Supplementary Material: Pose-graph via Adaptive Image Re-ordering

Daniel Barath[1]
https://people.inf.ethz.ch/dbarath/
Jana Noskova[2]
https://mat.fsv.cvut.cz/noskova/
Ivan Eichhardt[3]
http://cv.inf.elte.hu/
Jiri Matas[2]
https://cmp.felk.cvut.cz/~matas/

[1] Computer Vision and Geometry Group
ETH Zurich, Switzerland
[2] Visual Recognition Group,
FEE, CTU in Prague, Czech Republic
[3] Algorithms and Applications
Department, Eotvos Lorand University,
Budapest, Hungary

1 Path Scale Recovery

The algorithm proposed by Barath *et al*. [1] for recovering the relative pose between views v_s and v_d (s – source; d – destination) from a walk in the pose-graph, *i.e*. a chain of relative poses, provides only an approximation of the translation. Given the equation describing the pose chaining

$$\begin{aligned}
\phi(\mathcal{W}) &= \phi(f_{w_1}, f_{w_2}, \ldots, f_{w_{n-1}}) \\
&= \phi(f_{w_1}, f_{w_2}, \ldots, f_{w_{n-2}})\phi(f_{w_{n-1}}) \\
&= \phi(f_{w_1}, f_{w_2}, \ldots, f_{w_{n-3}})\phi(f_{w_{n-2}})\phi(f_{w_{n-1}}) \\
&= \ldots \\
&= \phi(f_{w_1})\phi(f_{w_2})\ldots\phi(f_{w_{n-1}}),
\end{aligned} \tag{1}$$

where $\mathcal{W} = (f_{w_1}, f_{w_2}, \ldots, f_{w_{n-1}})$ is a finite walk in the graph, for which there is a sequence of vertices $(v_{w_1}, v_{w_2}, \ldots, v_{w_n})$ such that $f_{w_i} \in \{e_{w_i}, e_{w_i}^{-1}\}$, $e_{w_i} = (v_{w_i}, v_{w_{i+1}})$ for $i = 1, 2, \ldots, n-1$, and $v_{w_1} = v_s$, $v_{w_n} = v_d$; e_{w_i} is a directed edge; $\phi : \mathcal{E} \to \mathrm{SE}(3)$ is a function returning the pose associated with an edge; and $\mathrm{SE}(3)$ is the manifold of 3D rigid transformations. Inverse $e_{w_i}^{-1}$ is calculated as $\phi(e_{w_i}^{-1}) = \phi(e_{w_i})^{-1}$ and by swapping the view indices.

The approximative nature of (1) comes from the fact the we are given unit-length translations that renders the final pose from (1) an approximation. In case the absolute scales of the edges are similar, the implied error is marginal. However, otherwise, it can lead to inaccurate solutions with a low number of inliers. In [1], this approximative nature is not crucial. It can, at most, make A* fail more often and, thus, RANSAC-based robust estimation applied at the cost of only a few milliseconds. However, A* can be made successful more often than in [1] when the path scales are recovered along the found walks. Using these recovered scales, (1) is not an approximation anymore. Consequently, the poses found by the A* algorithm become more accurate and A* fails less often than in [1].

Algorithm 1 Path Scale Recovery.

Input: $\mathcal{W} = (f_{w_1}, \ldots, f_{w_{n-1}})$ – walk;
$\mathcal{P}_{w_1}, \ldots, \mathcal{P}_{w_{n-1}}$ – point correspondences
Output: $s_{w_1}, \ldots, s_{w_{n-1}}$ – translation scales

$s_{w_1} \leftarrow 1$ ▷ Fixing the scale of the first edge.
for $i = 1 \,..\, n-2$ **do**
  $\mathcal{P}_{w_i w_{i+1}} \leftarrow \text{GetVisiblePoints}(\mathcal{P}_{w_i}, \mathcal{P}_{w_{i+1}})$ ▷ Eq. 2
  if $|\mathcal{P}_{w_i w_{i+1}}| = 0$ **then** ▷ No co-visible points.
    Break
  $\mathcal{S} \leftarrow \varnothing$ ▷ Scales from all correspondences.
  for $(\mathbf{p}, \mathbf{q}, \mathbf{r}) \in \mathcal{P}_{w_i w_{i+1}}$ **do**
    $[x_1, y_1, z_1]^{\mathrm{T}} \leftarrow \text{Triangulate}(\mathbf{q}, \mathbf{p}, \phi(f_{w_i})^{-1})$ ▷ Get the 3D coordinates.
    $[x_2, y_2, z_2]^{\mathrm{T}} \leftarrow \text{Triangulate}(\mathbf{q}, \mathbf{r}, \phi(f_{w_{i+1}}))$ ▷ *Default:* mid-point triangulation
    $\mathcal{S} \leftarrow \mathcal{S} \cup \{z_1 / z_2\}$ ▷ Saving the depth ratio of the image pairs.
  $s_{\text{med}} \leftarrow \text{Median}(\mathcal{S})$ ▷ Calculating the median depth ratio.
  if $i > 1$ **then** ▷ First edge has fixed scale
    $s_{w_i} \leftarrow s_{w_{i-1}} s_{\text{med}}$

Let us suppose that we are given a walk $\mathcal{W} = (f_{w_1}, f_{w_2}, \ldots, f_{w_{n-1}})$ in the pose-graph, and sets $\mathcal{P}_{w_i} = \{(\mathbf{p}, \mathbf{q}) \mid \mathbf{p}, \mathbf{q} \in \mathbb{R}^2\}$ of point correspondences in every consecutive image pair. In order to recover the translation scale, we use a similar approach as in [2] and iterate through the consecutive edge pairs, *i.e.* image triplets, $(f_{w_i}, f_{w_{i+1}})$ in walk $\mathcal{W}$, $i \in [1, n-2]$ forming local stellate graphs. For each edge pair, point correspondences that are visible in all three images are selected as

$$\mathcal{P}_{w_i w_{i+1}} = \{(\mathbf{p}, \mathbf{q}, \mathbf{r}) \mid \mathbf{p}, \mathbf{q}, \mathbf{r} \in \mathbb{R}^2 \wedge (\mathbf{p}, \mathbf{q}) \in \mathcal{P}_{w_i} \wedge (\mathbf{q}, \mathbf{r}) \in \mathcal{P}_{w_{i+1}}\}. \tag{2}$$

In case there are no such triplet correspondences and $|\mathcal{P}_{w_i w_{i+1}}| = 0$, we reject the walk and let A* find another one. Otherwise, without loss of generality, we can select the coordinate system of the middle image as the origin and triangulate each triplet correspondence $(\mathbf{p}, \mathbf{q}, \mathbf{r}) \in \mathcal{P}_{w_i^{-1} w_{i+1}}$ using both relative poses $\phi(f_{w_i})^{-1}$ and $\phi(f_{w_{i+1}})$. Note that w_i^{-1} means that the points in the correspondences are swapped. The pose inversion and point swapping is done to ensure that the middle image is in the origin. After the triangulation, we are given the scale ratio from the projective depths $s_{p,q}$ and $s_{q,r}$ of the 3D points as $s_{p,q}/s_{q,r}$. We take the median ratio over all point correspondences. Finally, the ratios are used to scale the translation vectors along the path. Since the global scale is still unknown, we fix the length of the first edge to one.

In the actual algorithm, we use the mid-point triangulation [3] since it is extremely fast and the main goal in this section is to efficiently find a relative pose that can be later improved by, *e.g.*, refitting on its inliers. Therefore, having very accurate results is not as crucial as speed. The algorithm is shown in Alg. 1. Also, we use only a maximum of 100 correspondences, to make the algorithm efficient.

A*	# successful paths ↑	# nodes visited ↓
w/ scale	**57331**	**4.9 × 10^8**
w/o scale	49949	5.1 × 10^8

Table 1. The average number of successful paths and total nodes visited by A*.

2 Additional Experiments

2.1 Results on each Scene

In Fig. 1, the $\log_{10}$ processing times (in seconds) spent on each component of the pipeline are shown for each tested algorithms. The compared algorithm are as follows:

1. (*Baseline*) Matching the image pairs, which survived the filtering by the inlier ratio predicted from GeM descriptors, by MAGSAC++.
2. (*A* w/o scale*) The A*-based technique proposed in [1] combined with MAGSAC++.
3. (*A* + scale + re-ord.*) A* with all the proposed algorithms and MAGSAC++.
4. (*A* + no scale + re-ord.*) A* without scale recovery with adaptive image pair re-ordering and MAGSAC++.
5. (*A* + scale + no re-ord.*) A* with scale recovery and MAGSAC++, without adaptive image pair re-ordering.
6. (*Baseline + re-ord.*) Matching the image pairs, which survived the filtering by the inlier ratio predicted from GeM descriptors, by MAGSAC++ and adaptive image pair re-ordering.

In all combinations, the maximum iteration number and the confidence of MAGSAC++ are set, respectively, to 5000 and 0.99.

In Fig. 1, the randomized robust pose estimation by MAGSAC++ clearly dominates the run-time in all cases. All other methods, *e.g.* A*, are 1–4 orders of magnitude faster. A* with scale recovery and adaptive re-ordering (A*+s+r) is the fastest on 6 out of the 11 scenes. On the other five scenes, A* with re-ordering (A*+r) is the fastest. Since we do not propose new algorithms for feature matching, we do not report the processing time spent on matching.

2.2 Scale Recovery

The average number of successful paths and total nodes visited by A* are reported in Table 1. A path is consider successful if the pose calculated from the pairs along the path lead to at least 20 inlier correspondences that can later be used for refitting the pose. The number of nodes visited indicates the effectiveness of the heuristic used in the A* algorithm. The fewer nodes are visited, the better the heuristic is. Table 1 clearly shows that the scale recovery leads to significantly more (by 15%) successful paths. Also, the number of nodes visited are decreased.

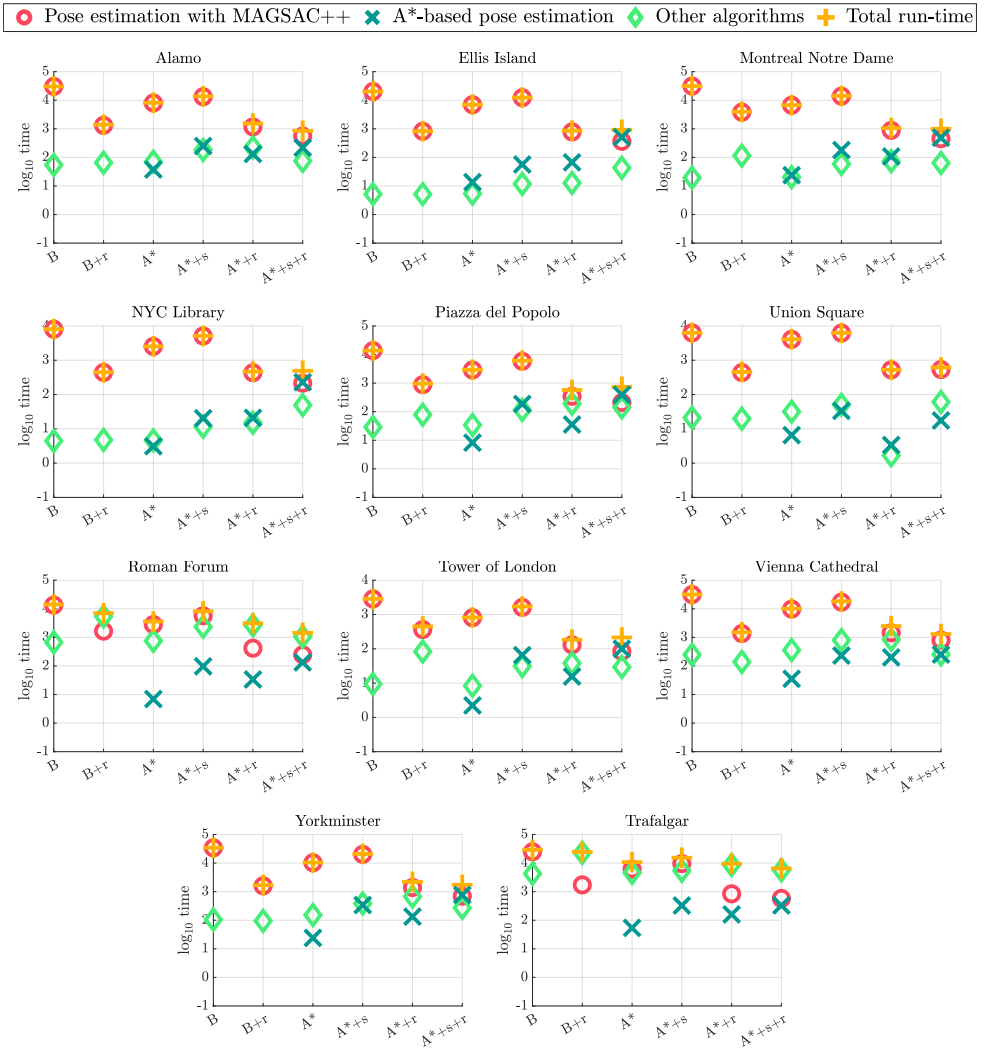

Figure 1. The $\log_{10}$ processing times (vertical axis) in seconds of each component of each tested algorithm (horizontal) on the 11 tested scenes from the 1DSfM dataset. The total run-time is shown by a yellow cross. The tested algorithms are the baseline (B); baseline with the proposed adaptive re-ordering (B+r); the algorithm from [0] (A*); A*-based pose estimation with the proposed path scale recovery (A* + s) and re-ordering (A* + r); and A* with all proposed algorithms (A* + s + r).

References

[1] Daniel Barath, Dmytro Mishkin, Ivan Eichhardt, Ilia Shipachev, and Jiri Matas. Efficient initial pose-graph generation for global sfm. In *CVPR*, pages 14546–14555, 2021.

[2] Zhaopeng Cui and Ping Tan. Global structure-from-motion by similarity averaging. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 864–872, 2015.

[3] Richard Hartley and Andrew Zisserman. *Multiple view geometry in computer vision*. Cambridge university press, 2003.