Large Scale Scene Matching for Graphics and Vision

James Hays

CMU-CS-09-152

July 29, 2009

School of Computer Science
Carnegie Mellon University
Pittsburgh, PA 15213

Submitted in partial fulfillment of the requirements for the degree of Doctor of Philosophy.

This research was sponsored in part by the National Science Foundation through a Graduate Research Fellowship to James Hays and grants IIS-0546547 and CCF-0541230. Funding was also provided by Google. The views and conclusions contained in this document are those of the author and should not be interpreted as representing the official policies, either expressed or implied, of any sponsoring institution, the U.S. government or any other entity.

Keywords: Scene Matching, Image Completion, Image Geolocation, Scene Classification, Object Detection, Context, Lazy Learning, SVM, Graphics, Vision, Internet Data, Computational Photography

Abstract

Our visual experience is extraordinarily varied and complex. The diversity of the visual world makes it difficult for computer vision to understand images and for computer graphics to synthesize visual content. But for all its richness, it turns out that the space of "scenes" might not be astronomically large. With access to imagery on an Internet scale, regularities start to emerge – for most images, there exist numerous examples of semantically and structurally similar scenes. Is it possible to sample the space of scenes so densely that one can use similar scenes to "brute force" otherwise difficult image understanding and manipulation tasks? This thesis is focused on exploiting and refining large scale scene matching to *short circuit* the typical computer vision and graphics pipelines for image understanding and manipulation.

First, in "Scene Completion" we patch up holes in images by copying content from matching scenes. We find scenes so similar that the manipulations are undetectable to naive viewers and we quantify our success rate with a perceptual study. Second, in "im2gps" we estimate geographic properties and global geolocation for photos using scene matching with a database of 6 million geo-tagged Internet images. We introduce a range of features for scene matching and use them, together with lazy SVM learning, to dramatically improve scene matching – doubling the performance of single image geolocation over our baseline method. Third, we study human photo geolocation to gain insights into the geolocation problem, our algorithms, and human scene understanding. This study shows that our algorithms significantly exceed human geolocation performance. Finally, we use our geography estimates, as well as Internet text annotations, to provide context for deeper image understanding, such as object detection.

Acknowledgements

I would like to acknowledge my amazing thesis advisor Alexei (Alyosha) Efros. He has given me valuable guidance and encouragement and he has been a selfless promotor of my work. He has provided me with many opportunities professionally and personally. Prior to this thesis work, I had the fortune of being advised by Yanxi Liu and Irfan Essa who both helped me grow tremendously as a researcher.

I would like to thank my thesis committee – Jessica Hodgins, Martial Hebert, Takeo Kanade, and Rick Szeliski – for their guidance. I've had the pleasure of interacting with Jessica and Martial throughout my time at CMU. They've helped me in numerous ways and made me feel at home with their graphics and vision research groups, respectively. CMU is a great environment for a researcher and I thank the numerous collaborators who have helped me along the way.

I thank Flickr for keeping their site open to researchers such as myself and Yahoo! and Intel Research Pittsburgh for sharing their facilities with me as part of the data intensive supercomputing initiative.

And finally, I thank all the friends I've made in Pittsburgh for their support (especially recent babysitting). Most of all I am grateful to my wife, Sonia, who has been tirelessly supportive of me and our son Alex throughout my thesis work.

Table of Contents

List of Figures

List of Tables

Chapter 1

Introduction

We are in the midst of an imaging revolution enabled by inexpensive digital cameras and the Internet. Consumers bought 130 million standalone digital cameras last year and it is estimated that cell phones will put cameras into the pockets of one-third of the world population by 2011. Photographers share their images on massive, rapidly growing web sites – Flickr and Facebook have over three billion and ten billion photographs respectively. These photos are often accompanied by informative user or community created metadata. For the "Facebook generation", visual content is a core element of online communication. Meanwhile, computers, unable to cope with the diversity of our visual world, are nearly blind to image content and progress in image-based graphics is held back by this lack of image understanding. But just as large quantities of real data pushed speech recognition from an "AI-hard" problem to a widely deployed tool, Internet scale imagery could lead to vision systems reliable enough for everyday use.

Computer vision and graphics are difficult because of the exceeding complexity of our visual world. The traditional computer vision or computer graphics pipeline mitigates this complexity with a bottom up, divide and conquer strategy (e.g. segmenting then classifying, assembling part-based models, or using scanning window detectors). This thesis research is fundamentally different – we use holistic scenes as our elementary visual unit. We observe that while the space of images is infinite, the space of "scenes" might not be astronomically large. With access to imagery on an Internet scale, regularities start to emerge – for most images, there exist numerous semantically and structurally similar scenes. It is possible to sample the space of scenes so densely that one can "brute force" otherwise difficult image understanding and manipulation tasks. This thesis is focused on exploiting, refining, analyzing the large scale scene matching that lets us *short circuit* the typical computer vision and graphics pipelines for image understanding and manipulation.

First of all, let us be absolutely clear – brute force image matching will not solve all of computer vision. Image space is *effectively infinite* with thousands of meaningful dimensions. A photograph of a complex, dynamic scene of articulated figures is not just one in a million, it is one in the lifetime of the universe. Deeper computer vision reasoning about the precise pose, arrangement, and geometry of objects is necessary to understand the visual world as humans do. But in this thesis, we are concerned with understanding images at a global scene level. We hope that the image understanding provided by scene matching can provide priors and context which make deeper image understanding successful.

The key contributions of this thesis are:

- *General Scene Matching.* The key enabling observation for this thesis is that we've reached a *tipping point* where the flood of Internet imagery rivals the entire diversity of our visual experience. We can use this Internet imagery and annotation to push back against the complexity that makes scene understanding so difficult. Scene matching is the mechanism that we use towards this end. Throughout this thesis we examine several different image features for scene matching and we propose some of our own.

- *Scene Completion.* Using scene matching, we develop the first image completion algorithm that leverages data from outside imagery. Scene matching gives the algorithm implicit high-level scene manipulation capabilities – the objects or scene elements that are inserted adhere to valid scene semantics because they came from real scenes with similar layouts. We are the first to create a standardized test set for image completion and perform a perceptual study to quantify the effectiveness of our algorithm.

- *Geographic Reasoning for Images.* We are the first to approach the problem of *global* image geolocation by utilizing the huge amount of geotagged data online. We consider not just landmarks but general scenes. Beyond location, we can estimate a litany of geographic properties. With collaborators, we build global movement models from the spatiotemporal statistics of our Internet data and use this to geolocate sequences of images. We measure human image geolocation performance and show that our computational methods are more accurate.

- *Context for Object Detection.* We use scene matching to provide novel sources of context for object detection – geographic information from geotags and semantic information from Internet keywords. With collaborators, we build a state-of-the-art object detection system and evaluate numerous contextual cues.

- *Large Scale Learning.* We show that "lazy learning" can provide dramatic improvements to scene matching applications with little additional complexity. We use a multiclass Kernel SVM to learn geographic classifiers from the nearest neighbor scene matches. We demonstrate significantly improved performance for image geolocation.

In the remainder of the Introduction we provide a rigorous definition of what we mean by "scene" (Section 1.1), give an overview of our scene matching architecture and features (Section 1.2), and discuss the data sources and data sets used throughout this thesis (Section 1.3).

1.1 Definitions

Our usage of "scene" and "scene similarity" are similar to a layperson's intuition. A *Scene* can be considered an abstraction of an image that is less concerned with the exact appearance, structure, and arrangement of elements than with the high-level semantics depicted. In traditional computer vision, reasoning in "scene" space might imply seeing through "image" space to more meaningful, possibly 3d real world structure. But we do not necessarily make this distinction. We can find similar scenes while working in directly image space, although more invariant features can also be useful.

It is easiest to define "scene" by asking *what properties do we expect similar scenes to share?* Different applications may be concerned with different aspects of scene similarity, but in general if two scenes are said to be "similar" or "matching" then that implies the following –

- The coarse layout of scene elements and the coarse geometric structure are similar. Scene elements with similar semantic meaning (roads, buildings, sky) occupy corresponding image regions and have roughly the same scale. The scene elements should be similarly shaped and oriented.

- The view or camera parameters are similar. This includes the focal length of the camera, the height of the camera, the in-focus depth of field, and the orientation of the camera with respect to the horizon.

- The appearance and materials of the scenes are similar. This includes color, texture, illumination, weather, vegetation / land cover type, and architectural style.

- The high-level semantics of the situations represented are similar. This includes object likelihoods (there could be a hairdryer, but not an airplane), navigation possibilities (I can turn left but not right), function (kitchen, highway, or basketball court), whether the scenes are natural or manmade, and location (indoors versus outdoors, Japan versus California).

Defined as such, the space of scenes is intrinsically quite high-dimensional. For two scenes to be similar does *not* imply that all of these properties are precisely the same. It is very unlikely that the exact articulation of objects or exact geometry would be the same between two scenes. Scene similarity is a continuous notion – given a query scene, we can say that one scene is a better match than another scene.

Our usage of "scene" is not limited to meaning a physical, real-world, navigable scene as it has been previously defined in the literature. For instance, Aude Oliva and Antonio Torralba state –

> In an attempt to define what a "scene" is, as opposed to an "object" or a "texture", we propose to consider the absolute distance between the observer and the fixated zone. Therein, if an image represents an "object" when the view subtends 1 to 2 meters around the observer, a "view on a scene" begins when there is actually a larger space between the observer and the fixated point, usually after 5 meters. Thus, whilst most of the "objects" are at a hand distance, a scene is mainly characterized as a place in which we can move. [74]

Our usage of "scene" is more general. For instance, it would be reasonable to try and find a similar scene given a portrait, a close-up view of an object, a macro view of a flower, or even an abstract illustration. Other authors seem to have relaxed their usage of the terminology as well, using the phrase "scene matching" to describe the retrieval of similar images of graffiti [89] or portraits and close-up object views [103], while other authors use more generic phrases such as "image matching" [100].

1.2 Scene Matching and Features

Our scene matching process is simple – for a novel photo, we find nearest neighbors scenes according to various image features. We make no attempt to optimize the search process even though it can be cumbersome with millions of images. Instead we focus on the design of features and distance metrics.

With large amounts of data, surprisingly basic image features can find similar scenes, even tiny RGB images [100]. In Scene Completion (Chapter 3), we rely on the gist descriptor [72] which simply measures edge intensity and direction over a coarse grid. Even though the gist descriptor is a very low level measure of scene statistics, it can often retrieve scenes with high level semantic agreement as long as one has millions of images to search through. In im2gps (Chapter 4) we employ several more features that measure texture, color, and straight line statistics of a scene and show that these improve performance considerably. In Chapter 6, we add SIFT [65] histograms and higher level features that measure the statistics of geometric regions of images (e.g. the color of vertical surfaces). The features are described in more detail as they are introduced throughout the thesis.

There is no universal, ideal feature set. As feature sets become larger and over-complete it is more important to use learning, as in Chapter 6, to focus on relevant features for a given task or a given query.

1.3 Data

For this thesis I use Flickr.com as the sole source of data. As of July 2009, Flickr has ∼1.3 billion *high resolution photographs* available for download, with ∼50 million added in the past month. While search engines such as Google Image Search index more images, a significant portion of those images are low quality photos or illustrations. Furthermore, search engines truncate the results for any given query to a few thousand results [33].

Flickr photographs can be weakly labeled with text “tags”, “titles”, and “descriptions” but most are not. Furthermore, most of these labels do not correspond to object and scene descriptions (e.g. “chair” has ∼169 thousand hits and “kitchen” has ∼300 thousand hits), but rather to place and event names (e.g. “Japan” has ∼5 million hits and “wedding” has ∼8.2 million hits). Object annotations tend to be accurate, although many Flickr images are quite unusual or artistic. [6] reports that half of photos labelled with landmark names represent those concepts poorly.

Since August 2006, Flickr users have been able to “geotag” their photographs. As of July 2009, there are ∼77 million publicly available photographs with GPS coordinates attached, with ∼2 million added in the past month. These coordinates are either hand annotated by the photographer using a map applet or added automatically be GPS-enabled cameras (such as an iPhone).

Two large databases of images were created for this thesis:

- *Scene Completion database.* ∼2.3 million Flickr images used for the original Scene

Completion experiments. Approximately half of the database, downloaded December 2006, consists of hand-selected Flickr "Groups" that specialize in outdoor photography. The other half, downloaded April 2007, was built from keyword searches. The images are fairly high quality in general and well suited to outdoor image completion. The images have text annotations but they are never used. More details about the data are provided in Chapter 3.

- *im2gps database.* ∼6.5 million Flickr images introduced in im2gps and downloaded November 2007. These images all have geotags as well as text and time labels. Geotagged images also appear to be of slightly higher quality than the general pool of Flickr photos (in general, photo annotation seems correlated with picture quality). This database is utilized in every chapter of this thesis. It is used as unlabelled data for Scene Completion, as geotagged data for im2gps and its follow up work, as temporally tagged data to build global travel models (Chapter 4.6), and as text tagged data to provide context for object detection (Chapter 5). More details about the data and its geographic distribution are provided in Chapter 4.

These databases are available to other researchers, but it is generally easier to recreate equivalent databases directly from Flickr. We provide code online for this purpose.[1] Approximately 1 Terabyte of storage is required for each 5 million JPEG compressed photographs. Computing all scene features for each million images requires ∼0.15 CPU years for Scene Completion features, ∼1 CPU year for the im2gps features, and ∼2 CPU years for the additional features described in Chapter 6.

Paired with these databases are several test sets:

- *Scene Completion test set.* 51 image completion test cases used to quantitatively evaluate Scene Completion (Chapter 3), as well as 27 additional test cases. Each test case is an unmanipulated photo paired with a mask that indicates which pixels must be replaced. This is the first shared and rigorously evaluated test set for image completion. The images are taken from either the LabelMe database [84] or personal photos and do not overlap with either image database. The test set is available online.[2]

- *im2gps test set.* 237 geotagged images used to evaluate im2gps. These photos were manually filtered from a random subset of the 6.5 million im2gps database to remove corrupt or privacy-violating images.

[1] http://graphics.cs.cmu.edu/projects/im2gps/flickr_code.html
[2] http://graphics.cs.cmu.edu/projects/scene-completion/

- *2k random test set.* 2000 geotagged images randomly sampled from the im2gps database, without filtering.
- *geo-uniform test set.* 955 geotagged images randomly sampled from the im2gps database such that their distribution is roughly uniform over the land area of the Earth. See Section 4.5 for more details.
- *Human geolocation test set.* 64 geotagged images manually selected from a random subset of the im2gps database. Images that had no geographic information were intentionally removed. Landmark, landscape, and urban photographs are all well represented. This test set was used to measure human geolocation performance (Chapter 7) and compare it with our computational methods.
- *Image sequence test set.* 4117 geotagged photos by 20 Flickr users which were uploaded after November 2007 (and thus do not overlap with the im2gps database). These sequences were filtered from a larger set of sequences by the heuristics described in Section 4.6. This test set is used to evaluate Image Sequence Geolocation.

The im2gps and human geolocation test sets are available online.[3] Because the test sets overlap with the im2gps database, care must be taken to avoid scene matching to the same photos or the same photographers.

[3]http://graphics.cs.cmu.edu/projects/im2gps/

Chapter 2

Background and Related Work

This chapter discusses research related to holistic image representations and semantic image matching in computing as well as visual memory and scene context in cognitive neuroscience. These references cover work that is most closely related to the scene matching process that is at the core of this thesis. References related to the specific applications of scene matching such as image completion, geolocation, object detection, and learning can be found in the appropriate chapters.

2.1 Holistic Image Representations in Computing

2.1.1 Content Based Image Retrieval

Content based image retrieval (CBIR) is concerned with allowing users to easily and efficiently search libraries of visual data. Queries to a CBIR system may take the form of an example image, a user painted sketch, or constraints on the layout of scene elements such as buildings or trees. See [61] for a survey of the field. While early CBIR systems "could only be used effectively by scientists", more recent efforts focus on "bridging the semantic gap" to find results that don't just match with respect to low-level features but also with respect to high-level semantic concepts [61]. Thus CBIR and our definition of "scene matching" overlap considerably. Popular image features in CBIR such as color and texture histograms have also proved useful for scene matching applications [39], and conversely features used heavily in scene matching such as the gist descriptor [74] have been used in CBIR [81]. Recent work focuses on improving retrieval for weakly and noisily labelled Internet images (such as the

Flickr images we use) [6, 81].

CBIR might be distinguished from "scene matching" applications (Section 2.1.4) in that CBIR's ultimate goal is the user-guided image retrieval itself, perhaps for a setting such as an internet search engine, whereas scene matching in the computer graphics and vision communities is often concerned with finding similar images to drive some secondary task such as object recognition [86], image parsing [63, 103], image completion [38], or image-based rendering [89]. With more specific tasks in mind, scene matching applications can be more precise about what domain of imagery they expect to work with and how they require images to be similar. For instance [38] works only with outdoor scenes and expects matching photos to share semantics, coarse geometric structure, and camera parameters.

2.1.2 Scene Categorization

Scene categorization is the problem of classifying an image into one of a small number (typically less than 20) of predefined scene categories such as mountain, beach, kitchen, or city [32, 108]. Features and algorithms first demonstrated for scene categorization such as the gist descriptor of [74] and the spatial pyramid of [60] have proven to be very useful for tasks driven by scene matching (Section 2.1.4). The categorization task offers an easy way to quantify scene matching performance. However, setting up the data sets and ground truth is tricky – how does one decide on categories? Does one pick photographs knowing the categories? How is membership into categories decided? Scenes don't necessarily fit neatly into categories. Humans will disagree on whether a scene belongs to a "street" class or an "inside of city" class. How much of our visual experience lies on the boundary between these categories? For instance, in one 5-way classification task ("coasts", "rivers/lakes", "mountains", "plains", "forests") human labellers agreed with each other only 89.7% of the time [109].

When the scene categories are specific real-world locations, e.g. "corridor 6b", "elevator 400/1", and "Draper Plaza", scene categorization becomes place recognition. [104] uses the gist descriptor to recognize these types of environments. [116] introduces a new global scene feature, the spatial PACT, to categorize the location of a mobile robot.

2.1.3 Global Image Features for Object Recognition

Because similar scenes are likely to contain similar objects, many object classification algorithms include global image features that are similar to those used for scene matching. For instance, all of the entries to the PASCAL VOC 2007 object classification challenge, in which the goal is to decide whether an image contains an object of a certain class, use some type of global features computed over the entire image or at interest points rather than trying to specifically detect and localize the object in question. [30]. Indeed there are questions as to whether the "methods are learning more about context/scene appearance than object appearance?" [30].

Object detection, the task of recognizing and localizing objects in an images, also benefits from global image features. "Statistical context priming for object detection" [102] learns relationships between gist-like global image features and likely object locations. "Object detection and localization using local and global features" [69] makes use of this relationship to provide context to more discriminative local features for object detection.

2.1.4 Applications of Scene Matching

There are numerous recent projects that utilize non-parametric scene matching with global image representations. Similar to my work in this thesis, these papers all have an explicit division between finding or grouping similar scenes and then utilizing those scenes for secondary computer graphics or computer vision tasks.

"Object Recognition by Scene Alignment" [86] improves object detections in an image by first finding similar scenes from the LabelMe database [84] and then examining labelled objects in those scenes to provide priors on object location and scale. Because the space of scenes is very diverse, the relationship between scene features and object locations can be hard to capture with compact parametric models. So while previous work used Gaussian mixture models to relate scene features to object priors [102], the main novelty of "Object Recognition by Scene Alignment" is the examination of non-parametric scene matching for this application.

"Tiny Images" [100] provides the most rigorous examination of large scale data-driven computer vision to date. The authors build a database of 80 million 32x32 images by querying image search engines with 70 thousand concrete English nouns. Each image has one label associated with it – the noun used in the query that found it. The authors were inspired to use 32x32 images by the fact that humans can parse such low resolution images surprisingly well. For instance, humans recognize cars in 32x32 scenes better than the best computational methods recognize cars in full resolution images. Using nearest-neighbor image retrieval the authors examine many tasks such as object recognition, scene categorization, image colorization, and image orientation correction. They use WordNet [33] to group labels and recognize objects at different levels of semantic specificity (eg. "artifact", "instrument", "device",

"vice"). The authors evaluate many tasks with variably sized databases and conclude that simple non-parametric image retrieval with Internet scale data sets is competitive at many computer vision tasks. However, the rate of performance growth is not promising – to increase performance linearly the data set must grow logarithmically in many cases. Growing data sets orders of magnitude beyond 80 million images will be difficult in the near future. But the performance observed in Tiny Images could be a manifestation of the computational difficulty of leveraging 32x32 images. In such low resolution images there is no notion of texture and objects in scenes must be recognized by context more than appearance. While this is at the limit of human abilities, it is far beyond the capabilities of current computational methods. Scene matching benefits tremendously from texture and gist features [39] and object detection benefits from a diverse set of features [106]. It could be that by using such an impoverished image representation, "Tiny Images" is more vulnerable to the combinatorial explosion of image space (which is huge, even at 32x32) because it cannot "see through" semantically meaningless image variations without richer image features. But the study provides is a compelling argument that computer vision will not be solved by brute force with 32x32 images.

One of the major limitations of data-driven scene matching is the computational expense it requires. In the naive case, finding similar scenes means a linear scan through tens or hundreds of gigabytes of scene descriptors [38,39]. For some loss of accuracy one can perform coarse to fine search where the initial search finds candidate matches using low resolution descriptors or PCA coefficients [100]. In "Small codes" [103] the authors go even smaller, describing scenes with binary codes from 8 to 256 bits in size. A mapping from gist scene descriptors to small codes is learned such that scenes with small ground truth distances will have small Hamming distances between their binary codes. Ground truth scene distances are defined by object overlap (for 20,000 LabelMe images) or by the full dimensional gist distance (in the case of a larger database of 13 million unlabelled web images). Scene matching using the small codes is faster and more accurate than other dimensionality reduction schemes such as Locality Sensitive Hashing. It even outperforms the full gist descriptor when trained and tested with subsets of the labelled data. When the codes are small enough ($\sim$ 30 bits) to enumerate all possible codes in a hash table in memory, the retrieval can be extremely fast (6 *microseconds*). Longer codes can be exhaustively searched in less than a second if they still fit in memory. The authors use their scene matching to directly vote on image segmentation and labelling. For a novel image, the most similar scenes from the LabelMe database are retrieved, and for each image location the scene matches vote on the object label of the query scene. The method works reasonably well for labelling large scene elements such as "building", "sky", and "road', but since smaller scene elements won't precisely align (e.g. the people in a street scene) the combinatorics of object configurations make this simple approach unlikely to scale well.

"Can Similar Scenes Help Surface Layout Estimation?" [23] tries to improve geometric classification of images by using similar *unlabelled* examples of similar scenes. For a query image, similar scenes are found using [39]. These unlabelled examples are segmented and classified using the baseline geometric context system [40]. The hope is that 1) similar scenes will share similar surface layouts, 2) the baseline geometric classifier will be mostly accurate, and 3) any classification errors because of ambiguities in a specific scene can be smoothed over by correct classifications in the majority of scenes. Ideally, the scene matches will be geometrically identical to each other and the query but still visually varied. It turns out that segmentations and classifications from the similar scenes can indeed be used to help improve the geometric classification of the original scene, although the effect is somewhat weak. However, as scene matching improves the effect should become stronger.

"Creating and exploring a large photorealistic virtual space" [89] stitches together similar scenes to hallucinate plausible virtual environments. It is similar to the well known "Photo Tourism" work [90] except that instead of finding geometric relationships between multiple images of the same real-world location using structure from motion, it uses scene matching to piece together images which have no real world connection. Users can navigate between scenes by "rotating", "translating", and "zooming" a virtual camera. For each of these navigation possibilities, a new scene is found and fused with the previous scene(s) after simulating the camera motion with image warping. Scenes are matched using gist descriptors and small color images, similar to [38]. To improve the scene matching, the authors train classifiers to group the scenes into categories (referred to as "themes") such as "people", "landscape", or "skyline". When navigating a virtual environment, only scenes from the same theme will be used to extend the environment unless a transition is explicitly requested. To maintain the apparent height of the virtual camera the horizon in each scene needs to be identified. The authors estimate horizon locations using by relating gist features to horizon height as in [102].

The combinatorics of object layout and scene geometry make it unlikely that any two scenes will align at high resolution. This is a problem for applications that need precise correspondence between scenes to build object location priors [86], to parse scenes [103], or to share segmentations and geometric labels [23]. In "Sift Flow" [63,64], the authors compute a dense correspondence between scenes in order to match visually similar scene elements while also preserving local coherence and minimizing the total amount of shift that scene elements undergo. The algorithm is able to correspond semantically like regions between scenes with significant structural variation. The paper uses an increasingly popular scene descriptor – densely sampled, quantized SIFTs histogrammed with a spatial pyramid [60]. The authors demonstrate their dense correspondence between scenes by using scene matches from videos to predict motion fields in still images. The authors also show that re-ranking scene matches based on the cost of their dense correspondence leads to qualitatively better scene matches.

In [63] the authors parse scenes by label transfer and show promising results for recognizing "stuff" (large objects with variable shape such as buildings, trees, sky, and road) but not "things" (smaller scale, better defined objects such as fire hydrants or birds).

"Image Restoration using Online Photo Collections" [20] improves white balance, exposure, and contrast for photographs by finding similar scenes from a database of 1 million Internet photographs and transferring the matched scene statistics to the query photograph. The assumption is that the database images have correct (or at least better) white balance, exposure, and contrast. In "CG2Real" [48] this idea is taken even further – the query photographs are computer generated images. The scene matches are used not just to guide color corrections but also as texture sources to replace the query image content. A parameter controls how well a region needs to match before it is replaced by texture from a scene match. For most queries, the output ends up being a relatively seamless composite of real and computer generated content.

2.2 Human Visual Memory and Scene Perception

Biological vision systems have long been of interest to computer vision researchers. For example, the design of the Laplacian Pyramid [11] was justified by the understanding of the limits and biases of human perception. [65], in introducing the SIFT descriptor, devotes a section to connections with primate vision. The gist scene descriptor [74] was inspired by human rapid scene recognition capability.

It might seem that massively data-driven approaches to image understanding are not biologically plausible. Surely human visual memory capacity cannot compare to a 6 million image database [39]? It's not entirely clear. According to [16], "the empirical study of long-term memory capacity has been virtually absent because of the impractical requirements of testing humans over extended periods." Because of this, the authors instead perform such an experiment on pigeons. Two pigeons are trained in 700 sessions over 3 years to remember specific, binary responses in association with example images. Afterwards, the pigeons can correctly respond to 73% of photos out of a set of 1,825 which the authors estimate corresponds to a long term visual memory capacity of more than 800 images.

Studies over shorter time frames have shown that humans could remember 83% out of a set of 10,000 pictures [94]. But at the same time, humans are often amazingly bad at remembering the exact configuration of a specific, real-world scene – a phenomena referred to as "change blindness" [83]. Thus there was speculation that human visual memory recorded only high-level scene semantics and would be confused by scenes with similar semantics but different layouts. However, Oliva and colleagues have recently shown that human object and scene memory is surprisingly robust to interference from intentionally similar distracting imagery [73]. In one experiment, participants were shown a sequence of 3,000 scenes with only 3 seconds to study each image. After this two and a half hour visual marathon, participants' memory was measured with a paired, forced-choice examination asking them which photograph they had already seen. Many of the images came from semantically identical scene categories (e.g. "hair salon", "iceberg", "country road", "library", or "crashing wave") photographed in similar ways. Increasing numbers of similar scenes caused more confusion, but the effect was surprisingly minor. With 4 similar scenes in a category, participants were 84% accurate, but with 64 similar scenes, participants were still 76% accurate! These results, along with similar results concerning object memory [10], lead Oliva to state that "A massive memory for details lend credence to object recognition approaches that require brute force storage of multiples viewpoints and exemplars" and that "Massive memory capacity is the infrastructure of scene gist recognition [in humans]" [73].[1]

Humans use their massive scene memory to provide context for deeper image understanding. The psychology and neuroscience literature related to context in human vision is too broad to review here but [101] and [4] provide an excellent overview. Humans can use contextual information from numerous cues to help in image understanding, but there are two views on what the dominant source of context is – relationships between objects or relationships to a holistic scene representation. For instance, in trying to recognize a blurry object in a city scene, object context might provide information of the form "this object is on top of a sidewalk and beside a fire hydrant" whereas global scene context might provide information of the form "this object is near the horizon in a street scene with this rough layout and scale". Object context would be utilized in a serial process, perhaps recognizing easy objects and then moving on to more difficult ones. Global scene context does not require the recognition of any objects. It is consistent with the idea that image understanding proceeds top down from an understanding of global structure to recognition of local details [70]. Humans probably employ both forms of context, but it is an open question as to whether the sources of context are acquired through a single mechanism or through parallel pathways using different brain regions (although [29] suggests specialized brain hardware related to scene layout). In this thesis, scene matching is used to provide computer vision with contextual information analogous to the global scene context that humans employ.

[1]Oliva's usage of "scene gist" is not referring to the gist scene descriptor, but to the coarse understanding of a scene that humans rapidly acquire when viewing a novel image.

Chapter 3

Scene Completion Using Millions of Photographs

Chapter Summary. We present a new image completion algorithm powered by a database of 2.3 million photographs gathered from the Web. The algorithm patches up holes in images by finding similar image regions in the database that are not only seamless but also semantically valid. Our algorithm is entirely data-driven, requiring no annotations or labelling by the user. Unlike existing image completion methods, our algorithm can generate a diverse set of image completions and we allow users to select among them. We quantify the effectiveness of our algorithm, and compare to an existing image completion method, with a perceptual study. This work was described, in part, in [38].

[1]Project Page: http://graphics.cs.cmu.edu/projects/scene-completion/

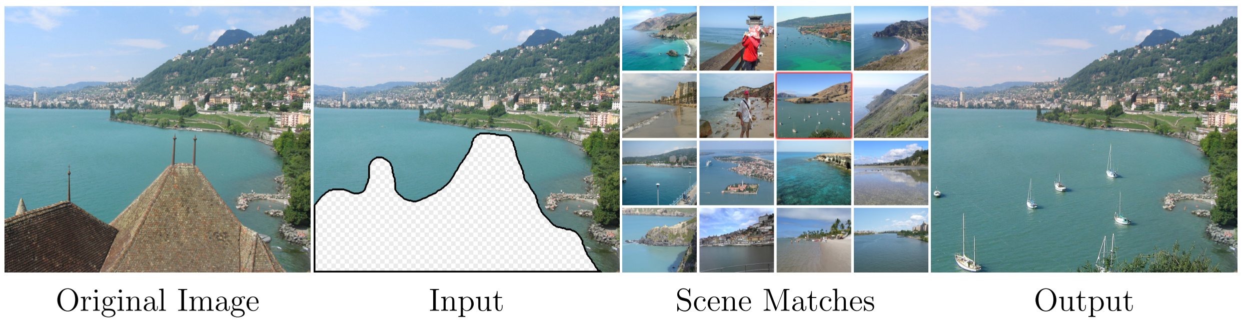


Figure 3.1: Given an input image with a missing region, we use matching scenes from a large collection of photographs to complete the image.

3.1 Introduction

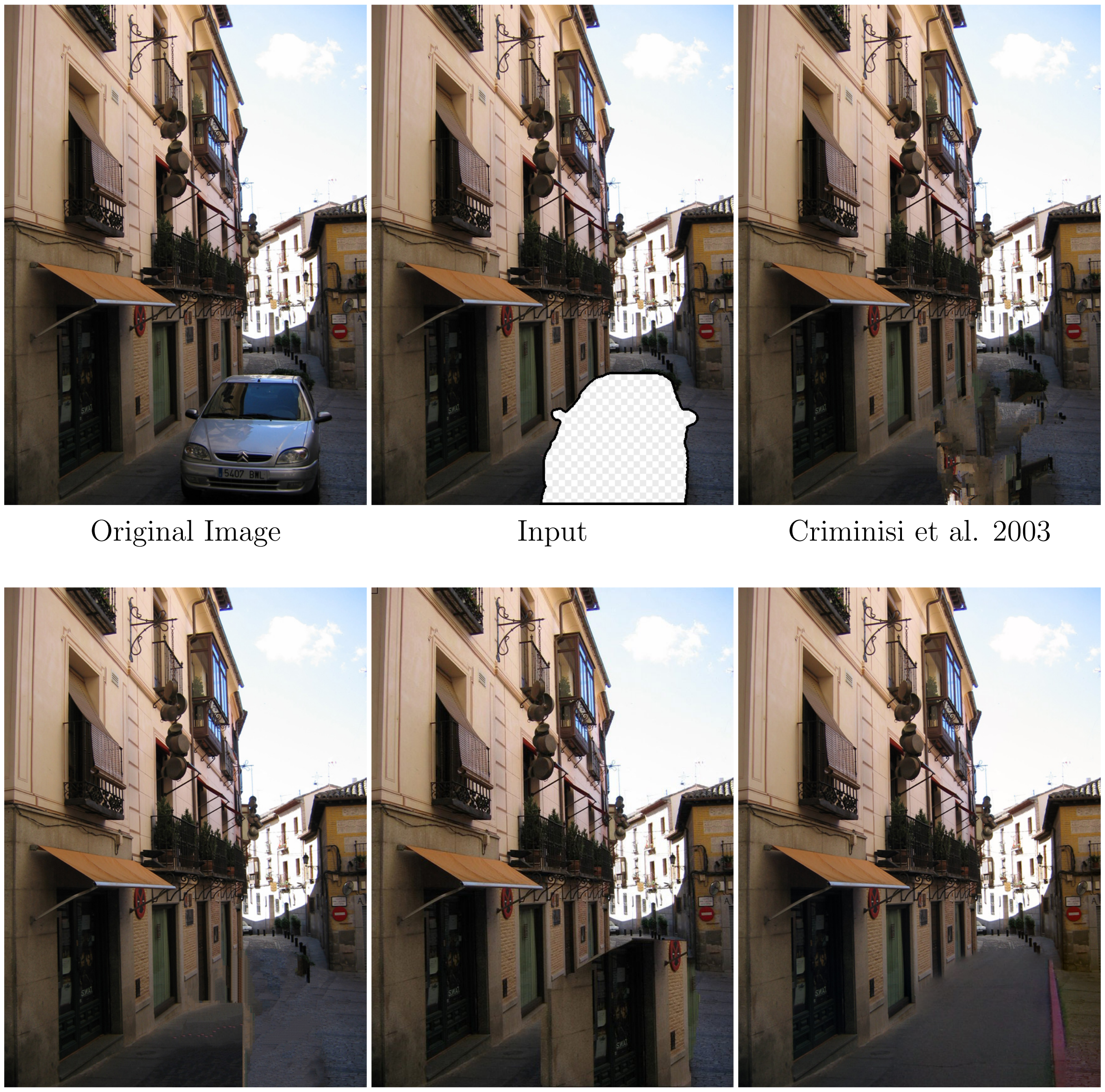
Every once in a while, we all wish we could erase something from our old photographs. A garbage truck right in the middle of a charming Italian piazza, an ex-boyfriend in a family photo, a political ally in a group portrait who has fallen out of favor [51]. Other times, there is simply missing data in some areas of the image. An aged corner of an old photograph, a hole in an image-based 3D reconstruction due to occlusion, a dead bug on the camera lens. Image completion (also called inpainting or hole-filling) is the task of filling in or replacing an image region with new image data such that the modification can not be detected.

There are two fundamentally different strategies for image completion. The first aims to reconstruct, as accurately as possible, the data that *should have been* there, but somehow got occluded or corrupted. Methods attempting an accurate reconstruction have to use some other source of data in addition to the input image, such as video (using various background stabilization techniques, e.g. [42]) or multiple photographs of the same scene [1, 90].

The alternative is to try finding a plausible way to fill in the missing pixels, hallucinating data that *could have been* there. This is a much less easily quantifiable endeavor, relying instead on the studies of human visual perception. The most successful existing methods [18,26,54,114, 115] operate by extending adjacent textures and contours into the unknown region. These algorithms are similar to texture synthesis algorithms such as [27,28,56,57], sometimes with additional constraints to explicitly preserve Gestalt cues such as *good continuation* [113], either automatically [18] or by hand [96]. Importantly, all of the existing image completion methods operate by filling in the unknown region with content from the known parts of the input source image.

Searching the source image for usable texture makes a lot of sense. The source image often has textures at just the right scale, orientation, and illumination as needed to seamlessly fill in the unknown region. Some methods [26, 115] search additional scales and orientations to gain additional source texture samples. However, viewing image completion as constrained texture synthesis limits the type of completion tasks that can be tackled. The assumption present in all of these methods is that all the necessary image data to fill in an unknown region is located somewhere else in *that same image*. We believe this assumption is flawed and that the source image simply doesn't provide enough data except for trivial image completion tasks.

Typical demonstrations of previously published algorithms are object removal tasks such as removing people, signs, horses, or cars from relatively simple backgrounds. The results tend to be fairly sterile images because the algorithms are only reusing image content that appeared somewhere else in the same image. For situations in which the incomplete region is


Original Image Input Criminisi et al. 2003

Smart Erase Wilczkowiak et al. Our algorithm

Figure 3.2: Results from various image completion algorithms including Microsoft Digital Image Pro Smart Erase.

not bounded by texture regions, or when there is too little useful texture, existing algorithms have trouble completing scenes (Figure 3.2).

3.2 Overview

In this study we perform image completions by leveraging a massive database of images. There are two compelling reasons to expand the search for image content beyond the source image. 1) In many cases, a region will be impossible to fill plausibly using only image data from the source image. For instance, if the roof of a house is missing or the entire sky is masked out. 2) Even if there is suitable content in the source image, reusing that content would often leave obvious duplications in the final image, *e.g.* replacing a missing building with another building in the image. By performing hole filling with content from other images, entirely novel objects and textures can be inserted.

However, there are several challenges with drawing content from other images. The first challenge is computational. Even in the single image case some existing methods report running times in the hours [26, 28] because of the slow texture search. Texture Synthesis based image completion methods are difficult to accelerate with traditional nearest-neighbor or approximate nearest-neighbor methods because of the high dimensionality of the features being searched and because the known dimensions of the feature being matched on change depending on the shape of the unknown region at each iteration.

The second challenge is that as the search space increases, there is higher chance of a synthesis algorithm finding locally matching but semantically invalid image fragments. Existing image completion methods might produce sterile images but they don't risk putting an elephant in someone's back yard or a submarine in a parking lot.

The third challenge is that content from other images is much less likely to have the right color and illumination to seamlessly fill an unknown region compared to content from the same image. More than other image completion methods we need a robust seam-finding and blending method to make our image completions plausible.

In this work we alleviate both the computational and semantic challenges with a two-stage search. We first try to find images depicting semantically similar scenes and then use only the best matching scenes to find patches which match the context surrounding the missing region. Scene matching reduces our search from a database of one million images to a manageable number of best matching scenes (60 in our case) which are used for image completion. We use a low dimensional scene descriptor [72] so it is relatively fast to find the nearest scenes,

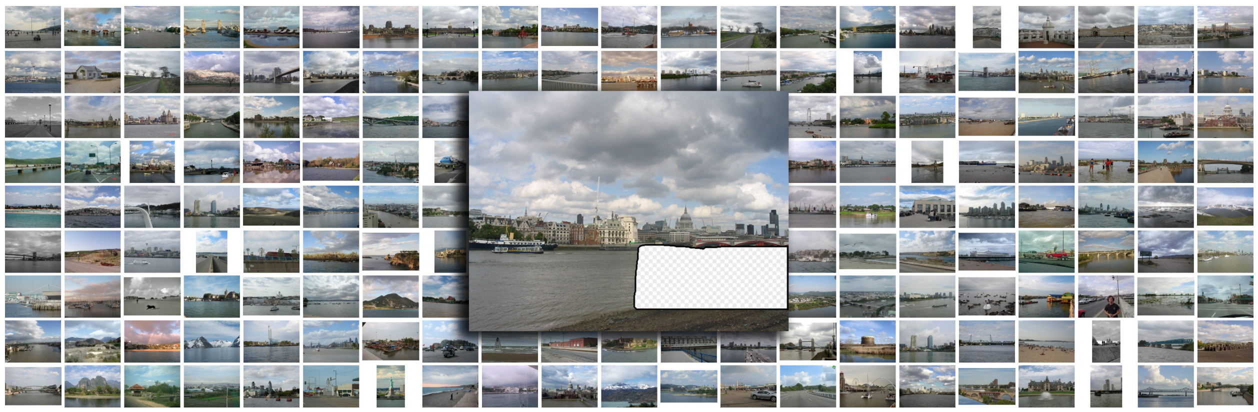


Figure 3.3: The first 164 nearest neighbor scenes for the incomplete image in the center. Most of the scenes are semantically and structurally similar; many are even from the same city (London).

even in a large database. Our approach is purely data-driven, requiring no labelling or supervision.

In order to seamlessly combine image regions we employ Poisson blending. To avoid blending artifacts, we first perform a graph cut segmentation to find the boundary for the Poisson blending that has the minimum image gradient magnitude. This is in contrast to minimizing the intensity domain difference along a boundary [115] or other heuristics to encourage a constant intensity offset for the blending boundary [46]. In section 3.4 we explain why minimizing the seam gradient gives the most perceptually convincing compositing results.

The image completion work most closely resembling our own, Wilczkowiak et al. [115], also demonstrates the search of multiple images. However, in their case it was only a few images that were hand selected to offer potentially useful image regions. Also related are methods which synthesize semantically valid images either from text or image constraints [22, 49]. These methods create semantically valid images through explicit sematic constraints using image databases with semantically labelled regions. The database labelling process must be supervised [22] or semi-supervised [49].

3.3 Semantic Scene Matching

Since we do not employ user-provided semantic constraints or a labelled database, we need to acquire our semantic knowledge from the data directly. This requires us to sample the set of visual images as broadly as possible. We constructed our image collection by downloading all of the photographs in thirty Flickr.com groups that focus on landscape, travel, or city photography. Typical group names are “lonelyplanet,” “urban-fragments,” and “ruraldecay.” Photographs in these groups are generally high quality. We also downloaded images based on keyword searches such as “travel,” “landscape,” and “river.” We discarded all duplicate images and all images that did not have at least 800 pixels in their largest dimension and 500 pixels in their smallest dimension. All images were down-sampled, if necessary, such that their maximum dimension was 1024 pixels. Our database downloading, pre-processing, and scene matching are all distributed among a cluster of 15 machines. In total we acquired about 2.3 million unique images totalling 396 gigabytes of JPEG compressed data.

In order to successfully complete images we need to find image regions in our database that are not just seamless and properly textured but also *semantically* valid. We do not want to complete hillsides with car roofs or have ducks swimming in city pavement which locally resembles a lake. To help avoid such situations we first look for *scenes* which are most likely to be semantically equivalent to the image requiring completion. The use of scene matching is the most important element of our image completion method. Without it, our search would not be computationally feasible and our image completion results would rarely be semantically valid. Our scene matching, in combination with our large database, allows us to do image completion without resorting to explicit semantic constraints as in previous photo synthesis work [22, 49].

We use the gist scene descriptor [69, 72] which has been shown to perform well at grouping semantically similar scenes (eg. city, tall buildings, office, fields, forest, beach) and for place recognition. It must be noted that a low-level scene descriptor in and of itself cannot encode high-level semantic information. Scenes can only be trusted to be semantically similar if the distance between them is very small. The way we address this issue is by collecting a huge dataset of images making it more likely that a very similar scene to the one being searched is available in the dataset. Indeed, our initial experiments with the gist descriptor on a dataset of ten thousand images were very discouraging. However, increasing the image collection to one million yielded a qualitative leap in performance. Independently, Torralba et al. [100] have observed a similar effect with a dataset of up to 80 million tiny (32x32) images.

The gist descriptor aggregates oriented edge responses at multiple scales into very coarse spatial bins. We found a gist descriptor built from 6 oriented edge responses at 5 scales aggregated to a 4x4 spatial resolution to be most effective. We augment the scene descriptor with the color information of the query image down-sampled to the spatial resolution of the gist.

Searching for similar scenes first rather than directly looking for similar patches speeds up our search dramatically. Instead of looking for image patches in all one million images at multiple offsets and scales we can instead eliminate 99.99% of the database quickly by finding the nearest neighbor scenes based on the relatively low dimensional scene descriptor.

Given an input image to be hole-filled, we first compute its gist descriptor with the missing regions excluded. This is done by creating a mask which weights each spatial bin in the gist in proportion to how many valid pixels are in that bin. We calculate the SSD between the the gist of the query image and every gist in the database, weighted by the mask. The color difference is computed in the L*a*b* color space. The distance is weighted such that each of the 4 different gist scales and the color information contribute roughly equally to the distance calculations. Examples of the nearest scenes for two particularly successful queries are shown in Figure 3.3.

Because we match scenes using arbitrary dimensions of the descriptor (depending on which regions of a query image are missing), we can not use PCA dimensionality reduction as suggested in [72]. For the same reason we do not use any approximate nearest-neighbor scheme since they tend to incur large relative errors when matching on arbitrary dimensions of descriptors with hundreds of dimensions. However, the descriptors are small enough to exhaustively search in a few minutes.

3.4 Local Context Matching

Having constrained our search to semantically similar scenes we can use traditional template matching to more precisely align those matching scenes to the local image context around the missing region. The local context is all pixels within an 80 pixel radius of the hole's boundary. This context is compared against the 200 best matching scenes across all valid translations and 3 scales (.81, .90, 1) using pixel-wise SSD error in L*a*b* color space. Only placements (translations and scales) for which the context is fully contained in the matching scene are considered. Since we expect our matching scenes to already be roughly aligned with the incomplete image we discourage spurious, distant matches by multiplying the error at each placement by the magnitude of the offset. At the minimum error placement of each matching scene we calculate a *texture energy distance* as the sum of the pixel-wise differences in edge energy $\parallel \nabla Scene(x,y) \parallel - \parallel \nabla Context(x,y) \parallel$ as an indication of how similarly textured each matching scene is with the incomplete image. This score will be used as part of the ranking criteria among candidate image completions.

We composite each matching scene into the incomplete image at its best scoring placement using a form of graph cut seam finding [57] and standard poisson blending [76]. Using a seam finding operation to composite the images is arguably undesirable for hole filling since a user might want to preserve all of the image data in the input image. Past image completion algorithms [18] have treated the remaining valid pixels in an image as hard constraints which are not changed. We relax this, as in [115], and allow the seam-finding operation to remove valid pixels from the query image but we discourage the cutting of too many pixels by adding a small cost for removing each pixel in the query image which increases with distance from the hole (Equation 2).

When performing a seam-finding operation and gradient domain fusion in sequence it makes sense to optimize the seam such that it will minimize the artifacts left behind by the gradient domain fusion. Jia et al. [46] uses dynamic programming to find a seam which has minimum intensity difference between the two images after allowing some constant intensity offset. The intuition is that humans are not sensitive to relatively large shifts in color and intensity as long as the shifts are seamless and low frequency.

We argue that it is better to minimize the gradient of the image difference along the seam instead of intensity differences. While the heuristic in [46] will indeed minimize the seam gradient if it succeeds in finding a constant color offset seam, that is impossible in many compositing situations (Figure 3.4). Minimizing intensity domain differences (even after optimizing for the best global color offset) can cause the seam to pass through many high frequency edges which will leave obvious blending artifacts after Poisson blending (Figure 3.4, a). If we instead minimize the gradient of the image difference along the seam (Figure 3.4, c), the seam tends to pass through regions of the images which either match (and thus have low absolute difference and thus low difference gradient) or are both constant colored (in which case poisson blending will hopefully do a good job of hiding the color difference at low frequencies). This is especially true in our scene completion domain – the query and scene matches are similarly colored. A seam that crosses an image edge produces more noticeable artifacts than small color differences across the seam.

We find our seam by minimizing the following cost function.

$$C(L) = \sum_{p} C_d(p, L(p)) + \sum_{p,q} C_i(p, q, L(p), L(q)) \tag{3.1}$$

Where $C_d(p, L(p))$ are the unary costs of assigning any pixel, p, to a specific label $L(p)$, and $C_i(p, q, L(p), L(q))$ is the higher-order cost for two pixels p and q taking labels $L(p)$ and $L(q)$ respectively. The label of each pixel, $L(p)$, has two values *patch* and *exist* corresponding to a pixel being taken from the scene match or the incomplete image. For missing regions of the existing image, $C_d(p, exist)$ is a very large number. For regions of the image not covered

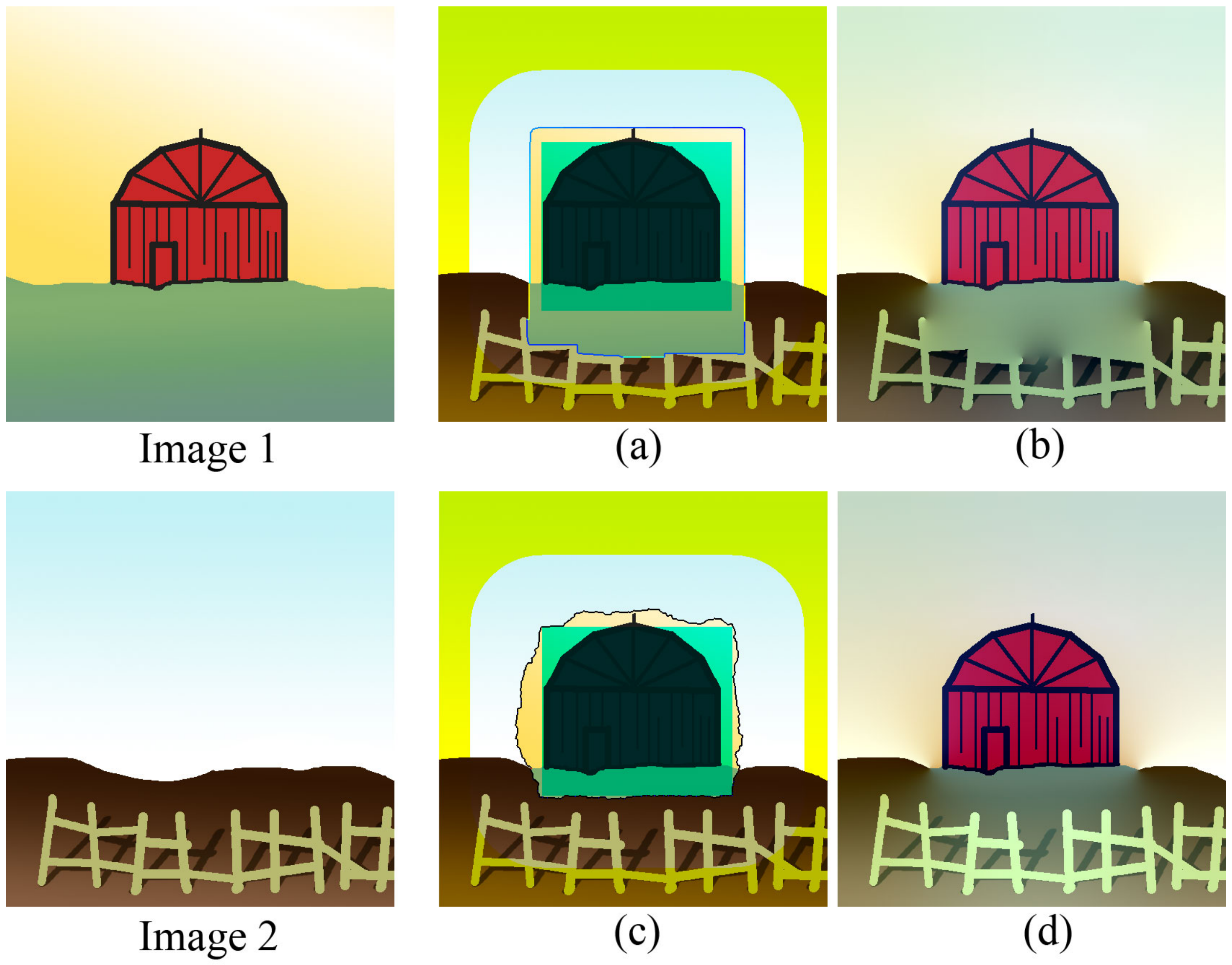


Figure 3.4: In this synthetic compositing example, we are placing the barn from Image 1 into the field of Image 2. The cyan region is constrained to come from image 1, and the Yellow region is constrained to come from image 2. (a) shows the optimal seam which minimizes intensity differences (even after some best, constant color shift). (c) shows the optimal seam which minimizes the gradient of the image difference. (b) and (d) are the Poisson blending results from (a) and (c) respectively.

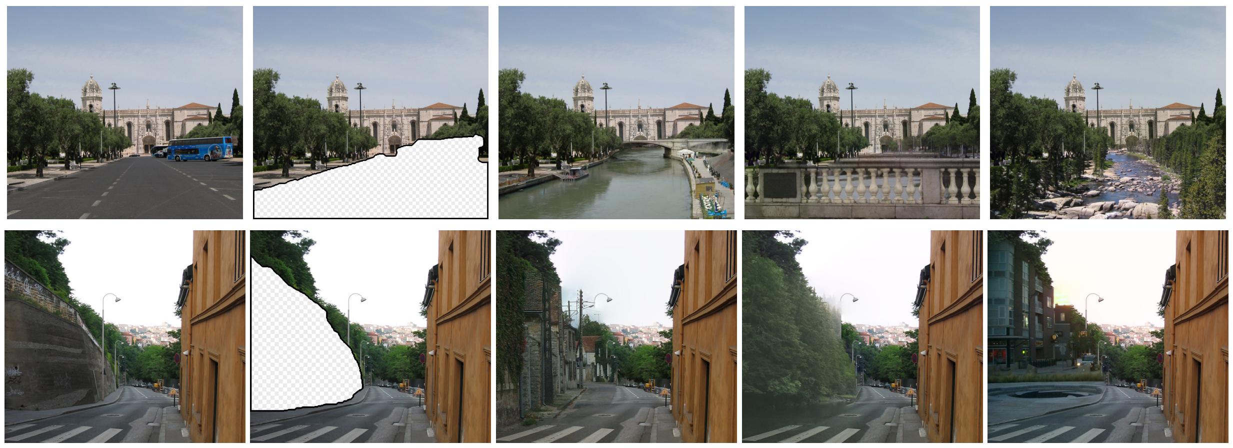


Figure 3.5: The algorithm presents to the user a set of image completions alternatives for each input. Here we show three such alternatives.

by the scene match, $C_d(p, patch)$ is very large. For all other pixels,

$$C_d(p, patch) = (k * Dist(p, hole))^3 \tag{3.2}$$

This small penalty assigns a cost to each pixel taken from the scene match which increases with a pixel's distance from the hole, $Dist(p, hole)$. We use an empirically found $k = .002$. Wilczkowiak et al. [115] use a similar penalty.

$C_i(p, q, L(p), L(q))$ is non-zero only for immediately adjacent, four-way connected pixels. When neighbors p and q share the same label, $L(p) = L(q)$, the cost is also zero. Along the seams, when $L(p) \neq L(q)$, $C_i(p, q, L(p), L(q)) = \nabla diff(p, q)$ where $\nabla diff(p, q)$ is the magnitude of the gradient of the SSD between the existing image and the scene match at pixels p and q.

Using this cost function has the added advantage that it can be globally minimized quickly using a graph cut [9] unlike the heuristic in [46] which has to iterate until convergence while estimating the best color offset. A very similar metric was also mentioned but not demonstrated in [1]. As in [1] we allow the Poisson blending to operate on the entire image domain instead of correcting one of the images. We use the poisson solver of [2]. It is worth noting that two image compositing techniques developed at the same time as our research [59, 112] may considerably improve the quality of our results.

Finally we assign each composite a score which is the sum of the scene matching distance, the local context matching distance, the local texture energy distance, and the cost of the graph cut. All four components of the score are scaled to contribute roughly equally. We present the user with the 20 composites with the lowest scores.

3.5 Results and Comparison

We test our algorithm on a set of 51 incomplete images. All test images come from either the LabelMe database [84] or our own personal photo collections, none of which are contained in our downloaded database. Images in the test set, about a half megapixel each, are higher resolution than the images used in most previous works [18, 26] and likewise the missing regions are quite large (56,000 pixels on average). The regions we removed from the photos all had semantic meaning such as unwanted objects, storefronts, walls with graffiti, roads, etc. The test set is made freely available on the authors' web page.

Image completion is an inherently under-constrained problem. There are many viable ways to fill a hole in an image. Previous approaches, which operate by reusing texture from the input image can offer relatively few viable, alternative completions (perhaps by changing parameters such as the patch size). While some such results might look slightly better than others, the semantic interpretation of the image is unlikely to change. On the other hand, our algorithm can offer a variety of semantically valid image completions for a given query image (Figure 3.5). After compositing the best-matching patches we present a user with the 20 top image completions (roughly equivalent to one page of image search results). In some cases, many of these completions are of acceptable quality and the user can select the completion(s) which they find most compelling. In other cases, only a few or none of the results were acceptable. The quality of our results benefits from this very simple user interaction and it is difficult for conventional image completion methods to offer an analogous selection of results.

Some of our image completions are shown in Figures 3.6, 3.7, and 3.8. The bottom result in Figure 3.8 is interesting because the scaffolding on the cathedral that was masked out has been replaced with another image patch of the same cathedral. The database happened to contain an image of the same cathedral from a similar view. It is not our goal to complete scenes and objects with their *true* selves in the database, but with an increasingly large database such fortuitous events do occur.

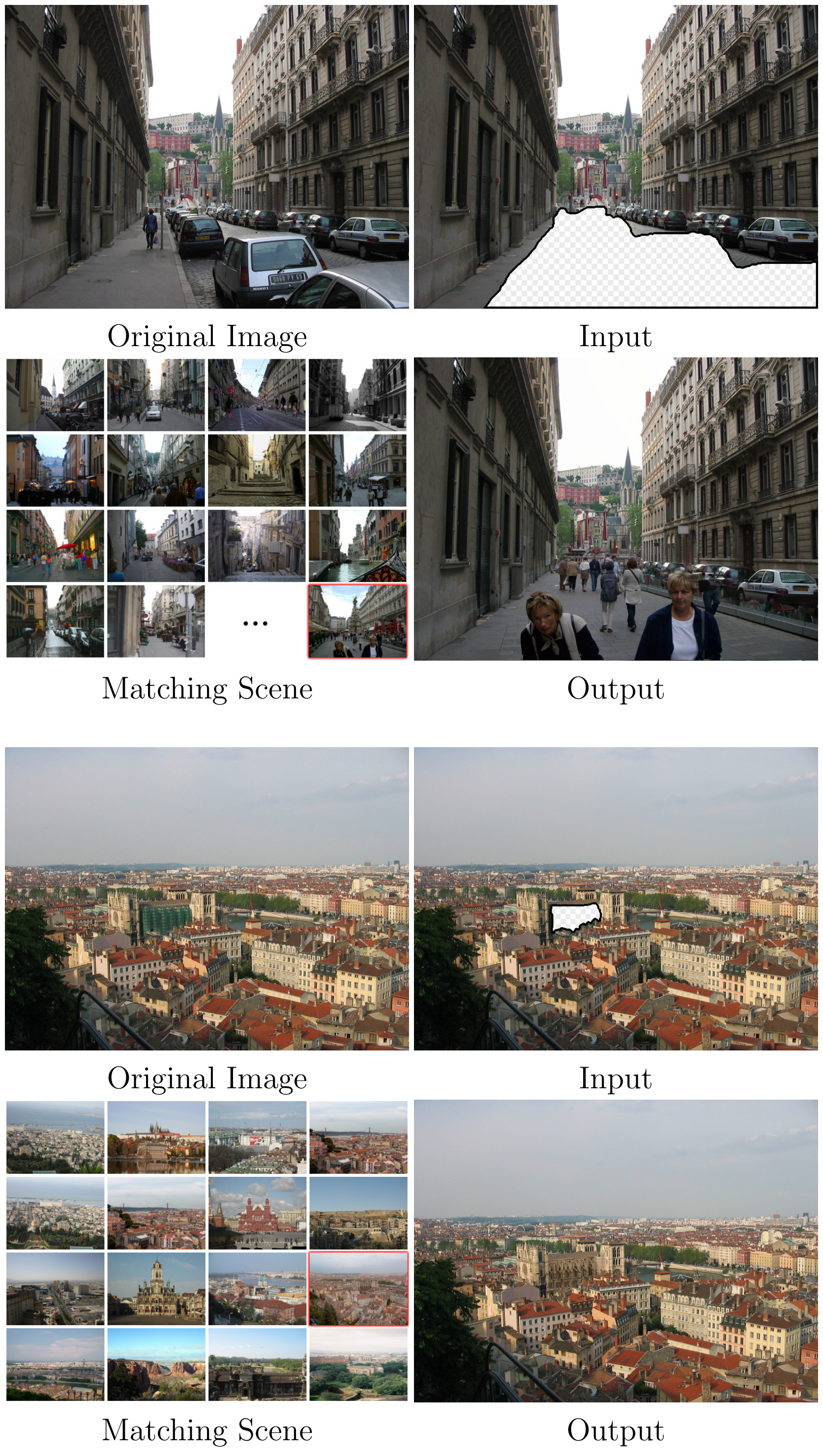


Figure 3.6: Results 1. The input and the matching scenes are composited together to create the outputs. The matching scene used in each output is highlighted in red. Note that the algorithm can handle a large range of scenes and missing regions. On rare occasions, the algorithm is lucky enough to find another image from the same physical location as seen in the bottom example.

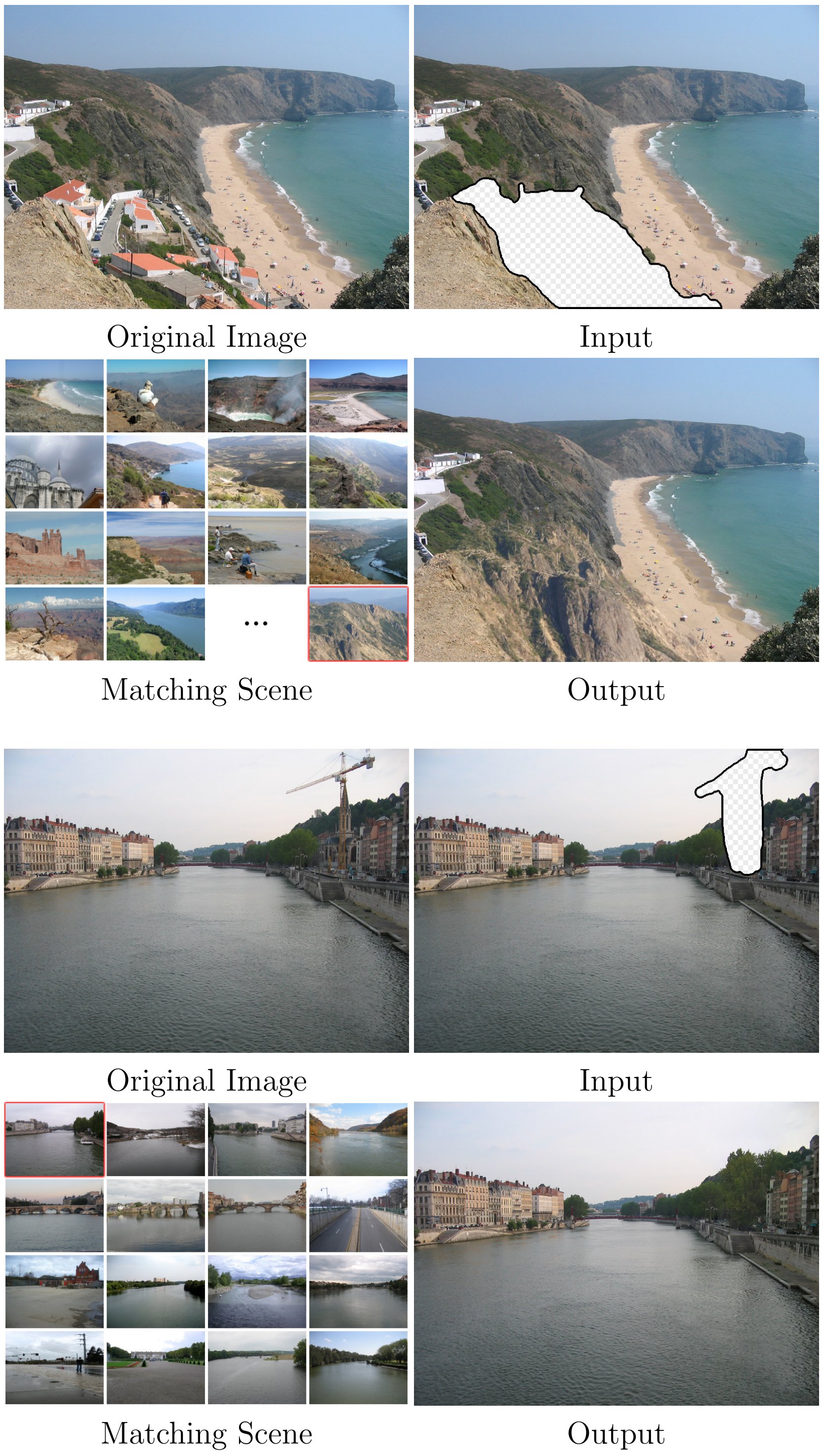


Figure 3.7: Results 2.

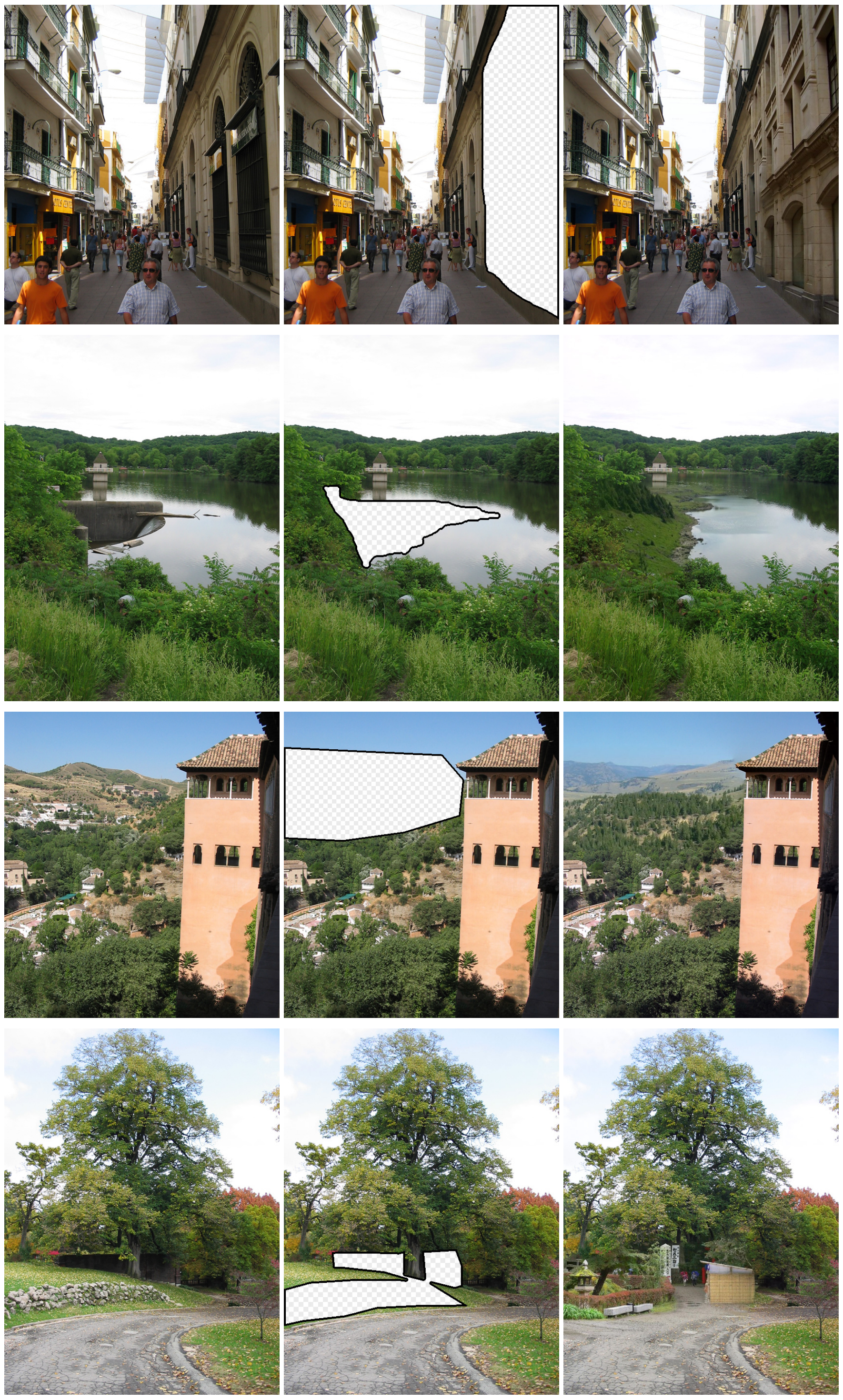


Figure 3.8: Results 3.

In all of the successful cases the completion is semantically valid but there might be slight low-level artifacts such as resolution mismatch between the image and patch, blurring from Poisson blending, or fine-scale texture differences between the image and patch. For failure cases these low level artifacts are often much more pronounced (Figure 3.12, top). Another source of failure is a lack of good scene matches which happens more often for atypical scenes (Figure 3.12, middle). Semantic violations (e.g. half-objects) account for another set of failures. The latter is not surprising since the algorithm has no object recognition capabilities and thus no notion of object boundaries.

For uniformly textured backgrounds (Figure 3.13, top), existing image completion algorithms perform well. However, our algorithm struggles since our scene matching is unlikely to find the exact same texture in another photograph. Furthermore, image completion algorithms such as Criminisi et al. [18] have explicit structure propagation which helps in some scenes (Figure 3.13, bottom).

As we get more data the scene matches improves and the image completions are better. Figures 3.9 and 3.10 show results obtained by using the 6.5 million image im2gps database instead of the 2.3 million image database used throughout this Chapter.

The scene matching, local context matching, and compositing take about 3, 10, and 4 minutes respectively on a single CPU but we parallelize all of these across many CPUs. Our algorithm is implemented in Matlab.

3.5.1 Quantitative Evaluation

It is difficult to rigorously define success or failure for an image completion algorithm because so much of it depends on human perception. While previous approaches demonstrate performance qualitatively by displaying a few results, we believe that it is very important to also provide a quantitative measure of the algorithm's success. Therefore, to evaluate our method, we performed an IRB-approved perceptual study to see how well naive viewers could distinguish our results, as well as those of a previous approach [18], from real photographs.[1] The study was performed on a set of 51 test images that were defined *a priori* and spanning different types of completions. We were careful not to include any potentially recognizable scenes or introduce bias that would favor a particular algorithm. We generated three versions of each image– the real photograph from which the image completion test

[1]One could argue that we should evaluate not only how reliably the image completion result fools novel viewers, but also how faithfully the result preserves the original image and how satisfied the user is with the novel composition.

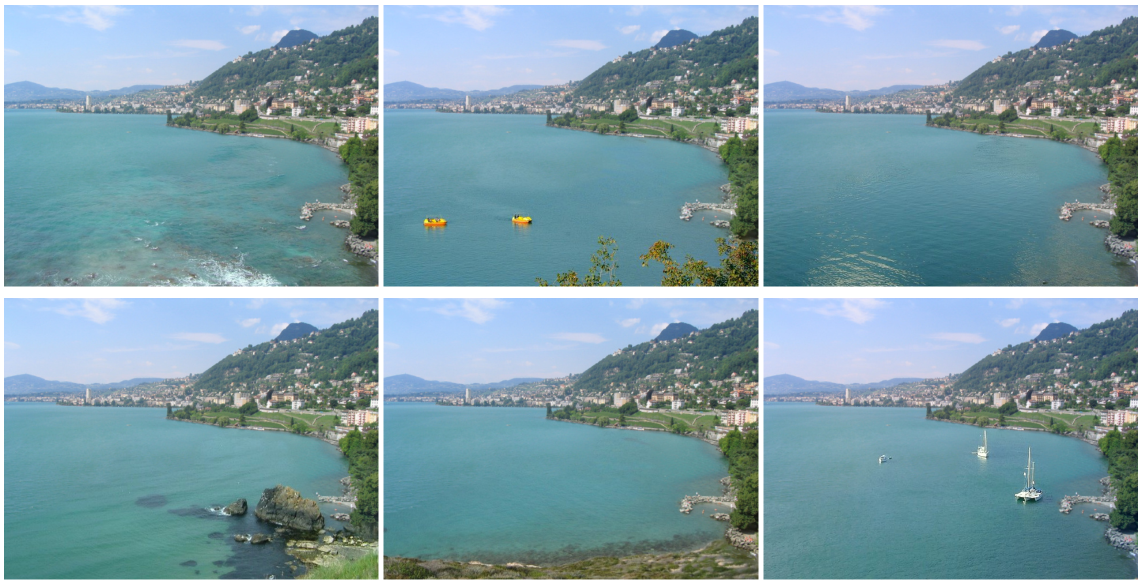


Figure 3.9: Six possible completions for the photograph shown in Figure 3.1. These results were produced with the larger photo dataset and features described in im2gps (Chapter 4).

cases were constructed, the result from Criminisi et al., and the result from our algorithm.

Each of our 20 participants viewed a sequence of images and classified them as real or manipulated. Of the 51 images each participant examined, 17 were randomly chosen from each source, but such that they do not see multiple versions of the same image. The order of presentation was also randomized. The participants were told that some of the images would be real, but they were not told the ratio of real vs manipulated images. We also told the participants that we were timing their responses for each image but that they should try to be accurate rather than fast. Overall the participants classified 80% of the images correctly. No effort was made to normalize for the differences in individual aptitude (which were small).

With unlimited viewing the participants classified our algorithm's outputs as real 37% of the time compared with 10% for Criminisi et al. [18]. Note that participants identified real images as such only 87% of the time. Participants scrutinized the images so carefully that they frequently convinced themselves real images were fake.

It is interesting to examine the responses of participants over time. In Figure 3.14 we measure the proportion of images from each algorithm that have been marked as fake with an increasing limit on the amount of time allowed. We claim that if a participant who has been specifically tasked with finding fake images cannot be sure that an image is fake within

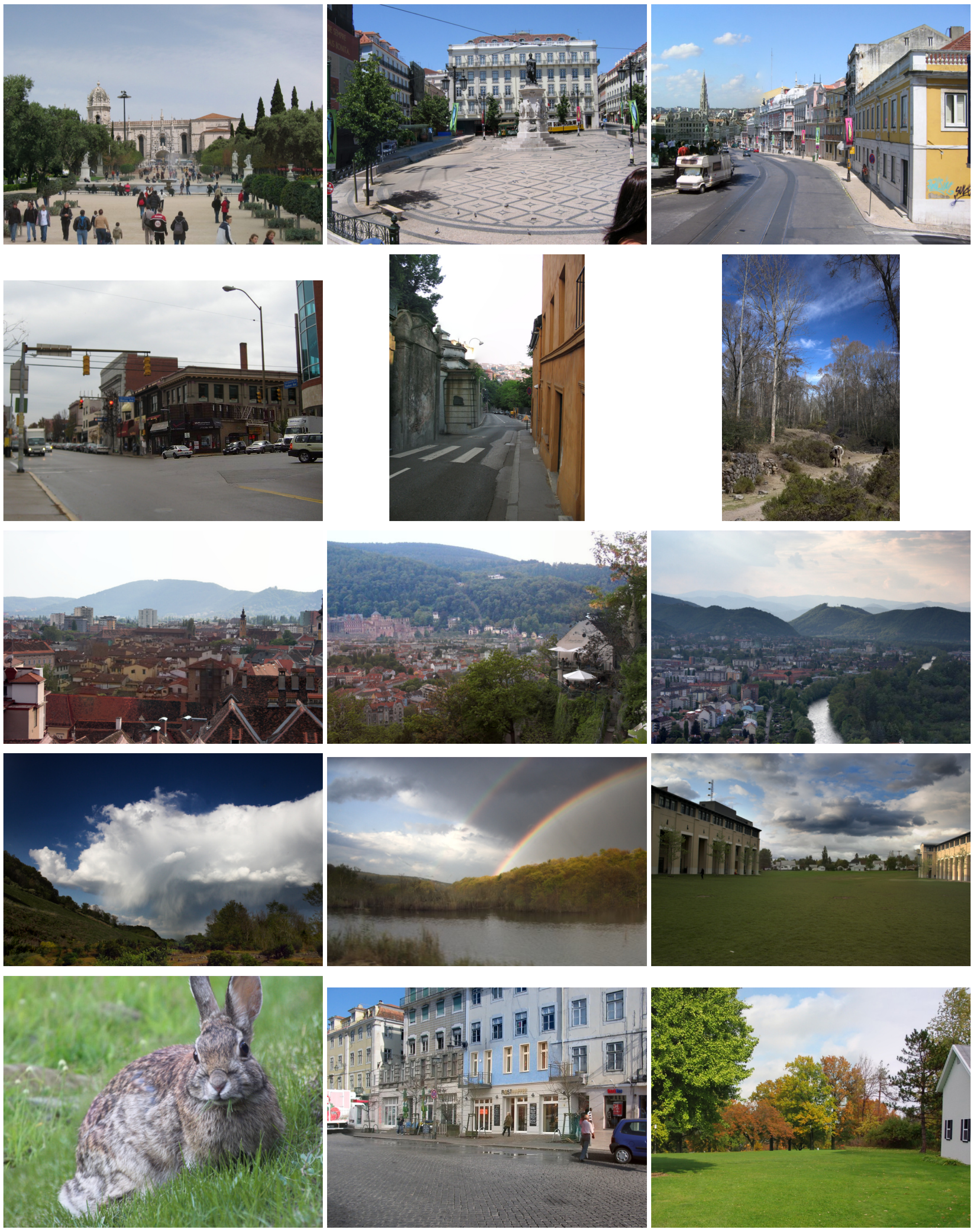


Figure 3.10: Additional results. Can you spot the manipulations? The inputs are shown separately in Figure 3.11. These completions were produced with the larger photo collection and features described in im2gps (Chapter 4).

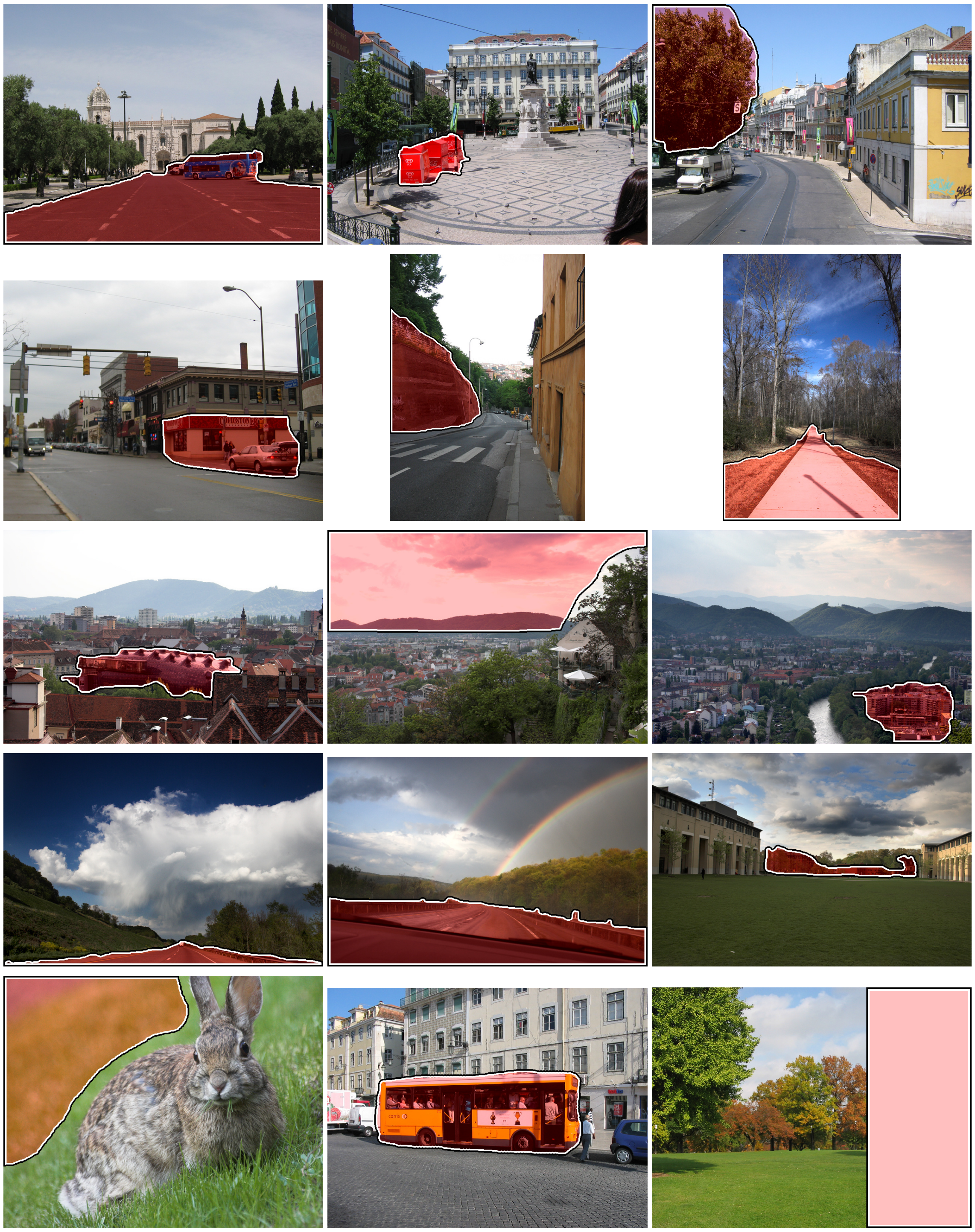


Figure 3.11: The input photos for Figure 3.10. The regions to be replaced are highlighted in red.

ten seconds, it is unlikely that an unsuspecting, casual observer would notice anything wrong with the image. After ten seconds of examination, participants have marked our algorithm's results as fake only 34% of the time (the other 66% are either undecided or have marked the image as real already). For Criminisi et al. participants have marked 69% of the images as fake by ten seconds. For real photographs, only 3% have been marked as fake. All pairwise differences are statistically significant ($p < 0.001$).

3.6 Discussion

This work approaches image completion from an entirely new direction – orthogonal and complementary to the existing work. While previous algorithms [18, 26, 28, 115] suggest clever ways to reuse visual data within the source image, we demonstrate the benefits of utilizing semantically valid data from a large collection of unlabelled images. Our approach successfully fills in missing regions where prior methods, or even expert users with the Clone brush, would have no chance of succeeding because there is simply no appropriate image data in the source image to fill the hole. Likewise, expert users would have trouble leveraging such a large image collection – it would take 10 days just to view it with one second spent on each image. Additionally, this is the first study in the field of image completion to undertake a full perceptual user study and report success rates on a large test set. While the results suggest substantial improvement over previous work, image completion is extremely difficult and far from solved. Given the complimentary strengths of our method and single-image techniques, a hybrid approach is likely to be rewarding.

It takes a large amount of data for our method to succeed. We saw dramatic improvement when moving from ten thousand to one million images. But our database is still tiny in comparison with the billions of high quality photographs available on sites like Picasa or Flickr. As we are able to leverage more data we expect the algorithm to perform better.

Because our algorithm requires a large amount of data it would be logistically challenging to distribute and use as a standalone desktop application. Our algorithm would be better suited to a web-based interface, where a user would submit an incomplete photo and a remote service would search a massive database, in parallel, and return results.

Beyond the particular graphics application, the deeper question for all appearance-based data-driven methods is this: *would it be possible to ever have enough data to represent the entire visual world?* Clearly, attempting to gather all possible images of the world is a futile task, but what about collecting the set of all *semantically differentiable scenes*? That is, given any input image can we find a scene that is “similar enough” under some metric? The

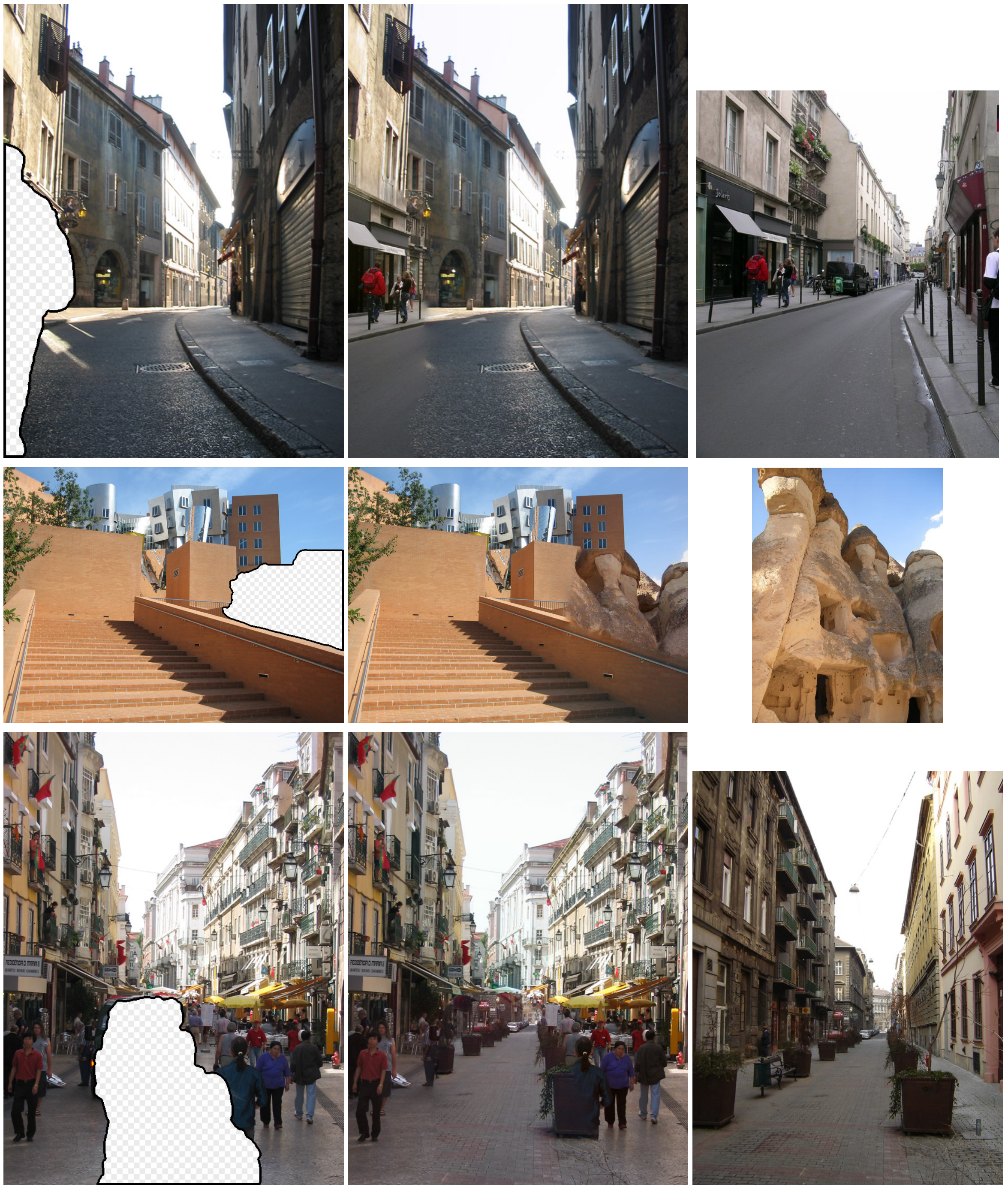

Input Output Scene Match

Figure 3.12: Typical failure cases. Some results exhibit pronounced texture seams (top). Others are failures of scene matching (middle). The last failure mode (bottom), shared with traditional image completion algorithms, is a failure to adhere to high-level semantics (e.g. entire people).

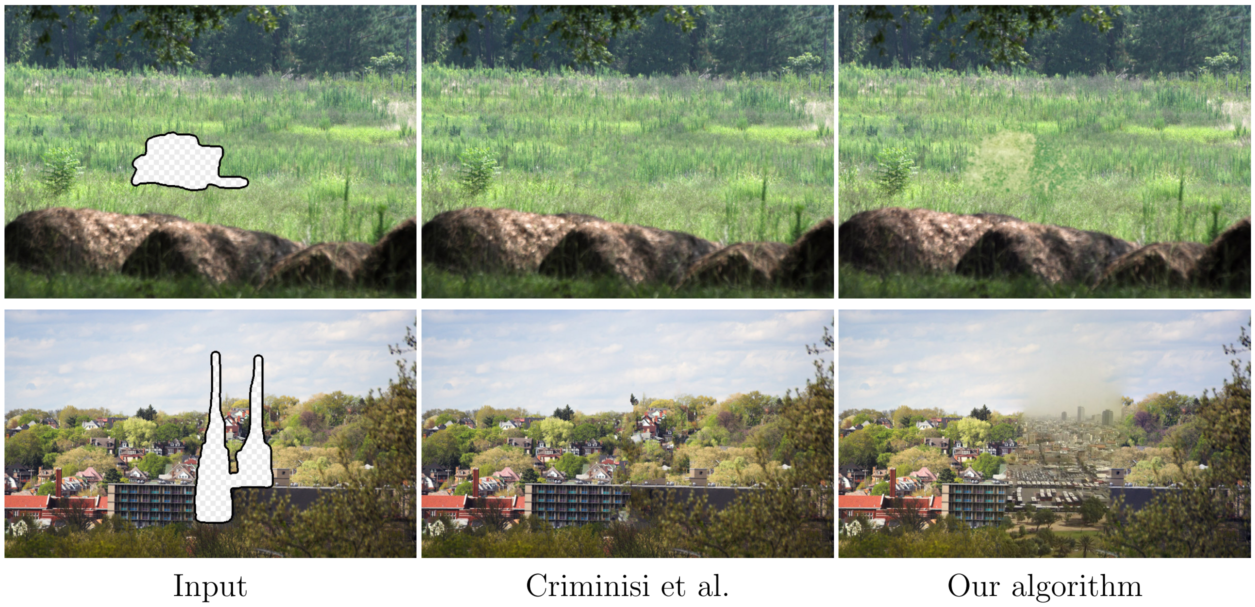


Figure 3.13: Cases where existing image completion algorithms perform better than our algorithm.

truly exciting (and surprising!) result of our work is that not only does it seem possible, but the number of required images might not be astronomically large. This study, along with work by Torralba et al. [100], suggest the feasibility of sampling from the entire space of scenes as a way of exhaustively modelling our visual world. This, in turn, might allow us to "brute force" many currently unsolvable vision and graphics problems.

Acknowledgments: We would like to thank Antonio Torralba for the gist code and useful comments and Gabriel Brostow for sharing their hole-filling software. We also thank Tamara Berg, Alex Berg, and Rob Fergus for sharing their image downloading scripts.

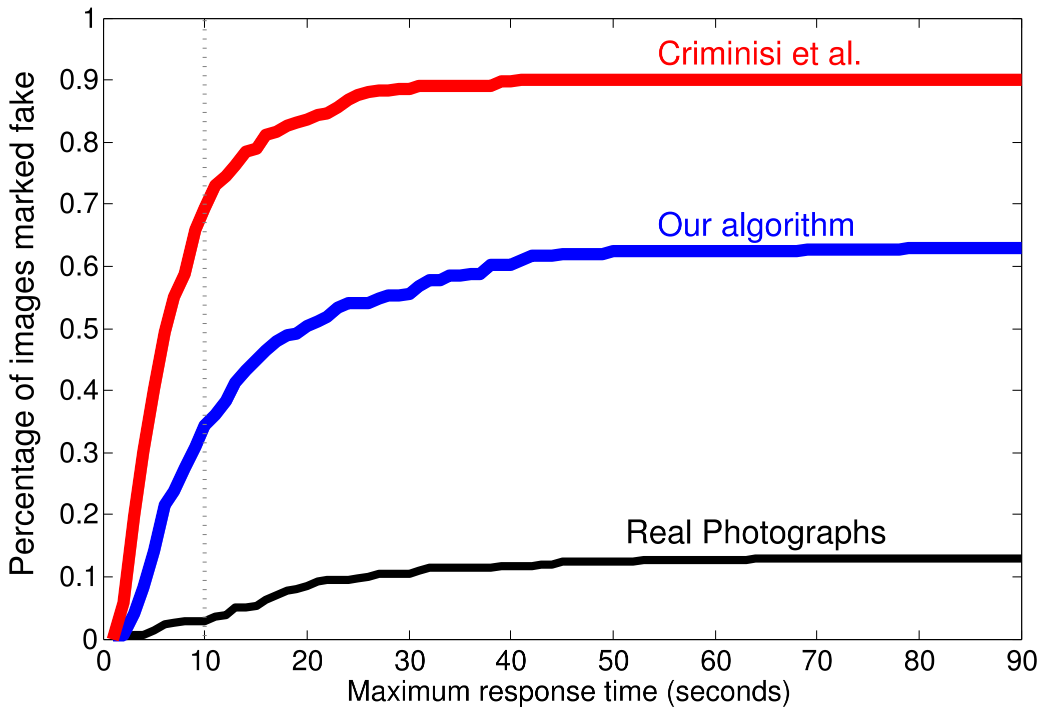


Figure 3.14: For any amount of viewing time, the percentage of images that have been marked as fake by participants in our perceptual study.

Chapter 4

Image Geolocation

Chapter Summary. We propose a simple algorithm for geolocating a single image using a purely data-driven scene matching approach. We leverage a dataset of over 6 million GPS-tagged images from the Internet. We represent the estimated image location as a probability distribution over the Earth's surface. We quantitatively evaluate our approach in several geolocation tasks and demonstrate encouraging performance. Additionally we consider the case of geolocating a sequence of timestamped photographs. We show that geolocation estimates can provide the basis for numerous other image understanding tasks such as population density estimation, land cover estimation or urban/rural classification. This work was described, in part, in "im2gps" [39] and "image sequence geolocation" [50], a collaboration with Evangelos Kalogerakis, Olga Vesselova, and Aaron Hertzmann.

4.1 Introduction

Consider the photographs in Figure 4.1. What can you say about where they were taken? The first one is easy – it's an iconic image of the Notre Dame cathedral in Paris. The middle photo looks vaguely Mediterranean, perhaps a small town in Italy, or France, or Spain. The rightmost photograph is the most ambiguous. Probably all that could be said is that it's a picture of a seaside in some tropical location. But note that even this vague description allows us to disregard all non-coastal, non-tropical areas – more than 99.9% of the Earth's surface! Evidently, we humans have learned a reasonably strong model for inferring location distribution from photographs. Moreover, even in cases when our geo-

[1]Project Page: http://graphics.cs.cmu.edu/projects/im2gps/

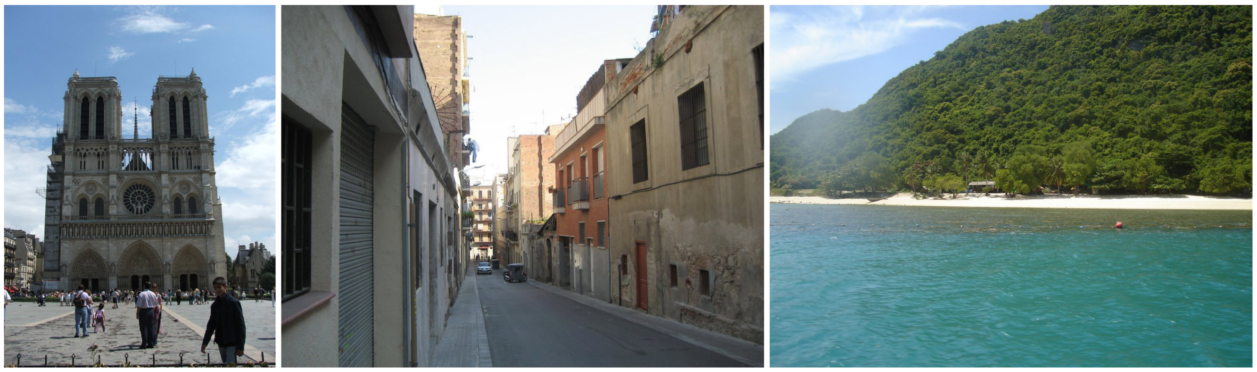


Figure 4.1: What can you say about where these photos were taken?

localization performance is poor, we are still able to give fairly confident estimates to other related questions: How hot/cold does it get? How many people live there? How well-off are they? etc.

What explains this impressive human ability? Semantic reasoning, for one, is likely to play a big role. People's faces and clothes, the language of the street signs, the types of trees and plants, the topographical features of the terrain – all can serve as semantic clues to the geographic location of a particular shot. Yet, there is mounting evidence in cognitive science that *data association* (ask not "What is it?" but rather "What is it *like*?") may play a significant role as well [5]. In the example above, this would mean that instead of reasoning about a beach scene in terms of the tropical sea, sand and palm trees, we would simply remember: "I have seen something similar on a trip to Hawaii!". Note that although the original picture may not actually be from Hawaii, this association is still extremely valuable in helping to implicitly define the *type* of place that the photo belongs to.

Of course, computationally we are quite far from being able to semantically reason about a photograph (although encouraging progress is being made). On the other hand, the recent availability of truly gigantic image collections has made data association, such as brute-force scene matching, quite feasible [38, 100].

In this study, we propose an algorithm for estimating a distribution over geographic locations from an image using a purely data-driven scene matching approach. For this task, we leverage a dataset of over 6 million GPS-tagged images from the Flickr online photo collection. We represent the estimated image location as a probability distribution over the Earth's surface, and geolocation performance is analyzed in several tasks. Additionally, the usefulness of image localization is demonstrated with meta-tasks such as land cover estimation and urban/rural classification.

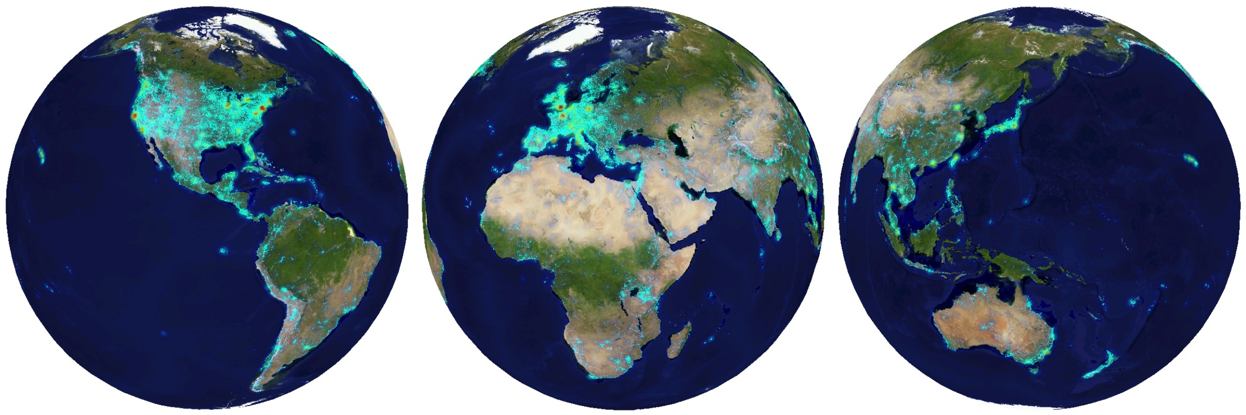


Figure 4.2: The distribution of photos in our database. Photo locations are cyan. Density is overlaid with the "jet" color map (log scale).

4.1.1 Background

Visual localization on a topographical map has been one of the early problems in computer vision, which turned out to be extremely challenging for both computers and humans [98]. But the situation improves dramatically if more sources of data are available. Jacobs et al. [43] proposes a very clever and simple method of geolocating a webcam based on correlating its video-stream with satellite weather maps over the same time period.

The recent availability of GPS-tagged images of urban environments coupled with advances in multi-view geometry and efficient feature matching led to a number of groups developing place recognition algorithms, some of which competed in the "Where am I?" Contest [97] at ICCV'05 (winning entry described in [118]). Similar feature-based geometric matching approaches have also been successfully applied to co-registering online photographs of famous landmarks for browsing [90] and summarization [87], as well as image retrieval in location-labeled collections, e.g. [14]. Landmark photos are linked to Wikipedia photos and articles within a specified city in [80]. Since the publication of im2gps, [17] and [119] have attacked the global landmark recognition problem, in the latter case scaling up to thousands of landmarks with high accuracy.

But can these geometric feature-based matching approaches scale up to all photos of world? This is unlikely in the near future, not just because of computational cost, but simply because the set of all existing photographs is still not large enough to exhaustively sample the entire world. Yes, there are tens of thousands of photos of a many landmarks, but some ordinary streets or even whole cities might be entirely missing. Even with a dense visual sample, much of the world is too self-similar (e.g. the 300 thousand square kilometers of corn fields in the US). Clearly, a generalization of some sort is required.

On the other side of the spectrum is the philosophy that all forests look more or less the same, as do deserts, mountains, cities, kitchens, bathrooms, etc. A large body of work exist on scene recognition [62,74,82,108], which involves defining a handful of scene categories and using various low-level features to classify a novel image into one of these categories. While impressive results are typically obtained, classification is not a difficult task if the number of categories is small. Moreover, the choice of categories is often not very scientific.

The approach we are proposing in this study neatly straddles these two extremes, seamlessly adapting to the amount of data available. If the query image is a famous landmark, there will likely be many similar images of the same exact place in the database, and our approach is likely to return a precise GPS location. If the query is more generic, like a desert scene, many different deserts could match, producing a location probability that is high over the dry, sandy parts of the world. In fact, our approach provides a more scientifically valid method of defining scene categories – based on geographic location as well as appearance.

4.2 Building a Geo-tagged Image Dataset

In order to reason about the global location of an arbitrary scene we first need a large number of images that are labelled with geographic information. This information could be in the form of text keywords or it could be in the form of GPS coordinates. Fortunately there is a huge (and rapidly growing) amount of online images with both types of labels. For instance, Flickr.com has hundreds of millions of pictures with either geographic text or GPS coordinates.

But it is still difficult to create a useful, high-quality database based on user collected and labelled content. We are interested in collecting images that depict some amount of geographic uniqueness. For instance, pictures taken by tourists are ideal because they often focus on the *unique* and *interesting* qualities of a place. Many of these images can be found because they often have geographic keywords associated with them (i.e. city or country names). But using geographic text labels is problematic because many of them are ambiguous (e.g. Washington city/state, Georgia state/country, Mississippi river/state, and LA city/state) or spatially broad (e.g. Asia or Canada).

Images annotated with GPS coordinates are geographically unambiguous and accurate, but are more likely to be visually irrelevant. Users tend to geo-tag all of their pictures, whether they are pet dog pictures (less useful) or hiking photos (more useful). In fact, the vast

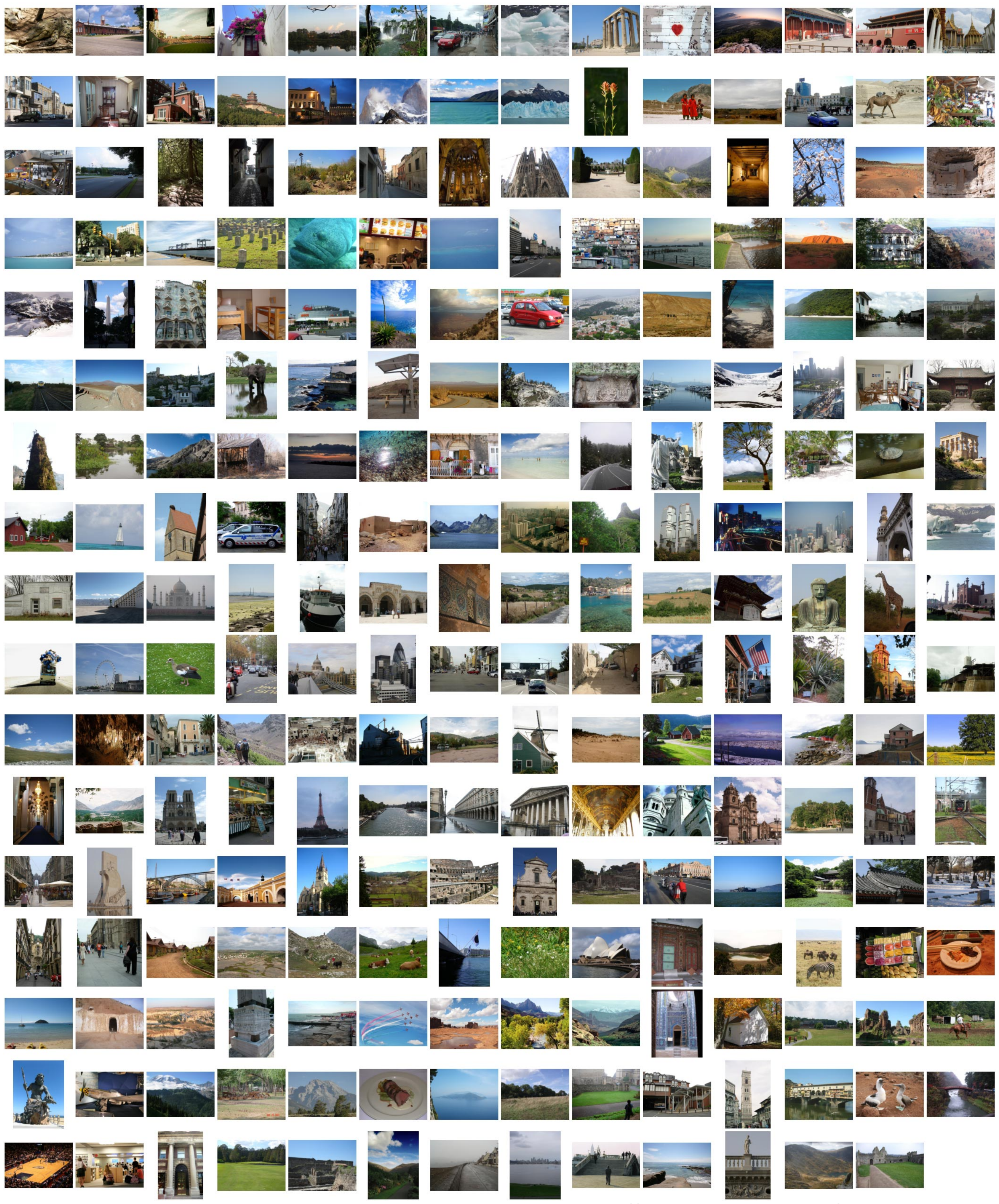


Figure 4.3: The 237 image im2gps test set. Note how difficult it is to specifically geolocate most of the images.

majority of online images tagged with GPS coordinates and to a lesser extent those with geographic text labels are not useful for image-based geolocation. Many of the images are poor quality (low resolution, noisy, black and white) or depict scenes which are only marginally useful for geolocation (most portraits, wedding pictures, abstracts, and macro photography). While these types of photos can sometimes reveal geographic information (western-style weddings are popular in Europe and Japan but not India; pet dogs are popular in the US but not Syria) the customs are so broadly distributed that it is not very useful for geolocation.

However, we found that by taking the intersection of these groups, images with both GPS coordinates and geographic keywords, we greatly increased the likelihood of finding accurately geolocated *and* visually useful data. People may geo-tag images of their cats, but they're less likely to label that image with "New York City" at the same time. Our list of geographic keywords includes every country and territory, every continent, the top 200 most populated cities in the world, every US state, and popular tourist sites (e.g. "Pisa", "Nikko", "Orlando").

This results in a pool of approximately 20 million geotagged and geographic text-labelled images from which we excluded all photos which were also tagged with keywords such as "birthday", "concert", "abstract" and "cameraphone". In the end we arrived at a database of $6,472,304$ images. All images were downsized to max dimension 1024 and JPEG compressed for a total of 1 terabyte of data.

While this is a tremendous amount of data it cannot be considered an exhaustive visual sampling of Earth. Our database averages only 0.0435 pictures per square kilometer of Earth's land area. But as figure 4.2 shows the data is very non-uniformly distributed towards places where people live or travel which is fortunate since geolocation query images are likely to come from the same places.

4.2.1 Evaluation Test Set

To evaluate the performance of our method, we also need a separate, held-out test set of geo-located images. We built the test set by drawing 400 random images from the original data set. From this set we manually removed the types of undesirable photos that we tried to excluded during database construction – abstract photos, overly processed or artistic photos, and black and white photos. We also excluded photos with significant artifacts such as motion blur or extreme noise. Finally we removed pictures with easily recognizable people or other situations that might violate someone's privacy. To ensure that our test set and database are independent we exclude from the database not just test images, but all other images from the same photographers.

Of the 237 resulting images, about 5% are recognizable as specific tourist sites around the globe but the great majority are only recognizable in a generic sense (See Figure 4.3). Some of the images contain very little geographic information, even for an astute human examiner. We think this test set is extremely challenging but representative of the types of photos people take.

4.3 Scene Matching

Is it feasible to extract geographic information from generic scenes? One of the main questions addressed by this study is as much about the Earth itself as it is about computer vision. Humans and computers can recognize specific, physical scenes that they've seen before, but what about more generic scenes which make up most of our database and our test set. Many of these scenes may be impossible to specifically localize. We know that our world is self-similar not just locally but across the globe. Film creators have long taken advantage of this (e.g. "Spaghetti Westerns" films that were ostensibly set in the American Southwest but filmed in Almería, Spain.) Nonetheless, it must be the case that certain visual features in imagery correlate strongly with geography even if the relationship is not strong enough to specifically pinpoint a location. Beach images must be near bodies of water, jungles must be near the equator, and glaciated mountains cover a relatively small fraction of the Earth's surface.

What features can we extract from images that will best allow us to examine and exploit this correlation between image properties and geographic location? In this study we evaluate an assortment of popular features from literature:

Tiny Images: The most trivial way to match scenes is to compare them directly in color image space. Reducing the image dimensions drastically makes this approach more computationally feasible and less sensitive to exact alignment. This method of image matching has been examined thoroughly by Torralba et al. [100] for the purpose of object recognition and scene classification. Inspired by this work we will use 16 by 16 color images as one of our features.

Color histograms: In the spirit of most image retrieval literature, we build joint histograms of color in CIE L*a*b* color space for each image. Our histograms have 4, 14, and 14 bins in L, a, and b respectively for a total of 784 dimensions. We have fewer bins in the intensity dimension because other descriptors will measure the intensity distribution of each image.

We compute distance between these histograms using χ^2 distance.

Texton Histograms: Texture features might help distinguish between geographically correlated properties such ornamentation styles or building materials in cities or vegetation and terrain types in landscapes. We build a 512 entry universal texton dictionary [67] by clustering our dataset's responses to a bank of filters with 8 orientations, 2 scales, and 2 elongations. For each image we then build a 512 dimensional histogram by assigning each pixel's set of filter responses to the nearest texton dictionary entry. Again, we use χ^2 distances between texton histograms. Note that this representation is quite similar to dense visual words.

Line Features: We have found that the statistics of straight lines in images are useful for distinguishing between natural and man-made scenes and for finding scenes with similar vanishing points. We find straight lines from Canny edges using the method described in Video Compass [55]. For each image we build two histograms based on the statistics of detected lines- one with bins corresponding to line angles and one with bins corresponding to line lengths. We use L1 distance to compare these histograms.

Gist Descriptor + Color: The gist descriptor [72] has been shown to work well for scene categorization [74] and for retrieving semantically and structurally similar scenes [38]. We create a gist descriptor for each image with 5 by 5 spatial resolution where each bin contains that image region's average response to steerable filters at 6 orientations and 4 scales. We also create a tiny L*a*b image, also at 5 by 5 spatial resolution.

Geometric Context: Finally, we compute the geometric class probabilities for image regions using the method of Hoiem et al. [40]. We use only the primary classes- ground, sky, and vertical since they are more reliably classified. We reduce the probability maps for each class to 8x8 and use L2 distance to compare them.

We precomputed all features for the database which took about 15 seconds per image on a contemporary Xeon processor for a total of 3.08 CPU years. Using a cluster of 400 processors we computed the features over 3 days.

4.4 Data-driven Geolocation

After all the preprocessing is complete, the geolocation framework is quite simple. For each input image in our test set we build the same features as discussed in Section 4.3 and compute the distance in each feature space to all 6 million images in the database. We scale each feature's distances so that their standard deviations are roughly the same and thus they influence the ordering of scene matches equally. For each query image we use the aggregate feature distances to find the nearest neighbors in the database and we derive geolocation estimates from those GPS tagged nearest neighbors.

To produce a geolocation estimate, the simplest heuristic is to take the GPS coordinate of the first nearest neighbor (1-NN) scene match. Of course, 1-NN approaches are often not robust. Alternatively, we can consider a larger set of k-NN ($k = 120$ in our experiments). This set of nearest neighbors together forms an implicit estimate of geographic location – a probability map over the entire globe. The hope is that the location of peak density in this probability map corresponds to the true location of the query image. One way to operationalize this is to consider the modes of the distribution by performing mean-shift [15] clustering on the geolocations of the matches. We represent the geolocations as 3d points and re-project the mean-shift clusters to the Earth's surface after the clustering procedure. We use a mean-shift bandwidth of $500km$ (although other settings work similarly) and disregard clusters with fewer than 4 matches, resulting in between 6 to 12 clusters containing about two thirds of the original 120 matches. The clustering serves as a kind of geographic outlier rejection to clean up spurious matches, but can be unfavorable to locations with few data-points.

To compute a geolocation estimate, one approach is to pick the cluster with the highest cardinality and report the GPS coordinate of its mode. For some applications, it might be acceptable to return a list of possible location estimates, in which case the modes of the clusters can be reported in order of decreasing cardinality. We show qualitative results for several images in Figure 4.8. More results can be found on our project web page.

4.4.1 Is the data helping?

The most interesting research question for us is *how strongly does image similarity correlate with geographic proximity*? To geolocate a query we don't just want to find images that are similarly structured or of the same semantic class (e.g. "forest" or "indoors"). We want image matches that are specific enough to be geographically distinct from otherwise similar scenes. How much data is needed start to capture this geography-specific information? In Figure 4.4 we plot how frequently the 1-NN scene match is within $200km$ of the query's true location as we alter the size of the database. With a tiny database of 90 images, the 1-NN scene match is as likely to be near the query as a random image from the database. With the full database we perform 16 times better than chance.

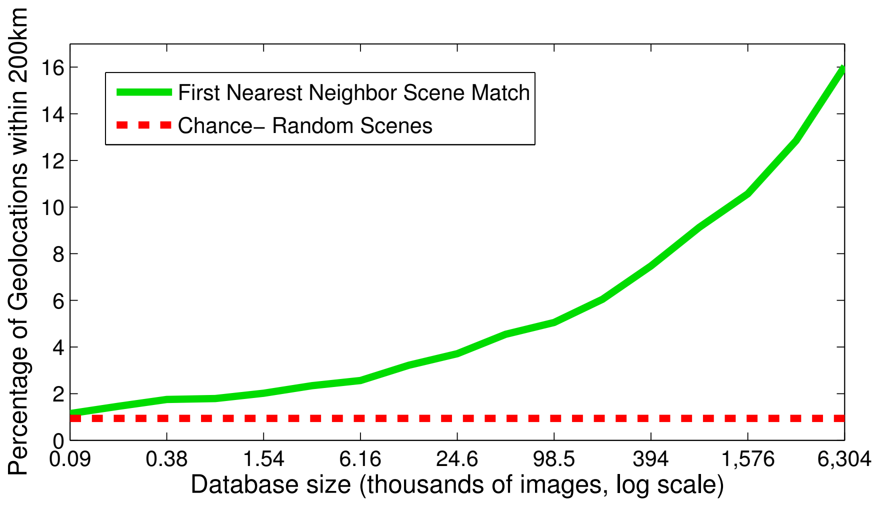


Figure 4.4: *Geolocation performance across database sizes.* Percentage of test set images that were correctly geolocated within 200km of ground truth as function of dataset size using 1-NN. As the database shrinks the performance converges to chance.

4.4.2 Which features are most geo-informed?

Another interesting question we consider is which *visual characteristics are more helpful in disambiguating between locations*? In Figure 4.5 we examine the geolocation accuracy when using each of the features from Section 4.3 in isolation as well as in unison. For each feature we consider the geolocation accuracy of 1-NN as well the largest cluster. The latter is indeed more robust than using 1-NN, although performance is similar when using all features. The richer feature combinations seem less prone to geographic outliers which disrupt the 1-NN approach.

Using all features together performed considerably better than any one by itself, suggesting that the information they are capturing is somewhat independent. The most geographically discriminative features are the gist, color histogram, and texton histogram. The gist, especially, performs well, even in the 1-NN regime. Surprisingly, color also does extremely well (but only after discarding geographic outliers), which suggests that it is a more diverse and location-specific feature than previously assumed (artists have long talked about "that special color" of a particular location). The least geographically discriminative feature is the 8x8 geometric context class likelihoods. This seems reasonable – the geometric context framework is inspired by the observation that the vast majority of scenes can be succinctly modelled by ground, sky, and vertical components. A view down a forest path can share the same class distribution as a view down Wall St. (when considering only the primary

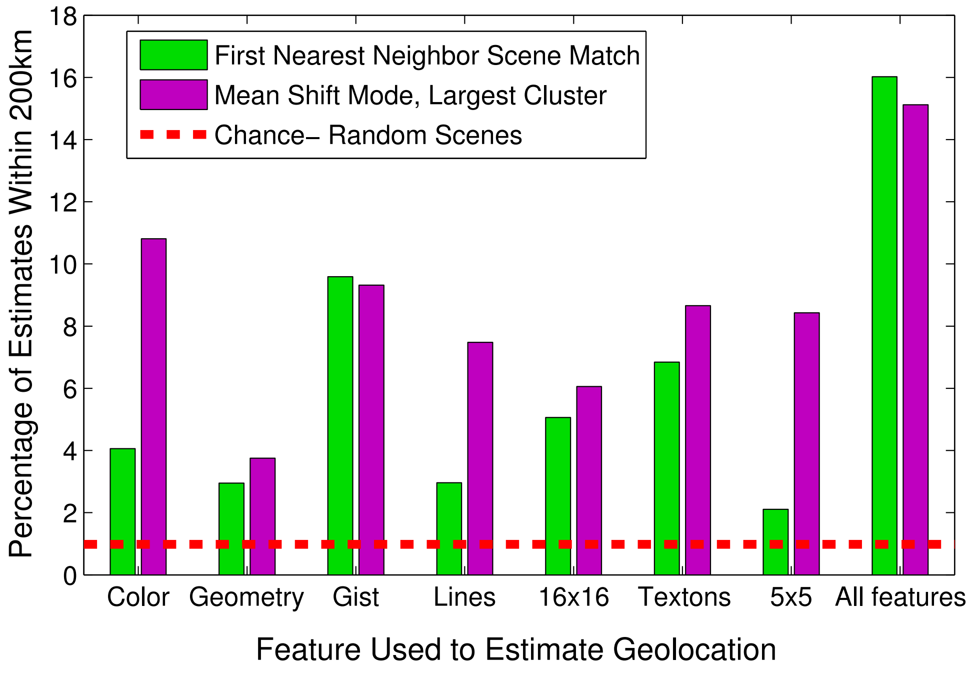


Figure 4.5: *Geolocation performance across features.* Percentage of test cases geolocated to within 200km for each feature. We compare geolocation by 1-NN vs. largest mean-shift mode.

geometric classes). The 16x16 tiny images also scored low. In fact, after geographic outlier rejection, they performed worse than humble 5x5 color images, suggesting that perhaps they are too noisy for this task. In the rest of the evaluations, we will use all features except the geometric context and the 16x16 tiny images.

4.4.3 How accurate are the estimates?

Given a photo, how often can we pin-point the right city? Country? Continent? So far we have evaluated geolocation accuracy only in terms of a distance threshold. In Figure 4.6 we more closely examine the distribution of geolocation errors (distances between estimated and ground truth locations) across our test set. For the two heuristics (1-NN, and mean-shift mode) plus three baselines (chance, and two best case scenarios), we sort the errors on the test set independently, from best to worst. We see that both heuristics are able to localize about 25% of the data within the scale of a (small) country. The 1-NN approach performs better at precise localization (within a city). Both methods outperform chance by a large

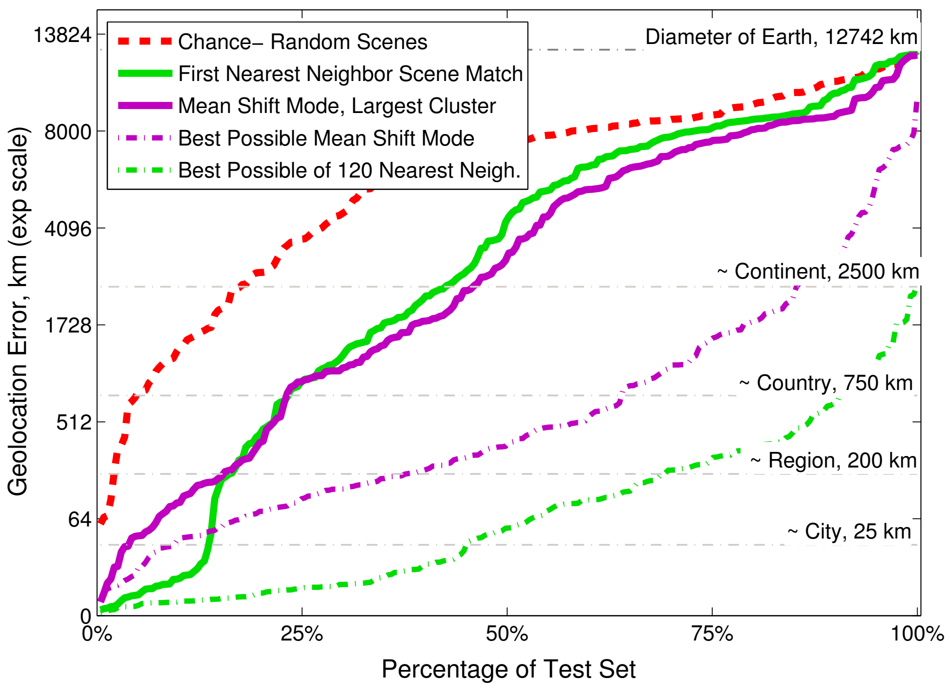


Figure 4.6: *Accuracy of geolocation estimates across the test set.* Localization errors (distance between predicted and ground truth location), are shown for 1-NN and mean-shift estimates. Errors are sorted from best to worst independently for each curve, thus showing the proportion of images geolocated within any error threshold. Chance performance (random matches) and two best-case scenarios – picking the mean-shift mode or scene match which is spatially closest to the ground truth query location – are shown for comparison.

margin.

It is instructive to note that although the largest one-third of errors across the test set are very large (nearly as bad a chance), for almost all queries there is *some* scene match or mean-shift cluster that is quite close to the query (see “best case” curves on Figure 4.6). In other words, among the 120 nearest neighbors there are almost always several geographically accurate matches but the heuristics sometimes have trouble disambiguating those from other visually similar, spatially dissimilar matches (e.g. Hawaii vs Martinique or New York City vs Hong Kong). If we allow ourselves N “guesses” as to the location of a query we can rapidly get closer to the ground truth location (Figure 4.7). For this task we compare geolocation estimates from N nearest neighbors and the modes of the N largest clusters. The probability map modes are especially accurate for this task. With 6 guesses for each query, the median error for the test set is less than $500km$ (nearly half the error of 6-NN, and one quarter the error of 6 random scenes).

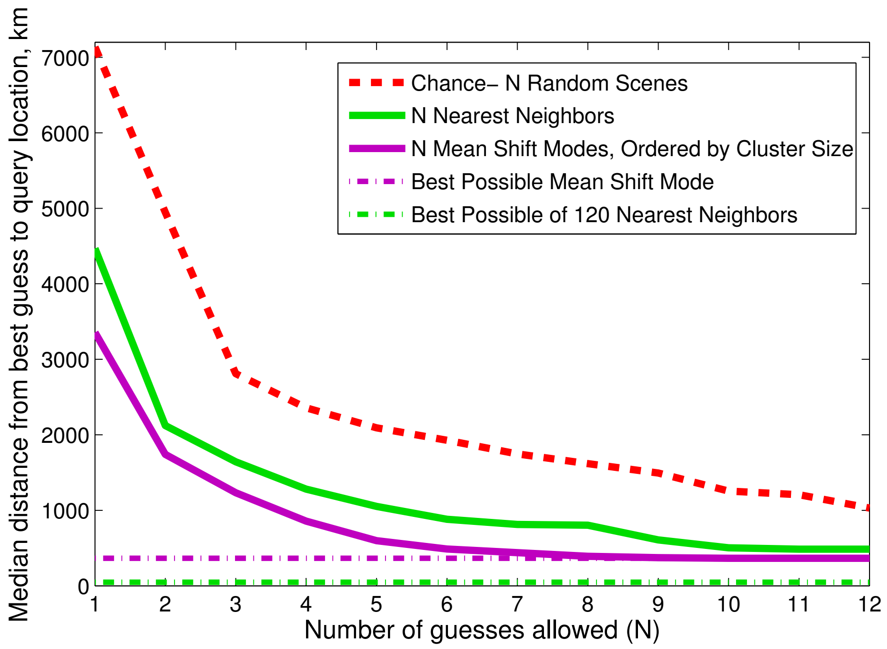


Figure 4.7: *Geolocation error with multiple guesses.* Median geolocation error for NN and mean-shift modes with increasing numbers of guesses allowed. The error is the distance from an algorithm's best guess to the query's ground truth location. Although the geolocation of a query may be ambiguous among several possibilities, after multiple guesses it is likely that one of the estimates is near the ground truth location.

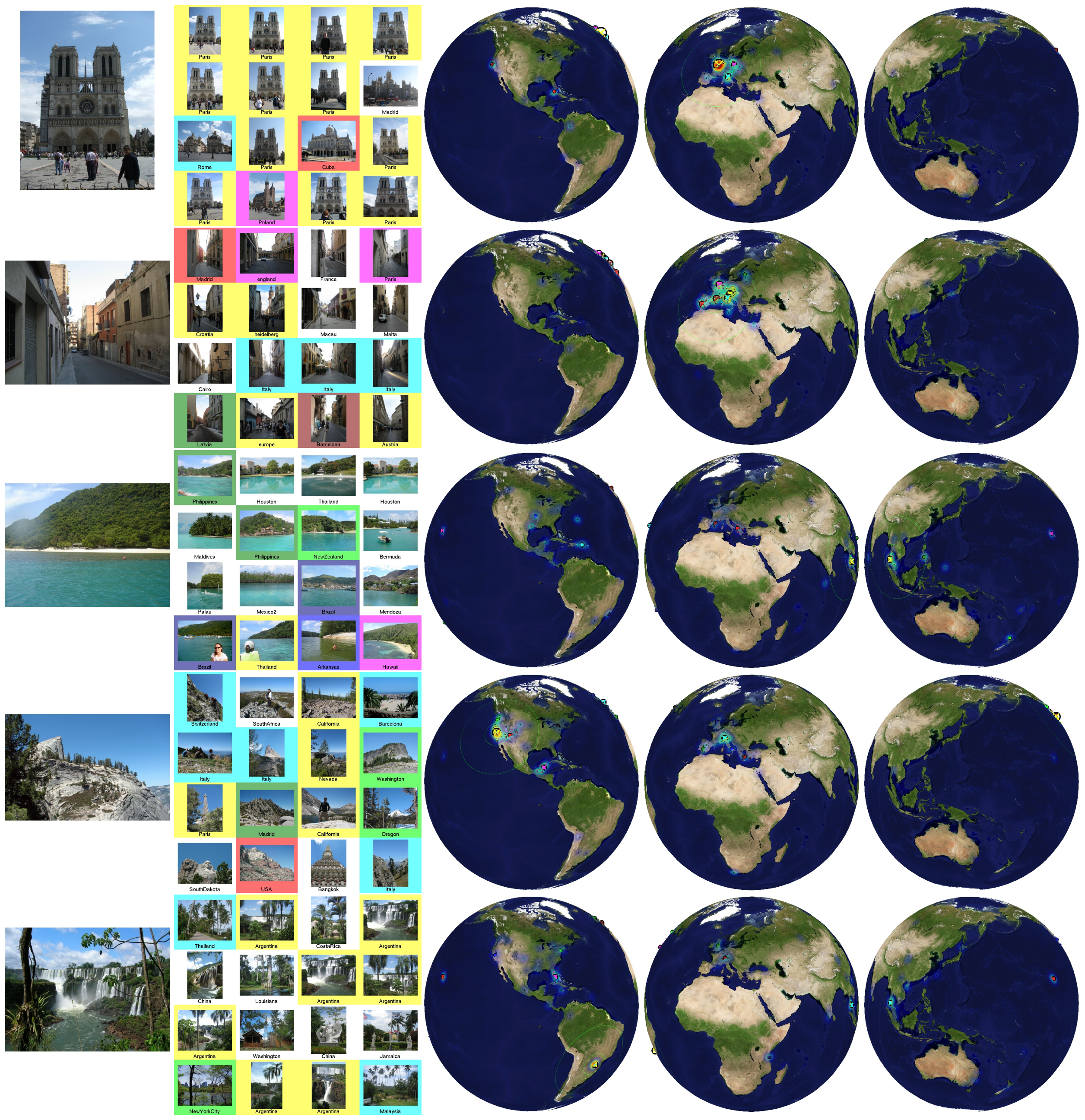


Figure 4.8: *Results 1.* From left to right: query images, nearest neighbors, and three visualizations of the estimated geolocation probability map. The probability map is shown as a jet-colorspace overlay on the world map. Cluster modes are marked with circumscribed "X"'s whose sizes are proportional to cluster cardinality. If a scene match is contained in a cluster it is highlighted with the corresponding color. The ground truth location is a cyan asterisk surrounding by green contours at radii of 200km, 750km, and 2500km. From top to bottom, these photos were taken in Paris, Barcelona, Thailand, California, and Argentina.

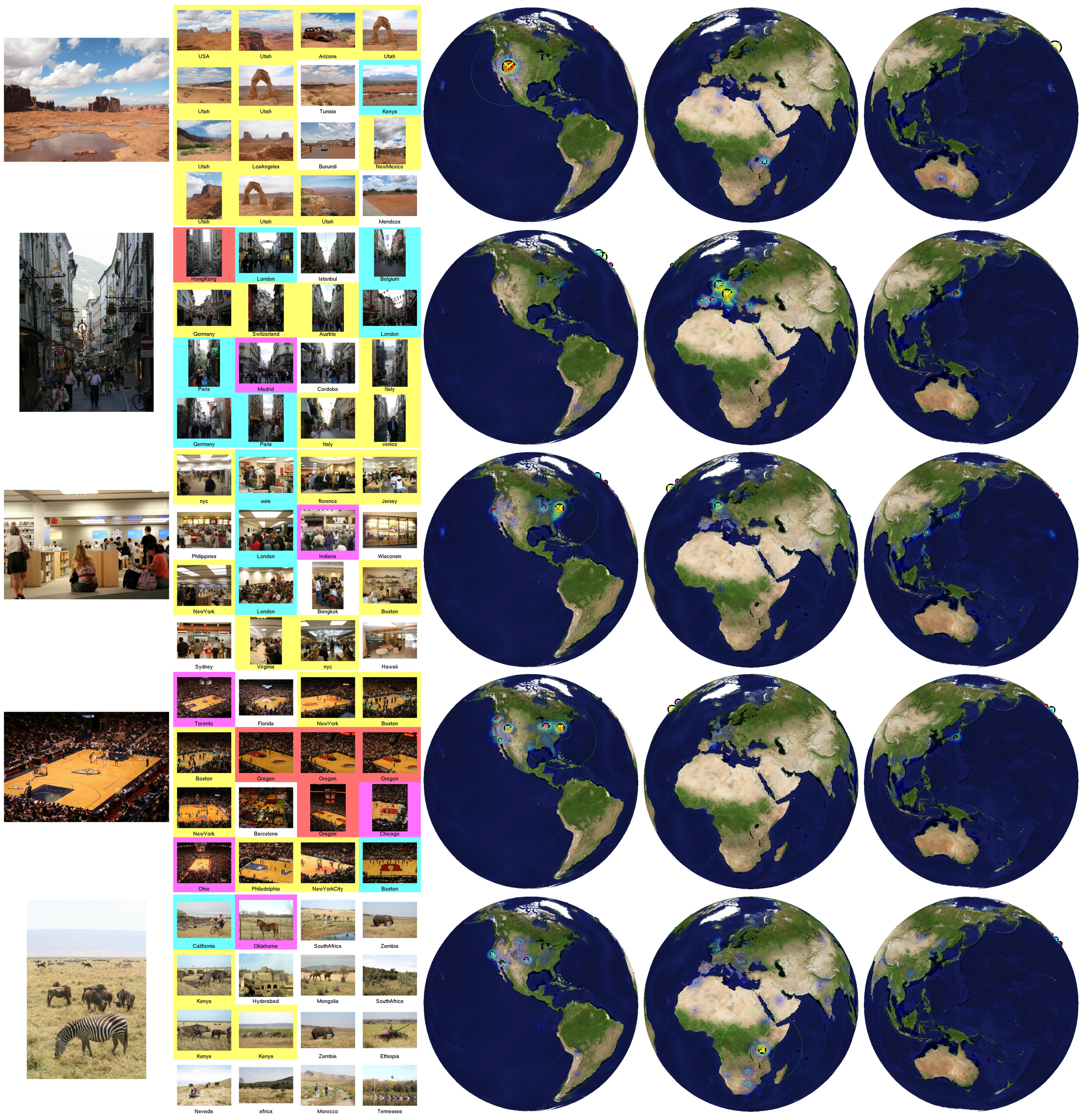


Figure 4.9: *Results 2.* From top to bottom, these photos were taken in Utah, Germany, New York City, New York City, and Tanzania. The Apple Store in Manhattan is a surprisingly popular location for geo-tagged photographs, possibly due to the GPS-enabled iPhone.

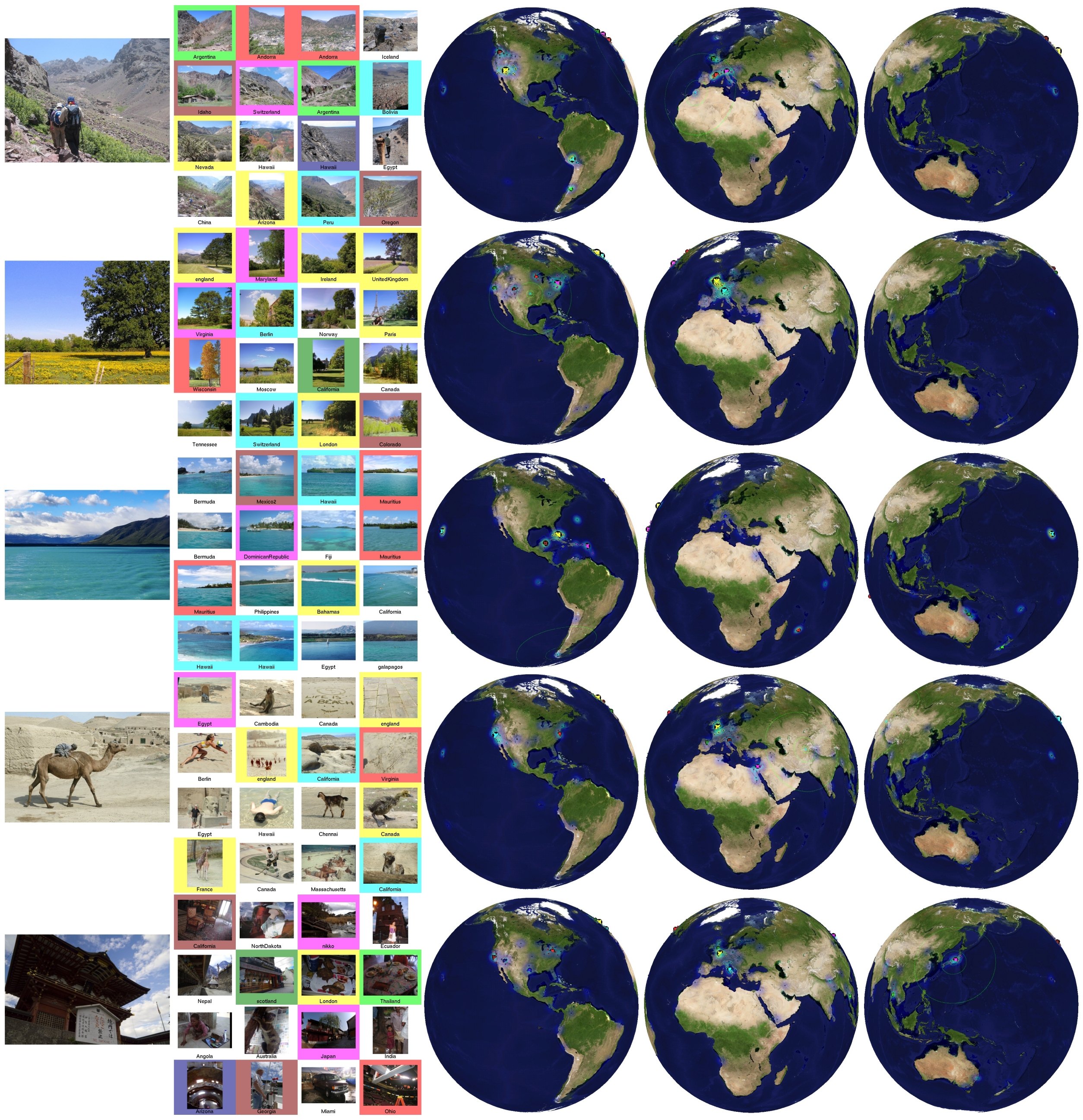


Figure 4.10: *Results 3.* In all of these cases the geolocation estimate was quite poor. From top to bottom, these photos were taken in Morocco, Oklahoma, Argentina, Afghanistan, and Japan. In the top two cases, the scene matches are fairly good, but the scenes are non-distinctive and there are not enough geolocated reference photos in Morocco or Oklahoma. In the middle photo, the matched scenes are largely tropical locations even though the photo is taken in frigid, Southern Argentina. In the bottom two photographs, the scene matches are poor.

4.5 Why Does it Work? Deeper Performance Analysis

This section was added after the initial publication of im2gps. Using the im2gps test set, we have found that the first nearest neighbor scene has a 16% chance of being within 200km of a query. The mode of the largest geographic cluster has a 15% chance of being within 200km. In this section we examine what factors account for the this performance and try to measure their influence.

4.5.1 Measuring performance without geographic bias.

> *The peaked geographic distribution of the data accounts for the performance that we see.* – Dan Huttenlocher, Boris Epshtein, Slashdot user comments, CSD Immigration course attendees

The most common concern about im2gps, stated above, is that since the geographic distribution of data appears peaked in relatively few places (Figure 4.2), our performance could be a result of random guessing. In fact, the chance that two photos are within 200km in the im2gps database is about 1.2%. For our test set of 237 images randomly sampled from the database chance is 0.97%. For individual test cases, chance ranges from less than 0.01% in Libya, Colombia, and Greenland to 4.9% near London. That is to say, 4.9% percent of the im2gps database is within 200km of London. For other cities the values are: New York City 4.3%, San Francisco 3.1%, Paris 2.8%, Chicago 1.9%, Tokyo 1.8%, and Barcelona 1.5%.

How would im2gps perform if the test set distribution wasn't geographically peaked? To quantitatively evaluate this issue we define a new *geographically uniform* test set. We tessellate the globe into quadrilateral regions roughly 400km on edge (Figure 4.11). We take one query from each of the 955 regions in that have at least ten photographs. Chance is an order of magnitude lower for this database – only 0.13%.[1] Figure 4.12 shows the geographic distribution of the test set, as well as the geolocation accuracy for each photo. Large regions of South America, Africa, and Central Asia have no accurate geolocations. Overall, for 2.3% of the photos the first nearest neighbor is within 200km of the query photo's location, and only 4.15% within 750km. Interestingly, relative to chance, this is just as much of an improvement as on the im2gps test set ($\sim$ 16 times better).

[1]This value was calculated by counting the number of database photos close enough to each query in the test set. Alternatively, each geolocation guess has an area of 126,663km^2 and the land area of the Earth is $148,940,000km^2$, suggesting that a truly uniform test set would have a chance guessing accuracy of 0.084%. Chance is higher for our test set because our database (and thus test set) contain no photographs in some regions of Siberia, Sahara, and Antarctica.

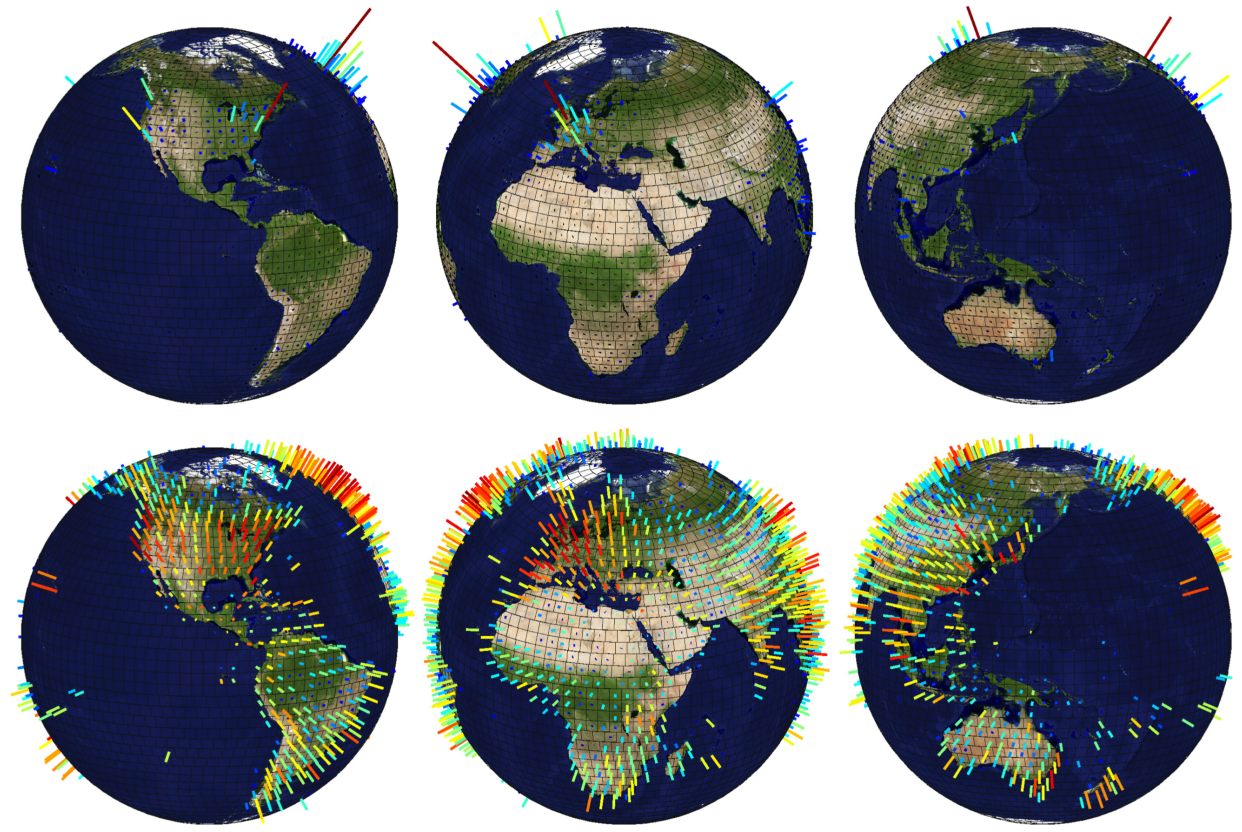


Figure 4.11: *Photo density in im2gps database*, linear scale (top) and natural log scale (bottom). The height of each bar is proportional to the density of geotagged photos in each equal area region. The bars are colored according to the Matlab "jet" color scheme. Regions with zero photos have no bar.

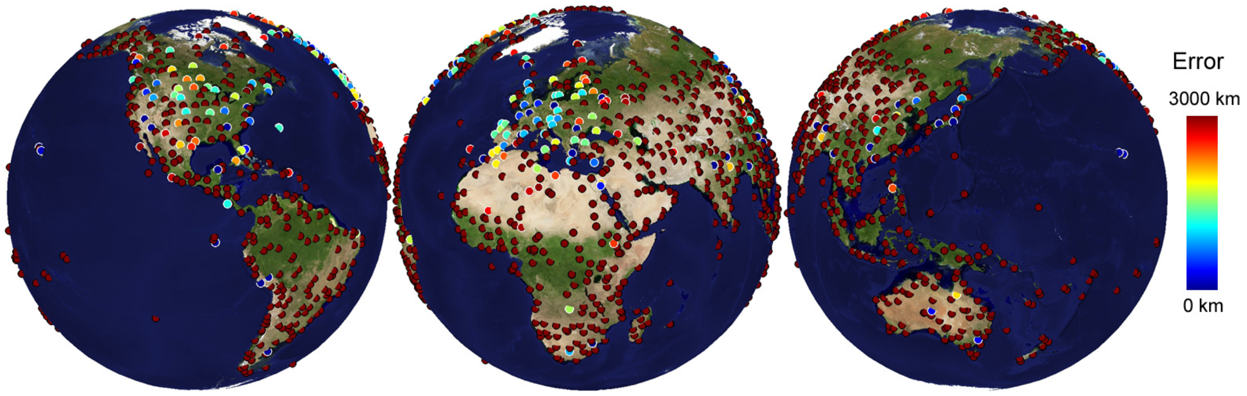


Figure 4.12: *Accuracy on Geographically Uniform Test Set.* For each photo in the test set, the marker color indicates how accurately the photo was geolocated.

For this data set, using the largest mean shift mode is less geographically accurate than the first nearest neighbor. With a fixed bandwidth for all parts of the globe, scene matches that are found in sparse areas will not be grouped together. For example, if there are 4 scene matches in England and 20 in South America, the England matches will most likely form a larger cluster than any in South America. As an alternative to mean shift we tried geographically clustering the matches using a Gaussian Mixture Model (GMM). GMM is appealing because its clusters can significantly vary in spatial density. The clustering is initialized with k-means, then an Expectation Maximization (EM) procedure optimizes the model parameters. We used $K = 25$ clusters.

Figure 4.13 shows an example of clustering from mean shift and GMM. The GMM clustering is arguably more consistent with human expectations – photos in West Africa, sub-Saharan Africa, South America, and Central Asia form clusters even though they are spread out. For this query, mean shift does not even form a cluster within 3000km of the ground truth photo location.[2] Any geolocation heuristic using the clusters (e.g. picking the biggest cluster, or using machine learning to decide among clusters (Chapter 6)) is hopeless in the mean shift case. Unfortunately, the GMM clustering can not fare much better here – even if the nearest cluster is picked, its mean coordinate is more than 1000km away from the photo location. With an accuracy threshold of 200km, GMM clustering performs quantitatively worse than MS clustering on all of our test sets. There is nothing stopping us from using multiple, overlapping clusterings of the scene matches. Including both sets of clusters does not improve performance in practice, though. Mean shift clustering, with its fixed bandwidth, is simply

[2]We could increase the mean shift bandwidth, but this does not help performance. In Chapter 6, we decrease the minimum cluster size. This does not affect the "largest cluster" heuristic, but it does give our learning an opportunity to choose isolated photographs.

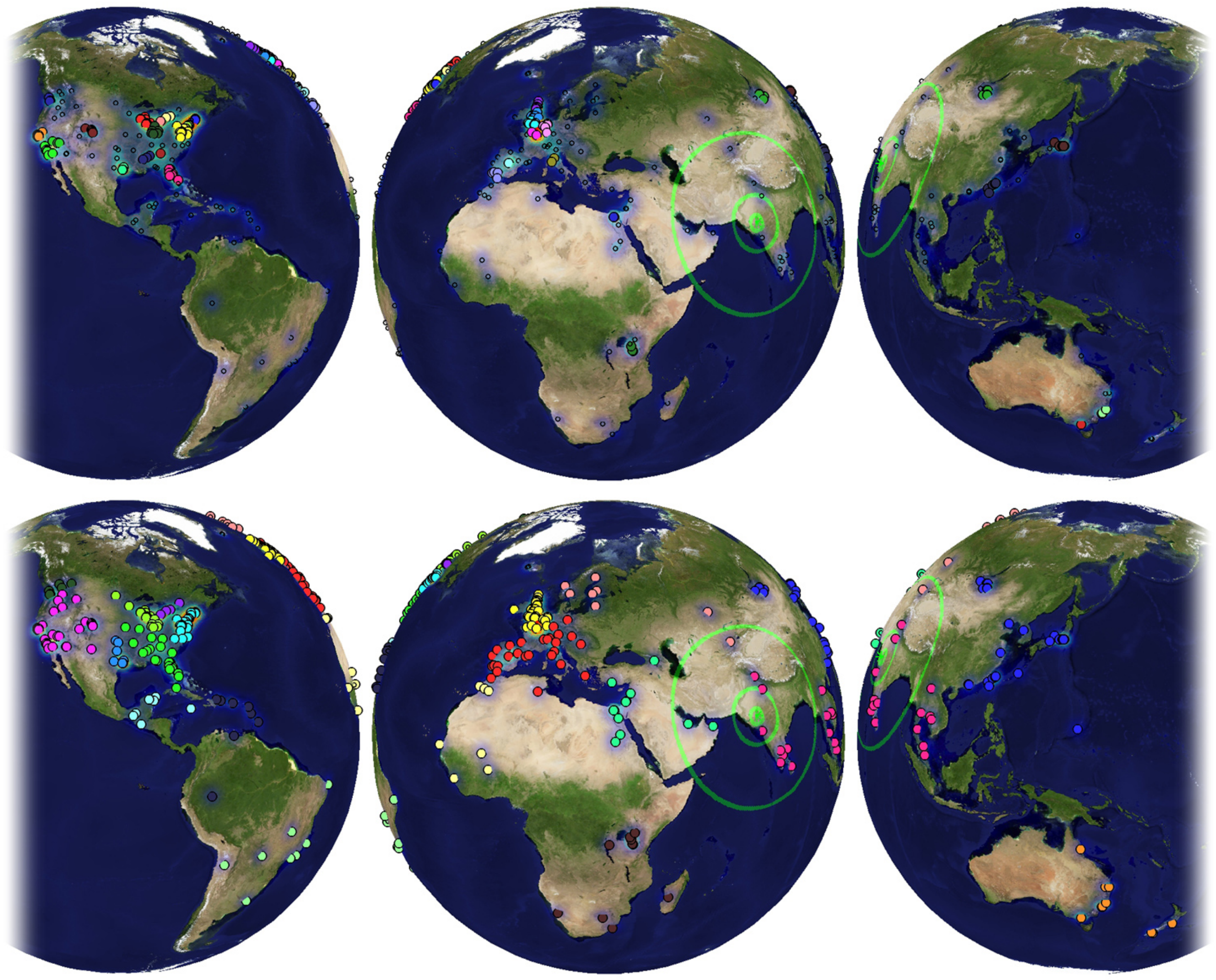


Figure 4.13: *GMM vs MS clustering.* Clustering of the same set of scene matches using Gaussian Mixture Model (top) and Mean Shift (Bottom). The mean shift clustering uses a bandwidth of 200km. Cluster membership is indicated by colored markers. The query image is from Ahmedabad. The GMM does a better job of grouping photos in sparse regions.

the most appropriate clustering method when we used a fixed distance threshold to evaluate performance. With a different metric to evaluate geolocation accuracy the GMM clustering might appear more effective.

The fundamental, unavoidable issue is that we do not have enough data for many locations around the world. A generic photo of Brazilian rain forest will find many more matches in Hawaii, Thailand, or more temperate locations than in the correct location. It is not a matter of database peakedness drowning out the correct matches – if a scene is visually distinct it will often be geolocated even if it is rarely photographed. But for generic scenes, where the visual features distinguishing one location from another are extremely subtle, a large amount of reference data is needed. So it is certainly the case that im2gps performance is inextricably tied to the geographic distribution of our test set and database. A biased sampling strategy at database creation time could help smooth out these differences, but there is not enough geotagged data on Flickr to completely remove the geographic bias of photo taking.

4.5.2 Measuring category level geolocation performance.

While we have demonstrated that our geolocation accuracy is far better than chance, random guessing is arguably not a realistic baseline comparison. Just by retrieving scenes of the same broad semantic category as a query (for instance "beach", "mountain", "city", "forest", "indoors", etc...) chance must rise considerably. Does category level guessing account for im2gps performance, or is it picking up on more subtle geographic variations?

As we increase the size of the im2gps database we see a slow but steady increase in performance (Figure 4.4). If random matching within the same scene broad scene category could account for im2gps performance, it is likely that performance would saturate with a dramatically smaller database. Previous work has shown 90% accuracy in 4-way categorization using a couple thousand training examples and nearest neighbor classification with the gist descriptor [74]. Why does our performance double as our database increases from six hundred thousand to six million geolocated examples? Part of the gain is likely because the scene matches become more discriminative (not just forest but rain forest, not just cities but European cities).

Figure 4.14 shows three queries that would fit into a broadly defined "city" category. Notice how different the geographic distribution of scene matches is for each query. The German city geolocation estimate is correctly peaked in central Europe. The Hong Kong skyline is confused with other skylines (New York, Shanghai, Tokyo, and Chicago). Hong Kong is the 5th largest cluster. The Alabama warehouse matches many paved areas or streets in the US,

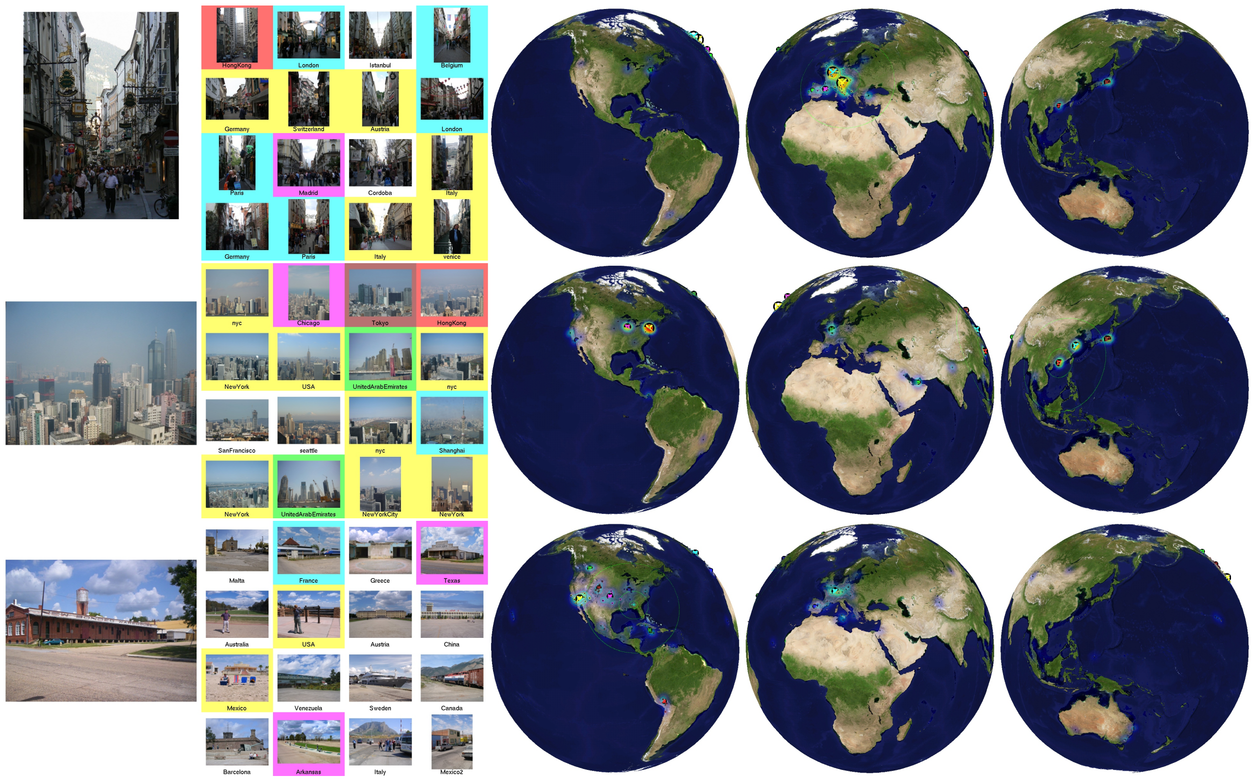


Figure 4.14: *im2gps results for different cities.* These city queries from Germany, Hong Kong, and Alabama produce very different geolocation estimates.

although none near the correct location. The im2gps scene matches can definitely be more specific than typically defined scene categories.

We can quantify how accurately im2gps would perform with perfect category level scene recognition and random guessing within that category for our test sets. We use the NASA map of land cover classification (Figure 4.22) to assign a ground truth geographic scene category to each image in a test set. The categories are “city”, “forest”, “water”, “shrubland”, “rain forest”, “barren”, “snow and ice”, “crops and grassland”, and “savanna”. We classify the entire im2gps database into the same 9 categories. Then for each photo in a test set, we calculate the probability that randomly matching to scenes of the same category will produce a geolocation guess within 200km. This is something of an ideal case because it assumes that any two categories, e.g. “shrubland” and “savannah”, can always be distinguished. Chance under this perfect categorical matching is still quite low – 2.09% for the im2gps test set (up from 0.97%) and 0.36% for the geographically uniform test set (up from 0.13%). We can safely say that our geolocation method is discriminating far more than just scene categories.

4.5.3 Measuring Landmark Geolocation Performance

Perhaps 5% to 7% of photos in the im2gps test set are readily recognizable landmarks such as Sagrada Familia or the Sydney Opera House. A very geographically knowledgeable person might even recognize the exact physical scene for 10% of the test cases. Landmarks are visually distinctive and often photographed so it makes sense that they contribute a large amount to im2gps performance. Of the 16% of queries whose first nearest neighbor is within 200km, 58% of the 1NN matches depict the same *physical scene*. Many of these would not be considered “landmarks” by a layperson – an aircraft in the Smithsonian, an Apple store in New York City, or a bridge in Portugal. At the same time certain possible landmarks, such as the Millennium Wheel in London, are missed by the first nearest neighbor.

We also evaluate the contribution of instance level matching within the larger *2k random test set*. This set contains 2000 random, unfiltered images from the im2gps database. No corrupt, abstract, portrait, or artistic photos were removed, as they were for the im2gps test set, so it is much more challenging. For this experiment we use lazy learning and more sophisticated features (Chapter 6), both of which significantly increase instance level recognition. Within the 13% of test cases that are successfully geolocated, 40% of the 1nn matches are the same physical scene. The cluster chosen by the learning contains a landmark match 58% of the time. In some of these cases it requires a great deal of study to determine if any of the matches are the same physical location and the geolocation probably would have been correct even without those matches.

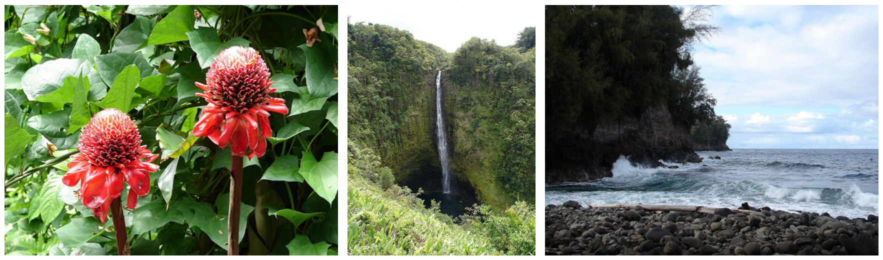


Figure 4.15: *Toy Test Set 1.* Considered individually, none of these photos are successfully geolocated. But considered as a sequence, the photos are successfully geolocated in Hawaii.

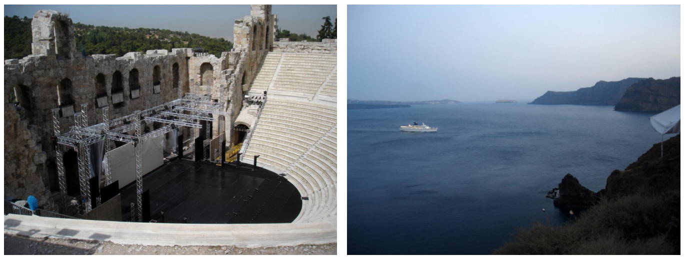


Figure 4.16: *Toy Test Set 2.* These two images separated by two hours and several hundred kilometers. The first image is easy to geolocate in Athens, but the second is more difficult. Considered as a sequence, the second image is successfully geolocated in the Greek Isles.

Thus instance level recognition does account for a slim majority of successful geolocations with both test sets and both geolocation methods. But we are also able to geolocate a significant number of photos that are not landmarks and would presumably fall through the cracks of methods such as [17, 119]. Our own experiments using SIFT features in Chapter 6 also support this.

4.6 Image Sequence Geolocation With Human Travel Priors

Above we considered the task of geolocating a *single* image. In this section, we investigate the task of geolocating a *sequence* of timestamped photographs taken by the same photographer. This section is a high level overview of [50], which was primarily the work of **Evangelos Kalogerakis**, **Olga Vesselova**, and **Aaron Hertzmann**, with **Alexei A. Efros** and myself assisting.

[2]Project Page: http://www.dgp.toronto.edu/ kalo/papers/images2gps/

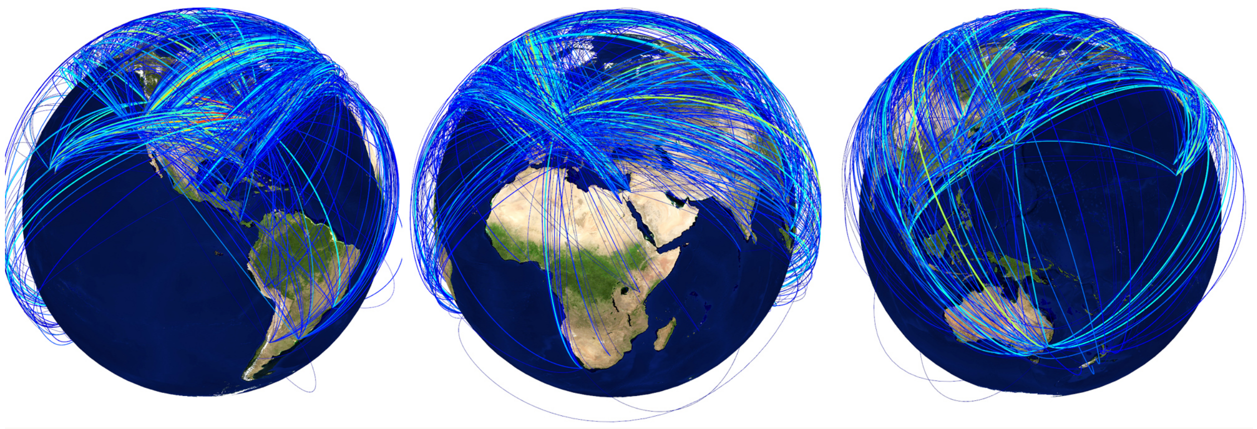


Figure 4.17: Common trips derived from geotagged Flickr data, as estimated from pairs of photographs taken by the same user 5+ days apart. Travel frequency is shown in jet color map, natural log scale.

Single image geolocation is difficult because many photographs contain little or no geographic information. Some photos contain enough geographic information to narrow down the possibilities, but not to decide on a single location. But given a *sequence* of photographs from the same user, it is often possible to geolocate otherwise ambiguous photos. In the simplest case, if the timestamps for a set of photos are very close together then the pictures must have been taken in the same small geographic area. This means that the individual geolocation estimates for each photo can reinforce each other – while many of the photos might be generic European street scenes, if just one photo depicts the Eiffel tower then all of the photos become unambiguous.

For a more subtle case, consider the photos in Figure 4.15. The geolocation estimates of each individual photo are quite ambiguous. But if we know that all the photos are taken in the same region, the most probable geolocation emerges, correctly, as Hawaii. The photos reinforce each other because their geolocation ambiguities are independent. For this toy example, we did not use any temporal information. But consider the photos in Figure 4.16, which are separated both spatially and temporally. Can we use the fact that the first photo is easily recognized as part of the Acropolis of Athens to help determine the location of the second, more generic coastal photograph? We can, but to do so we need some understanding of how people move across the planet to help constrain the geolocation. We would like a model that tells us, for every location on Earth, and every possible destination, what is the likelihood that someone would travel there after a given amount of time has elapsed.

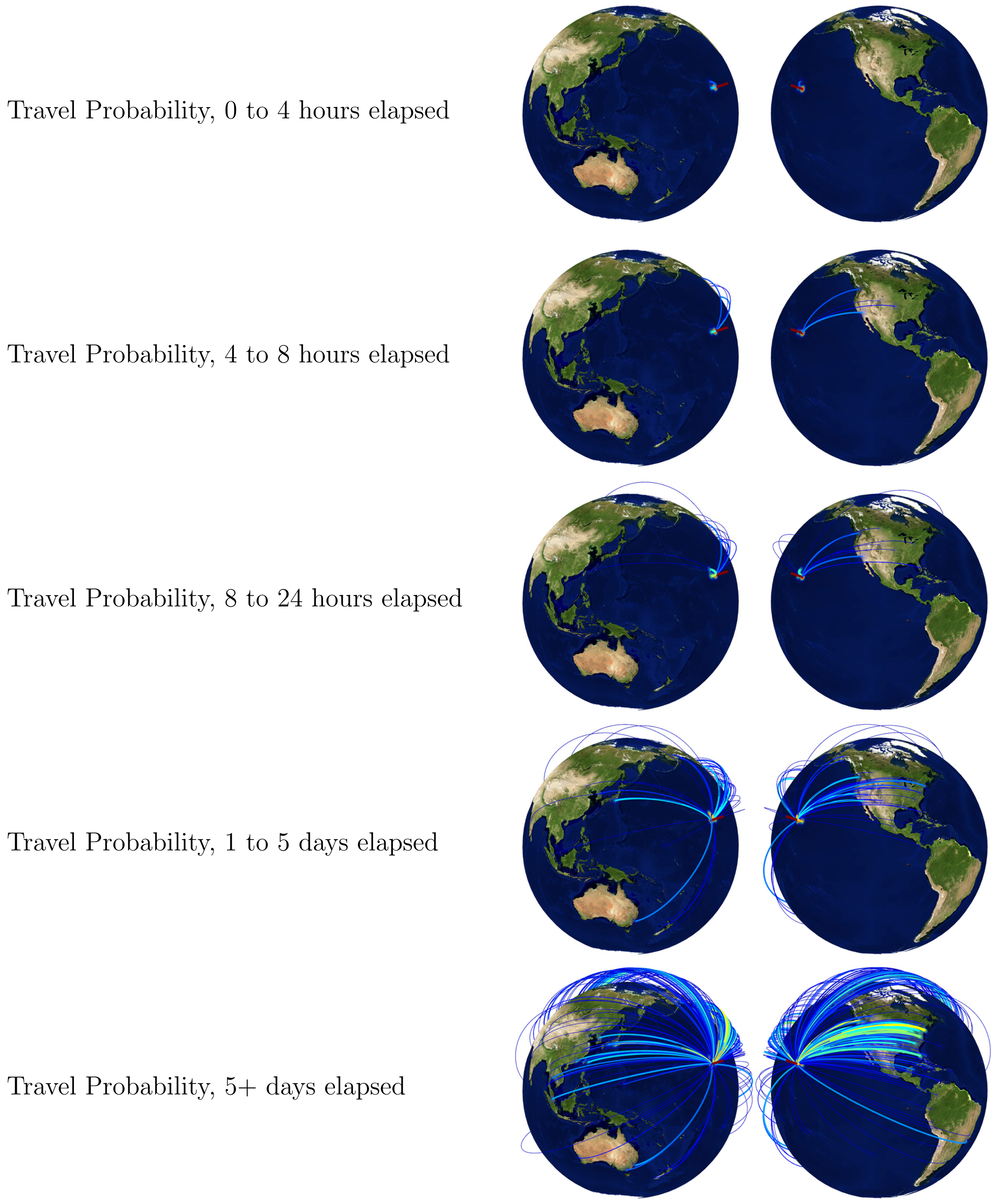

Travel Probability, 0 to 4 hours elapsed
Travel Probability, 4 to 8 hours elapsed
Travel Probability, 8 to 24 hours elapsed
Travel Probability, 1 to 5 days elapsed
Travel Probability, 5+ days elapsed

Figure 4.18: Given that a photo in a sequence was taken on the big island of Hawaii, these figures visualize the frequency of the next photo being taken at different locations around the globe for different ranges of time elapsed. With 0 to 4 hours elapsed, the next photo must be on one of the Hawaiian islands. From 4 to 8 hours elapsed it is possible to fly to the Western United States. The longer the time elapsed, the less likely it is that the photographer has remained on Hawaii, although Hawaii remains the single most probable follow-up location. Travel frequency is shown in jet color map, natural log scale.

It turns out that modelling human motion across the globe is an active and important research problem. It is useful for traffic forecasting, disaster response planning, urban planning and infectious disease simulation. Our approach is to directly learn a global photographer movement prior from the im2gps dataset of geotagged Flickr images. To build this model, and to geolocate a novel sequence, we require that the time differentials between pairs of photographs to be known. In practice, this is not an onerous requirement because cameras automatically include this information in photo EXIF data. Even if the time is inaccurate (e.g. a photographer never set her camera's clock), the time *differences* should be accurate.

We build our travel model on top of the same global location discretization described above in Section 4.5 and shown in Figure 4.11. Using the im2gps database, we count how often photographers move from every bin to each other bin to take their next photograph, for a given range of elapsed time (e.g. 4 to 8 hours). A visualization of this model is shown in Figure 4.17, and a smaller slice, showing travel probabilities from the big island of Hawaii for different time ranges, is shown in Figure 4.18.

Finally, we can geolocate a novel sequence using the single image geolocation estimates from im2gps and our travel model. We describe the unknown photo locations with a Hidden Markov Model and use the Forward-Backward algorithm (an instance of dynamic programming) to find the maximum likelihood locations. For a complete description of the model and inference process see [50].

Returning to our toy example in Figure 4.16, the maximum likelihood solution correctly places the second photograph in the Greek Isles. To more rigorously evaluate the algorithm, we build a test set of 4117 geotagged images from 20 Flickr users. We can geolocate these photos to within 400km of their correct bin 58% of the time compared to 15% for single image geolocation. Results for three sequences are visualized in Figure 4.19.

What accounts for this dramatic increase in performance with sequences? One possibility, that might obviate the need for the sophisticated learning machinery and travel models that we employ is that photos are taken in temporal clusters and many clusters contain at least one landmark. Thus, simply propagating the geolocation from landmarks to temporally nearby photos would account for the performance increase. To test this, we identify all of the test set images for which single image geolocation was correct (peaked in the ground truth bin). We then replace the geolocation estimate for these *distinctive images* with delta functions in the correct bin (reinforcing the confidence of the landmark), and replace the image likelihoods of all other photographs with uniform distributions. Thus the Hidden Markov Model sees a series of photos which are either correctly and confidently geolocated or completely ambiguous. In this scenario, the geolocation accuracy drops from 58% to 39%. Clearly landmarks alone do not account for the performance.

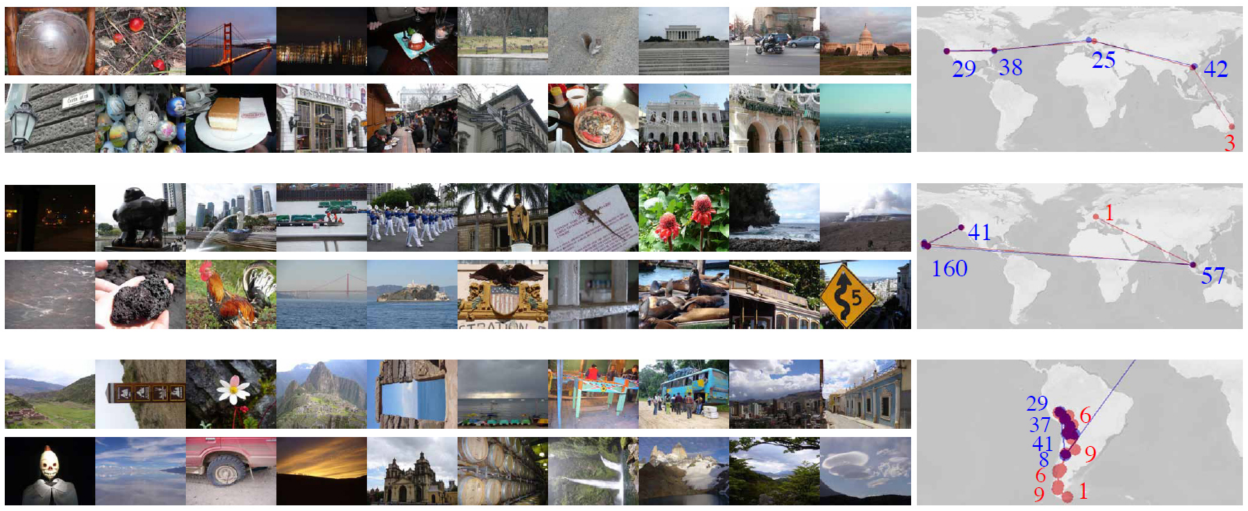


Figure 4.19: Sample images from three of our test users, and their routes. Ground-truth routes are shown in red, and estimated routes in blue. The number of images in each location is shown, with blue numbers indicating correctly-tagged regions, and red indicating errors. Top: A user with 137 photos of San Francisco, Washington DC, Budapest, Macau, and Sydney. We geolocate this sequence with 97.8% accuracy, as compared to 37.7% with single image geolocation. The only errors are in Sydney, which is shown in only three aerial views. Middle: A user with 259 photos from Switzerland, Singapore, Hawaii, and San Francisco. We geolocate this sequence with 99.6% accuracy, as compared to 18.5% with single image geolocation. The only error is in Switzerland, which is only shown in a single blurry night-time photo. Bottom: A user with 146 photos from South America. We geolocate this sequence with 79% accuracy, as compared to 10% with single image geolocation. The algorithm incorrectly labels the last leg of the trip as being in the United Kingdom. Figure reproduced with permission of Evangelos Kalogerakis and Aaron Hertzmann.

We can further test how informative the ambiguously geolocated photographs are by replacing the geolocation estimates for all *distinctive images* with uniform distributions. Thus our algorithm is forced to geolocate sequences for which *every single* individual geolocation estimate is wrong. Amazingly, the performance is still 27.5%, much better than the single image geolocation performance. The ambiguous single image geolocations can often reinforce each other. This is compelling evidence that generic scenes, and not just landmarks, can be used to geolocate photos.

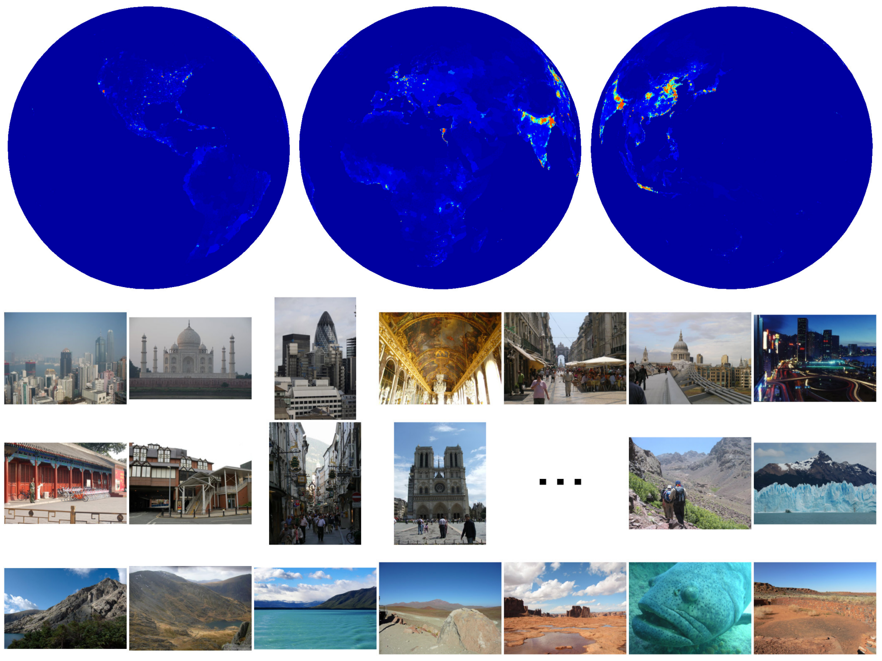
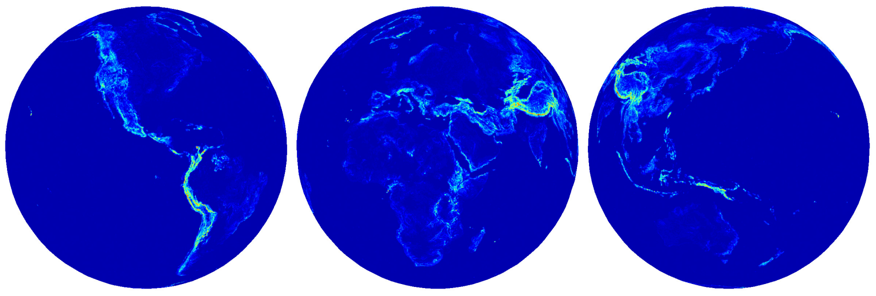


Figure 4.20: Top: global population density map, shown with "jet" color map. Bottom: in scanline order, the test cases with the highest and lowest estimated population density.

4.7 Secondary Geographic Tasks

Once we have a geolocation estimate (either in the form of a specific location estimate or a probability map), we can use it to index into any geographic information system (GIS) or map. There is a vast amount of freely available geolocated information about our planet such as climate information, crime rates, real estate prices, carbon emissions, political preference, etc. Even if an image cannot be geolocated accurately, its geographic probability map might correlate strongly with some features of the planet. For instance, given a map of population density and a query image, we can sample the pop. density map at the estimated geolocation(s) and assign the average value as an explicit pop. density estimate for the query image. Using this approach we estimate the pop. density (Figure 4.20) and elevation gradient magnitude (Figure 4.21) for each of our 237 test images.

We also produce land cover estimates for each of our test images by sampling from a land cover classification map (Figure 4.22) according to each image's geolocation probability map. We show the test images which are most likely to be "barren" (Figure 4.23), "forest" (Figure 4.24), "water" (Figure 4.25), and "savanna" (Figure 4.26).



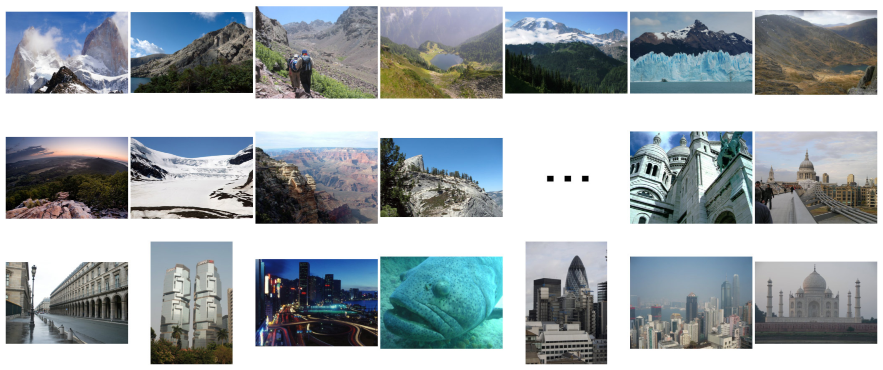


Figure 4.21: Top: global elevation gradient map, shown with “jet” color map. Bottom: in scanline order, the test cases with the largest and smallest estimated elevation gradient. Major cities tend to be in flat regions.

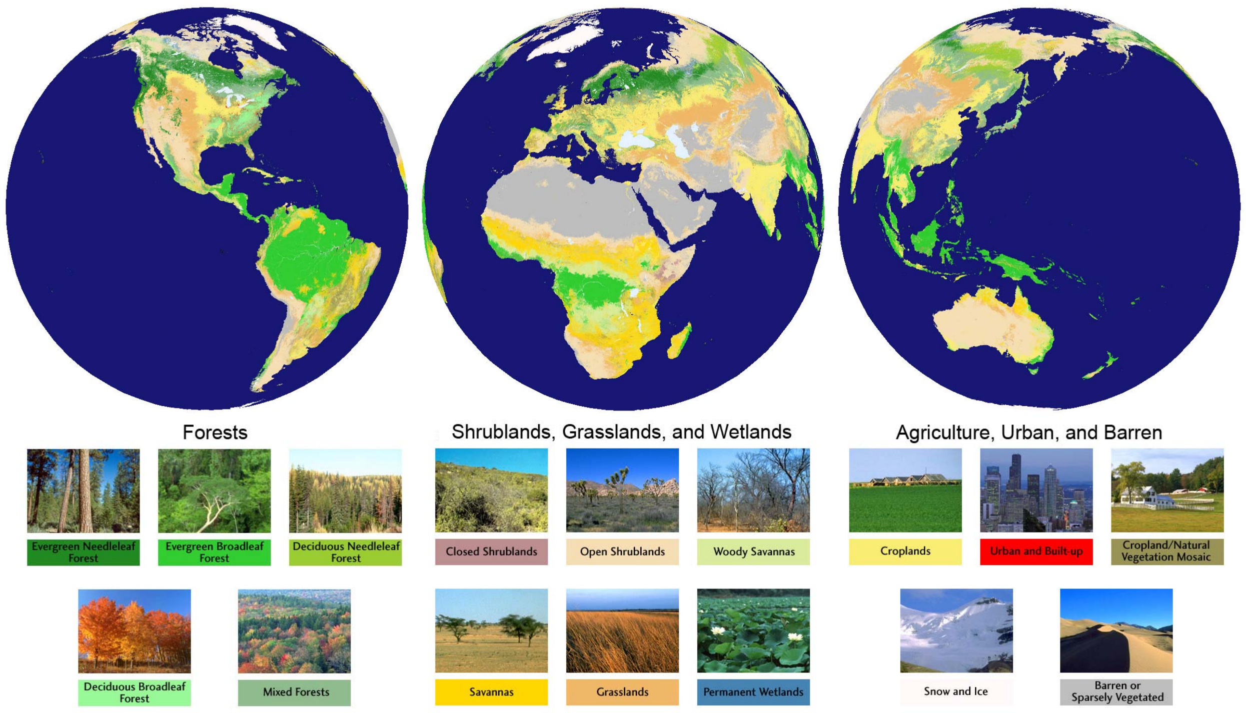

Forests
Shrublands, Grasslands, and Wetlands
Agriculture, Urban, and Barren
Evergreen Needleleaf Forest
Evergreen Broadleaf Forest
Deciduous Needleleaf Forest
Closed Shrublands
Open Shrublands
Woody Savannas
Croplands
Urban and Built-up
Cropland/Natural Vegetation Mosaic
Deciduous Broadleaf Forest
Mixed Forests
Savannas
Grasslands
Permanent Wetlands
Snow and Ice
Barren or Sparsely Vegetated

Figure 4.22: Land cover classification map and key.

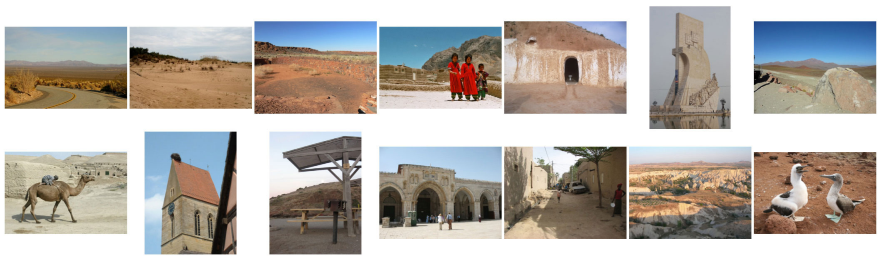


Figure 4.23: Test images with the highest “barren” likelihood.

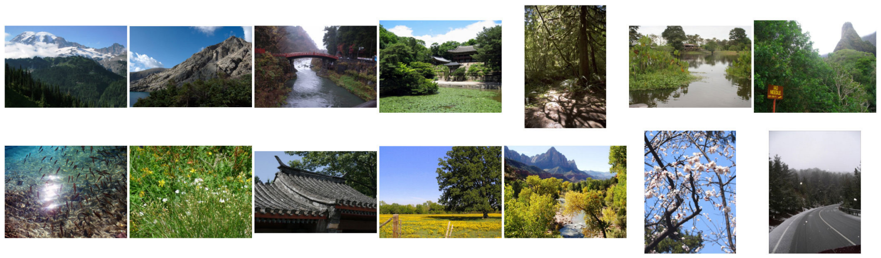


Figure 4.24: Test images with highest "forest" likelihood. Note that there is no "mountain" class in the land cover map– most mountains are labelled as "forest" or "barren" according to their land cover. The mountains above are indeed forested.

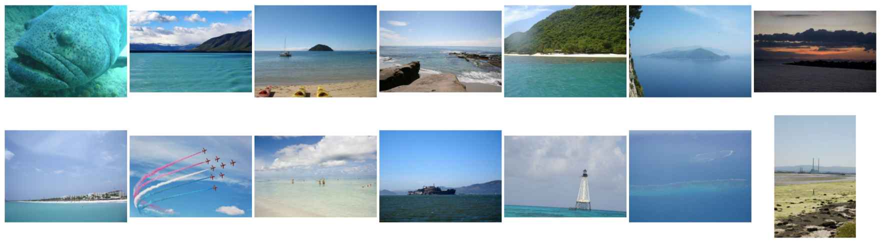


Figure 4.25: Test images with the highest "water" likelihood.

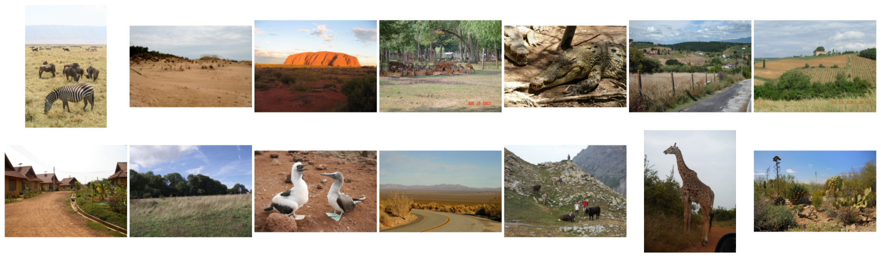


Figure 4.26: Test images with the highest "savanna" likelihood. Perhaps only a couple of the images in our test set actually depict "savanna", but many of these images contain similar visual elements.

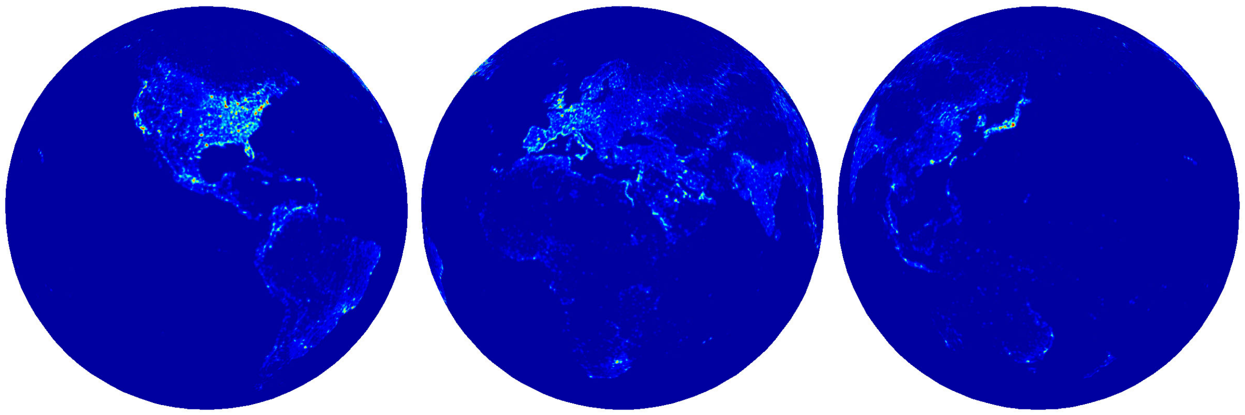


Figure 4.27: Map of global light pollution, shown with “jet” color map.

This framework can also be used to retrieve geographically relevant images out of an unlabelled collection, i.e. “Which images in my photo collection are from my trip to India?”. In this case the secondary geographic data source could be a global map where India=1.

Additionally, we can perform image classification using properties derived from a secondary geographical data source according to our geolocation estimates. For example, using a global map of light pollution (Figure 4.27), we look up the light pollution magnitude at the ground truth location of each of our test cases. We divide the test images along their median light pollution value into “urban” and “rural” classes (Figure 4.28). This is a difficult classification problem because the classes are not cleanly separated, but it is a principled way to generate ground truth scene categorization. Having defined a ground truth classification, we try to predict each test image’s class without using its ground truth location but instead using its estimated geolocation (Figure 4.29). Figure 4.30 shows the ROC curve for this task with several different heuristics to estimate geolocation from scene matches. Using the entire geolocation probability map instead of a single, explicit geolocation estimate performs best (.82 area under ROC).

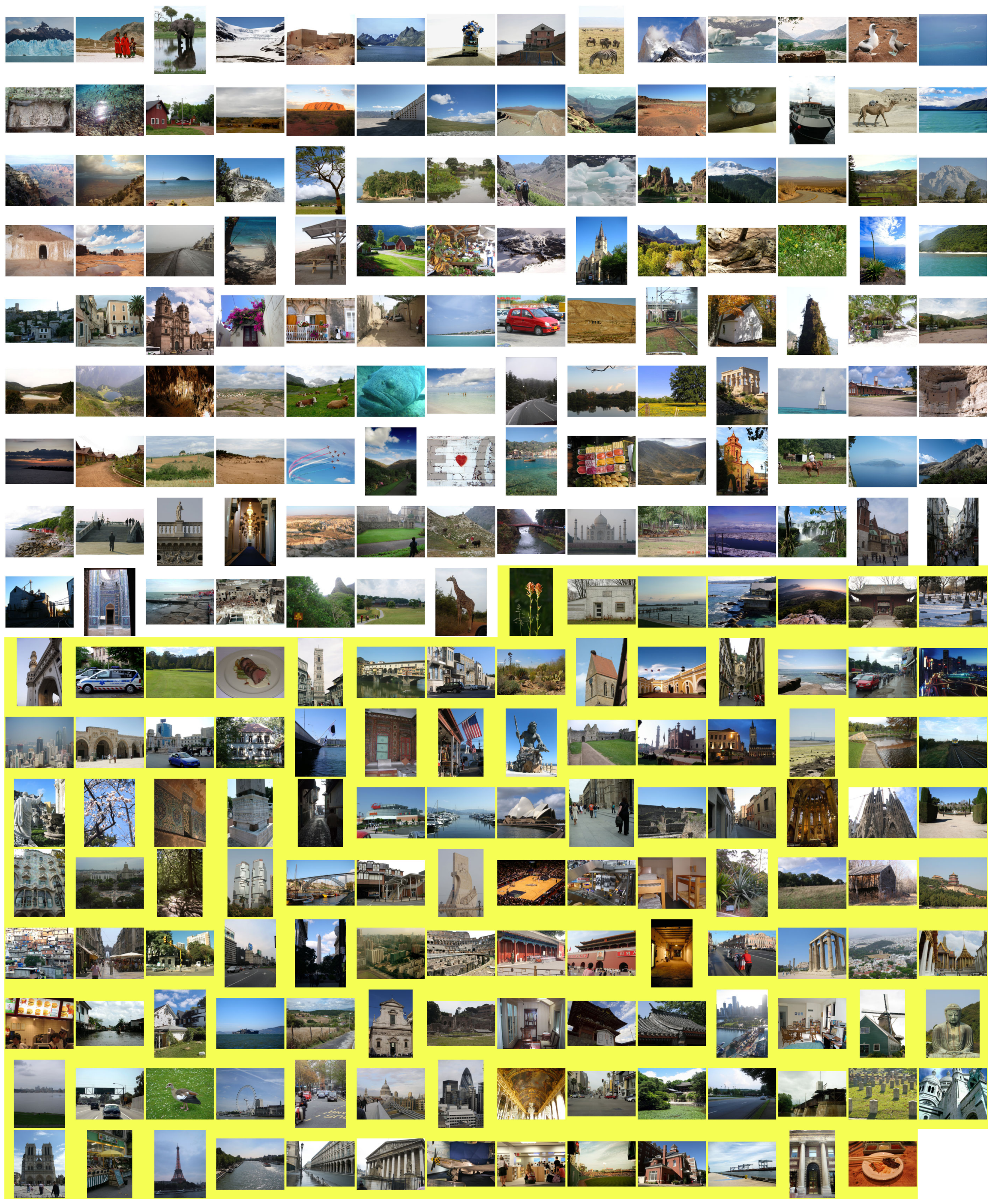


Figure 4.28: The *ground truth* ranking of test cases by the amount of light pollution at their locations. The images are classified into rural (white) and urban (yellow) by splitting along the median score. Note how un-intuitive some of classifications are- fields, flowers, and forests marked as urban because they are actually in or very near a large city. In addition, some scenes that look like cities fall below the threshold because they are actually rather small villages.

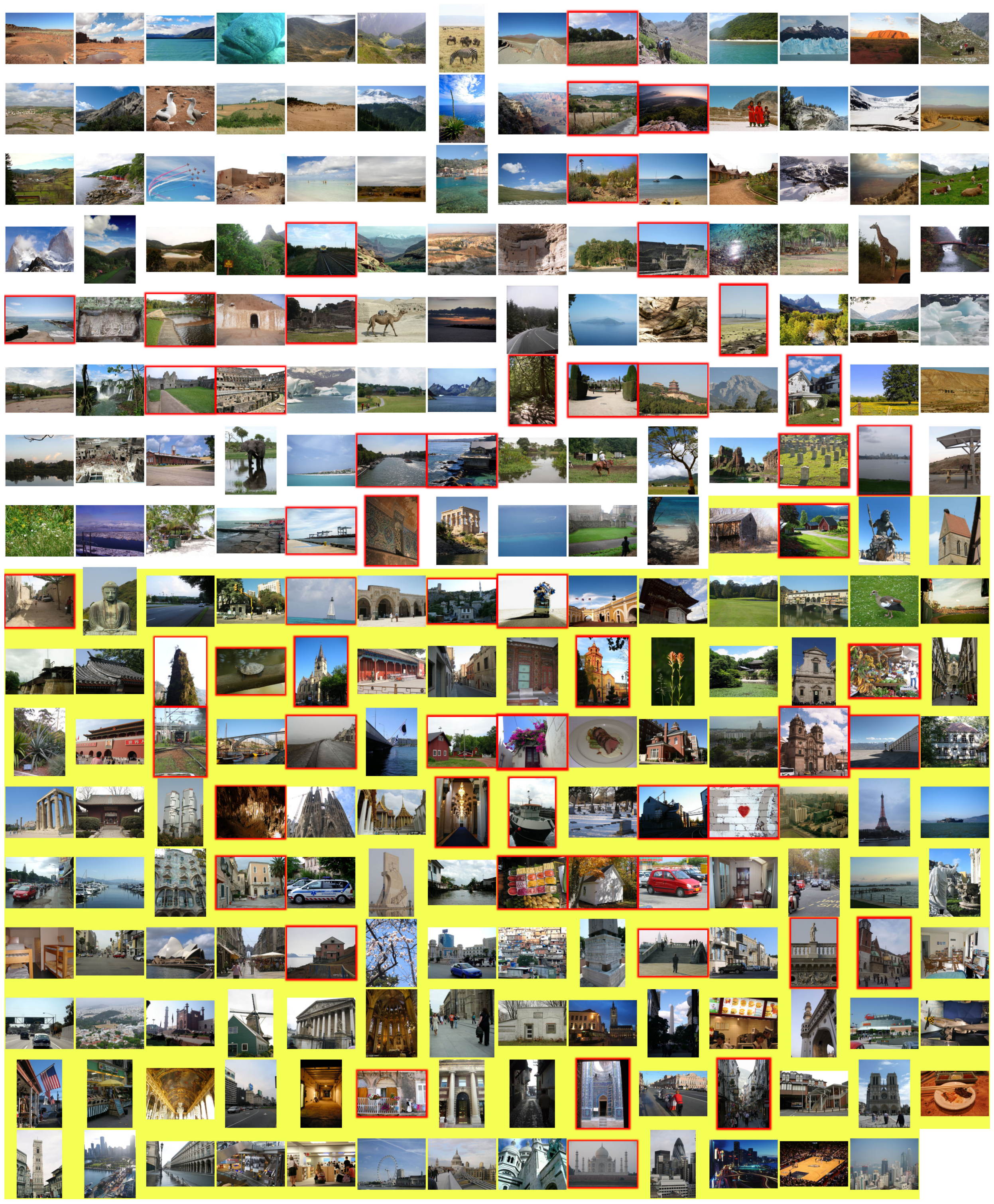


Figure 4.29: Our algorithm's ranking of test cases by the amount of light pollution at their estimated geolocation. The misclassifications are highlighted in red. Note the diversity of the concepts that are being classified as urban- basketball arenas, indoor shopping centers, indoor apartments, fruit markets, and dinner plates. Most of the misclassifications are reasonable and likely consistent with human performance on this task.

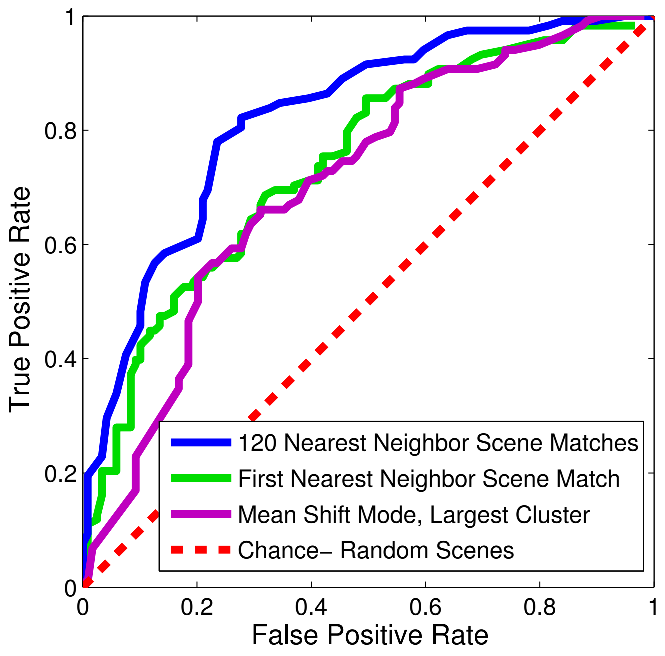


Figure 4.30: *ROC curve for urban/rural classification.* Areas under ROC curve are .82, .74, and .71 for 120-NN, 1-NN, and 1 mean-shift mode, respectively.

4.8 Discussion

We believe that estimating geographic information from images is an excellent, difficult, but very much doable high-level computer vision problem whose time has come. The emergence of so much geographically-calibrated image data is an excellent reason for computer vision to start looking globally – on the scale of the entire planet! Not only is geo-location an important problem in itself, but it could also be tremendously useful to many other vision tasks:

i) Knowing the distribution of likely geolocations for an image provides huge amounts of additional meta-data for climate, average temperature for any day, vegetation index, elevation, population density, per capita income, average rainfall, etc.

ii) Even a coarse geo-location can provide a useful object prior for recognition (See Chapter 5). In the ideal case, knowing that a picture is somewhere in Japan would allow one to prime object detection for the appropriate type of taxi cabs, lane markings, average pedestrian height, etc.

iii) Geolocation provides a concrete task that can be used to quantitatively evaluate scene matching algorithms as well as provide a more scientific basis for scene recognition studies, both for humans and machines. We return to geolocation for this purpose in Chapter 6 to show the effectiveness of new features and "lazy learning" for scene matching.

In conclusion, this work is the first to be able to extract geographic information from a single image. It is also the first time that a database of over 6 million geolocated images has been used in computer vision. While our results look quite promising, much work remains to be done. We hope that this work might jump-start a new direction of research in *geographical computer vision*.

Acknowledgements: We thank Steve Schlosser, Julio Lopez, and Intel Research Pittsburgh for helping us overcome the logistical and computational challenges of this project. All visualizations and geographic data sources are derived from NASA data.

Chapter 5

Geographic and Keyword Context for Object Detection

Chapter Summary We improve object detection accuracy using two *novel* sources of context – scene geography, as estimated by im2gps (Chapter 4) and keyword context, derived from the keywords of matching Internet scenes. These sources of context are among several explored in [24] in collaboration with **Santosh Divvala**, **Derek Hoiem**, **Alexei A. Efros**, and **Martial Hebert**.

5.1 Introduction

Object detection – the task of identifying and localizing categories of objects in an image – is one of the most actively studied problems in computer vision. The state-of-the-art methods are "sliding window" approaches that exhaustively scan regions in an image for areas that locally resemble an object category. For example, Felzenszwalb et al. [34] searches multiple offsets and scales for rectangles that resemble discriminatively trained, deformable templates of object categories in the "Histogram of Gradients" feature space popularized by [19].

But local appearance can be ambiguous or misleading. For example, in Figure 5.1, the Felzenszwalb et al. detector has mistaken a carriage and horse legs for a dining table. To a human, this seems like a particularly egregious error because the scene context (outdoor, street), and object context (nearby cars and trees) make a dining table extraordinarily unlikely. Humans use context to aid object detection (See [75] for an overview), and there have

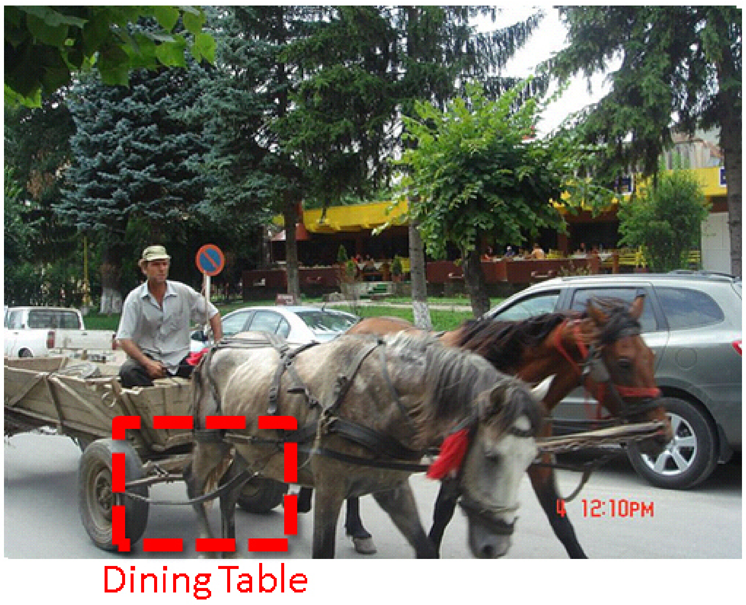


Figure 5.1: The detector of Felzenszwalb et al. [34] confidently detects a dining table in an unlikely context.

been numerous recent computer vision methods for employing context in object detection (see [24] for an overview), but these methods have not been proven effective on a dataset as diverse as PASCAL.[1]

Here we introduce two novel sources of scene-level context for object detection and evaluate them as part of an entry to the PASCAL VOC 2008 challenge. Specifically, we examine context from *geography* and *keywords* as estimated from scene matching. The goal of this scene-level context is to predict the likelihood of observing an object o given the image I i.e., $P(o|I)$. In Section 5.6 we will combine these estimates with the local object detector of Felzenszwalb et al. [34].

5.2 Geographic Context

In im2gps [39] (Chapter 4) we showed that scene matching can be used to estimate geographic properties of an image such as population density, elevation, land cover class, and urban vs

[1]The PASCAL Visual Object Challenge [31] is an annual competition for object classification and detection. Its data sets are frequently used as benchmarks for detection systems. PASCAL is very challenging, requiring the detection of 20 object categories in diverse Flickr photographs.

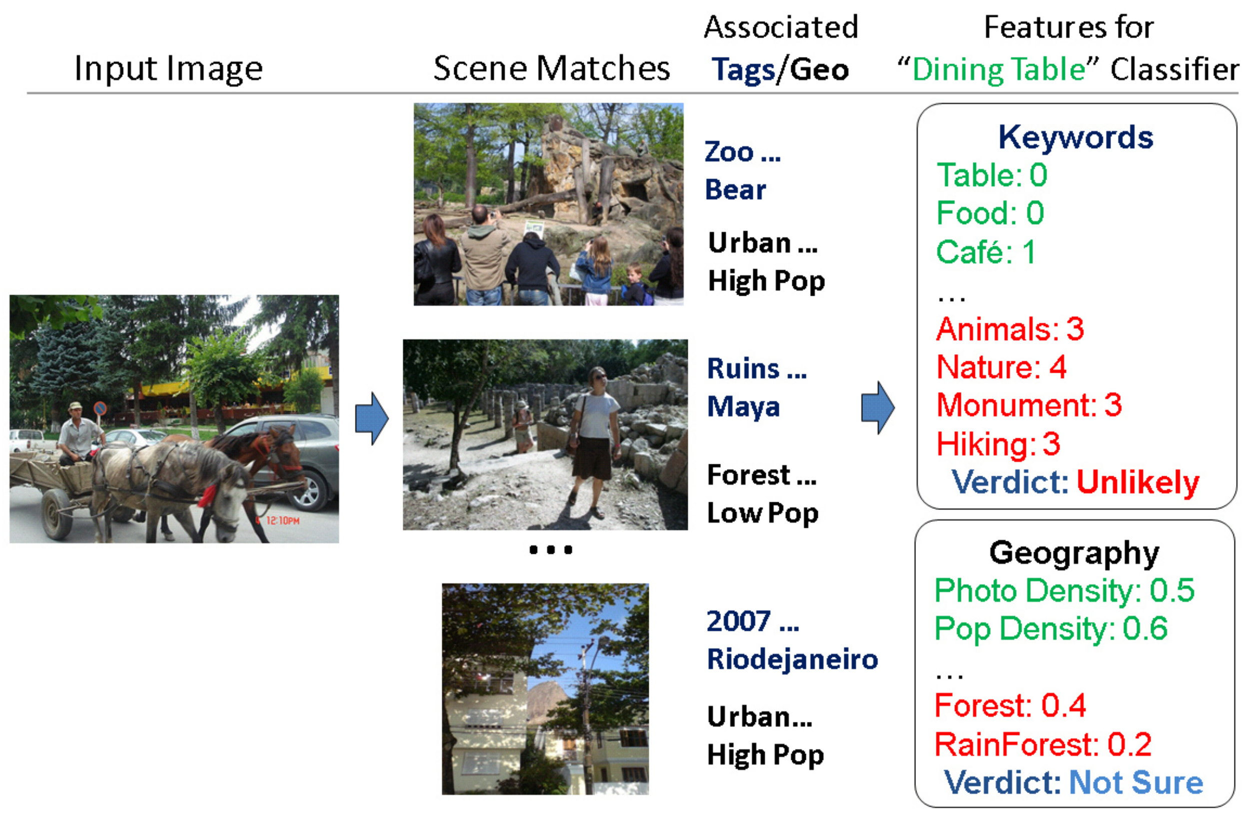


Figure 5.2: *Geographic and Keyword context.* Geographic properties and keywords associated with the scene can help predict object presence in an image. The base detector finds a dining table in this input image (see figure 5.1), while the context indicates that a dining table is unlikely.

rural class. Object occurrence is correlated with scene geography – dining tables are rare in forests, boats are frequently found in water scenes, people are more likely in high population density scenes. We would like to test how strong this relationship is and whether knowing scene geography could improve object detection accuracy.

To build a "geographic context", we estimate geographic properties for a novel image by finding matching scenes within our im2gps database of approximately 6 million geotagged Flickr photographs with known geographic properties. We compute 15 geographic properties such as land cover probability (e.g., 'forest', 'cropland', 'barren', or 'savanna'), vegetation density, light pollution, and elevation gradient magnitude by averaging the values of the matching scenes. Separately for each object class, we train a logistic regression classifier [53] to predict object presence from these geographic properties.

Figure 5.1 visualizes our context pipeline at test time. The geography context is undecided in this case: the high likelihood of forest suggests a dining table is unlikely, but the high likelihood of a city suggests a dining table is possible. The relationship between our estimated geography and object presence is often weak. For instance, ten of the VOC object classes are predominantly indoor and thus their contexts cannot be well distinguished by geography. Furthermore, the geographic context can not reliably discriminate indoor vs outdoor photographs.

5.3 Keyword (Semantic) Context

A more direct way to estimate object presence from scene matches might be to examine the keywords attached to those scene matches. For example, if an image's scene matches are all labeled "cat" that is a strong indicator that the image contains a cat. Unfortunately, inaccuracies in the scene matching process and the sparsity of text labels rarely make it this straightforward. Useful keywords are very sparse – on average there is just one content descriptive keyword per image in the im2gps dataset. Contrary to other studies [21], we find the existing keywords to be fairly accurate (e.g. an image tagged "sheep" will usually contain a sheep).

To build a "keyword context", we first manually filter the 2000 most popular words appearing in Flickr tags and titles into 50 categories. 20 categories are for words that correspond unambiguously to each of the VOC object classes (e.g. 'bottle', 'beer', 'wine', and 'vino' for the bottle category), and 30 categories for additional visual categories (e.g. a category for 'church', 'cathedral', and 'temple'). Roughly 75% of the popular keywords are discarded because they are weakly correlated with visual content (e.g. "the", "november", and "nikon"). The purpose of grouping the keywords into categories is to try and provide more robust features to the learning. If the learning were presented with a 2000 word histogram it would be more likely to overfit. A more sophisticated alternative would be to automatically learn useful categories or to use a lexical database such as WordNet [33] to know that 'A320' is a plane, '350z' is a car, and 'csx' is a train. However, WordNet does not contain any of these popular, proper names and for others it is misleading ('Toyota' is a city) so we did not use it.

For a novel image we build a histogram of the keyword categories that appear among the 80 nearest neighbor scenes. We use logistic regression to predict object class based on this histogram. In Figure 5.1, the keywords associated with the scene matches clearly indicate an outdoor setting. The logistic regression has learned that keywords relating to categories "ruins", "hiking", and "monuments" mean the image is unlikely to contain a dining table.

Both geographic and keyword context classifiers are trained on the VOC'08 trainset. Due to the large size of the VOC data set ($\sim$ 10 thousand images), we approximated the distance metrics of im2gps with the faster to compute L2 distance, which decreases the relevance of the retrieved scenes somewhat. Both PASCAL VOC 2007 and 2008 partially overlap with the im2gps database. During the scene matching process, we are careful to remove any exact matches from consideration during the scene as well as all other photographs by the same user.

Contemporary with our research, Wang, Hoiem, and Forsyth [110] examined image classification using keywords from matched Internet images. Although they did not use their classifier for object detection context, it would likely be effective in that role.

5.4 Direct, VOC-learned context

The high-level strategy for geography and keyword context is: find similar Internet images, and determine what their geographic and keyword labels indicate about object presence. The relationship between those scene match labels and object presence is learned within the VOC trainset. The advantage of using a large collection of Internet images is that it allows us to reliably find similar scenes. But a disadvantage is that the labels on those scenes are weak (in the case of geography) or sparse (in the case of keywords).

A logical alternative would be to instead find similar scenes within the VOC trainset. In this case, the labelling is perfectly accurate and gives us information about not just object presence, but object scale and location, as well. The difficulty of this approach is that the VOC trainset is 2000 times smaller than our im2gps data set. With a data set as diverse as PASCAL VOC, this amounts to a very sparse sampling of the space of observed scenes. This alternative is explored in our work [24]. Within the VOC trainset, a direct relationship is learned between the scene gist descriptor and geometric context of each image and the presence of objects in that image. The scene gist of an image is computed in the standard way as described in [74]. The geometric context for an image is computed as a set of seven geometric class (ground, left, right, center, sky, solid, porous) confidence maps as described in [40]. These confidence maps are re-sized to 12×12 grids and vectorized to serve as a coarse "geometric gist" descriptor. For each of the descriptors, logistic regression is used to train object presence classifiers for every object class. The idea of relating global scene statistics to object presence was also explored in [105].

Context source	Average Precision
Geographic	15.1%
Keyword	25.6%
Scene gist	23.9%
Geometry	21.5%
Combined	31.2%

Table 5.1: *Accuracy of context types.* The average precision of object presence classifiers trained on multiple sources of context and tested on the PASCAL VOC 2008 validation set.

5.5 Object Presence Accuracy

In Table 5.1 we report the average precision of each source of context for the task of predicting whether an object is present in an image or not. This is exactly equivalent to the "object classification" task of the PASCAL VOC. All of the methods are significantly better than chance ($\sim$ 7%)[2], but none of the methods are competitive with the state of the art in object classification (which is roughly 55% average precision). However, these sources of context are meant to be *complementary* with a local object detector, and thus focus only on scene statistics. VOC object classes that occur in very similar scene contexts (horse, sheep, and cow; car and bus; dog and cat; bicycle and motorcycle; chair and sofa) are thus indistinguishable to our context methods unless object appearance significantly influences the scene descriptors (which is possible for large objects).

The Keyword context, derived from Internet scene matching, is the most effective. Thus our use of large scale Internet data, although weakly labelled, is justified. The sources of context are somewhat complementary – combining them with a provides a modest increase in average precision.

[2]Chance is calculated as following: For every VOC test image, the confidence of that image containing each class is assigned randomly. Calculation of average precision then proceeds normally. Because the confidences are random, the precision will be uniform through the range of recall values. The precision for each class will be the chance that a random test image actually contains that class. This chance varies for each VOC class – maximum at 46% for person and minimum at 1.5% for sheep. Averaging across classes, and assuming that the test set distribution matches the trainval distribution, one arrives at chance of 7.08%.

5.6 Combining the scores for PASCAL 2008 Entry

Our goal was to use our geographic and keyword features as context for *object detection* (localization) and not in isolation for *object classification* (Table 5.1). We start with the object detector of Felzenszwalb et al. [34] which had the top performance in the most categories in the PASCAL 2007 VOC object detection challenge [30]. Our approach is to "rescore" each candidate detection based on the object presence likelihood as determined by context. In addition to the object presence context described above, we also employed contextual cues for scale and position (see [24] for details). To train a final classifier which incorporates these contextual cues we use the VOC validation set. For the top candidate detections in each image we compute all of the context scores (which will be confidences from their respective logistic regression classifiers). If the detection has at least 50% overlap with a ground truth bounding box, we consider it correct. We train a final, combined L1 regularized logistic regression classifier using the confidence of the object detection and the confidence of the context cues. Lastly, the bounding box for each detected object is refined using a local graph cut segmentation.

5.7 Results

The addition of context improves the average precision of the Felzenszwalb et al. [34] detector from 18.2% to 19.4%. Improving the bounding boxes of the detections further increases the average precision to 22.0%. This improvement was enough to put us among the top performers in the 2008 challenge, although our entry was technically the sole entrant in its competition because we use data beyond the PASCAL VOC training sets to build our system. Another top performing system was a more recent version of [34]. We added the same context to this version and attained slightly smaller improvements (from 22.4% to 23.4% with context and 23.9% with improved bounding boxes).

In Figures 5.3 and 5.4 we show the images for which the context has most strongly increased and decreased the confidence of local object detections. In essence this shows the cases where the context and local appearance most strongly *disagreed*, or, equivalently, where the context has the strongest effect.

5.8 Discussion

The scene-level, object presence context described in this chapter accounted for a 1% gain in average precision for each of the local object detectors we used. This may seem trivial, but the PASCAL VOC is difficult and improvements are hard to come by. A deeper analysis of the results reveals that our scene context significantly changes the *types of errors* the detector makes – there are fewer false positives on background, but more false positives between some objects that share contexts (e.g. birds and planes; horses and sheep). Small and occluded objects, for which local appearance alone is more ambiguous, are more accurately detected. See [24] for more detailed analysis.

While we have shown that Internet data, though weakly and sparsely labelled, is useful for gaining image understanding and aiding object detection, the effect is somewhat limited. There are numerous avenues for improvement – the quality of the scene matches can be improved with better features and learning (Chapter 6). If large scale image collections could be better annotated (and such efforts are underway in "Games With a Purpose" [3] and ImageNet [21]), then the scene matching could provide much more reliable information.

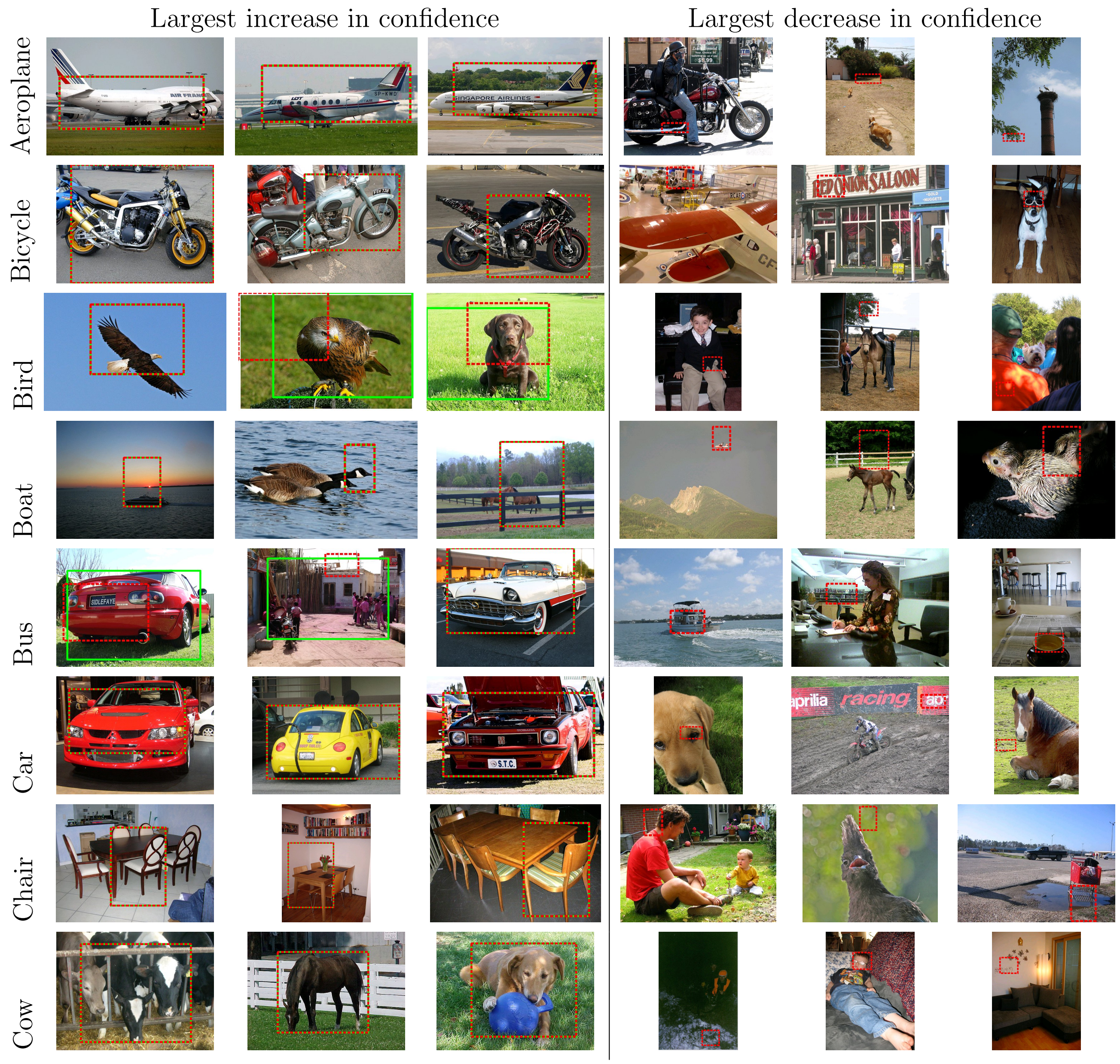

Largest increase in confidence
Largest decrease in confidence
Aeroplane
Bicycle
Bird
Boat
Bus
Car
Chair
Cow

Figure 5.3: Images for which the best detections had the largest increase and decrease in confidence after the addition of context. In these cases the local appearance and global context disagree most strongly. When the addition of context increases confidence (left) it is because a detection is in a reasonable setting for the object class, even if the local appearance does not match well (motorbikes on second row share context with bicycles). When the addition of context decreases confidence (right) it is typically pruning away spurious detections that had high confidence scores from the local detector. (Red Dotted: Detector, Green Solid: Detector+context).

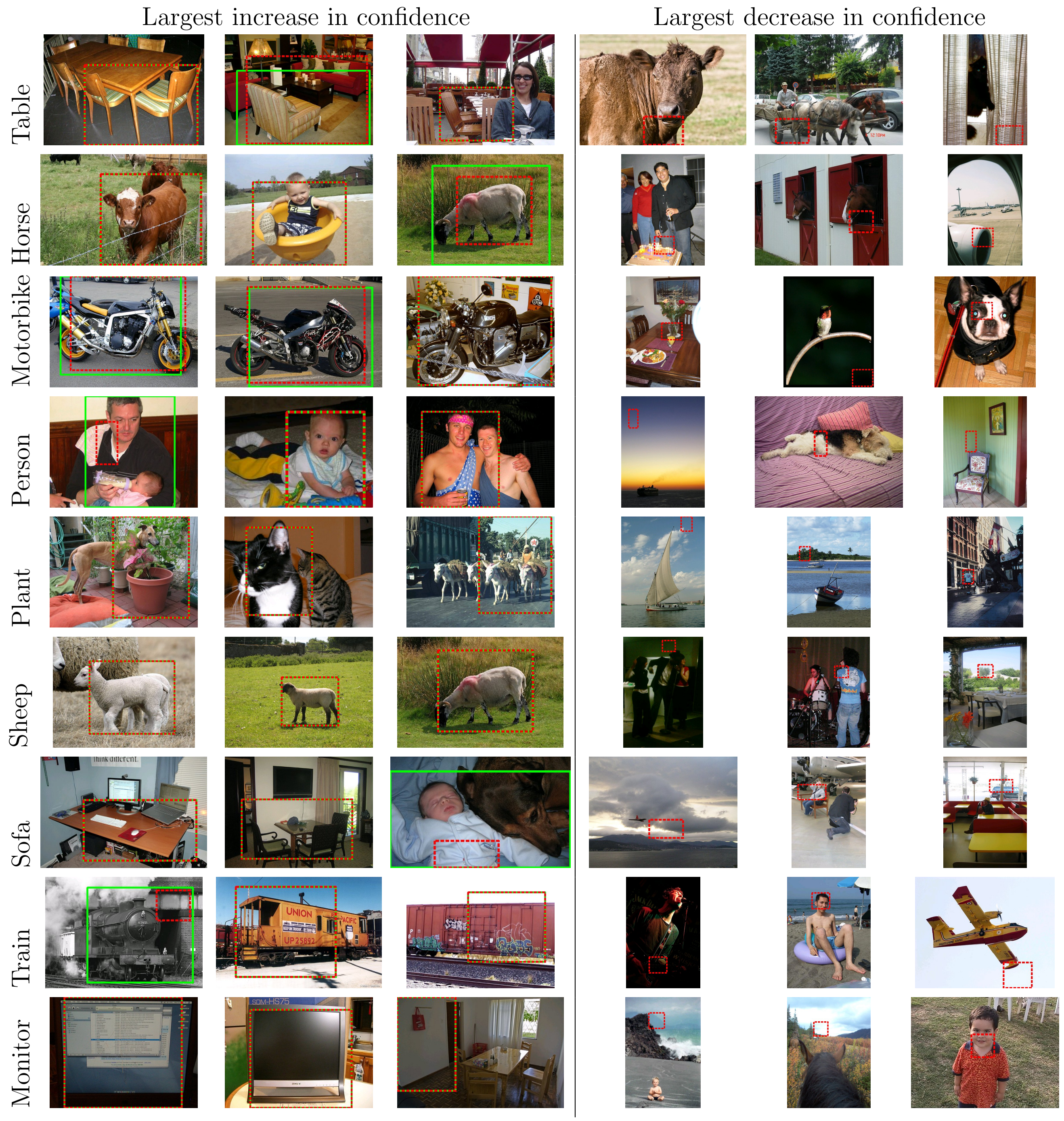

Largest increase in confidence
Largest decrease in confidence
Table
Horse
Motorbike
Person
Plant
Sheep
Sofa
Train
Monitor

Figure 5.4: Images for which the best detections had the largest increase and decrease in confidence after the addition of context. In these cases the local appearance and global context disagree most strongly. (Red Dotted: Detector, Green Solid: Detector+context).

Chapter 6

Lazy Learning for Large Scale Scene Matching

Chapter Summary We use "lazy learning" and additional features to improve large scale scene matching. First, we revisit the image geolocation task and train a multiclass, kernel SVM to choose which geographic cluster of scene matches a query belongs to. Second, we describe the additional features, some novel, that we utilized in the learning to address shortcomings of the existing im2gps features (Chapter 4). In concert, the lazy learning and new features double the baseline im2gps performance.

6.1 Learning for Scene Matching

Recently there has been significant interest in large scale image matching for computer vision and computer graphics applications. These retrieval approaches are typically non-parametric – finding nearest neighbors according to one or two image features and then using those nearest neighbors for various tasks (object classification [100], instance level recognition [45, 71, 78], scene parsing [63, 103], synthesizing virtual environments [89]), as well as all of the applications in this thesis such as Image Completion (Chapter 3), Image Geolocation (Chapter 4), and Object Detection (Chapter 5).

Most of these methods rely on a narrow range of image features – either SIFT [65] or scene gist [72]. However, in this chapter we propose numerous other image features that could be useful for scene matching – local and global measures of color, texture, and geometry

(Section 6.2). This presents a problem because nearest neighbor methods suffer as feature dimensionality increases. Nearest neighbor methods have no inherent ability to learn which features are relevant and irrelevant. Furthermore, the relative importance of features is not just task specific but *query* specific. There may not exist a satisfactory, fixed feature weighting or distance function for a given task. Ideally, an algorithm would focus on the set of features that is discriminative for each specific query.

In this chapter we adopt a "lazy learning" approach, inspired by SVM-KNN [117], to improve large scale nearest neighbor methods. We demonstrate the effectiveness of this approach on the task of global image geolocation as introduced in im2gps (Chapter 4). Our supervised lazy learning can be seen as a post-process to refine the nearest-neighbor search we employed in im2gps. A point of interest about our approach is that our *classes* for the supervised learning emerge in a lazy manner (after a nearest neighbor search) rather than being predefined as in SVM-KNN [117].[1] We demonstrate that our learning, in concert with our new features, dramatically improves geolocation accuracy. We believe this lazy learning framework is applicable to other scene matching problems.

6.1.1 Related Work

Before we explain our lazy learning approach we will briefly review related learning methods:

Support Vector Machines

Typical kernel SVMs have a training complexity of $\mathcal{O}(N^2 * D)$ where N is the number of training examples and D is the feature dimensionality.[2] With millions of data points, this approach is intractable. There are numerous large scale SVM approaches in the machine learning literature, but any method that requires an all-pairs distance computation as a starting point, such as SMO [79], is intractable with millions of high dimensional training examples. Many large scale methods approximate the SVM solution without necessarily seeing all of the training data. However, for the tasks that we are interested in, such as

[1]Because the output of a geolocation estimation system is a real-valued GPS coordinate, it might seem like it is more naturally a regression problem. But for any query scene, the geolocation problem ends up being a decision between several discrete, disconnected possibilities (e.g. Alps vs Cascades vs Rockies vs Andes). Therefore we think it is natural to treat it as a classification problem.

[2]Although it is often stated that SVMs trained in the dual have a complexity of $\mathcal{O}(N^2 * D)$ and SVMs trained in the primal have a complexity of $\mathcal{O}(N * D^2)$, [12] argues that it is possible to reformulate either approach to have complexity $\mathcal{O}(max(N, D) * min(N, D)^2)$.

geolocation, we know that performance depends on utilizing as much training data as possible (Chapter 4, Figure 4.4). Other approximation methods limit the number of support vectors (e.g. NORMA [52]). This strategy was found to hurt image classification performance in [111].

Learning for Image Retrieval

There has been significant recent work in applying machine learning methods (often Kernel Support Vector Machines) to image retrieval and scene classification problems. While there is not room for an exhaustive review of the literature, the following methods provide a sampling of different approaches. While some of these methods learn distance functions that are applied to data sets of one million+ images, none are trained on more than tens of thousands of examples.

Frome et al. [36] learn individual distance functions for *every exemplar* in a data set, referred to as local distance function learning. Malisiewicz and Efros [66] continue this line of work and propose and SVM-like optimization to learn local distance functions. These methods do not naively scale to millions of data points because each training example requires a large scale learning procedure involving all other points. Bosch et al. [8] explore KNN and SVM-based scene classification with pLSA and bag-of-words models of various SIFT-like features. The largest data sets explored contain thousands of images. In “small codes”, Torralba et al. [103] learn a mapping from gist scene descriptors to small binary codes such that scenes with small ground truth distances will have small Hamming distances between their binary codes. Ground truth scene distances are defined by object overlap as indicated by annotations for 22,000 LabelMe images. Two scenes with similar objects in similar positions will have a high overlap score. The learned scene codes produce quantitatively better scene matching within the LabelMe dataset than the original gist descriptor. Jain et al. [44] learn distance metrics, coupled with hash functions, that allow rapid image retrieval with higher accuracy than KNN.

Perhaps the most promising recent research for large scale SVMs, Wang et al. [111] train SVMs to classify images into Flickr groups and PASCAL categories. They use the fast and compact “histogram intersection kernel” coupled with an online stochastic SVM training method. With 80 thousand example images and image features analogous to our own, they can train an SVM in 150 seconds. Performance is nearly as good as traditional, batch-trained SVMs.

Lazy Learning Methods

Nearest neighbor methods are attractive because they require no training, they are trivially parallelizeable, they perform well in practice, and their query complexity scales linearly with the size of the dataset. In fact, it is often possible to perform nearest neighbor search in less than linear time, especially if one is willing to adopt approximate methods. However, nearest neighbor methods lack one of the fundamental advantages of supervised learning methods – the ability to focus on the relevant features for a given task or query.

Our approach is a lazy learning technique inspired by SVM-KNN [117] and prior supervised, KNN enhancements [25,35,37,95,107]. [13] provides a nice overview. Lazy learning methods are hybrids of non-parametric, KNN techniques and parametric, supervised learning techniques. The philosophy driving these works is that learning becomes *easier* when examining the local space around a query instead of the entire problem domain.

Consider the image geolocation problem. The boundary between geographic classes (e.g. Tokyo and London) is extraordinarily complex because it must divide a wide spectrum of scenes types (indoor, urban, landscape, etc...) that occur in both locations. There is no simple parametric boundary between these geographic classes. However, within a space of similar scenes (e.g. subway carriage photos) it may be trivially easy to divide the classes and this allows one to employ simpler, faster, and easier to interpret learning methods. Thus lazy learning is promoted not as an approximation method, but as a learning enhancement. But it is the scalability to very large data sets (which SVM-KNN [117] makes note of) that makes lazy learning attractive to us.

The approach of SVM-KNN [117] is straightforward – for any query, find K nearest neighbors and then train a multiclass, kernel SVM using those K nearest neighbors and then use it to classify the original query point. The K nearest neighbors are found with a coarse, fast to compute distance metric while the kernel matrix for the SVM is built with a more sophisticated, slow distance metric. As K becomes small, the technique reduces to 1NN. As K becomes large, the technique reduces to a standard kernel SVM.

6.1.2 Learning Approach

Our approach follows the same basic outline as SVM-KNN [117], although our classes emerge at query time and we use different features, distance metrics, and multiclass SVM approaches. For a novel query, our algorithm is:

1. Find $K_{sl} = 2000$ rough nearest neighbors using the "base" features defined in im2gps (Chapter 4).
2. Reduce the K_{sl} nearest neighbors to K using both "base" features and the additional features introduced in this chapter.
3. Cluster the K nearest neighbors according to their geographic locations using mean shift. We use a bandwidth of 200km. Each of the $\mathcal{C}$ clusters is now considered a distinct class for the sake of learning. Typical values of $\mathcal{C}$ are 30 to 60, depending on the minimum allowed cluster size.
4. Compute the all-pairs distances between all K nearest neighbors using both the "base" and additional features with L1 and χ^2 (chi squared) distances.
5. Convert the all-pairs distances into a positive semi-definite kernel matrix (i.e. the "Kernel Trick") and use it to train $\mathcal{C}$ 1-vs-all non-linear SVMs.
6. For each of the $\mathcal{C}$ classifiers, compute how far the query point is from the decision boundary. The class for which this distance is most positive is the "winner", and the query is assigned to that class.
7. The final geolocation estimate for the query is then the average GPS coordinate of all members of the winning class.

Steps 1 and 3 are largely unmodified from im2gps (Chapter 4), except that in im2gps the mean shift clustering was explained as a form of "geographic outlier rejection" where the largest cluster was the winner. It was effectively a KNN voting scheme. This can improve performance (as seen in Section 6.1.3), but it is very sensitive to the value of K – if K is too small, it reduces to 1NN. If K is large then popular locations in the database (e.g. London) will always win. With SVM-KNN, tiny clusters can be chosen. Performance is not sensitive to the minimum cluster cardinality, although disallowing singleton clusters can provide a speed up.

Our approach depends on the nearest-neighbor search in steps 1 and 2 retrieving enough geographically relevant scenes to train the SVM. If a query photo is from Pittsburgh and none of the retrieved scenes are nearby, the learning can not hope to recover. We know from im2gps (Chapter 4, Figure 4.6) that for 75% of queries, an image within 200km of the ground truth location is among the $K_{sl} = 120$ nearest neighbors. Thus we can have some confidence that geographically nearby scenes are being included among our nearest neighbors and taking part in the learning process. In Table 6.1 we see that the algorithm is not especially sensitive to the value of K once it is above ~ 100.

In step 5, to convert the all-pairs distance matrix into a kernel matrix, we first find $\mathcal{M}$, the maximum distance between any of the points (including the query point). We subtract all distances from $\mathcal{M}$, thus "flipping" the all-pairs distance matrix into an affinity matrix – large distances become small affinities, and vice versa. This operation does not yet guarantee that the matrix is positive semi-definite, as is required by an SVM. Like [117], we check that the smallest eigenvalue of the kernel matrix is negative and if it is we subtract it from the diagonal of our kernel matrix. In our experiments this condition never triggered when using all features and rarely triggered when using the base im2gps features. The distances from the query point to each of the K nearest neighbors are likewise "flipped" by subtracted them from $\mathcal{M}$.

We train our non-linear SVMs using Chapelle's primal SVM code [12]. We tried a large range of values for the regularization constant λ, but performance was relatively unchanged.[3] For each of $\mathcal{C}$ training operations, the kernel matrix is unchanged – only the class memberships change. The output of each training step is a set of weights on the K training examples and a bias term. The dot product of the these weights and the query distances plus the bias term is the distance from the non-linear decision boundary. The class for which this value is most positive wins. Our geographic classes are of unequal sizes (anywhere from 1 to 100+ photos). The SVMs we learn are implicitly biased towards picking the classes trained with more positive examples and we make no effort to counter this.

We also experimented with passing our all-pairs distances through an RBF-like kernel, $\mathcal{K}(x_i, x_j) = exp(-\frac{dist(x_i,x_j)}{2\sigma})$. We do not get a performance increase and furthermore the performance is very sensitive to values of the kernel parameter σ and the SVM regularization parameter λ.

Complexity and Running Time

As with KNN-SVM, our complexity is linear with respect to N, the number of "base" distances we compute to find K nearest neighbors, and quadratic with respect to K. In our case, $N = 6471706$ and $K =\sim 200$ so our running time is still dominated by the initial search which takes ~ 2.5 minutes (amortized over many queries). The total time per query is ~ 3 minutes. In our implementation we parallelize batches of queries across ~ 100 CPUs.

[3]In general, performance was better when λ was smaller and support vectors were sparser. As λ approaches infinity, all exemplars become support vectors with uniform weighting and the SVM classification degenerates into picking the cluster with the smallest average distance. This heuristic performs nearly the same as picking the largest cluster. Performance with different values of λ and $K = 200$: $\lambda = 1$, 13.75%. $\lambda = 10$ 13.40%. $\lambda = 500$ 13.25%. $\lambda = 50,000$ 9.65%.

K	Biggest Cluster	Lazy SVM
1	8.85%	8.85%
25	10.75%	11.10%
50	10.65%	12.20%
100	10.50%	13.10%
200	9.70%	**13.75%**
400	8.40%	13.00%
800	8.15%	13.15%

Table 6.1: *Performance across values of K.* These experiments were carried out on the im2gps 2k random test set with a 200km accuracy threshold. Picking the largest cluster of scene matches (a heuristic proposed in im2gps) improves performance over 1NN for a range of K. Our lazy learning approach produces significant performance gains as long as K is large enough – about 100 in our experiments.

We have made little effort to optimize the initial search although "tiny images" [100] reports good results from a very low dimensional initial search of PCA bases. Step 1 is amenable to approximation because it does not need to have high precision, only high recall, assuming that step 2 will filter out spurious matches.

6.1.3 Image Geolocation Learning Results

Our learning increases geolocation accuracy on the 2k random test set from 6.25% (1NN) to 10.65% when using the im2gps features and from 8.85% (1NN) to 13.75% when using all features (Section 6.2). As long as K is small, our im2gps heuristic of picking the largest mean shift mode can improve over 1NN, but not nearly as much as our learning (See Table 6.1). Furthermore, our learning is not as sensitive to the value of K.

6.1.4 Discussion

Our results are qualitatively and quantitatively greatly improved by new features and learning. In fact, our geolocation performance now exceeds that of humans (Chapter 7). But there is still a great deal of room for improvement. As Figure 6.2 shows, we are still nearly hopeless at geolocating photographs from sparsely sampled regions unless they show distinct landmarks. While it was hoped that our scene matching might be able to pick up on subtle landscape, vegetation, or architecture cues to geolocate images this is rarely observed. Our

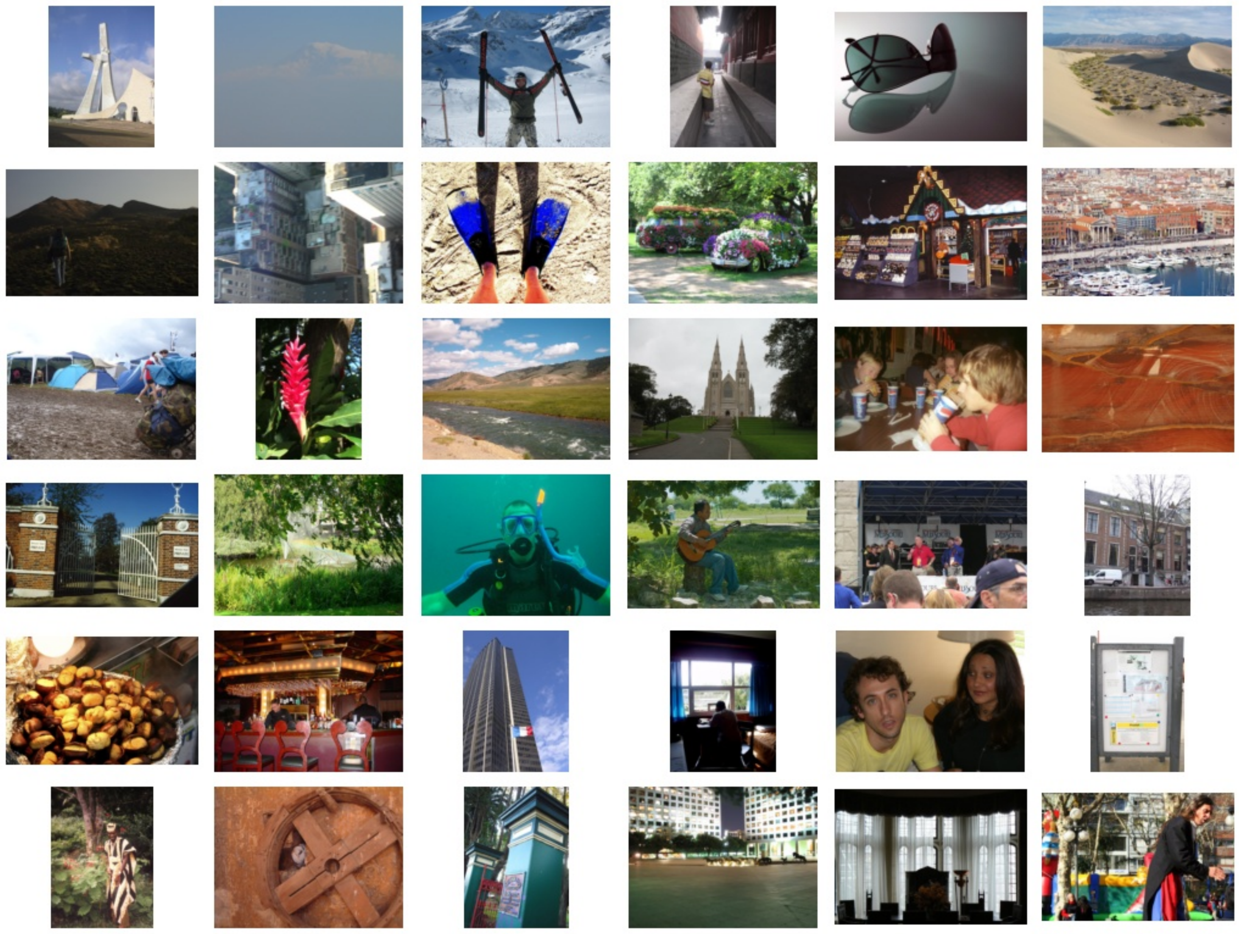


Figure 6.1: *Sample of 2k random test set.* 36 of the 2000 random images in the 2k random test set. Many of these photos are difficult or impossible to confidently geolocate.

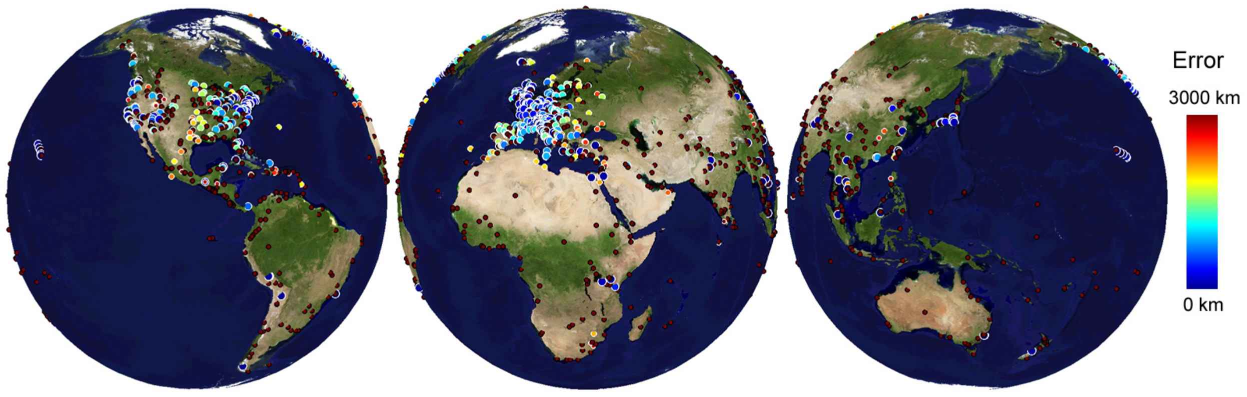


Figure 6.2: *Geolocation learning accuracy across the 2k random test set.* The location of each photo in the 2k random test set is plotted. Each marker is colored and sized according to how accurately it was geolocated by the lazy learning.

algorithm's advantage over humans is its large visual memory, not its ability to relate scene statistics to geographic trends.

Although $\sim$ 13% is low in absolute terms, we saw in Section 4.6 that reasoning about temporal sequences of photographs could raise performance dramatically. Specifically, [50] showed that for each of the 20 test sequences, the geolocation accuracy tends to increase by a factor of four once sequence information is considered.

We have several good lower bounds for geolocation performance, but it is unclear what a reasonable upper bound is? Many of the photos, especially in the 2k random test set, are hopeless. But many others contain clear geographic cues – architecture, writing, clothing, plant and animal species – that the scene matching is not sensitive enough to leverage.

Future Work. Lazy learning is an instance of *transductive* (as opposed to inductive) learning – we are only interested in classifying a specific query point and we get to see that point before building our classifiers. If the classifier knows that it should only try to minimize classification error at one point instead of the entire space it can sometimes perform better. There exist algorithms for transductive SVMs which have been shown to improve performance in the text domain [47], and which could improve performance in this domain as well.

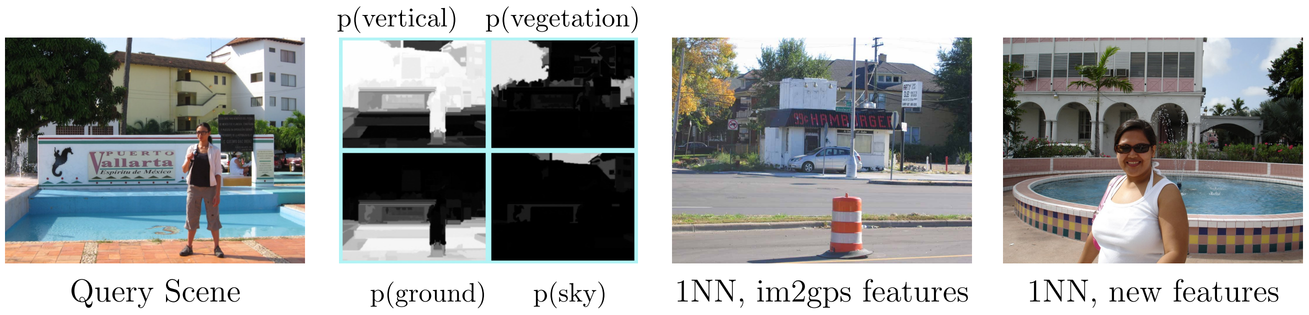


Figure 6.3: For each geometric class we build separate color and texton histograms. Scene matching is improved by restricting the histogram comparisons to corresponding geometric regions.

6.2 Additional Features

This section describes additional scene matching features which are intended to be more robust. These features are used with lazy learning in the previous section to improve geolocation performance. These features are intended to improve upon the "base" im2gps features described in Chapter 4. Two shortcomings of the im2gps features are 1) lack of invariance to scene layout and 2) poor performance at instance-level recognition. To address these problems we describe additional geometry derived and SIFT derived features.

6.2.1 Geometry Derived Features

The scene descriptors used in Scene Completion (Chapter 3) and im2gps (Chapter 4) are all "global"– encompassing statistics of the entire image or built rigidly on top of the entire image. This means that scene transformations which, for most applications, are not meaningful (e.g. cropping the image, shifting the horizon) produce huge changes in the global descriptors and thus huge distances according to our distance metrics. This lack of invariance effectively reduces the size of any database we use, because inconsequential image differences will prevent otherwise good scene matches from being retrieved. We introduce several new features to measure scene statistics in a more robust, content aware manner which can help "see through" these transformations.

Geometry Specific Color and Texton Histograms

Global color and texture histograms are effective features for scene matching but they do have drawbacks – semantically unrelated images can coincidentally have similar histograms, and semantically similar images with viewpoint variations can have very different histograms. To address both of these and try and make the histogram comparisons more meaningful we build histograms for *each geometric class* in an image. For example, texture histograms for vertical surfaces in an image. By restricting texture and color comparisons to geometrically-like regions of images we expect their distances to be more reliable.

We use geometric context [40] to estimate the probability of each image region being "ground", "vertical", "sky", or "porous" (i.e. vegetation). For any pixel, the probability of "ground", "sky", and "vertical" sums to one, while "porous" is a subset of "vertical". We build color and texture histograms for each geometric class by weighting each pixel's contribution to each histogram according to the geometric class probabilities. We also build global texture and color histograms in which the "vertical" pixels get much higher contribution (the intuition being that the appearance of vertical image content is more likely to be correlated with geolocation than the sky or ground). Our approach is similar to the "illumination context" proposed in Photo Clip Art [59] in which scenes are matched with color histograms built from ground, sky, and vertical image regions.

The geometric context classification is not entirely reliable, especially for unusual scenes, but the mistakes tend to be fairly *consistent* which is arguably more important than accuracy in this task (e.g. if clouds were 100% classified as "vertical", our feature distances would still be reasonable because the scenes would be decomposed into consistent, although mixed, semantic groups). The geometric context probability maps are themselves resized to 8x8 image features.[4]

Geometric Region of Interest Gist

We introduce a variant of the scene gist descriptor (the single most successful feature in im2gps). The gist descriptor is very sensitive to the alignment of scene structures, which is a useful property if we want to retrieve scenes with the same "spatial envelope", but a limiting factor for geolocation where we would rather be invariant to the location of the horizon, the focal length of the camera, or whether the photograph is landscape or portrait oriented. To make the gist robust to the transformations, we try to consistently build it centered on the

[4]im2gps explored such a geometric context probability map, but for computational reasons it was computed at lower resolution and thus less accurate.

horizon and ignoring empty "negative space". We build the gist descriptor over the square region of interest that maximizes the following score:

$$score(x_{min}, y_{min}, width) = mean(p_{gnd})^{\lambda_{gnd}} * mean(p_{vrt})^{\lambda_{vrt}} * mean(p_{sky})^{\lambda_{sky}} \\ * std(p_{gnd}) * std(p_{sky}) \qquad (6.1)$$

Where p_{gnd}, p_{vrt}, and p_{sky} are vectors containing the per-pixel ground, vertical, and sky probabilities within the region and $\lambda_{gnd} = .25, \lambda_{vrt} = 2, \lambda_{sky} = .25$. There is also a bias towards larger, more centered regions of interest. A homogenous region of sky or ground would score badly, while a geometrically diverse region with equal amounts of sky and ground would score well. Figures 6.2.1 and 6.2.1 show examples of the regions of interest found with this heuristic.

As an alternative to parameterizing features with respect to geometry, we attempted to build features with respect to the horizon. Inspired by [41], we manually labelled the horizon in 6500 random scenes and estimated the horizon for a novel scene from the k nearest neighbors according to the gist descriptor. Whereas [41] worked within the LabelMe database, we work with more diverse Flickr images. Even after labelling 6500 scenes, the non-parametric horizon estimation was not accurate enough to be useful. Flickr images require vastly more training examples before scenes can be matched reliably. One indicator of the amount of non-traditional scenes in our Flickr data is that for 37% of the Flickr scenes we labelled, the horizon was ambiguous (e.g. a close up of an object) or out of frame (e.g. looking up at a skyscraper). Another horizon estimation heuristic is proposed in [58] – select the y value halfway between the lowest sky pixel and highest ground pixel in the geometric context map. Our geometric region of interest is similarly derived from a geometric context map, although with 3 degrees of freedom (x, y, scale) rather than 1 (y).

6.2.2 Bags of SIFTs

SIFT [65] derived features have been used for scene matching with spatial pyramids composed of sparsely [60] or densely [64] sampled SIFTs.[5] In these cases the quantization of visual words is typically rather coarse (500 visual words). These scene matching strategies are potentially advantageous over the gist descriptor because they can cope with moderate

[5]Histograms of densely sampled, quantized SIFT features are arguably capturing the same information as texton histograms. However, sparse SIFT features, sampled only at interest points, are capturing something qualitatively different than texture.

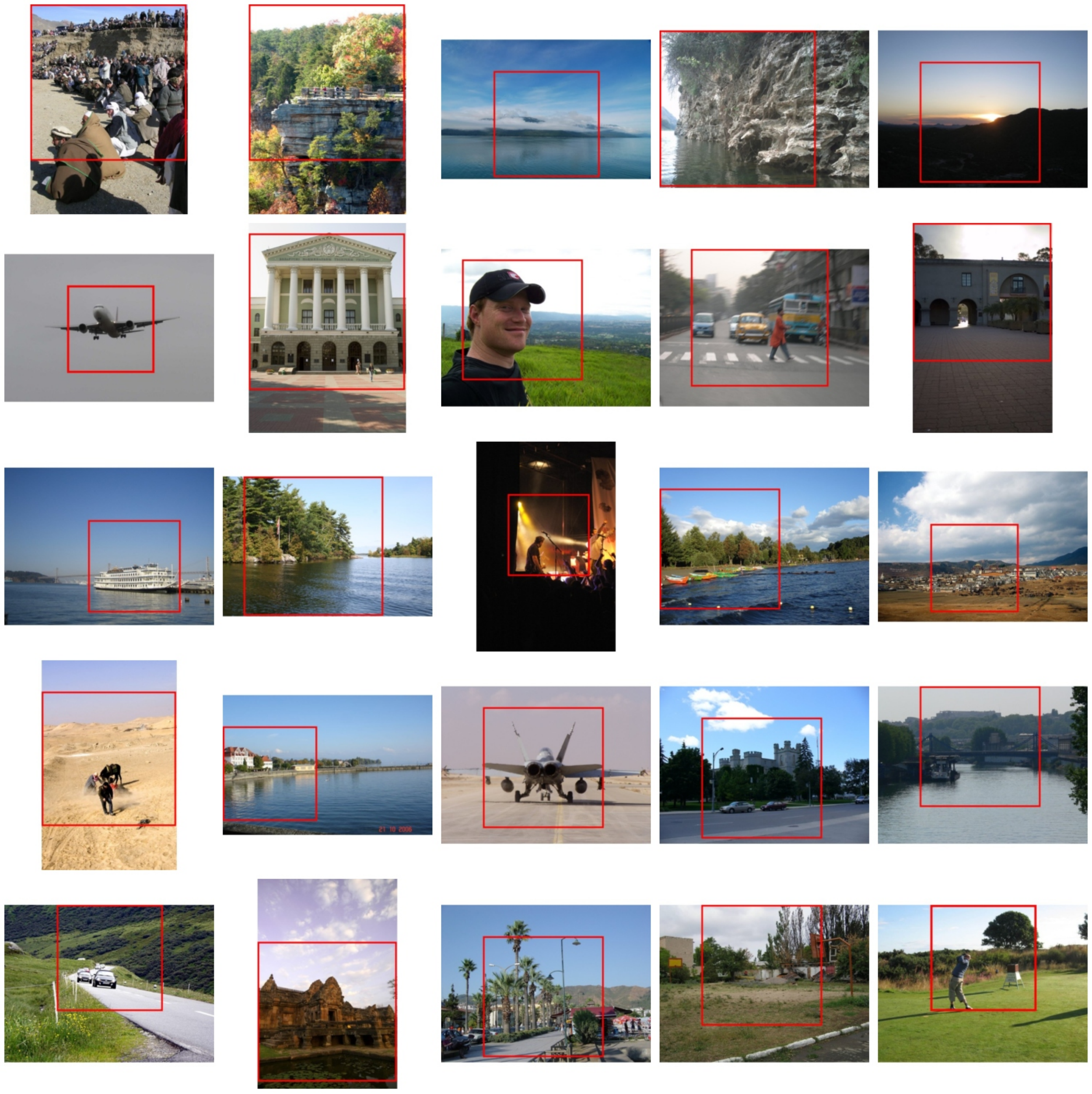


Figure 6.4: *Geometric Regions of Interest.* The regions of interest tend to center on the horizon. However, if the horizon is near the top or bottom of the image this requires the region of interest to become quite small because its aspect ratio is fixed.

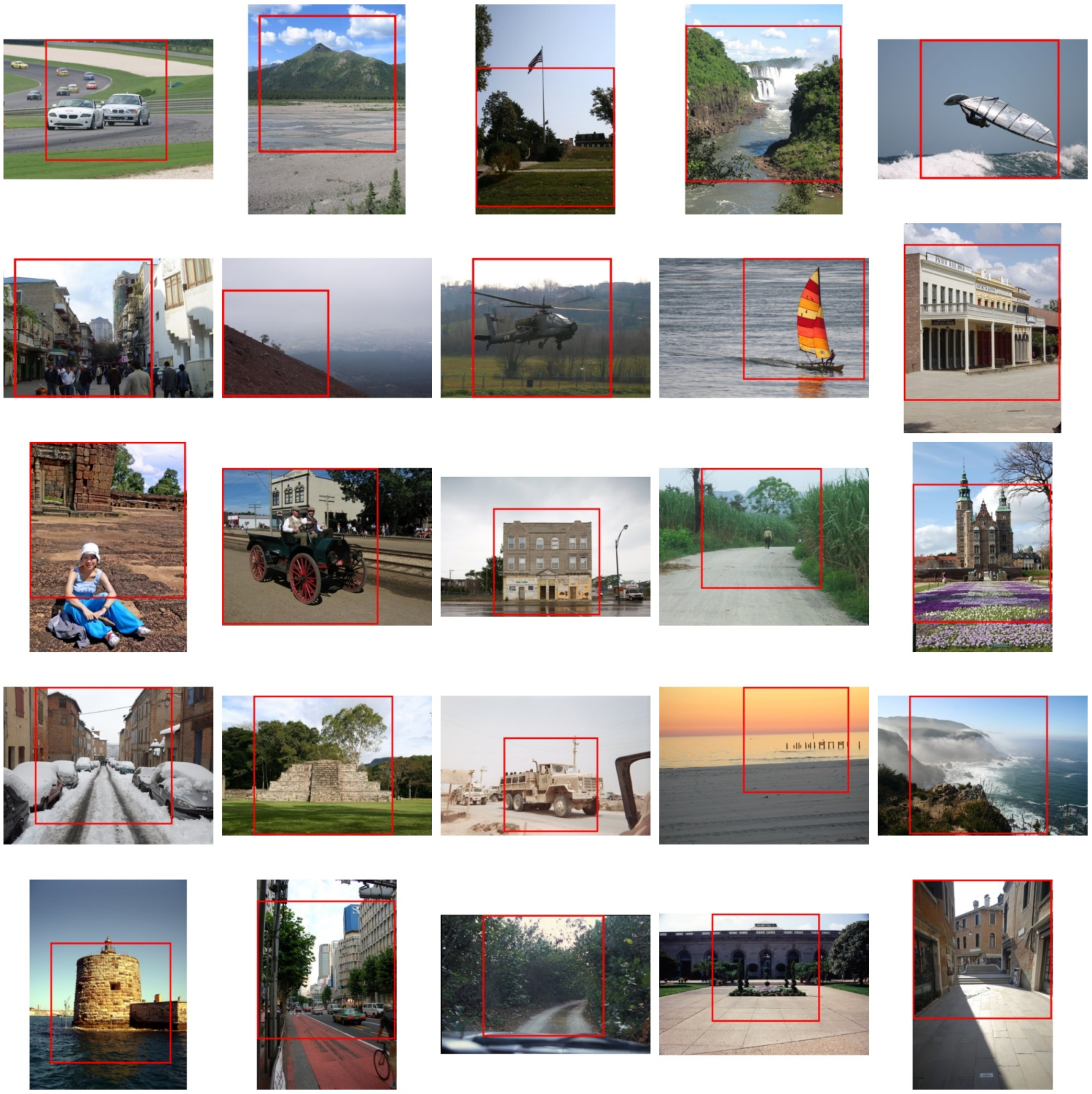


Figure 6.5: *Additional Geometric Regions of Interest.*

misalignments between otherwise similar scenes more easily than the fixed spatial binning of the gist. Quantized SIFT features have also been shown to work well for instance-level recognition in large data sets [88]. Larger vocabularies (1 million visual words) and geometric verification of candidate matches improve performance further [77]. Recent landmark geolocation works [17, 119] have relied entirely on these types of features.

Inspired by these successes, we compute SIFT features at image locations and scales of interest detected by Hessian-affine and MSER [68] interest point detectors. For each interest point detector, we build vocabularies of 1,000 and 50,000 visual words based on a random subset of the database.[6] The intuition is that a vocabulary of 1,000 captures texture qualities of the scene while a vocabulary of 50,000 captures instance specific (landmark) image elements.

6.2.3 Feature Results

In total, including the features from im2gps, we have an over-complete set of 22 elementary features.[7], The hope is that learning will be able to focus on the appropriate feature set for any query. As we showed in the previous section, lazy learning dramatically improved geolocation accuracy when using these additional features. But as a baseline we evaluate the use of all of the features in unison for the image geolocation task. We use L1 distance for all image features (gist, geometric context maps) and χ^2 (chi squared) measure for all histograms (texture, color, lines, SIFT). The distances are scaled so that on average they have equal standard deviation among the top 1000 matches and thus influence the ranking equally. The scene matching process is implemented hierarchically – first, 2000 nearest neighbors are found with the im2gps features and then distances are computed for the new geometry derived and SIFT features and the matches are reranked.

Compared to the im2gps features, the new features perform significantly better at instance level recognition, as would be expected from the new large-vocabulary SIFT histograms. Scene matches for more "generic" scenes are also improved. Figures 6.2.3 and 6.2.3 show cases where the first nearest neighbor with the new features is dramatically improved from

[6]To build the visual vocabularies, we use 20 million SIFTS sampled from roughly 1 million images. To build the 50,000 entry vocabularies a two level hierarchy is used, as k-means would otherwise be prohibitively slow. The hierarchy is only used to construct the vocabulary after which the leaf nodes are treated as a flat vocabulary. We use "soft assignment" as described in [78], assigning each SIFT descriptor to its nearest 5 vocabulary centers, inversely weighted by distance. Because we use soft assignment, the 50,000 entry histograms are not sparse enough to merit an inverted file system search.

[7]The im2gps features total 2201 dimensions, while the features proposed in this chapter total 109,436 dimensions (dominated by the two, 50 thousand entry SIFT histograms).

the im2gps features. For common scene types under canonical viewpoints the difference is less noticeable.

Using the 237 image im2gps test set and base im2gps features, the first nearest neighbor is within 200km of a query 16.5% of the time. Using the 4 SIFT histograms by themselves (after the initial hierarchical search) gives an accuracy 18.6%. Using all features raises accuracy to 21.1%. With the more challenging 2k random image test set, the im2gps 1NN is correct only 6.35%, SIFT features 8.05%, and all features 8.85%.

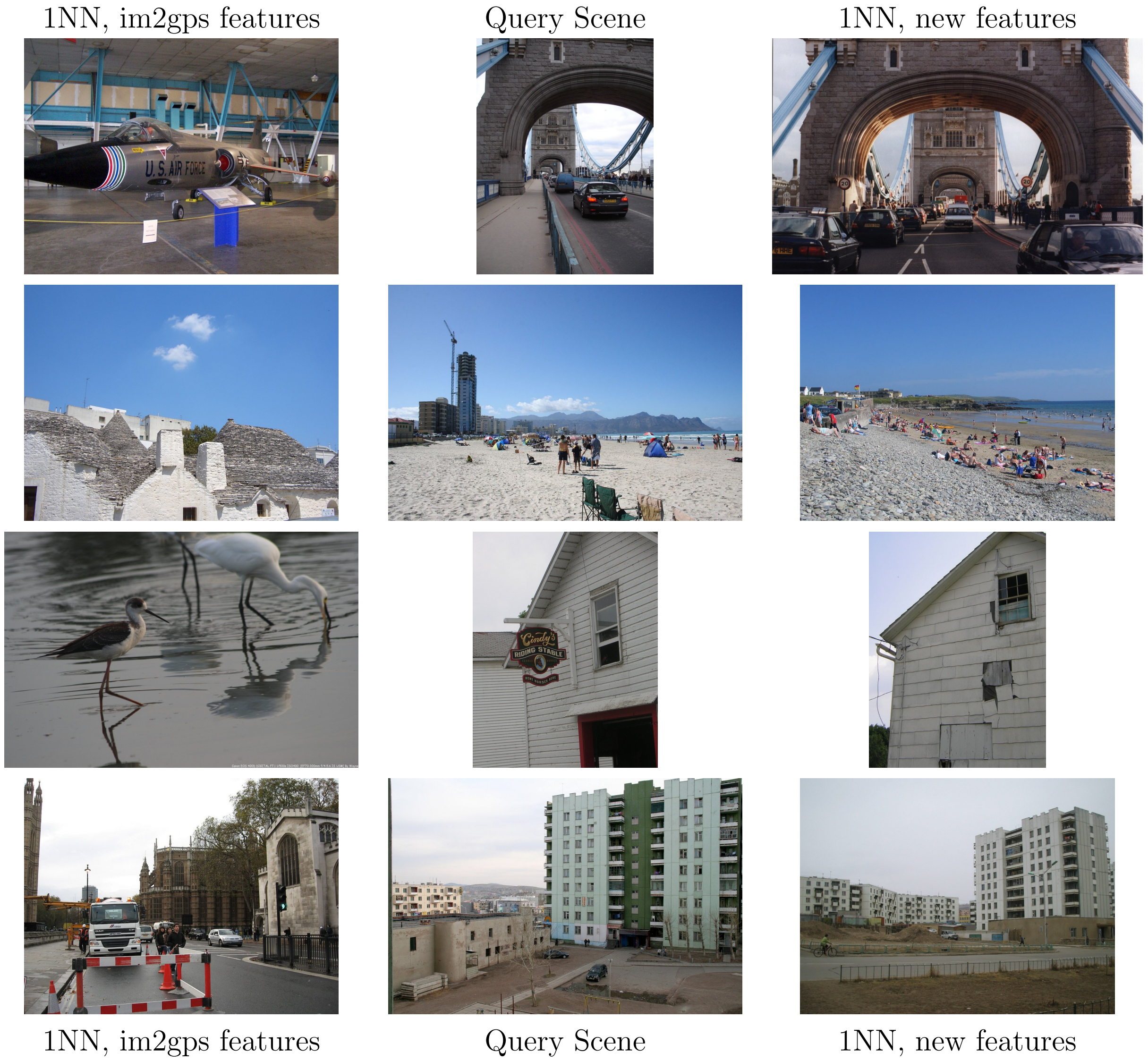


Figure 6.6: *Scene Matches with New Features.* The new features are dramatically better at landmark recognition, especially when the viewpoints do not match, as in the top row. This is to be expected from the SIFT features. The remaining figures show non-landmark scenes for which the matches are much better. The last row is an ideal case – even though an exact, instance-level match can not be found, the new features have found a scene that is architecturally very similar. Even more impressive, both photos are in Mongolia where there are few photos to match to.

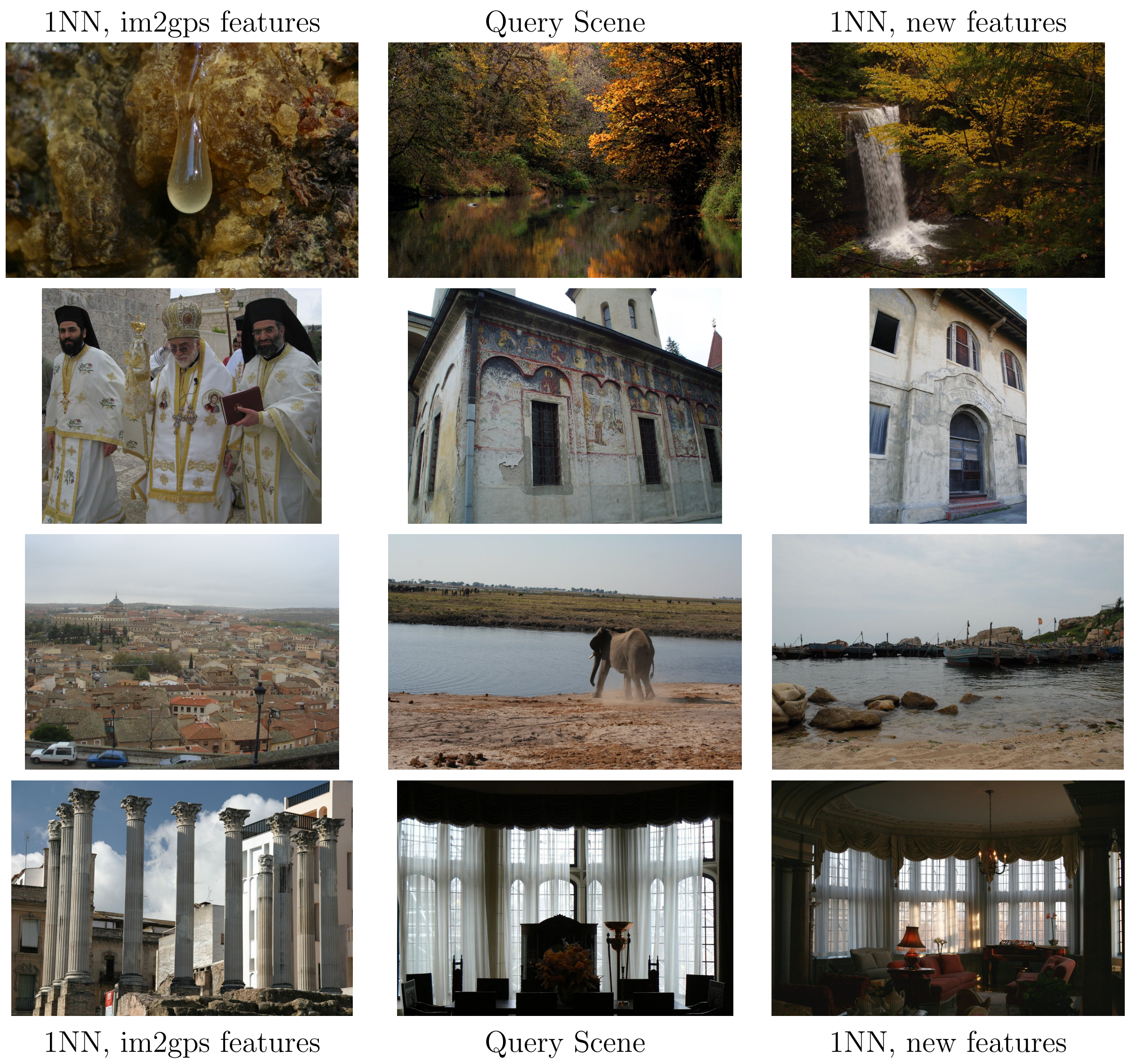


Figure 6.7: *Additional Scene Matches with New Features.* In the last row, the query and new feature 1NN are both in Toronto, but it is not the same physical scene.

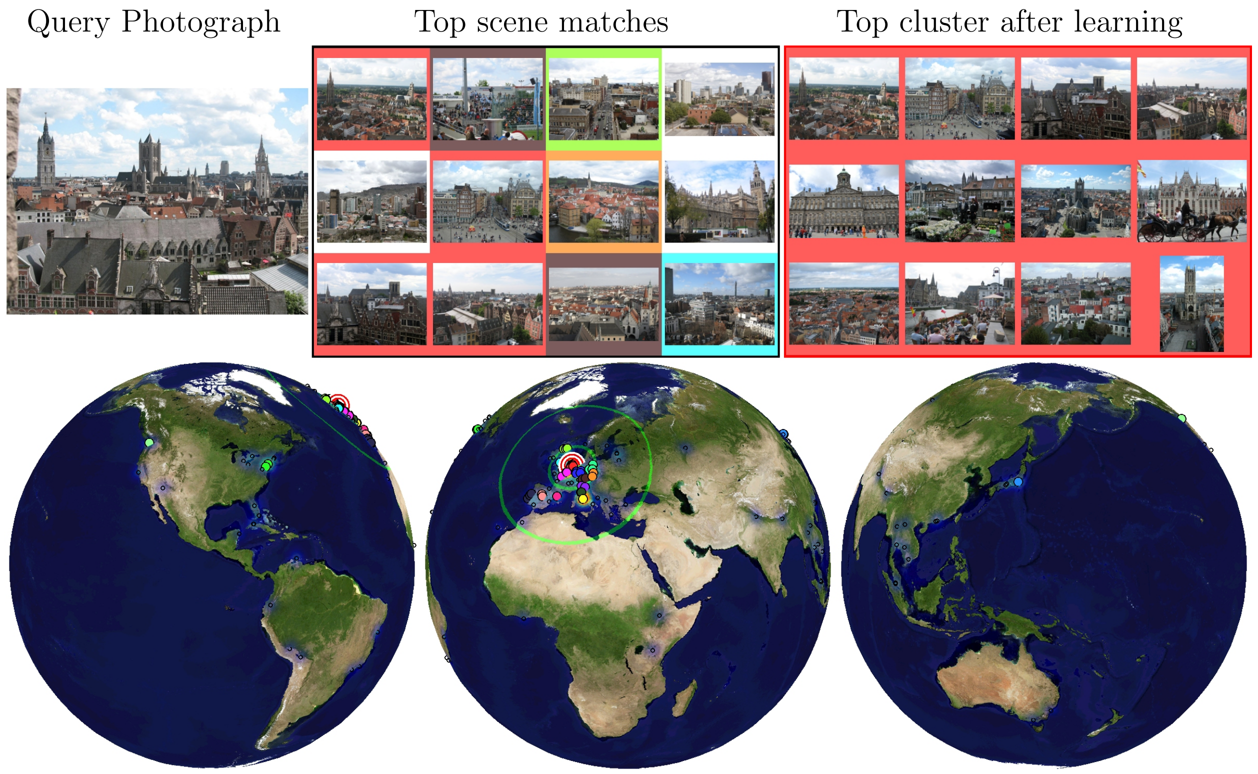


Figure 6.8: *Results 1.* All results were generated from $K = 200$ nearest neighbors clustered with a mean shift bandwidth of 200km and a minimum cluster size of 3. The scene match montages are scanline ordered according to scene match distances. The colors of scene match borders and globe markers indicate cluster membership. The coloring of the clusters indicates their ordering by cardinality – yellow is largest, then cyan, magenta, red, green, and blue. The geolocation estimate from learning is indicated by the red and white concentric rings. The ground truth location is marked by concentric green rings of radius 200km, 750km, and 3000km. The density of scene matches on the globe is indicated by a jet colormap overlay. Scene matches without a cluster are plotted as black rings.

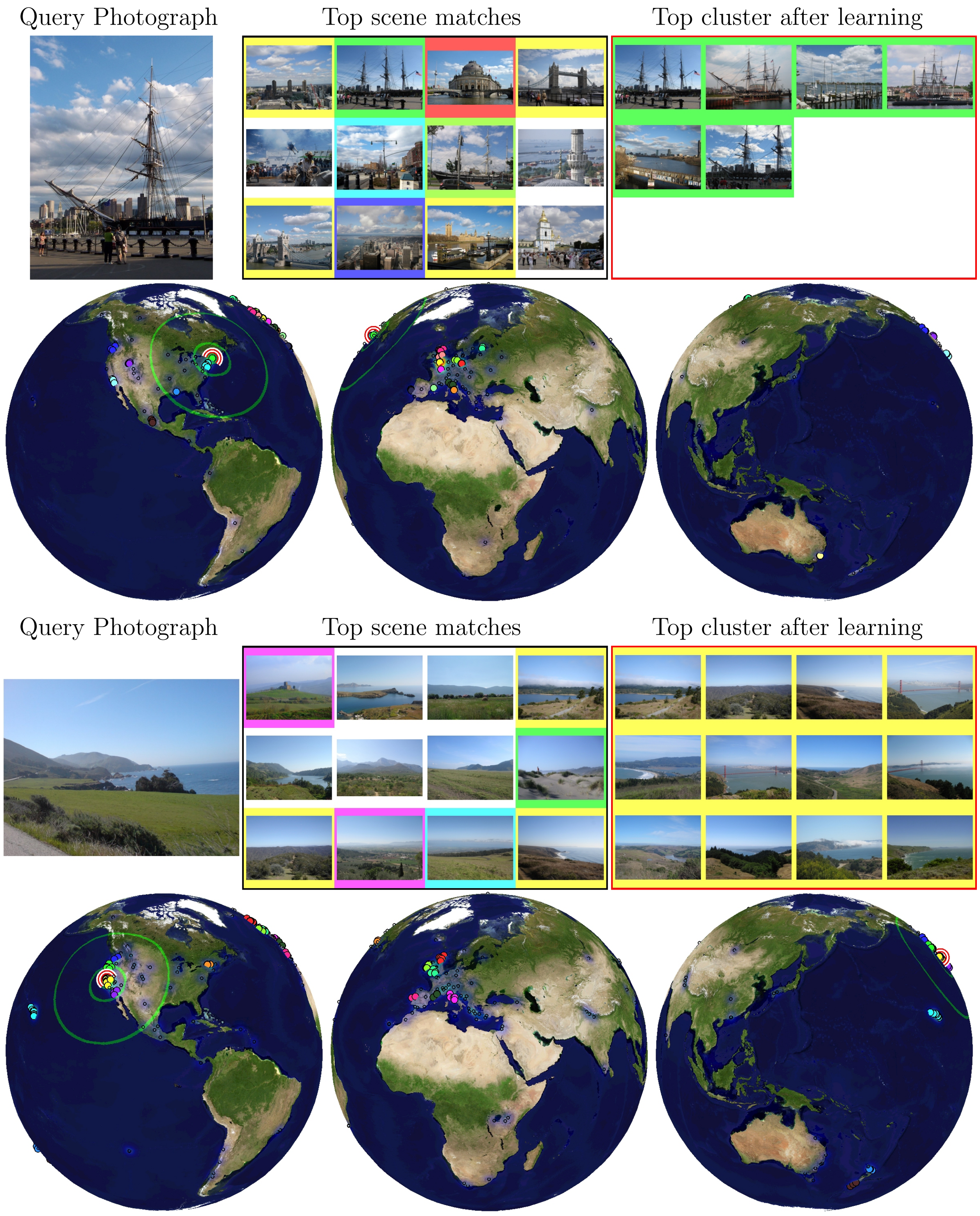


Figure 6.9: *Results 2.*

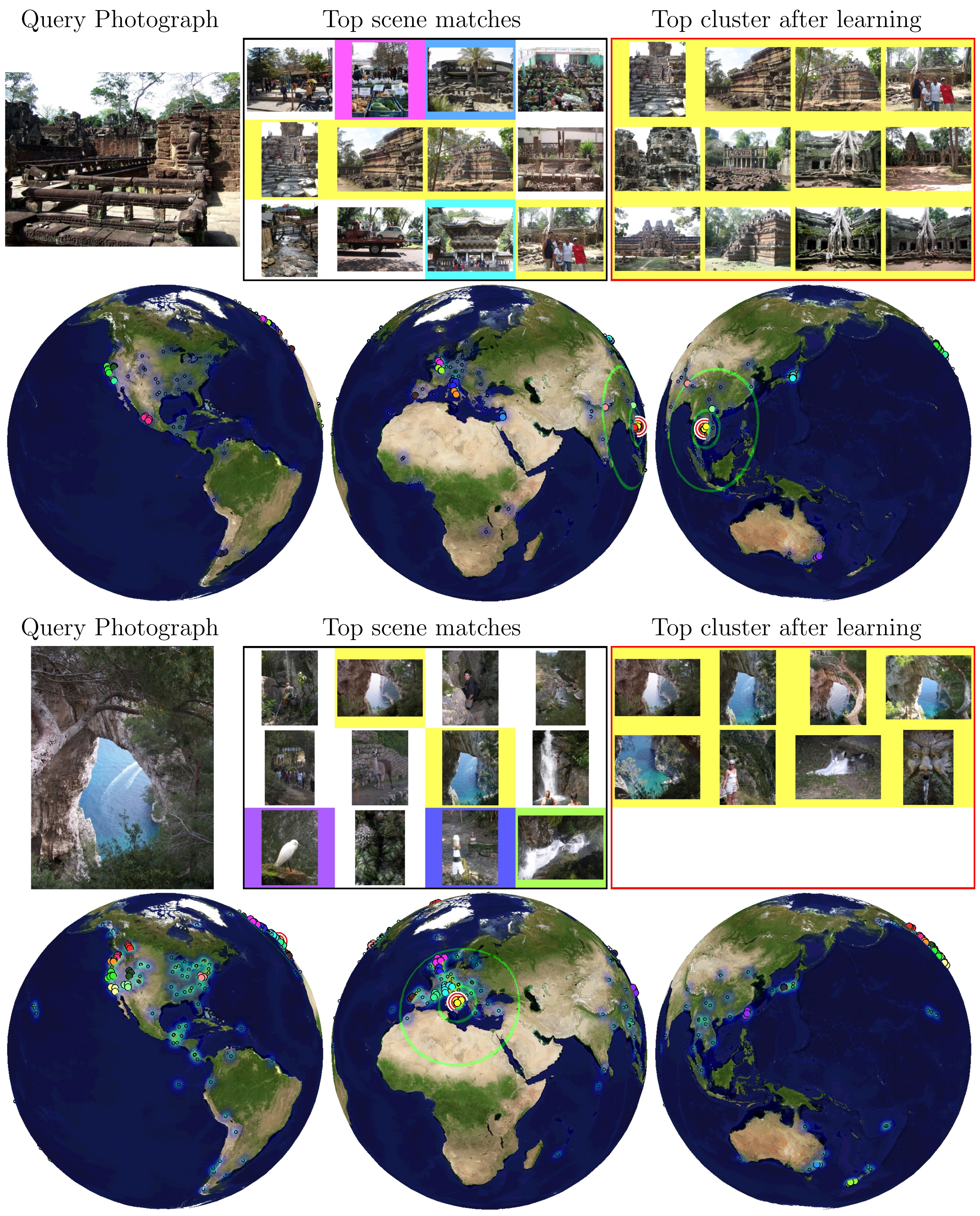
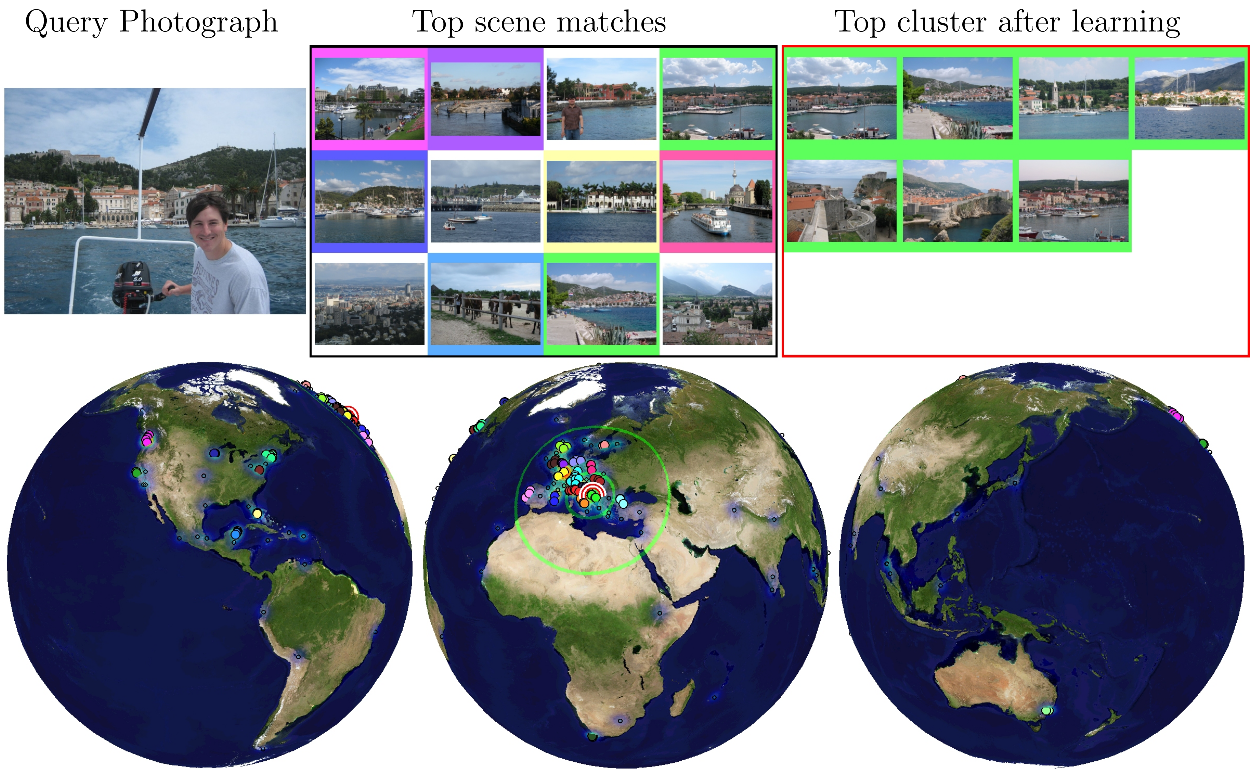


Figure 6.10: *Results 3.*



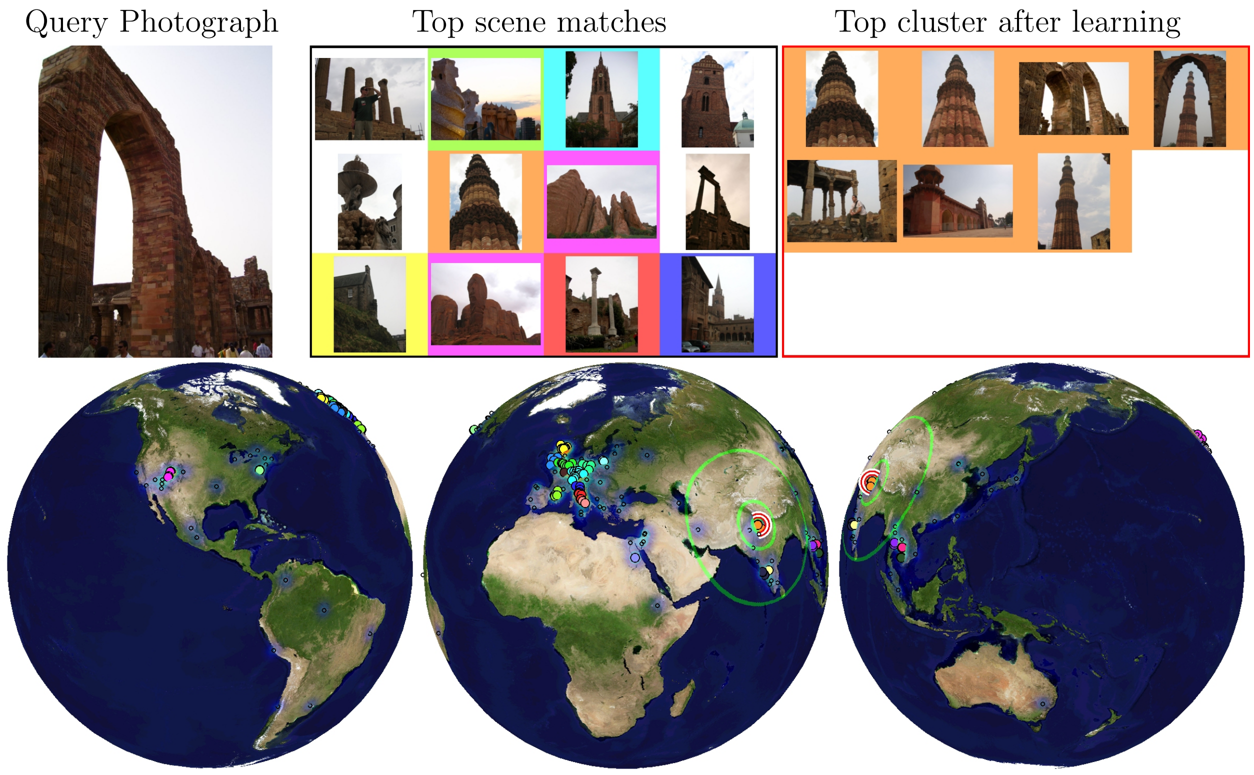
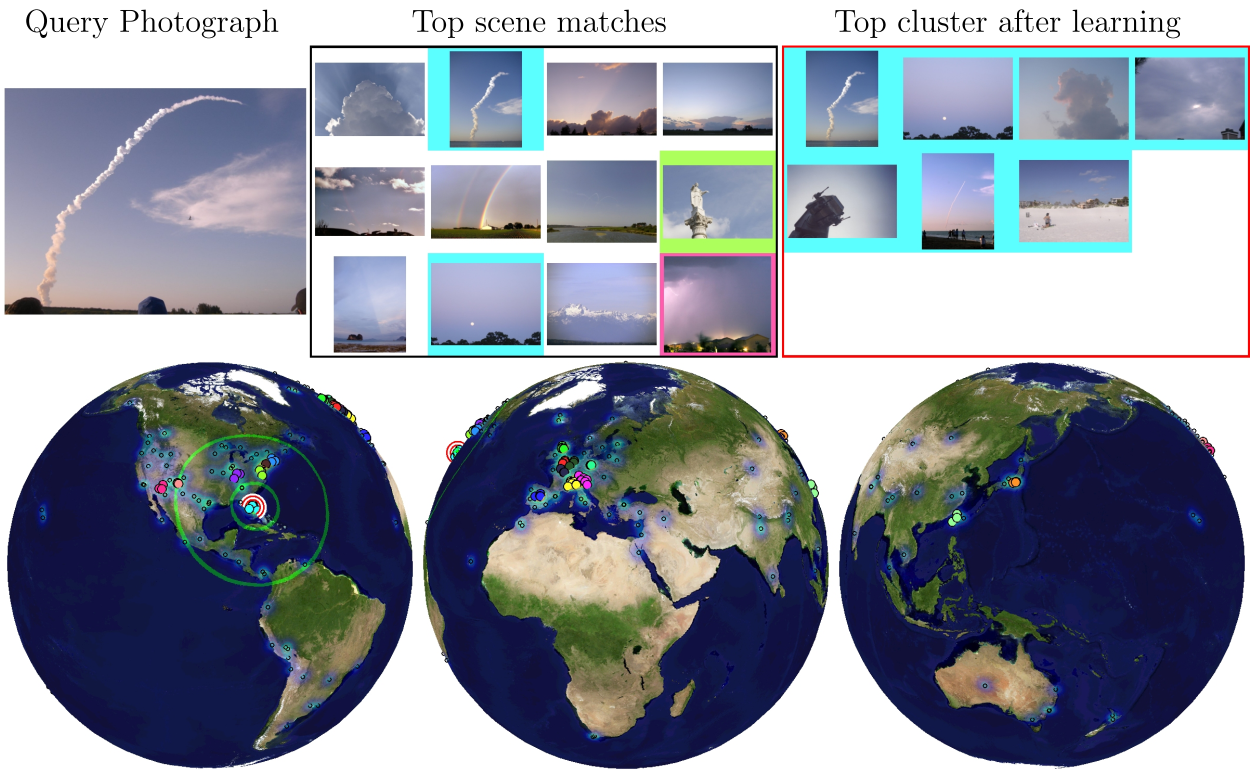


Figure 6.11: *Results 4.*



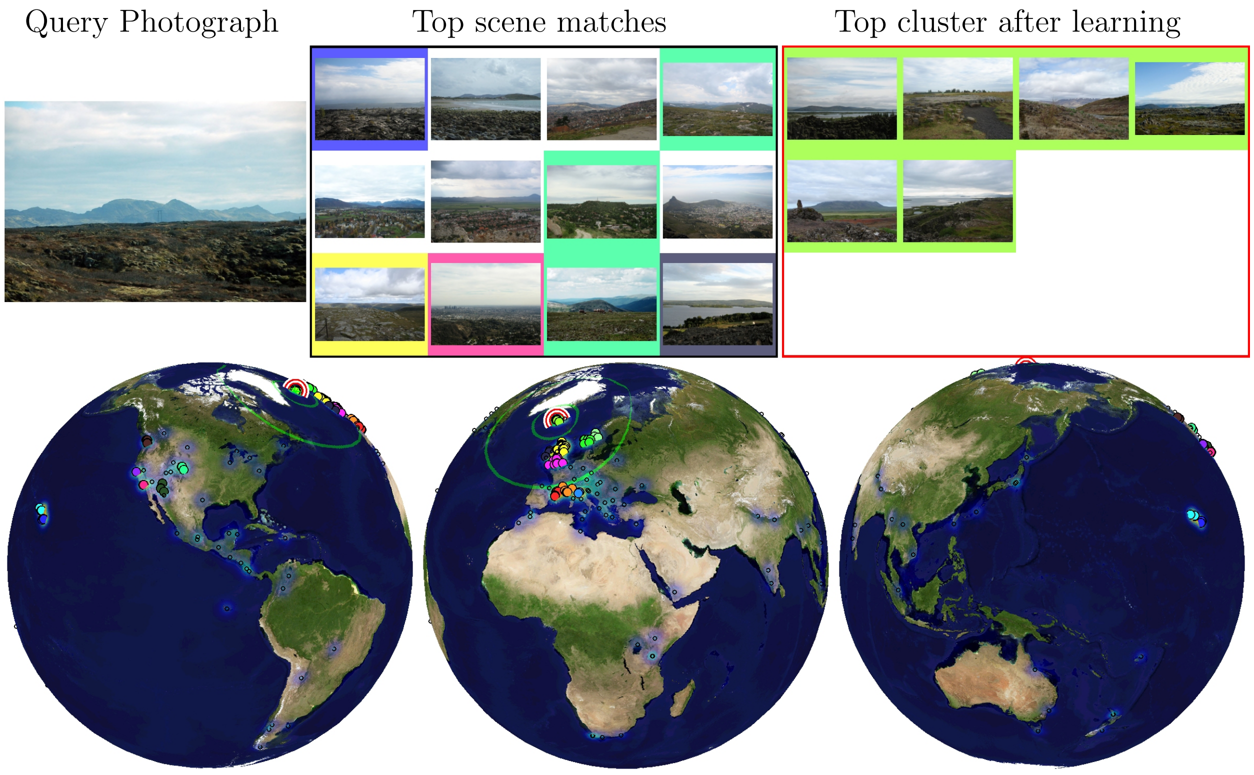


Figure 6.12: *Results 5.*

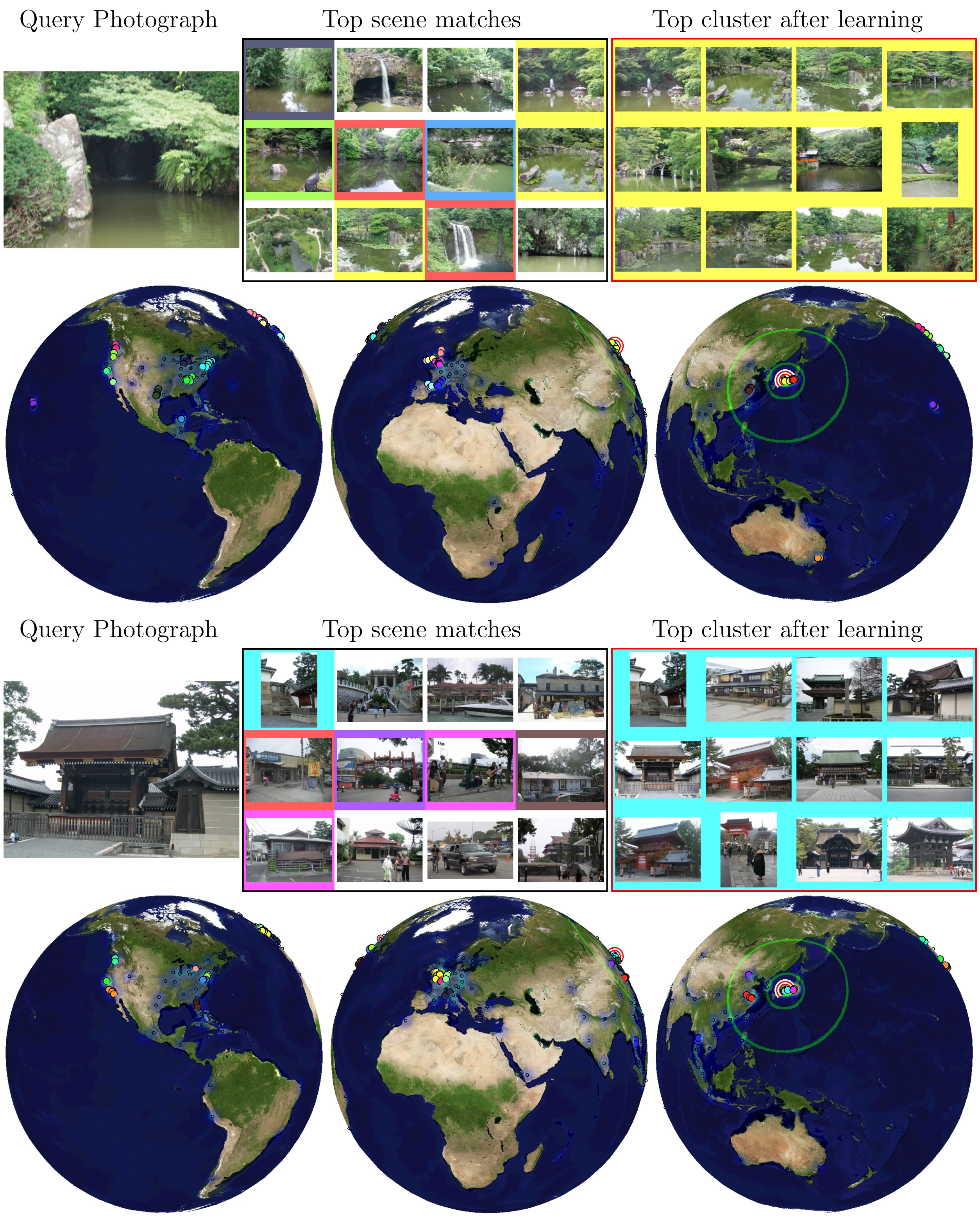


Figure 6.13: *Results 6.*

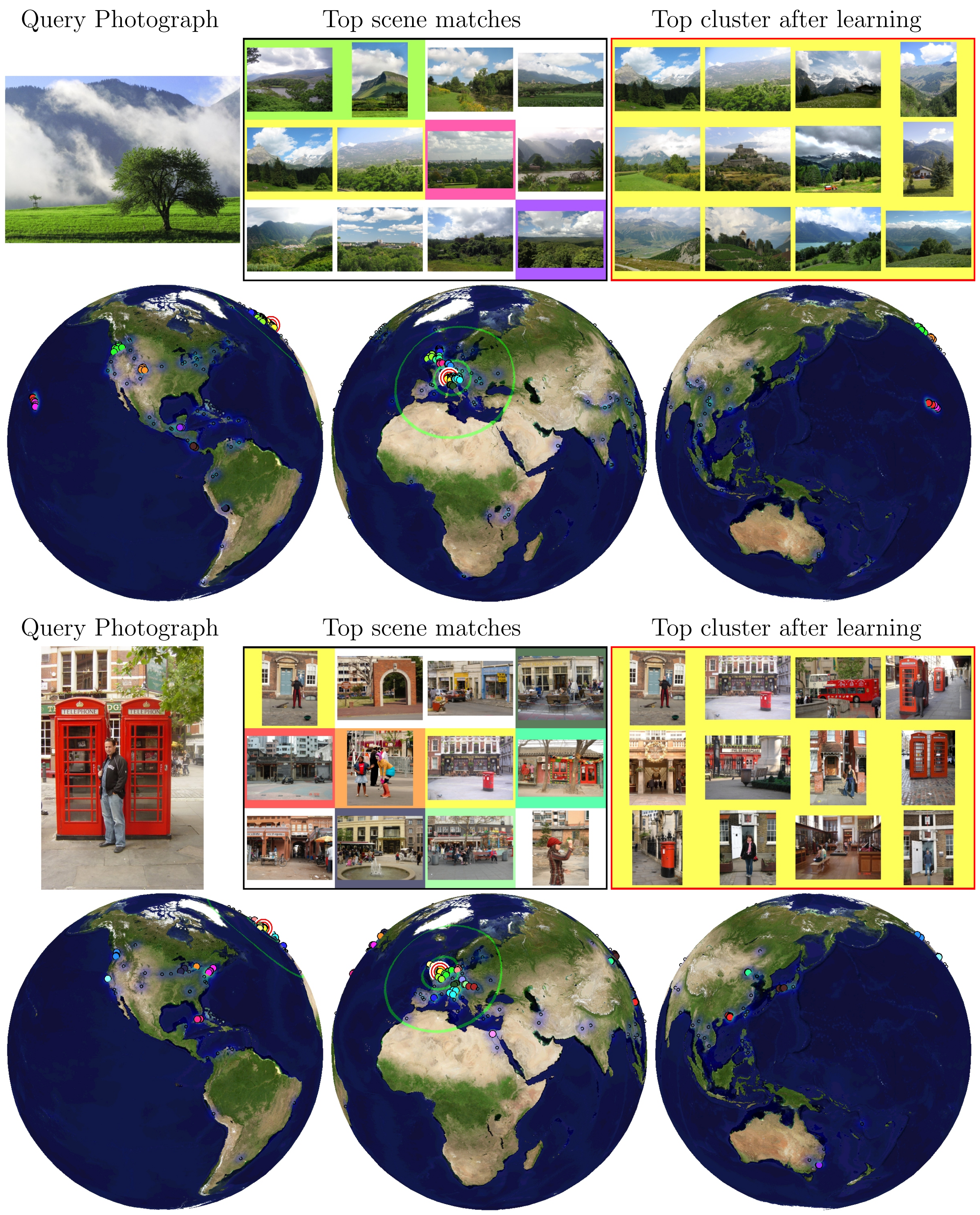


Figure 6.14: *Results 7.*

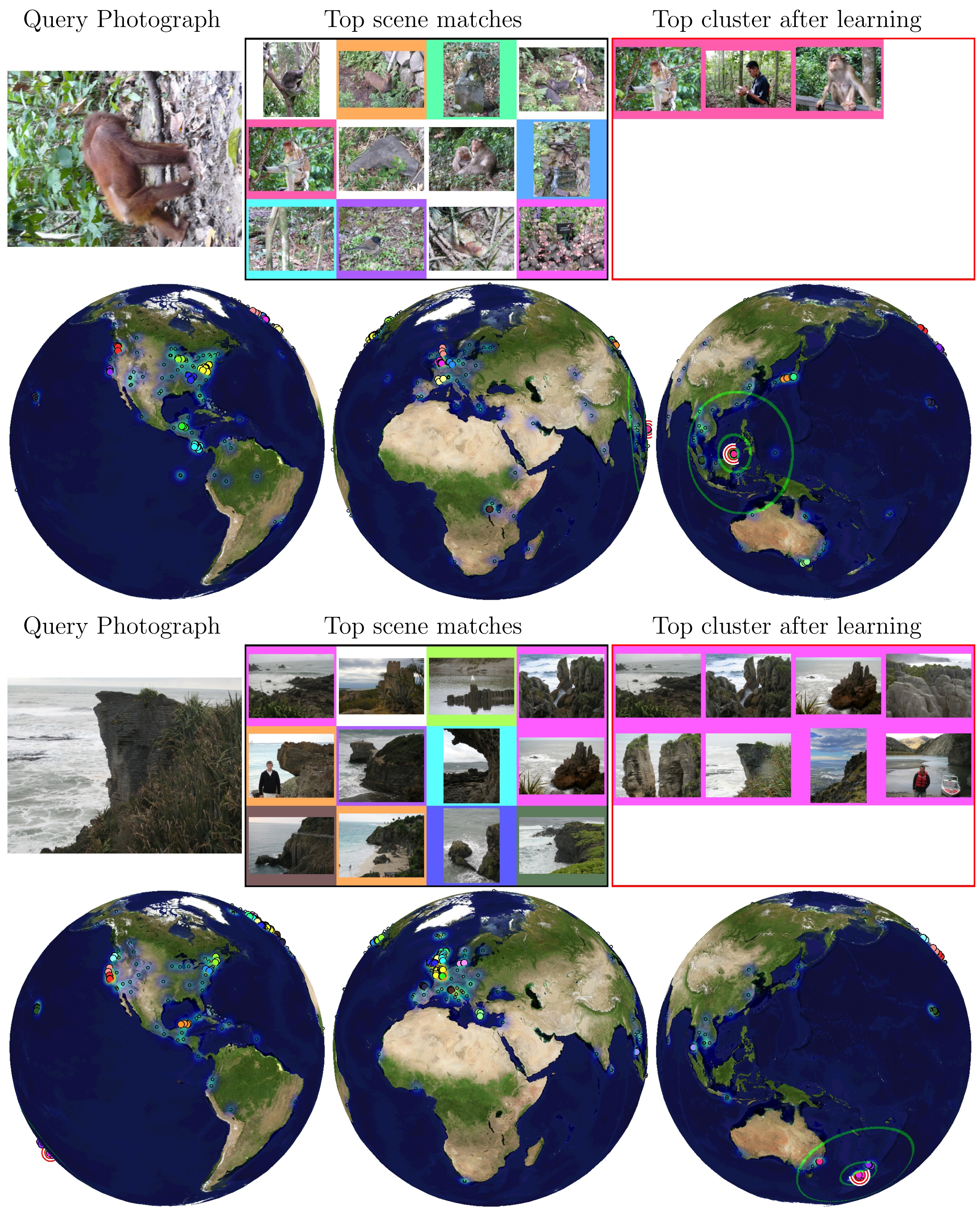


Figure 6.15: *Results 8.*

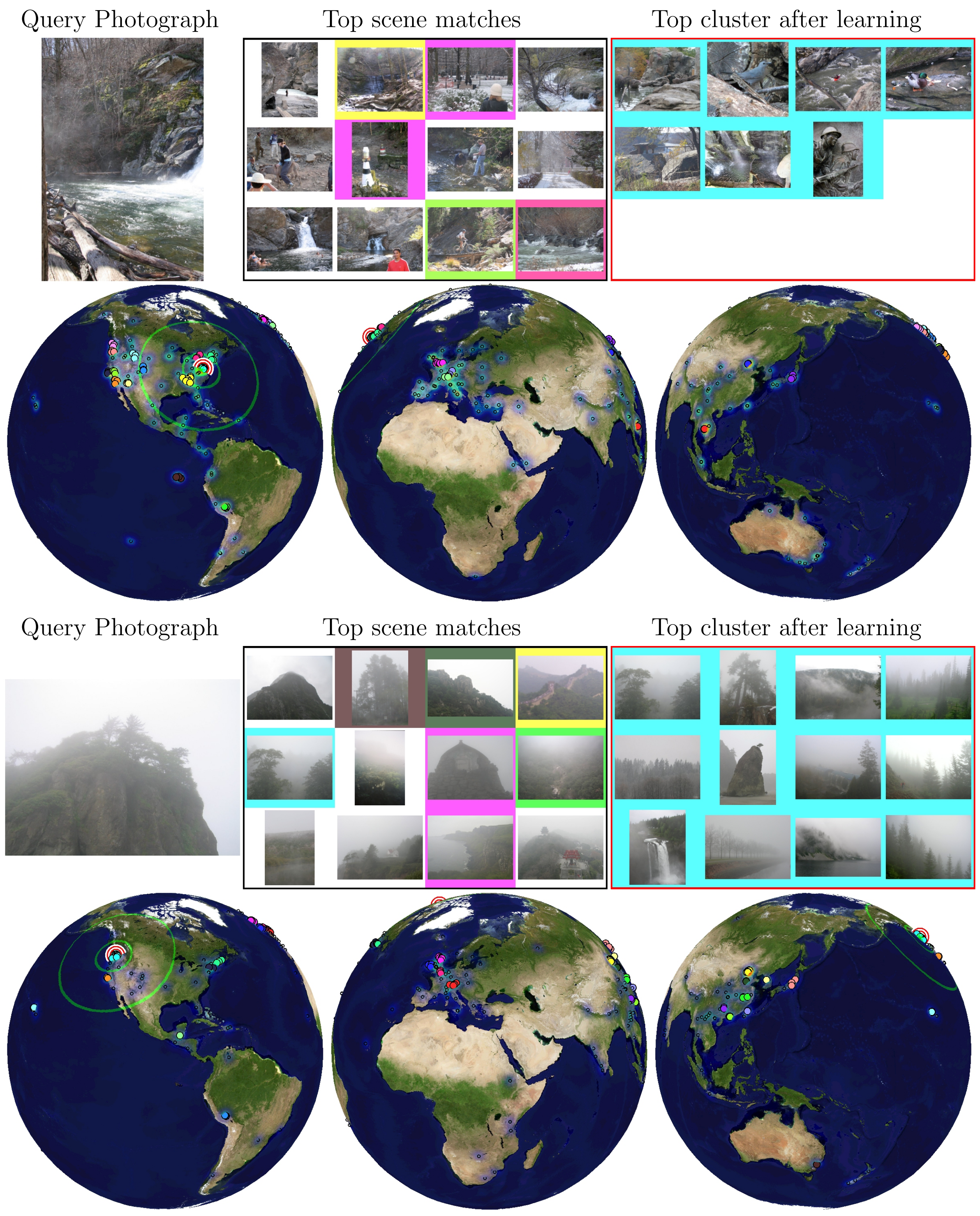


Figure 6.16: *Results 9.*

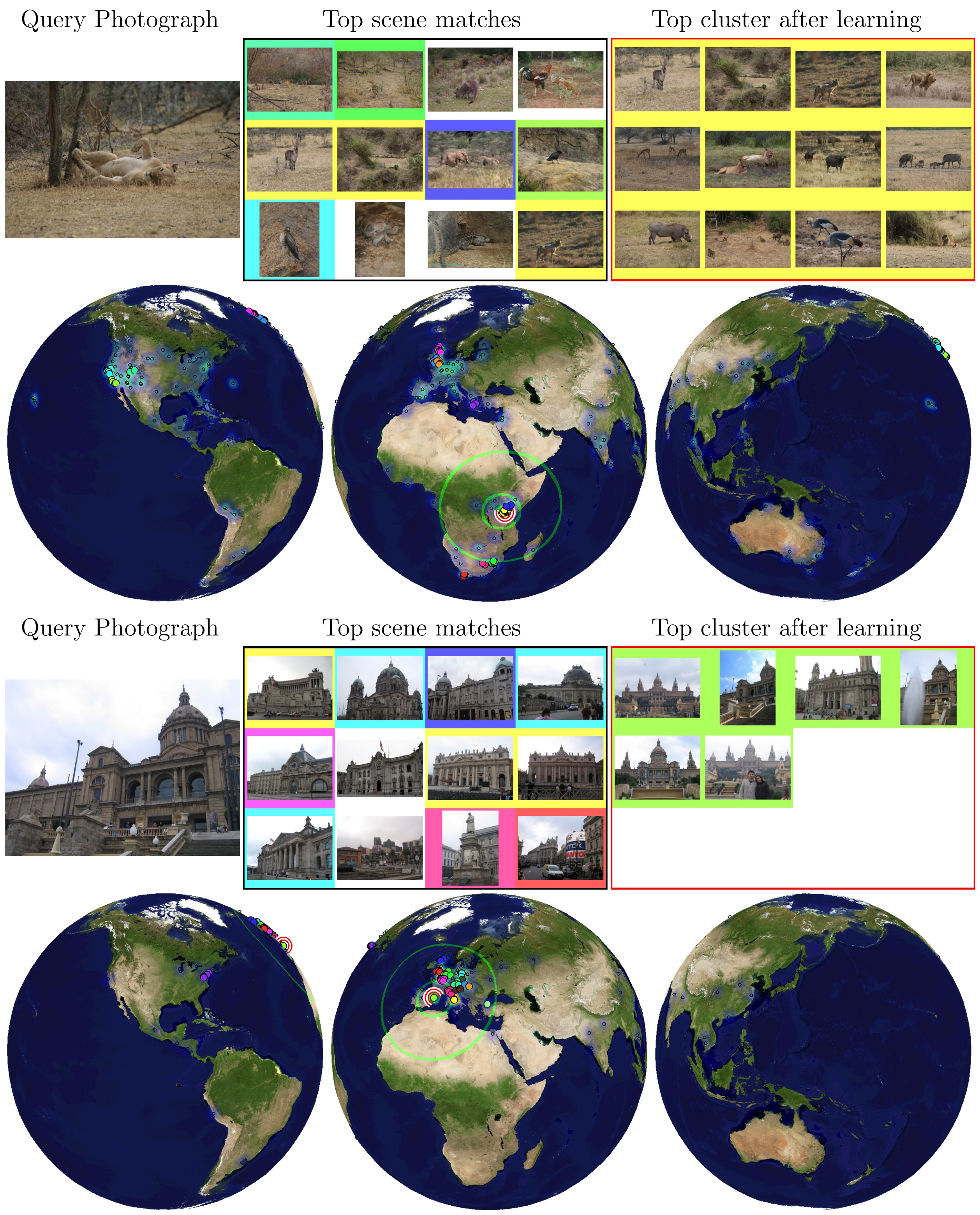


Figure 6.17: *Results 10.*

Chapter 7

Human Geolocation of Images

Chapter Summary In this chapter, we examine human photo geolocation accuracy and compare that to our computational methods. Our goals are 1) to learn about human scene understanding strategies and capabilities and 2) to gain a point of comparison to help us understand our computational methods.

7.1 Introduction

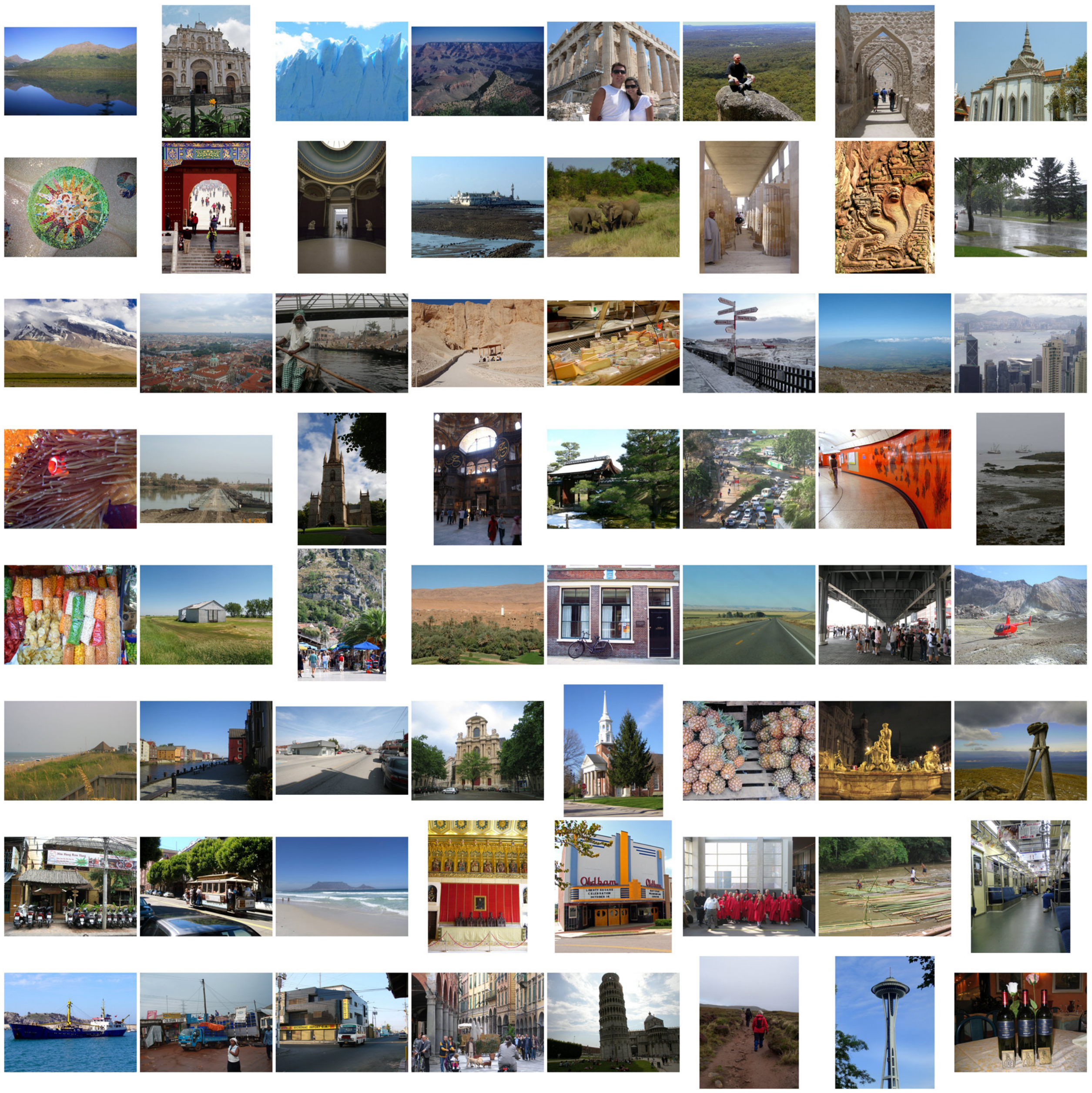
Previously, both human and computer vision researchers have mostly been interested in scene classification as it relates to rigid semantic categories (e.g. kitchen, bedroom, forest, city, etc...). It turns out that in this regime, simple gist or texture features can classify scenes almost as well as humans [82, 105], even with a small amount of training data. However, the success of computational methods might simply be due to the small number of scene categories, and the ease with which these hand-designed categories can be separated by low-level features. Here we investigate a much harder task. We examine human performance at organizing scenes according to geographic location on the Earth rather than hand-defined semantic categories. Participants are shown novel images and asked to pick the location on a globe where the photograph was taken. This is a difficult, memory intensive task – many scenes are geographically ambiguous while others require high-level scene understanding and knowledge of cultural or architectural trends across the Earth.

In addition to gaining some insight into human scene understanding strategies and capabilities, human performance gives us a benchmark that will help us understand our data-driven computational method, im2gps (Chapter 4). This helps us overcome one difficulty in evaluating the im2gps – that we don't necessarily have an intuition for how challenging the photo geolocation task is.

7.2 Perceptual Study

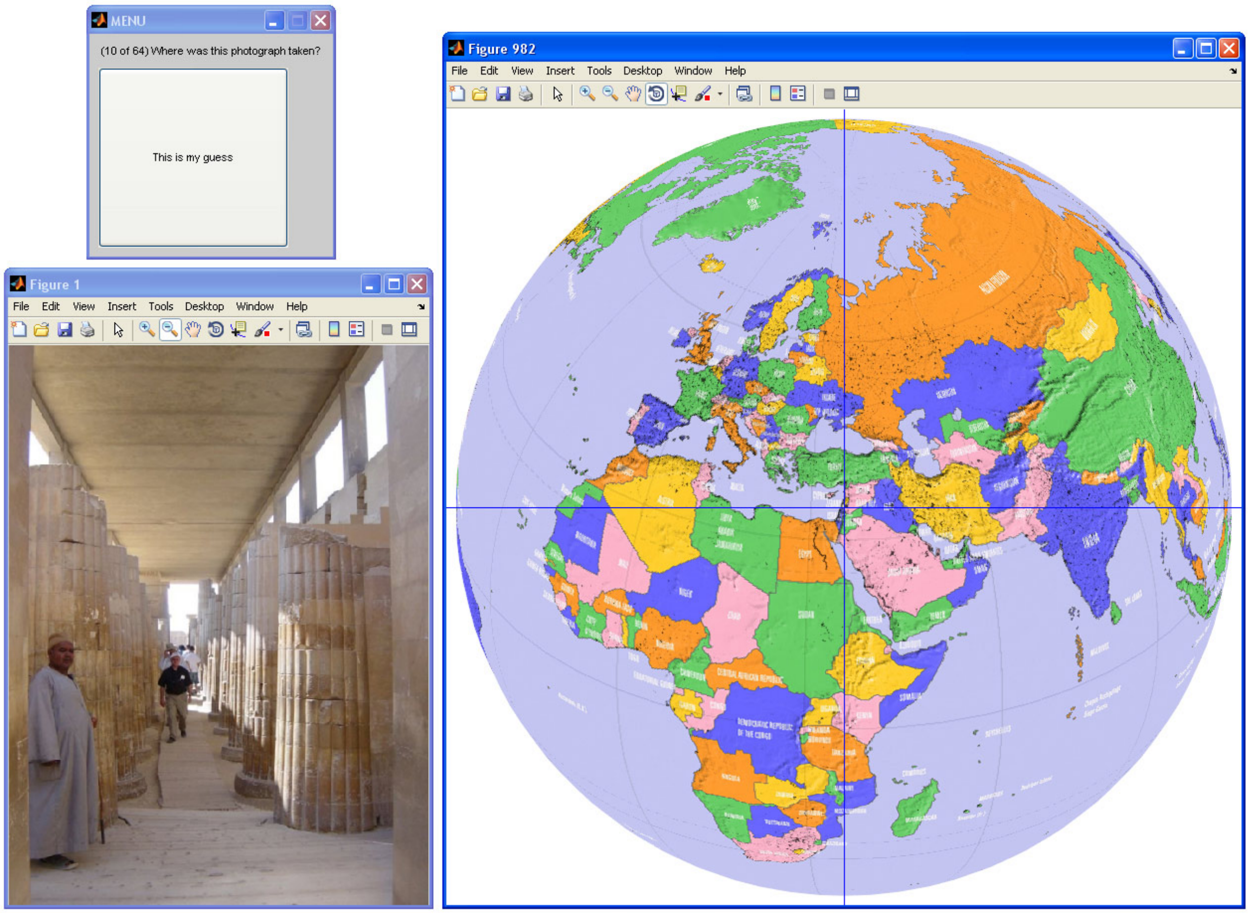
Our first task is to design a test set to present to both human participants and computational methods. We want each participant to view the entire test set so that the per-image performance statistics are robust. We also want the test set to be small enough such that humans do not become fatigued during the study. Therefore we limit ourselves to 64 photographs. The test set is built by manually filtering 1000 random images from the im2gps data set. We ensure that the test set is roughly geographically uniform, spanning all continents, and that indoor, landscape, and urban scene types are evenly represented. Because human studies are time consuming and expensive, we did not want to waste time with images that would be impossible to geolocate. Such images could also frustrate the participants in our study. Therefore we remove images which are deemed to have very little geographic information. But as you can see in Figure 7.1, the test set is still very challenging.

The interface that participants used to geolocate images is shown in Figure 7.2. The interface design hinges on the sticky, somewhat philosophical, issue of "What does it mean to know where a photo was taken?". For example, one could argue that knowing the *name* of the location is enough (e.g. "Oh, that's the Forbidden Palace in *Beijing*"). But what if the participant has no idea where Beijing is? One could instead argue that a participant needs to point to a specific location on a map without relying on labels. In the extreme, one could say that any map is providing assistance and the human task should be the exact same as the computational task – provide GPS coordinates. In the end, we believe that the relevant issues for this study are visual scene understanding and visual memory, not spatial memory, so we provide a fairly detailed map. Participants see a 3d globe, textured with a political map with light pollution and elevation cues. To simplify the interface there is no zooming, which means that city names do not fit on the map and country names are often illegible. When shown a photograph, participants rotate the globe with mouse dragging until a fixed crosshair is over the location where they think the photo was taken. The users then click an on-screen button to finalize their choice. When participants start the study they undergo a supervised training session to ensure that they are comfortable with the interface.



Alaska	Antigua	Argentina	Arizona	Athens	Australia	Bahrain	Bangkok
Barcelona	Beijing	Berlin	Bombay	Botswana	Cairo	Cambodia	Canada
China	Czech Repub.	Dhaka	Egypt	France	Greenland	Hawaii	Hongkong
Indonesia	Iraq	Ireland	Istanbul	Japan	Kenya	London	Maine
Mexico City	Minnesota	Montenegro	Morocco	Netherlands	Nevada	New York City	New Zealand
Nor. Carolina	Norway	Oklahoma	Paris	Pennsylvania	Philippines	Rome	Russia
Saigon	San Francisco	South Africa	Spain	Tennessee	Texas	Thailand	Tokyo
Turkey	Uganda	Venezuela	Florence	Pisa	Scotland	Seattle	Venice.

Figure 7.1: *Human Geolocation Test Set.* How well can you do? The rough location of each photo is indicated by the corresponding table of geographic place names. The test set was filtered by hand such that all images had some hint of geographic information, although many photos are still quite ambiguous.


(10 of 64) Where was this photograph taken?

This is my guess

Figure 7.2: *Human Geolocation Interface.* Photographs are shown with the maximum dimension resized to 650 pixels. The globe is rotated by dragging with the mouse.

After each geolocation guess, participants are presented with the actual location of the photo and the amount of error in their estimate. If their error is less than 400km they get a congratulatory message. We believe this feedback is very important – it makes the study seem like a game and increases participant attention. After finishing the study many participants volunteered that it was fun and intellectually challenging. In fact, Google's Picasa photo sharing service has an analogous game on their web site.

In order to gain deeper insight into what strategies and cues participants are using to geolocate images, the study consists of two separate photo viewing conditions – ''long" or "100ms". In the 'long' condition, participants have unlimited time to view the photograph. In the "100ms" condition, there is a three second countdown, then the photograph is flashed for only 100 milliseconds followed by a gray blank (See Figure 7.3). In both conditions

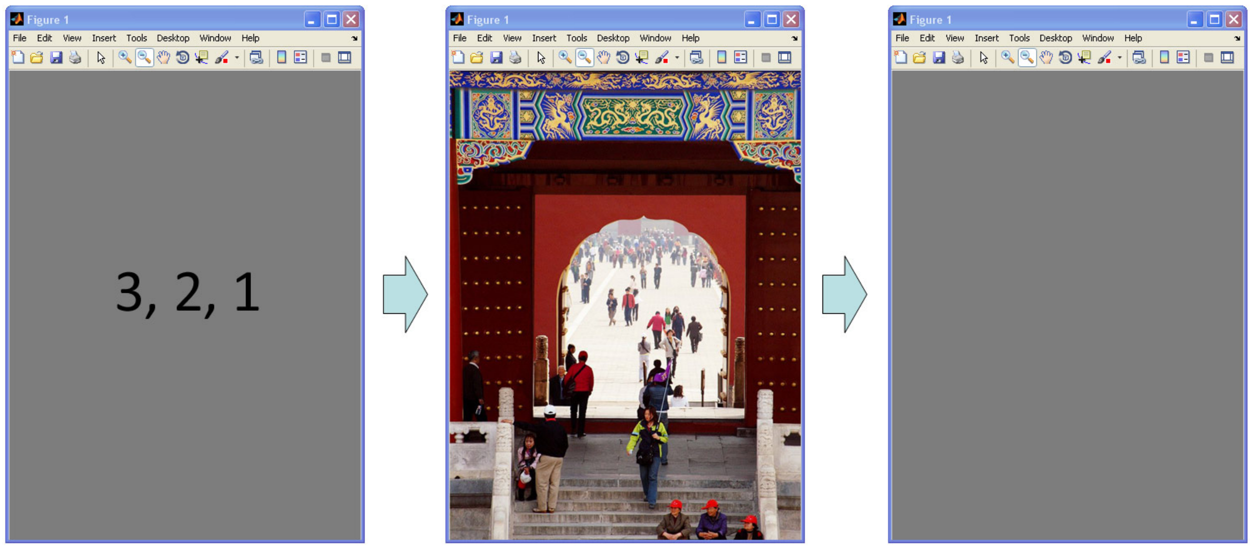


Figure 1
File Edit View Insert Tools Desktop Window Help
3, 2, 1
Figure 1
File Edit View Insert Tools Desktop Window Help
Figure 1
File Edit View Insert Tools Desktop Window Help

Figure 7.3: *"100ms" Viewing Condition.* The last 32 images for each participant are flashed for only 100ms after a countdown. This is long enough to get a very good "gist" of the scene, but not long enough to fixate on any details.

participants have unlimited time to make their decision and use the interface. The short viewing condition is inspired by numerous psychophysics and neuroscience studies showing that humans gain some degree of scene understanding very rapidly and under very short viewing conditions. For example, Thorpe's well known animal vs non-animal study showed that with a 20ms presentation, participants were 94% accurate at saying whether or not a photo contained an animal [99]. With a presentation time of 100ms it is not possible for a viewer to perform a saccade and fixate anywhere in the image beyond the initial focal point (which in our case will be the center of the image where the countdown is displayed). Therefore this viewing condition is interesting because it strongly inhibits that ability of participants to scrutinize an image for subtle cues such a text, car manufacturers, clothing styles, etc. We claim that it reduces the geolocation task to a more memory-based decision dependent on scene statistics. This would be more analogous with our computational methods, which cannot understand scenes as well as humans and instead rely on scene statistics and a huge visual memory.

Participants see the first 32 photos under the 'long' viewing condition and the last 32 under the '100ms' viewing condition. We wanted to make sure participants were comfortable with the study before giving them the more challenging viewing condition. The order of the presentation is randomized, but we ensure that every photo is seen an equal number of times in each viewing condition.

7.3 Results

A total of 20 participants completed the study. No learning or fatigue effects were observed with the viewing condition blocks. After the main portion of the study the participants completed an 11 question computer survey. Question topics included age, sex, level of education, amount of domestic and international travel, number of languages spoken, amount of nature or travel television watched, and photography experience. None of the responses were strongly correlated with geolocation performance, although travel experience was weakly positively correlated.

The geolocation accuracy of each participant, sorted by performance, is shown in Figure 7.4. A geolocation is considered correct if it is within 400km of the ground truth location, which is a larger threshold than we have used previously. This increased tolerance is to account for some users being imprecise with the interface. There is a large variance in performance between participants. This is not too surprising – the geolocation task is very dependent on prior visual memory, and if a participant unfamiliar with the geographic cues (architecture, vegetation, etc.) then it is hard to narrow down the most likely geolocation.

The per-image performance of the base im2gps system described in Chapter 4 is correlated with human performance (r = .51). In general, the images that are computationally difficult are also difficult for humans. Exceptions are highlighted in Figures 7.5 and 7.6 which show cases where humans are better and worse, respectively, than the baseline im2gps method. The new features and lazy learning (Chapter 6) increase performance dramatically – The size of the "memory" of the computational method hasn't changed, but its ability to "understand" scenes has.

In Figure 7.7, performance is examined with increasingly large spatial thresholds for geolocation accuracy. In this figure the human accuracies for different viewing conditions are plotted separately. As expected, longer viewing durations increase geolocation accuracy. But the effect is *surprisingly small* (and not statistically significant at the 400km threshold). This is one of the most interesting results of the study. It suggests that participants weren't so much "playing detective" with each image to decipher subtle geographic cues, but instead performance was heavily memory dependent and they either knew it or they didn't. This is not to suggest that more careful examination of the images could not provide useful information. The participants only spent an average of 7 additional seconds per image on the "long" viewing condition photographs. They might have simply been lazy, although we hoped the game-like design of the study encouraged them to sincerely try.

In Figure 7.8, accuracy is broken down between landmark, landscape, and urban scene types. There were a few images in the test set that did not fit into these categories (indoor and

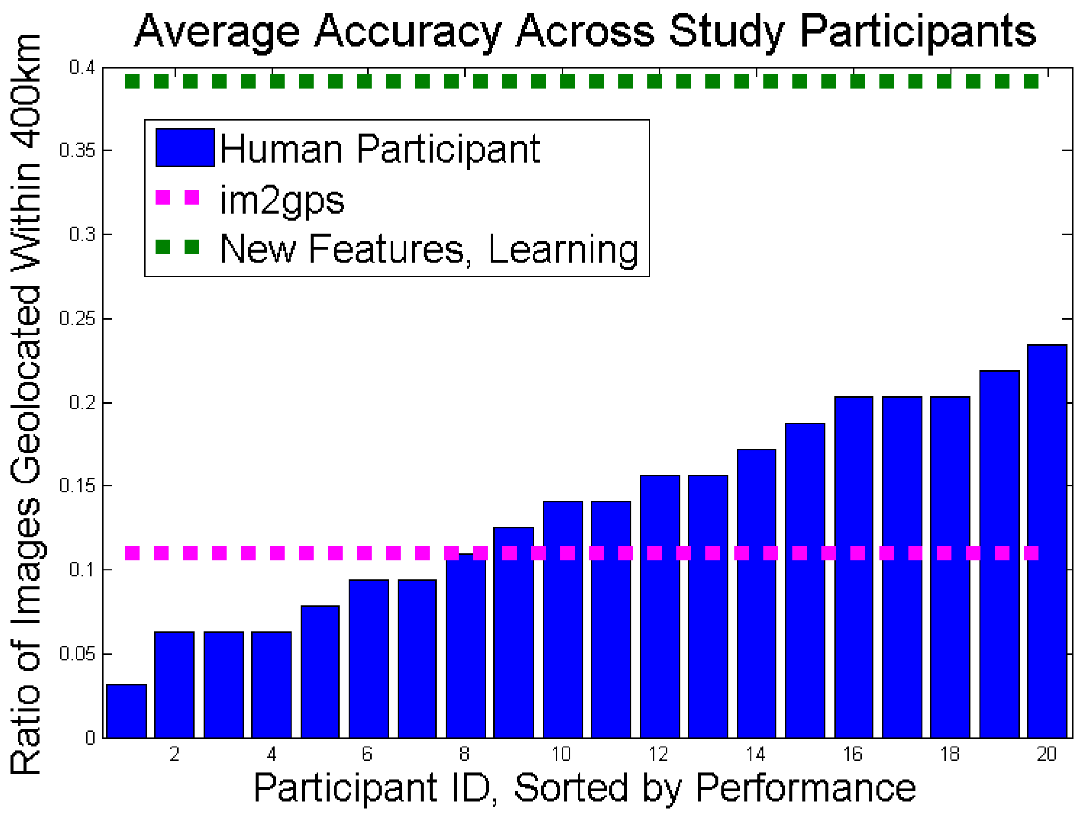


Figure 7.4: *Human Geolocation Performance, by Participant.* There is a large variance in performance among the participants. On average, participants do slightly better than im2gps, but far worse than geolocation with new features and learning.

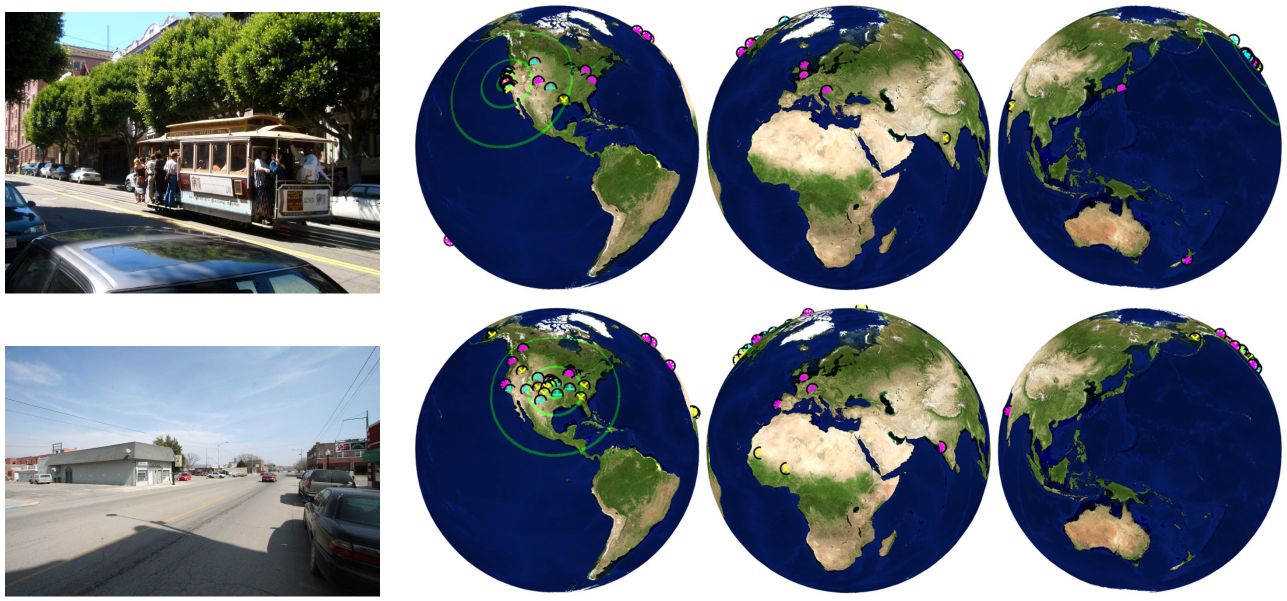


Figure 7.5: *Examples Where Human Geolocation is Better than im2gps.* Magenta markers are im2gps nearest neighbors. Yellow markers are human geolocations under '100ms' viewing conditions. Cyan markers are human geolocations under 'long' viewing conditions. While im2gps fails to recognize this San Francisco landmark, the new features and learning do not. Humans, especially under 'long' viewing condition, are exceptionally good at recognizing a generic Midwest U.S. scene.

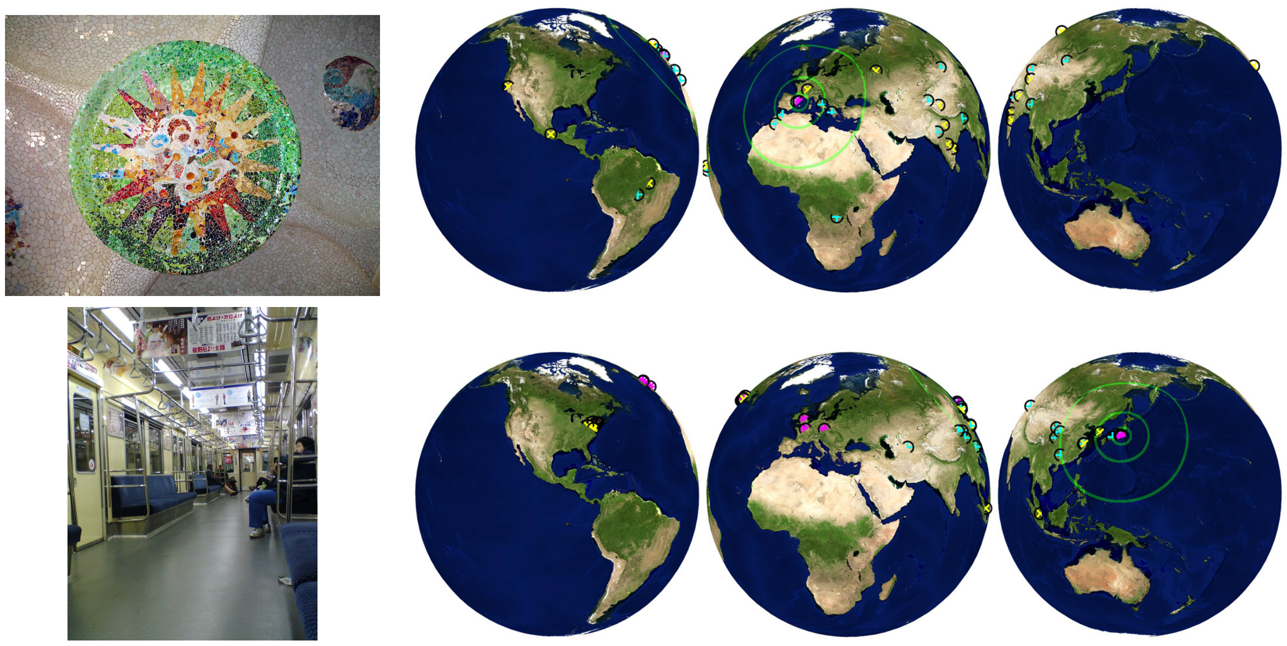


Figure 7.6: *Examples Where Human Geolocation is Worse than im2gps.* Magenta markers are im2gps nearest neighbors. Yellow markers are human geolocations under '100ms' viewing conditions. Cyan markers are human geolocations under 'long' viewing conditions. The Gaudi mosaic is often photographed and visually distinct, making it easy for the computational methods. The Tokyo subway was confused with other metropolitan areas under the '100ms' viewing condition. However, under 'long' viewing, Oriental characters are noticeable and all of the guesses are in Asia (although many times still wrong).

object photos). For all methods, the relative difficulty of these categories is the same – landmarks are easiest, then urban, then landscape. Landscape images are especially vexing for the humans and im2gps method, although still well above chance (which would be near 1%). The dramatic gain on landmark images for the new features and learning is not entirely surprising – bag of SIFT features have been shown to do well at instance level recognition. The large gains in the the other types of scenes show that the increase in performance goes beyond landmark recognition, though. See Chapter 6 for more examples of geolocation results.

We hypothesized that the short viewing condition might cause the human geolocation performance to be more correlated with the computational performance, because the statistics a human gathers under short viewing conditions more closely resembles the features used by our computational method. This was not the case, though. The correlation of im2gps performance to human performance was similar for both viewing conditions.

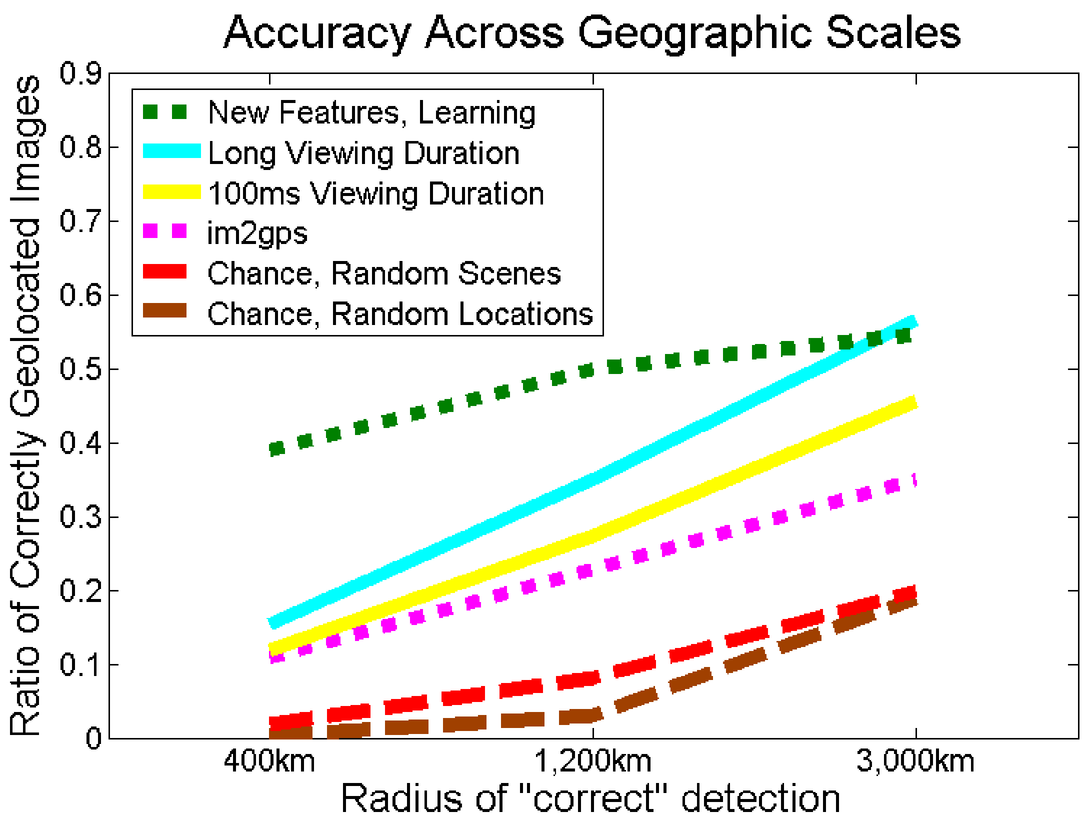


Figure 7.7: *Human Geolocation Performance vs Scale.* Chance is calculated in two ways. 1) Random Scenes. This is chance if geolocation estimates are the locations of random scenes in the im2gps database. 2) Random Locations. This is chance if the geolocation estimates are random locations on land.

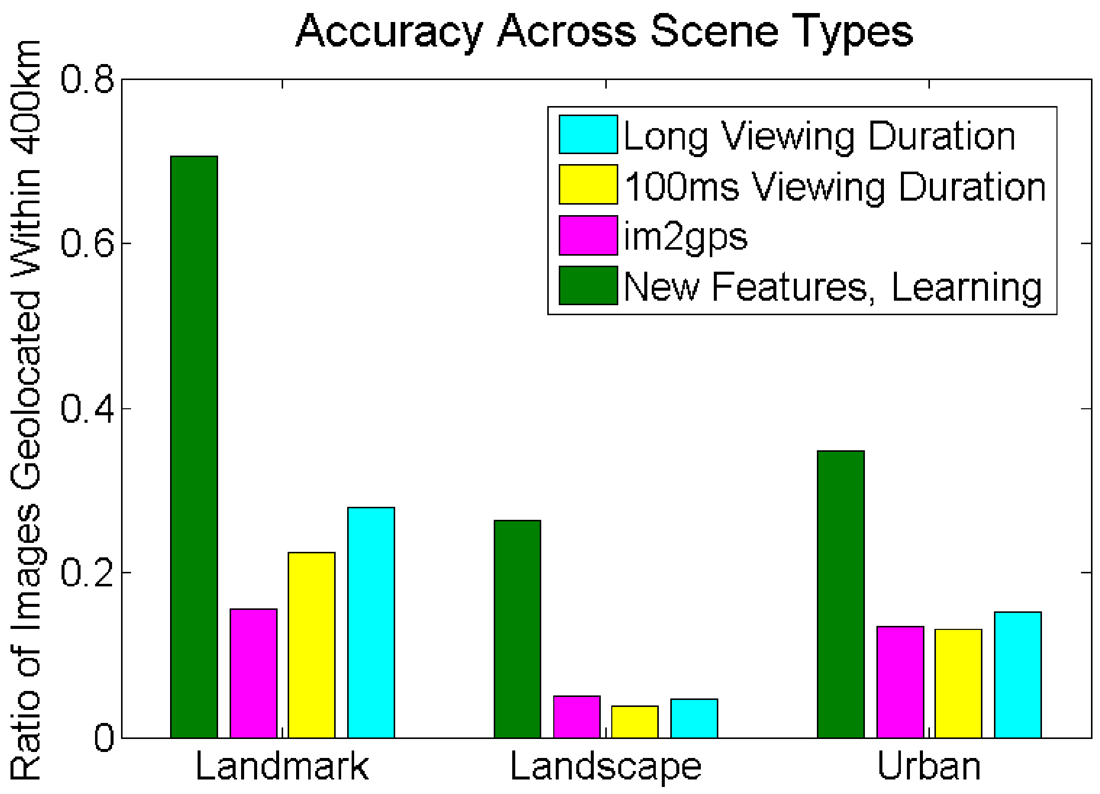


Figure 7.8: *Human Geolocation Performance, by Scene Type.*

Chapter 8

Conclusion

In this thesis, we showed that scene matching can effectively leverage Internet scale image collections with keyword, temporal, and GPS metadata for numerous graphics and vision tasks. We used scene matching to advance the state-of-the-art in image completion and object detection and to enable previously unapproachable types of image understanding such as global image geolocation. We showed that lazy learning together with a rich set of image features can dramatically improve scene matching. Overall, this thesis suggests that the quantity of imagery on the Internet is large enough to push back against the complexity of our visual world and make computer vision and computer graphics easier.

However, while this thesis concretely demonstrates the effectiveness of scene matching for many tasks, it does not rigorously explain *why* scene matching works at all. Unfortunately, there is no simple answer to this question. Below, we speculate about many of the relevant factors. In the final section we map out potentially interesting future research directions related to this thesis.

8.1 Why Does Scene Matching Work?

The intrinsic dimensionality of a scene according to our definition (Section 1.1) is quite high. Because of this complexity, several million images can not provide a dense sampling of the space of scenes. Instead, all of the applications in this thesis rely critically on the space of *likely* scenes being much smaller than the space of *possible* scenes. This dependency means that the performance of our algorithms is not just a function of our visual world, but a function of our *visual experience* as influenced by numerous factors, many of them social in nature. We work exclusively with photographs captured by humans which by itself provides a strong bias because of the physical constraints imposed by the real world. But imagine a mobile robot that captured random photographs of the world – this would be a tremendously different space of scenes than the one we've sampled from Flickr. The multitude of factors that influence whether or not someone takes a picture and uploads it to Flickr all change the distribution of scenes that we observe. Some of these factors are social (People tend to photograph other people), cultural (Is it acceptable to capture and share photographs of this situation?), technological (Are cameras and Internet connections widely available in this region?), aesthetic (Is this subject matter or subject matter interesting and attractive? Is this photograph reasonably composed and oriented?), etc. Most of these factors tend to narrow the range of observed scenes and make our tasks easier. The aesthetic factors, though, can sometimes push in the other direction by compelling people to capture a typical experience in a very unusual manner. This all means that our large photo collections are measuring not just the real, visual world but also the human factors involved in photo capturing and sharing. This is potentially an advantage – we can use our data to learn about humans, as we did in Image Sequence Geolocation (Section 4.6) where we built a prior on human travel from Flickr data. But it can also be a hindrance – we might not expect our algorithms to generalize to random robot imaging.

Beyond these issues related to sampling the space of scenes, one could also ask if scenes are somehow a privileged unit of representation in our visual world. It it easier to reason about scenes than objects, geometries, poses, and other visual phenomena? This is a difficult question to rigorously answer, but it does seem that scenes have fewer ambiguities and polysemes[1] than these other phenomena, at least when imaged by human photographers. With a large enough database and simple image features, it is rarely the case that a nearest neighbor scene will be semantically unrelated to a query photograph. It might simply be a property of our visual world that scenes are easy to reason about, which provides further motivation for a vision pipeline in which scene reasoning precedes and assists more difficult difficult cases of visual reasoning, as is the case with human vision (Section 2.2).

[1]In the computer vision literature, polysemy refers to the phenomena of the same visual signal having multiple meanings. This differs from the linguistic definition of polysemy, in which a word has different *related* meanings, often branching from the same etymological origin. In computer vision, there is no expectation that the ambiguities are semantically related.

8.2 Future Areas of Investigation

Since the publication of the first parts of this thesis, several other authors have published complementary research (Section 2.1.4) and there is undoubtedly a rich set of future work concerning the efficient use of large scale data for graphics and vision problems.

For many applications, the “limiting reactant” for large scale scene matching is not the quantity or quality of image data, but the quantity and quality of image *annotation*. With better annotation, large scale imagery can be utilized to tackle core computer vision problems such as scene parsing, object recognition, single view geometry, and pose estimation. There are numerous potential source for annotation – relying on the social motivations of photographers (this thesis), making a game out of image labelling [3], offering 3d visualizations after labelling [85], and, of course, paying people [91]. There is no clear winner yet among these possibilities – each offers advantages and disadvantages related to quality, scale, versatility, and cost. Further investigation is warranted.

This thesis has been exclusively concerned with images, but analogous methods might be effective in the video domain. Videos are higher dimensional than images, but like scenes the space of likely videos is a very small manifold of the space of possible videos. The general trend of using ever-larger data sets, which this thesis continues, will undoubtedly continue in all relevant computer vision and graphics research domains.

Appendix A

How Heavy-Tailed is Our Visual World?

The frequency of scenes and objects is heavy-tailed and thus brute-force approaches are infeasible. – David Forsyth, Jitendra Malik (paraphrased)

The frequency of scenes and objects follows Zipf's Law and makes im2gps performance with respect to database size questionable. – Antonio Torralba (paraphrased)

Within the computer vision community there is something of a philosophical debate concerning the recent emergence of massively data driven scene understanding methods such as those described in this thesis. Our visual world is massively complex and varied. The expectation of data-driven methods – that you've seen it all before – cannot be realistic or workable, one could argue. How diverse is our visual experience, and how hard is it to sample exhaustively? In this appendix we touch on these fundamentally difficult questions.

We begin with a discussion of "Zipf's law" which states that the frequency of a word in any natural language corpus is inversely proportional to its rank when sorted by decreasing frequency ($P_n \propto 1/n^a$ where P_n is the probability of the n^{th} most likely item and a is a constant particular to the data set). This implies that the distribution follows a the power law and is heavy-tailed. When category frequency is plotted against category rank in log-log space, such a "Zipfian" distribution will appear linear. Figure A.1 shows a linguistic example – the distribution of words on the English version of Wikipedia in November 2006.

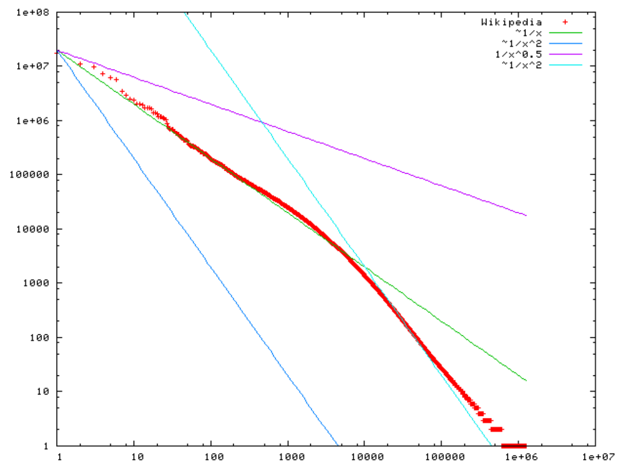


Figure A.1: *Word frequency vs rank on Wikipedia.* On the x-axis is the rank of each word, on the y-axis is the frequency. Both axes are in the log domain. The most frequent words are “the”, “of”, and “and”. The distribution is not perfectly “Zipfian” because the slope is not constant. Credit to Victor Grishchenko, licensed under LGPL.

It is claimed that Zipf’s law extends to many other real-world, categorical data distributions, including visual data such as object and scene frequencies.

If a set of data follows Zipf’s law (or any other heavy-tailed distribution) it is generally thought to be discouraging to data driven approaches because there is a long tail of categories that are not represented frequently enough to sample densely. For example, in Figure A.1, 84% of words appear less than 10 times. If a data driven method could not function with less than 10 samples of each category (for instance visual examples of objects or scene types, or motion capture examples of actions), then this specific distribution would imply that the algorithm fails on 84% of (thus far) observed categories. On the other hand, if the test set frequencies are the same as in the training set, 99.1% of the test cases are from categories that are sampled densely enough. The important thing for this thesis is the question of whether or not our data distribution follows Zipf’s law and is heavy-tailed.

A.1 Measuring the “heavy-tail”.

Scenes that are too rare for us to sample densely yet still show up in test data are likely to cause our algorithms to fail. The work in this thesis relies on sampling the space of scenes densely with respect to *geographic location*, *visual appearance*, and *object presence*.

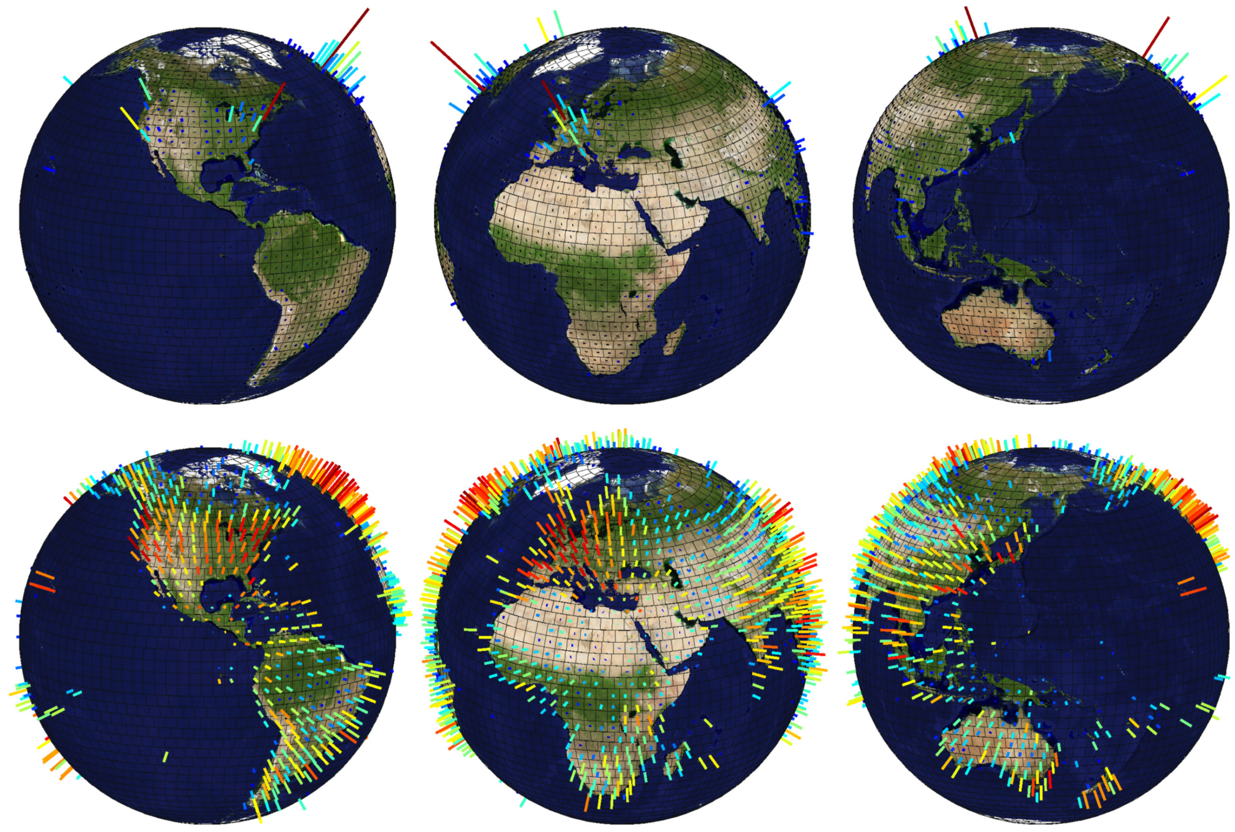


Figure A.2: *Photo density in im2gps data set*, linear scale (top) and natural log scale (bottom). The height of each bar is proportional to the density of geotagged photos in each equal area region. The bars are colored according to the Matlab "jet" color scheme. Regions with zero photos have no bar.

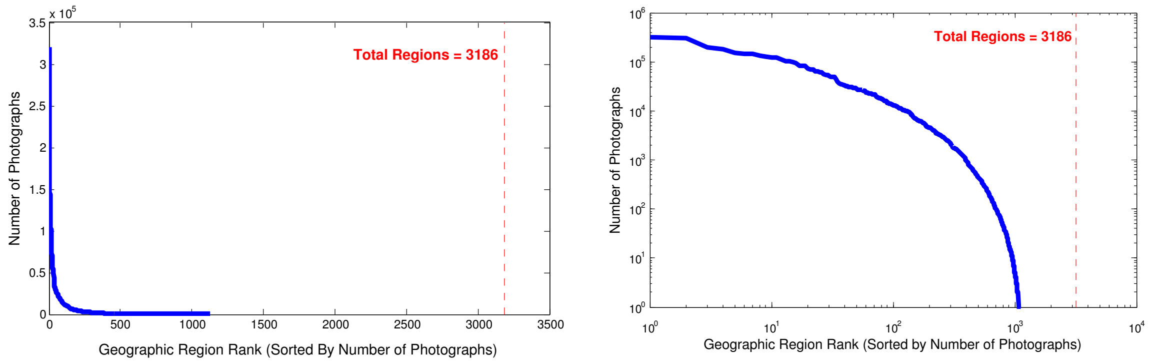


Figure A.3: *Photograph density versus rank in the im2gps database.* The geographic distribution is appears heavy-tailed (left), but in the log-log plot (right) we can see that the distribution is less heavy-tailed than Zipf's law would predict.

Geographic Distribution of Photos. Im2gps benefits from a high density of photographs at all locations across the Earth. Section 4.5.1 shows that im2gps performs very poorly when asked to geolocate images from undersampled regions. Figure A.2 visualizes the density of our Flickr photographs across the world. For this analysis, the Earth's surface is tesselated with 3,186 roughly equal area quadrilaterals which are $\sim$ 400km on edge. While some regions are much more heavily sampled, the natural log scale visualization shows that very few regions are completely devoid of photographs (mostly the Arctic, Antarctic and parts of Siberia and the Sahara desert). Unsurprisingly, there are almost no photographs in the open oceans. The geographic categorization that we have created with this tessellation is arguably not semantically meaningful – Boston and New York City have fallen into a single region while the Taiga of Siberia spans fifty. But the evaluation used in im2gps does not take the semantics of geographic areas into account either. It uses the same radius (often 200km, thus the 400km quads) to evaluate geolocation estimates anywhere on the globe.

Figure A.3 plots the distribution of geographic region photo densities in the same frequency vs rank log-log format used for Figure A.1. The distribution does not appear to follow Zipf's law – it is saturating near approximately 1,000 categories (a bit less than the number of regions containing land). As we collect more data, we expect regions to reach sufficient density faster than new categories emerge. Furthermore, there is a clear upper bound on how many categories *can* emerge unlike, for instance, language.

Visual Distribution of Scenes. The geographic distribution of photographs is amenable to data driven methods, but can we say anything about the visual distribution of scenes? If we trust that the distance our scene matching algorithm attributes to a pair of scenes is semantically meaningful, we could say something about the density of similar scenes for

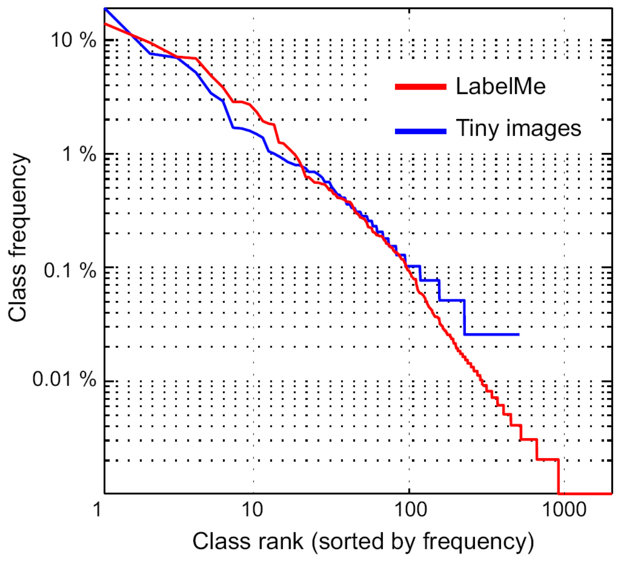


Figure A.4: *Object category frequency versus rank in "Tiny Images" and "LabelMe".* On the x-axis is the rank of each object category, on the y-axis is the frequency. Both axes are in the log domain. Credit to Antonio Torralba. The most popular object descriptions in LabelMe as of August 26th, 2008 are "person walking" (28718), "window" (22789), "tree" (10023), "head" (8548), "building" (8309), "car" (7895), "sky" (6238), "leg" (5716), "arm" (4764), "trees" (4563), "sidewalk" (4211), "wall" (4189), "road" (3940), "torso" (3095), "sign" (3064), "car occluded" (2539), "door" (2441), "chair" (2179), "needle crystal" (2063), and "person" (1943).

a large sampling of scenes and thus get some measure of how heavy-tailed the distribution of scenes is without having to define categories. An example result might be of the form "10% of scenes have no neighbors within some distance threshold", but no single threshold or kernel will produce semantically meaningful results. Whether or not two scenes are visually "similar enough" is task specific.

Thus one of the attractions of scene matching applications like im2gps is that they provide a quantitative way to evaluate large scale scene matching. Plotting the performance of geolocation against database size shows an encouraging rate of growth (Figure 4.4), but there are many other phenomena influencing the performance of our geolocation algorithm such as the dependence on geographic sampling density discussed above, the presence of landmarks, and the self-similarity of the earth (thus perfect scene matches with trillions of reference photos could still fail the geolocation task). These factors might explain why Figure 4.4 shows performance growth faster than large scale data driven tasks analyzed in [100].

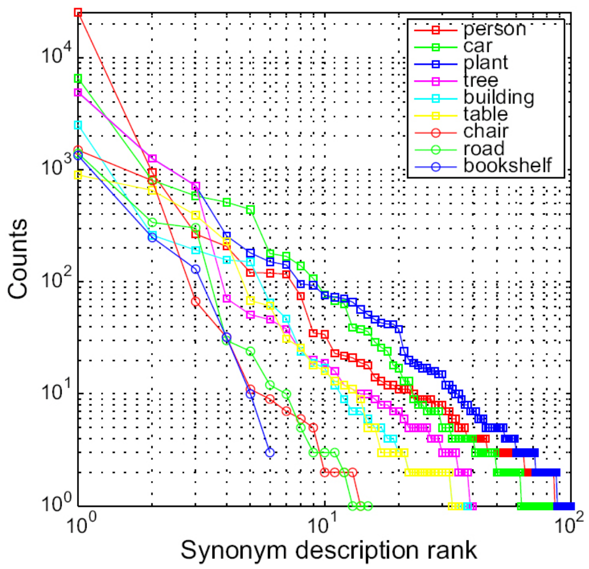


Figure A.5: *Description frequency versus rank for groups of synonyms in "LabelMe"*. On the x-axis is the rank of each object description, on the y-axis is the frequency. Both axes are in the log domain. For each visual concept, there seems to be a Zipf-like distribution due to the English language. Credit to Bryan Russell.

Distribution of Objects. The number [7] and frequency [84,93,100] of visual categories in our world has been studied in the psychology and computer vision literature. [7] estimates from linguistics, child development, and dictionary counting that there are less than 1,500 basic level object categories on the level of "penguin", "sparrow", "lamp", "cup", and "typewriter". From this, Biederman arrives at an upper bound of 30,000 *visual object categories* by doubling 1,500 and hypothesizing that there are an average of 10 visual exemplars for each object category (e.g. different types of lamps, cups, or penguins). [100] analyzes object name distributions in the "Tiny Images" and "LabelMe" databases (Figure A.4). Both distributions appear to be heavy-tailed and follow Zipf's law.

Of the 20 most popular LabelMe descriptions (Figure A.4), note that "person walking" and "person", as well as "car" and "car occluded" are separate categories. It may be reasonable to break up these categories (for instance if they differ enough visually to require separate detectors). However, "tree" and "trees" being two different categories points to different labelling strategies more than different visual categories. In fact, LabelMe annotators frequently use different vocabulary to describe the same visual phenomena. Figure A.5 shows that several popular object categories have up to 100 synonyms, and that these synonyms themselves follow a Zipf-like distribution [84].

Examining the descriptions that make up the heavy-tail of LabelMe annotations (5390 of the 9259 descriptions are used only once) is informative. Out of 100 random, once-used descriptions[1] there are many typos ("bacony", "windshiel", and "electical cord"), compound descriptions ("goat left near forest", "counter/bench top", "bowl of vegetables"), and unusual phrases ("identifying parking", "cute gal", "freeze"). Only about 30% of the descriptions indicate a clear, isolated visual object category. Of these, about half are synonyms or more specific versions of more commonly used descriptions – "sewer lid" appears once, but "manhole" appears 223 times and "manhole cover" appears 55 times. "bar of soap" appears once, but "soap" appears 44 times. It's not clear that these are actually new visual categories. Perhaps 15% of the descriptions in the heavy-tail are unambiguously new object categories with no more frequently used synonyms, for instance "leash", "dam", "money", and "router". It may be that the overall object distribution appears to follow a power law because of noise in the annotation process and the use of the heavy-tailed English language. Maybe that the visual world is not so heavily tailed.

The issues of annotation noise and English language synonyms are addressed in [92]. The authors analyze the visual variation of our world using image descriptions collected with their "WhatWhere" game. In the "What" portion of the game, players are shown an image and given points for every object they can name. A bonus is given for rarely used object names such as "pupil" and "tine". In the "Where" portion of the game, users given 5 objects found by other users and asked to locate them in an image as a form of verification. The authors correct misspellings by hand and use WordNet [33] to collapse all synonyms into the same category. They observe that object category frequency follows a power law. However, their data set is less extensive than LabelMe, covering only 250 images and totalling less than 1000 objects, so it is hard to extrapolate and reason about the overall number and frequency of object categories.

[1] "asphalt (road)", "carback occluded", "tea boxes", "perso standing", "park sign", "prjection screen", "newpaper stand", "antenna tower", "carwindow", "cdromdrive", "calander", "screen az150deg", "mountainside", "line persons", "dam", "skly", "freeze", "bacony", "motobikes side", "light, street light, lamp", "money", "windshiel", "sec house", "boat with rowers", "persons sitting", "coast line", "chopping board", "bar of soap", "seashell", "toilette", "tv screen", "plate on stand", "toy horse", "butter dish", "mixing bowl", "dish with lobsters", "bowl of vegetables", "crackers", "cas", "vegtables", "bull skull", "foosball game", "pato", "lion right mid water", "goat left near forest", "bird front mid water", "rear fender", "person_walking_back_occluded", "face (side view)", "art object", "airport buildings", "roofs", "stone steps", "field, grass", "piece of wood", "whiskers right", "cheetah cub", "wood carving", "arbor", "spoon occluded", "cctv, camera", "plastic bottle", "vehicle occluded", "walking lane", "pedestrian person walking crop", "samuel adams statue", "cute gal", "identifying parking", "electical cord", "street sign back", "fishing boat", "sewer lid", "christmas wreath", "silver colored hub cab-car", " car frontal", "playboy sign", "card display", "person walking crop az90deg", "290", "overhad projector arm", "leash", "chaild walking", "truck cropped", "rock2", "isa", "sunset sky", "ancient building", "glass window", "pedestrians (grouping)", "drawers empty", "piano crop", "table salt", "counter/bench top", "vanity top", "people art decor", "router", "french press", "mop handle", "sink unit", "chair arm"

Just as there are rare objects in our world there are undoubtedly rare scenes. Since there is no clear visual categorization of scenes[2] it is difficult to analyze their distribution in the same way that object distributions were analyzed above. It is unclear how strongly correlated the distribution of objects is to the distribution of scenes. Some rare objects (e.g. "metal detector", "x-ray machine") belong to or define rare scenes (e.g. "security checkpoint"). On the other hand, some rarely photographed objects (e.g. "leash", "router", and "money" from above) occur in very common settings like homes and offices. Also, common objects could be arranged in such as way as to constitute a rare scene ("cars" and "dirt" into "demolition derby"). The frequency distributions of visual objects and scenes are still open questions.

The distribution of objects and scenes in our visual world might turn out to be irrelevant, in practice, because the Internet provides us with an ever growing avalanche of data. For instance, "Leash" might be a rare object category, and "Demolish Derby" might be a rare event, but there are 13,000 and 17,000 examples on Flickr, respectively. Undoubtedly, some objects are not well sampled. But if our ultimate goal is to give machines the type of qualitative understanding of images that humans enjoy, the fact that we can find relatively few annotated visual examples of a "maniple" is not likely to be a showstopper.[3]

[2]Visual scenes continuously transition from one category to another ("beach" to "lake", "street" to "city", etc...) or occupy multiple categories ("forest" and "mountain" or "kitchen" and "bedroom" in the case of Parisian apartments).

[3]Maniple, noun: a liturgical vestment draped over the left arm. Flickr examples: ~ 20.

Bibliography

[1] Aseem Agarwala, Mira Dontcheva, Maneesh Agrawala, Steven Drucker, Alex Colburn, Brian Curless, David Salesin, and Michael Cohen. Interactive digital photomontage. *ACM Trans. Graph.*, 23(3):294–302, 2004.

[2] Amit Agrawal, Ramesh Raskar, and Rama Chellappa. What is the range of surface reconstructions from a gradient field? In *ECCV*, 2006.

[3] Luis Von Ahn. Games with a purpose. *IEEE Computer Magazine*, pages 96–98, June 2006.

[4] M. Bar. Visual objects in context. *Nature Neuroscience Reviews*, 5:617–629, 2004.

[5] Moshe Bar. The proactive brain: using analogies and associations to generate predictions. *Trends in Cognitive Sciences*, 11(7), 2007.

[6] Tamara Lee Berg and David Forsyth. Automatic ranking of iconic images. Technical Report UCB/EECS-2007-13, EECS Department, University of California, Berkeley, Jan 2007.

[7] I. Biederman. Recognition by components: a theory of human image interpretation. *Pyschological review*, 94:115–147, 1987.

[8] A. Bosch, A. Zisserman, and X. Munoz. Scene classification using a hybrid generative/discriminative approach. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30(4), 2008.

[9] Yuri Boykov, Olga Veksler, and Ramin Zabih. Fast approximate energy minimization via graph cuts. *IEEE Trans. Pattern Anal. Mach. Intell.*, 23(11):1222–1239, 2001.

[10] T.F. Brady, T. Konkle, G.A. Alvarez, and A. Oliva. Visual long-term memory has a massive storage capacity for object details. *Proceedings of the National Academy of Sciences (in press)*, 2008.

[11] Peter J. Burt and Edward H. Adelson. The laplacian pyramid as a compact image code. *IEEE Transactions on Communications*, COM-31,4:532–540, 1983.

[12] O. Chapelle. Training a support vector machine in the primal. *Neural Computation*, 19(5):1155–1178, 2007.

[13] Stefan Schaal Chris Atkeson, Andrew Moore. Locally weighted learning. *AI Review*, 11:11–73, April 1997.

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

[15] Dorin Comaniciu and Peter Meer. Mean shift: A robust approach toward feature space analysis. *IEEE Trans. Pattern Anal. Mach. Intell.*, 24(5):603–619, 2002.

[16] Robert G. Cook, Deborah G. Levison, Sarah R. Gillett, and Aaron P. Blaisdell. Capacity and limits of associative memory in pigeons. *Psychonomic Bulletin and Review*, 12(2):350–358, April 2005.

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

[18] A. Criminisi, P. Perez, and K. Toyama. Object removal by exemplar-based inpainting. *CVPR*, 02:721, 2003.

[19] Navneet Dalal and Bill Triggs. Histograms of oriented gradients for human detection. In *CVPR*, 2005.

[20] Kevin Dale, Micah K. Johnson, Kalyan Sunkavalli, Wojciech Matusik, and Hanspeter Pfister. Image restoration using online photo collections. In *International Conference on Computer Vision*, 2009.

[21] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2009.

[22] Nicholas Diakopoulos, Irfan Essa, and Ramesh Jain. Content based image synthesis. In *Conference on Image and Video Retrieval (CIVR)*, 2004.

[23] S. Divvala, A.A. Efros, , and M. Hebert. Can similar scenes help surface layout estimation? In *First IEEE Workshop on Internet Vision at CVPR*, 2008.

[24] Santosh K. Divvala, Derek Hoiem, James Hays, Alexei Efros, and Martial Hebert. An empirical study of context in object detection. In *CVPR*, 2009.

[25] Carlotta Domeniconi and Dimitrios Gunopulos. Adaptive nearest neighbor classification using support vector machines. In *NIPS*, 2001.

[26] Iddo Drori, Daniel Cohen-Or, and Hezy Yeshurun. Fragment-based image completion. *ACM Trans. Graph.*, 22(3):303–312, 2003.

[27] Alexei A. Efros and William T. Freeman. Image quilting for texture synthesis and transfer. *Proceedings of SIGGRAPH 2001*, pages 341–346, August 2001.

[28] Alexei A. Efros and Thomas K. Leung. Texture synthesis by non-parametric sampling. In *ICCV*, pages 1033–1038, Corfu, Greece, September 1999.

[29] R.A. Epstein. The cortical basis of visual scene processing. *Psychological Science*, 12:954–978, 2005.

[30] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The PASCAL Visual Object Classes Challenge 2007 (VOC2007) Results. http://www.pascal-network.org/challenges/VOC/voc2007/workshop/index.html, 2007.

[31] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The PASCAL Visual Object Classes Challenge 2008 (VOC2008) Results. http://www.pascal-network.org/challenges/VOC/voc2008/workshop/index.html, 2008.

[32] Li Fei Fei and Pietro Perona. A bayesian hierarchical model for learning natural scene categories. In *CVPR 2005*, pages II: 524–531, 2005.

[33] C. Fellbaum. *Wordnet: An Electronic Lexical Database.* Bradford Books., 1998.

[34] Pedro Felzenszwalb, David McAllester, and Deva Ramanan. A discriminatively trained, multiscale, deformable part model. In *CVPR*, 2008.

[35] J. H. Friedman. Flexible metric nearest neighbor classification. Technical report, Stanford, Nov. 1994.

[36] A. Frome and J. Malik. Image retrieval and recognition using local distance functions. In *NIPS*, 2006.

[37] T. Hastie and R Tibshirani. Discriminant adaptive nearest neighbor classification. *IEEE PAMI*, 18:607–616, 1996.

[38] James Hays and Alexei A Efros. Scene completion using millions of photographs. *ACM Transactions on Graphics (SIGGRAPH 2007)*, 26(3), 2007.

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

[40] D. Hoiem, A.A. Efros, and M. Hebert. Recovering surface layout from an image. *Int. J. Comput. Vision.*, 75(1), 2007.

[41] D. Hoiem, A.A. Efros, and M. Hebert. Putting objects in perspective. *Int. J. Comput. Vision. (In Press)*, 2008.

[42] M. Irani, P. Anandan, and S. Hsu. Mosaic based representations of video sequences and their applications. In *ICCV*, 1995.

[43] Nathan Jacobs, Scott Satkin, Nathaniel Roman, Richard Speyer, and Robert Pless. Geolocating static cameras. In *Proc. ICCV*, 2007.

[44] P. Jain, B. Kulis, and K. Grauman. Fast image search for learned metrics. *CVPR*, June 2008.

[45] Hervé Jégou, Matthijs Douze, and Cordelia Schmid. Hamming embedding and weak geometric consistency for large scale image search. In *European Conference on Computer Vision*, volume I, pages 304–317, oct 2008.

[46] Jiaya Jia, Jian Sun, Chi-Keung Tang, and Heung-Yeung Shum. Drag-and-drop pasting. *ACM Trans. Graph.*, 2006.

[47] Thorsten Joachims. Transductive inference for text classification using support vector machines. In *International Conference on Machine Learning (ICML)*, 1999.

[48] M. K. Johnson, K. Dale, S. Avidan, H. Pfister, W. T. Freeman, and W. Matusik. Cg2real: Improving the realism of computer-generated images using a large collection of photographs. Technical Report MIT-CSAIL-TR-2009-034, CSAIL MIT, July 2009.

[49] Matthew Johnson, Gabriel J. Brostow, Jamie Shotton, Ognjen Arandjelović, Vivek Kwatra, and Roberto Cipolla. Semantic photo synthesis. *Computer Graphics Forum (Proc. Eurographics)*, 25(3):407–413, September 2006.

[50] Evangelos Kalogerakis, Olga Vesselova, James Hays, Alexei A. Efros, and Aaron Hertzmann. Image sequence geolocation with human travel priors. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV '09)*, 2009.

[51] D. King. *The Commissar Vanishes.* Henry Holt and Company, 1997.

[52] J. Kivinen, A. J. Smola, and R. C. Williamson. Online learning with kernels. *IEEE Transactions on Signal Processing*, 52(8):2165–2176, 2004.

[53] K. Koh, S.-J. Kim, and S. Boyd. An interior-point method for large-scale l1-regularized logistic regression. In *Journal of Machine Learning Research*, pages 1519–1555, June 2007.

[54] Nikos Komodakis. Image completion using global optimization. In *CVPR*, pages 442–452, 2006.

[55] Jana Kosecka and Wei Zhang. Video compass. In *ECCV '02: Proceedings of the 7th European Conference on Computer Vision-Part IV*, pages 476–490, 2002.

[56] Vivek Kwatra, Irfan Essa, Aaron Bobick, and Nipun Kwatra. Texture optimization for example-based synthesis. In *ACM Trans. Graph.*, pages 795–802, 2005.

[57] Vivek Kwatra, Arno Schodl, Irfan Essa, Greg Turk, and Aaron Bobick. Graphcut textures: Image and video synthesis using graph cuts. *ACM Trans. Graph.*, 22(3):277–286, July 2003.

[58] Jean-François Lalonde, Alexei A. Efros, and Srinivasa G. Narasimhan. Estimating natural illumination from a single outdoor image. In *IEEE International Conference on Computer Vision*, 2009.

[59] Jean-François Lalonde, Derek Hoiem, Alexei A. Efros, Carsten Rother, John Winn, and Antonio Criminisi. Photo clip art. *ACM Transactions on Graphics (SIGGRAPH 2007)*, 26(3), August 2007.

[60] S. Lazebnik, C. Schmid, and J. Ponce. Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. In *CVPR*, 2006.

[61] Michael S. Lew, Nicu Sebe, Chabane Djeraba, and Ramesh Jain. Content-based multimedia information retrieval: State of the art and challenges. *ACM Trans. Multimedia Comput. Commun. Appl.*, 2(1):1–19, 2006.

[62] Li-Jia Li and Li Fei Fei. What, where and who? classifying events by scene and object recognition. In *Proc. ICCV*, 2007.

[63] C. Liu, J. Yuen, , and A. Torralba. Nonparametric scene parsing: label transfer via dense scene alignment. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2009.

[64] C. Liu, J. Yuen, A. Torralba, J. Sivic, and W. T. Freeman. Sift flow: dense correspondence across different scenes. In *European Conference on Computer Vision (ECCV)*, 2008.

[65] D.G. Lowe. Object recognition from local scale-invariant features. *ICCV*, pages 1150–1157 vol.2, 1999.

[66] Tomasz Malisiewicz and Alexei A. Efros. Recognition by association via learning per-exemplar distances. In *CVPR*, 2008.

[67] D. Martin, C. Fowlkes, D. Tal, and J. Malik. A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics. In *Proc. ICCV*, July 2001.

[68] J. Matas, O. Chum, M. Urban, and T. Pajdla. Robust wide-baseline stereo from maximally stable extremal regions. *Image and Vision Computing*, 22(10):761 – 767, 2004.

[69] K. Murphy, A. Torralba, D. Eaton, and W. T. Freeman. Object detection and localization using local and global features. In *Lecture Notes in Computer Science (unrefeered). Sicily workshop on object recognition*, 2005.

[70] D Navon. Forest before trees: The precedence of global features in visual perception. *Cognitive Psychology*, 9:353–383, 1977.

[71] D. Nister and H. Stewenius. Scalable recognition with a vocabulary tree. In *CVPR*, volume 2, pages 2161–2168, 2006.

[72] A. Oliva and A. Torralba. Building the gist of a scene: The role of global image features in recognition. In *Visual Perception, Progress in Brain Research*, volume 155, 2006.

[73] Aude Oliva, Talia Konkle, Timothy Brady, and George Alvarez. The high fidelity of scene representation in visual long-term memory, 2009. Oral Presentation, Vision Science Society (VSS) meeting 2009.

[74] Aude Oliva and Antonio Torralba. Modeling the shape of the scene: A holistic representation of the spatial envelope. *Int. J. Comput. Vision*, 42(3):145–175, 2001.

[75] Aude Oliva and Antonio Torralba. The role of context in object recognition. *Trends in Cognitive Sciences*, 11(12):520–527, November 2007.

[76] Patrick Perez, Michel Gangnet, and Andrew Blake. Poisson image editing. *ACM Trans. Graph.*, 22(3):313–318, 2003.

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

[78] J. Philbin, O. Chum, M. Isard, J. Sivic, and A. Zisserman. Lost in quantization: Improving particular object retrieval in large scale image databases. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2008.

[79] John C. Platt. Sequential minimal optimization: A fast algorithm for training support vector machines, 1998.

[80] Till Quack, Bastian Leibe, and Luc Van Gool. World-scale mining of objects and events from community photo collections. In *CIVR '08: Proceedings of the 2008 international conference on Content-based image and video retrieval*, 2008.

[81] R. Raguram and S. Lazebnik. Computing iconic summaries for general visual concepts. In *First IEEE Workshop on Internet Vision at CVPR*, 2008.

[82] Laura Walker Renninger and Jitendra Malik. When is scene recognition just texture recognition? *Vision Research*, 44:2301–2311, 2004.

[83] Ronald A. Rensink, J. Kevin O'Regan, and James J. Clark. To see or not to see: the need for attention to perceive changes in scenes. *Psychological Science*, 8(5):368–373, 1997.

[84] B. C. Russell, A. Torralba, K. P. Murphy, and W. T. Freeman. Labelme: a database and web-based tool for image annotation. *International Journal of Computer Vision*, 77:157–173, 2008.

[85] Bryan Russell and Antonio Torralba. Building a database of 3d scenes from user annotations. In *CVPR*, 2009.

[86] Bryan C. Russell, Antonio Torralba, Ce Liu, Rob Fergus, and William T. Freeman. Object recognition by scene alignment. *Neural Information Processing Systems (NIPS)*, 2007.

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

[88] J. Sivic and A. Zisserman. Video Google: A text retrieval approach to object matching in videos. In *ICCV*, volume 2, pages 1470–1477, October 2003.

[89] Josef Sivic, Biliana Kaneva, Antonio Torralba, Shai Avidan, and Bill Freeman. Creating and exploring a large photorealistic virtual space. In *First IEEE Workshop on Internet Vision at CVPR*, 2008.

[90] Noah Snavely, Steven M. Seitz, and Richard Szeliski. Photo tourism: exploring photo collections in 3d. *ACM Trans. Graph.*, 25(3):835–846, 2006.

[91] Alexander Sorokin and David Forsyth. Utility data annotation with amazon mechanical turk. In *First IEEE Workshop on Internet Vision at CVPR 08*, 2008.

[92] Merrielle Spain and Pietro Perona. Measuring and predicting importance of objects in our visual world. Technical Report CNS-TR-2007-002, California Institute of Technology Tech Report, 2007.

[93] Merrielle Spain and Pietro Perona. Some objects are more equal than others: measuring and predicting importance. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2008.

[94] L. Standing. Learning 10,000 pictures. *Quarterly Journal of Experimental Psychology*, 25:207–222, 1973.

[95] Craig Stanfill and David Waltz. Toward memory-based reasoning. *Communications of the ACM*, 29(12):1213–1228, 1986.

[96] Jian Sun, Lu Yuan, Jiaya Jia, and Heung-Yeung Shum. Image completion with structure propagation. *ACM Trans. Graph.*, 24(3):861–868, 2005.

[97] Richard Szeliski. "Where am I?": ICCV 2005 Computer Vision Contest. http://research.microsoft.com/iccv2005/Contest/.

[98] W.B. Thompson, C.M. Valiquette, B.H. Bennett, and K.T. Sutherland. Geometric reasoning for map-based localization. *Spatial Cognition and Computation*, 1(3), 1999.

[99] S. Thorpe, D. Fize, and C. Marlot. Speed of processing in the human visual system. *Nature*, 581:520–522, 1996.

[100] A. Torralba, R. Fergus, and W. T. Freeman. 80 million tiny images: a large dataset for non-parametric object and scene recognition. *IEEE PAMI*, 30(11):1958–1970, 2008.

[101] A. Torralba, A. Oliva, M. Castelhano, and J.M Henderson. Contextual guidance of eye movements and attention in real-world scenes: the role of global features in object search. *Psychological Review*, 113:766–786, 2006.

[102] A. Torralba and P. Sinha. Statistical context priming for object detection. In *Proceedings of the IEEE International Conference on Computer Vision, ICCV*, pages 763–770, 2001.

[103] Antonio Torralba, Rob Fergus, and Yair Weiss. Small codes and large image databases for recognition. In *CVPR*, 2008.

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

[105] Antonio Torralba and Aude Oliva. Statistics of natural images categories. *Network: Computation in Neural Systems*, 14:391–412, 2003.

[106] A. Vedaldi, V. Gulshan, M. Varma, and A. Zisserman. Multiple kernels for object detection. In *Proceedings of the International Conference on Computer Vision*, 2009.

[107] Pascal Vincent and Yoshua Bengio. K-local hyperplane and convex distance nearest neighbor algorithms. In *NIPS*, 2002.

[108] Julia Vogel and Bernt Schiele. Semantic modeling of natural scenes for content-based image retrieval. *Int. J. Comput. Vision*, 72(2):133–157, 2007.

[109] Julia Vogel, Adrian Schwaninger, Christian Wallraven, and Heinrich H. Bülthoff. Categorization of natural scenes: local vs. global information. In *APGV '06: Proceedings of the 3rd symposium on Applied perception in graphics and visualization*, pages 33–40, 2006.

[110] G. Wang, D. Hoeim, and D. A. Forsyth. Building text features for object image classification. In *CVPR*, 2009.

[111] G. Wang, D. Hoeim, and D. A. Forsyth. Learning image similarity from flickr groups using stochastic intersection kernel machines. In *ICCV*, 2009.

[112] Jue Wang and Michael Cohen. Simultaneous matting and compositing. *ACM Trans. Graph.*, 2007.

[113] M. Wertheimer. Laws of organization in perceptual forms (partial translation). In W.B. Ellis, editor, *A sourcebook of Gestalt Psychology*, pages 71–88. Harcourt Brace and Company, 1938.

[114] Y. Wexler, E. Shechtman, and M. Irani. Space-time video completion. *CVPR*, 01:120–127, 2004.

[115] Marta Wilczkowiak, Gabriel J. Brostow, Ben Tordoff, and Roberto Cipolla. Hole filling through photomontage. In *BMVC*, pages 492–501, July 2005.

[116] Jianxin Wu and James M. Rehg. Where am i: Place instance and category recognition using spatial pact. In *CVPR*, 2008.

[117] Hao Zhang, Alexander C. Berg, Michael Maire, and Jitendra Malik. Svm-knn: Discriminative nearest neighbor classification for visual category recognition. In *CVPR '06*, 2006.

[118] Wei Zhang and Jana Kosecka. Image based localization in urban environments. In *3DPVT '06*, 2006.

[119] Yantao Zheng, Ming Zhao, Yang Song, Hartwig Adam, Ulrich Buddemeier, Alessandro Bissacco, Fernando Brucher, Tat-Seng Chua, and Hartmut Neven. Tour the world: building a web-scale landmark recognition engine. In *CVPR*, 2009.