BANSAC: A dynamic BAyesian Network for adaptive SAmple Consensus
(SUPPLEMENTARY MATERIALS)

Valter Piedade
Instituto Superior Técnico, Lisboa
valter.piedade@tecnico.ulisboa.pt

Pedro Miraldo
Mitsubishi Electric Research Labs
miraldo@merl.com

These supplementary materials present new quantitative experiments (Appendix A) and some additional derivations and pseudo-code (Appendix B).

Contents

A Additional Experiments

This section provides additional experiments with real-world and synthetic data. Appendices A.1 and A.2 show results with each scene from the PhotoTourism dataset for the essential and fundamental matrices estimation. Appendix A.3 offers results for curve and circle fitting problems using synthetic data. Appendix A.4 contains ablation studies on the conditional probability tables (CPTs), sampling weights, and stopping criteria.

For both of the relative pose problem experiments (Appendices A.1 and A.2), we use the following scenes from the PhotoTourism dataset, with a matching score cutoff of 0.85: 0) `brandenburg_gate` with 43% inliers; 1) `palace_of_westminster` with 32% inliers; 2) `westminster_abbey` with 49% inliers; 3) `taj_mahal` with 57% inliers; 4) `prague_old_town_square` with 32% inliers; and 5) `st_peters_square` with 46% inliers; 6) `buckingham_palace` with 45% inliers; 7) `colosseum_exterior` with 36% inliers; 8) `grand_place_brussels` with 31% inliers; 9) `notre_dame_front_facade` with 46% inliers; 10) `pantheon_exterior` with 62% inliers; 11) `temple_nara_japan` with 60% inliers; 12) `trevi_fountain` with 33% inliers; and 13) `sacre_coeur` with 51% inliers. As in the main document, we use 4K pairs for each scene and repeated each trial 5 times.

All experiments presented in this document and on the main paper were performed on an Intel(R) Core(TM) i7-7820X CPU @ 3.60GHz processor.

A.1 Calibrated relative pose

This subsection presents additional results for the calibrated relative pose estimation problem, comparing BANSAC and P-BANSAC against the baselines (RANSAC, NAPSAC, P-NAPSAC, and PROSAC). As estimation parameters, we use an error threshold of $1\mathrm{e}{-3}$ (normalized points), 1000 maximum iterations, and a confidence of 0.999, and set the BANSAC stopping criteria threshold τ to 0.01 in BANSAC and 0.1 in P-BANSAC (same parameters as in the main paper's results). Results are shown in Tab. 1.

We observe that, in accuracy, BANSAC and P-BANSAC are the best methods overall. In execution time, P-BANSAC is the best, with BANSAC second best in most scenes.

A.2 Uncalibrated relative pose

This subsection presents further results for the uncalibrated relative pose estimation problem, using the same baselines as in the previous subsection. As estimation parameters, we use an error threshold of 0.5, 10000 maximum iterations, and a confidence threshold of 0.999, and set the BANSAC stopping criteria threshold τ to 0.01 in BANSAC and 0.1 in P-BANSAC (same parameters as in the main paper's results). Results are shown in Tab. 1.

The results obtained are similar to those obtained in estimating the essential matrix. BANSAC is the best method in accuracy, followed by P-BANSAC in most scenes. Both are also the fastest methods overall.

Table 1. *Experimental results for the calibrated and uncalibrated relative pose estimation problems for each scene in the PhotoTourism dataset.*

Seq.	Calibrated Relative Pose Estimation (essential matrix estimation)						Uncalibrated Relative Pose Estimation (fundamental matrix estimation)					
	RANSAC	**NAPSAC**	**P-NAPSAC**	**PROSAC**	**BANSAC**	**P-BANSAC**	**RANSAC**	**NAPSAC**	**P-NAPSAC**	**PROSAC**	**BANSAC**	**P-BANSAC**
	Rotation mAA (10°) $\uparrow$											
0	0.711	0.245	0.740	0.754	0.773	**0.775**	0.574	0.260	0.585	0.593	**0.608**	0.593
1	0.555	0.218	0.604	0.612	0.624	**0.625**	0.482	0.253	0.504	0.514	**0.546**	0.537
2	0.714	0.417	0.709	0.714	**0.719**	0.717	0.686	0.455	0.684	0.686	**0.693**	0.689
3	**0.866**	0.259	0.797	0.820	**0.866**	0.848	0.863	0.567	0.859	0.857	**0.883**	0.863
4	0.322	0.142	0.290	0.302	**0.331**	0.311	0.269	0.111	0.267	0.246	**0.280**	0.264
5	0.759	0.251	0.745	0.772	0.803	**0.804**	0.628	0.336	0.617	0.617	**0.661**	0.642
6	0.684	0.216	0.658	0.693	**0.730**	0.724	0.569	0.275	0.566	0.571	**0.605**	0.576
7	0.434	0.136	0.448	0.449	0.467	**0.468**	0.374	0.191	0.375	0.377	**0.409**	0.394
8	0.357	0.133	0.359	0.368	**0.380**	0.379	0.301	0.186	0.295	0.300	**0.317**	0.306
9	0.669	0.271	0.698	0.716	0.730	**0.731**	0.582	0.258	0.593	0.599	**0.635**	0.625
10	0.762	0.204	0.718	0.707	**0.786**	0.784	0.467	0.306	0.420	0.434	**0.480**	0.438
11	0.829	0.255	0.797	0.815	**0.838**	0.816	0.762	0.484	0.746	0.744	**0.783**	0.728
12	0.532	0.217	0.568	0.579	**0.605**	0.598	0.458	0.212	0.468	0.471	**0.501**	0.490
13	0.827	0.201	0.836	0.844	**0.867**	0.862	0.804	0.418	0.819	0.819	**0.846**	0.839
	Translation mAA (10°) $\uparrow$											
0	0.581	0.150	0.599	0.613	0.643	**0.647**	0.363	0.0970	0.360	0.377	**0.394**	0.376
1	0.504	0.162	0.548	0.558	0.565	**0.567**	0.418	0.183	0.436	0.446	**0.480**	0.466
2	0.494	0.184	0.480	0.486	**0.506**	0.501	0.377	0.121	0.373	0.377	**0.390**	0.384
3	0.641	0.116	0.544	0.570	**0.649**	0.626	0.610	0.265	0.597	0.596	**0.635**	0.609
4	0.282	0.0970	0.244	0.253	**0.292**	0.267	0.176	0.0390	0.167	0.153	**0.192**	0.167
5	0.601	0.141	0.570	0.601	**0.642**	0.635	0.351	0.121	0.328	0.331	**0.379**	0.357
6	0.622	0.178	0.590	0.627	**0.661**	0.651	0.274	0.110	0.281	0.287	**0.311**	0.294
7	0.403	0.107	0.409	0.409	0.432	**0.434**	0.257	0.100	0.252	0.259	**0.296**	0.279
8	0.274	0.0850	0.267	0.277	**0.297**	0.296	0.140	0.0590	0.137	0.140	**0.151**	0.141
9	0.592	0.209	0.614	0.631	0.655	**0.662**	0.413	0.150	0.416	0.430	**0.461**	0.456
10	0.611	0.114	0.540	0.524	**0.620**	**0.620**	0.213	0.0770	0.169	0.180	**0.214**	0.185
11	0.617	0.0830	0.549	0.563	**0.630**	0.600	0.378	0.108	0.338	0.332	**0.389**	0.323
12	0.417	0.122	0.446	0.453	**0.491**	0.486	0.217	0.0460	0.215	0.215	**0.247**	0.235
13	0.798	0.173	0.792	0.800	**0.837**	0.832	0.757	0.329	0.761	0.759	**0.792**	0.777
	Avg. execution time $[ms]$ $\downarrow$											
0	26.6	40.3	19.9	20.5	**17.3**	17.4	15.4	29.4	10.9	11.8	12.6	**9.75**
1	34.9	43.3	29.1	31.1	21.7	**20.6**	21.9	30.2	18.8	20.1	14.6	**12.3**
2	17.9	35.8	15.4	15.2	**10.1**	10.7	10.1	25.9	12.6	9.67	5.84	**5.43**
3	13.0	37.1	9.48	9.49	9.02	**8.02**	7.90	25.5	5.76	5.60	6.72	**4.49**
4	33.5	42.4	28.8	28.8	17.2	**15.9**	20.7	28.8	18.4	18.1	11.6	**8.82**
5	22.1	37.9	17.3	16.8	15.8	**15.6**	12.7	24.6	9.39	9.29	11.5	**8.92**
6	28.0	39.9	21.7	23.5	18.1	**17.8**	14.4	27.1	9.17	12.1	10.6	**8.29**
7	32.4	42.2	28.1	29.9	17.1	**16.5**	19.4	27.5	18.5	18.8	9.86	**8.81**
8	37.6	43.2	32.7	34.4	19.7	**19.4**	22.3	26.3	20.1	21.2	11.7	**10.3**
9	26.6	40.1	21.2	22.0	**16.4**	**16.4**	14.7	28.3	12.1	12.4	10.1	**8.12**
10	13.4	38.2	**8.96**	10.7	9.41	9.73	5.67	12.3	4.43	4.72	5.08	**3.34**
11	13.8	36.9	8.90	8.98	9.33	**8.85**	5.16	16.3	3.71	3.45	4.51	**2.61**
12	37.1	43.2	32.0	32.6	**22.7**	23.2	21.4	30.8	19.2	19.4	14.4	**13.0**
13	21.4	41.3	17.1	16.8	**14.7**	14.8	11.3	35.0	9.74	9.15	8.61	**7.12**

A.3 Synthetic data

We consider two simple problems: curve and circle-fitting. For each, we ran several experiments, varying the inlier rate between 15 and 50%. Each experiment has 300 data points ranging between $[-1, 1]$. Inliers are disturbed by a Gaussian noise of mean 0 and variance 0.02, and outliers are modeled by a uniform distribution with a maximum absolute value of 1.0. We evaluate BANSAC against RANSAC and BaySAC, which we implemented from scratch since no code is available. As estimation parameters, we use an error threshold value of 0.02, 3000 maximum iterations, and an estimation confidence of 0.99. In BANSAC, the initial probabilities $\mathcal{P}^0$ are set to 0.5 for all data points, and the stopping criterion threshold τ is set to 0.01. We measure the root mean squared error (RMSE) of the geometric distance of points in the estimated model to the desired model and the number of iterations made. We present the mean values obtained after 1000 randomly generated trials. The results are shown in Fig. 1.

We observe that BANSAC has an accuracy similar to or better than the baselines requiring significantly fewer iterations, even for lower inlier rates. Figure 2 illustrates the BANSAC probability update for the curve fitting problem.

A.4 Ablation studies

Next, we test different configurations for three components of the proposed algorithm. We present experiments using diverse conditional probability tables (CPTs) parameters, various activation functions for sampling, and com-

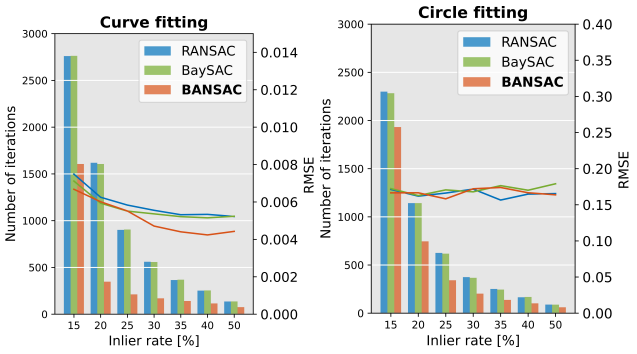

Figure 1. *Experimental results for the curve (left) and circle (right) fitting. We compare RANSAC, BaySAC, and BANSAC based on the number of iterations and RMS error for different inlier rates.*

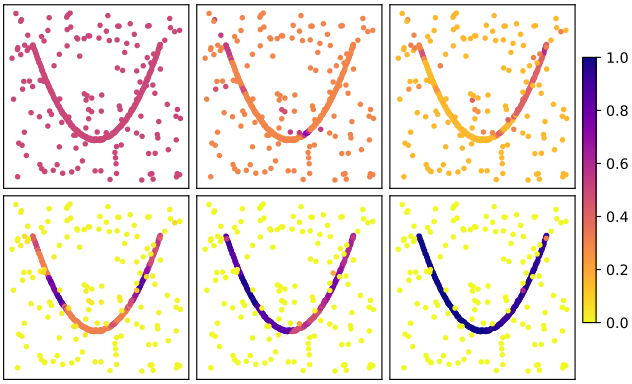

Figure 2. *Example of BANSAC inlier probability update over iterations for a curve fitting problem (color code at the right). In the first row, from left to right, we show iterations* 0*,* 2*, and* 4*. Second row shows iteration* 9*,* 11*, and* 14*.*

binations of different stopping criteria. The results were obtained using PhotoTourism sequence `sacre_coeur` (all pairs) for the uncalibrated relative pose problem (fundamental matrix estimation), using an error threshold of 0.5, 10000 maximum iterations, and a confidence of 0.999, as in the main paper. BANSAC stopping criterion threshold τ is set to 0.01. We evaluate the mAA of the rotation and translation errors at 5 and 10 degrees and the average execution time.

A.4.1 Conditional probability tables

To infer $P(x_n^k = \text{inlier} \mid c_n^{1:k})$ we need to define the CPTs of $P(c_n^k \mid x_n^{k-1})$ and $P(x_n^{k-1} \mid c_n^k, x_n^{k-1})$ for the 1st order Markov assumption. We present these CPTs in Tab. 2. The values for the CPT of $P(x_n^{k-1} \mid c_n^k, x_n^{k-1})$ were obtained empirically after testing different variations. We found that probability update is robust to slight variations of the reported parameters. The parameters of the CPT of $P(c_n^k \mid x_n^{k-1})$ are defined using a function $\gamma(\cdot)$. We want

Table 2. *Conditional probability table of* $P(c_k^n|\mathbf{x}_{k-1}^n)$ *and* $P(\mathbf{x}_k^n|c_k^n, \mathbf{x}_{k-1}^n)$.

c_n^k	x_n^{k-1}	$P(c_n^k \mid x_n^{k-1})$
Inlier	Inlier	$\gamma(\epsilon^k)$
Inlier	Outlier	$1-\gamma(\epsilon^k)$

x_n^k	x_n^{k-1}	c_n^k	$P(x_n^k \mid c_n^k, x_n^{k-1})$
Inlier	Inlier	Inlier	1.0
Inlier	Inlier	Outlier	1.0
Inlier	Outlier	Inlier	0.2
Inlier	Outlier	Outlier	0.0

Table 3. *Evaluation of BANSAC using different activation functions to define the parameters of the conditional probability table of* $P(c_n^k \mid x_n^{k-1})$.

Metrics	**Activation functions**		
	$\gamma_1(\psi)$	$\gamma_2(\psi)$	$\gamma_3(\psi)$
Rotation mAA (5°) ↑	0.836	0.793	0.790
Rotation mAA (10°) ↑	0.864	0.827	0.827
Translation mAA (5°) ↑	0.775	0.738	0.732
Translation mAA (10°) ↑	0.825	0.795	0.791
Avg. execution time $[ms]$ ↓	13.9	17.4	17.5

this function to give a high probability to classifications made by good models and vice versa. Since the quality of a model is defined by its inlier ratio, we define this function as $\gamma(\epsilon^k)$, where ϵ^k is the inlier ratio at iteration k. We test the following functions $\gamma(\epsilon^k)$ (variations of these functions with different values were tested, we are listing the ones that produced the best results):

$$\gamma_1(\epsilon^k) = \begin{cases} 0.62 \cdot \epsilon^k + 0.5, & \epsilon^k < 0.7143 \\ 0.2 \cdot \epsilon^k + 0.8, & \text{otherwise} \end{cases}, \quad (1)$$

$$\gamma_2(\epsilon^k) = \frac{0.5}{0.5 + e^{-10 \cdot (\epsilon^k - 0.3)}}, \text{and} \quad (2)$$

$$\gamma_3(\epsilon^k) = \tanh(3 \cdot \epsilon^k). \quad (3)$$

We present results using these functions with BANSAC and P-BANSAC in Tab. 3.

We achieved the best results in accuracy and execution time when using $\gamma_1(\psi)$. Based on these experiments, we decided to use $\gamma_1(\psi)$ in $P(c_n^k \mid x_n^{k-1})$ in all the experiments.

In the experiment shown in the main paper where we use the 2nd and 3rd orders of the Markov assumption, we use the CPTs shown in Tabs. 4 and 5, respectively. Similar to the CPT for the 1st order of the Markov assumption, the outlined parameters were obtained empirically.

A.4.2 Weighted sampling

In each iteration k, we perform a sampling weighted by the probabilities estimated in the previous iteration $\mathcal{P}^{k-1}$. Instead of simply using the probability values directly, we test the use of activation functions to increase the range of weights. The goal is to increase the chances of choosing

Table 4. *Conditional probability table of* $P(x_n^k \mid c_n^k, x_n^{k-2:k-1})$.

x_n^k	x_n^{k-1}	x_n^{k-2}	c_n^k	$P(x_n^k \mid c_n^k, x_n^{k-2:k-1})$
Inlier	Inlier	Inlier	Inlier	1.0
Inlier	Inlier	Inlier	Outlier	0.8
Inlier	Inlier	Outlier	Inlier	0.9
Inlier	Inlier	Outlier	Outlier	0.7
Inlier	Outlier	Inlier	Inlier	0.2
Inlier	Outlier	Inlier	Outlier	0.1
Inlier	Outlier	Outlier	Inlier	0.1
Inlier	Outlier	Outlier	Outlier	0.0

Table 5. *Conditional probability table of* $P(x_n^k \mid c_n^k, x_n^{k-3:k-1})$.

x_n^k	x_n^{k-1}	x_n^{k-2}	x_n^{k-3}	c_n^k	$P(x_n^k \mid c_n^k, x_n^{k-3:k-1})$
Inlier	Inlier	Inlier	Inlier	Inlier	1.0
Inlier	Inlier	Inlier	Inlier	Outlier	0.8
Inlier	Inlier	Inlier	Outlier	Inlier	0.9
Inlier	Inlier	Inlier	Outlier	Outlier	0.7
Inlier	Inlier	Outlier	Inlier	Inlier	0.6
Inlier	Inlier	Outlier	Inlier	Outlier	0.5
Inlier	Inlier	Outlier	Outlier	Inlier	0.4
Inlier	Inlier	Outlier	Outlier	Outlier	0.2
Inlier	Outlier	Inlier	Inlier	Inlier	0.3
Inlier	Outlier	Inlier	Inlier	Outlier	0.2
Inlier	Outlier	Inlier	Outlier	Inlier	0.1
Inlier	Outlier	Inlier	Outlier	Outlier	0.3
Inlier	Outlier	Outlier	Inlier	Inlier	0.2
Inlier	Outlier	Outlier	Inlier	Outlier	0.1
Inlier	Outlier	Outlier	Outlier	Inlier	0.05
Inlier	Outlier	Outlier	Outlier	Outlier	0.0

points with higher inlier probabilities. We tested the following activation functions (different functions were tested, and we are showing the ones that gave the best results):

$$\rho_1(\psi) = \psi \cdot 100 \tag{4}$$

$$\rho_2(\psi) = \begin{cases} 100 \cdot \psi & \psi > 0.3 \\ 10 \cdot \psi & \text{otherwise} \end{cases}, \tag{5}$$

$$\rho_3(\psi) = \frac{100}{1 + e^{-10 \cdot (\psi - 0.5)}} \tag{6}$$

$$\rho_4(\psi) = 130 \cdot \tanh(\psi) \tag{7}$$

where $\psi \triangleq P(x_n^k = \text{inlier} \mid C_n^{1:k})$ is the estimated probability for the n$^{\text{th}}$ data point at iteration k. In Tab. 6, we show results using these activations functions in BANSAC.

Of the tested functions, only $\rho_1(\psi)$ is linear. This function equally converts all points probabilities to the desired sampling range. The remaining give greater weights to points with higher probabilities and vice versa. Overall, we observed that $\rho_1(\psi)$ was the one that gave better results in accuracy and execution time. Based on these experiments, we chose to use $\rho_1(\psi)$ in all other experiments.

Table 6. *Evaluation of BANSAC using different activation functions to generate sampling weights.*

Metrics	**Sampling activation functions**			
	$\rho_1(\psi)$	$\rho_2(\psi)$	$\rho_3(\psi)$	$\rho_4(\psi)$
Rotation mAA (5°) ↑	0.836	0.825	0.823	0.834
Rotation mAA (10°) ↑	0.864	0.853	0.851	0.861
Translation mAA (5°) ↑	0.775	0.760	0.758	0.776
Translation mAA (10°) ↑	0.825	0.813	0.811	0.826
Avg. execution time $[ms]$ ↓	13.9	13.5	13.6	14.2

Table 7. *Evaluation of BANSAC with different stopping criteria.*

Stopping Criteria				**Results**				
				Rotation		Translation		Time
RANSAC	SPRT	PROSAC	BANSAC	mAA(5°)	mAA(10°)	mAA(5°)	mAA(10°)	Avg. $[ms]$
✓				0.845	0.868	0.792	0.837	16.2
	✓			0.837	0.864	0.775	0.825	13.9
		✓		0.839	0.865	0.782	0.829	15.1
			✓	0.850	0.870	0.818	0.854	33.5
✓			✓	0.845	0.867	0.793	0.837	16.1
	✓		✓	0.836	0.864	0.775	0.825	13.9
		✓	✓	0.838	0.864	0.782	0.829	14.4

A.4.3 Stopping criteria

Finally, we assess the different kinds and combinations of stopping criteria we can use with our method: RANSAC, SPRT, PROSAC, BANSAC, and BANSAC combined with RANSAC, SPRT, or PROSAC. We show results using these different combinations of stopping criteria in Tab. 7.

We observe that, although BANSAC stopping criteria ensure the output results are accurate, it is slow. However, when we combine our stopping condition with others, we consistently improve execution time with a slight drop in accuracy.

B Other Markov Assumptions

In this section, we present new derivations on probability updates. We show how to get exact inferences for second and third-order Markov assumptions.

B.1 Second-order Markov assumption

For the second-order assumption, in addition to the conditional independence constraints presented in the main paper, we have

$$x_n^j \perp x_n^{0:j-3} \mid x_n^{j-1}, x_n^{j-2}, c_n^j \;\; \forall j, \tag{8}$$

which means

$$P(x_n^j \mid x_n^{0:j-1}, c_n^j) = P(x_n^j \mid x_n^{j-2:j-1}, c_n^j), \;\; \forall j. \tag{9}$$

Now, similar to what is done in the main document, by applying the chain rule of probabilities, we write the joint probability at iteration k as

$$\widetilde{P}(x_n^{0:k}, c_n^{1:k}) = P(x_n^0) \prod_{j=1}^{k} \widetilde{\phi}(x_n^j, c_n^j), \tag{10}$$

Algorithm 1: *BANSAC algorithm outline. In the algorithm below, I means inlier and O outlier.*
Input – Data $\mathcal{Q}$, and without pre-computed scores
Output – Best model θ^*, and $\mathcal{C}^*$

$k \leftarrow 1$;
$\Phi_n^+ \leftarrow 0.5,\ \forall n$; ▷ for x_n^k = true (a pre-computed score can be used here)
$\Phi_n^- \leftarrow 0.5,\ \forall n$; ▷ for x_n^k = false (a pre-computed score can be used here)
$P_n = \frac{\Phi_n^+}{\Phi_n^+ + \Phi_n^-}$; ▷ current weights used for sampling
while $k < K$ **do**
  ...
  Other RANSAC steps as listed in the main paper;
  ...
  for *all* n **do**
    if $c_n^k = I$ **then**
      $\widehat{\Phi}_n^+ \leftarrow P(x_n^k = I, c_n^k = I, x_n^{k-1} = I)P(c_n^k = I, x_n^{k-1} = I)\Phi_n^+ + P(x_n^k = I, c_n^k = I, x_n^{k-1} = O)P(c_n^k = I, x_n^{k-1} = O)\Phi_n^-$;
      $\widehat{\Phi}_n^- \leftarrow P(x_n^k = O, c_n^k = I, x_n^{k-1} = I)P(c_n^k = I, x_n^{k-1} = I)\Phi_n^+ + P(x_n^k = O, c_n^k = I, x_n^{k-1} = O)P(c_n^k = I, x_n^{k-1} = O)\Phi_n^-$;
    else
      $\widehat{\Phi}_n^+ \leftarrow P(x_n^k = I, c_n^k = O, x_n^{k-1} = I)P(c_n^k = O, x_n^{k-1} = I)\Phi_n^+ + P(x_n^k = I, c_n^k = O, x_n^{k-1} = O)P(c_n^k = O, x_n^{k-1} = O)\Phi_n^-$;
      $\widehat{\Phi}_n^- \leftarrow P(x_n^k = O, c_n^k = O, x_n^{k-1} = I)P(c_n^k = O, x_n^{k-1} = I)\Phi_n^+ + P(x_n^k = O, c_n^k = O, x_n^{k-1} = O)P(c_n^k = O, x_n^{k-1} = O)\Phi_n^-$;
    $\Phi_n^+ \leftarrow \widehat{\Phi}_n^+$;
    $\Phi_n^- \leftarrow \widehat{\Phi}_n^-$;
    $P_n = \frac{\Phi_n^+}{\Phi_n^+ + \Phi_n^-}$;
  ...
  Other RANSAC steps as listed in the main paper;
  ...

where

$$\widetilde{\phi}(x_n^j, c_n^j) = \begin{cases} P(x_n^j \mid x_n^{j-2:j-1}, c_n^j)P(c_n^j \mid x_n^{j-1}), & j \geq 2 \\ P(x_n^1 \mid x_n^0, c_n^1)P(c_n^1 \mid x_n^0), & j = 1 \end{cases}. \tag{11}$$

We use $\widetilde{P}(.)$ to distinguish from the joint probability derived in the main document. Following the same steps derived in the main document, from Eqs. 10 and 11 the exact inference is given by

$$P(x_n^k = \text{inlier} \mid C_n^{1:k}) = \alpha \widetilde{\Phi}(x_n^k = \text{inlier}, x_n^{0:k-1}, C_n^{1:k}), \tag{12}$$

where again α is the normalization factor, and

$$\widetilde{\Phi}(x_n^k, x_n^{0:k-1}, C_n^{1:k}) = \sum_{x_n^{k-1}} \widetilde{\Phi}^\dagger(x_n^k, x_n^{0:k-1}, C_n^{1:k}) \tag{13}$$

where

$$\widetilde{\Phi}^\dagger(x_n^k, x_n^{0:k-1}, C_n^{1:k}) = \sum_{x_n^{k-2}} \widetilde{\phi}(x_n^k, C_n^k) \sum_{x_n^{k-3}} \widetilde{\phi}(x_n^{k-1}, C_n^{k-1}) \cdots \sum_{x_n^1} \widetilde{\phi}(x_n^3, C_n^3) \sum_{x_n^0} \widetilde{\phi}(x_n^2, C_n^2)\widetilde{\phi}(x_n^1, C_n^1)P(x_n^0). \tag{14}$$

As in the main document, a convenient result of Eq. 14 is that $\widetilde{\Phi}^\dagger(.)$ can be calculated recursively as follows:

$$\widetilde{\Phi}^\dagger(x_n^k, x_n^{0:k-1}, C_n^{1:k}) = \begin{cases} \sum_{x_n^{k-2}} \widetilde{\phi}(x_n^k, C_n^k)\widetilde{\Phi}^\dagger(x_n^{k-1}, x_n^{0:k-2}, C_n^{1:k-1}) & k \geq 2 \\ \widetilde{\phi}(x_n^1, C_n^1)P(x_n^0) & k = 1 \end{cases}. \tag{15}$$

For the second-order Markov assumption experiments, the only difference compared to what is described for the first-order is the use of the conditional probability in Eq. 12 as the sampling weights.

B.2 Third-order Markov assumption

For the third-order Markov assumption, we have the conditional independence constraints

$$x_n^j \perp x_n^{0:j-4} \mid x_n^{j-1}, x_n^{j-2}, x_n^{j-3}, c_n^j \ \ \forall j, \tag{16}$$

which means

$$P(x_n^j \mid x_n^{0:j-1}, c_n^j) = P(x_n^j \mid x_n^{j-3:j-1}, c_n^j), \ \ \forall j. \tag{17}$$

Again, by applying the chain rule of probabilities, we write the joint probability at iteration k as

$$\widetilde{\widetilde{P}}(x_n^{0:k}, c_n^{1:k}) = P(x_n^0) \prod_{j=1}^{k} \widetilde{\widetilde{\phi}}(x_n^j, c_n^j), \tag{18}$$

where

$$\widetilde{\widetilde{\phi}}(x_n^j, c_n^j) = \begin{cases} P(x_n^i \mid x_n^{j-3:j-1}, c_n^j) P(c_n^j \mid x_n^{j-1}), & j \geq 3 \\ P(x_n^2 \mid x_n^{0:1}, c_n^2) P(c_n^2 \mid x_n^1), & j = 2 \\ P(x_n^1 \mid x_n^0, c_n^1) P(c_n^1 \mid x_n^0), & j = 1 \end{cases} . \tag{19}$$

Following the same steps shown in the main document, from Eqs. 18 and 19 the exact inference is given by

$$P(x_n^k = \text{inlier} \mid C_n^{1:k}) = \alpha \widetilde{\widetilde{\Phi}}(x_n^k = \text{inlier}, x_n^{0:k-1}, C_n^{1:k}), \tag{20}$$

where again α is the normalization factor, and

$$\widetilde{\widetilde{\Phi}}(x_n^k, x_n^{0:k-1}, C_n^{1:k}) = \sum_{x_n^{k-1}} \sum_{x_n^{k-2}} \widetilde{\widetilde{\Phi}}^{\dagger}(x_n^k, x_n^{0:k-1}, C_n^{1:k}), \tag{21}$$

where

$$\begin{aligned} &\widetilde{\widetilde{\Phi}}^{\dagger}(x_n^k, x_n^{0:k-1}, C_n^{1:k}) = \\ &\qquad \sum_{x_n^{k-3}} \widetilde{\widetilde{\phi}}(x_n^k, C_n^k) \sum_{x_n^{k-4}} \widetilde{\widetilde{\phi}}(x_n^{k-1}, C_n^{k-1}) \cdots \\ &\sum_{x_n^1} \widetilde{\widetilde{\phi}}(x_n^4, C_n^4) \sum_{x_n^0} \widetilde{\widetilde{\phi}}(x_n^3, C_n^3) \widetilde{\widetilde{\phi}}(x_n^2, C_n^2) \widetilde{\widetilde{\phi}}(x_n^1, C_n^1) P(x_n^0). \end{aligned} \tag{22}$$

Again, we can write Eq. 22 in a recursive way:

$$\widetilde{\widetilde{\Phi}}^{\dagger}(x_n^k, x_n^{0:k-1}, C_n^{1:k}) = \begin{cases} \sum_{x_n^{k-3}} \widetilde{\widetilde{\phi}}(x_n^k, C_n^k) \widetilde{\widetilde{\Phi}}^{\dagger}(x_n^{k-1}, x_n^{0:k-2}, C_n^{1:k-1}) & k \geq 3 \\ \widetilde{\widetilde{\phi}}(x_n^2, C_n^2) \widetilde{\widetilde{\phi}}(x_n^1, C_n^1) P(x_n^0) & k = 2 \\ \widetilde{\widetilde{\phi}}(x_n^1, C_n^1) P(x_n^0) & k = 1 \end{cases} . \tag{23}$$

Note for $k = 1$, Eq. 22 does not sum in x_n^{-2}, since there is no such variable.

Finally, the weights for the sampling are taken from the inference in Eq. 20.

B.3 Probability Updates Pseudo-code

The probability updates derived in this code are easy to implement. An algorithm with the pseudo-code for the first-order Markov assumption is shown in Algorithm 1, in which probabilities are taken from Tab. 2. The second and third-order constraints are derived similarly.