A. Supplementary Appendix for "Learning to Find Good Correspondences"

A.1. Dataset details

Here we detail the sequences used for training and testing in Table A.1. The image numbers reported for SUN3D [34] are after subsampling the video sequences by a factor of 10. For SUN3D we choose 9 sequences for training, and use the 15 sequences previously used by [31], collectively marked with ‡, only for testing. We generate disjoint training, validation and test subsets by splitting the images in each set in a 60-20-20 ratio, as explained in Section 4.1. For ‡ we take up to 500 images with a 0-0-100 ratio, as we do not train any model on them.

The last column assigns a label (a-u) to each set for convenience, which is used in Section A.3. Our best model is trained concatenating the datasets marked with ◊ and △.

Scene	Images	3D points	Avg. views	Label
Yahoo YFCC100M [28, 13]				
'Buckingham'	1676	152003	11.83	a
'Notre Dame'	3767	502017	35.41	b
'Sacre Coeur'	1179	152594	19.63	c
'Saint Peter's'	2506	235668	26.96	◊
'Reichstag'	75	19881	7.74	d
Multi-View Stereo [27]				
'Fountain'	11	–	–	e
'HerzJesu'	8	–	–	f
SUN3D [34] (training and validation)				
'Harvard 1' (harvard_conf_big/hv_conf_big_1)	455	–	–	–
'Harvard 2' (harvard_computer_lab/hv_c1_1)	543	–	–	–
'Harvard 3' (harvard_corridor_lounge/hv_lounge_corridor2_1)	540	–	–	–
'Harvard 4' (harvard_corridor_lounge/hv_lounge_corridor3_whole_floor)	629	–	–	–
'Brown 1' (brown_bm_3/brown_bm_3)	841	–	–	△
'Brown 2' (brown_cs_4/brown_cs4)	877	–	–	–
'Hotel 1' (hotel_ucla_ant/hotel_room_ucla_scan1_2012_oct_05)	1305	–	–	–
'Hotel 2' (hotel_pedraza/hotel_room_pedraza_2012_nov_25)	1065	–	–	–
'Home' (home_pt/home_pt_scan1_2012_oct_19)	2407	–	–	–
SUN3D [34] (test only, chosen by [31]) (‡)				
brown_cogsci_2/brown_cogsci_2	259	–	–	g
brown_cogsci_6/brown_cogsci_6	500	–	–	h
brown_cogsci_8/brown_cogsci_8	126	–	–	i
brown_cs_3/brown_cs3	340	–	–	j
brown_cs_7/brown_cs7	251	–	–	k
hotel_florence_jx/florence_hotel_stair_room_all	500	–	–	l
harvard_c4/hv_c4_1	224	–	–	m
harvard_c10/hv_c10_2	81	–	–	n
harvard_corridor_lounge/hv_lounge1_2	154	–	–	o
harvard_robotics_lab/hv_s1_2	159	–	–	p
mit_32_g725/g725_1	377	–	–	q
mit_46_6conf/bcs_floor6_conf_1	327	–	–	r
mit_46_6lounge/bcs_floor6_long	500	–	–	s
mit_w85g/g_0	387	–	–	t
mit_w85h/h2_1	500	–	–	u

Table 1. Datasets.

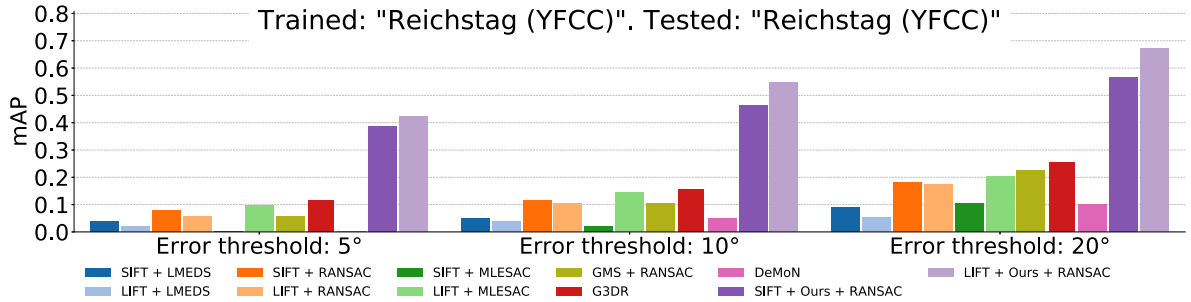

Figure 9. Results for the model trained and tested on 'Reichstag'.

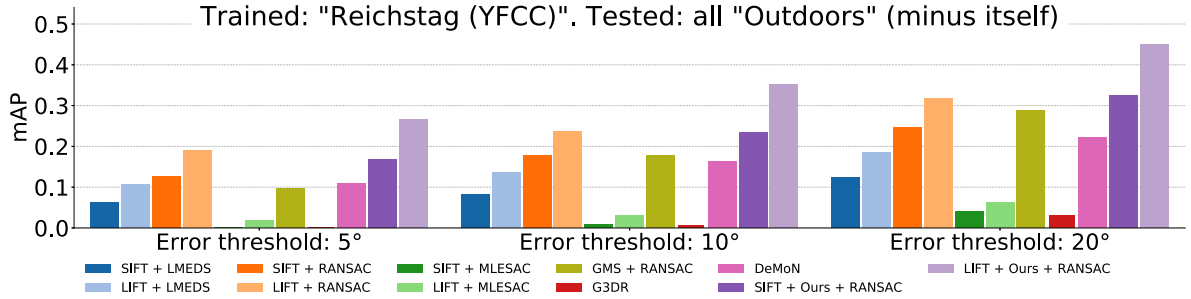

Figure 10. Results for the model trained on 'Reichstag' (d), and tested on every other 'Outdoors' sequence, *i.e.*, a-c, e, f and ◊. We average the results over each sequence.

A.2. Training with limited data

Due to space constraints, the paper only reports results with our best model, which is the concatenation of 'St. Peter's' (◊) and 'Brown 1' (△). Here we replicate the experiments of Section 4.5.1 and Section 4.5.2, *i.e.*, we train a model and evaluate it first on the same sequence and then on every other 'Outdoors' sequence, respectively, but now using only our *smallest* training sequence. The dataset in question is 'Reichstag' (d) from the 'Outdoors' subset, which contains only 59 images for training, 8 for validation and 8 for testing. Note that after accounting for visibility constraints, this still lets us extract over 1500 image pairs for training and about 35 for each validation and testing.

Fig. 9 shows results training and testing on (different subsets of) the same sequence, and Fig. 10 shows how the model generalizes over every 'Outdoors' sequence other than itself, *i.e.*, a-c, e, f, and ◊. We follow the same protocols as in Section 4.5.1 and Section 4.5.2. The best results are obtained with LIFT features, which is consistent with our previous observations. Our method outperforms all the baselines, with LIFT plus RANSAC and GMS plus RANSAC being the closest competitors. More importantly, when generalizing to other scenes with so little training data (Fig. 10) we still outperform GMS by 65%-200% relative at different error thresholds, and RANSAC by about 50% relative.

A.3. Per-sequence results

We could not show per-sequence results in the paper due to space constraints. Fig. 11 provides separate results for every testing sequence for our approach and for every baseline at multiple error thresholds. Again, we use our models trained on a single sequence from each data type, marked respectively with ◊ and △ (we do the same for G3DR [36]). The sequences used for testing include 'Outdoors' datasets a-f in Table A.1, which are averaged in the column marked $*$, and 'Indoors' datasets g-u, which are averaged in ‡. We provide numbers on top of the bars for $*$ and ‡. Our approach outperforms every baseline, with LIFT performing better than SIFT on the 'Outdoors' subset and the opposite for the 'Indoors' subset.

Note that DeMoN only achieves good performance for e and f in the 'Outdoors' sequences, and completely fails for YFCC100M sequences. G3DR shows even worse performance, hinting that sparse methods are preferable when it comes to photo-tourism datasets.

A.4. Ransac post-processing

As outlined in 4.4, RANSAC for post-processing allows us to greatly improve both the performance and the speed over RANSAC. In Table A.4 we provide results for the generalization experiments of Section 4.5.2, for stand-alone RANSAC,

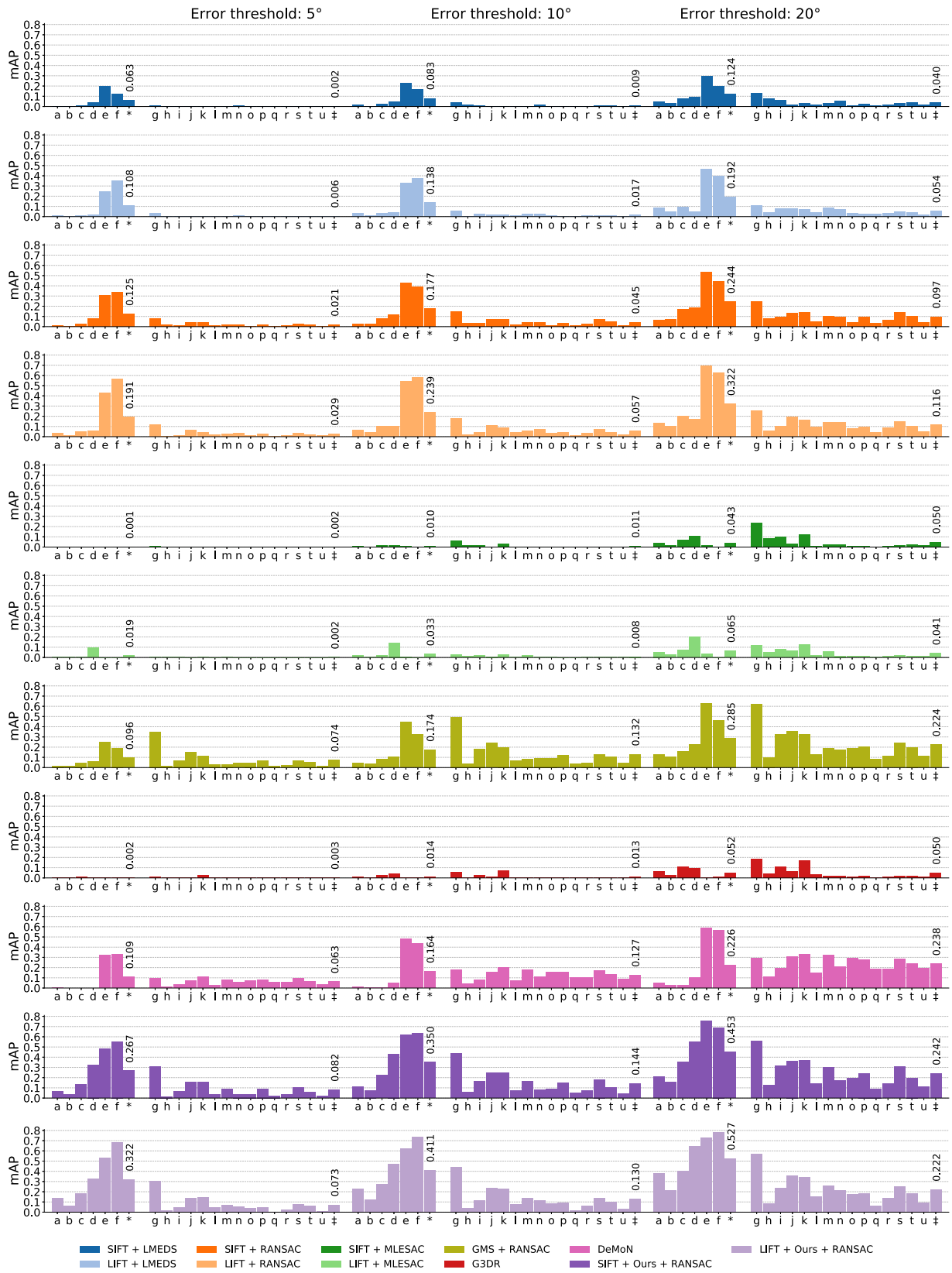

Figure 11. Results for every sequence in the 'Outdoors' subset (a-f) and the 'Indoors' subset (g-u). The entry labeled $*$ denotes the average performance over the 'Outdoors' subset, and ‡ the average performance over the 'Indoors' subset. Labels are listed in Table A.1.

and our method using either the 8-point algorithm or RANSAC for post-processing, which were not originally included in the paper due to spatial constraints. Note that this boost is only possible at test time, due to the differentiability requirement for training.

		Outdoors		Indoors		Average
		SIFT	LIFT	SIFT	LIFT	
RANSAC	mAP@20^o	0.221	0.291	0.097	0.115	—
Ours + 8-point	mAP@20^o	0.264	0.343	0.148	0.143	—
	w.r.t. RANSAC	+19.5%	+17.9%	+52.6%	+24.3%	+28.6%
Ours + RANSAC	mAP@20^o	0.462	0.530	0.242	0.222	—
	w.r.t. RANSAC	+109.0%	+82.1%	+149.5%	+93.0%	+108.4%

Table 2. RANSAC vs Ours with 8-point vs Ours with RANSAC. Both SIFT and LIFT use 2k keypoints.

A.5. Differentiating through eigendecomposition

To differentiate through the eigendecomposition, we rely on the TensorFlow implementation. Here, we provide a short definition for completeness. For more details, we refer the interested readers to [15]. In [15], it is shown that for a matrix $\mathbf{X}$, which can be decomposed into $\mathbf{X} = \mathbf{U}\mathbf{\Sigma}\mathbf{U}^\top$, where $\mathbf{U}$ is the matrix of eigenvectors and $\mathbf{\Sigma}$ is a diagonal matrix with eigenvalues, the derivative w.r.t. the eigenvectors are

$$d\mathbf{U} = 2\mathbf{U}\left(\mathbf{K} \odot (\mathbf{U}^\top d\mathbf{X}\mathbf{U})_{sym}\right) \ , \tag{10}$$

where $\mathbf{M}_{sym} = \frac{1}{2}(\mathbf{M}^\top + \mathbf{M})$, and

$$\mathbf{K}_{ij} = \begin{cases} \frac{1}{\sigma_i - \sigma_j}, & i \neq j \\ 0, & i = j \end{cases} \ , \tag{11}$$

and σ_i is the i-th eigenvalue.