Supplementary Material
Efficient and Effective Cascade Correspondence Search for Large-scale Image-based Localization

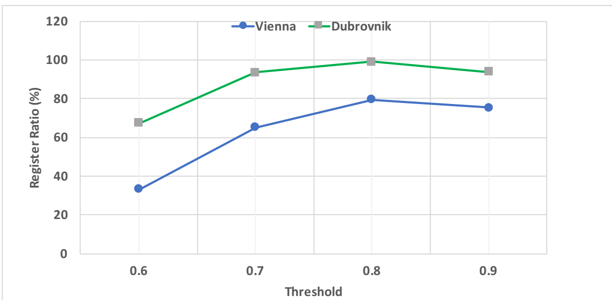

Figure 1: Studying the influence of test ratio thresholds on Dubrovnik and Vienna datasets. This experiment determines the good ratio threshold for precise search. Results show that threshold $\nu_h = 0.8$ achieves the highest registration rate. In the figure, the horizontal axis indicates the value of thresholds, and the vertical axis indicates the percentage of registered images.

References

Cao, S., and Snavely, N. 2013. Graph-based discriminative learning for location recognition. In *CVPR*.

Feng, Y.; Fan, L.; and Wu, Y. 2016. Fast localization in large-scale environments using supervised indexing of binary features. *IEEE Trans. Image Processing* 25(1):343–358.

Li, Y.; Snavely, N.; Huttenlocher, D.; and Fua, P. 2012. Worldwide pose estimation using 3d point clouds. In *ECCV*.

Li, Y.; Snavely, N.; and Huttenlocher, D. P. 2010. Location recognition using prioritized feature matching. In *ECCV*.

Sattler, T.; Weyand, T.; Leibe, B.; and Kobbelt, L. 2012. Image retrieval for image-based localization revisited. In *BMVC*.

Sattler, T.; Leibe, B.; and Kobbelt, L. 2011. Fast image-based localization using direct 2d-to-3d matching. In *ICCV*.

Sattler, T.; Leibe, B.; and Kobbelt, L. 2012. Improving image-based localization by active correspondence search. In *ECCV*.

Sattler, T.; Leibe, B.; and Kobbelt, L. 2016. Efficient effective prioritized matching for large-scale image-based localization. *TPAMI* (99):1–1.

Svarm, L.; Enqvist, O.; Oskarsson, M.; and Kahl, F. 2014. Accurate localization and pose estimation for large 3d models. In *CVPR*.

Zeisl, B.; Sattler, T.; and Pollefeys, M. 2015. Camera pose voting for large-scale image-based localization. In *ICCV*.

Dataset	# Cameras	# 3D Points	# Descriptors	# Query Images
Dubrovnik	6044	1,886,884	9,606,317	800
Rome	15,179	4,067,119	21,515,110	1000
Aachen	3047	1,540,786	7,281,501	369
Vienna	1324	1,123,028	4,854,056	266

Table 1: The information about four benchmark datasets we use in our experiments.

Method	#reg. images	Median	Quartiles [m]		#images with error		Time (s)
			1st Quartile	3st Quartile	< 18.3m	$>$400m	
Kd-tree	795	-	-	-	-	-	34.1*
(Li, Snavely, and Huttenlocher 2010)	753	9.3	7.5	13.4	655	-	
(Sattler, Leibe, and Kobbelt 2011)	782.0	1.3	0.5	5.1	675	13	2.32*
(Feng, Fan, and Wu 2016)	784.1	-	-	-	-	-	
(Sattler et al. 2012)	786	-	-	-	-	-	
(Sattler, Leibe, and Kobbelt 2012)	795.9	1.4	0.4	5.3	704	9	0.71*
(Sattler, Leibe, and Kobbelt 2016)	797	-	-	-	-	-	
(Cao and Snavely 2013)	796	-	-	-	-	-	
(Zeisl, Sattler, and Pollefeys 2015)	798	1.69	-	-	725	2	3.78^{+}
(Zeisl, Sattler, and Pollefeys 2015)**	794	**0.47**	-	-	749	13	-
(Svarm et al. 2014)	798	0.56	-	-	**771**	3	5.06^{+}
(Li et al. 2012)	**800**	-	-	-	-	-	
CCS	781	0.93	0.34	3.77	710	12	1.71
CCS (all)	791	0.85	0.34	3.34	716	9	11.03
$\mathbf{CCS_{R}}$	796	0.89	**0.31**	3.67	717	17	1.71
$\mathbf{CCS_{R}}$ (all)	798	0.87	0.34	**2.94**	734	11	11.03
$\mathbf{CCS_{RP}}$	794	1.06	0.39	4.15	711	10	**0.19**

Table 2: We compare our method to the state of the art on Dubrovnik dataset. Methods marked '*' are repeated on our machine, Methods marked '+' reports only the processing time of outlier rejection/voting scheme, taken from original papers (ignoring the execution time of 2D-3D matching). Methods marked '**' report results after bundle adjustment. We are also interested in our performance as using "all" descriptors of 3D point, we named them with "all" at the end. Using "all" descriptors improves the accuracy, e.g. can register more queries and more accurate. However, its execution time and memory requirement are not practical, thus we keep working with "mean" descriptors.

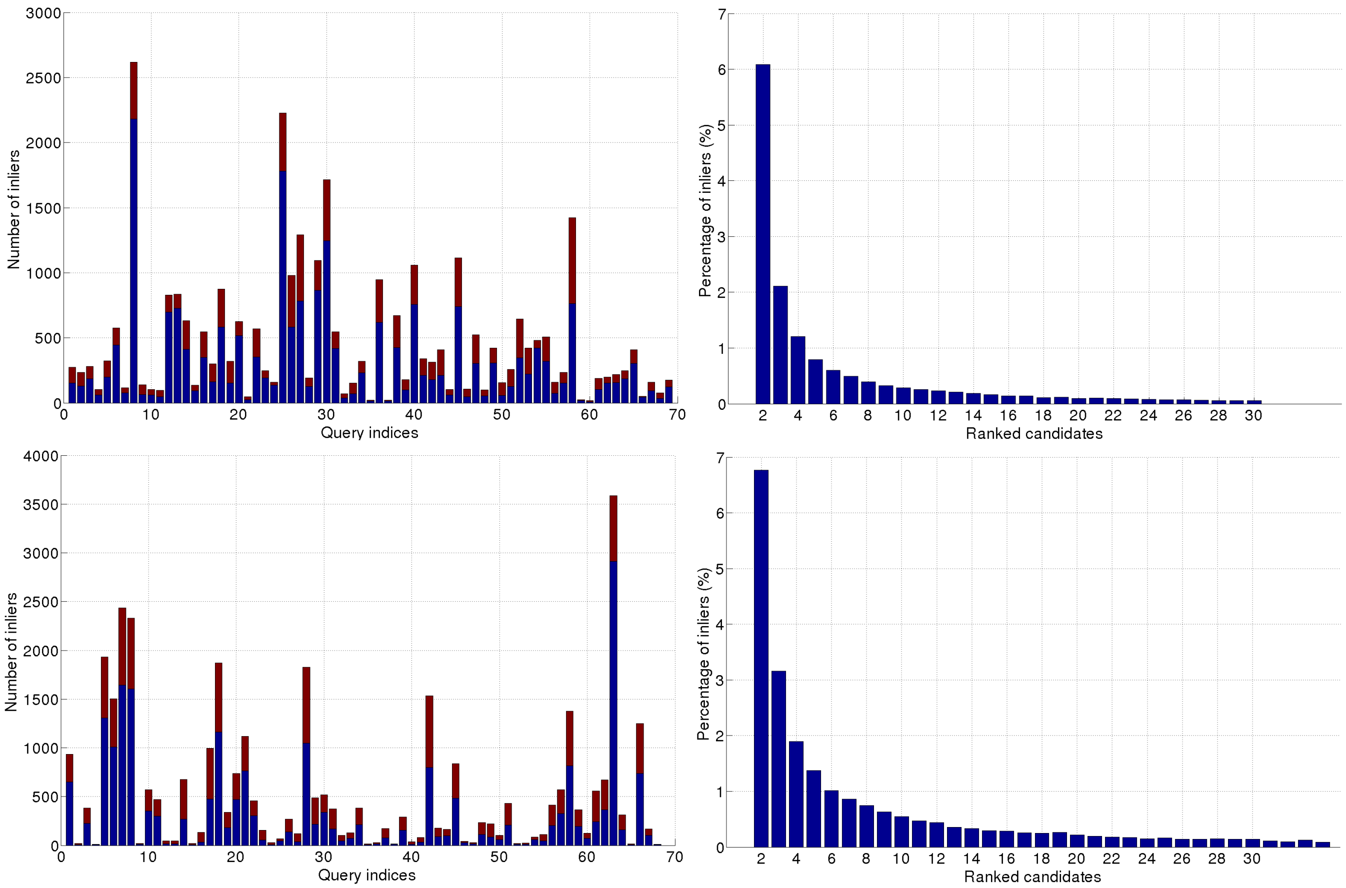

Figure 2: The number of inliers per query on Dubrovnik (first row) and Vienna (second row) datasets. Left figures shows the number of inliers found (on first 70 queries of Dubrovnik/Vienna) by threshold $\nu_h = 0.8$ (blue), and relaxed threshold $\nu = 0.9$ (red). Right figures are the percentage of number inliers contributed by candidates (from second order) in the list $\mathbf{L}_R$. On Vienna dataset, We increase approximately 100% of inliers as using relaxed threshold, and contribute about nearly 48% to total of number of inliers. The candidate list on Vienna dataset contributes slightly higher number of inliers than on Dubrovnik dataset. These explains why our method achieve better results on Vienna dataset. Fig. 3 shows inliers on one query example of Dubrovnik.

Figure 3: Left figure has 160 inliers found by CCS (1-1 matchings on threshold ν_h), and right figure has 278 inliers found by $\mathrm{CCS_R}$ (1-M matchings on relax threshold ν).