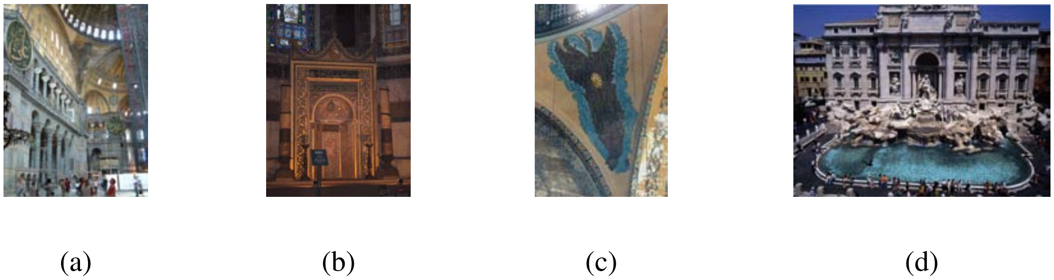
Chapter 4

SCENE SEGMENTATION USING THE WISDOM OF CROWDS

The distribution of photos in a large collection holds valuable information about the content of the scene. In the previous chapter, I used this information to select interesting *views* of the scene. In this chapter, I seek to leverage this information to automatically identify and segment interesting *objects*. While extremely challenging to solve purely by analyzing pixels in a single image, this problem is much more tractable with a large image collection. For example, a robust interest operator is obtained by simply finding features (e.g. SIFT [80]) that appear in numerous photos. By identifying oft-photographed features, this operator tends to highlight the parts of the scene that people find most interesting. The fact that this works robustly is a powerful demonstration of *the wisdom of crowds* [121], where a community of people combine to provide information that is otherwise difficult to obtain.

While detecting interesting features is straightforward via simple counting methods, identifying interesting objects is more challenging, as it necessitates segmentation — another difficult, ill-posed computer vision problem. To address this segmentation problem, I propose a new *field-of-view* cue for inferring perceptual grouping information from large photo collections. The key idea is very simple—it's based on the observation that people tend to take photos of "objects" as opposed to arbitrary regions of the scene, and these objects are usually framed within the field of view of the photo. Consequently, if two points are on the same object, those two points will likely appear together in many photos. I therefore use the co-occurence of features in many images as a cue for grouping them together.

In this chapter, I introduce this field-of-view cue and techniques for leveraging it for identifying and segmenting objects in images and point clouds. I also demonstrate the use of this analysis to display the relative importances of objects, automatically compute textual labels for these objects from noisy user-contributed Flickr tags (which are attached to images, not image regions), and browse a scene using an interactive map.



(a) (b) (c) (d)

Figure 4.1: (a) A “bad” image of the Hagia Sophia. (b,c) Two “good” images of objects in the Hagia Sophia. Good images provide useful segmentation cues. (d) An image of the Trevi Fountain showing why photo collections of some scenes may not contain many good images. In this case, there are interesting statues in the façade, but it is difficult for photographers to get close enough to photograph them individually.

4.1 Related work

The problem of decomposing a set of images into recurring objects in an unsupervised manner has been the subject of much recent work in computer vision, such as Fergus et al. [38], Sivic et al. [116] and Sudderth et al. [120]. In this chapter, I address this problem for static scenes, where objects always have the same 3D context and variation among images arises from differing camera positions and viewing directions.

Russell et al. [104] use grouping cues in an unsupervised approach to object category segmentation. They compute multiple segmentations for each image and then apply a latent topic model where the *segments* serve as documents, relying on the fact that each object will appear against several different backgrounds, so “bad” segments will contain multiple latent topics. In this thesis, however, as I deal only with static scenes, any image region will almost always appear in the same context. I instead rely on the field-of-view of human-taken photographs to provide grouping cues. Similar camera cues are used in other work. For example, Philbin and Zisserman [98] use spectral clustering to extract objects from a set of photographs. Unlike my approach in this chapter, they operate under the assumption that each image only contains a single object. Epshtein et al. [35] use the distribution of viewing frusta (and ignoring visibility) in a photo collection to organize the photos into a hierarchy. Campbell et al. [18] use a camera fixation cue that is similar to a field-of-view cue to solve a somewhat different problem, computing a consistent 3D shape from a set of photographs and using it to segment an object in a set of images.

In Section 4.5.2, I describe my approach for automatically annotating images containing the extracted objects. Related work on image auto-annotation is discussed in Chapter 2.

4.2 Segmentation of static scenes

Much existing work deals with identifying and segmenting object categories from visual scenes, where each object may appear against different backgrounds in different images. I address a different question: how can one identify and segment interesting objects from a static scene? The fact that the background changes was important for previous work, and enabled a solution using latent topic models. While handling static scenes may seem like a simpler problem, the fact that the background is not changing is a challenge — different cues are needed.

I instead use information provided by the distribution of photos taken of the scenes. To segment the scenes, I use what could be called a *field-of-view* or *incidence* constraint. For scenes that contain individual objects of interest, I hypothesize that multiple photographers are going to take pictures "of" each object: pictures in which the object is prominent and takes up most of the frame. I call such images "good", and other images "bad". Good images, for our purposes, are ones which provide useful segmentation cues. Figure 4.1 shows a few example images of both types. When enough images are good, field-of-view constraints can be used to accurately segment the scene into interesting objects. In the remainder of the paper, I describe a simple scene model which takes advantage of field-of-view constraints and evaluate the effectiveness of this model on several different scenes. I also demonstrate the three applications of scene segmentation mentioned above: an "interestingness" viewer, an automatic object labeler that uses user-submitted Flickr tags, and an overhead map browser. In addition, I discuss some observations about the relationship between objects and the photos people take.

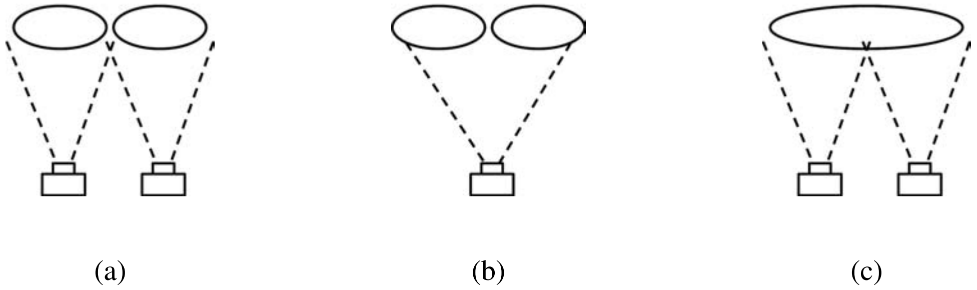


Figure 4.2: (a) A case in which field-of-view cues and spatial cues agree, indicating a pair of objects. (b) A case in which field-of-view cues indicate a single object but spatial cues indicate a pair of objects. (c) A case in which field-of-view cues indicate a pair of objects but spatial cues indicate a single object.

4.3 Algorithm

Given a set of images of a scene, I first follow the reconstruction pipeline described in Appendix A:

1. Use the SIFT keypoint detector [80] to extract feature regions from all images. These regions are represented using the SIFT descriptor.

2. For each pair of images, perform feature matching on the descriptors. Prune this set of matches by using RANSAC [41] to estimate a fundamental matrix, removing all inconsistent matches.

3. Organize the matches into tracks (connected components of features) removing tracks that contain multiple features in the same image.

4. Perform structure-from-motion [118] on the set of feature tracks, which returns a set of 3D point locations for all valid tracks, as well as camera parameters for each image.

I now operate on two structures, the (sparse) point-image incidence matrix, indicating which points appear in which images, and the set of 3D point locations. Let V be the set of images and X be the set of points, with $M = |V|$ and $N = |X|$. I use $x_j \in V_i$ if point j is visible in image i. The goal

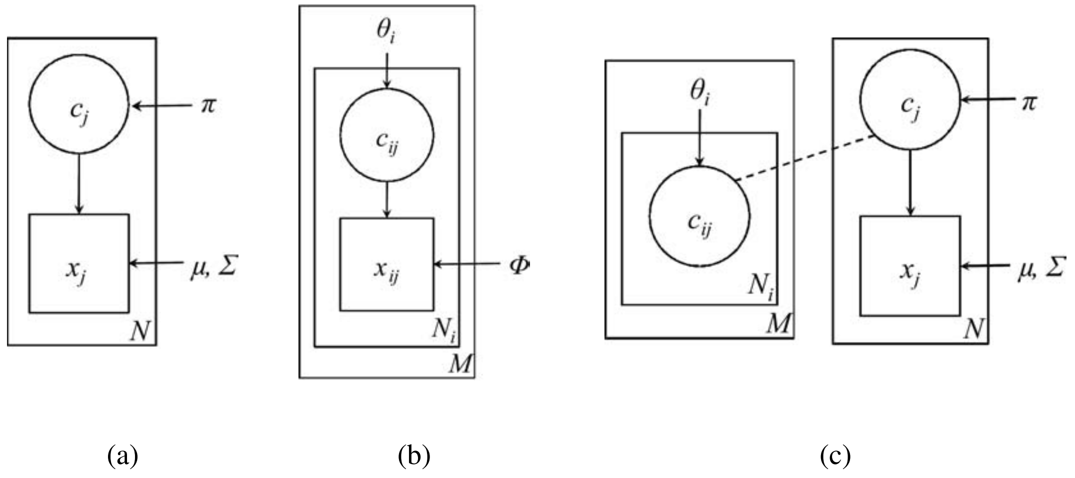


Figure 4.3: Graphical models for (a) a mixture of Gaussians on 3D point locations, (b) the pLSA model, and (c) the combined model that uses both point location and image incidence information. In each of these models, N is the number of scene points, M is the number of images, and N_i is the number of scene points visible in image i. The rectangular plates indicate repetition — the nodes within each plate are replicated for each image or point.

is to compute a clustering C over the points X, where two points belong to the same cluster if they are part of the same object.

My approach is to use both field-of-view cues and spatial cues to segment the scene. Figure 4.2 illustrates how these cues work at a basic level. Field-of-view cues encourage points seen in the same view to be part of the same object, while spatial cues encourage objects to be spatially localizable. I use image incidence to enforce field-of-view cues and a single 3D Gaussian distribution per object to enforce spatial cues. To construct a probabilistic model that takes advantage of both types of cues, I combine a mixture of 3D Gaussians (Figure 4.3a), which uses spatial cues only, and pLSA [58] (Figure 4.3b), which uses incidence cues only. My combined model (Figure 4.3c) uses both types of cues.

I briefly review the probability distributions specified by a mixture of Gaussians and pLSA, and combine them into a single model. A mixture of Gaussians corresponds to the following distribution:

$$\begin{aligned} P(C, X|\pi, \mu, \Sigma) &= \prod_j P(c_j|\pi) P(x_j|c_j, \mu, \Sigma) \\ P(c_j|\pi) &\sim \text{Mult}(\pi) \\ P(x_j|c_j, \mu, \Sigma) &\sim \mathcal{N}(\mu_{c_j}, \Sigma_{c_j}) \end{aligned} \tag{4.1}$$

In this model, there is a class variable c_j associated with each point x_j. The class variable is drawn from a multinomial distribution (Mult) with parameter π, where π_c is the probability that a point will belong to class c. The point locations are drawn from 3D Gaussians with parameters μ_{c_j} and Σ_{c_j}, where the point class c_j specifies which Gaussian to use. The pLSA model corresponds to the following distribution:

$$\begin{aligned} P(C, X|\theta, \Phi) &= \prod_i \prod_{j|x_j \in V_i} P(c_{ij}|\theta_i) P(x_{ij}|c_{ij}, \Phi) \\ P(c_{ij}|\theta_i) &\sim \text{Mult}(\theta_i) \\ P(x_{ij}|c_{ij}, \Phi) &\sim \text{Mult}(\Phi_{c_{ij}}) \end{aligned} \tag{4.2}$$

This model is usually used to decompose a set of text documents (represented as word incidences or counts) into latent *topics*. The generative process for a word incidence in an image in pLSA is the following: first draw a class c (a topic) from the image-specific class distribution θ_i, then draw a word from the class-specific word distribution Φ_c.

In ordinary pLSA, the same word can appear multiple times in a single document. In my case, however, the words come from a finite set of 3D points, none of which can appear more than once in a single image. In the pLSA model, there is a class variable c_{ij} for each point-image *incidence*. In other words, a point can belong to different objects in different images. This is not really desirable in my case, so I restrict the model to use a single class variable per point. In addition, I introduce a spatial term. The combined model corresponds to the following distribution:

$$\begin{aligned} P(C, X|\theta, \pi, \mu, \Sigma) &= \left(\prod_i \prod_{j|x_j \in V_i} P(c_{ij}|\theta_i) \right) \times \\ & \quad \left(\prod_j P(c_j|\pi) P(x_j|c_j, \mu, \Sigma) \right) \\ P(c_{ij}|\theta_i) &\sim \mathrm{Mult}(\theta_i) \\ P(c_j|\pi) &\sim \mathrm{Mult}(\pi) \\ P(x_j|c_j, \mu, \Sigma) &\sim \mathcal{N}(\mu_{c_j}, \Sigma_{c_j}) \\ c_{ij} &= c_j \end{aligned} \tag{4.3}$$

Instead of directly replacing the multinomial topic distributions from pLSA with 3D Gaussian distributions, I add a single class variable c_j for each 3D point x_j and tie the values of the incidence class variables c_{ij} with c_j. The resulting model is not strictly a valid Bayesian network, but could easily be converted to one in which the c_{ij} variables are eliminated and c_j is directly conditioned on all of the images in which the point appears (see Figure 4.3c). Another way to think about this model is as a mixture model where the data consists of point locations and *image* incidences — that is, the set of images in which a point appears. Under either interpretation, the above joint density is amenable to (local) maximization using the EM algorithm [28].

4.4 Evaluation

I performed several different experiments to test the validity of this model.

4.4.1 Hand-labeled ground truth

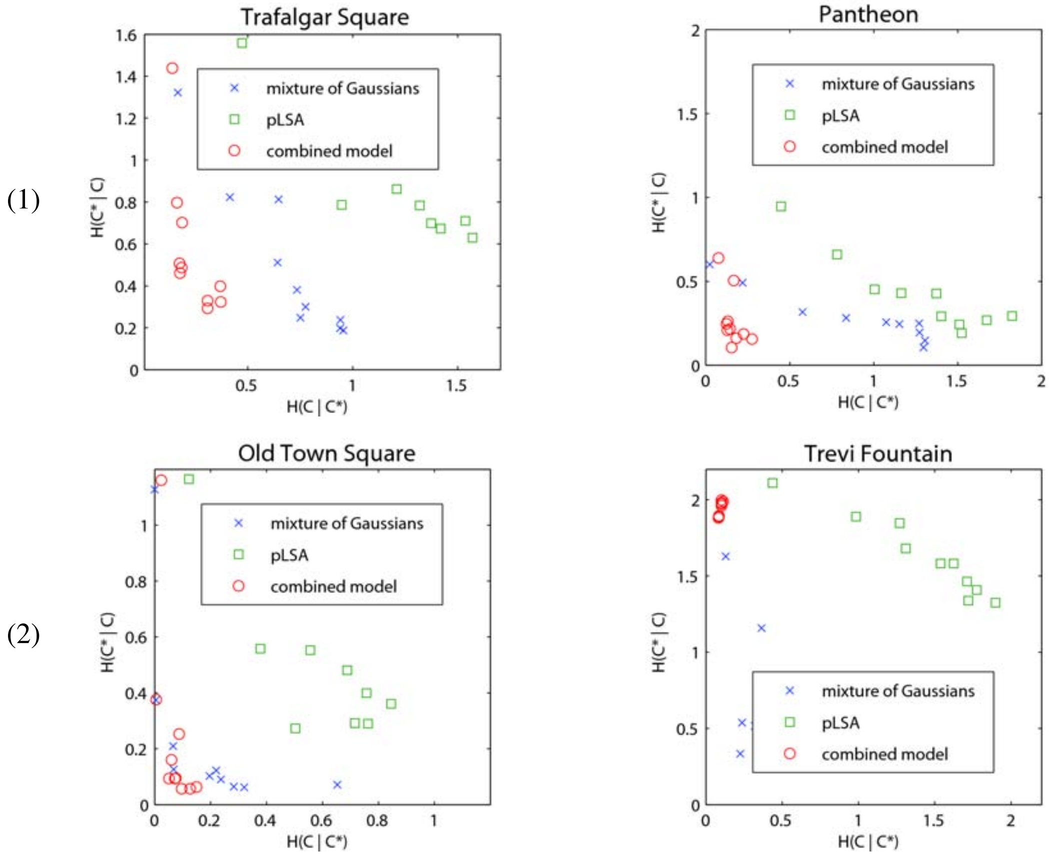
In the first experiment, I tested the model on six scenes with a hand-labeled ground-truth clustering. Each scene contains between 300 and 3000 images (exact numbers in Appendix A) downloaded from Flickr: Trafalgar Square, the Pantheon in Rome, Hagia Sophia, Trevi Fountain, Old Town Square in Prague, and Piazza Navona. These scenes all contain component objects which have names and can be identified visually. In order to test the automatic segmentations, I created ground-truth clusterings for each scene manually, as follows. For each scene, I reconstructed a set of 3D

	Trafalgar	Pantheon	Hagia Sophia	Trevi	Prague	Navona
mixture of Gaussians	1.15	1.36	0.63	**0.81**	0.35	0.68
pLSA	2.07	1.70	0.64	3.12	1.13	1.46
combined model	**0.69**	**0.38**	**0.53**	2.07	**0.20**	**0.45**

Table 4.1: Median values of the VI clustering metric $VI(C, C^*)$ for each algorithm and scene, over multiple runs of EM. Lower values indicate the computed clustering is closer to the hand-labeled clustering.

points using the system of Snavely et al. [118]. I then assigned 3D points to clusters by manually selecting regions in images and grouping all points in the selected region with a particular cluster. Though any hand-labeling of this nature is somewhat arbitrary, I attempted to label objects as uncontroversially as possible. I also used Wikipedia text and images to decide which objects should be included when there was some uncertainty. In other cases, there is a natural segmentation implied by the scene. For example, in the Trafalgar Square scene I labeled each building and statue as a separate object. Since the reconstructed scenes contain hundreds of thousands of 3D points, many of which don't belong to an easily nameable object, I only hand-labeled a small fraction of the points, and evaluate our algorithm on these points only. Note that the aforementioned manual steps were used only to create the ground truth used for evaluation purposes. The segmentation algorithms themselves are fully automatic.

For each of the six scenes, I tested three different models: a mixture of 3D Gaussians, the pLSA model, and the combined model that uses both spatial and incidence cues. Each model was tested using multiple different values of k, the number of clusters. For each value of k, I used 5 different random initializations and ran 100 iterations of the EM algorithm for each, then kept the single result with highest joint probability for each value of k. I created hard cluster assignments by assigning each point to its most likely cluster (or, for pLSA, the cluster under which the point has highest probability). I evaluate clusterings using Meila's Variation of Information metric [85]. Given a ground truth clustering C^* and computed clustering C, the VI metric $VI(C, C^*) = H(C|C^*) + H(C^*|C)$ measures the amount of information lost and gained between the two clusterings. For



(a) (b)

Figure 4.4: Evaluation of clustering C against ground truth clustering C^* for the (1a) Trafalgar Square, (1b) Pantheon, (2a) Old Town Square, and (2b) Trevi Fountain datasets. The horizontal axis $H(C|C^*)$ is a measure of over-segmentation, and the vertical axis $H(C^*|C)$ is a measure of under-segmentation. The lower left corner is optimal. Note that for the Trevi Fountain, both pLSA and the combined model are prone to undersegmentation, as most images of the Trevi Fountain are of the entire façade. This causes field-of-view cues to prefer larger objects, even when there are interesting details within the façade.

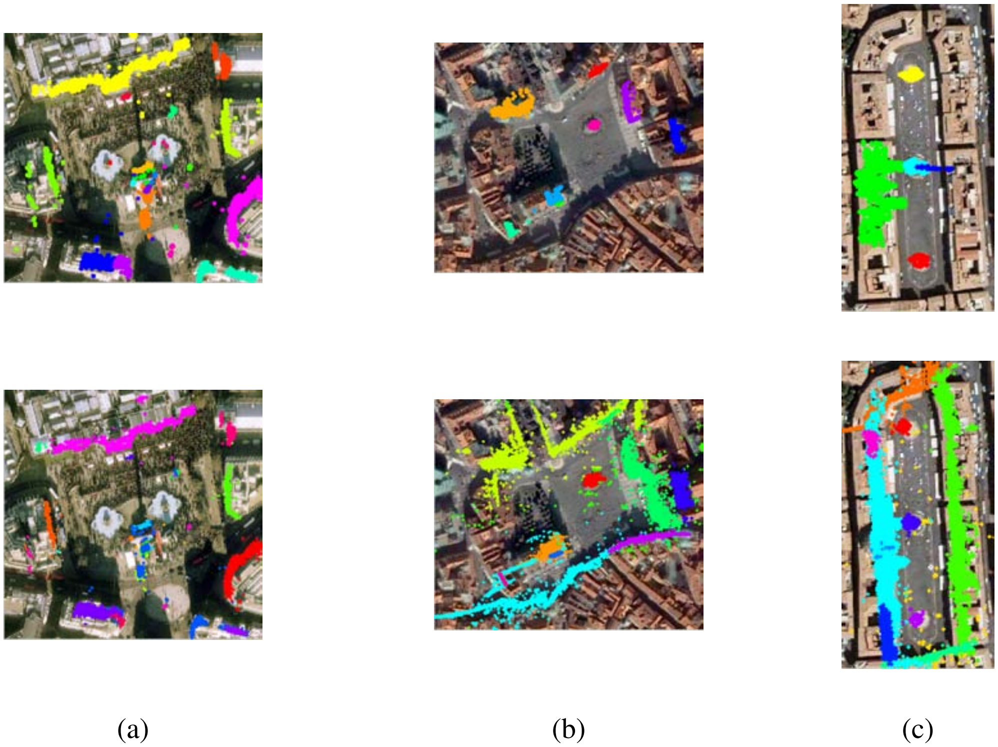
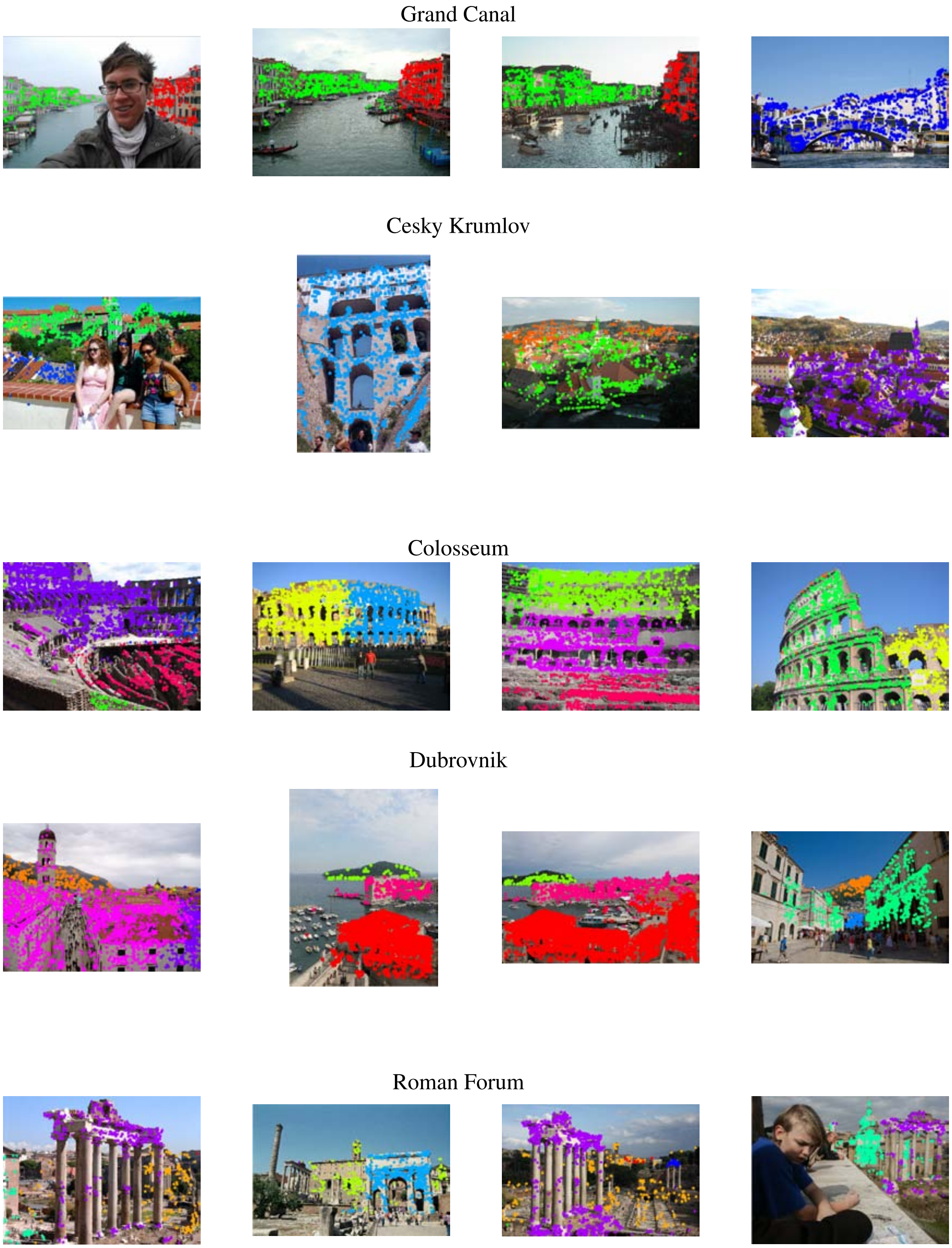


Figure 4.5: Satellite view of ground truth (top row) and computed segmentations (bottom row) for (a) Trafalgar Square, (b) Old Town Square, and (c) Piazza Navona. Each color corresponds to a different cluster. For the Trafalgar Square scene, not all reconstructed points are shown in the computed segmentation to avoid clutter. As there is no explicit correspondence between ground-truth clusters and computed clusters, the ground-truth clusterings do not use the same color scheme.

two identical clusterings, this value is zero. Also, by looking at the two conditional entropy terms separately, it is possible to get a sense of how over- and under-segmented C is.

Table 4.1 contains median VI distances for each of the three algorithms and six scenes. The combined model performs best on all scenes except the Trevi Fountain, which suffers from under-segmentation as photographers cannot get close enough to the façade to take closeup images of the interesting objects (without a zoom lens). Figure 4.4 shows our over- and under-segmentation results on four of the scenes. In general, when the collection of photographs contains many images of interesting objects, our combined model does well. Since what we are testing is whether or not



Grand Canal

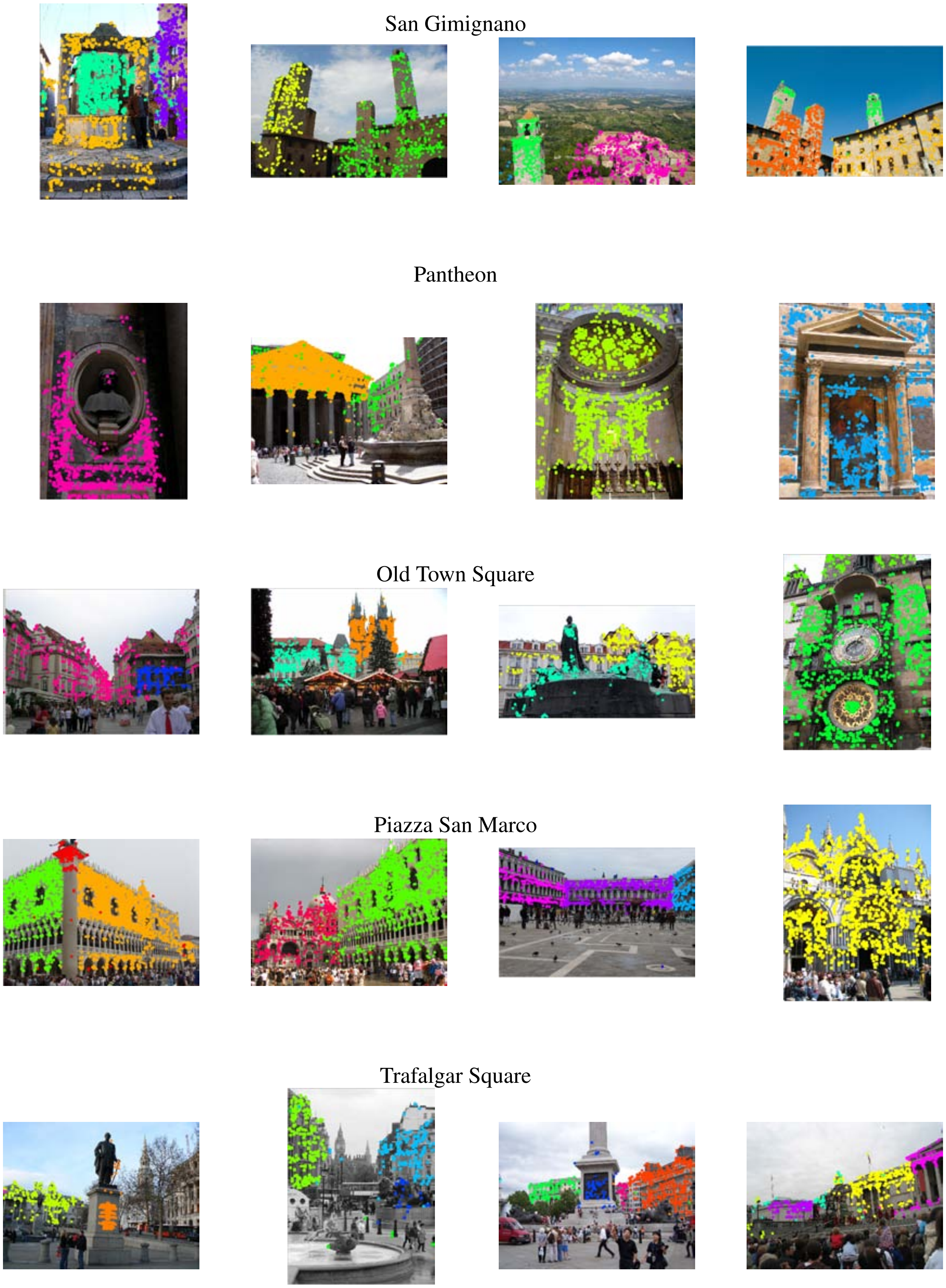
Cesky Krumlov

Colosseum

Dubrovnik

Roman Forum

Figure 4.6: Image views of scene point clusterings for several scenes.



San Gimignano

Pantheon

Old Town Square

Piazza San Marco

Trafalgar Square

Figure 4.7: Image views of scene point clusterings for several scenes.

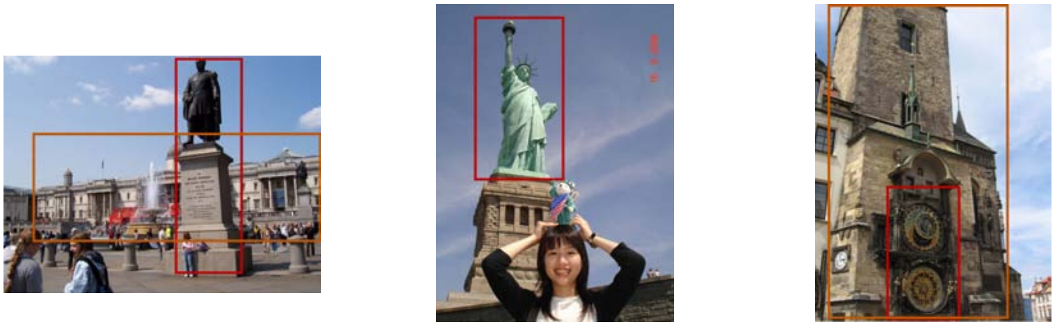


Figure 4.8: Example bounding boxes shown to Mechanical Turk workers.

field-of-view cues provide additional information that is useful for segmentation, our results demonstrate that this is true for many scenes. Visualizations of the clusterings themselves are shown in Figures 4.5, 4.6, and 4.7.

4.4.2 Mechanical Turk experiment

In addition to hand-labeled segmentations, I also performed an experiment in which I harvested rough image segmentations from Mechanical Turk and tested their agreement with my automatically computed segmentations. This is advantageous for a few reasons. First, hand-labeling scene points is quite tedious and can be done more efficiently via crowdsourcing. Second, since not all humans will necessarily agree on a segmentation, using all of these segmentations in aggregate as "ground truth" is a more principled approach than just choosing one (mine).

To obtain the image segmentations, I had Mechanical Turk users draw bounding boxes around the objects in an image, with the following instructions:

> Draw boxes around the prominent objects in the below image. Don't draw boxes around people, animals, vehicles, or anything else that is unlikely to be a permanent part of the scene. If only part of an object is visible, you may draw a box around the visible part. The above examples may be helpful.

The examples provided are shown in Figure 4.8.

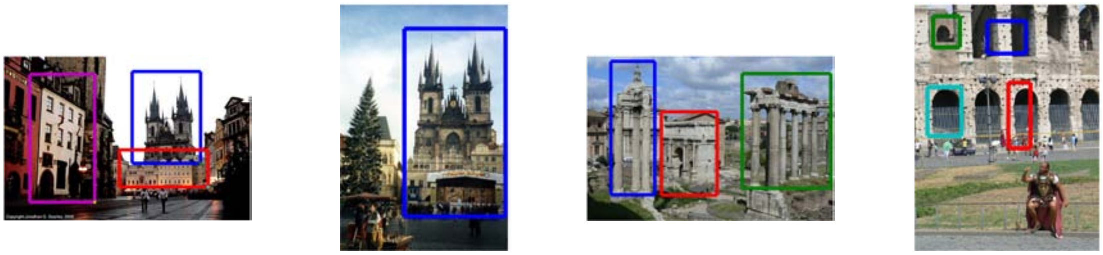


Figure 4.9: Bounding boxes drawn by Mechanical Turk workers.

In total, Mechanical Turk users drew bounding boxes in 10000 images across 15 scenes. For the most part, users behaved reasonably in the experiment. There were a few instances of seemingly random bounding boxes unrelated to any object, but most of the indicated objects were sensible. Some example bounding box segmentations are shown in Figure 4.9.

These bounding boxes provide another source of ground truth for testing our clustering algorithm based on the field-of-view cue. In the previous section, I compared the automatic clustering with a hand-labeled clustering of the scene points. With user-provided image segmentations, some difficulty arises, as there's no obvious way to combine the image bounding boxes into a global scene point clustering. Instead, I compare the clusterings in each image separately, and average the resulting VI distances. That is, for each image, I compute the VI distance between the clustering defined by the user-drawn bounding boxes and each automatically computed clustering, ignoring all scene points not visible in the image. I then consider the clustering error to be the average such distance over all images. Table 4.2 shows the median of this clustering error over multiple runs of EM with varying numbers of clusters for pLSA, a mixture of Gaussians, and my combined model. Again, the combined model using both spatial cues and field-of-view cues outperforms the other two.

4.4.3 View Framing

How do tourists frame their photos with respect to objects in the scene? The bounding boxes gleaned from Mechanical Turk can give us some insight into this question. Figure 4.4.2 shows the distribution of bounding box centers in all images. Note that it appears people are not positioning objects

	MoG	pLSA	comb.		MoG	pLSA	comb.
Grand Canal	0.71	0.84	**0.60**	Cesky Krumlov	0.49	0.80	**0.45**
Colosseum	0.42	0.68	**0.34**	Dubrovnik	0.55	0.76	**0.47**
Roman Forum	0.62	0.78	**0.54**	Hagia Sophia	0.58	0.63	**0.55**
San Gimignano	0.52	0.72	**0.48**	Piazza Navona	0.65	0.83	**0.64**
Pantheon	0.60	0.62	**0.51**	Pisa Duomo	0.58	0.88	**0.46**
Old Town Square	0.70	0.58	**0.48**	Piazza San Marco	0.57	0.57	**0.46**
St. Peter's Basilica	0.65	0.96	**0.55**	Trafalgar Square	0.58	0.71	**0.52**
Trevi Fountain	**1.12**	1.56	1.14				

Table 4.2: Median values of the VI clustering metric $VI(C, C^*)$ for each algorithm and scene, over multiple runs of EM. Lower values indicate the computed clustering agrees more with the bounding boxes drawn by Mechanical Turk users.

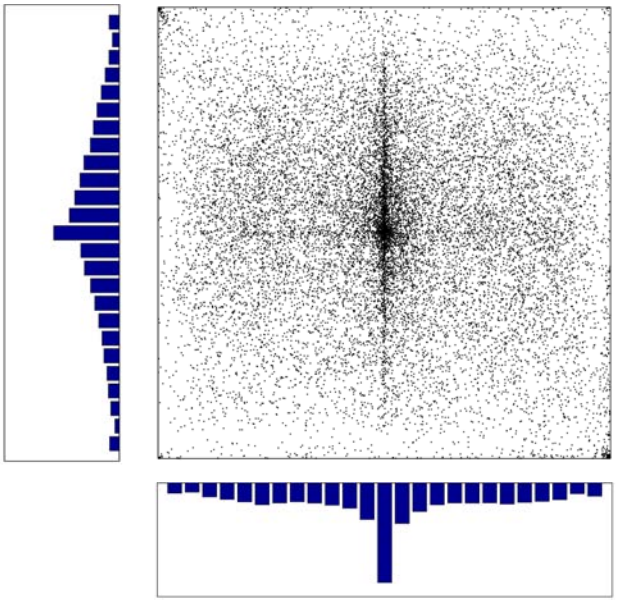
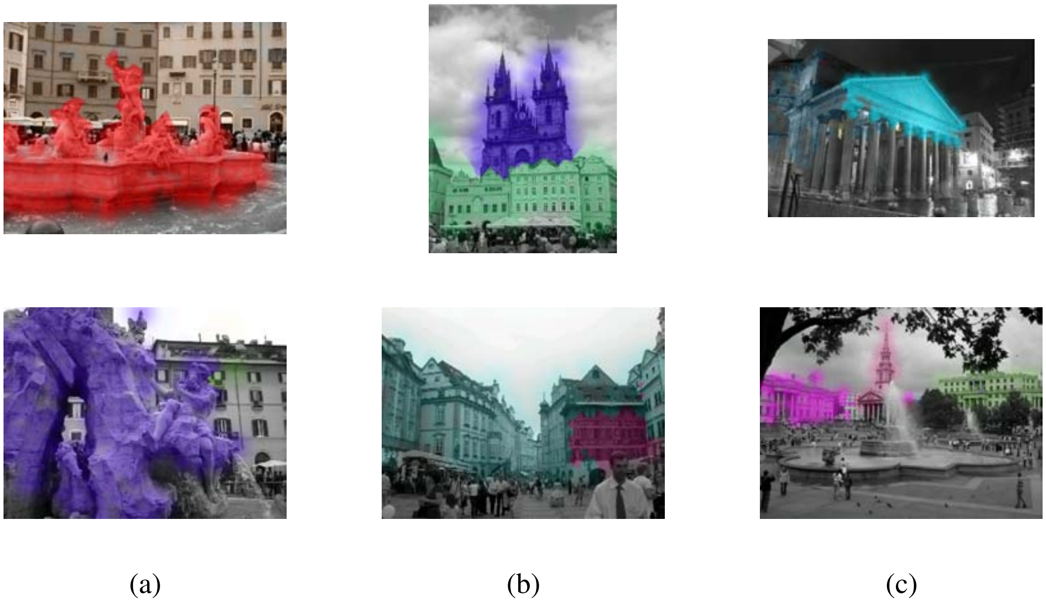


Figure 4.10: Object centers as indicated by Mechanical Turk users. Notice the increased density of centers along the vertical axis at the image center. This seems to indicate a stronger preference for horizontally centered objects than for vertically centered objects.



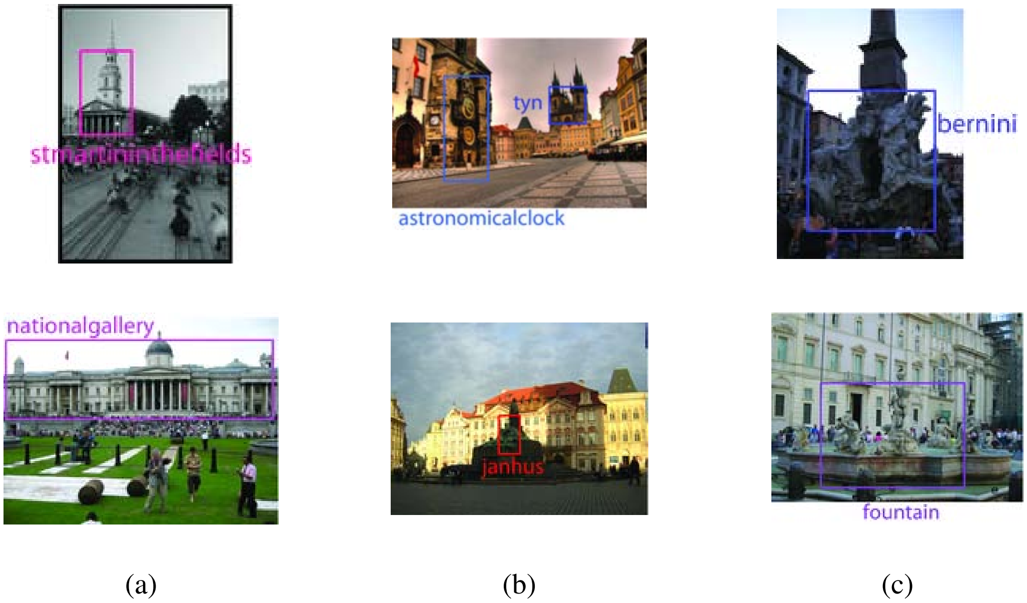
(a) (b) (c)

Figure 4.11: Importance images computed for (a) Piazza Navona, (b) Prague, and (c) the Pantheon (top) and Trafalgar Square (bottom). Importance is indicated by color saturation, with different hues for different clusters.

according to the rule-of-thirds — the only noticeable peaks are at the horizontal and vertical centers of the image. (It's possible, however, that human subjects in the photos, which I instructed Mechanical Turk users to ignore, are positioned according to the rule-of-thirds.) This agrees with the recent result of Palmer et al. [94]. In addition, the horizontal center of the image is a significantly more likely object location than the vertical center. Also, objects are slightly more likely to be located above the vertical center than below it.

4.5 Applications

Once we have computed a 3D point segmentation for a scene, we can use this segmentation to compute and display additional information about the scene. In this section, we describe two such applications: highlighting interesting objects in images and labeling image regions using noisy user-submitted Flickr tags.



(a) (b) (c)

Figure 4.12: Region tags automatically computed from our segmentation, and used to label two images of (a) Trafalgar Square, (b) Old Town Square, and (c) Piazza Navona. I compute these labels using noisy user-submitted tags downloaded from Flickr, automatically associating a single tag (or no tags) with each cluster. In these images, I manually moved the labels to make them more readable.

4.5.1 Importance viewer

Given an image in the collection, I want to identify regions which belong to objects that are important or interesting. I define an object as interesting if there are many photos of it in the collection. As this tends to overly reward large background objects, I also penalize objects for size. The importance score is:

$$\text{imp}(c) = \alpha \frac{1}{|\Sigma_c|} \sum_i \theta_i(c) \tag{4.4}$$

Here, α is a scene normalization coefficient that enforces a fixed total importance, $\frac{1}{|\Sigma_c|}$ penalizes clusters proportional to the determinant of their covariance matrices, and $\sum_i \theta_i(c)$ rewards clusters for appearing in many images. To visualize importance in an image, I assign each pixel to its nearest feature point in the image and highlight the pixel with intensity proportional to the importance of the cluster to which this point is assigned, falling off as distance to this point increases. Some resulting importance images can be seen in Figure 4.11.

4.5.2 Region labeling

Flickr and most other photo sharing sites give users the ability to attach textual tags to entire images. (Flickr also provides functionality for leaving rectangular notes on image regions, but this feature is much less utilized.) In general, these tags are noisy and the majority of them do not correspond to actual objects in the scene. For many objects, however, it is possible to compute accurate tags by examining tag-cluster co-occurrence statistics. To find good object tags, I first apply pLSA to the tags with fixed image-topic distributions θ computed from the point clustering. This gives a distribution over tags for each cluster $P(t|c)$, which also gives the joint distribution $P(c,t)$. For a particular cluster c and tag t, I compute the following score, as in Chapter 3:

$$\text{score}(c,t) = P(c,t)\log\frac{P(c,t)}{P(c)P(t)} \tag{4.5}$$

This gives high scores to cluster-tag pairs that appear much more frequently than would be expected given just the marginals, which indicates that the cluster and tag are probably related. I then assign the tag with highest score to each cluster if the score exceeds a specified threshold, otherwise I assign no tag to the cluster. Region tagging results can be seen in Figure 4.12.

4.5.3 Interactive map viewer

Many systems have been created to support interactive browsing of the visual content of a scene [118, 35, 87]. My scene segmentations segmentations allow for the possibility of object-centric interactive scene viewers. I have created a simple interactive viewer based on an overhead map or floor plan, shown in Figure 4.13.

To create the interactive floor plan, I first segment the scene using the algorithm from Section 4.3. I then manually align the scene points with an overhead view (though this step could possibly be automated with an approach similar to Kaminsky et al. [67]. Since I only want to include segments that can be localized at a reasonably-sized spot on the map, I remove all segments larger than a size threshold. To choose the representative image for each segment, I compute the Kullback-Leibler divergence [74] between the distribution of scene points in each image and each cluster.

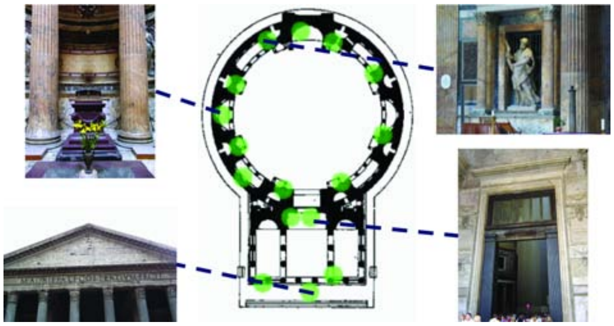


Figure 4.13: An interactive floor plan viewer showing the Pantheon (center). By moving the mouse over one of the highlighted circles, the user can see an image of the object at that location (sample images shown on left and right).

4.6 Discussion

In this chapter I have proposed a new field-of-view cue that can be used to extract objects from static 3D scenes, along with a probabilistic model that takes advantage of this cue and several applications of the method. It is important to stress that the probabilistic model is intended mainly to evaluate the usefulness of field-of-view cues, and not to provide a complete solution to the scene segmentation problem. Note in particular that I am not incorporating any of the more standard image segmentation cues such as intensity, color, contour, or region information, except through feature matching for estimating point correspondences. Including such additional terms would likely further improve upon these results. Still, my model that takes into account only field-of-view and spatial proximity is able to achieve segmentations that are good enough to enable a variety of applications.

Chapter 5

TOURIST FLOW: TRACKING PHOTOGRAPHERS USING FLICKR

As discussed in the preceding chapters, community photo collections, such as those found on Flickr, are rich sources of information about human perception, as they capture what people find interesting and worthy of photographing. In addition, these photo collections are a source of information about human behavior: where people go, what they view, and how they move from one place to another. Indeed a number of researchers have begun to explore GPS-tagged photos as a means of answering such questions [66, 25, 100, 101]. While these results are of great interest, the inherent limitations in GPS accuracy (a few meters) and coverage (outdoors only), limit their applicability for the most part to macro-scale analysis, at the level of neighborhoods, cities, countries, or patterns over the globe. Lacking, however, are effective tools for capturing and analyzing motion of people at *scene-level*, e.g., for a tourist site such as a plaza or building interior. One possible solution is tracking humans in video. However, installing video cameras at these sites is often impractical. Many webcams are already present at tourist sites, but these are typically low-resolution and updated only every several seconds, making it difficult to perform vision-based tracking of individuals.

To sidestep these issues, I propose mining photos on Flickr to derive human motion patterns for the world's tourist sites. My approach is based purely on photographs, as I do not use GPS or rely on video to track people, and hence can leverage the vast stores of publicly available imagery on the Internet. These photos are usually timestamped and we can compute the location at which each photo was taken using modern structure-from motion algorithms as described in Appendix A, giving us precise spacetime coordinates for each photo. Even though a single photographer typically provides only a few photos, often separated by minutes, by aggregating data over hundreds of photographers it is possible to get a more detailed picture of how people explore such sites, in the form of flow fields and other representations and visualizations.

I emphasize, however, that the goal in this chapter is not to do a rigorous study of human behavior, but rather to introduce computational and visualization tools for estimating and analyzing motion patterns from unstructured photo collections. As such, I focus on the algorithms, displaying the results, and mentioning a few of the most obvious observations, leaving a detailed, sociological analysis of the derived flows and human motion patterns to future research. I also emphasize that the motion patterns that I produce are derived from photos, and thus capture useful data only where people take photos. Hence, these approaches are best suited for popular tourist sites, and less applicable for many other scenes.

5.1 Related work

Uncovering human movement patterns through small scenes via tourist photographs requires estimating the position from which each photo was taken, then accumulating sequences of these positions into more general movement patterns. To estimate the camera positions, we use image matching and structure-from-motion (as GPS is too coarse). Previous work by Hays and Efros [55] proposed a more general vision-based solution to the geolocation problem, computing multiple types of image features and using a nearest-neighbor classifier with a database of geotagged images. Hays's approach, while coarser than direct matching and reconstruction methods, has the advantage of providing some information about the distribution of possible locations for an unmatchable image (of a beach or forest, for instance).

More accurate geolocation can be done if image sequences are considered. Kalogerakis et al. [66] geolocate image sequences using global human travel priors. Given a sequence of images by the same photographer, they use a hidden Markov model variant with a time-varying distribution over travel distance along with an image appearance model to geolocate each image in the sequence. Our goal is different, in that our photos are already geolocated and we are more interested in inferring human behavior patterns. Crandall et al. [25] use photos taken in close temporal proximity to aid in landmark identification, and also to visualize linear human pathways in a geospatial region. However, no further reasoning is done on these pathways.

Snavely et al. [117] describe a technique for computing paths through the photos of a scene, for the purpose of smoothly rendering a camera move from one position to another in a 3D photo browser. This technique uses information about where people take photos, since it tries to use paths that stay close to actual photos, in essence letting Flickr users specify with their photos which parts of the scene can be traversed. Actual sequential and temporal information about the photos is not used.

Other work has modeled human travel on a global scale using various data sources, such as currency [16], cell phone towers [52], and geotagged photos [48], though none of these approaches are applicable at the scene level.

5.2 Online photo sequences

I obtain the datasets needed for our technique from the photo sharing site Flickr. After downloading the images, we run the Bundler reconstruction pipeline of Snavely et al. [118] or Agarwal et al. [8] for larger scenes (see Appendix A). I also download photographer and timestamp metadata from Flickr. Note that, like Kalogerakis et al. [66], these timestamps need not be consistent across photographers. For each photographer, I operate predominantly on pairs of consecutive photos, removing pairs separated by more than an hour, as well as pairs which would require movement at an improbable speed.

I analyze these reconstructed photo sequences in several ways, addressing the following set of questions:

- At a particular location, which direction is a person most likely to move? How strong is this preference?
- Which parts of the scene are traversed most frequently, and in which direction?
- Is there a dominant order in which people visit areas of a scene?

5.3 Tourist flow

My objective in the *tourist flow* problem is to compute, for each scene location, the most likely direction that a person will move from that position. That is, the goal is to compute a flow field over the scene. More generally, it may be desirable to compute the *probability distribution over possible directions of movement* from each point in the scene.

As I have no information except at reconstructed camera locations, I simplify this objective to computing this distribution only at the cameras. Specifically, given a reconstructed set of photos, I construct a scene graph G with a node at each reconstructed camera location and edges between nearby cameras. The goal is then to compute, for each camera, a distribution over the nearby cameras that indicates likelihood of movement to each nearby camera's location. This discretized version of the problem treats the reconstructed cameras as states in a Markov chain, where photographers can move about the scene by transitioning between nearby cameras.

For each human photographer H_i, I have a sequence S_i of camera locations c_{ij} and times t_{ij} at which the photographer took a photo. Suppose you were given continuous GPS traces of the individuals as input — then recovering the flow field would be a simple matter of tabulating the histogram of observed motions at each point in the scene. Complicating the problem in our case (with no GPS) is that the instantaneous motion of each photographer is not provided, but rather the cumulative motion that the individual undergoes between photos. Computing flow therefore requires estimating each photographer's instantaneous motion, i.e., their trajectory between photos. One simple approach is to use the shortest path (geodesic) on the camera graph between photos. This approach, while only an approximation of the photographer's true motion, takes into account where people stand when visiting a scene. I weight each edge in G with the squared Euclidean distance between the two locations. (I use squared distance as non-squared distance tends to choose paths with too few edges. Note that using squared distance is also equivalent to a Gaussian probability distribution over possible next locations from each node.) While the shortest path is not necessarily accurate, it is still useful for addressing some of the questions we ask about movement patterns. In addition, it is possible to partially test the accuracy of this assumption by using triples of consecutive photos by the same photographer. Does the middle photo lie on the shortest path between the outer two? Figure 5.1 shows the results of such an experiment, where we make the additional assumption of constant travel speed. Shortest path prediction is slightly more accurate than inferring a straight-line path, and both are far more accurate than simply predicting the medoid of the entire set of reconstructed camera locations, confirming that behavior between photos is at least somewhat predictable. Once I have estimated the shortest path P_{ij} through each adjacent pair of locations c_{ij} and c_{ij+1} in each photo sequence S_i, I compute the transition distribution from each node $u \in V(G)$ as:

$$p(u \to v) = \frac{|P : (u, v) \in P|}{\sum_{w \in V(G)} |P : (u, w) \in P|} \tag{5.1}$$

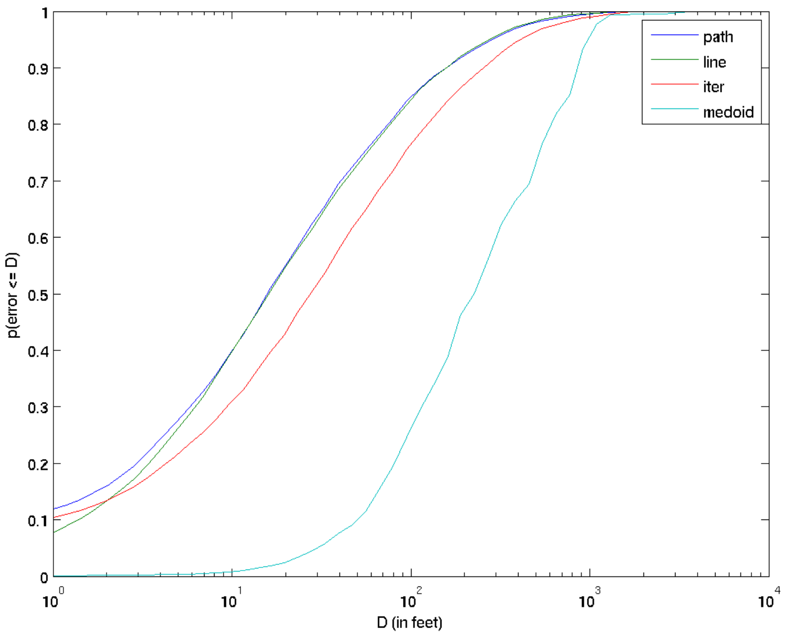


Figure 5.1: Cumulative histogram of error in predicting the location of the intermediate camera from a triple of consecutive photos, for four prediction methods. Shortest path prediction (path) uses the most likely path through other known camera positions. Linear prediction (line) uses a straight line between the first and third camera position. Iterated most likely path (iter) uses a path computed after an alternating optimization over paths and edge transition probabilities. The baseline predictor (medoid) predicts the medoid of all camera positions.

In other words, the probability of moving from node u to node v in one step is the number of paths that contained this edge divided by the number of paths that contained any edge out of u.

An alternative solution to this type of missing data problem involves alternating optimization such as the EM algorithm [28]. I experimented with such approaches with disappointing results. In particular, one can repeat the shortest path (most likely path) computation on the estimated transition distribution, and then re-estimate the transition distribution, until convergence. I find that this procedure results in an extremely low-entropy distribution that is not reflective of actual human behavior, and is in fact a worse predictor of intermediate camera locations than the initial shortest

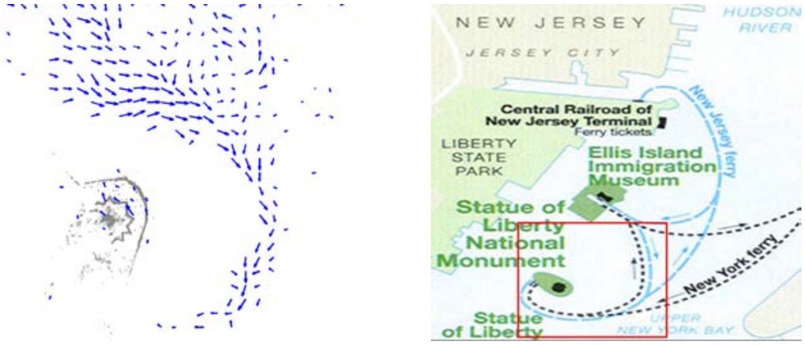


Figure 5.2: **(left)** Tourist flow field for the Statue of Liberty. Each arrow shows the expected direction of movement among nearby camera positions. Shorter arrows indicate uncertainty across cameras. **(right)** A map showing boat routes to and from the Statue of Liberty. The red box corresponds roughly to the region shown in the flow field (rotated $90°$).

paths. The main cause of this is the large space of transition distributions — it is easy to find a maze-like low-entropy distribution that happens to fit the observed pairs. Another way to avoid this problem is via regularization (e.g., encouraging the transition distribution for nearby cameras to be similar), but using the initial shortest paths achieves the same effect and is much simpler.

For the purpose of visualization, I compute flow at equally spaced points on a grid, where each flow vector is computed from the expected movement direction of the nearby cameras. Figure 5.2 shows such a flow field for the Statue of Liberty, along with a map[1] showing the paths taken by boats to and from Liberty Island. Not surprisingly, it appears that the dominant behavior is to take photos of the statue while approaching the island, rather than on the return trip. I show results for more scenes later in this paper.

Note that the expected movement direction may be misleading, as the distribution could be more uniform in some areas and more peaked in others. To explore this, I compute the *directionality* between each adjacent pair of camera locations in G — how likely it is that each edge will be taken in its preferred direction, under a uniform prior:

$$\mathrm{dir}(u, v) = \frac{\max(|P : (u, v) \in P|, |P : (v, u) \in P|) + 1}{|P : (u, v) \in P| + |P : (v, u) \in P| + 2} \tag{5.2}$$

[1] From http://www.hudsonlights.com.

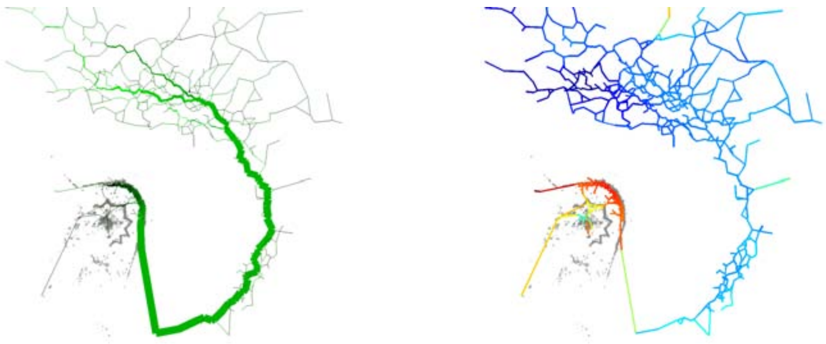


Figure 5.3: **(left)** Edge directionality for the Statue of Liberty. Each edge connects two nearby reconstructed cameras. Edge thickness correponds to the number of tourists crossing the edge in either direction (as estimated via shortest paths). Edge color corrsponds to the probability that the edge will be traversed in its more common direction, where black edges are equally likely to be crossed in either direction and green edges are more likely to be crossed in one direction only. **(right)** Spatial ordering for the Statue of Liberty. Each edge is colored according to the average rank of the two cameras it connects, where rank is an estimate of the order in which a person is likely to visit a part of the scene. Blue regions are visited earlier, and red regions are visited later.

Figure 5.3 plots directionality for the Statue of Liberty dataset, where strong directionality (indicated by green color) can be seen on the boat paths along the water, but much less on the island itself. Figure 5.3 also emphasizes (with thickness) edges that are traversed frequently. Also observe that boats begin approaching the statue across a relatively wide area, then converge to a much narrower path as they get closer. I show directionality visualizations for many other scenes later in this paper. I can also compute the directionality of an entire scene as a sum of the edge directionalities weighted by the number of traversals. Scene directionalities are shown in the last column of Table 5.1.

5.4 Scene ordering

Some scenes have a natural order of traversal, e.g., tourists visiting the Statue of Liberty tend to take photos from the boat as they approach *before* taking photos on the island. Other scenes have partial orders, or no dominant order at all. How can one represent, compute, and display this scene ordering? Given a set of ordered image pairs for each photographer H_i, I fit a function f over camera positions c that attempts to achieve $f(c_{ij}) < f(c_{ij+1})$. That is, I try to ensure that each pair of consecutive photos taken by the same person are ordered correctly by f.

To construct this ordering, I use a ranking SVM [65]. I minimize the standard SVM hinge loss error:

$$\frac{1}{2} w \cdot w + \lambda \sum_{ij} \xi_{ij} \tag{5.3}$$

where w is the feature weight vector, ξ_{ij} are the slack variables, and λ is the margin violation penalty, which controls the tradeoff between maximizing the margin and satisfying the constraints. I use the slightly modified margin constraint on the rank differences:

$$w \cdot (\phi(c_{ij+1}) - \phi(c_{ij})) + \xi_{ij} \geq t_{ij+1} - t_{ij} \tag{5.4}$$

where ϕ is the feature function (in our case defined implicitly by a kernel). Instead of a fixed margin, I vary the margin according to the time difference between each pair of consecutive photos, encouraging photos taken far apart in time to be far apart in the ordering. This modification appears to have only a minor effect on the results, as ordinary ranking SVMs as well as linear regression performed similarly in our experiments. I use a Gaussian kernel over camera locations and solve the optimization in the dual. A visualization of the ordering function computed for the Statue of Liberty is shown in Figure 5.3, and seems to confirm the overall behavior I inferred from our previous approaches — people prefer to take photos during the approach to the statue by boat, rather than on the way back. Orderings for other scenes are shown in Section 5.6.

It is still necessary to check if the orderings computed by the ranking SVM are meaningful, or if it's fitting an ordering to idiosyncrasies in the data set that are not indicative of how tourists behave at the scene. To do this, for each scene I run the ranking SVM on half of the photographers to compute an ordering, and test on the other half. If there is a true ordering, I should be able to predict the order of pairs of images by photographers in the held-out set. The first column of Table 5.1 shows the test set prediction rate for many of the reconstructed scenes. Since many camera pairs may be difficult to compare, I also evaluate the computed ordering on the half of the test set for which the ranking SVM is most certain. The prediction rates for this experiment are in the second column of Table 5.1.

The orderings for all scenes have an accuracy rate between 45% and 65%, however when I only consider the camera pairs on which our SVM is most certain, accuracy improves dramatically for a

	acc	**acc***	**dir**		**acc**	**acc***	**dir**
Statue of Liberty	0.64	0.63	0.76	San Gimignano	0.53	0.54	0.57
St. Peter's Basilica	0.63	0.75	0.67	Old Town Square	0.54	0.52	0.56
Pantheon	0.57	0.63	0.63	Colosseum	0.47	0.47	0.55
Roman Forum	0.57	0.56	0.60	Cesky Krumlov	0.48	0.51	0.54
Trafalgar Square	0.55	0.56	0.58	Dubrovnik	0.59	0.69	0.54
Piazza Navona	0.49	0.64	0.57				

Table 5.1: Accuracy of photo order prediction. The first column shows the accuracy rate of using our ranking SVM to predict the order of two consecutive photos by the same held-out user. The second column is the same, but only considers half the pairs, the ones for which the ranking SVM is most certain. The third column contains the total edge directionality for each scene, as described in Section 5.3.

few scenes, such as St. Peter's (0.63 to 0.75) and Dubrovnik (0.59 to 0.69). This suggests that while most scenes do not have a strong ordering, my approach is able to identify preferences in tourist behavior when they exist. These scores may also indicate that many scenes have directionality only in some areas, a conclusion which is also supported by the directionality visualizations in Section 5.6.

5.5 Patterns of photography

The last two sections tried to infer human motion patterns using reconstructed camera positions. Alternatively, I can look for patterns in the sequences of photos themselves, based on the content of each photograph. I consider the following type of questions:

- Are there views which are commonly photographed immediately before or after other views? That is, are there common view transitions?

- Do people photograph views in order of popularity, or some other order?

- Do people take multiple consecutive photos of the same or similar views?

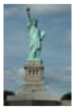
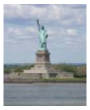
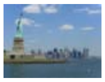
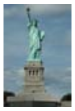
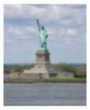
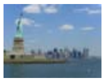
			
	223 (0.81)	29 (0.08)	23 (0.11)
	49 (0.31)	90 (0.57)	18 (0.11)
	21 (0.26)	10 (0.12)	50 (0.62)

Figure 5.4: Canonical view transition counts for the Statue of Liberty. Each entry contains the number of consecutive image pairs where the first image matches the row canonical view and the second image matches the column canonical view. Transition probabilities are in parentheses. Observe that when the first two canonical views are photographed consecutively, the second canonical view is usually photographed first.

To answer these questions, I extract a small set of canonical views for each scene. These are views which are taken by many photographers. I select the canonical views using the algorithm discussed in Chapter 3. In addition to selecting canonical views, this algorithm creates a clustering in which each photo is associated with the canonical view to which it is most similar.

I then examine the transition count matrix T between photos assigned to each of the canonical views. This matrix is easy to compute, since the canonical views algorithm in Chapter 3 also assigns each photo to its nearest canonical view. The transition count matrix for the top three canonical views of the Statue of Liberty is shown in Figure 5.4. Several things can be learned from this matrix. First, taking repeated photos of the same view (and uploading all of them to Flickr) is very common. Second, even though the dominant canonical view is photographed most frequently, when

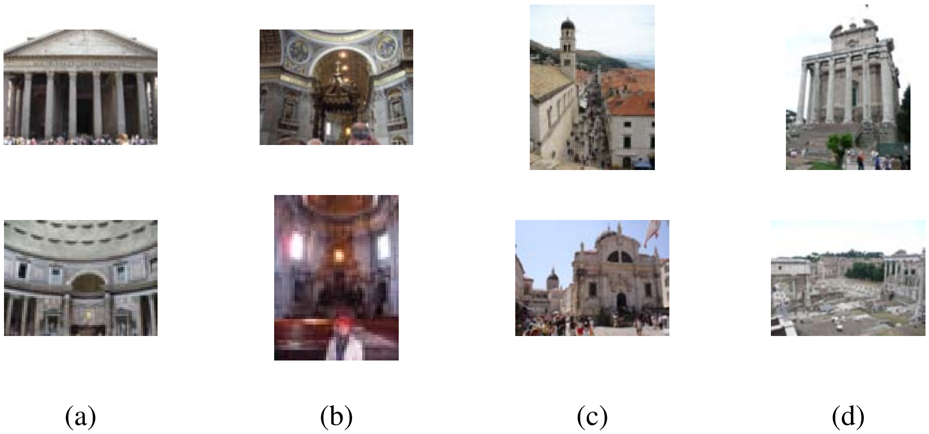


Figure 5.5: Asymmetric view pairs for **(a)** the Pantheon, **(b)** St. Peter's Basilica, **(c)** the city of Dubrovnik, and **(d)** the Roman Forum. Of the top 5 canonical views for each scene, these are the pairs which are most likely to appear in order (top before bottom).

the top two canonical views are photographed consecutively the second canonical view is usually taken first. This is not surprising given the observations about the Statue of Liberty from previous sections, as the second view is from much farther away and therefore earlier on the approach (though all three are taken from the water). In general, asymmetries in this matrix can identify pairs of views that are commonly photographed in one order rather than the other. Such asymmetric pairs for a few scenes are shown in Figure 5.5.

It is also possible to use a technique similar to the one in Section 5.4 to try to fit a global ordering to the views of a scene. Instead of fitting a ranking SVM based on camera position, it can be based on the visual content of the photographs. I use as the feature set (not the kernel) the similarity of each image to a set of 10 canonical views, as computed in Chapter 3. I tried other feature spaces such as the incidence of scene points in an image, but found that this led to severe overfitting. This is not surprising, as the number of scene points is much larger than the number of images. Figure 5.6 shows five canonical views from St. Peter's basilica both in the order they are chosen by the canonical views algorithm and in the order they are placed by the ranking SVM. I also evaluated the accuracy of the image-based SVM orderings using a training and test set (as in the spatial orderings), and I find that the spatial orderings from Section 5.4 are slightly more accurate, even for scenes like

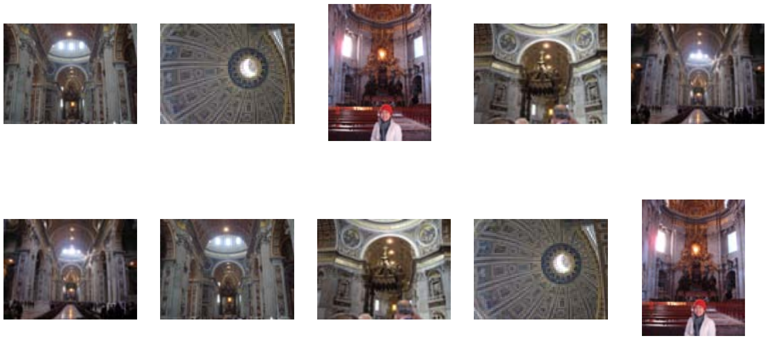


Figure 5.6: Unsorted and sorted canonical views for St. Peter's Basilica. **(top)** Canonical views in the order computed by the greedy k-means algorithm described in Chapter 3. **(bottom)** Canonical views ordered by the temporal ranking function. The sorted set clearly captures a linear traversal from the front entry of the cathedral to the back (gold throne in last image), glancing up at the dome above the altar on the way. Note that these images were taken by different people, and cannot be trivially ordered by timestamp.

the Statue of Liberty with a strong ordering preference. One possible explanation is that views of the same object from different distances look much the same, and it is the relative distances that determine the ordering.

5.6 Results

In this section, I show and interpret several visualizations for tourist movement at several of the 16 scenes used in this thesis. Figures 5.7 and 5.8 show three visualizations for several scenes. The left column of both figures shows the tourist flow, i.e., the expected direction of movement at each part of the scene. The center column shows the volume of tourists moving between nearby camera locations, as well as the directionality of each camera pair (how likely it is that tourists move in the more common direction). The right column colors edges between nearby cameras based on a global ordering of scene locations. The behavior at some of the scenes, such as the Statue of Liberty is relatively straightforward to interpret. For others, it may be difficult, especially if you've never been there. I provide a brief description of the movement patterns at a few scenes:

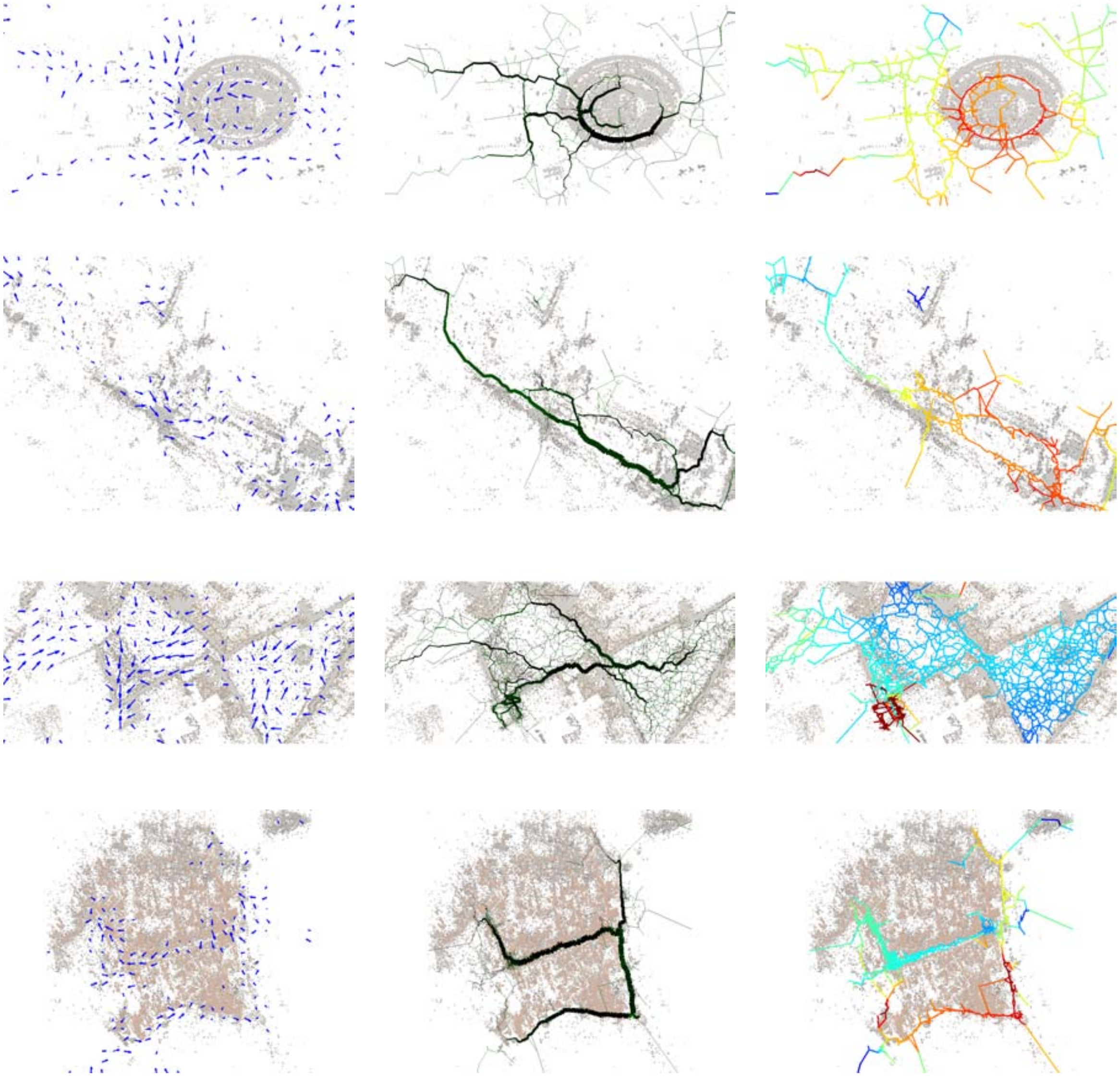


Figure 5.7: Flow, directionality, and ordering for **(top row)** the Colosseum, **(second row)** the Roman Forum, **(third row)** part of the town of San Gimignano, and **(bottom row)** the city of Dubrovnik.

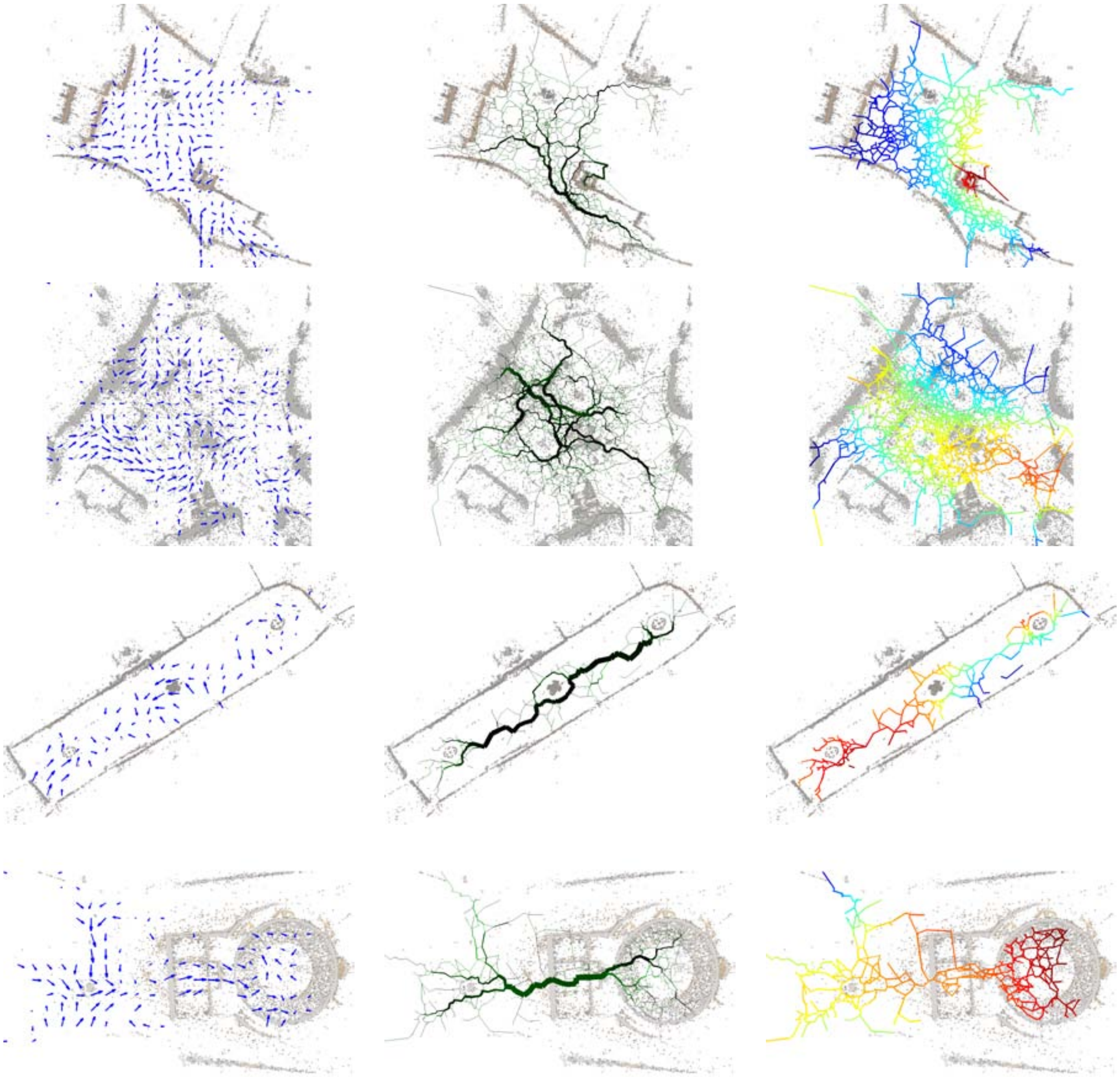


Figure 5.8: Flow, directionality, and ordering for **(top row)** Old Town Square in Prague, **(second row)** Trafalgar Square, **(third row)** Piazza Navona, and **(bottom row)** the Pantheon.

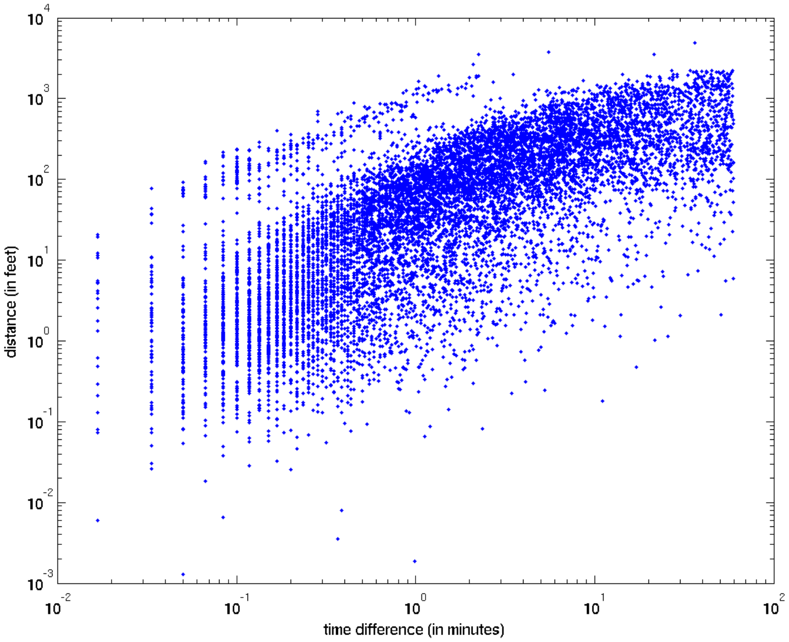


Figure 5.9: Scatter plot of time and distance between consecutive photos, over all scenes mentioned in this chapter.

St. Peter's Basilica — (See Figure 1.2.) The dominant behavior at St. Peter's is to walk from the entrance on the right to the altar at the center of the cross. There are interesting artworks to see in each of the three other directions from the altar, but a common final step is to climb up to the base of the dome.

Old Town Square — Here, movement about the square appears random, but a common final step is to climb up the clock tower.

San Gimignano — San Gimignano has many towers, though this reconstruction only includes one, and again, the final step in our ordering is to climb the tower.

Roman Forum — Tourists appear to enter the Forum from the east (upper left in our figures), which means they are probably coming from the Colosseum. Later pictures tend to be on the western side of the Forum, near the Curia and Arch of Septimus Severus.

I also believe online photos are a useful data source for describing more general human movement patterns, in addition to scene-specific patterns. The work of Kalogerakis et al. [66] or an extension thereof may be able to provide this on a global scale, while my approach is more useful at finer resolutions.

One interesting question is whether movement patterns derived from shared photos agree with previous studies on human behavior. There is some agreement [52, 16] that human movement follows a Lévy flight model, though this may not be detectable at finer scales. Figure 5.9 shows a scatter plot of time and spatial distance between consecutive photos over all scenes in this chapter. Note that the vast majority of photo pairs were taken within 5 minutes and 500 feet of each other. The average speed was 0.4 miles per hour, well below the average human walking speed of 3 miles per hour. The pairs that seem to indicate travel at faster than human walking speed are from the Statue of Liberty, where many of the photographs were taken from moving boats.

Another interesting question involves the relationship between photos and movement. Walt Disney is said to have designed his theme parks around a concept he called “weenies” [7] — at any location, there should be a view of a large visual icon that entices tourists to approach it. Can a similar effect be observed at other tourist sites? More precisely, do tourists tend to move in the direction of what they photograph? Figure 5.10 shows a histogram of the angle between camera viewing direction and human movement direction, and suggests that such an effect is indeed present. The effect is also considerably more powerful for the dominant canonical view for each scene. (This analysis does not use the inferred paths from Section 5.3, but instead just takes the vector between two consecutive camera locations as the movement direction.)

5.7 Discussion

The main contribution of this chapter is the use of online photo collections for tracking human movement through tourist sites. My analysis yields insight into how humans move and take photos while exploring these sites. While some of the observations may appear obvious to people familiar with the sites, an advantage of our approach is that these patterns can be extracted automatically from data that already exists and is freely available. There are several extensions to this work that I believe are worth exploring. In this chapter I analyze each scene independently, except for a few simple statistics I compute across data sets. There may be consistent patterns of behavior that hold

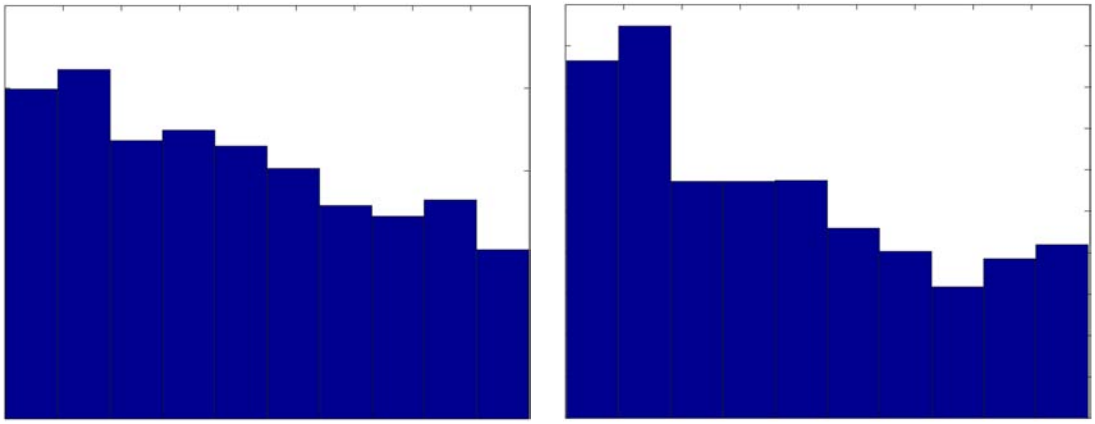


Figure 5.10: Histogram of angle between viewing direction and movement direction for **(left)** all views and **(right)** views similar to the dominant canonical view for each scene. The leftmost bin in each histogram corresponds to movement in the camera's viewing direction, and the rightmost bin corresponds to movement opposite the viewing direction. Note the correlation between viewing direction and movement, a correlation that is even stronger for the dominant canonical view.

across scenes. For example, in three of the scenes with a high vantage point looking over the scene, photos taken from this vantage point were taken near the end of a person's visit to the scene. It would be useful to automatically detect such behaviors. My analysis also didn't take into account the behavior of the same person across multiple scenes, and this is a possible direction for further exploration as well.

Chapter 6

CONCLUSION

In this thesis I explored the idea of using community photo collections to infer a human-centric perception of the world, through computer vision and statistical techniques. In particular, I made the following contributions:

- I introduced the idea of using online photo collections as a window into human perception. While previous work by Naaman [87] nominally used photo collections for this purpose, the photos were used merely as GPS markers — the visual content of the photos was not used. In my work, I take advantage of modern computer vision matching and reconstruction techniques to extract a kind of collective perception of concepts like canonical views and objects which are not discoverable by other methods.

- I described a technique for discovering canonical views of 3D scenes and using them to provide a simple summary-based visualization of the scene content. The key observation is that the distribution of photos taken by people is highly non-uniform, with certain views appearing far more frequently than others, as predicted by the original canonical views work by Palmer et al. [96]. I showed that these canonical views can be computed robustly and efficiently from large photo collections. In addition, I described a simple approach for annotating these views by using statistical techniques to choose from among the noisy user-submitted textual tags on Flickr.

- To select canonical views, I introduced to the vision community a greedy exemplar-finding algorithm used by Cornuejols et al. [24] in a different context. This algorithm is guaranteed to achieve an objective value within a constant factor of the optimum. In practice, the algorithm outperforms affinity propagation, the state-of-the-art algorithm, on data sets arising from visually-matched community photo collections.

- As another instance of using photo collections as a window into human perception, I discovered a useful new cue for identifying objects: the field-of-view cue, which states that photos are likely to be framed around an object of interest. I designed a simple scene model and associated inference algorithm to exploit this cue to produce 3D scene and image segmentations.

- I acquired data consisting of human-drawn bounding boxes around objects in photos taken at various tourist sites, and used this data to evaluate the manner in which people frame their photos with respect to objects.

- I introduced the idea of using 3D reconstructions of Internet photo collections to track the movement of people (photographers) at scales and in environments for which GPS and other approaches are not feasible. I described a technique for extracting common patterns of movement in a scene, and showed many ways in which these patterns can be visualized and interpreted.

6.1 Future work

The ideas presented in this thesis suggest future work in various directions, touching computer vision, psychology and perception, human-computer interaction, and computer-supported collaborative work. Also, future advances in other areas outside the scope of this thesis have the capability of greatly enhancing some of the methods I have presented. For instance, new developments in highly scalable image matching and 3D reconstruction techniques stand to increase by orders of magnitude the size of the image collections on which my techniques can be applied. And since my approach depends on an accurate estimate of the distribution of photos, increasing the size of the photo collections on which I operate can be an effective technique for combating noise, as well as reaching further out into the long tail of interesting content. Of course, my own methods must also be made to scale to photo collections of this size. Currently, the algorithms presented in this thesis have been run on city-scale image sets with hundreds of thousands of images and take minutes on a single CPU. However, one can imagine performing similar analyses on data sets of billions of images, which is likely to require algorithms that have been designed to explicitly support parallelism.

In the remainder of this section I describe future work that expands on and extends the ideas in this thesis.

6.2 Discovering new properties of large photo collections

My work in this thesis takes advantage of certain patterns present in the set of photos people take — people tend to take photos from canonical views, and with an object contained entirely within the field of view, and tend to take photos in predictable order. A promising line of work is to discover other such patterns, or adapt other principles from the psychology community (for instance, canonical object sizes [71]), and design new algorithms that exploit these patterns for computer vision or visualization tasks. Recognition- and segmentation-like problems seem most amenable to this type of approach, as the information gleaned from such patterns may provide an alternative to directly computing the interaction between a large number of low-level cues, as current approaches to these problems must.

One important aspect of tourist photos that I have ignored in this thesis is that they often contain people. By reasoning about the presence or absence of people in photographs, as well as their location with respect to the photographer and static objects in the 3D scene, it may be possible to discover new patterns and enable new applications. For instance, through a combination of face and clothing recognition, 3D reconstruction, and analysis of timestamps and Flickr IDs, one could potentially identify the same person in multiple photographs from an event [47] and, if that person also took photographs, discover his/her Flickr ID. Another interesting question is the difference between the distribution of photos containing people and the distribution of photos not containing people. Are these two view distributions different? Do the cues I exploit in this thesis work equally well for both?

More generally, can we take advantage of other advances in computer vision? All the work in this thesis depends upon advances in rigid matching [80] and reconstruction [118]. As computer vision matures, more robust algorithms for problems like recognizing object categories or identifying weather conditions may be usable in this context. Taking this further, we can consider a two-pronged approach towards scene understanding, where low-level techniques provide more structured data which can be mined for patterns in human photography and behavior, which in turn can be used as a prior for solving other computer vision problems.

6.3 Soliciting information from users and photographers

All of the data used in this thesis is produced through the ordinary behavior of users of photo-sharing sites. Photographers who upload their photos to Flickr and tag them are not motivated by solving computer vision problems, but by more personal and social reasons. (Much research [51, 82, 114, 113] has been done on the motivations behind tagging and photo-sharing.) However, there exist other user contribution paradigms, as exemplified by LabelMe [105], the ESP Game [126], Amazon Mechanical Turk [1], and the PhotoCity project [125]. Each of these is capable of providing data that is useful for scene understanding, but each also raises interesting issues. For instance, photos uploaded to PhotoCity are likely to be less useful in computing canonical views and scene segmentations, as particular views are rewarded by the game, presumably altering the photo-taking behavior of its users.

In Chapter 4, I described a field-of-view cue for segmenting objects in a 3D scene, and also solicited bounding box segmentations from Mechanical Turk users for comparison. A better approach might be some combination of user contribution and automatic techniques for object extraction and labeling, to minimize the amount of human work required while still achieving high quality results. I have experimented with a simple implementation of a "3D LabelMe" that converges to a precise 3D segmentation with user-drawn object outlines, but more work is needed to develop this concept into a fully-functional system, and future study is required to discover the optimal boundary between human and algorithmic contribution.

6.4 New applications and visualizations

Another area of future work is expanding the set of applications that are enabled by leveraging the distribution of online photos. One application that has been proposed recently by Choudhury et al. [26] is the construction of travel itineraries from Flickr photo collections. Though this doesn't use the visual content of the photos, it is closer to a "real" application than what I have described in this thesis — it directly solves a real-world problem that people have. Another example is the construction of landmark-based walking directions by Hile et al. [57], which does use the photo content. Are there other real-world problems for which online photo distributions provide a useful avenue of attack?

There are also questions regarding the evaluation of different 3D scene visualizations that should be addressed. What properties does a good scene summary have? When is a scene summary preferable to a virtual tour? What is the right way to combine summaries and virtual tours (one example is Snavely et al. [117])? Are virtual tours that reflect actual human movement patterns as discussed in Chapter 5 preferable to other tours? Answering these questions probably requires setting up a number of user studies, but the knowledge gained could be valuable, as many different scene visualizations have been proposed in recent years but very little human evaluation has been done.

BIBLIOGRAPHY

[1] Amazon Mechanical Turk. *http://www.mturk.com/*.

[2] Flickr. *http://www.flickr.com/*.

[3] Google Images. *http://images.google.com/*.

[4] Google Maps. *http://maps.google.com/*.

[5] Graphviz. *http://graphviz.org/*.

[6] Wikipedia. *http://www.wikipedia.org/*.

[7] Wikipedia: Cinderella Castle. *http://en.wikipedia.org/wiki/Cinderella_Castle*.

[8] Sameer Agarwal, Noah Snavely, Ian Simon, Steven M. Seitz, and Richard Szeliski. Building Rome in a Day. *Proceedings of the IEEE 12th International Conference on Computer Vision*, pages 72–79, September 2009.

[9] Shane Ahern, Mor Naaman, Rahul Nair, and Jeannie Yang. World Explorer: Visualizing aggregate data from unstructured text in geo-referenced collections. In *Proceedings of the 7th ACM/IEEE-CS Joint Conference on Digital Libraries*. ACM, 2007.

[10] Antonis A. Argyros and Manolis Lourakis. The design and implementation of a generic sparse bundle adjustment software package based on the Levenberg-Marquardt algorithm. *Technical Report: Institute of Computer Science-FORTH, Heraklion, Crete, Greece*, 2004.

[11] Kobus Barnard, Pinar Duygulu, David A. Forsyth, Nando de Freitas, David M. Blei, and Michael I. Jordan. Matching words and pictures. *Journal of Machine Learning Research*, 3(6):1107–1135, 2003.

[12] Tamara L. Berg, Alexander C. Berg, Jaety Edwards, Michael Maire, Ryan White, Yee-Whye Teh, Erik Learned-Miller, and David A. Forsyth. Names and faces in the news. *Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2:848–854, 2004.

[13] Volker Blanz, Michael J. Tarr, and Heinrich H. Bülthoff. What object attributes determine canonical views? *Perception*, 28(5):575–600, 1999.

[14] David M. Blei and Michael I. Jordan. Modeling annotated data. *Proceedings of the Twenty-Sixth Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 127–134, 2003.

[15] David M. Blei, Andrew Y. Ng, and Michael I. Jordan. Latent Dirichlet allocation. *Journal of Machine Learning Research*, 3(4-5):993–1022, May 2003.

[16] Dirk Brockmann, Lars Hufnagel, and Theo Geisel. The scaling laws of human travel. *Nature*, 439(7075):462–465, 2006.

[17] Heinrich H. Bülthoff and Shimon Edelman. Psychophysical support for a two-dimensional view interpolation theory of object recognition. *Proceedings of the National Academy of Sciences of the United States of America*, 89(1):60, 1992.

[18] Neill Campbell, George Vogiatzis, Carlos Hernandez, and Roberto Cipolla. Automatic 3D Object Segmentation in Multiple Views using Volumetric Graph-Cuts. *Image and Vision Computing*, 28(1):14–25, 2007.

[19] John F. Canny. A computational approach to edge detection. *Readings in computer vision: issues, problems, principles, and paradigms*, 1987.

[20] Chad Carson, Megan Thomas, Serge Belongie, Joseph Hellerstein, and Jitendra Malik. Blobworld: a System for Region-based Image Indexing and Retrieval. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 1999.

[21] Ondrej Chum, James Philbin, Josef Sivic, Michael Isard, and Andrew Zisserman. Total Recall: Automatic query expansion with a generative feature model for object retrieval. *Eleventh IEEE International Conference on Computer Vision*, 2007.

[22] Paul Clough, Hideo Joho, and Mark Sanderson. Automatically Organising Images using Concept Hierarchies. *Proceedings of the ACM SIGIR Workshop on Multimedia Information Retrieval*, 2005.

[23] Dorin Comaniciu and Peter Meer. Mean shift: a robust approach toward feature space analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(5):603–619, May 2002.

[24] Gerard Cornuejols, Marshall L. Fisher, and George L. Nemhauser. Location of Bank Accounts to Optimize Float: An Analytic Study of Exact and Approximate Algorithms. *Management Science*, 23(8):789–810, 1977.

[25] David Crandall, Lars Backstrom, Dan Huttenlocher, and Jon Kleinberg. Mapping the World's Photos. *Proceedings of the 18th International Conference on the World Wide Web*, 2009.

[26] Munmun De Choudhury, Moran Feldman, Sihem Amer-Yahia, Nadav Golbandi, Ronny Lempel, and Cong Yu. Automatic construction of travel itineraries using social breadcrumbs. *Proceedings of the 21st ACM Conference on Hypertext and Hypermedia*, pages 35–44, 2010.

[27] Scott Deerwester, Susan T. Dumais, George W. Furnas, Thomas K. Landauer, and Richard Harshman. Indexing by latent semantic analysis. *Journal of the American Society for Information Science*, 41(6):391–407, January 1999.

[28] Arthur P. Dempster, Nan M. Laird, and Donald B. Rubin. Maximum Likelihood from Incomplete Data via the EM Algorithm. *Journal of the Royal Statistical Society: Series B (Methodological)*, 39(1):1–38, 1977.

[29] Trip Denton, M. Fatih Demirci, Jeff Abrahamson, Ali Shokoufandeh, and Sven Dickinson. Selecting Canonical Views for View-Based 3-D Object Recognition. In *Proceedings of the 17th International Conference on Pattern Recognition*.

[30] Inderjit S. Dhillon and Dharmendra S. Modha. Concept Decompositions for Large Sparse Text Data Using Clustering. *Machine Learning*, 42(1):143–175, 2001.

[31] Delbert Dueck and Brendan J. Frey. Non-metric affinity propagation for unsupervised image categorization. *International Conference on Computer Vision*, 2007.

[32] Delbert Dueck, Brendan J. Frey, Nebojsa Jojic, Vladimir Jojic, Guri Giaever, Andrew Emili, Gabe Musso, and Robert Hegele. Constructing treatment portfolios using affinity propagation. *Research in Computational Molecular Biology*, pages 360–371, 2008.

[33] Pinar Duygulu, Kobus Barnard, Nando de Freitas, and David A. Forsyth. Object recognition as machine translation: Learning a lexicon for a fixed image vocabulary. *Proceedings of the Seventh European Conference on Computer Vision*, pages 97–112, 2002.

[34] Shimon Edelman and Heinrich H. Bülthoff. Orientation dependence in the recognition of familiar and novel views of three-dimensional objects. *Vision Research*, 32(12):2385–400, December 1992.

[35] Boris Epshtein, Eyal Ofek, Yonatan Wexler, and Pusheng Zhang. Hierarchical photo organization using geo-relevance. In *Proceedings of the 15th Annual ACM International Symposium on Advances in Geographic Information Systems*, Seattle, Washington, 2007. ACM.

[36] Andreas Feininger. *Principles of composition in photography*. American Photographic Book Publishing Company, 1972.

[37] Rob Fergus, Li Fei-Fei, Pietro Perona, and Andrew Zisserman. Learning Object Categories from Google's Image Search. *Tenth IEEE International Conference on Computer Vision*, 2:1816–1823, 2005.

[38] Rob Fergus, Pietro Perona, and Andrew Zisserman. Object class recognition by unsupervised scale-invariant learning. *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2003.

[39] Rob Fergus, Barun Singh, Aaron Hertzmann, Sam T. Roweis, and William T. Freeman. Removing camera shake from a single photograph. *ACM SIGGRAPH*, 1(212), 2006.

[40] George Field. *Chromatics; or, the analogy, harmony, and philosophy of colours*. Bogue, 1845.

[41] Martin A. Fischler and Robert C. Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24(6):381–395, June 1981.

[42] Fodor's. *See It Rome*. Fodor's, 2006.

[43] William T. Freeman. The generic viewpoint assumption in a framework for visual perception. *Nature*, 368(6471):542–545, April 1994.

[44] Brendan J. Frey and Delbert Dueck. Clustering by passing messages between data points. *Science*, 315(5814):972, 2007.

[45] Yasutaka Furukawa, Brian Curless, Steven M. Seitz, and Richard Szeliski. Reconstructing building interiors from images. *Proceedings of the International Conference on Computer Vision*, pages 80–87, 2009.

[46] Andrew C. Gallagher and Tsuhan Chen. Understanding images of groups of people. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 256–263, 2009.

[47] Rahul Garg. Personal communication, 2009.

[48] Fabien Girardin, Francesco Calabrese, Filippo Dal Fiore, Carlo Ratti, and Josep Blat. Digital footprinting: Uncovering tourists with user-generated content. *IEEE Pervasive Computing*, 7(4):36–43, 2008.

[49] Inmar E. Givoni and Brendan J. Frey. Semi-supervised affinity propagation with instance-level constraints. *Proceedings of the 12th International Conference on Artificial Intelligence and Statistics*, 2009.

[50] Michael Goesele, Noah Snavely, Brian Curless, Hugues Hoppe, and Steven M. Seitz. Multi-View Stereo for Community Photo Collections. *Proceedings of the Eleventh IEEE International Conference on Computer Vision.*

[51] Scott A. Golder and Bernardo A. Huberman. Usage patterns of collaborative tagging systems. *Journal of Information Science*, 32(2):198–208, April 2006.

[52] Marta C. González, César A. Hidalgo, and Albert-László Barabási. Understanding individual human mobility patterns. *Nature*, 453:779–782, 2008.

[53] Christopher D. Green. All that glitters: a review of psychological research on the aesthetics of the golden section. *Perception*, 24(8):937–68, January 1995.

[54] Peter Hall and Martin Owen. Simple canonical views. *Proceedings of the British Machine Vision Conference*, 1:7–16, 2005.

[55] James Hays and Alexei A. Efros. IM2GPS: estimating geographic information from a single image. *IEEE Conference on Computer Vision and Pattern Recognition*, 2008.

[56] James Hays and Alexei A. Efros. Scene completion using millions of photographs. *Communications of the ACM*, 51(10):87–94, 2008.

[57] Harlan Hile, Radek Grzeszczuk, Alan Liu, Ramakrishna Vedantham, Jana Košecka, and Gaetano Borriello. Landmark-based pedestrian navigation with enhanced spatial reasoning. *Pervasive Computing*, pages 59–76, 2009.

[58] Thomas Hofmann. Probabilistic latent semantic analysis. *Uncertainty in Artificial Intelligence*, 1999.

[59] Thomas Hofmann. The cluster-abstraction model: Unsupervised learning of topic hierarchies from text data. *Proceedings of the International Joint Conference on Artificial Intelligence*, 16:682–687, 1999.

[60] Derek Hoiem, Alexei A. Efros, and Martial Hebert. Putting objects in perspective. *International Journal of Computer Vision*, 80(1):3–15, 2008.

[61] Abigail Hole. *Best of Rome*. Lonely Planet Publications, 2006.

[62] Alexander Jaffe, Mor Naaman, Tamir Tassa, and Marc Davis. Generating summaries for large collections of geo-referenced photographs. *Proceedings of the International Conference on World Wide Web*, pages 853–854, 2006.

[63] Yushi Jing and Shumeet Baluja. Pagerank for product image search. *Proceedings of the 17th International Conference on World Wide Web*, pages 307–316, 2008.

[64] Yushi Jing, Shumeet Baluja, and Henry Rowley. Canonical image selection from the web. *Proceedings of the 6th ACM International Conference on Image and Video Retrieval*, pages 280–287, 2007.

[65] Thorsten Joachims. Optimizing search engines using clickthrough data. *Proceedings of the Eighth ACM International Conference on Knowledge Discovery and Data Mining*, pages 133–142, 2002.

[66] Evangelos Kalogerakis, Olga Vesselova, James Hays, Alexei A. Efros, and Aaron Hertzmann. Image sequence geolocation with human travel priors. *IEEE 12th International Conference on Computer Vision*, pages 253–260, 2010.

[67] Ryan Kaminsky, Noah Snavely, Steven M. Seitz, and Richard Szeliski. Alignment of 3D point clouds to overhead images. *IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*, pages 63–70, 2009.

[68] Biliana Kaneva, Josef Sivic, Antonio Torralba, Shai Avidan, and William T. Freeman. Infinite Images: Creating and Exploring a Large Photorealistic Virtual Space. *Proceedings of the IEEE*, 98(8):1391–1407, 2010.

[69] Lyndon Kennedy and Mor Naaman. Generating diverse and representative image search results for landmarks. *Proceeding of the 17th International Conference on World Wide Web*, pages 297–306, 2008.

[70] Lyndon Kennedy, Mor Naaman, Shane Ahern, Rahul Nair, and Tye Rattenbury. How flickr helps us make sense of the world: context and content in community-contributed media collections. *Proceedings of the 15th International Conference on Multimedia*, pages 631–640, 2007.

[71] Talia Konkle and Aude Oliva. Canonical visual sizes for real world objects. *Journal of Vision*, 9(8):815–815, April 2010.

[72] Michael Kubovy and Martin van den Berg. The whole is equal to the sum of its parts: a probabilistic model of grouping by proximity and similarity in regular patterns. *Psychological Review*, 115(1):131–154, January 2008.

[73] Michael Kubovy and Johan Wagemans. Grouping by proximity and multistability in dot lattices: a quantitative Gestalt theory. *Psychological Science*, 1995.

[74] Solomon Kullback and Richard A. Leibler. On information and sufficiency. *The Annals of Mathematical Statistics*, 1951.

[75] Jean-François Lalonde, Derek Hoiem, and Alexei A. Efros. Photo clip art. *ACM SIGGRAPH*, 2007.

[76] Li-Jia Li, Gang Wang, and Li Fei-Fei. OPTIMOL: automatic Online Picture collecTion via Incremental MOdel Learning. *International Journal of Computer Vision*, 2010.

[77] Xiaowei Li, Changchang Wu, Christopher Zach, Svetlana Lazebnik, and Jan-Michael Frahm. Modeling and recognition of landmark image collections using iconic scene graphs. *Proceedings of the 10th European Conference on Computer Vision*, pages 427–440, 2008.

[78] Yunpeng Li, David Crandall, and Dan Huttenlocher. Landmark classification in large-scale image collections. *IEEE 12th International Conference on Computer Vision*, pages 1957–1964, September 2009.

[79] Stuart P. Lloyd. Least squares quantization in PCM. *IEEE Transactions on Information Theory*, 28(2):129–137, March 1982.

[80] David G. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2):91–110, 2004.

[81] George Markowsky. Misconceptions about the Golden Ratio. *The College Mathematics Journal*, 23(1):2, January 1992.

[82] Cameron Marlow, Mor Naaman, Danah Boyd, and Marc Davis. HT06, tagging paper, taxonomy, Flickr, academic article, to read. *Proceedings of the Seventeenth Conference on Hypertext and Hypermedia*, pages 31–40, 2006.

[83] David Marr and Ellen C. Hildreth. Theory of edge detection. *Proceedings of the Royal Society of London: Series B, Biological Sciences*, 207:187–217, February 1980.

[84] David R. Martin, Charless Fowlkes, Doron Tal, and Jitendra Malik. A Database of Human Segmented Natural Images and its Application to Evaluating Segmentation Algorithms and Measuring Ecological Statistics. *Proceedings of the 8th International Conference on Computer Vision*, 2:416–423, 2001.

[85] Marina Meila. Comparing clusterings by the variation of information. *Learning Theory and Kernel Machines*, 2003.

[86] Mor Naaman, Andreas Paepcke, and Hector Garcia-Molina. From where to what: metadata sharing for digital photographs with geographic coordinates. *On The Move to Meaningful Internet Systems*, pages 196–217, 2003.

[87] Mor Naaman, Yee Jiun Song, Andreas Paepcke, and Hector Garcia-Molina. Automatic organization for digital photographs with geographic coordinates. *Proceedings of the Joint ACM/IEEE Conference on Digital Libraries*, pages 53–62, 2004.

[88] David Nistér and Henrik Stewénius. Scalable recognition with a vocabulary tree. *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2006.

[89] Aude Oliva and Antonio Torralba. Modeling the shape of the scene: a holistic representation of the spatial envelope. *International Journal of Computer Vision*, 42(3):145–175, 2001.

[90] Larry Page, Sergei Brin, Rajeev Motwani, and Terry Winograd. The PageRank citation ranking: bringing order to the web. *Stanford InfoLab Technical Report*, 1998.

[91] Stephen E. Palmer. Common region: a new principle of perceptual grouping. *Cognitive Psychology*, 1992.

[92] Stephen E. Palmer. *Vision Science: Photons to Phenomenology*. MIT Press, Cambridge, 1999.

[93] Stephen E. Palmer and Diane M. Beck. The repetition discrimination task: an objective method for studying perceptual grouping. *Perception & Psychophysics*, 2007.

[94] Stephen E. Palmer, Jonathan S. Gardner, and Thomas D. Wickens. Aesthetic issues in spatial composition: effects of position and direction on framing single objects. *Spatial Vision*, 21(3-5):421–49, January 2008.

[95] Stephen E. Palmer and Irvin Rock. Rethinking perceptual organization: the role of uniform connectedness. *Psychonomic Bulletin & Review*, 1994.

[96] Stephen E. Palmer, Eleanor H. Rosch, and Paul Chase. Canonical perspective and the perception of objects. *Attention and Performance IX*, pages 135–151, 1981.

[97] James Philbin, Ondrej Chum, Michael Isard, Josef Sivic, and Andrew Zisserman. Object retrieval with large vocabularies and fast spatial matching. *IEEE Conference on Computer Vision and Pattern Recognition*, 2007.

[98] James Philbin and Andrew Zisserman. Object mining using a matching graph on very large image collections. In *Sixth Indian Conference on Computer Vision, Graphics & Image Processing*, pages 738–745. IEEE, December 2009.

[99] Pavel Praks, Jiří Dvorský, and Václav Snášel. Latent semantic indexing for image retrieval systems. *SIAM Linear Algebra Proceedings*, 2003.

[100] Till Quack, Bastian Leibe, and Luc Van Gool. World-scale mining of objects and events from community photo collections. *Proceedings of the 2008 International Conference on Content-Based Image and Video Retrieval*, pages 47–56, 2008.

[101] Tye Rattenbury, Nathan Good, and Mor Naaman. Towards automatic extraction of event and place semantics from Flickr tags. *Proceedings of the 30th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 103–110, 2007.

[102] Stefan Roth and Michael J. Black. Fields of experts: a framework for learning image priors. *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 860–867, 2005.

[103] Carsten Rother, Sanjiv Kumar, Vladimir Kolmogorov, and Andrew Blake. Digital tapestry. *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 589–596, 2005.

[104] Bryan C. Russell, Alexei A. Efros, Josef Sivic, William T. Freeman, and Andrew Zisserman. Using multiple segmentations to discover objects and their extent in image collections. *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2:1605–1614, 2006.

[105] Bryan C. Russell, Antonio Torralba, Kevin P. Murphy, and William T. Freeman. LabelMe: A Database and Web-Based Tool for Image Annotation. *International Journal of Computer Vision*, 77(1):157–173, May 2008.

[106] Gerard Salton and Christopher Buckley. Term-weighting approaches in automatic text retrieval. *Information Processing & Management*, 24(5):513–523, 1988.

[107] Christoph Schlieder and Christian Matyas. Photographing a city: an analysis of place concepts based on spatial choices. *Spatial Cognition & Computation*, 9(3):212–228, July 2009.

[108] Patrick Schmitz. Inducing ontology from Flickr tags. *Collaborative Web Tagging Workshop at the 15th International World Wide Web Conference*, pages 210–214, 2006.

[109] Erick Schonfeld. Who Has The Most Photos Of Them All? Hint: It Is Not Facebook. *TechCrunch*, 2009.

[110] Florian Schroff and Antonio Criminisi. Harvesting image databases from the web. *IEEE 11th International Conference on Computer Vision*, October 2007.

[111] Patricia Schultz. *Rome*. Berlitz Publishing, 2003.

[112] Steven M. Seitz, Brian Curless, James Diebel, Daniel Scharstein, and Richard Szeliski. A comparison and evaluation of multi-view stereo reconstruction algorithms. *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2006.

[113] Shilad Sen, F. Maxwell Harper, Adam LaPitz, and John Riedl. The quest for quality tags. *Proceedings of the 2007 International ACM Conference on Supporting Group Work*, pages 361–370, 2007.

[114] Shilad Sen, Shyong K. Lam, Al Mamunur Rashid, Dan Cosley, Dan Frankowski, Jeremy Osterhouse, F. Maxwell Harper, and John Riedl. Tagging, communities, vocabulary, evolution. *Proceedings of the 2006 Conference on Computer Supported Cooperative Work*, pages 181–190, 2006.

[115] Jianbo Shi and Jitendra Malik. Normalized cuts and image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(8):888–905, 2000.

[116] Josef Sivic, Bryan C. Russell, Alexei A. Efros, Andrew Zisserman, and William T. Freeman. Discovering objects and their location in images. *Tenth IEEE International Conference on Computer Vision*, 1:370–377, 2005.

[117] Noah Snavely, Rahul Garg, Steven M. Seitz, and Richard Szeliski. Finding paths through the world's photos. *ACM Transactions on Graphics*, 27(3), 2008.

[118] Noah Snavely, Steven M. Seitz, and Richard Szeliski. Photo Tourism: exploring photo collections in 3D. *ACM SIGGRAPH*, pages 835–846, 2006.

[119] Noah Snavely, Steven M. Seitz, and Richard Szeliski. Skeletal sets for efficient structure from motion. *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2008.

[120] Erik Sudderth, Antonio Torralba, William T. Freeman, and Alan Willsky. Describing visual scenes using transformed dirichlet processes. *Advances in Neural Information Processing Systems*, 18:1297–1304, 2005.

[121] James Surowiecki. *The Wisdom of Crowds: Why the Many Are Smarter Than the Few and How Collective Wisdom Shapes Business, Economies, Societies and Nations*. Random House, 2004.

[122] Richard Szeliski. Image alignment and stitching: a tutorial. *Microsoft Research Technical Report*, 2006.

[123] Marshall F. Tappen, Bryan C. Russell, and William T. Freeman. Exploiting the sparse derivative prior for super-resolution and image demosaicing. *IEEE Workshop on Statistical and Computational Theories of Vision*, 2003.

[124] Antonio Torralba and Aude Oliva. Statistics of natural image categories. *Network: Computation in Neural Systems*, 14(3):391–412, August 2003.

[125] Kathleen Tuite, Noah Snavely, Dun-Yu Hsiao, Adam M. Smith, and Zoran Popović. Reconstructing the world in 3D: bringing games with a purpose outdoors. *Proceedings of the Fifth International Conference on the Foundations of Digital Games*, pages 232–239, 2010.

[126] Luis von Ahn and Laura Dabbish. Labeling images with a computer game. *Proceedings of the 2004 Conference on Human Factors in Computing Systems*, pages 319–326, 2004.

[127] Jingdong Wang, Jian Sun, Long Quan, Xiaoou Tang, and Heung-Yeung Shum. Picture collage. *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 1:347–354, 2006.

[128] Daphna Weinshall, Michael Werman, and Yoram Gdalyahu. Canonical views, or the stability and likelihood of images of 3D objects. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19:97–108, 2002.

[129] Max Wertheimer. Laws of organization in perceptual forms. *A source book of Gestalt psychology*, 1938.

Appendix A
SCENE RECONSTRUCTION PIPELINE

The work described in this thesis relies on the scene reconstruction pipelines of Snavely et al. [118] and Agarwal et al. [8], mainly as preprocessing. In this appendix, I describe this pipeline, including the data acquisition, image matching, 3D reconstruction, and final cleanup phase.

Figures A.1 and A.2 show the number of images included in the reconstructions of each of the scenes used in the experiments for this thesis, as well as an overhead view of the reconstructed scene points and cameras and a representative image (the first canonical view chosen by my greedy k-means algorithm). In addition, I have also used two city-scale reconstructions generated by Agarwal et al. [8], of Rome and Venice, reconstructed from 150,000 and 250,000 input images, respectively, though most of the images do not end up in the final reconstruction.

A.1 Retrieving the images and metadata

All of the images used in this thesis were obtained from the Internet photo-sharing site Flickr [2]. The images, tags, and other metadata were downloaded using modified versions of scripts originally written by Noah Snavely [118] and James Hays [55]. The images vary greatly in almost every conceivable way, though most share the crucial property that they are actual photographs taken by a human photographer at the scene. (Though there are often photos which were not taken at the desired scene, either due to polysemy or incorrect tagging, these photos are usually identified and discarded during the matching and reconstruction process.) A few exceptions that survive the reconstruction are panoramas constructed from multiple photos and photos of other photos, but these are uncommon enough to not significantly affect my results.

A.2 Computing feature tracks

The first step is to compute a feature-image incidence matrix from the set of photos from each scene. To do so, I use the SIFT keypoint detector [80] to find feature points in all of the images

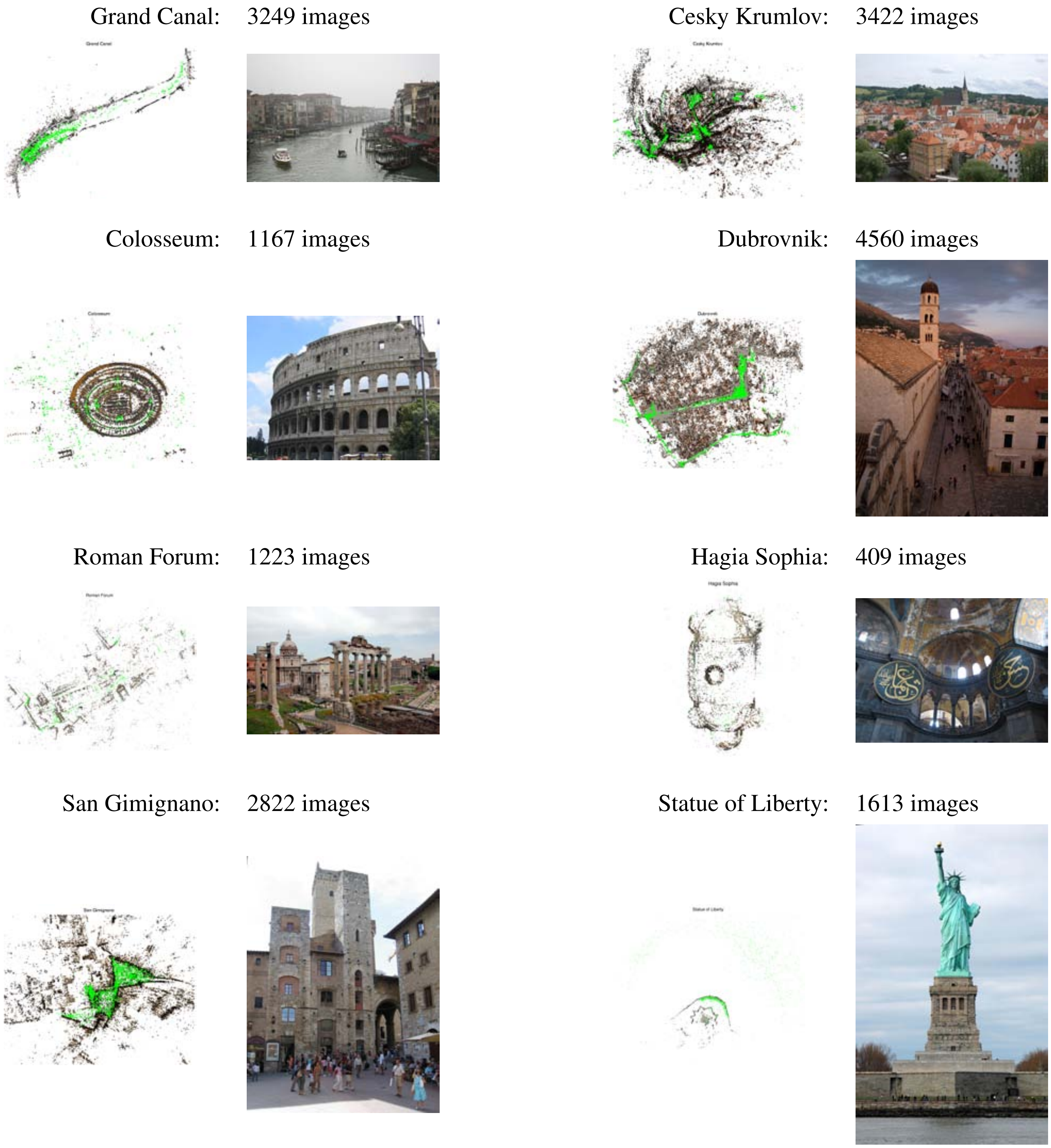


Figure A.1: A list of the Internet photo collections used in this thesis, and the number of images in each reconstruction, as well as an overhead view of the reconstructed scene, and a representative image. Green dots are camera positions.

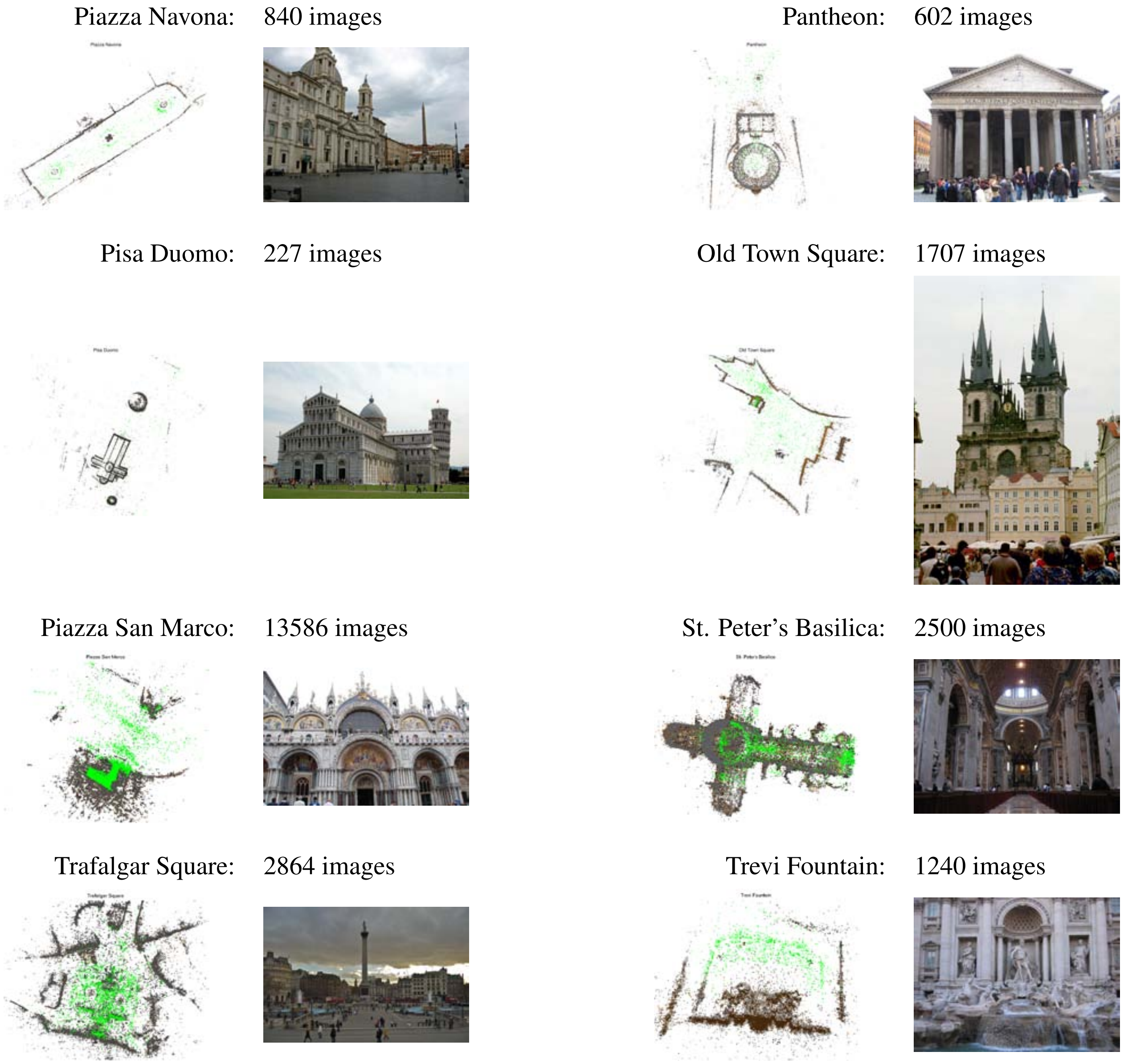

Piazza Navona: 840 images
Pantheon: 602 images
Pisa Duomo: 227 images
Old Town Square: 1707 images
Piazza San Marco: 13586 images
St. Peter's Basilica: 2500 images
Trafalgar Square: 2864 images
Trevi Fountain: 1240 images

Figure A.2: A list of the Internet photo collections used in this thesis, and the number of images in each reconstruction, as well as an overhead view of the reconstructed scene, and a representative image. Green dots are camera positions.

in $\mathcal{V}$. The feature points are represented using the SIFT descriptor. Then, for each pair of images, I perform feature matching on the descriptors to extract a set of candidate matches. The set of candidates is further pruned by estimating a fundamental matrix using RANSAC [41] and removing all inconsistent matches. After the previous step is complete for all images, we organize the matches into tracks, where a track is a connected component of features. I remove tracks containing fewer than two features total, or at least two features in the same image. At this point, I consider each track as corresponding to a single 3D scene point (though before computing the 3D reconstruction, the coordinates of the point are unknown). From the set of tracks, it is easy to construct the sparse feature-image incidence matrix, where entry (i, j) is 1 if feature track i is present in image j and 0 otherwise.

The above matching process is feasible when the number of images in the collection is in the hundreds or even thousands, but for city-scale collections of hundreds of thousands of images the pairwise matching step becomes infeasible. For these collections, I use the process of Agarwal et al. [8]. After SIFT features are computed, each image is represented as a sparse vector of quantized features using the vocabulary tree approach of Nister and Stewenius [88]. Pairs of potentially matching images are then proposed by finding, for each image, a set of other images for which the sparse vector pairs have large inner product. These potential matches (which are far fewer in number than the total number of image pairs) can then be verified by a full pairwise match and geometric verification step. Further matches are then generated by one or more query expansion steps, which proposes image pairs that share an already-verified match. The entire matching process is distributed across many nodes using a greedy scheduling algorithm.

A.3 Structure-from-motion

Given a set of feature tracks, I use the structure-from-motion algorithm of Snavely et al. [118] to estimate all camera positions and a sparse set of 3D scene points. At a high level, this algorithm adds images to the reconstruction one at a time, performing bundle adjustment at each step using the sparse bundle adjustment library of Lourakis and Argyros [10]. For larger data sets, there are additional speedups: using the skeletal set method of Snavely et al. [119], which computes an initial reconstruction on a reduced set of images, and using the more efficient bundle adjustment solver of Agarwal et al. [8]. For city-scale scenes that consist of more than one connected component, structure-from-motion is run on each component independently.

A.4 Postprocessing

Once a 3D reconstruction has been computed, a few other steps are required for some of the visualizations I produce. First, the sparse 3D point cloud is rotated so that the y-axis is aligned with the direction of gravity estimated by the technique of Szeliski [122]. I then manually align the point cloud to a satellite image or floor plan, and compute the scale factor between the point cloud and the real world. Once the gravity vector is known, I detect images that are rotated by 90°, 180° (much less common), or 270°, and rotate them so that the gravity vector is best aligned with the vertical image direction.