Improved Geometric Verification for Large Scale Landmark Image Collections

BMVC 2012 Submission # 217

Abstract

In this work, we address the issue of geometric verification, with a focus on modeling large-scale landmark image collections gathered from the internet. In particular, we show that we can compute and learn descriptive statistics pertaining to the image collection by leveraging information that arises as a by-product of the matching and verification stages. In turn, this learned information can be used to significantly improve the efficiency of the overall system. Our approach is based on the intuition that matching numerous image pairs of the same geometric scene structures quickly reveals useful information about two aspects of the image collection: (a) the validity of individual visual words and (b) the appearance of landmarks in the image collection. Both of these sources of information can then be used to drive any subsequent processing, thus allowing the system to bootstrap itself. While current techniques make use of dedicated training/preprocessing stages, our approach elegantly integrates into the standard geometric verification pipeline of typical internet photo collection reconstruction systems, by simply leveraging the information revealed during the verification stage. The main result of this work is that this "learning-as-you-go" approach significantly improves performance; our experiments on large scale internet photo collections demonstrate significant improvements in efficiency and completeness over standard techniques.

1 Introduction

Our main focus in this work is the issue of geometric verification, which is a fundamental component of any system that seeks to model large scale contaminated photo collections gathered from the internet [1, 6, 16, 18]. Recent years have seen remarkable progress in this area, and current systems are capable of producing 3D models from city-scale datasets containing hundreds of thousands, or even millions of images, within a fairly short time span [1, 6]. In this work, we seek to improve the efficiency of these state of the art approaches by addressing one of the most computationally expensive operations in this process.

In designing a 3D reconstruction system for internet photo collections, one of the key considerations is robustness to "clutter" – when operating on datasets downloaded using keyword searches on community photo sharing websites (such as Flickr), it has been observed that invariably, a large fraction of images in the collection are unsuitable for the purposes of 3D reconstruction [7, 16]. In addition, these datasets are "unorganized", meaning that we have no information about the spatial relationships, if any, between the images. Thus, one of the fundamental steps in a 3D reconstruction system is *geometric verification*: the process of determining which images in an internet photo collection are geometrically related to each

other (i.e., images of the same 3D scene). This is a computationally expensively step; a simple exhaustive pairwise comparison of images leads to a quadratic algorithm that cannot scale to handle large scale image collections with hundreds of thousands of images. Thus, much work in recent years has focused on developing efficient ways to perform this comparison. For example, Agarwal et al. [1] use image retrieval techniques to detemine, for every image in the dataset, a small set of candidate images to match against. An alternate approach, adopted by Frahm et al. [6], is to first cluster the images based on global image descriptors, which provides a rough grouping based on viewpoint, and to then perform the verification within each cluster. These approaches have proved to be very efficient – for instance, in [6], it was shown that datasets containing up to 3 million images could be processed in approximately 24 hours, leading to dense 3D models. While this is extremely promising, there are still some limitations to this approach. For instance, even the carefullly optimized approach described in [6] spends approximately 50% of the processing time simply verifying image pairs against each other. In addition, the approach in [6] suffers from "incompleteness"; due to the coarse clustering, a very large fraction of images are discarded immediately following the clustering and verification steps (for e.g., over 95% of the images in the collection remain unmatched following these steps). In this work, we aim to overcome these limitations.

Thus far, the typical way to peform geometric verification has been to estimate the geometric relationship between pairs *independently*, which does not fully exploit the specific characteristics of the dataset. Our key insight in this work is simple: as the geometric verification progresses, we learn information about the image collection, and subsequently use this learned infomation to improve efficiency and completeness. More specifically, since images of the same geometric structures are being repeatedly verified against each other, this process of repeated matching reveals useful information about (a) the stability and validity of low-level image features and (b) the global appearance of the various landmarks in the image collection. While current techniques either ignore this information, or leverage it for other tasks via an offline processing stage, our system feeds this information directly back into the verification pipeline. Our approach, while extremely simple, is also very effective – our results on a variety of challenging datasets demonstrate significant improvements in efficiency compared to current techniques.

2 Related Work

Recent years have seen remarkable advances with respect to the modeling, organization and visualization of large-scale, heavily contaminated image collections gathered from the internet. As noted earlier, the recent approaches of [1, 6] are capable of producing 3D reconstructions of city-scale landmark image collections containing millions of images. To handle datasets of this magnitude, these approaches have primarily focused on exploiting the *parallelism* inherent in the problem, either by using clusters of computers [1] or GPUs [6]. However, far less attention has been paid to *redundancy*, in that images of the same geometric structures are verified against each other time and time again. While this cue has gone mostly ignored, we show that incorporating this information into the standard reconstruction pipeline can result in a significant computational benefit.

Also relevant to our approach are techniques for the related problem of location recognition, where the goal is to efficiently identify and return images that are geometrically related to a given query image. Given that efficiency and accuracy are important in this setting, a number of recent approaches have addressed the problem of learning how to select informa-

tive image features (or, alternatively, suppressing uninformative features) during the process of image retrieval [8, 10, 12, 19]. While similar in spirit to our idea, our goal is quite different – our aim is to fullly utilize this information in an *online* way. This is a distinguishing characteristic from current techniques that have thus far obtained this information via an offline, preprocessing step, or through a *post-hoc* phase that simply uses the output of structure-from-motion on the entire dataset. We argue that it is possible to take this simple idea one step further – i.e., to incorporate it into the standard reconstruction pipeline itself. Since this information is, in some sense, a byproduct of the process of geometric verification, it can easily be fed back into the system, instead of being ignored or recomputed later. In addition, in contrast to techniques such as [8, 10, 19], which operate at the *feature* level (i.e., using the results of exhaustive geometric verification to either prioritize or prune image features for every image in the dataset), our approach explores the weighting of *visual words*. In this respect, our approach is similar to that of [12], which uses an offline training stage to identify a subset of the visual vocabulary that contains information that is most useful for landmark identification. However, in contrast to this approach, we do not require a dedicated learning stage or any labeled training datasets; our system incrementally learns as it processes new image pairs, and efficiently uses this information to bootstrap any subsequent processing.

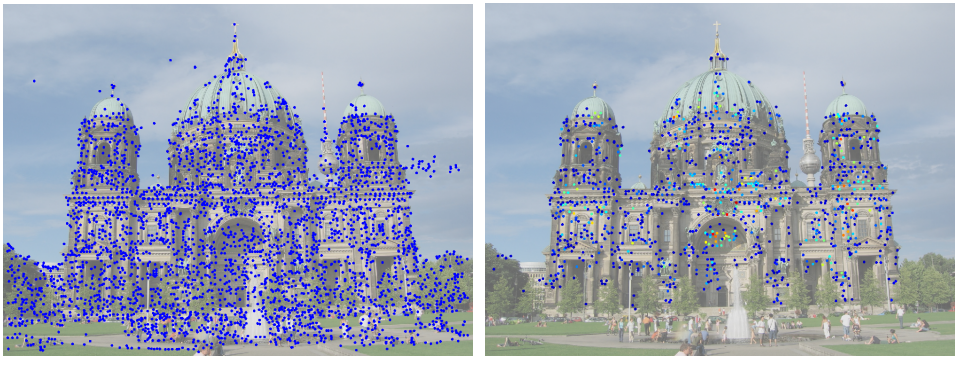
There has been some recent work related to the problem of identifying landmark images in large-scale image collections [9, 20, 21]. Our approach differs from these in that we do not require any manually labeled training imagery, as in [9]. In addition, we do not require images to have any associated geotags or GPS information, as required in [21], and we also do not require full structure-from-motion to be carried out on the entire dataset, as in [20]. Also note that our goal in this work is somewhat different - we aim to improve the efficiency of the geometric verification stage, by making use of information that is revealed as a byproduct of this step; our focus is thus on simple, but very efficient methods that do not require computationally intensive preprocessing stages, or manual labeling effort.

3 Efficient Large-scale Image Registration

3.1 Identifying useful visual words

As noted earlier, current approaches take a somewhat pessimistic view to the problem of geometric verification, by independently computing the two-view geometry for each image pair. In other words, given a pair of images, features are matched between the images to obtain a set of putative correspondences, and then a robust estimation algorithm (e.g., RANSAC [5], or one of its more efficient variants) is used to identify a set of inliers. This process then repeats for the next pair of images, typically ignoring the results produced by any previous rounds of verification. We adopt a different strategy: namely, our intuition is that this process of repeated verification reveals useful information about the validity of low-level image features. More specifically, considering a bag-of-visual-words framework, our goal is to identify visual words that are more stable, and more likely to be geometrically consistent.

As an example, consider Figure 1(a), which shows all detected SIFT [11] features for a single image. Note that a large number of features lie in areas of the image that are very unlikely to pass any geometric consistency check (for e.g., features detected on vegetation, on people, and in the sky). Now, if we have previously verified *other* images of the same scene, we can weight each visual word in the current image by the number of times that the word has previously passed the geometric consistency check in other image pairs – these



(a) (b)

Figure 1: (a) All detected features for a single image (b) Features filtered based on the results of geometric verification. In this example, we only display features corresponding to visual words that were found to be inliers in at least 10 previous image pairs. The features in (b) are heatmap colour coded based on the inlier counts.

results are visualized in Figure 1(b). Note, in particular, that this weighting has two effects: (1) it emphasizes visual words that are stable, repeatable, and more likely to be geometrically consistent and (2) suppresses visual words that are very unlikely to pass the geometric consistency check. Thus, the main idea is to incrementally accumulate this kind information, and then leverage it to improve the efficiency of the overall system. This idea has been explored to some extent in recent work [8, 19], but at the feature-level. In other words, a preprocessing step determines, for every image in the dataset, which features are likely to be useful (or, alternatively, which are likely to be "confusing"). Our approach extends on this idea in two ways: (1) we work at the visual word level, which in turn allows us to predict, for a never before seen image, which features are likely to be geometrically consistent, and (2) since our goal is geometric verification, we incorporate this visual word prioritization strategy into the verification step itself (i.e., no preprocessing or labeling of images is required).

3.1.1 Computing visual word priorities

In the interests of computational efficiency, we adopt a very simple strategy to identify potentially useful visual words. Consider a visual vocabulary, $W = \{w_1, w_2, ..., w_N\}$, consisting of N visual words. Typically, this vocabulary is generated by (approximate) k-means clustering, using a diverse set of image descriptors [14]. In addition, consider a set of visual word *priorities*, $C = \{c_1, c_2, ..., c_N\}$, where each c_i represents, in some sense, a score that is proportional to the validity of the visual word. In the absence of any prior information, we start by assigning each of these visual words the same priority (i.e., $c_i = 0, \forall i$). We then carry out geometric verification on the image collection, selecting image pairs using either the retrieval-based method as in [1], or the clustering based-method as in [6]. For each pair of images, this step typically involves matching features between the images to obtain a set of putative correpondences, and then running RANSAC [5] to identify a set of inliers.

Each pair of matching features is associated with a visual word from the set W (for

simplicity, we ignore for now the case where a pair of matched features gets assigned to different visual words in each image; we will return to this point later). For every pair of images that we *successfully* verify (where a "success" is considered to be a pair of images with $> I$ inliers; commonly chosen values of I range between 15-20), we then update the priority of the inlier visual words based on the results of this process. The simplest possible scheme would consider the set C to be a set of inlier counts – in other words, for each feature match that was found to be an inlier, we update a count c_i for the corresponding visual word. Intuitively, over time, this weighting has two effects: (1) it emphasizes visual words that are matched across many images pairs – thus, prioritizing words that are more likely to be geometrically consistent and (2) suppresses visual words that repeatedly fail the consistency check. We expect that this set of counts will gradually reveal useful information about the validity of the words. Below, we describe how these counts can be leveraged for improving the efficiency of robust estimation.

3.1.2 Improving RANSAC sampling

One immediate application of the visual word priorities is to improve the efficiency of robust estimation. For simplicity, assume we are given a set of counts $C = \{c_1, c_2, ..., c_N\}$, obtained by matching a set of image pairs. This weighting of visual words can then be incorporated into a RANSAC framework that biases the sampling in favour of the more reliable words. While current techniques either ignore this information, or leverage it for other tasks via an offline preprocessing/postprocessing stage, we propose to feed this information directly back into the verification pipeline.

Over the past decade, a number of improvements to RANSAC have been proposed, each addressing a specific weakness of the original algorithm [2, 3, 13, 15, 17]. Most relevant to this work are techniques that perform *non-uniform* sampling of the data points, using some form of prior information. Two recent examples of this category are PROSAC [2], which uses ordering information to preferentially sample points based on their rank and GroupSAC [13], which partitions points into groups based on some kind of similarity information, with the intuition being that inlier points will tend to form large, salient groupings. Given the set of inlier counts C, it is clear that this information can very easily be incorporated into a PROSAC-style sampling strategy. In particular, given a pair of images, consider a set $\mathcal{S}$ containing M matched points, $\mathbf{x}_i \leftrightarrow \mathbf{x}'_i$, for $i = 1, ..., M$. For each matched feature in $\mathcal{S}$, we have a corresponding visual word w_i, with associated count c_i. We then order the matches in $\mathcal{S}$ based on the counts c_i, and then carry out PROSAC-style sampling. PROSAC can be viewed as a process that starts by deterministically testing the most promising hypotheses (generated from the most promising data points), and gradually shifting to the sampling strategy of RANSAC as the confidence in the *a priori* sorting based on quality scores decreases. The improvement in efficiency rests on the weak assumption that the ordering defined by the quality score is no worse than a random ordering. Indeed, PROSAC is designed to draw the same samples as RANSAC, but in a more meaningful order.

It is worth noting that thus far, virtually the only kind of ordering information that has been used in PROSAC has been purely image-to-image [2, 13, 15, 17]. In other words, given a set of matched features for an image pair, the ordering of matches is determined, for instance, using some function of the SIFT matching score (e.g., ordering matches based on the ratio of the distances in the SIFT space of the best and second best match). Note that this ordering does not leverage any information from prior matching rounds – in other words, each pair of images is verified completely independently of the others. This scheme thus discards

potentially very useful information that arises as a byproduct of the verification. Particularly for the case of photo collections, where images of the same 3D scenes are repeatedly encountered, there is much to be gained by altering the ordering scheme to take prior matching results into account. We adopt precisely this strategy, sorting the set of matches based on the number of times the correponding visual words have been previously verified as being inliers. As we will show in the results in Section ZZZ, this ordering strategy results in an appreciable improvement in efficiency compared to the standard image-to-image ordering technique.

3.2 Identifying landmark images

Section 3.1 described an approach to identifying promising visual words, using information generated via repeated matching of image pairs. In many cases, it is possible to learn additional useful information as well during this process. For instance, once we have obtained a sufficiently large set of succesfully verified image pairs, we hypothesize that this set captures useful information about the global appearance of various landmarks in the dataset. More specifically, consider the system of [6], which uses a clustering-based approach to first group images by (approximate) viewpoint, and then verifies these viewpoint clusters to obtain a set of iconic images. These iconic images, in some sense, represent a concise summary of the entire image collection, and typically contain a diverse set of views of the various landmarks present in the dataset. We observe that this information can then be used to train a classifier to recognize landmark images. One of the limitations of the approach of [6] is that the clustering and verification steps are only approximate, and often, a significant fraction of images in the dataset are rejected as being irrelevant. One possible approach to increasing the number of registered images is to carry out a second "re-verification" stage, where each rejected image is matched to a small set of iconics (obtained, for instance, by using image retrieval techniques). However, this step is prohibitively expensive, particularly for very large scale image collections, where the number of rejected images is on the order of a few million. In this context, having a trained model of *landmark appearance* is potentially very useful, since this would allow us to only verify those images that are likely to be landmark images and discard the rest.

This stage of our system operates as follows: as images are processed in the pipeline described in [6], we identify a subset of these images as verified landmark images ("iconics"). These images constitute a set of positive training examples to train a landmark-vs.-non landmark classifier. In order to obtain negative training examples, we sample randomly from the set of rejected images, and attempt to register the sampled image against the set of iconics. If this process fails, it is very likely that the sampled image is a non-landmark image, and we add this to the negative training pool. This process repeats until a sufficient number (on the order of a few thousand) negative training images have been found. Following this, we train a simple binary classifier to distinguish landmark images from non-landmark images. To build this classifier, we continue to leverage our same visual vocabulary (W), but now use it to build a standard bag of visual words (a histogram of the visual words) image descriptor for each training image. We use this descriptor to train a linear support vector machine (SVM) classifier. Once trained, we run the classifier on each image before verification. If the classifier has a positive response we continue with geometric verification, but if the response is negative we reject the image immediately, thus saving the compute time of geometric verification. Detailed analysis of the performance of this classifier can be found in Section **??**.

(a) (b)

Figure 2: (a) Putative features matches and (b) inliers. In this case, the inlier ratio is $\approx 20\%$.

4 Results

4.1 Robust estimation

We first evalute the effect of the modified ordering scheme based on visual words counts, on the robust estimation stage. We adopt a recent high-performance RANSAC variant, called ARRSAC [15], which integrates PROSAC-style sampling into a real-time robust estimation framework. We report results on two different experiments, representing different usage scenarios. In the first case, we consider a dataset of 10000 images representing a single landmark (the Berlin Dome). This dataset is relatively clean, though a small fraction ($\approx 5\%$) of unrelated images are present in the dataset. We process this dataset using the approach of [1], by retrieving 20 match candidates for each image in the dataset, and performing geometric verification. In the second experiment, we process a much larger dataset, containing 2.8 million images of Berlin. To handle datasets of this magnitude, we use the implementation described in [6]. In both of these experiments, we compare the performance of (a) baseline RANSAC, (b) ARRSAC with image-to-image ordering (based on SIFT matching scores) and (c) ARRSAC with ordering based on visual words.

A specific example is shown Figure 2(a). A subset of the putative feature matches are shown. The inlier ratio for this image pair is significantly low ($\approx 20\%$), due to large changes in viewpoint and scale, coupled with repetitive patterns and symmetries. For this low inlier ratio, standard RANSAC would require close to 360000 samples, which is computationally prohibitive. A more optimized technique, such as PROSAC (refer Section 3.1.1), which leverages non-uniform sampling based on feature matching scores, requires about 20000 samples on average. It is worth noting that thus far, virtually the only weighting scheme that has been explored with PROSAC and other non-uniform sampling techniques is purely image-to-image; in other words, only using the feature matching scores between a single pair of images. Altering this weighting scheme to take the results of geometric verification into account has a significant benefit; for the image pair in Figure 2, this reduces the number of samples required to approximately 500 on average - this represents a factor of 40 reduction compared to the traditional weighting employed in PROSAC. More generally, our initial experiments with small-scale datasets (containing on the order of a thousand images of a single landmark) indicate that this is a promising direction for research that could potentially accelerate the geometric verification step significantly. We propose to investigate this approach more thoroughly on challenging large-scale datasets containing multiple landmarks.

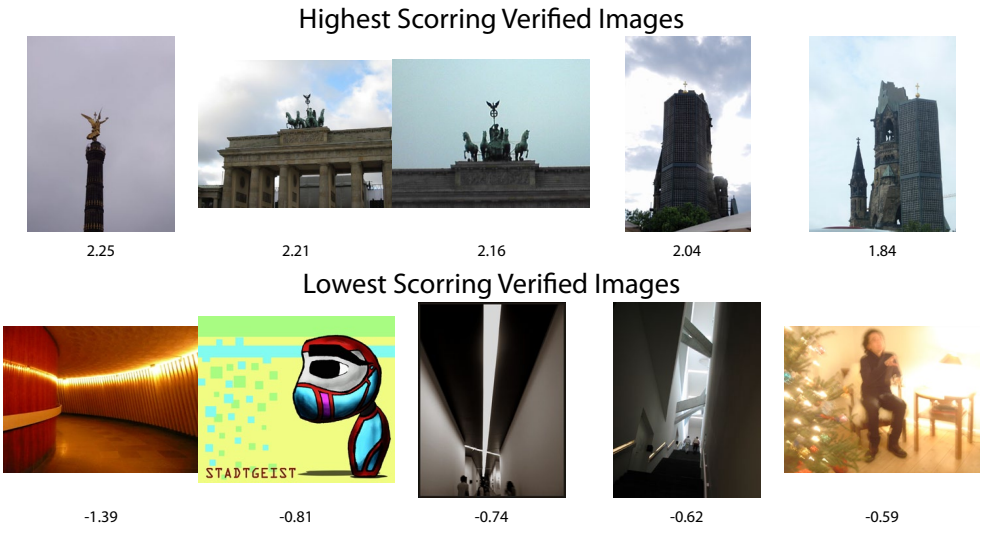


Figure 3: The top and bottom five images according to our classifier that do geometrically verify to some iconic image. Notice that, though the lowest scoring images do verify, it is actually good the classifier rejected them as they are not of landmarks, while the top five images are clearly landmark images.

4.2 Landmark classification

4.3 Identifying landmark images

After we have identified our iconic images we have only verified roughly 3% of the images in our dataset. We now wish to run geometric verification between the remaining 97% of the dataset and our iconic images. This however is quite slow as most will fail verification (failed verifications are much slower than successful verifications). To reduce the number of images we verify we use our linear SVM classifier to first weed out images that do not resemble any iconic image.

To train our SVM we take all iconic images as our positive training images. We then run our verification on random images until we have found an equal number of images that do not verify to any of our iconic images, these are our negative training images. For each image we compute a bag-of-visual-words histogram from the visual words already computed for RANSAC. Because on average there are < 2000>? visual words per image and our histogram has 1 million dimensions, our feature is very sparse. We take advantage of this sparsity at training time by using the very fast linear svm library of Fan et al. [4]. We are able to train our svm with 20,000 training images with 1 million dimensional feature vectors in 8.6 seconds. At test time the classifier evaluation is a sparse inner product which amounts to a 1-3 thousand multiplications. In out tests a classifier evaluation took less the 10^{-6} seconds.

To measure the performance of our classifier we create a test set by running geometric verification on a random subset of the images that have yet to be verified, positive examples being images that do verify to an iconic image and negative examples being images that do not verify to any iconic image. We would like our classifier to have a number of key properties. First, we would like it to not miss too many images that would verify against

an iconic image. To measure this we look at the true positive rate, which on our test set is 69.5%. This may seem low but we found that, while the images the classifier rejected did verify to some iconic, they were most often not landmark images as shown in Figure 3.

Next, we want the total number of images for which our classifier fires on, and for which we then run geometric verification on, to be low. Our classifier fires on 25.9% of test images, allowing 74.1% of images to be rejected without verification, amounting to at least a 4x speed up. Finally, the verification runs much faster on images that do verify than ones that don't so it is also important that of the images that the classifier fires on a high percentage of those images are verified. On our test set 31.1% of these images do verify compared to 11.5% across the full test set, which in effect gives and additional XX speedup.

Over all we get a speed up of XX from the classifier alone and XX when combined with the visual word based RANSAC. <— REWORD

5 Conclusion

References

[1] Sameer Agarwal, Noah Snavely, Ian Simon, Steven M. Seitz, and Richard Szeliski. Building Rome in a Day. In *International Conference on Computer Vision*, 2009.

[2] O. Chum and J. Matas. Matching with PROSAC - Progressive Sample Consensus. In *Computer Vision and Pattern Recognition, IEEE Conference on*, 2005.

[3] Ondřej Chum and Jiří Matas. Optimal Randomized RANSAC. *IEEE Trans. Pattern Anal. Mach. Intell.*, 30(8):1472–1482, 2008.

[4] Rong-en Fan, Kai-Wei Chang, Cho-Jui Hsieh, Xiang-rui Wang, and Chih-jen Lin. LIBLINEAR: A library for large linear classification. *The Journal of Machine Learning Research*, 9:1871–1874, 2008.

[5] Martin A. Fischler and Robert C. Bolles. Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography. *Communications of the ACM*, 24(6):381–395, 1981.

[6] Jan-Michael Frahm, Pierre Fite-Georgel, David Gallup, Tim Johnson, Rahul Raguram, Changchang Wu, Yi-Hung Jen, Enrique Dunn, Brian Clipp, Svetlana Lazebnik, and Marc Pollefeys. Building Rome on a Cloudless Day. In *European Conference on Computer Vision*, volume 6314, pages 368–381, 2010.

[7] L. Kennedy, S.-F. Chang, and I. Kozintsev. To Search or To Label?: Predicting the Performance of Search-based Automatic Image Classifiers. In *ACM Multimedia Information Retrieval Workshop (MIR 2006)*, 2006.

[8] Jan Knopp, Josef Sivic, and Tomas Pajdla. Avoiding confusing features in place recognition. In *Proceedings of the 11th European conference on Computer vision: Part I*, ECCV'10, pages 748–761, Berlin, Heidelberg, 2010. Springer-Verlag.

[9] Yunpeng Li, D.J. Crandall, and D.P. Huttenlocher. Landmark classification in large-scale image collections. In *International Conference on Computer Vision*, pages 1957 –1964, 2009.

[10] Yunpeng Li, Noah Snavely, and Daniel P. Huttenlocher. Location recognition using prioritized feature matching. In *Proceedings of the 11th European conference on Computer vision: Part II*, ECCV'10, pages 791–804, Berlin, Heidelberg, 2010. Springer-Verlag.

[11] David G. Lowe. Distinctive Image Features from Scale-Invariant Keypoints. *International Journal of Computer Vision*, 60:91–110, 2004.

[12] Nikhil Naikal, Allen Yang, and S. Shankar Sastry. Informative Feature Selection for Object Recognition via Sparse PCA. Technical report, 2011.

[13] Kai Ni, Hailin Jin, and Frank Dellaert. GroupSAC: Efficient Consensus in the Presence of Groupings. In *International Conference on Computer Vision*, 2009.

[14] James Philbin, Ondrej Chum, Michael Isard, Josef Sivic, and Andrew Zisserman. Object retrieval with large vocabularies and fast spatial matching. In *Computer Vision and Pattern Recognition*, pages 1–8, 2007.

[15] Rahul Raguram, Jan-Michael Frahm, and Marc Pollefeys. A Comparative Analysis of RANSAC Techniques Leading to Adaptive Real-Time Random Sample Consensus. In *European Conference on Computer Vision*, pages II: 500–513, 2008.

[16] Rahul Raguram, Changchang Wu, Jan-Michael Frahm, and Svetlana Lazebnik. Modeling and Recognition of Landmark Image Collections Using Iconic Scene Graphs. *Int. J. Comput. Vision*, 95(3):213–239, 2011.

[17] T. Sattler, B. Leibe, and L. Kobbelt. SCRAMSAC: Improving RANSAC's Efficiency with a Spatial Consistency Filter. In *International Conference on Computer Vision*, 2009.

[18] N. Snavely, S. M. Seitz, and R. Szeliski. Modeling the World from Internet Photo Collections. *International Journal of Computer Vision*, 80(2):189–210, 2008.

[19] Panu Turcot and David G. Lowe. Better matching with fewer features: The selection of useful features in large database recognition problems. In *ICCV Workshop on Emergent Issues in Large Amounts of Visual Data (WS-LAVD)*, 2009.

[20] Xian Xiao, Changsheng Xu, and Jinqiao Wang. Landmark image classification using 3D point clouds. In *International Conference on Multimedia (MM)*, pages 719–722, 2010.

[21] Yan-Tao 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 *Computer Vision and Pattern Recognition*, June, 2009.