Visual SLAM: Why Bundle Adjust?

Alvaro Parra\textsuperscript{1} Tat-Jun Chin\textsuperscript{1} Anders Eriksson\textsuperscript{1} Ian Reid\textsuperscript{1}

\textsuperscript{1}School of Computer Science, The University of Adelaide
\textsuperscript{2}School of Electrical Engineering and Computer Science, Queensland University of Technology

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 → 3D map mainained 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 motions.

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 → Simpler SLAM systems.
- It is also more capable of handling slow motions or pure rotational motions.

BUNDLE ADJUSTMENT (BA)

Let $z_{ij}$ be the coordinates of the i-th scene point as seen in the j-th image $Z_j$. Given $z_{ij}$, SfM aims to estimate the coordinates $X_i = \{X_i\}$ of the scene points and poses $(R_{ij}, t_{ij})$ of the images $\{Z_j\}$. The BA formulation is

$$\min_{(X_i,\{R_{ij}, t_{ij}\})} \sum_{i,j} \|z_{ij} - f(X_i, R_{ij}, t_{ij})\|_C^2,$$  \hspace{1cm} (1)

where $f(X_i, R_{ij}, t_{ij})$ is the projection function.

BA-SLAM (ADAPTED FROM [1])

1. $X \leftarrow \text{Initialise points}(Z_0)$
2. for each keyframe step $t = 1, 2, \ldots$ do
3. $s \leftarrow t - \text{window size} + 1$ if a number of $s \geq 1$ points left field of view then
4. $X \leftarrow X \cup \text{Initialise new points}(Z_t)$
5. end if
6. $R_{ij}, t_{ij}, X \leftarrow \text{BA}(R_{ij}, t_{ij}, X, Z_t)$
7. if loop is detected in $Z_t$ then
8. $R_{ij}, t_{ij}, X \leftarrow \text{BA}(R_{ij}, t_{ij}, X, Z_t)$
9. end if
10. end for

L-infinity SLAM

1. for each keyframe step $t = 1, 2, \ldots$ do
2. $s \leftarrow t - \text{window size} + 1$
3. $(R_{ij}, t_{ij}) \leftarrow \text{relative_rotation}(Z_{t-s:s})$
4. $R_{ij} \leftarrow \text{rotation_averaging}(R_{ij})$
5. $t_{ij} \leftarrow \text{known_rotation_prob}(R_{ij}, t_{ij})$
6. if loop is detected in $Z_t$ then
7. $(R_{ij}, t_{ij}) \leftarrow \text{known_rotation_prob}(R_{ij}, t_{ij})$
8. end if
9. end for

THE KNOWN ROTATION PROBLEM (KRot)

Given $(R_{ij}, \gamma)$, KRot [2] optimises the camera positions $[t_{ij}]$ and 3D points $[X_i]$ as

$$\min_{(X_i, t_{ij})} \max_{ij} \|z_{ij} - f(X_i, R_{ij}, t_{ij})\|_C,$$  \hspace{1cm} (3)

subject to cheirality constraints.

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

$$\min_{(X_i, t_{ij})} \gamma,$$

subject to

$$A_{ij} \cdot X_i \leq \gamma,$$

$$b_{ij} \cdot X_i \leq \gamma,$$

$$\gamma \geq 0,$$

where

$$A_{ij} = [S_{ij}, I_{s \times 2} - z_{ij}], \ \ b_{ij} = [R_{ij}^T \left[ \begin{array}{c} 0 \\ 0 \\ 1 \end{array} \right]]^T, \ \ S_{ij} = R_{ij}^{-1} - z_{ij}R_{ij}^{-1}.$$

(3) is quasi-convex $\rightarrow$ it is amenable to efficient global solution [2, 3].

We use Res-Inf [4] as the KRot routine in Line 5 in L-infinity SLAM. It outperformed existant methods by alternating between pose estimation and triangulation to efficiently partition the problem into smaller sub-problems - About 3s 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-Inf performance is still inadequate for loop closure ($> 10,000$ 3D points, > 100 images).

REFERENCES


RESULTS

- 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.

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.