Supplementary Material

Our theoretical results are proved and further explained in [28]. Additional numerical results and discussion are provided in §A. We refer to references, equation numbers and sections from the main text. We also follow the same numbering of figures, tables and equations, and thus start, for example, with Figure 3.

A. Supplementary Details for the Numerical Experiments

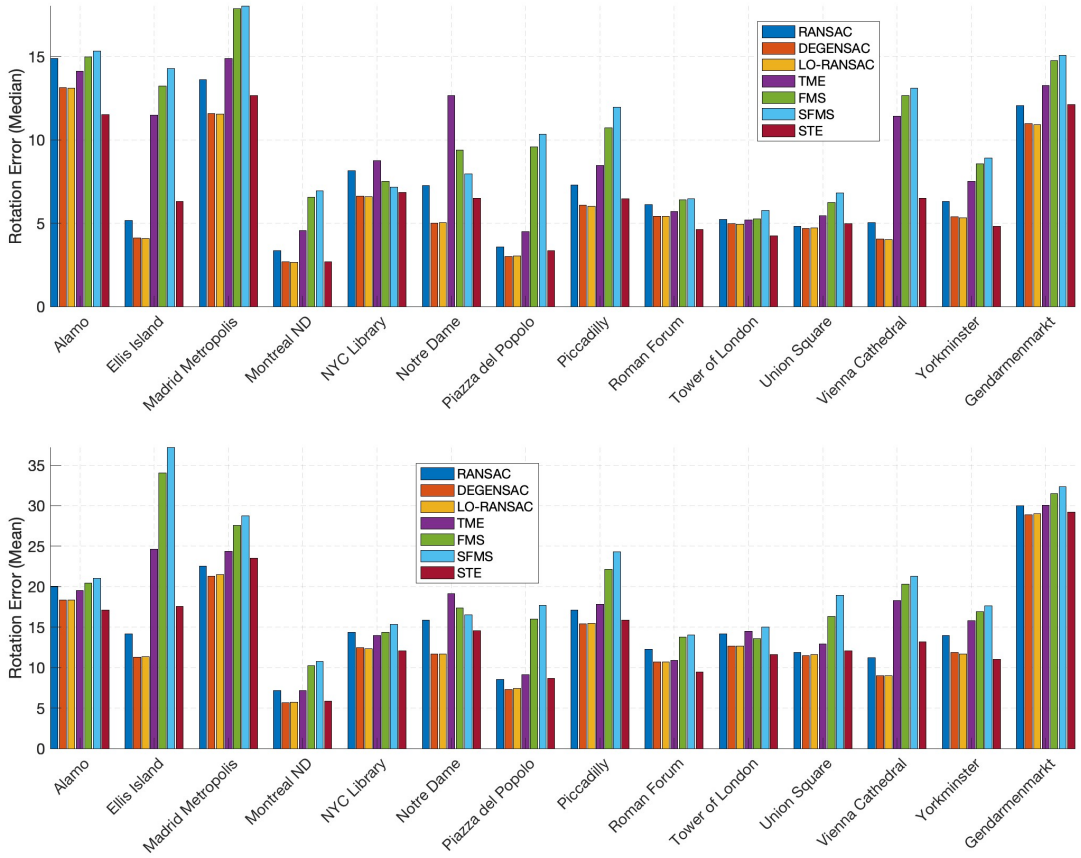

Figure 3. Median (top) and mean (bottom) relative rotation errors (in degrees) obtained by seven algorithms for the 14 datasets of Photo Tourism. The numerical results are reported in Table 1

We provide additional numerical results and details for our two different types of applications in §A.1 and §A.2.

A.1. Supplementary Details for Fundamental Matrix Estimation

We review some details about data normalization applied for RSR methods in §A.1.1 and the implementation of the RSR methods in §A.1.2. We provide additional numerical results in §A.1.3, and runtime comparison of only the RSR algorithms (including RANSAC) in §A.1.4.

A.1.1 Details of Data Normalization

Fundamental matrix estimation requires proper normalization of the input data [17]. We chose to use a normalization that is very similar to the commonly used one in the 8-point algorithm; nevertheless, we apply the normalization to the full dataset as our method is applied to the full dataset. We notice that there are even slightly better normalization techniques, however, to make sure that our competitive performance is due to the proposed method and not the normalization itself, we use the direct generalization of the standard method.

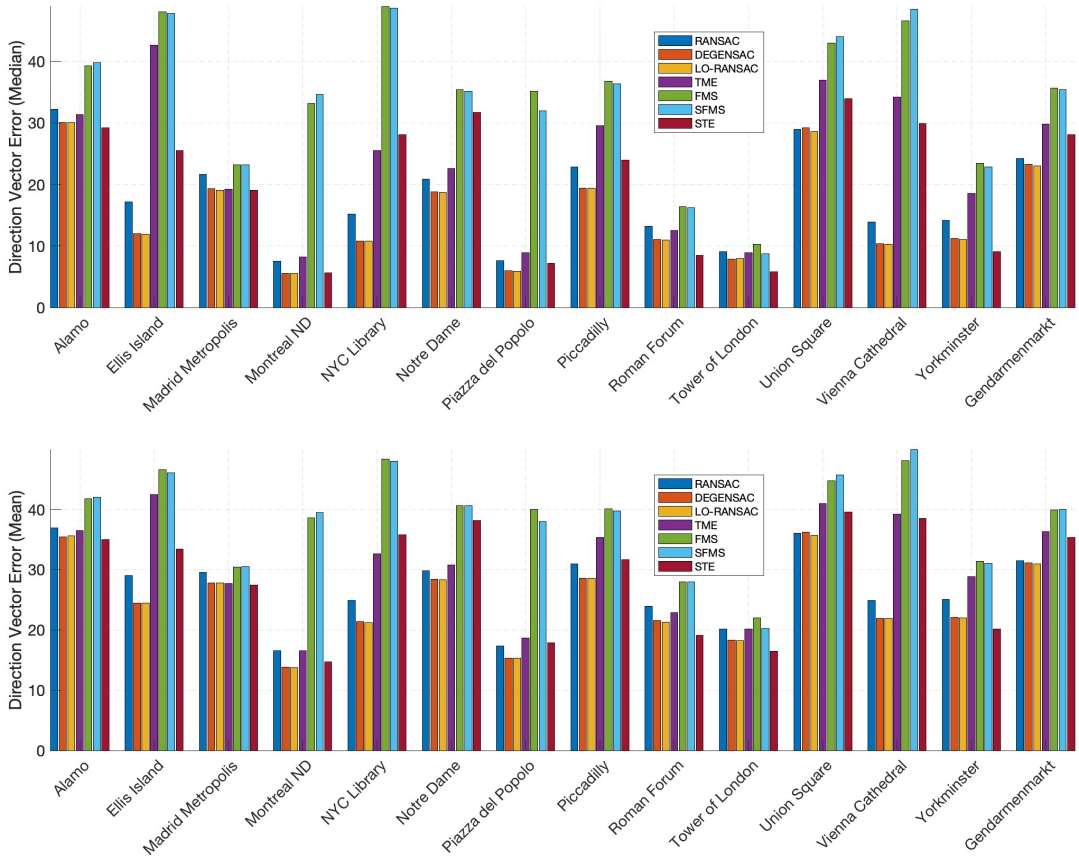

Figure 4. Median (top) and mean (bottom) errors of direction vectors (in degrees) obtained by seven algorithms for the 14 datasets of Photo Tourism. The numerical results are reported in Table 2

Each data point is represented by a column vector of homogeneous coordinates, $\boldsymbol{x} \in \mathbb{R}^3 = (x_1, x_2, 1)^\top$. For the N data points, $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N$, we form the data matrices $\mathbf{X} = [\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N] \in \mathbb{R}^{3\times N}$. Let (μ_1, σ_1) and (μ_2, σ_2) be the means and standard deviations of the first and the second rows of $\mathbf{X}$, respectively. We normalize the points as follows:

$$\hat{\boldsymbol{x}}_i = \mathbf{T}\boldsymbol{x}_i,\ 1 \le i \le N,$$

where the normalizing transformation $\mathbf{T} \in \mathbb{R}^{3\times 3}$ is given by

$$\mathbf{T} = \begin{bmatrix} \sigma_1^{-1} & 0 & 0 \\ 0 & \sigma_2^{-1} & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & -\mu_1 \\ 0 & 1 & -\mu_2 \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} \sigma_1^{-1} & 0 & -\sigma_1^{-1}\mu_1 \\ 0 & \sigma_2^{-1} & -\sigma_2^{-1}\mu_2 \\ 0 & 0 & 1 \end{bmatrix}.$$

This normalization is applied to any RSR method that uses all data points at once to form the subspace. In our experiments, these methods include FMS, SFMS, TME and STE. For RANSAC and its variants (which compute subspaces for 8 points at a time) we use the common normalization [17], which is exactly the normalization above, but applied to the 8 points chosen each time.

A.1.2 Review of the Implementation of the RSR Algorithms

We apply each RSR method (TME, FMS, SFMS and STE) as follows. Given two normalized data matrices (see Section A.1.1) $\hat{\mathbf{X}} = [\hat{\boldsymbol{x}}_1, \ldots, \hat{\boldsymbol{x}}_N] \in \mathbb{R}^{3\times N}$ and $\hat{\mathbf{X}}' = [\hat{\boldsymbol{x}}'_1, \ldots, \hat{\boldsymbol{x}}'_N] \in \mathbb{R}^{3\times N}$ whose columns are normalized feature points for the two given images, we form the data matrix $\tilde{\mathbf{X}} = [\tilde{\boldsymbol{x}}_1, \ldots, \tilde{\boldsymbol{x}}_N] \in \mathbb{R}^{9\times N}$ with the following columns: $\tilde{\boldsymbol{x}}_i = \mathrm{vec}(\hat{\boldsymbol{x}}_i \hat{\boldsymbol{x}}_i'^\top)$, $i = 1, \ldots, N$. We apply each RSR method with $d = 8$ and $D = 9$ (that is, we aim to recover an 8-subspace in $\mathbb{R}^9$) to $\tilde{\mathbf{X}}$. We find the orthogonal vector to this 8-dimensional subspace in $\mathbb{R}^9$ and reshape it in into a 3 by 3 matrix $\tilde{\mathbf{F}}$. In order to report a proper fundamental matrix with rank

2, we replace its lowest singular value with 0. Finally, we obtain the estimated fundamental matrix for the original data (before normalization) as follows: $\mathbf{T}^\top \hat{\mathbf{F}} \mathbf{T}'$, where $\mathbf{T}$ and $\mathbf{T}'$ are the normalization transformations defined in Appendix A.1.1.

A.1.3 Additional Numerical Results for Fundamental Matrix Estimation

We assess the quality of the fundamental matrix estimation by the median and mean errors of relative rotation and direction vectors directly obtained by the fundamental matrices for the various methods. We recall that these methods include STE (our proposed algorithm), TME [47, 50], FMS [24], SFMS [24], vanilla RANSAC [10], DEGENSAC [7] and LO-RANSAC [6]. We first explain how we compute the above errors for any of these methods.

For any two cameras, i and j, We estimate the fundamental matrix $\widetilde{\mathbf{F}}_{ij}$ from the correspondence pairs of the images of these cameras and then extract from $\widetilde{\mathbf{F}}_{ij}$ the relative rotation, $\widetilde{\mathbf{R}}_{ij}$, and the direction vector (that is, relative translation normalized to have norm 1) in the coordinates of camera i, $\widetilde{\boldsymbol{t}}_{ij}$ [17]. We remark that we need to normalize the relative translation to obtain direction vectors due to the scale ambiguity of the fundamental matrix. For cameras i and j, the relative rotation error compares the estimated relative rotation, $\widetilde{\mathbf{R}}_{ij}$, and the ground-truth one (provided by the Photo Tourism database), $\{\mathbf{R}^*_{ij}\}$, as follows:

$$e_R = \cos^{-1}\left(\frac{\mathrm{tr}\left(\mathbf{R}^{*\top}_{ij}\widetilde{\mathbf{R}}_{ij}\right)-1}{2}\right).$$

Note that the estimated direction vectors, $\{\widetilde{\boldsymbol{t}}_{ij}\}$, are defined up to a global orientation and in order to align them with the ground-truth information (given in the frame of camera i) obtained from the Photo Tourism data set, $\{\boldsymbol{t}^*_{ij}\}$, we find a rotation matrix $\mathbf{R}_{\text{align}}$, which minimizes $\sum_{1\le i,j\le N}\|\boldsymbol{t}^*_{ij}-\mathbf{R}_{\text{align}}\widetilde{\boldsymbol{t}}_{ij}\|^2_F$. For cameras i and j, the error of direction vectors is

$$e_T = \cos^{-1}(|\boldsymbol{t}^*_{ij}\cdot\mathbf{R}_{\text{align}}\widetilde{\boldsymbol{t}}_{ij}|).$$

We remark that the estimated direction vector, $\widetilde{\boldsymbol{t}}_{ij}$ in the frame of camera i implicitly requires estimating both the direction vector in world's coordinates and the absolute rotation matrix (up to global scale) that maps from world's coordinates to the frame of camera i, and thus seems to be susceptible to large errors.

Location	RANSAC		DEGENSAC		LO-RANSAC		TME		SFMS		FMS		STE	
	$\tilde{e}_R$	$\hat{e}_R$	$\tilde{e}_R$	$\hat{e}_R$	$\tilde{e}_R$	$\hat{e}_R$	$\tilde{e}_R$	$\hat{e}_R$	$\tilde{e}_R$	$\hat{e}_R$	$\tilde{e}_R$	$\hat{e}_R$	$\tilde{e}_R$	$\hat{e}_R$
Alamo	14.90	20.02	13.13	18.38	13.12	18.35	14.13	19.50	14.98	20.44	15.33	21.02	11.51	18.65
Ellis Island	5.17	14.17	4.10	11.26	4.09	11.38	11.48	24.64	13.22	34.06	14.27	37.16	6.71	17.57
Madrid Metropolis	13.61	22.52	11.57	21.28	11.56	21.48	14.89	24.38	17.89	27.58	18.05	28.78	12.11	21.63
Montreal N.D.	3.36	7.14	2.69	5.69	2.66	5.72	4.55	7.18	6.56	10.22	6.94	10.74	2.72	5.80
NYC Library	8.16	14.34	6.64	12.49	6.60	12.33	8.76	13.97	7.50	14.35	7.18	15.35	6.99	12.25
Notre Dame	7.25	15.84	4.99	11.67	5.03	11.68	12.67	19.16	9.40	17.36	7.96	16.54	6.86	15.11
Piazza del Popolo	3.59	8.54	3.00	7.32	3.02	7.45	4.51	9.10	9.59	15.98	10.35	17.69	3.28	8.21
Piccadilly	7.30	17.11	6.08	15.42	6.03	15.47	8.47	17.85	10.73	22.12	11.95	24.29	6.81	16.38
Roman Forum	6.11	12.29	5.42	10.69	5.41	10.68	5.69	10.90	6.41	13.75	6.46	14.05	4.55	9.60
Tower of London	5.24	14.16	4.96	12.69	4.94	12.64	5.20	14.47	5.27	13.55	5.77	15.02	4.18	12.03
Union Square	4.82	11.89	4.70	11.46	4.73	11.60	5.44	12.95	6.24	16.34	6.82	18.91	4.81	11.94
Vienna Cathedral	5.04	11.23	4.05	8.98	4.03	9.01	11.44	18.29	12.67	20.31	13.12	21.27	7.05	13.75
Yorkminster	6.31	13.97	5.39	11.85	5.33	11.70	7.52	15.83	8.55	16.93	8.92	17.63	4.87	11.75
Gendarmenmarkt	12.05	29.99	10.99	28.89	10.93	28.99	13.28	30.05	14.77	31.48	15.09	32.38	11.83	29.10

Table 1. Median and mean relative rotation errors (in degrees) obtained by seven algorithms for the 14 datasets of Poto Tourism. $\tilde{e}_R$ is the median relative rotation error, $\hat{e}_R$ is the mean relative rotation error.

Figure 3 is a bar plot presenting the errors of relative rotations (in angles) for the different methods and all 14 datasets and Figure 4 showcases the errors of direction vectors. For completeness, we record the presented numerical values in Table 1 (for relative rotations) and Table 2 (for direction vectors). Figure 5 presents the mean Average Accuracy (mAA) [18] for rotation with threshold 10°, where as opposed to the former errors which need to be small, higher mAA is better.

We note that STE significantly outperforms FMS and SFMS. STE also outperforms TME, where the only cases where TME outperforms STE is for direction vector estimation for the NYC Library and Notre Dame datasets. In terms of relative rotation

Location	RANSAC		DEGENSAC		LO-RANSAC		TME		SFMS		FMS		STE	
	$\tilde{e}_T$	$\hat{e}_T$	$\tilde{e}_T$	$\hat{e}_T$	$\tilde{e}_T$	$\hat{e}_T$	$\tilde{e}_T$	$\hat{e}_T$	$\tilde{e}_T$	$\hat{e}_T$	$\tilde{e}_T$	$\hat{e}_T$	$\tilde{e}_T$	$\hat{e}_T$
Alamo	32.20	36.97	30.09	35.46	30.11	35.57	31.36	36.49	39.31	41.74	39.85	41.98	29.25	35.00
Ellis Island	17.16	29.04	11.98	24.43	11.90	24.45	42.62	42.42	48.11	46.62	47.79	46.02	25.48	33.44
Madrid Metropolis	21.67	29.56	19.36	27.83	19.09	27.82	19.23	27.71	23.22	30.47	23.18	30.52	19.04	27.47
Montreal N.D.	7.49	16.51	5.57	13.80	5.53	13.75	8.25	16.55	33.20	38.61	34.65	39.51	5.62	14.71
NYC Library	15.17	24.87	10.77	21.35	10.80	21.25	25.48	32.62	48.89	48.34	48.69	48.02	28.09	35.79
Notre Dame	20.91	29.83	18.82	28.40	18.75	28.30	22.61	30.74	35.38	40.63	35.16	40.60	31.68	38.12
Piazza del Popolo	7.58	17.35	5.95	15.31	5.93	15.34	8.88	18.70	35.17	39.99	31.98	37.96	7.16	17.83
Piccadilly	22.89	30.99	19.40	28.58	19.39	28.58	29.58	35.36	36.83	40.06	36.33	39.76	23.95	31.64
Roman Forum	13.24	23.96	11.07	21.52	10.97	21.30	12.51	22.86	16.42	27.96	16.24	27.95	8.52	19.13
Tower of London	9.10	20.18	7.90	18.30	7.95	18.20	8.88	20.15	10.27	22.00	8.77	20.21	5.80	16.46
Union Square	28.97	36.03	29.23	36.21	28.60	35.73	36.98	40.98	42.97	44.76	44.02	45.71	33.93	39.60
Vienna Cathedral	13.88	24.94	10.33	21.89	10.26	21.89	34.24	39.24	46.63	48.11	48.46	49.88	29.88	38.53
Yorkminster	14.17	25.11	11.20	22.05	11.05	21.99	18.54	28.87	23.48	31.42	22.82	31.03	9.09	20.17
Gendarmenmarkt	24.25	31.51	23.32	31.11	23.03	30.96	29.82	36.33	35.70	39.88	35.44	40.00	28.09	35.37

Table 2. Median and mean errors of direction vectors (in degrees) obtained by seven algorithms for the 14 datasets of Photo Tourism. $\tilde{e}_T$ is the corresponding median error, $\hat{e}_T$ is the corresponding mean error.

estimation, STE outperforms vanilla RANSAC, where Ellis Island and Vienna Cathedral are the only two datasets where RANSAC outperforms STE. In terms of direction vector estimation, STE outperforms vanilla RANSAC in 7 out of the 14 datasets and in the rest of them vanilla RANSAC outperforms STE. Anyway, the errors of direction vectors are higher than relative rotation errors because of an issue discussed above and many references do not present these errors. As shown in Figure 5, STE has a higher mAA(10°) than other RSR methods. Overall, STE is slightly better than RANSAC. Indeed, the averaged mAA(10°) for STE is 0.44 for STE and 0.41 for RANSAC. However, RANSAC has a noticeable higher mAA (10°) for the following 3 datasets: Ellis Island, Notre Dame and Vienna Cathedral. On the other hand, DEGENSAC and LO-RANSAC are overall slightly better than STE (with averaged mAA(10°) 0.45 for both methods), but STE has a noticeably higher mAA (10°) for the following 5 datasets: Alamo, Montreal Notre Dame, Roman Forum, Tower of London and York Minster.

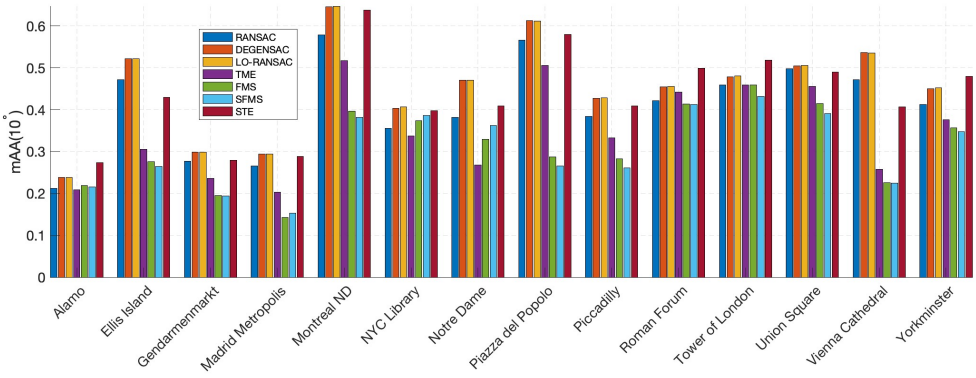

Figure 5. mAA(10°) obtained by seven algorithms for the 14 datasets of Photo Tourism.

A.1.4 Runtime Comparison of the RSR components

We compare the elapsed time of the different RSR algorithms, including RANSAC and its variants. Since our current implementation of STE uses Algorithm 2 with $m=5$ values of γ ($\gamma \in \{(2i)^{-1}\}_{i=1}^{5}$), it is slower than the basic implementation of Algorithm 1 by a factor 5 . This choice keeps the similar accuracy of STE with more values of γ but reduces the overall runtime.

We compared the runtime of STE (with $m=5$) with RANSAC, TME, FMS and SFMS for the 14 Photo Tourism datasets (using Matlab implementations). These experiments were carried out on an Apple Silicon M2. For each dataset, the average runtime among all image pairs is reported in Table 3. Overall, the runtimes of STE and RANSAC are somewhat comparable, where the average

over all datasets of the runtime of RANSAC is 22.91 msec and of STE is 21.91 msec. However, the variance of the runtime of RANSAC is rather large, unlike STE. We further note that TME was the fastest method, however running STE with $m=1$, instead of $m=5$, is slightly faster than TME. FMS and SFMS are slower than TME and overall faster than STE (with $m=5$) and RANSAC.

Locations	Algorithm Runtimes (ms)				
	STE	TME	SFMS	FMS	RANSAC
Alamo	19.2	4.0	13.7	13.2	11.8
Ellis Island	19.8	4.8	9.8	9.3	44.6
Madrid Metropolis	18.6	3.9	9.3	8.8	11.3
Montreal N.D.	18.0	3.6	18.5	17.8	31.6
NYC Library	21.1	4.5	22.5	22.0	51.7
Notre Dame	19.4	4.7	18.5	17.9	15.5
Piazza del Popolo	19.1	4.2	9.3	8.8	22.7
Piccadilly	19.0	4.4	8.8	8.2	24.0
Roman Forum	18.1	3.7	16.0	15.5	7.3
Tower of London	17.6	3.7	15.4	14.8	4.5
Union Square	19.0	4.2	9.6	9.1	10.2
Vienna Cathedral	20.5	4.5	21.3	20.8	37.5
Yorkminster	17.4	3.6	15.1	14.5	18.2
Gendarmenmarkt	18.7	3.7	8.5	8.0	29.9

Table 3. Elapsed runtimes (in milliseconds) of STE, TME, SFMS, FMS and RANSAC for the 14 datasets of Photo Tourism. The runtimes are averaged over all image pairs in the dataset.

The runtime of RANSAC mainly depends on the fraction of the outliers. For higher fractions of oultiers, RANSAC significantly slowed down compared to other methods, showcasing exponential dependence of time on the fraction of outliers. Indeed, if ε denotes the fraction of inliers and d the subspace dimension and one requires 99% confidence, the expected number of iterations of RANSAC is given as $\log(1-.99)/\log(1-\varepsilon^d)$ [44].

To further investigate the expected correlation of the runtime and the fraction of outliers, we also considered artificial instances, which we repeatedly generated 100 times. We fixed $N=400$ and generated two data matrices $\mathbf{X}=[\boldsymbol{x}_1,...,\boldsymbol{x}_N]\in\mathbb{R}^{3\times N}$ and $\mathbf{X}'=[\boldsymbol{x}'_1,...,\boldsymbol{x}'_N]\in\mathbb{R}^{3\times N}$, whose columns contain inlier and outlier feature correspondence pairs in homogenous coordinates as follows. We varied the fraction of designated outliers, where a fixed fraction of columns of $\mathbf{X}$ and $\mathbf{X}'$ (with the same index) were designated as outliers and the rest as inliers. For generating an outlier correspondence pair, the first two coordinates of $\boldsymbol{x}_i$ and $\boldsymbol{x}'_i$ were uniformly sampled in $[0,1000]$ and the third coordinates was set to be 1. For an inlier correspondence pair, we followed the procedure of [19] to sample $(\boldsymbol{x}_i,\boldsymbol{x}'_i)$ satisfying the epipolar constraint.

Given the generated matrices $\mathbf{X}$ and $\mathbf{X}'$, we first normalized them according to the procedure described in §A.1.1 and then used the procedure described in §A.1.2 to estimate the fundamental matrix from these two matrices using any of the RSR methods with $d=8$ and $D=9$. The application of RANSAC is also similar to before. The relative rotation was extracted from the estimated fundamental matrix and its error was computed based on the given ground truth rotation for the inliers, which are clarified when following the details of generating them in [19]. Using these 100 samples we computed mAA(10°).

Figure 6 reports mAA(10°) and runtimes of STE (with $m=5$), TME, SFMS, FMS and RANSAC for different fractions of outliers. The maximum number of iteration of RANSAC was set to be 1000. We note that STE (with $m=5$) achieved the highest mAA(10°) when the fraction of outliers was at most 50% and in this case the mAA(10°) values of FMS are slightly below those of STE. On the other hand, when the fraction of outliers is at least 60% the mAA(10°) values of STE are higher than STE, but they indicate low accuracy. The values of RANSAC, SFMS and TME are below the ones of STE. Moreover, the mAA(10°) values of TME are rather low and indicate low accuracy.

Regarding runtime, we first note that FMS, SFMS and TME have fixed times, since the number of data points, N, is fixed. Moreover, TME is faster then FMS and SFMS whose time is comparable. On the other hand, STE has a smaller run time for low fraction of outliers, since it converges faster then. Furthermore, the runtime of STE stabilizes when the fraction of outliers is large and it is slightly above FMS and SFMS. RANSAC is relatively fast for a small fraction of outliers, but it is much slower than all other method when the fraction of outliers is at least 50%.

The above mentioned formula of expected number of iterations implies that RANSAC can easily fail with large d. In the setting of 3-

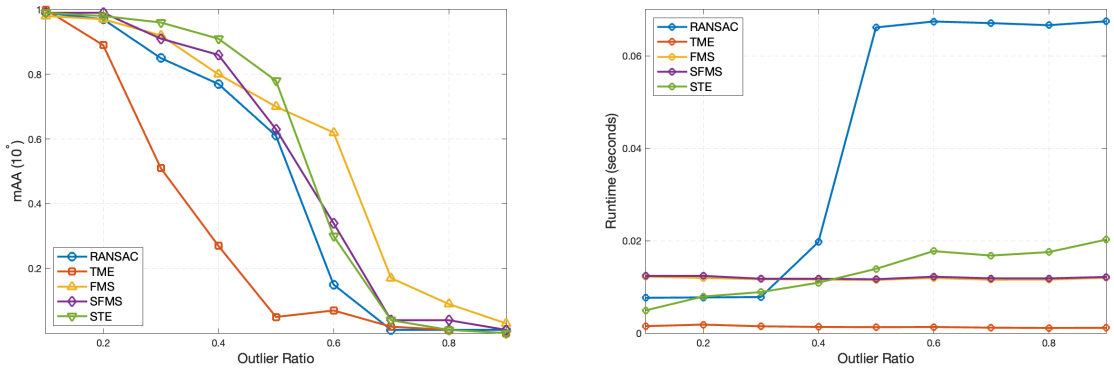

Figure 6. mAA(10°) (left) and runtime (right, in seconds) of five algorithms for the synthetic feature points data. The x-axis indicates the fraction of the outliers, where the total number of points is fixed as 400.

view homography estimation, $D=27$ and $d=26$. We thus tried the standard haystack model with $D=27$, $d=26$, $N=400$ and varied fractions of outliers. To measure error, we use the angle between L_*, the inlier subspace of the haystack model, and L, the estimated subspace. Figure 7 compares the errors and runtimes of the five algorithms for different fractions of outliers. We note that STE has the lowest error, but they notably increase for fraction of outliers larger than 50%. The errors of SFMS and TME are only slightly above than those of STE. RANSAC performed poorly, even for very small fractions of outliers and FMS is even worse than RANSAC.

We note that the runtimes of the RSR methods behave similarly to the ones in Figure 6. In each one of the experiments RANSAC needed the maximal number of iterations, which was set as 1000, and obtained the same runtime, which is significantly larger than the other runtimes.

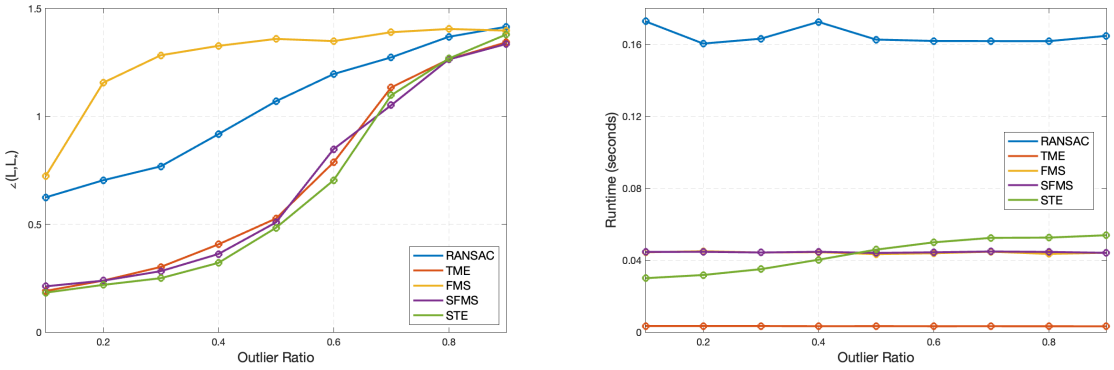

Figure 7. Angular error (left, in radius) and runtime (right, in seconds) of five algorithms for the generalized haystack synthetic data. The error is angle between L_* and the estimated subspace $\hat{L}$ recovered by the algorithm. The x-axis is the fraction of outliers.

A.2. Additional Numerical Results for Initial Camera Removal for SfM

We use all 14 datasets of the Photo Tourism database. As mentioned in the main text, scale factors are first obtained by the LUD algorithm. Many of the blocks $\mathbf{E}_{ij}$ are absent when there are no matching features between images i and j. On the other hand, the geometric relationship between cameras i and j still exists and the ground-truth essential matrix is thus well defined. In such scenarios, a common strategy is to assign to $\mathbf{E}_{ij}$ a zero matrix [39]. For completeness, we apply our proposed procedure with this strategy in Section A.2.1. However, the results should be more accurate when using matrix completion to fill the missing blocks of $\mathbf{E}$. We follow such a strategy here and provide its details as follows. Let Ω denote the set of indices, ij, of the blocks $\mathbf{E}_{ij}$ that can be directly obtained by the images. We denote by $\mathbf{M}\in\mathbb{R}^{3n\times 3n}$ the desired matrix with blocks $\{\lambda_{ij}\mathbf{E}_{ij}\,|\,(i,j)\in\Omega\}$ and additional

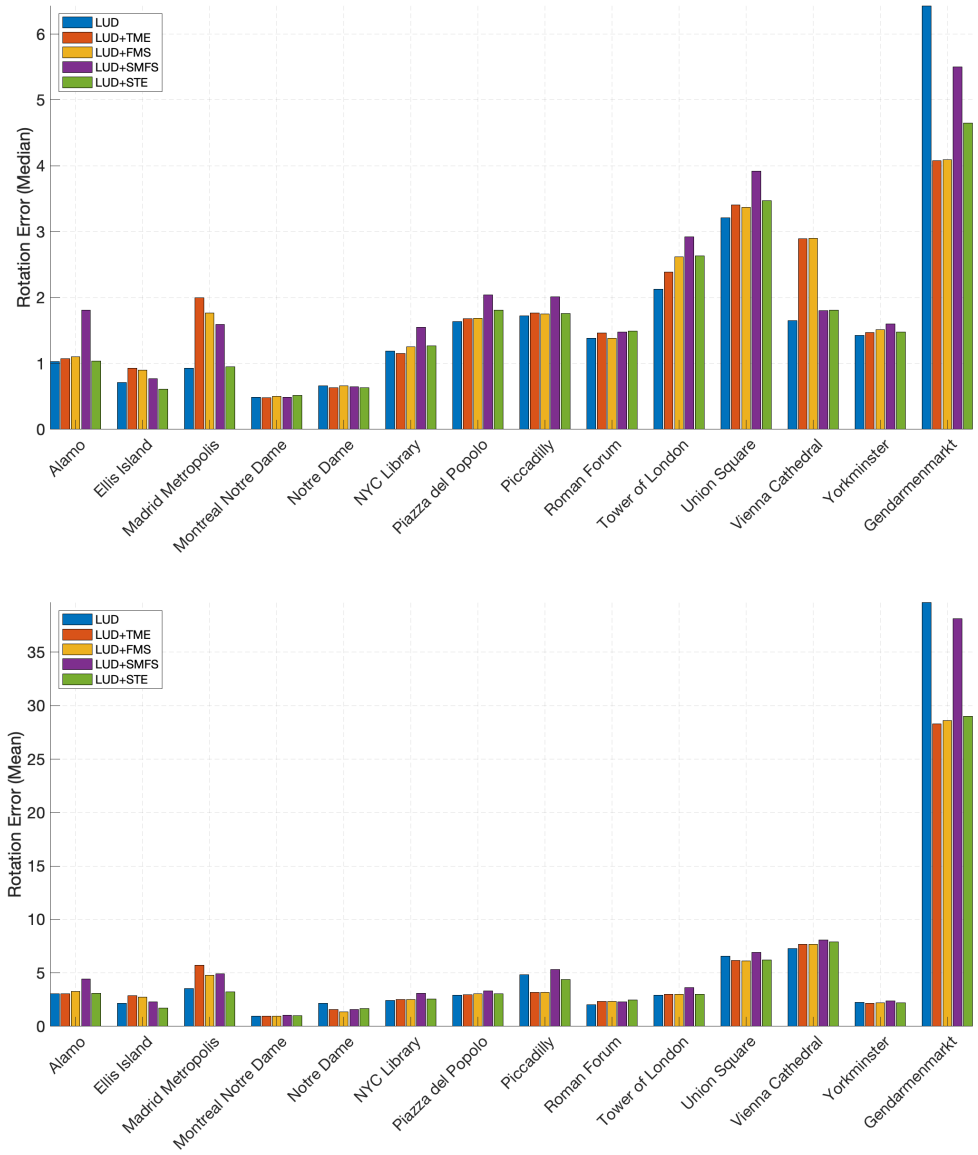

Figure 8. Median (top) and mean (bottom) absolute rotation errors (in degrees) of LUD and four RSR methods used to initially screen bad cameras within LUD applied to the 14 datasets of Photo Tourism. For Gendarmenmarkt, the median rotation error of SFMS was 31.04 degrees and mean error of SFMS was 62.89, which are above the figure range.

completed ones. We approximate $\mathbf{M}$ by solving the following relaxed optimization problem (see e.g., [3]):

$$\widehat{\mathbf{M}} = \underset{\mathbf{M}\in\mathbb{R}^{3n\times 3n}}{\operatorname{argmin}} \|\mathbf{M}\|_*, \quad \text{subject to } \mathbf{M}_{ij} = \lambda_{ij}\mathbf{E}_{ij}, \quad (i,j)\in\Omega. \tag{6}$$

The Singular Value Thresholding (SVT) algorithm [3] is applied to solve Equation (6). It is worth noting that for some datasets, the sampling ratio $|\Omega|/n^2$ is smaller than 10%. To prevent the SVT algorithm from diverging, we set a step size $\delta = n^2/(10|\Omega|)$, which is smaller than the suggested value in [3, Section 5.1.2]. Since the block $\mathbf{M}_{ij}$ is defined as an essential matrix, we further

project it on the essential matrix manifold. Specifically, we define $\widetilde{\mathbf{E}} \in \mathbb{R}^{3n \times 3n}$ as follows:

$$\widetilde{\mathbf{E}}_{ij} = \begin{cases} \lambda_{ij}\mathbf{E}_{ij}, & \text{if } (i,j) \in \Omega \\ \mathbf{P}_E(\widehat{\mathbf{M}}_{ij}), & \text{if } (i,j) \notin \Omega \\ \mathbf{0} & \text{if } i = j, \end{cases}$$

where for $\widehat{\mathbf{M}}_{ij} = \mathbf{U}\mathbf{S}\mathbf{V}^\top$, $\mathbf{P}_E(\widehat{\mathbf{M}}_{ij}) = \mathbf{U}\mathrm{diag}([1,1,0])\mathbf{V}^\top$ is the projector to the essential manifold. Following the procedures described in Section 4.2, we apply RSR with $d = 6$ by treating $\widetilde{\mathbf{E}}$ as a data matrix of $D = N = 3n$, recover a d-dimensional robust subspace and identify the outlying columns whose distance is largest from this subspace.

We compare mean and median errors of rotations and translations and runtime for the LUD pipeline and the LUD+RSR pipeline for the following four RSR algorithms: STE, FMS, SFMS, and TME. By LUD+RSR we refer to our proposed camera filtering process by the chosen RSR algorithm followed by LUD. Figure 8 is a bar plot of rotation errors (in degrees) for all methods and all 14 datasets, Figure 9 is a bar plot of translation errors (in meters), and Figure 10 is a bar plot of run times. As explained in the main text, we only aim to compare the effect of the camera screening on the overall time (and this screening may potentially increase time due to issues with parallel rigidity), but it is not the actual time as LUD+RSR uses information for scaling factors obtained by LUD. For completeness, Table 4 summarizes the results of LUD, LUD+STE, and LUD+SFMS, whereas Table 5 presents the results of LUD+TME and LUD+FMS.

Overall, STE yields better accuracy than other RSR methods. We observe that LUD+STE generally improves the estimation of camera parameters (both rotations and translations) over LUD, though the median rotation errors are comparable. The improvement of LUD+STE is significant for both the Roman Forum and Gendarmenmarkt. In terms of runtime, both LUD+STE and LUD+SFMS demonstrate improvements, where LUD+SFMS is even faster than LUD+STE. While this does not yet imply faster handling of the datasets (as we use initial scaling factors obtained by LUD), it indicates some efficiency in the removal of outliers.

Table 6 reports results for Gendarmenmarkt when removing 45% of outlying columns. We note that STE significantly reduces the pipeline's runtime and improves the accuracy for both rotations and translations. TME also exhibits improvement, but it is less accurate and is also slower than STE. On the other hand, FMS and SFMS do not improve the accuracy. While the resulting errors are still large, their improvement shows some potential in dealing with difficult SfM structure by initially removing cameras in a way that may help eliminate some scene ambiguities, which are prevalent in Gendarmenmarkt.

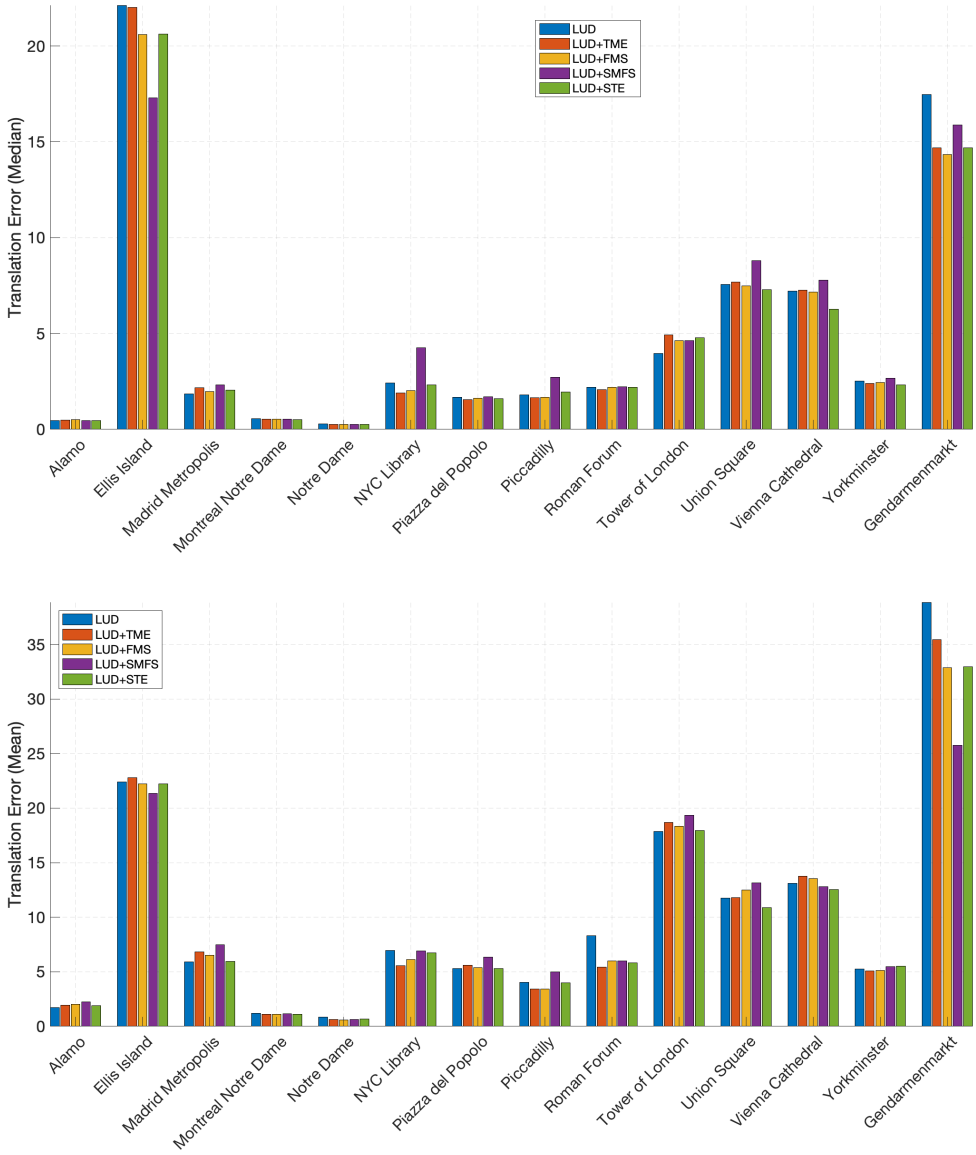

Figure 9. Median (top) and mean (bottom) absolute translation errors (in meters) of LUD and four RSR methods used to initially screen bad cameras within LUD applied to the 14 datasets of Photo Tourism.

Finally, in order to get an idea about the behavior of the outliers, Figure 11 plots the distances of all data points to the STE subspace, where it uses a logarithmic scale. Distinguished outliers are noticed in all datasets, but Piazza del Popolo. Moreover, such significant outliers constitute only a small portion of the columns. As mentioned in the main text, we preferred to avoid heuristic methods for the cutoff of outliers, and thus assumed a fixed percentage of 20% outlying column vectors.

			LUD						LUD+STE						LUD+SFMS					
Location	n	N	n	$\hat{e}_T$	$\tilde{e}_T$	$\hat{e}_R$	$\tilde{e}_R$	T_{total}	n	$\hat{e}_T$	$\tilde{e}_T$	$\hat{e}_R$	$\tilde{e}_R$	T_{total}	n	$\hat{e}_T$	$\tilde{e}_T$	$\hat{e}_R$	$\tilde{e}_R$	T_{total}
Alamo	570	606,963	563	1.74	0.47	3.04	1.03	328.8	497	1.90	0.45	3.08	1.04	242.6	466	2.24	0.46	4.44	1.81	183.3
Ellis Island	230	178,324	227	22.41	22.10	2.14	0.71	44.6	203	22.22	20.62	1.73	0.61	34.9	194	21.35	17.30	2.28	0.77	34.7
Madrid Metropolis	330	187,790	319	5.93	1.84	3.56	0.93	45.0	281	5.97	2.05	3.21	0.95	27.3	288	7.47	2.33	4.91	1.59	28.5
Montreal N.D.	445	633,938	437	1.22	0.56	0.96	0.49	237.8	388	1.12	0.52	1.02	0.52	159.7	381	1.15	0.54	1.05	0.49	149.5
Notre Dame	547	1,345,766	543	0.85	0.29	2.15	0.66	894.0	473	0.69	0.25	1.68	0.63	763.7	480	0.62	0.25	1.60	0.65	785.6
NYC Library	313	259,302	309	6.95	2.42	2.41	1.19	53.7	276	6.76	2.31	2.58	1.27	39.5	237	6.93	4.27	3.09	1.55	87.7
Piazza del Popolo	307	15,791	298	5.29	1.67	2.90	1.64	65.2	258	5.30	1.60	3.06	1.81	36.6	245	6.34	1.70	3.33	2.04	36.2
Piccadilly	2,226	1,278,612	2,171	4.04	1.81	4.82	1.72	1,649.0	1,937	3.98	1.95	4.37	1.76	857.4	1,827	4.98	2.72	5.33	2.01	654.4
Roman Forum	995	890,945	967	8.32	2.20	2.04	1.38	272.2	810	5.84	2.19	2.47	1.49	170.4	811	5.98	2.23	2.28	1.48	182.9
Tower of London	440	474171	431	17.86	3.96	2.92	2.13	65.2	355	17.95	4.79	3.01	2.63	44.9	338	19.35	4.62	3.64	2.92	43.1
Union Square	733	323933	715	11.75	7.57	6.59	3.21	53.7	651	10.90	7.29	6.21	3.47	53.8	571	13.14	8.81	6.93	3.92	36.5
Vienna Cathedral	789	1,361,659	774	13.10	7.21	7.28	1.65	762.6	685	12.56	6.28	7.91	1.81	547.1	666	12.82	7.78	8.08	1.80	365.6
Yorkminster	412	525,592	405	5.25	2.51	2.23	1.43	91.9	359	5.51	2.32	2.21	1.48	62.9	335	5.49	2.66	2.40	1.60	52.2
Gendarmenmarkt	671	338,800	654	38.82	17.46	39.63	6.43	124.2	596	32.94	14.70	29.01	4.65	79.2	564	25.77	15.88	38.16	5.50	77.8

Table 4. Performance of the LUD, LUD+STE and LUD+SFMS pipelines on the Photo Tourism datasets: n and N are the number of cameras and key points, respectively (when applying screening, n also denotes the number of the remaining cameras); $\hat{e}_R$,$\tilde{e}_R$ indicate mean and median errors of absolute camera rotations in degrees, respectively; $\hat{e}_T$,$\tilde{e}_T$ indicate mean and median errors of absolute camera translations in meters, respectively; T_{total} is the runtime of the pipeline (in seconds).

			LUD+TME						LUD+FMS					
Location	n	N	n	$\hat{e}_T$	$\tilde{e}_T$	$\hat{e}_R$	$\tilde{e}_R$	T_{total}	n	$\hat{e}_T$	$\tilde{e}_T$	$\hat{e}_R$	$\tilde{e}_R$	T_{total}
Alamo	570	606,963	463	1.96	0.48	3.04	1.07	216.6	480	2.05	0.51	3.26	1.10	203.0
Ellis Island	230	178,324	205	22.80	22.01	2.87	0.93	39.6	207	22.22	20.60	2.75	0.90	38.4
Madrid Metropolis	330	187,790	288	6.83	2.17	5.73	2.00	34.0	289	6.52	1.97	4.80	1.77	40.0
Montreal N.D.	445	633,938	381	1.10	0.53	0.96	0.48	155.8	389	1.12	0.54	0.97	0.50	167.0
Notre Dame	547	1,345,766	463	0.62	0.25	1.59	0.63	747.4	479	0.61	0.26	1.37	0.66	765.9
NYC Library	313	259,302	267	5.55	1.91	2.53	1.15	41.6	263	6.14	2.03	2.52	1.25	42.1
Piazza del Popolo	307	15,791	251	5.61	1.56	2.96	1.68	41.8	257	5.40	1.62	3.05	1.69	41.3
Piccadilly	2,226	1,278,612	1,980	3.44	1.65	3.19	1.77	1,211.7	1,888	3.43	1.68	3.20	1.75	1,172.9
Roman Forum	995	890,945	828	5.42	2.08	2.34	1.46	157.0	814	6.01	2.19	2.35	1.38	185.8
Tower of London	440	474,171	364	18.67	4.93	3.02	2.39	48.8	359	18.34	4.62	3.02	2.62	40.9
Union Square	733	323,933	612	11.80	7.68	6.18	3.41	49.4	610	12.49	7.48	6.13	3.37	44.1
Vienna Cathedral	789	1,361,659	694	13.77	7.26	7.69	2.89	468.7	692	13.54	7.17	7.70	2.90	472.0
Yorkminster	412	525,592	360	5.07	2.40	2.15	1.47	70.8	358	5.11	2.44	2.19	1.51	73.3
Gendarmenmarkt	671	338,800	602	35.43	14.69	28.30	4.08	96.7	600	32.87	14.34	28.59	4.09	85.0

Table 5. Performance of the LUD+TME and LUD+FMS pipelines on the Photo Tourism datasets: n and N are the number of cameras and key points, respectively (when applying screening, n also denotes the number of the remaining cameras); $\hat{e}_R$,$\tilde{e}_R$ indicate mean and median errors of absolute camera rotations in degrees, respectively; $\hat{e}_T$,$\tilde{e}_T$ indicate mean and median errors of absolute camera translations in meters, respectively; T_{total} is the runtime of the pipeline (in seconds).

Method	n	$\hat{e}_T$	$\tilde{e}_T$	$\hat{e}_R$	$\tilde{e}_R$	T_{total}
LUD	654	38.82	17.46	39.63	6.43	124.2
LUD+STE	497	22.82	12.53	26.76	3.60	51.24
LUD+TME	516	25.07	14.72	24.93	4.18	64.35
LUD+FMS	559	40.74	18.6	40.28	6.54	102.14
LUD+SFMS	549	41.31	19.33	40.65	6.47	100.40

Table 6. Comparison of the four pipelines when 45% outlying columns are eliminated in Gendarmenmarkt. Recall that $\hat{e}_R$,$\tilde{e}_R$ indicate mean and median errors of absolute camera rotations in degrees, respectively; $\hat{e}_T$,$\tilde{e}_T$ indicate mean and median errors of absolute camera translations in meters, respectively; and T_{total} is the runtime of the pipeline (in seconds).

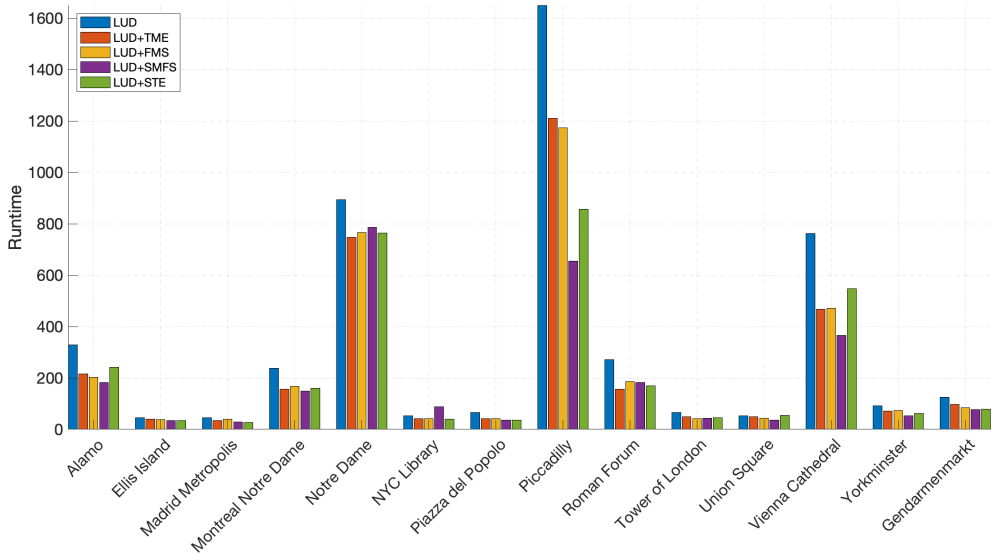

Figure 10. Runtime (in seconds) of the LUD pipeline and four LUD-type pipelines including an RSR method to initially screen bad cameras, applied to the 14 datasets of Photo Tourism.

A.2.1 Initial Camera Removal for SfM without Matrix Completion

For completeness, we also provide the numerical results for the initial camera removal experiment when the absent blocks $\mathbf{E}_{ij}$ are assigned zero matrices. Table 7 and Table 8 showcase the performance of the LUD+RSR piplines without the matrix completion step. Overall, integrating matrix completion into the LUD+RSR pipelines enhances their performance. The most notable enhancement for STE is for the mean translation error of the Roman Forum and Gendarmenmarkt. For other RSR algorithm we notice even more significant improvement with matrix completion. In other words, without matrix completion, the higher accuracy of STE in comparison to other RSR methods is more noticeable.

			LUD						LUD+STE						LUD+SFMS					
Location	n	N	n	$\hat{e}_T$	$\tilde{e}_T$	$\hat{e}_R$	$\tilde{e}_R$	T_{total}	n	$\hat{e}_T$	$\tilde{e}_T$	$\hat{e}_R$	$\tilde{e}_R$	T_{total}	n	$\hat{e}_T$	$\tilde{e}_T$	$\hat{e}_R$	$\tilde{e}_R$	T_{total}
Alamo	570	606,963	563	1.74	0.47	3.04	1.03	328.8	518	1.80	0.45	3.02	1.03	283.2	466	2.24	0.47	4.44	1.82	219.0
Ellis Island	230	178,324	227	22.41	22.10	2.14	0.71	44.6	209	22.51	20.78	2.81	0.90	41.0	195	21.51	18.50	2.27	0.77	33.3
Madrid Metropolis	330	187,790	319	5.93	1.84	3.56	0.93	45.0	290	5.69	1.81	3.24	0.99	37.6	278	5.82	2.25	3.54	1.12	32.1
Montreal N.D.	445	633,938	437	1.22	0.56	0.96	0.49	237.8	400	1.09	0.55	1.01	0.50	186.5	383	1.14	0.54	1.03	0.53	146.5
Notre Dame	547	1,345,766	543	0.85	0.29	2.15	0.66	894.0	495	0.79	0.27	2.15	0.64	798.4	476	0.72	0.26	1.87	0.66	815.6
NYC Library	313	259,302	309	6.95	2.42	2.41	1.19	53.7	278	6.78	2.51	2.57	1.20	42.6	238	6.82	4.61	3.13	1.49	42.2
Piazza del Popolo	307	15,791	298	5.29	1.67	2.90	1.64	65.2	247	5.60	1.69	2.22	1.24	35.7	242	6.62	1.59	3.51	2.15	45.9
Piccadilly	2,226	1,278,612	2171	4.04	1.81	4.82	1.72	1,649.0	1976	4.29	1.98	4.76	1.71	950.7	1827	4.98	2.72	5.33	2.01	661.6
Roman Forum	995	890,945	967	8.32	2.20	2.04	1.38	272.2	849	10.04	2.33	2.18	1.45	175.6	814	9.46	2.68	2.15	1.50	150.5
Tower of London	440	474,171	431	17.86	3.96	2.92	2.13	65.2	373	18.83	3.21	2.80	2.01	53.2	351	22.31	4.10	3.18	2.46	27.0
Union Square	733	323,933	715	11.75	7.57	6.59	3.21	53.7	624	11.92	7.55	6.24	3.59	49.6	558	13.42	9.04	6.87	3.48	38.5
Vienna Cathedral	789	1,361,659	774	13.10	7.21	7.28	1.65	762.6	715	12.78	6.26	7.66	1.66	590.2	666	12.82	7.78	8.08	1.80	444.0
Yorkminster	412	525,592	405	5.25	2.51	2.23	1.43	91.9	368	5.26	2.50	2.22	1.50	73.2	338	5.67	3.13	2.41	1.56	48.0
Gendarmenmarkt	671	338,800	654	38.82	17.46	39.63	6.43	124.2	616	37.60	15.49	31.29	4.75	108.3	543	35.34	27.53	62.89	31.04	83.7

Table 7. Performance of the LUD, LUD+STE and LUD+SFMS pipelines **without matrix completion** on the Photo Tourism datasets: n and N are the number of cameras and key points, respectively (when applying screening, n also denotes the number of the remaining cameras); $\hat{e}_R$,$\tilde{e}_R$ indicate mean and median errors of absolute camera rotations in degrees, respectively; $\hat{e}_T$,$\tilde{e}_T$ indicate mean and median errors of absolute camera translations in meters, respectively; T_{total} is the runtime of the pipeline (in seconds).

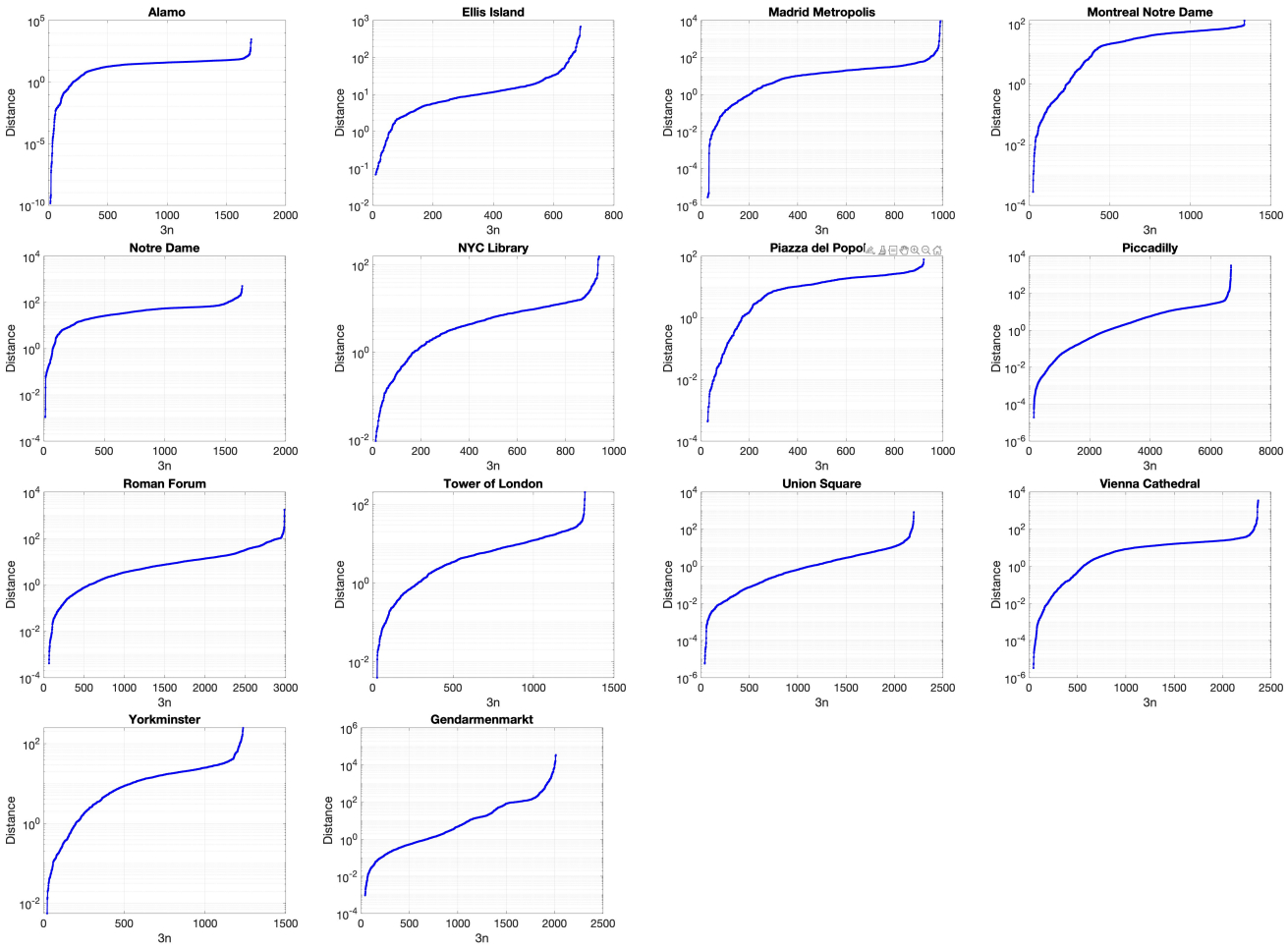

Figure 11. Distance of columns of $\mathbf{E}$ to the subspace recovered by STE (in log scale). The distances are sorted and plotted in descending order. All datasets exhibit significant outlying columns.

			LUD+TME						LUD+FMS					
Location	n	N	n	$\hat{e}_T$	$\tilde{e}_T$	$\hat{e}_R$	$\tilde{e}_R$	T_{total}	n	$\hat{e}_T$	$\tilde{e}_T$	$\hat{e}_R$	$\tilde{e}_R$	T_{total}
Alamo	570	606,963	463	1.96	0.48	3.04	1.07	216.6	480	2.05	0.51	3.26	1.10	203.0
Ellis Island	230	178,324	205	22.80	22.01	2.87	0.93	39.6	207	22.22	20.60	2.75	0.90	38.4
Madrid Metropolis	330	187,790	288	6.83	2.17	5.73	2.00	34.0	289	6.52	1.97	4.80	1.77	40.0
Montreal Notre Dame	445	633,938	381	1.10	0.53	0.96	0.48	155.8	389	1.12	0.54	0.97	0.50	167.0
Notre Dame	547	1,345,766	463	0.62	0.25	1.59	0.63	747.4	479	0.61	0.26	1.37	0.66	765.9
NYC Library	313	259,302	267	5.55	1.91	2.53	1.15	41.6	263	6.14	2.03	2.52	1.25	42.1
Piazza del Popolo	307	15,791	251	5.61	1.56	2.96	1.68	41.8	257	5.40	1.62	3.05	1.69	41.3
Piccadilly	2,226	1,278,612	1980	3.44	1.65	3.19	1.77	1,211.7	1888	3.43	1.68	3.20	1.75	1,172.9
Roman Forum	995	890,945	828	5.42	2.08	2.34	1.46	157.0	814	6.01	2.19	2.35	1.38	185.8
Tower of London	440	474,171	364	18.67	4.93	3.02	2.39	48.8	359	18.34	4.62	3.02	2.62	40.9
Union Square	733	323,933	612	11.80	7.68	6.18	3.41	49.4	610	12.49	7.48	6.13	3.37	44.1
Vienna Cathedral	789	1,361,659	694	13.77	7.26	7.69	2.89	468.7	692	13.54	7.17	7.70	2.90	472.0
Yorkminster	412	525,592	360	5.07	2.40	2.15	1.47	70.8	358	5.11	2.44	2.19	1.51	73.3
Gendarmenmarkt	671	338,800	602	35.43	14.69	28.30	4.08	96.7	600	32.87	14.34	28.59	4.09	85.0

Table 8. Performance of the LUD+TME and LUD+FMS pipelines **without matrix completion** on the Photo Tourism datasets: n and N are the number of cameras and key points, respectively (when applying screening, n also denotes the number of the remaining cameras); $\hat{e}_R$,$\tilde{e}_R$ indicate mean and median errors of absolute camera rotations in degrees, respectively; $\hat{e}_T$,$\tilde{e}_T$ indicate mean and median errors of absolute camera translations in meters, respectively; T_{total} is the runtime of the pipeline (in seconds).