Improved Geometric Verification for Large Scale Landmark Image Collections

Rahul Raguram
rraguram@cs.unc.edu
Joseph Tighe
jtighe@cs.unc.edu
Jan-Michael Frahm
jmf@cs.unc.edu

Department of Computer Science
University of North Carolina
Chapel Hill, NC, USA.

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. Our approach is based on the intuition that validating numerous image pairs of the same geometric scene structures quickly reveals useful information about two aspects of the image collection: (a) the reliability 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, by simply leveraging the information revealed during the verification stage. The main result of this work is that this unsupervised "learning-as-you-go" approach significantly improves performance; our experiments 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, 19, 22]. 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 [8, 19]. 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 structure). This is a computationally expensive 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 determine, 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. For instance, even the carefully 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., for some datasets, 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 perform geometric verification has been to estimate the geometric relationship between pairs *independently*, which does not fully exploit the specific characteristics of the dataset being processed. Our main idea in this work is simple: as the geometric verification progresses, we learn information about the image collection, and subsequently use this learned information 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, we feed this information directly back into the verification pipeline. This approach, while simple, is also very effective; our results 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 se-

lect informative image features (or, alternatively, how to suppress uninformative features) [9, 11, 14, 21, 23]. Also closely related is the work of [13], which uses the output of geometric verification to learn a probabilistic similarity measure between visual words, thus linking together words that are likely to be related. While these ideas are very similar in spirit to ours, our goal is quite different – we aim to utilize this information in an *online* way. This is a distinguishing characteristic from current techniques that obtain this information via an offline, preprocessing step, or through a *post-hoc* phase that uses the output of structure-from-motion. In addition, in contrast to techniques such as [9, 11, 23], which operate at the *feature* level , our approach explores the weighting of *visual words*. In this respect, our approach is similar to [14], which uses offline training to identify a subset of the visual vocabulary containing information that is most useful for landmark identification. However, we do not require a dedicated learning stage or labeled training data.

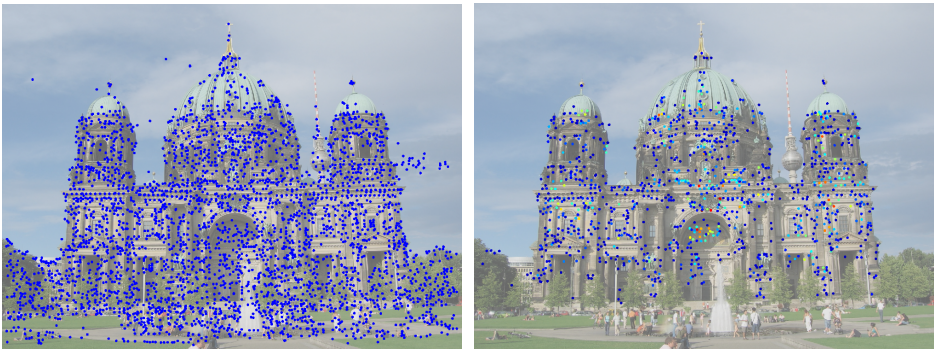
There has been some recent work related to the problem of identifying landmark images in large-scale image collections [10, 24, 25]. Our approach differs from these in that we do not require any manually labeled training imagery, as in [10]. In addition, we do not require images to have any associated geotags or GPS information, as required in [25], and we also do not require full structure-from-motion to be carried out on the entire dataset, as in [24]. Indeed, our method is in fact designed to *support* the structure from motion process and seamlessly integrate into it.

3 Efficient Large-scale Image Registration

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 [6], 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: our intuition is that repeated verification reveals useful information about both the validity of low-level image features (i.e., visual words), as well as the appearance of landmark images (i.e., bags of visual words).

3.1 Identifying useful visual words

As a motivating example, consider Figure 1(a), which shows all detected SIFT [12] 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 on vegetation, 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 (see Figure 1(b)). Note, in particular, that this weighting emphasizes visual words that are stable, reliable, and more likely to be geometrically consistent (for instance, those in the central portion of the structure), while also suppressing spurious visual words. Similar ideas have been explored to some extent in recent work [9, 23], but at the feature-level. Our approach extends 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).



(a) (b)

Figure 1: (a) All detected features for a single image (b) Features filtered based on the results of geometric verification – only visual words that were inliers in at least 10 previous image pairs are shown. The features in (b) are heatmap colour coded based on the inlier counts.

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 [17]. In addition, consider a set of visual word *priorities*, $C = \{c_1, c_2, ..., c_N\}$, where each c_i represents 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, we match SIFT features, and then run a robust estimator 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. In the case where the matched points are assigned to different words, we simply update the counts for both visual words. Intuitively, over time, we expect that these counts will help identify visual words that are frequently matched as inliers, as well as words that repeatedly fail the geometric consistency check. Note that the set C is maintained globally and is used across all image pairs.

3.1.2 Improving RANSAC sampling

One immediate application of the visual word priorities is in improving robust estimation. In recent years, a number of improvements to RANSAC have been proposed, each addressing a specific weakness of the original algorithm [2, 3, 15, 18, 20]. Most relevant to this work are

techniques that perform *non-uniform* sampling of the data points using some form of prior information. In our case, this prior comes from the computed visual word priorities.

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. Two recent estimators in this category are PROSAC [2], which uses ordering information to preferentially sample points based on their sorted rank and GroupSAC [15], which partitions points into groups based on similarity information. 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 decreases. Simply put, 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, 15, 18, 20]. In other words, given a set of matched features for an image pair, the ordering of matches is usually determined using some function of the SIFT matching score (e.g., 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. 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 corresponding visual words have been previously verified as being inliers. As we will show in the results in Section 4, 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 that operated at the level of individual visual words. We now show that it is possible to learn additional useful information that captures higher-level information. 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 present in the dataset. More specifically, consider the system of [6], which first clusters the images into groups based on (approximate) viewpoint, and then verifies these viewpoint clusters to obtain a set of iconic images. These iconic images represent a concise summary of the entire image collection (see [19]), and typically contain a diverse set of views of the various landmarks in the dataset. We observe that this information can then be used to train a classifier that distinguishes between landmark and non-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). How-

ever, 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.

We propose the following approach to efficiently increase the number of registered images: 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 leverage our same visual vocabulary (W), but now use it to build a standard bag of visual words (a histogram of the visual words) descriptor for each training image. We use this descriptor to train a linear support vector machine (SVM) classifier. Once trained, in the re-verification stage, we first 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 significantly reducing the overall compute time.

4 Results

4.1 Robust estimation

We first evaluate the effect of the modified ordering scheme based on visual words counts (3.1.2) on the robust estimation stage. We use a high-performance RANSAC variant called ARRSAC [18], which integrates PROSAC-style sampling into a real-time robust estimation framework. We report results on two experiments, representing different usage scenarios:

Experiment 1: We consider a dataset of 10000 images representing a single landmark (downloaded by doing a keyword search for "Berlin Dome" on Flickr). 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. We compare the performance of three estimators: (a) baseline RANSAC (denoted as **R1**) (b) ARRSAC with ordering based on SIFT matching scores (**R2**) and (c) ARRSAC with the proposed ordering based on visual words (**R3**). The only difference between **R2** and **R3** is in the ordering used to prioritize matches. In all cases, we estimate the epipolar geometry using the 7-point algorithm [7]. For this experiment, we use a visual vocabulary with $20,000$ words, computed by k-means clustering using a set of randomly chosen descriptors from the dataset.

Experiment 2: In this experiment, we process a much larger dataset, consisting of 2.77 million images of Berlin, obtained from [6]. To handle datasets of this magnitude, we use the clustering based approach described in [6], which is capable of scaling well to these larger datasets. We extract binarized gist features for each image, and then cluster using k-medoids with $k = 100000$ centers. We then perform geometric verification independently on each cluster, by first trying to identify a set of m (=3, in our experiments) consistent images in each cluster, denoting the image with the most inliers as the "iconic". We then register all remaining images in the cluster to this iconic. Compared to the approach used for Experi-

Table 1: Experiment 1 (BerlinDome-10k)

	R1	R2	R3
Mean time per image pair (ms)	389.6	41.2	28.9
Mean # of hypotheses per image pair	14218.4	604.1	399.7
Total runtime (hours:minutes)	23:48	04:08	03:16

Table 2: Experiment 2 (Berlin-2.77M)

	R2	R3
Mean time per image pair (ms)	26.7	18.8
Total runtime (hours:minutes)	02:50	02:28
# iconics	9841	9912
# registered images	132,719	133,830

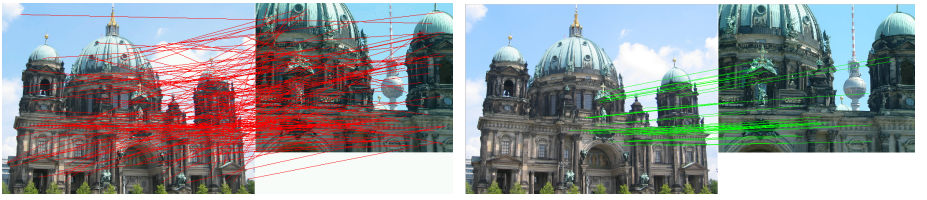
ment 1, this strategy dramatically reduces the total number of pairwise image verifications that need to be performed [6]. For this experiment, we compare the performance of **R2** and **R3**, since RANSAC (**R1**) is impractical for datasets of this size. In this case, we use a larger vocabulary with 10^6 visual words, computed using approximate k-means clustering [17].

In all cases, note that there is no dedicated "training" phase for **R3**. Initially, we start with all visual word priorities set to zero, and then progressively accumulate inlier counts. As the matching progresses, the efficiency of **R3** increases rapidly, and after verifying about 500 pairs on average, it becomes more efficient than the matching score based ordering. This is an empirical observation at this stage, and investigating this progression in more detail is an interesting direction for future work. In the results we report, for method **R3**, we start by using the matching score based ordering, and then switch over to the visual word based ordering after 500 image pairs. The results for the two experiments are shown in Tables 1 and 2. For the smaller BerlinDome-10k dataset, it can be seen that the new ordering scheme improves on the runtime of the matching score based ordering by about 30%. This provides strong evidence that using information accumulated during the matching process helps improve efficiency. For Expt. 1, we do not distribute our processing across multiple CPU/GPU cores; thus, the reported total runtime is for a single thread. For the large-scale Berlin-2.77M dataset, much the same trend holds for the per-image pair estimation time, which indicates that the weighting based on visual words is still reliable even with the presence of multiple landmarks in the dataset. It can be seen that the mean time to verify a single image pair is reduced compared to the results in Table 1; this is a consequence of the viewpoint-based clustering which increases the mean inlier ratio by grouping similar images in the same cluster. Finally, the overall runtime numbers reported in Table 2 are for a parallel implementation that distributes image pairs across 16 CPU threads for geometric verification. Note that the overall runtimes are not directly comparable between Tables 1 and 2, since these use two different strategies ([1, 6]) to perform the verification.

A specific example is shown in Figure 2(a).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 requires close to 1.07×10^6 samples, which is computationally prohibitive. ARRSAC with feature matching scores requires about 22000 samples, while ARRSAC with visual word ordering takes 595 samples, which represents a 36x improvement compared to ordering using matching scores.

4.2 Landmark classification

As noted earlier, one of the limitations of the approach in [6], is that a significant fraction of images are discarded following the clustering and verification stage. On the large scale Berlin dataset, observe from Table 2 that only $\approx 5\%$ of the entire dataset has been registered at this point. Thus, to increase completeness and coverage, we now wish to run geometric verification in order to attempt to register as much as possible of the remaining 95% of the



(a) (b)

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

Figure 3: A random sample of the training set derived from the first stage of verification.

dataset, against the iconic images that we have identified. This, however, is a very slow process, since most image pairs will fail verification (note that when performing robust estimation, failed verifications are much slower than successful verifications). Thus, to reduce the number of images that need to be verified, we use our linear SVM classifier (Section 3.2) to first weed out images that do not visually resemble landmark images.

To train our SVM classifier, we take positive and negative training images as described in Section 3.2, resulting in 9,471 positive and 10,529 negative training images. A sample of this training set can be seen in Figure 3. Since this training set is derived from the verification step, rather than a manual labeling, it does have a small fraction of non-landmark images in the positive set as well as landmark images in the negative set. What is interesting, however, is the trained classifier seems to be robust to this incorrect training data and is able to, at least partially, exclude non-landmark images from being incorrectly verified (refer Figure 4).

For each image we compute a bag-of-visual-words histogram using the 10^6 word vocabulary (Sec. 4.1). Since, on average there are $\approx 2,000$ visual words per image, and our histogram has one million dimensions, our feature vectors are 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 use the coordinate descent dual-based solver to solve the L2-regularized L2-loss SVM objective function and use five-fold cross validation on the training set to find the normalization constant (C), the only parameter for the SVM, by finding the parameter setting that gives the highest average classification rate (see [4] for details). We are able to cross-validate and train our SVM on 20,000 training images with one million dimensional feature vectors in 312 seconds. At test time, the classifier evaluation is a sparse inner product which, for our problem, amounts to on average $2,000$ addition operations. In our tests, evaluating the classifier on an image took less than 10^{-6} seconds, which is orders of magnitude faster than full geometric verification.

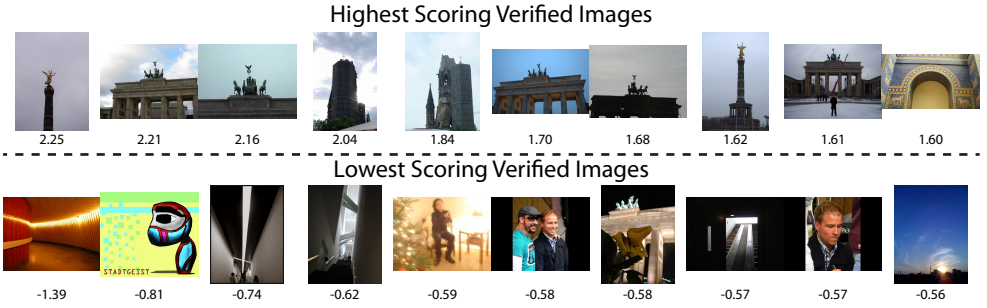


Figure 4: The top and bottom ten images (according to classifier scores) that geometrically verify to some iconic image. Notice that, though the lowest scoring images do verify, it is actually useful that the classifier rejected them, since these images are mostly not landmarks.

Table 3: Re-verification statistics (Berlin-2.77M)

	Full re-verification	SVM + re-verification
Number of newly registered images	273,639	202,538
Total runtime (days:hours:mins)	01:10:26	00:10:43

To measure the accuracy of our classifier, we create a test set of 44,289 images by running geometric verification on a random subset of the images that have yet to be verified. For positive test examples, we take images that are successfully registered to an iconic image and for negative examples, we choose images that do not verify to any iconic image. After this process we have 5,119 positive and 39,170 negative test images. We would like our classifier to have a number of favourable properties. First, we would like it to not miss too many images that would successfully verify against an iconic image. To measure this, we look at the true positive rate, which on our test set is 69.5%. While this might seem low at first glance, we have found that, while the images the classifier rejected do verify to some iconic, they are most often not "truly" landmark images (refer to Figure 4). This discrepancy arises from the fact that non-landmark images can be present in the set of iconics, if groups of geometrically consistent non-landmark photos exist in the dataset (these usually correspond to groups of near-duplicate images taken by a single user and often incorrectly tagged). As a second property of our classifier, we would like the total number of images for which our classifier has a positive response to, and for which we then run geometric verification on, to be low. Our classifier has a positive response to 25.9% of test images, allowing 74.1% of images to be rejected without verification, amounting to at least a 4x speed up.

Table 3 reports results for this approach on the Berlin-2.77M dataset. We start with 2,638,136 unregistered images, and perform re-verification. To do so, we compare two approaches: (a) full re-verification, where we retrieve 20 iconics for each unregistered image (using vocabulary-tree based image retrieval [16]), and then perform geometric verification; and (b) SVM + re-verification, where we first classify an unregistered image as being landmark/non-landmark, and then only verify the landmark images. It can be seen from Table 3 that incorporating the learned model of appearance into the re-verification stage reduces the overall runtime by $\approx 70\%$. There is a trade-off, however: doing so reduces the number of registered images by about 26%. Note that these numbers are in close agreement with those predicted from the SVM test set. In future work, we plan to investigate the impact of these missed images in the context of a complete 3D reconstruction.

5 Conclusion

In this paper, we have presented techniques for taking advantage of the information generated during geometric verification, to improve the overall efficiency of the process. We show that reliable statistics can be computed on both the visual-word level, as well as at an image-level. Our approach thus integrates online knowledge extraction seamlessly into structure-from-motion systems, and is particularly relevant for large-scale image collections. Our results demonstrate both improved efficiency, as well as higher image registration performance, potentially yielding more complete 3D models for these large-scale datasets.

6 Acknowledgments

This work was supported by NSF grant IIS-0916829, Department of Energy Award DE-FG52-08NA28778 , and the US Navy Spawar Center.

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] R. I. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, 2000.

[8] 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.

[9] Jan Knopp, Josef Sivic, and Tomas Pajdla. Avoiding confusing features in place recognition. In *European Conference on Computer vision*, pages 748–761, 2010.

[10] 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.

[11] Yunpeng Li, Noah Snavely, and Daniel P. Huttenlocher. Location recognition using prioritized feature matching. In *European Conference on Computer vision*, pages 791–804, 2010.

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

[13] Andrej Mikulík, Michal Perdoch, Ondřej Chum, and Jiří Matas. Learning a fine vocabulary. pages 1–14, 2010.

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

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

[16] David Nister and Henrik Stewenius. Scalable Recognition with a Vocabulary Tree. In *Computer Vision and Pattern Recognition*, pages 2161–2168, 2006.

[17] 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.

[18] 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.

[19] 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.

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

[21] T. Sattler, B. Leibe, and L. Kobbelt. Fast image-based localization using direct 2D-to-3D matching. In *International Conference on Computer Vision*, pages 667 –674, 2011.

[22] 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.

[23] 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.

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

[25] 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.