Chapter 3
Large-Scale Image Geolocalization

James Hays and Alexei A. Efros

Abstract In this chapter, we explore the task of global image geolocalization—estimating where on the Earth a photograph was captured. We examine variants of the "im2gps" algorithm using millions of "geotagged" Internet photographs as training data. We first discuss a simple to understand nearest-neighbor baseline. Next, we introduce a lazy-learning approach with more sophisticated features that doubles the performance of the original "im2gps" algorithm. Beyond quantifying geolocalization accuracy, we also analyze (a) how the nonuniform distribution of training data impacts the algorithm (b) how performance compares to baselines such as random guessing and land-cover recognition and (c) whether geolocalization is simply landmark or "instance level" recognition at a large scale. We also show that geolocation estimates can provide the basis for image understanding tasks such as population density estimation or land cover estimation. This work was originally described, in part, in "im2gps" [9] which was the first attempt at global geolocalization using Internet-derived training data.

3.1 Introduction

Is it feasible to estimate the location of 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 that may be impossible to specifically localize? We know that our world is self-similar not just locally but across the globe.

J. Hays (✉)
Brown University, Providence, RI, USA
e-mail: hays@cs.brown.edu

A.A. Efros
University of California, Berkeley, CA, USA
e-mail: efros@eecs.berkeley.edu

J. Choi and G. Friedland (eds.), *Multimodal Location Estimation of Videos and Images*,
DOI 10.1007/978-3-319-09861-6_3

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 images 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.

Consider the photographs in Fig. 3.1. 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 even this vague description allows us to disregard all noncoastal, nontropical 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.

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 [2]. 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 [8, 33].

In this chapter, we examine algorithms for estimating a distribution over geographic locations from an image using a data-driven scene matching approach. For this task, we leverage a dataset of over 6 million GPS-tagged images from Flickr.com.

Fig. 3.1 What can you say about where these photos were taken?

We measure how often a geolocation strategy can correctly locate a query photo, where "correct" is defined as "within 200 km of the actual location." with meta-tasks such as land cover estimation and urban/rural classification.

A key idea of the im2gps work is that humans or algorithms can estimate the location of a photograph *without* having to perform "instance level" or "landmark" recognition. While instance-level recognition techniques are impressive, we show that such matches only account for about one half of successful geolocalizations.

3.1.1 Background

There exist a variety of geolocalization algorithms operating on different input modalities. While we will review a few techniques, see [19] for a broader survey. Im2gps assumes the input is an unlabeled photograph while other methods make use of sequences of photos [12] or try to relate ground level views to aerial imagery [17]. Jacobs et al. [11] propose a clever method to geolocalize a webcam by correlating its video-stream with satellite weather maps over the same time period. Visual localization on a topographical map was one of the early problems in computer vision. It turns out to be challenging for both computers and humans [32], but recent methods [1] based on terrain "curvelet" features work surprisingly well.

The availability of GPS-tagged images of urban environments coupled with advances in multiview 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 [31] at ICCV'05 (winning entry described in [37]). Similar local feature geometric matching approaches have also been successfully applied to co-registering online photographs of famous landmarks for browsing [30] and summarization [28], as well as image retrieval in location-labeled collections, e.g. [4]. Landmark photos are linked to Wikipedia photos and articles within a specified city in [26]. Since the publication of im2gps, [6] and [38] have attacked the global landmark recognition problem, in the latter case scaling up to thousands of landmarks with high accuracy.

But can these geometric local feature 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 300,000 square kilometers of corn fields in the USA). Clearly, a generalization of some sort is required.

On the other side of the spectrum from instance-level recognition is the task of scene categorization which tries to group forests with forests, kitchens with kitchens, deserts with deserts, etc. A large body of work exists on scene recognition [16, 22, 27, 34, 35] which involves defining a fixed taxonomy of scene categories and using various features to classify a novel image into one of these categories.

We use a combination of features from both the local feature matching literature (best suited for instance-level recognition) as well as features more commonly seen in category recognition (best suited for recognizing broader geographic concepts, e.g., "Mediterranean"). 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.

3.1.2 Chapter Outline

In Sect. 6.4 we create training and testing databases from geotagged Internet images. In Sect. 3.3 we discuss the original, relatively simple "im2gps" geolocalization algorithm [9]. In Sect. 3.4 we add new features and utilize a lazy learning technique to nearly double the original "im2gps" performance. In Sect. 3.5, we analyze factors affecting geolocalization accuracy such as geographic bias in the influence of instance-level landmark matching.

3.2 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 labeled 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 labeled 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 only 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 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 in India; pet dogs are popular in the USA but not in Syria) the customs are so broadly distributed that it is not very useful for geolocation.

However, we find 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 relevant training data. People may geotag 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-labeled 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 Fig. 3.2 shows the data is very nonuniformly distributed towards places where people live or travel. We will revisit this nonuniform distribution in Sect. 3.5.1. It can be seen as a desirable property in that this is the same distribution from which people would generate query images or undesirable since it leaves huge portions of the world under-sampled.

3.2.1 Evaluation Test Set

To evaluate geolocalization performance we use a separate, held-out test set of geolocated images. We built the test set by drawing 400 random images from the original dataset. From this set, we manually remove the types of undesirable photos that we

Fig. 3.2 The distribution of photos in our database. Photo locations are cyan. Density is overlaid with the "jet" color map (log scale)

Fig. 3.3 A sample of the 237 image im2gps test set. Note how difficult it is to specifically geolocalize most of the images

tried to excluded during database construction—abstract photos, overly processed or artistic photos, and black and white photos. We also exclude photos with significant artifacts such as motion blur or extreme noise. Finally we remove 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 Fig. 3.3). 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.

3.3 Simple, Baseline Geolocalization Method

This section briefly describes the original "im2gps" method [9]. We treat this as a baseline for later studies in Sects. 3.4 and 3.5. In this section, we first look at a handful of relatively simple "baseline" global image features. We hope that some of these image properties correlate with geographic location.

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. [33]. Inspired by this work we will use 16 by 16 color images as one of our base features.

Color histograms. 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 [20] by clustering our dataset's responses to a bank of filters with eight orientations, two scales, and two 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. This representation is quite similar to dense "visual words" of local features.

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 [13]. 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 [23] has been shown to work well for scene categorization [22] and for retrieving semantically and structurally similar scenes [8]. 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. [10]. 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 8×8 and use L2 distance to compare them.

We precompute all features for the 6.5 million images. At 15 s per image this requires a total of 3.08 CPU years, but is trivially parallelized.

Our baseline geolocation algorithm is quite simple—for each query we find the nearest neighbor scene in our database according to these features. We then take the GPS coordinate of that nearest neighbor match as our geolocation estimate.

3.3.1 Is the Data Helping?

A key question for us is *how strongly does image similarity correlate with geographic proximity*? To geolocalize 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 Fig. 3.4 we plot how frequently the 1-NN increase 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.

Given a photo, how often can we pin-point the right city? Country? Continent? With our simple baseline geolocalization algorithm, the first nearest neighbor is within 64 km of the true location 12 % of the time, within 200 km 16 % of the time, within 750 km 25 % of the time, and within 2,500 km 50 % of the time.

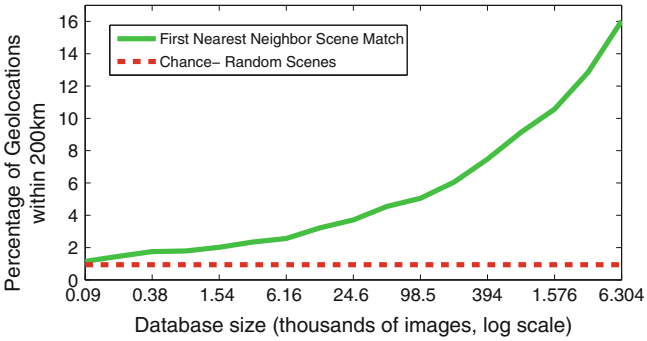

Fig. 3.4 *Accuracy of simple geolocalization baseline across database sizes*. Percentage of test set images that were correctly localized within 200 km of ground truth as function of dataset size using 1-NN. As the database shrinks the performance converges to chance

3.3.2 Grouping Geolocation Estimates

1-NN approaches are sensitive to noise. Alternatively, we also consider a larger set of kNN ($k = 120$ 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 [5] 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 200 km (although performance is not especially sensitive to this parameter). 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. In practice, this works no better than 1-NN, but we will use these mean shift clusters as the basis for our learning algorithm in Sect. 3.4.4. 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 Fig. 3.5. Cluster membership is indicated with a colored border around the matching scenes and with colored markers on the map.

3.4 Improving Geolocalization with More Features and Lazy Learning

The features (global histograms, gist, bag of textons, etc) and prediction method (1 nearest neighbor) in the previous section represent the capabilities of the original im2gps system [9]. However, we can dramatically improve global geolocalization accuracy with more advanced features and more sophisticated learning.

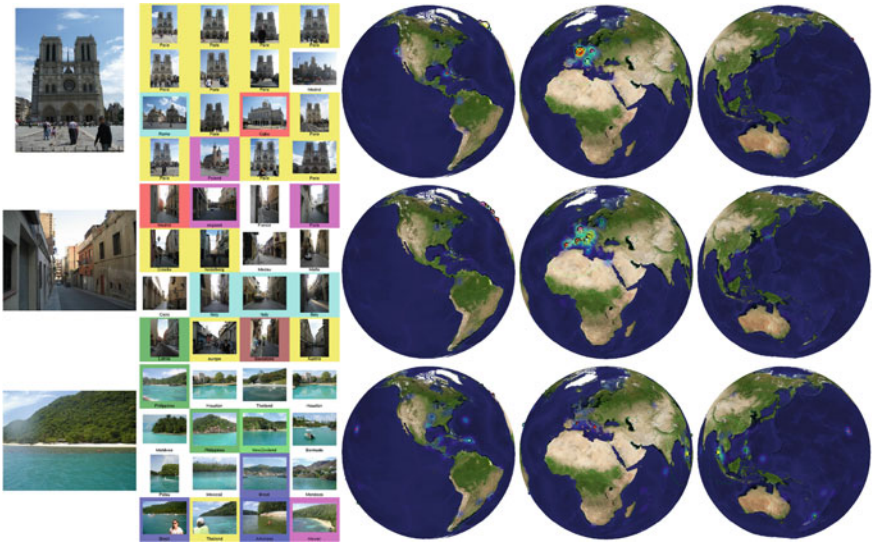

Fig. 3.5 *Results of simple geolocalization baseline*. From left to right: query images, nearest neighbors, and three visualizations of the estimated geolocation probability map. The probability map is shown as a jet-colorspace overlay on the world map. Cluster modes are marked with circumscribed "X"'s 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 200, 750, and 2,500 km. From top to bottom, these photos were taken in Paris, Barcelona, and Thailand

First, we describe additional scene matching features which are intended to be more robust than those used in the previous section. Two shortcomings of the baseline features are (1) sensitivity to scene layout and (2) poor performance at instance-level recognition. To address these problems we describe additional geometry derived and SIFT-derived features.

Second, we use "lazy learning" with these additional features. We train a multiclass, kernel SVM to decide which mean shift cluster of scene matches a query belongs to. Together, the new features and lazy learning double the baseline im2gps performance.

3.4.1 Geometry Specific Color and Texton Histograms

The baseline scene descriptors are all "global"—encompassing statistics of the entire image or built on fixed image grid regardless of scene layout. This means that irrelevant scene transformations (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 means that inconsequential image differences

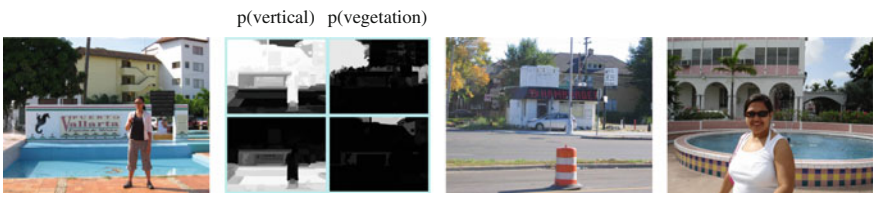

Fig. 3.6 For each geometric class we build separate color and texton histograms. Scene matching is improved by restricting the histogram comparisons to corresponding geometric regions

will prevent otherwise good scene matches from being retrieved. To address this and make histogram comparisons more meaningful we build color and texton 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 (Fig. 3.6).

We use geometric context [10] 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 [14] 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 8×8 image features.

3.4.2 Bags of SIFT Features

SIFT [18] derived features have been used for scene representations with spatial pyramids composed of densely sampled local features still near the state of the art for scene recognition [15]. In these cases, the quantization of visual words is typically rather coarse (e.g., 500 visual words). Quantized SIFT features have also been shown to work well for instance-level recognition in large datasets [29]. Larger vocabularies (1 million visual words) and geometric verification of candidate matches improve performance further [24]. Landmark geolocation methods [6, 38] have relied entirely on these types of features.

Inspired by these successes, we compute SIFT features at interest points detected by Hessian-affine and MSER [21] detectors. For each interest point type, we build vocabularies of 1,000 and 50,000 visual words based on a random subset of the database. 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. 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 [25], 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.

3.4.3 Geolocalization with Additional Features

While these features perform especially well when coupled with a more sophisticated machine learning method (Sect. 3.4.4), as a baseline we again use the first nearest neighbor method. 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 scene matching process is implemented hierarchically—first, 2,000 nearest neighbors are found with the baseline features and then distances are computed for the new geometry derived and SIFT features and the matches are reranked.

Compared to the baseline 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. Figure 3.7 shows cases where the first nearest neighbor with the new features is dramatically improved from the baseline features. For common scene types under canonical viewpoints, the difference is less noticeable.

Recall that with the 237 image im2gps test set and base im2gps features, the first nearest neighbor is within 200 km of a query 16 of the time. Using the four SIFT histograms by themselves (after the initial hierarchical search) gives an accuracy 18.6 %. Using all features improves accuracy to 21.1 %.

3.4.4 Lazy Learning for Large-Scale Scene Geolocalization

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 learn which dimensions are relevant for a particular task.

Fig. 3.7 *Nearest Neighbors with New Features*. The features introduced in this section 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 nonlandmark 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

This is critical because our feature representation is quite high-dimensional. In total, including the features from the baseline method, we have an over-complete set of 22 elementary features. The baseline features total 2,201 dimensions, while the features proposed in this section total 109,436 dimensions dominated by the two 50,000 entry SIFT histograms. Such high feature dimensionality is problematic for nearest-neighbor methods. Unfortunately, more sophisticated learning approaches are difficult to apply when the number of training samples is large (over 6 million in our case) and the feature dimensionality is high (over 100,000 in our case).

We adopt a "lazy learning" approach inspired by SVM-KNN [36] and prior supervised, KNN enhancements (See [3] for an overview of "local" learning methods). Lazy learning methods are hybrids of nonparametric, KNN techniques and parametric, supervised learning techniques. Our supervised lazy learning can be seen as a post-process to refine the nearest-neighbor search we use as a baseline.

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 datasets that makes lazy learning attractive to us.

For a novel query, our algorithm is:

1. Find $K_{sl} = 2{,}000$ nearest neighbors using the "baseline" features defined in Sect. 3.3.
2. Reduce the K_{sl} nearest neighbors to K using both "baseline" features and the additional features introduced in this section.
3. Cluster the K nearest neighbors according to their geographic locations using mean shift. We use a bandwidth of 200 km. Each of the $\mathscr{C}$ clusters is now considered a distinct class for the sake of learning. Typical values of $\mathscr{C}$ are 30–60, 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 $\mathscr{C}$ 1-vs-all nonlinear SVMs.
6. For each of the $\mathscr{C}$ 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 mean shift cluster.
7. The final geolocation estimate for the query is then the average GPS coordinate of all members of the winning cluster.

As K becomes small, the technique reduces to 1NN. As K becomes large, the technique reduces to a standard kernel SVM (which is intractable with our scale of data).

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. However, for 75 % of queries, the $K_{sl} = 120$ nearest neighbors according to the baseline features have at least one image within 200 km of the ground truth location. Thus we can have some confidence that geographically nearby scenes are being included among our nearest neighbors and taking part in the learning process.

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 pre-defined as in SVM-KNN [36]. 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.

3.4.4.1 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 $K = \sim 200$ and our running time is still dominated by the initial search which takes ~2.5 min (amortized over many queries). We have made little effort to optimize the initial search although "tiny images" [33] 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.

3.4.5 Geolocalization Results with New Features and Lazy Learning

With a one nearest neighbor algorithm, our accuracy is 16 % with baseline features and 21 % with more advanced, higher dimensional features. Replacing the one nearest neighbor prediction with the lazy learning method raises our accuracy to 31 %, nearly doubling the performance of the original im2gps publication [9]. We show four geolocalization results in Figs. 3.8 and 3.9.

3.5 Why Does it Work? Deeper Performance Analysis

3.5.1 Measuring Performance Without Geographic Bias.

Since the geographic distribution of data appears to be peaked in relatively few places (Fig. 3.10), one concern is that our performance could be a result of random guessing. In fact, the chance that two photos are within 200 km in the im2gps database is about 1.2 %. For our test set of 237 images 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 % of the im2gps database images (and probably 4.9 % of Internet images) are within 200 km 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 our simple baseline geolocalization algorithm perform if the test set distribution was not geographically peaked? To quantitatively evaluate this issue we

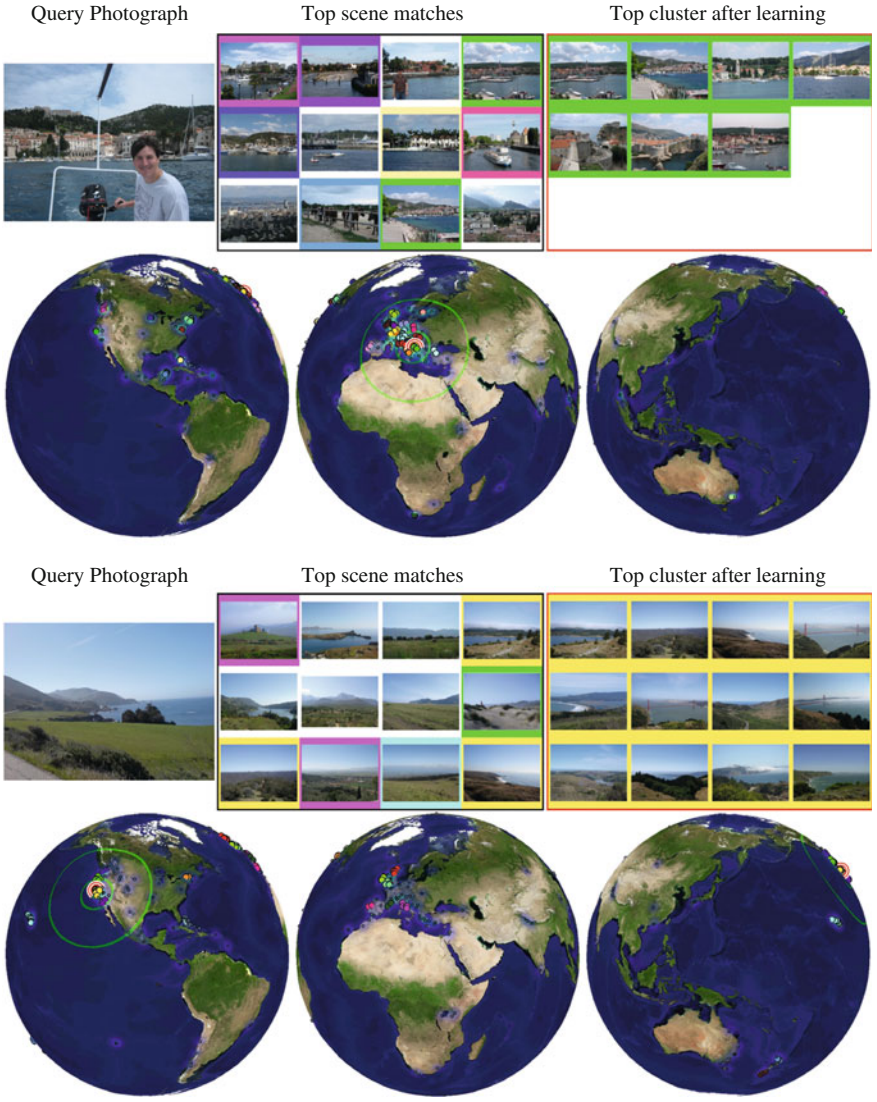

Fig. 3.8 *Geolocalization Results with Lazy Learning*. Results are generated from $K = 200$ nearest neighbors clustered with a mean shift bandwidth of 200 km and a minimum cluster size of 3. The scene match montages are scanline ordered according to scene match distances. The colors of scene match borders 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 200, 750, and 3,000 km. 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*

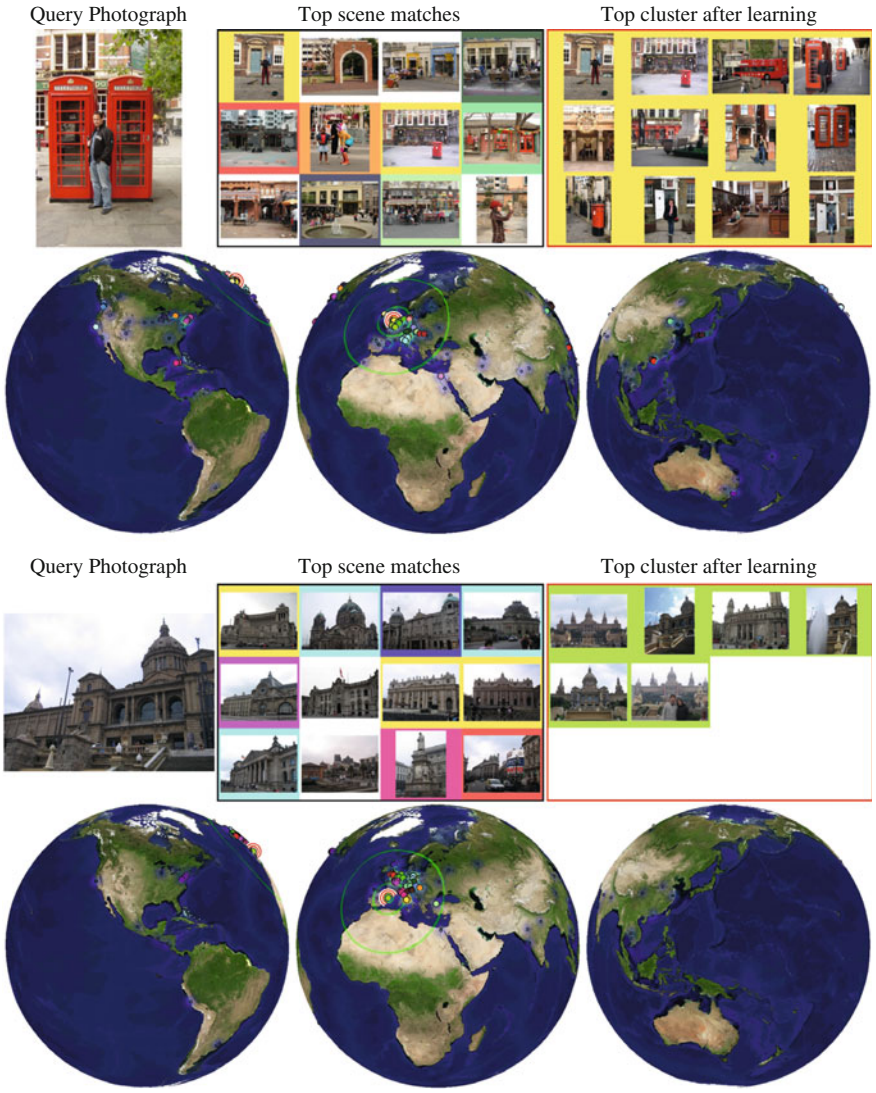

Fig. 3.9 Additional Geolocalization Results with Lazy Learning

define a new *geographically uniform* test set. We tessellate the globe into quadrilateral regions roughly 400 km on edge (Fig. 3.10). We take one query from each of the 955 regions in that have at least ten photographs. Chance is an order of magnitude lower

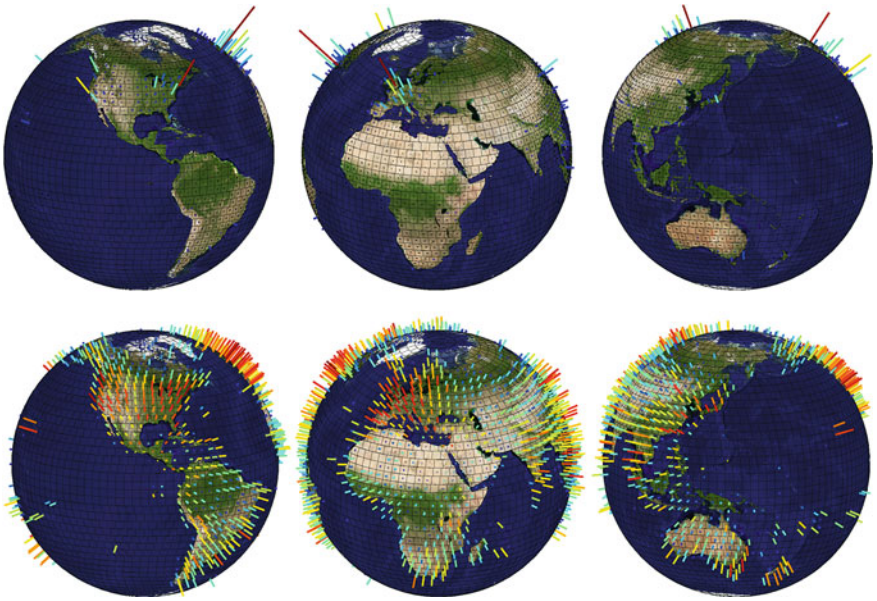

Fig. 3.10 *Photo density in im2gps database*, linear scale (*top*) and natural log scale (*bottom*). The height of each bar is proportional to the density of geotagged photos in each equal area region. The bars are colored according to the Matlab "jet" color scheme. Regions with zero photos have no bar

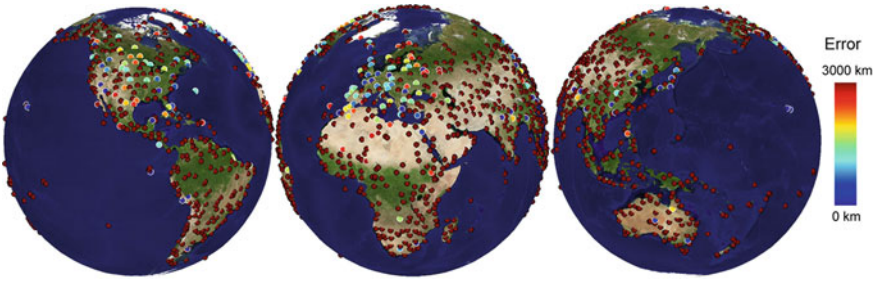

Fig. 3.11 *Accuracy on Geographically Uniform Test Set*. For each photo in the test set, the marker color indicates how accurately the photo was geolocated

for this database—only 0.13 %.[1] Figure 3.11 shows the geographic distribution of the test set, as well as the geolocation accuracy for each photo. We are unable to correctly localize any queries in large regions of South America, Africa, and Central Asia. Overall, for only 2.3 % of the test set is the first nearest neighbor is within

[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 126,663 km^2 and the land area of the Earth is 148,940,000 km^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.

200 km of the query photo's location. Interestingly, relative to chance, this is just as much of an improvement as on the im2gps test set (∼16 times better).

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.

3.5.2 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 (Fig. 3.4). 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 [22]. Why does our performance double as our database increases from 600,000 to 6 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 3.12 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 USA, 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 land cover maps 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."

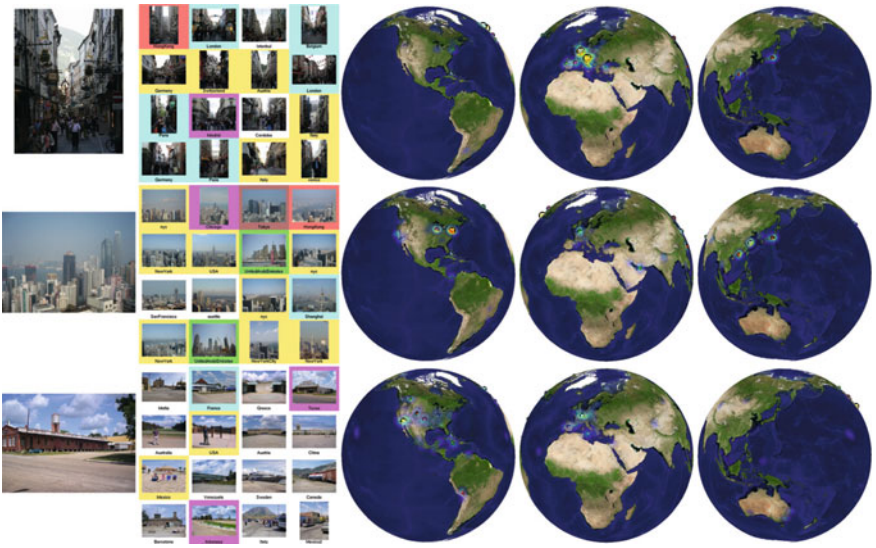

Fig. 3.12 *im2gps results for different cities*. These city queries from Germany, Hong Kong, and Alabama produce very different geolocation estimates

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 200 km. 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.

3.5.3 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. For our baseline algorithm, of the 16 % of queries whose first nearest neighbor is within 200 km, 58 % of the 1 NN 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 when using the higher dimensional features and lazy learning introduced in Sect. 3.4. With the improved features, the first nearest neighbor is the same scene for 40 % of successfully localized queries. The cluster chosen by the learning contains an instance-level match 58 % of the time. In some of these cases, the geolocation probably would have been correct even without the instance matches.

Thus, instance-level recognition does account for a slim majority of successful geolocalizations for both the simpler and more complex geolocalization strategies. But we are also able to localize a significant number of photos that are not landmarks and would presumably fall through the cracks of methods such as [6, 38].

3.6 Discussion

Not only is photo geolocalization an important problem in itself, but it could also be tremendously useful to many other vision tasks. Knowing the distribution of likely locations 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. Even a coarse geo-location can provide a useful object prior for recognition. For example, 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.

Im2gps [9] was the first study of global image geolocation, a task that only became possible because of the emergence of large-scale geotagged Internet imagery. While the baseline im2gps approach was relatively simple, with the additional features and learning discussed in Sect. 3.4, our results are qualitatively and quantitatively greatly improved. In fact, our geolocalization accuracy exceeds that of nonexpert humans [7]. Typically, humans are implicitly treated as an upper bound for performance in vision tasks (e.g., object detection). Have we saturated performance for automatic image geolocalization? Definitely not. There is still a great deal of room for improvement. As Fig. 3.11 shows, the algorithm has trouble localizing photographs from sparsely sampled regions of the world unless they contain distinct landmarks. While it was hoped that our scene matching might be able to pick up on subtle landscape, vegetation, or architecture cues to geolocalize images this is rarely observed. Our algorithm's advantage over humans is its large visual memory, not its ability to relate scene statistics to geographic locations. Geolocalization performance should increase as algorithms include more high-level reasoning about architecture, writing, clothing, lighting direction, geology, and plant and animal species.

Acknowledgments 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. Funding for this work was provided by an NSF fellowship to James Hays and NSF grants CAREER 1149853, CAREER 0546547, and CCF-0541230.

References

1. G. Baatz, O. Saurer, K.Köser, M. Pollefeys, Large scale visual geo-localization of images in mountainous terrain, In *Proceedings of the 12th European Conference on Computer Vision - Volume Part II*, (2012), pp. 517–530
2. M. Bar, The proactive brain: using analogies and associations to generate predictions. Trends Cogn. Sci. **11**(7), 280–289 (2007)
3. S.S. Chris Atkeson, Andrew Moore, Locally weighted learning. AI. Review **11**, 11–73 (1997)
4. O. Chum, J. Philbin, J. Sivic, M. Isard, A. Zisserman, Total recall: Automatic query expansion with a generative feature model for object retrieval, in *Proceedings of ICCV*, 2007
5. D. Comaniciu, P. Meer, Mean shift: A robust approach toward feature space analysis. IEEE Trans. Pattern Anal. Mach. Intell. **24**(5), 603–619 (2002)
6. D.J. Crandall, L. Backstrom, D. Huttenlocher, J. Kleinberg. Mapping the world's photos, in *WWW '09: Proceedings of the 18th international conference on World wide web 2009*, pp. 761–770, 2009
7. J. Hays, A. Efros. Where in the world? human and computer geolocation of images, in *Vision sciences society meeting*, 2009
8. J. Hays, A.A. Efros. Scene completion using millions of photographs, in *ACM Transactions on Graphics (SIGGRAPH 2007)*, 26(3), 2007
9. J. Hays, A.A. Efros. im2gps: estimating geographic information from a single image, in *CVPR*, 2008
10. D. Hoiem, A. Efros, M. Hebert, Recovering surface layout from an image. Int. J. Comput. Vision. **75**(1), 151–172 (2007)
11. N. Jacobs, S. Satkin, N. Roman, R. Speyer, R. Pless, *Geolocating static cameras*, in *Proceedings, ICCV*, 2007
12. E. Kalogerakis, O. Vesselova, J. Hays, A.A. Efros, A. Hertzmann. Image sequence geolocation with human travel priors, in *Proceedings of the IEEE International Conference on Computer Vision (ICCV '09)* (2009)
13. J. Kosecka, W. Zhang. Video compass, in *ECCV '02: Proceedings of the 7th European Conference on Computer Vision-Part IV*, 2002, pp. 476–490
14. J.-F. Lalonde, D. Hoiem, A.A. Efros, C. Rother, J. Winn, A. Criminisi. Photo clip art. *ACM Transactions on Graphics (SIGGRAPH 2007)*, vol. 26(3) (August 2007)
15. S. Lazebnik, C. Schmid, J. Ponce. Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories, in *CVPR* (2006)
16. L.-J. Li, L.F. Fei, *What, where and who? classifying events by scene and object recognition*, in *Proceedings, ICCV*, (2007)
17. T.-Y. Lin, S. Belongie, J. Hays. Cross-view image geolocalization, in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (Portland, OR, June 2013)
18. D. Lowe, Object recognition from local scale-invariant features. ICCV **2**, 1150–1157 (1999)
19. J. Luo, D. Joshi, J. Yu, A. Gallagher, Geotagging in multimedia and computer visiona survey. Multime'd Tools Appl. **51**, 187–211 (2011)
20. D. Martin, C. Fowlkes, D. Tal, J. Malik. A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics, in *Proceedings ICCV* (July 2001)
21. J. Matas, O. Chum, M. Urban, T. Pajdla, Robust wide-baseline stereo from maximally stable extremal regions. Image Vis. Comput. **22**(10), 761–767 (2004)
22. A. Oliva, A. Torralba, Modeling the shape of the scene: a holistic representation of the spatial envelope. Int. J. Comput. Vision **42**(3), 145–175 (2001)
23. A. Oliva, A. Torralba. Building the gist of a scene: The role of global image features in recognition, in *Visual Perception, Progress in Brain Research*, 2006, vol. 155
24. J. Philbin, O. Chum, M. Isard, J. Sivic, A. Zisserman. Object retrieval with large vocabularies and fast spatial matching, in *CVPR* (2007)

25. J. Philbin, O. Chum, M. Isard, J. Sivic, 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)
26. T. Quack, B. Leibe, L. 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)
27. L.W. Renninger, J. Malik, When is scene recognition just texture recognition? Vis. Res. **44**, 2301–2311 (2004)
28. I. Simon, N. Snavely, S.M. Seitz. Scene summarization for online image collections, in *Proceedings, ICCV* (2007)
29. J. Sivic, A. Zisserman, Video Google: A text retrieval approach to object matching in videos. ICCV **2**, 1470–1477 (2003)
30. N. Snavely, S.M. Seitz, R. Szeliski, Photo tourism: exploring photo collections in 3d. ACM Trans. Graph. **25**(3), 835–846 (2006)
31. R. Szeliski. "Where am I?": ICCV 2005 Computer Vision Contest. http://research.microsoft.com/iccv2005/Contest/
32. W. Thompson, C. Valiquette, B. Bennett, K. Sutherland, Geometric reasoning for map-based localization. Spatial Cogn. Comput **1**(3), 291–321 (1999)
33. A. Torralba, R. Fergus, W.T. Freeman, 80 million tiny images: a large dataset for non-parametric object and scene recognition. IEEE PAMI **30**(11), 1958–1970 (2008)
34. J. Vogel, B. Schiele, Semantic modeling of natural scenes for content-based image retrieval. Int. J. Comput. Vis. **72**(2), 133–157 (2007)
35. J. Xiao, J. Hays, K. Ehinger, A. Oliva, A. Torralba. Sun database: Large-scale scene recognition from abbey to zoo, in *CVPR* (2010)
36. H. Zhang, A.C. Berg, M. Maire, J. Malik. Svm-knn: Discriminative nearest neighbor classification for visual category recognition, in *CVPR '06* (2006)
37. W. Zhang, J. Kosecka. Image based localization in urban environments, in *3DPVT '06* (2006)
38. Y. Zheng, M. Zhao, Y. Song, H. Adam, U. Buddemeier, A. Bissacco, F. Brucher, T.-S. Chua, H. Neven. Tour the world: building a web-scale landmark recognition engine, in *CVPR* (2009)