# Visual SLAM: Why Bundle Adjust?


Álvaro Parra[1] Tat-Jun Chin[1] Anders Eriksson[2] Ian Reid[1]

[1]School of Computer Science, The University of Adelaide
[2]School of Electrical Engineering and Computer Science, Queensland University of Technology


THE UNIVERSITY of ADELAIDE


## Abstract

Bundle adjustment (BA) is performed to estimate the 6DOF camera trajectory and 3D map from the input feature tracks in many modern SLAM pipelines.

**Two fundamental weaknesses plague SLAM systems based on BA:**

- The need to carefully **initialise** BA $\rightarrow$ 3D map mainainted over time; it makes the overall algorithm cumbersome.
- Since estimating the 3D structure (which requires sufficient baseline) is inherent in BA, the SLAM algorithm will encounter difficulties during periods of **slow motion** or **pure rotational motion**.

**We propose a different SLAM optimisation core:**

- We conduct **rotation averaging** to incrementally optimise *only camera orientations*.
- *Given* the orientations, we estimate the camera positions and 3D points via a **quasi-convex** formulation that can be solved efficiently and *globally optimally*.

**Our approach:**

- Obviates the need to estimate and maintain the camera positions and 3D map at keyframe rate $\rightarrow$ Simpler SLAM systems.
- It is also more capable of handling slow motions or pure rotational motions.


## Bundle Adjustment (BA)

Let $\mathbf{z}_{i,j}$ be the coordinates of the $i$-th scene point as seen in the $j$-th image $\mathsf{Z}_j$. Given $\{\mathbf{z}_{i,j}\}$, SfM aims to estimate the coordinates $\mathsf{X} = \{\mathbf{X}_i\}$ of the scene points and poses $\{(\mathbf{R}_j, \mathbf{t}_j)\}$ of the images $\{\mathsf{Z}_j\}$. The BA formulation is

$$\min_{\{\mathbf{X}_i\},\{(\mathbf{R}_j,\mathbf{t}_j)\}} \sum_{i,j} \|\mathbf{z}_{i,j} - f(\mathbf{X}_i \mid \mathbf{R}_j, \mathbf{t}_j)\|_2^2, \quad (1)$$

where $f(\mathbf{X}_i \mid \mathbf{R}_j, \mathbf{t}_j)$ is the projection function.

## Rotation Averaging

Given a set of **relative rotations** $\{\mathbf{R}_{j,k}\}$ between pairs of overlapping images $\{\mathsf{Z}_j, \mathsf{Z}_k\}$, estimate the **absolute rotations** $\{\mathbf{R}_j\}$ that are consistent with the relative rotations.

$$\min_{\{\mathbf{R}_j\}} \sum_{j,k \in \mathsf{N}} \|\mathbf{R}_{j,k} - \mathbf{R}_k \mathbf{R}_j^{-1}\|_F^2, \quad (2)$$

where $\mathsf{N}$ is the covisibility graph.

## BA-SLAM (adapted from [1])

```
X ← Initialise_points(Z_0).
for each keyframe step t = 1, 2, ... do
    s ← t − (window size) + 1.
    if a number of n ≥ 1 points left field of view then
        X ← X ∪ initialise_n_new_points(Z_t).
    end if
    R_{s:t}, t_{s:t}, X ← BA(R_{s:t}, t_{s:t}, X, Z_{0:t}).
    if loop is detected in Z_t then
        R_{1:t}, t_{1:t}, X ← BA(R_{1:t}, t_{1:t}, X, Z_{0:t}).
    end if
end for
```

## L-infinity SLAM

```
1:  for each keyframe step t = 1, 2, ... do
2:      s ← t − (window size) + 1.
3:      {R_{j,k}}_{j,k∈N_win} ← relative_rotation(Z_{(s−1):t}).
4:      R_{s:t} ← rotation_averaging({R_{j,k}}_{j,k∈N_win}).
5:      t_{s:t}, X ← known_rotation_prob(R_{s:t}, Z_{0:t}).
6:      if loop is detected in Z_t then
7:          {R_{j,k}}_{j,k∈N_sys} ← relative_rotation(Z_{0:t}).
8:          R_{1:t} ← rotation_averaging({R_{j,k}}_{j,k∈N_sys}).
9:          t_{1:t}, X ← known_rotation_prob(R_{1:t}, Z_{0:t}).
10:     end if
11: end for
```

## The Known Rotation Problem (KRot)

Given $\{\mathbf{R}_j\}$, KRot [2] optimises the camera positions $\{\mathbf{t}_j\}$ and 3D points $\{\mathbf{X}_i\}$ as

$$\min_{\{\mathbf{X}_i\},\{\mathbf{t}_j\}} \max_{i,j} \|\mathbf{z}_{i,j} - f(\mathbf{X}_i \mid \mathbf{R}_j, \mathbf{t}_j)\|_2, \quad \text{where} \quad f(\mathbf{X}_i \mid \mathbf{R}_j, \mathbf{t}_j) := \frac{\mathbf{R}_j^{(1:2)}\mathbf{X}_i + \mathbf{t}_j^{(1:2)}}{\mathbf{R}_j^{(3)}\mathbf{X}_i + \mathbf{t}_j^{(3)}}, \quad (3)$$

subject to cheirality constraints.

(3) can be rewritten by adding an extra variable $\gamma$ as

$$P_1: \quad \min_{\{\mathbf{X}_i\},\{\mathbf{t}_j\}} \gamma \quad (4a)$$

$$\text{subject to} \quad \left\| \mathbf{A}_{i,j} \begin{bmatrix} \mathbf{X}_i \\ \mathbf{t}_j \end{bmatrix} \right\|_2 \le \gamma\, \mathbf{b}_j^\top \begin{bmatrix} \mathbf{X}_i \\ \mathbf{t}_j \end{bmatrix}, \ \forall i, j, \quad (4b)$$

$$\gamma \ge 0, \quad (4c)$$

where: $\mathbf{A}_{i,j} = \begin{bmatrix} \mathbf{S}_{i,j} & \mathbf{I}_{2\times 2} & -\mathbf{z}_{i,j} \end{bmatrix}$, $\mathbf{b}_j = \begin{bmatrix} \mathbf{R}_j^{(3)} & 0 & 0 & 1 \end{bmatrix}^\top$, $\mathbf{S}_{i,j} = \mathbf{R}_j^{(1:2)} - \mathbf{z}_{i,j}\mathbf{R}_j^{(3)}$.

- (3) is **quasi-convex** $\rightarrow$ it is amenable to *efficient global solution* [2, 3].
- We use **Res-Int** [4] as the KRot routine in Line 5 in L-infinity SLAM. It outperformed existent methods by alternating between pose estimation and triangulation to efficiently partition the problem into small sub-problems - About 3 s in around 15 images and 3000 3D points.
- We use **KRot-TDC** as the KRot routine for loop closure in Line 9 in L-infinity SLAM. Res-Int performance is still inadequate for loop closure ($> 10,000$ 3D points, $> 100$ images).

## KRot with translation direction constraints (KRot-TDC)

We propose to address loop closure (Step 9 in L-infinity SLAM) over a sample of the input with a formulation that incorporates relative camera translation directions $\mathbf{t}_{j,k}^{(E)}$ in the in KRot problem. We constrained camera positions $\mathbf{C}_j = -\mathbf{R}_j^\top \mathbf{t}_j$ to agree up to an angular threshold to the relative translation direction:

$$\angle(\mathbf{t}_{j,k}, \mathbf{C}_k - \mathbf{C}_j) \le \alpha \quad \forall j, k, \quad (5)$$

where $\mathbf{t}_{j,k} := (\mathbf{K}^{-1}\mathbf{R}_j)^\top \mathbf{t}_{j,k}^{(E)}$ is the relative translation direction in world coordinates.

$$P_2: \quad \min_{\{\mathbf{X}_i\},\{\mathbf{C}_j\}} \gamma \quad (6a)$$

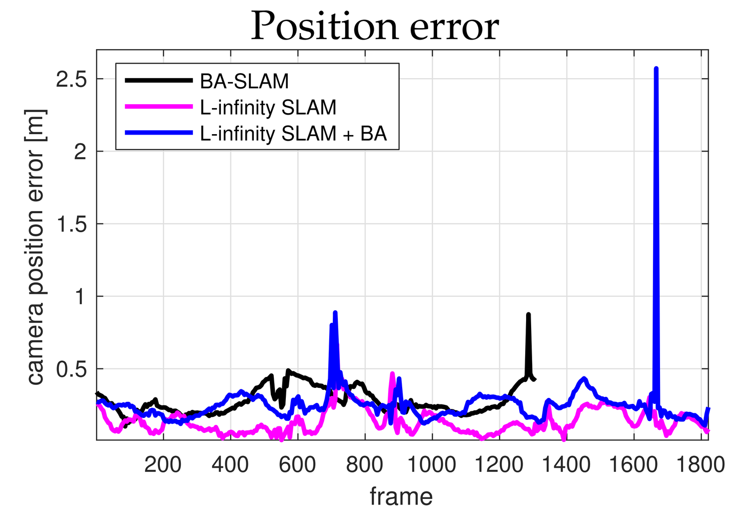
$$\text{subject to} \quad \left\| \mathbf{B}_{i,j} \begin{bmatrix} \mathbf{X}_i \\ \mathbf{C}_j \end{bmatrix} \right\|_2 \le \gamma\, \mathbf{c}_j^\top \begin{bmatrix} \mathbf{X}_i \\ \mathbf{C}_j \end{bmatrix}, \ \forall i, j, \quad (6b)$$

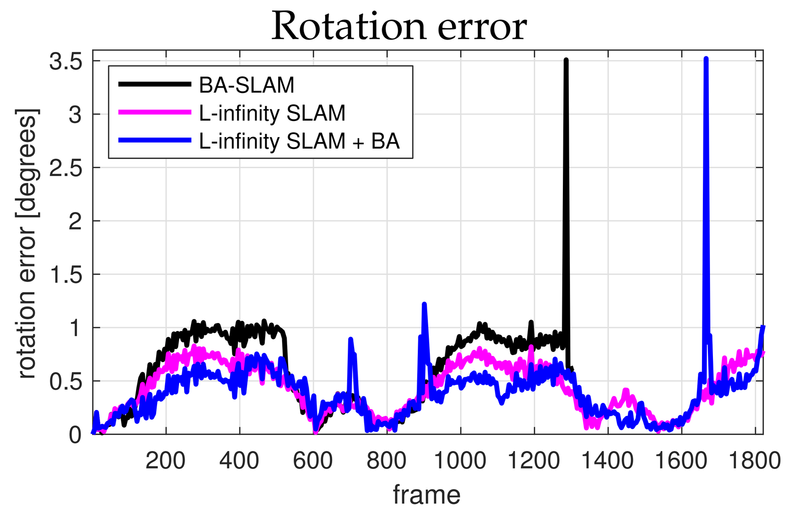
$$\left\| \mathbf{D}_{j,k} \begin{bmatrix} \mathbf{C}_j \\ \mathbf{C}_k \end{bmatrix} \right\|_2 \le \mathbf{e}_{j,k}^\top \begin{bmatrix} \mathbf{C}_j \\ \mathbf{C}_k \end{bmatrix}, \ \forall j, k, \quad (6c)$$

$$\gamma \ge 0, \quad (6d)$$

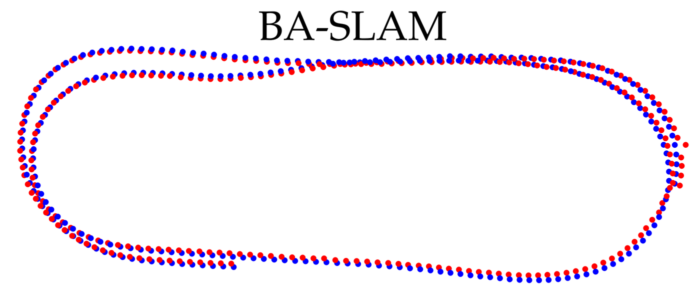
where:
$\mathbf{B}_{i,j} = \begin{bmatrix} \mathbf{S}_{i,j} & -\mathbf{S}_{i,j} \end{bmatrix}$, $\mathbf{c}_j = \begin{bmatrix} \mathbf{R}_j^{(3)} & -\mathbf{R}_j^{(3)} \end{bmatrix}$, $\mathbf{D}_{j,k} = \begin{bmatrix} \mathbf{Z}_{j,k}^{(1:2)} & -\mathbf{Z}_{j,k}^{(1:2)} \end{bmatrix}$, $\mathbf{e}_{j,k} = \tan(\alpha) \begin{bmatrix} \mathbf{t}_{j,k}^\top & -\mathbf{t}_{j,k}^\top \end{bmatrix}^\top$
and $\mathbf{Z}_{j,k}$ is a rotation matrix such that $\mathbf{Z}_{j,k}\mathbf{t}_{j,k} = [0\ 0\ 1]^\top$.

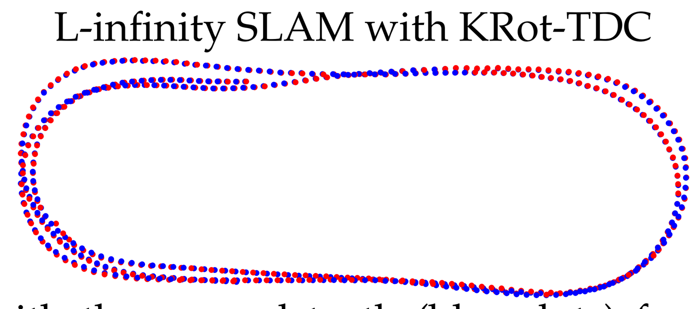
## Results

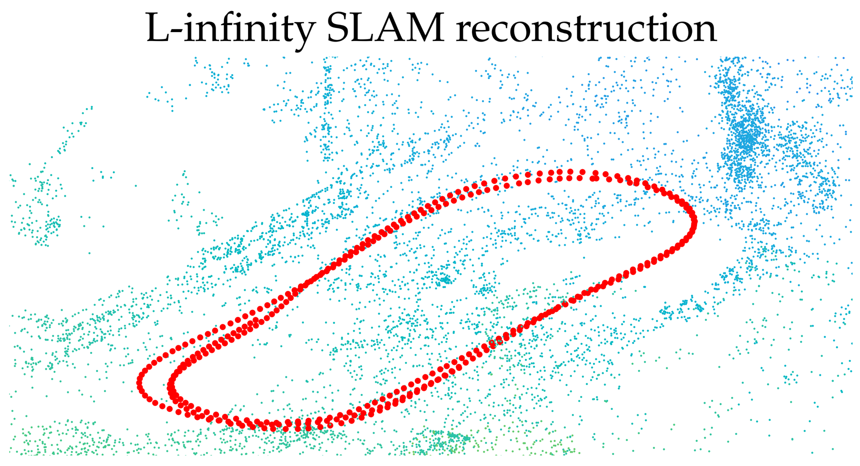



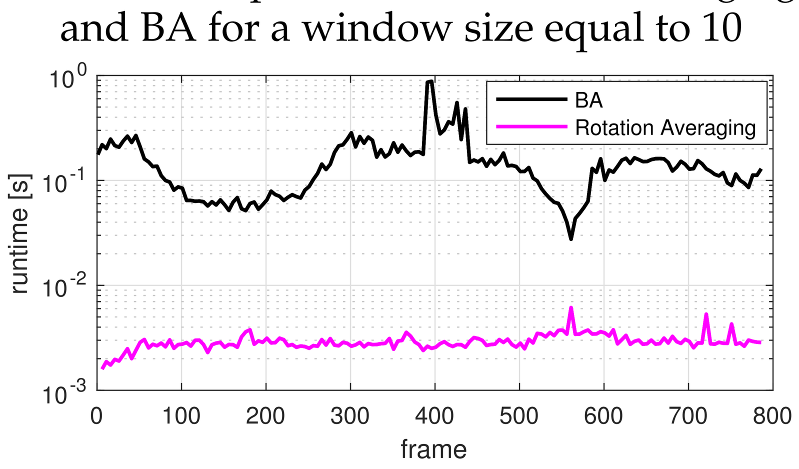

Camera position and rotation error for BA-SLAM and L-infinity SLAM in the Maptek dataset. The comparison includes the result of BA after KRot-TDC fed with same feature tracks.




Estimated camera positions (red dots) superposed with the ground truth (blue dots) for BA-SLAM, and L-infinity SLAM solving loop closure with the proposed KRot-TDC.




## Conclusions

- We presented L-infinity SLAM to be a simpler alternative to SLAM systems based on BA.
- Globally optimal quasi-convex optimisation $\rightarrow$ No need to maintain an accurate map and camera motions at key-frame rate as demanded by systems based on BA.
- The online effort is devoted to estimating camera orientations through rotation averaging.
- To efficiently solve loop closure, we proposed a variant of KRot which incorporates relative translation directions to accurately solve camera drifts over a sample of feature tracks.
- L-infinity SLAM is a simple and efficient alternative for applications requiring estimating slow motions or only rotational motions.

## References


1. H. Strasdat, J. Montiel, and A. Davison, “Visual SLAM: why filter?,” Image and Vision Computing, vol. 30, no. 2, 2012.
2. F. Kahl, “Multiple view geometry and the $L_\infty$-norm,” in ICCV, 2005.
3. C. Olsson, A. Eriksson, and F. Kahl, “Efficient optimization for $L_\infty$-problems using pseudoconvexity,” in ICCV, 2007.
4. Q. Zhang, T.-J. Chin, and H. M. Le, “A fast resection-intersection method for the known rotation problem,” in CVPR, 2018.