Hyperpoints and Fine Vocabularies for Large-Scale Location Recognition – Supplementary Material

Torsten Sattler[1], Michal Havlena[2], Filip Radenović[3], Konrad Schindler[2], Marc Pollefeys[1]
[1]Department of Computer Science, ETH Zürich, Switzerland
[2]Institute of Geodesy and Photogrammetry, ETH Zürich, Switzerland
[3]CMP, Faculty of Electrical Engineering, Czech Technical University in Prague
{sattlert,pomarc}@inf.ethz.ch,{havlena,schindler}@geod.baug.ethz.ch,radenfil@cmp.felk.cvut.cz

1. Overview

In this supplementary material, we present more details on the performance of our approach on the Landmarks dataset (Sec. 2) and a discussion of the memory requirements of our approach (Sec. 3). Both were not included in the paper due to space constraints.

Tab. 1 lists details about the datasets used for experimental evaluation.

2. Landmarks

Tab. 2 compares the performance of previous approaches and our method on the Landmarks dataset. We present detailed results from the paper for our approach, where we used only the nearest word for each query feature and estimated camera poses for the top-10 ranked images. The rest of the parameters were set the same as for the SF-0 experiment, that is employing all the proposed enhancements, constraining the camera orientation by 30°, and using the effective inlier rate to re-rank the poses. Following [1,2,4], we report the percentage of query images for which a pose with a certain number of inliers can be estimated instead of precision/recall pairs. As can be seen, our approach clearly outperforms the image retrieval approach from [2]. Similarly, our method can localize more query images than the model compression method from [1]. At the same time, our method also compresses the model w.r.t. [4]; [4] require 6.07 GB and 22.71 GB for their method when using mean and all point descriptors, respectively (*c.f.* Sec. 3). In contrast, our method requires 5.51 GB, out of which 2 GB are used to store the visual vocabulary. It is worth noting that storing the vocabulary is a constant overhead. Adding new points to the model does not require to add additional SIFT descriptors, which are the most memory consuming parts of the stored 3D points.

As evident from Tab. 2, our approach does not achieve the same localization performance as [4]. Compared to SF-0, the query images in the Landmarks dataset contain significantly more features (*c.f.* Tab. 1). As a result, more votes for cameras in the reconstruction are cast during image retrieval. Consequently, more cameras unrelated to the query can be found in the beginning of the ranked lists after image retrieval, often pushing related cameras out of the top-10 cameras that we use in the experiments. Using more top-ranked cameras or a better retrieval engine would enable our approach to come closer to the performance of [4]. Using more nearest neighboring words would in turn retrieve even more unrelated cameras and further deteriorate the result. Notice that [4] essentially solve the dataset and that our approach already outperforms the state-of-the-art retrieval approach from [2].

3. Memory Consumption

In this section, we compare the memory consumption of our method with the approach from [4] on SF-0. The memory consumption we report includes the memory required to store the descriptors, the 3D point positions, as well as indices required for the inverted files (our method) or to compute the co-occurrence probabilities used to generate RANSAC samples ([4]). We thereby assume that a 3D point is represented by 3 double values (24 bytes) and that each SIFT descriptor can be stored using 128 bytes by quantizing each descriptor entry (the cluster centers of our visual vocabulary are stored this way). Indices are stored using 32 bit (4 bytes) unsigned integers. We ignore the memory overhead for storing search structures as it is highly implementation dependent.

In the following, let P be the number of 3D points, N the number of descriptors contained in the model (which corresponds to the number of point observations in the model), M the number of SIFT descriptors used by a method, and C the number of cameras in a model.

Dataset	# Cameras	# 3D Points	# Descriptors	# Query Images	Mean # Features / Query Image
Landmarks [4]	205.16M	38.19M	177.82M	10k	8378.67
San Francisco (SF-0) [4]	610.77M	30.34M	149.30M	803	1860.42

Table 1: Details on the datasets used for experimental evaluation.

Method	Success Criterion	Localized Query Images (%)
Full model, full descriptors		
[4] (all descriptors, co-occurrence prior, bidirectional matching)	≥12 inliers	**∼99.00**
[4] (mean descriptors, co-occurrence prior, bidirectional matching)	≥12 inliers	**98.95**
Compressed model, full descriptors (mean descriptor per point)		
[1] (applying [4] with mean desc. on 0.37% of all 38M points)	≥12 inliers	45.90
[1] (applying [4] with mean desc. on 0.58% of all 38M points)	≥12 inliers	61.50
[1] (applying [4] with mean desc. on 0.81% of all 38M points)	≥12 inliers	71.87
[1] (applying [4] with mean desc. on 1.50% of all 38M points)	≥12 inliers	81.45
Image retrieval		
[2] (top-10 ranked cameras)	-	81.17
Ours: full model, quantized descriptors, image retrieval for disambiguation		
ours (1 nearest word, top-10 ranked cameras, SIFT)	≥12 inliers	92.09
ours (1 nearest word, top-10 ranked cameras, RootSIFT)	≥12 inliers	94.00

Table 2: Evaluation on the Landmarks dataset. We report the percentage of query images for which a pose that satisfies the success criterion can be estimated. The results for [4] were taken from Fig. 3 and Tab. 3 in [4]. Our method clearly outperforms the image retrieval-based approach from [2] and the method from [1] that uses a compressed model. Our method does not reach the same performance as [4]. The reason for this drop in performance is the large number of query features (*c.f*. Tab. 1), which result in casting many votes during image retrieval. Casting many votes results in polluted lists of top-ranked cameras such that many unrelated cameras are contained within the top-10. Considering more than 10 top-ranked cameras or using a better retrieval scheme will further boost the performance of our method.

[4] uses either all descriptors for every point or a mean descriptor for each point. In order to compute the co-occurrence prior they use to generate random samples inside RANSAC, [4] need to store the indices of all cameras observing it for each point. Consequently, [4] requires

$$128 \cdot M + 24 \cdot P + 4 \cdot N \ \text{ bytes.} \tag{1}$$

For each 3D point, our approach stores its 3D position, the list of cameras that observe this point, and the feature orientation of each such observation (which is required for the weak geometry filter from [3]). Each orientation is stored as a float using 4 bytes. Overall, $8 \cdot N$ bytes are required to store the indices and the orientations for all points. For each camera, we need to store its position in 3D to evaluate the advanced inlier measure (*c.f*. Eq. 8 in Sec. 4.5 of the paper). We also store a rotation value for each camera that denotes the rotation around the principal axis for use with the geometry filter. Thus, 28 bytes are required for each camera. In addition, each inverted file entry stores the indices of the 3D points mapped to the corresponding word, resulting in an additional $4 \cdot N$ bytes. In order to perform match expansion, we also store for each camera the indices of the points visible in it, requiring an additional $4 \cdot N$ bytes overall. In total, our approach uses

$$128 \cdot 2^{24} + 24 \cdot P + 16 \cdot N + 28 \cdot C \ \text{ bytes,} \tag{2}$$

where the first term denotes the memory required to store the visual vocabulary.

	SF-0	
Method	Memory (GB)	Recall for 95% precision
[4] (mean descriptors)	**4.85**	50.2
[4] (all descriptors)	19.03	54.2
ours	4.92	**59.1**

Table 3: Comparing memory consumption and localization performance of our approach and [4]. The ground truth from 2011 was used for both methods. As can be seen, our approach significantly outperforms [4] at a comparable or lower memory consumption.

Tab. 3 compares the memory consumption of our approach and the one from [4] on the SF-0 dataset. As can be seen, our approach significantly outperforms [4] when both methods use about the same amount of memory and still performs clearly better even when [4] uses much more memory.

References

[1] S. Cao and N. Snavely. Minimal Scene Descriptions from Structure from Motion Models. In *CVPR*, 2014.

[2] S. Cao and N. Snavely. Graph-Based Discriminative Learning for Location Recognition. *IJCV*, 112(2):239–254, 2015.

[3] H. Jegou, M. Douze, and C. Schmid. Hamming Embedding and Weak Geometric Consistency for Large Scale Image Search. In *ECCV*, 2008.

[4] Y. Li, N. Snavely, D. P. Huttenlocher, and P. Fua. Worldwide Pose Estimation Using 3D Point Clouds. In *ECCV*, 2012.