Rotation-Only Bundle Adjustment

Seong Hun Lee and Javier Civera

CVPR 2021

Universidad Zaragoza

1542

ROBOTICS

UNIVERSIDAD DE ZARAGOZA

Presentation outline

1. Introduction to Rotation-Only Bundle Adjustment
2. Normalized Epipolar Error (Lee & Civera 2020)
3. Two-view Rotation-Only Optimization (Kneip & Lynen 2013)
4. Rotation-Only Bundle Adjustment
5. Results
6. Limitations and Future work
7. Conclusion

1. Introduction

- Structure-from-Motion (SfM):
Reconstructing the camera poses and the scene structure from the list of unordered images.

1. Introduction

- Structure-from-Motion (SfM): Reconstructing the camera poses and the scene structure from the list of unordered images.

[Image from https://www.cs.cornell.edu/~snavely/bundler/]

1. Introduction

- Structure-from-Motion (SfM): Reconstructing the camera poses and the scene structure from the list of unordered images.

[Image from https://www.cs.cornell.edu/~snavely/bundler/]

- Global SfM pipeline:

1. Introduction

- Structure-from-Motion (SfM): Reconstructing the camera poses and the scene structure from the list of unordered images.

[Image from https://www.cs.cornell.edu/~snavely/bundler/]

- Global SfM pipeline:

(1) Feature extraction and matching across all images (e.g., SIFT descriptors).

1. Introduction

- Structure-from-Motion (SfM): Reconstructing the camera poses and the scene structure from the list of unordered images.

[Image from https://www.cs.cornell.edu/~snavely/bundler/]

- Global SfM pipeline:

(1) Feature extraction and matching across all images (e.g., SIFT descriptors).

(2) Relative pose estimation between image pairs (e.g., Nister's 5-pt method in RANSAC).

1. Introduction

- Structure-from-Motion (SfM): Reconstructing the camera poses and the scene structure from the list of unordered images.

[Image from https://www.cs.cornell.edu/~snavely/bundler/]

- Global SfM pipeline:

(1) Feature extraction and matching across all images (e.g., SIFT descriptors).

(2) Relative pose estimation between image pairs (e.g., Nister's 5-pt method in RANSAC).

(3) Global rotation estimation by multiple rotation averaging.

1. Introduction

- Structure-from-Motion (SfM): Reconstructing the camera poses and the scene structure from the list of unordered images.

[Image from https://www.cs.cornell.edu/~snavely/bundler/]

- Global SfM pipeline:

(1) Feature extraction and matching across all images (e.g., SIFT descriptors).

(2) Relative pose estimation between image pairs (e.g., Nister's 5-pt method in RANSAC).

(3) Global rotation estimation by multiple rotation averaging.

(4) Global translation estimation.

1. Introduction

- Structure-from-Motion (SfM): Reconstructing the camera poses and the scene structure from the list of unordered images.

[Image from https://www.cs.cornell.edu/~snavely/bundler/]

- Global SfM pipeline:

(1) Feature extraction and matching across all images (e.g., SIFT descriptors).

(2) Relative pose estimation between image pairs (e.g., Nister's 5-pt method in RANSAC).

(3) Global rotation estimation by multiple rotation averaging.

(4) Global translation estimation.

(5) Multiview triangulation.

1. Introduction

- Structure-from-Motion (SfM): Reconstructing the camera poses and the scene structure from the list of unordered images.

[Image from https://www.cs.cornell.edu/~snavely/bundler/]

- Global SfM pipeline:

(1) Feature extraction and matching across all images (e.g., SIFT descriptors).

(2) Relative pose estimation between image pairs (e.g., Nister's 5-pt method in RANSAC).

(3) Global rotation estimation by multiple rotation averaging.

(4) Global translation estimation.

(5) Multiview triangulation.

(6) Bundle adjustment, i.e., a joint optimization of the camera poses and the 3D points.

1. Introduction

- Global SfM pipeline:

(1) Feature extraction and matching

(2) Relative pose estimation

(3) Global rotation estimation by multiple rotation averaging.

(4) Global translation estimation

(5) Multiview triangulation

(6) Bundle adjustment (BA)

1. Introduction

- Global SfM pipeline:

(1) Feature extraction and matching

(2) Relative pose estimation

(3) Global rotation estimation by multiple rotation averaging.

(4) Global translation estimation

(5) Multiview triangulation

(6) Bundle adjustment (BA)

→ Steps (4), (5) and (6) are influenced by the rotation accuracy of step (3).

1. Introduction

- Global SfM pipeline:

(1) Feature extraction and matching
(2) Relative pose estimation
(3) Global rotation estimation by multiple rotation averaging.
(4) Global translation estimation
(5) Multiview triangulation
(6) Bundle adjustment (BA)

→ Steps (4), (5) and (6) are influenced by the rotation accuracy of step (3).
To improve the accuracy, we propose an intermediate step (3.5).

1. Introduction

- Global SfM pipeline:

(1) Feature extraction and matching

(2) Relative pose estimation

(3) Global rotation estimation by multiple rotation averaging.

(4) Global translation estimation

(5) Multiview triangulation

(6) Bundle adjustment (BA)

(3.5) Rotation-Only Bundle Adjustment (ROBA)

→ Steps (4), (5) and (6) are influenced by the rotation accuracy of step (3).
To improve the accuracy, we propose an intermediate step (3.5).

1. Introduction

- Global SfM pipeline:

(1) Feature extraction and matching

(2) Relative pose estimation

(3) Global rotation estimation by multiple rotation averaging.

(4) Global translation estimation

(5) Multiview triangulation

(6) Bundle adjustment (BA)

(3.5) Rotation-Only Bundle Adjustment (ROBA)

→ Steps (4), (5) and (6) are influenced by the rotation accuracy of step (3).
To improve the accuracy, we propose an intermediate step (3.5).
Advantages:

1. Introduction

- Global SfM pipeline:

(1) Feature extraction and matching

(2) Relative pose estimation

(3) Global rotation estimation by multiple rotation averaging.

(4) Global translation estimation

(5) Multiview triangulation

(6) Bundle adjustment (BA)

(3.5) Rotation-Only Bundle Adjustment (ROBA)

→ Steps (4), (5) and (6) are influenced by the rotation accuracy of step (3).
To improve the accuracy, we propose an intermediate step (3.5).
Advantages:

(i) Unlike rotation averaging, the optimization uses the image measurements as direct input. This leads to the accuracy improvement.

1. Introduction

- Global SfM pipeline:

(1) Feature extraction and matching

(2) Relative pose estimation

(3) Global rotation estimation by multiple rotation averaging.

(4) Global translation estimation

(5) Multiview triangulation

(6) Bundle adjustment (BA)

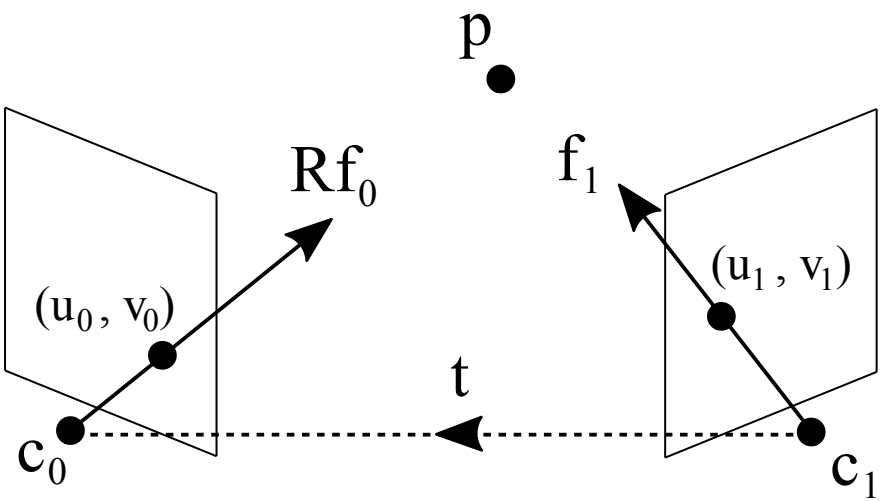
(3.5) Rotation-Only Bundle Adjustment (ROBA)

→ Steps (4), (5) and (6) are influenced by the rotation accuracy of step (3).
To improve the accuracy, we propose an intermediate step (3.5).
Advantages:

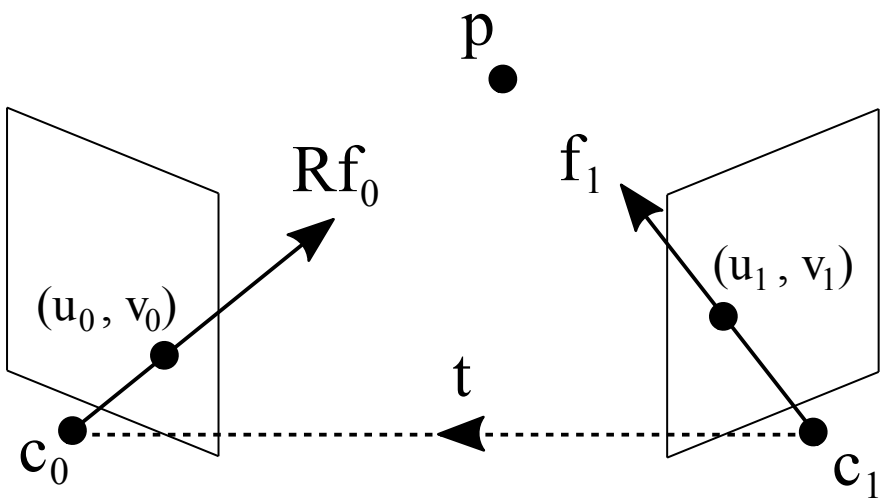
(i) Unlike rotation averaging, the optimization uses the image measurements as direct input. This leads to the accuracy improvement.

(ii) Unlike BA, it is completely independent of the translations and structure.

2. Normalized Epipolar Error



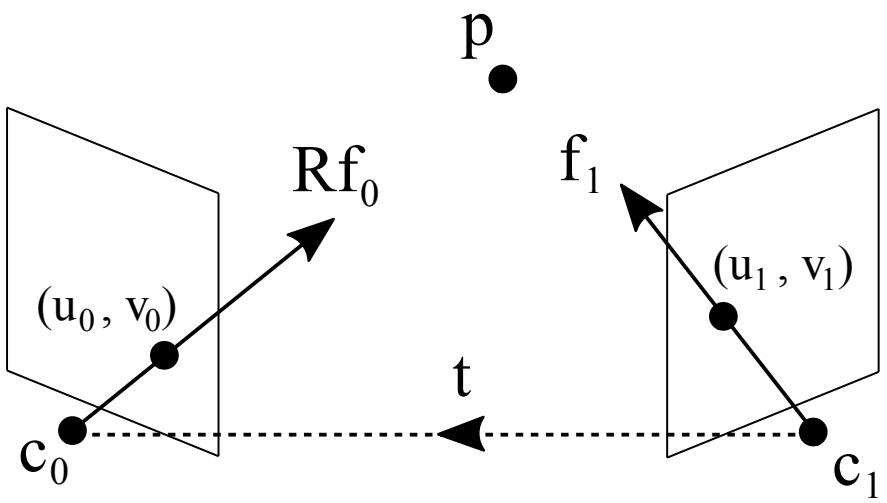
2. Normalized Epipolar Error



$$f_0 = K_0^{-1} \begin{bmatrix} u_0 \\ v_0 \\ 1 \end{bmatrix} = \begin{bmatrix} x_0/z_0 \\ y_0/z_0 \\ 1 \end{bmatrix},$$ f_1 can be defined analogously.

K_0 is the camera intrinsic matrix of c_0.
(x_0, y_0, z_0) is the coordinates of point p in the camera reference frame c_0.

2. Normalized Epipolar Error

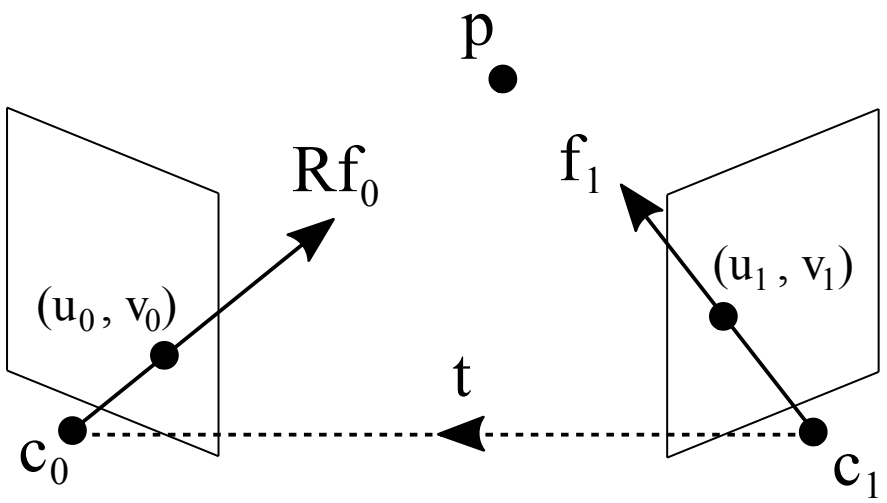


$$f_0 = K_0^{-1} \begin{bmatrix} u_0 \\ v_0 \\ 1 \end{bmatrix} = \begin{bmatrix} x_0/z_0 \\ y_0/z_0 \\ 1 \end{bmatrix},$$

f_1 can be defined analogously.

K_0 is the camera intrinsic matrix of c_0.
(x_0, y_0, z_0) is the coordinates of point p in the camera reference frame c_0.

2. Normalized Epipolar Error



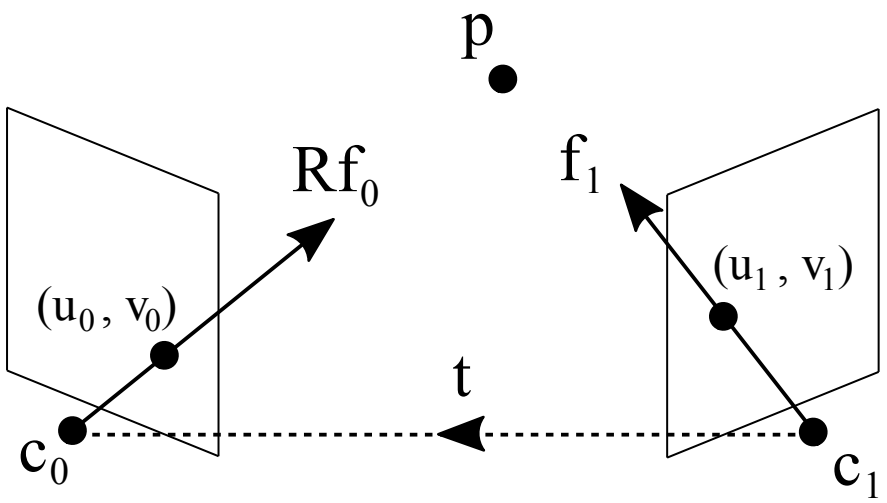
$$f_0 = K_0^{-1} \begin{bmatrix} u_0 \\ v_0 \\ 1 \end{bmatrix} = \begin{bmatrix} x_0/z_0 \\ y_0/z_0 \\ 1 \end{bmatrix},$$ f_1 can be defined analogously.

K_0 is the camera intrinsic matrix of c_0.
(x_0, y_0, z_0) is the coordinates of point p in the camera reference frame c_0.

- Epipolar error $e := \left| \mathbf{f}_1 \cdot \left(\widehat{\mathbf{t}} \times \mathbf{R}\mathbf{f}_0 \right) \right| = \left| \mathbf{f}_1^\top \mathbf{E}\mathbf{f}_0 \right|$

with $\mathbf{E} = [\widehat{\mathbf{t}}]_\times \mathbf{R}$

2. Normalized Epipolar Error



$$f_0 = K_0^{-1} \begin{bmatrix} u_0 \\ v_0 \\ 1 \end{bmatrix} = \begin{bmatrix} x_0/z_0 \\ y_0/z_0 \\ 1 \end{bmatrix},$$ f_1 can be defined analogously.

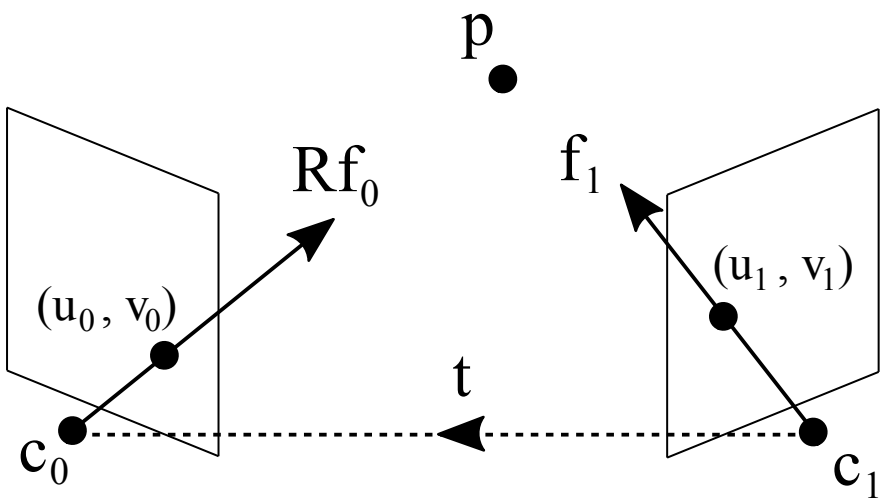
K_0 is the camera intrinsic matrix of c_0.
(x_0, y_0, z_0) is the coordinates of point p in the camera reference frame c_0.

- Epipolar error $e := \left|\mathbf{f}_1 \cdot \left(\widehat{\mathbf{t}} \times \mathbf{R}\mathbf{f}_0\right)\right| = \left|\mathbf{f}_1^\top \mathbf{E}\mathbf{f}_0\right|$

with $\mathbf{E} = [\widehat{\mathbf{t}}]_\times \mathbf{R}$

- Normalized epipolar error $\widehat{e} := \left|\widehat{\mathbf{f}}_1 \cdot \left(\widehat{\mathbf{t}} \times \mathbf{R}\widehat{\mathbf{f}}_0\right)\right| = \left|\widehat{\mathbf{f}}_1^\top \mathbf{E}\widehat{\mathbf{f}}_0\right|$

2. Normalized Epipolar Error



$$f_0 = K_0^{-1} \begin{bmatrix} u_0 \\ v_0 \\ 1 \end{bmatrix} = \begin{bmatrix} x_0/z_0 \\ y_0/z_0 \\ 1 \end{bmatrix},$$ f_1 can be defined analogously.

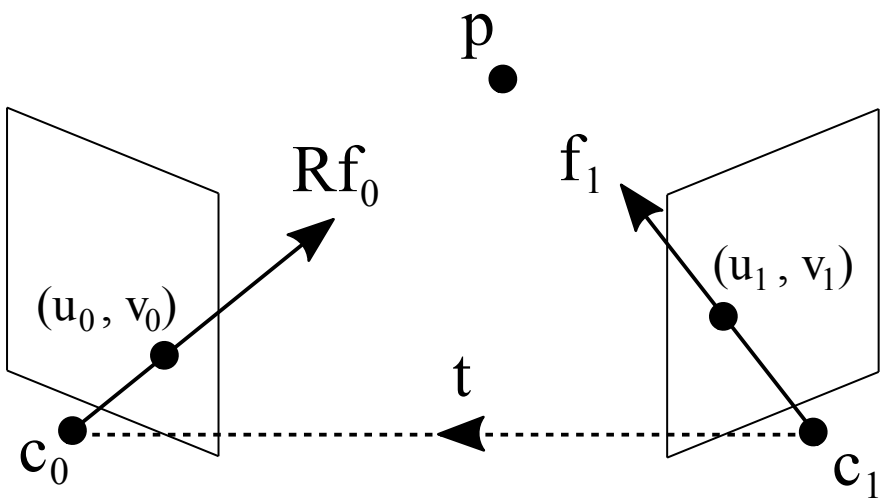
K_0 is the camera intrinsic matrix of c_0.
(x_0, y_0, z_0) is the coordinates of point p in the camera reference frame c_0.

- Epipolar error $e := \left| \mathbf{f}_1 \cdot \left(\hat{\mathbf{t}} \times \mathbf{R}\mathbf{f}_0 \right) \right| = \left| \mathbf{f}_1^\top \mathbf{E} \mathbf{f}_0 \right|$ → The 3rd element of f_0 and f_1 is one.

with $\mathbf{E} = [\hat{\mathbf{t}}]_\times \mathbf{R}$

- Normalized epipolar error $\hat{e} := \left| \hat{\mathbf{f}}_1 \cdot \left(\hat{\mathbf{t}} \times \mathbf{R}\hat{\mathbf{f}}_0 \right) \right| = \left| \hat{\mathbf{f}}_1^\top \mathbf{E} \hat{\mathbf{f}}_0 \right|$

2. Normalized Epipolar Error



$$f_0 = K_0^{-1} \begin{bmatrix} u_0 \\ v_0 \\ 1 \end{bmatrix} = \begin{bmatrix} x_0/z_0 \\ y_0/z_0 \\ 1 \end{bmatrix},$$ f_1 can be defined analogously.

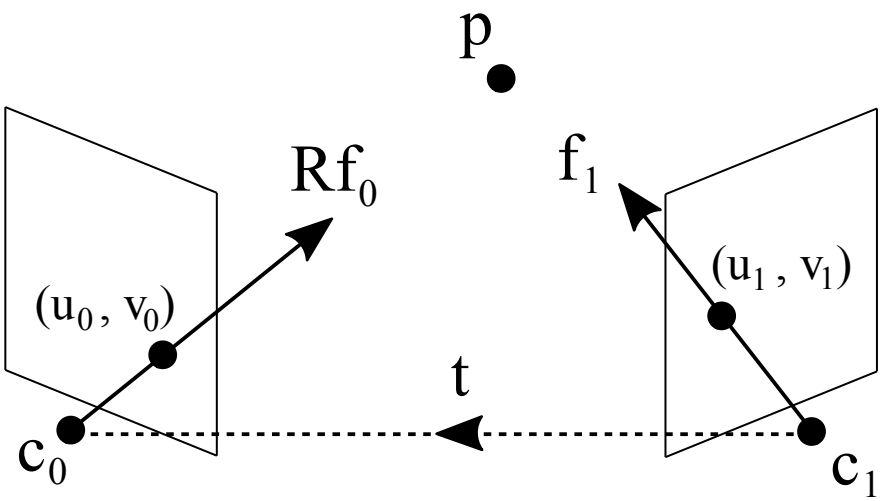
K_0 is the camera intrinsic matrix of c_0.
(x_0, y_0, z_0) is the coordinates of point p in the camera reference frame c_0.

- Epipolar error $e := \left|\mathbf{f}_1 \cdot \left(\hat{\mathbf{t}} \times \mathbf{R}\mathbf{f}_0\right)\right| = \left|\mathbf{f}_1^\top \mathbf{E}\mathbf{f}_0\right|$ → The 3rd element of f_0 and f_1 is one.

with $\mathbf{E} = [\hat{\mathbf{t}}]_\times \mathbf{R}$

- Normalized epipolar error $\hat{e} := \left|\hat{\mathbf{f}}_1 \cdot \left(\hat{\mathbf{t}} \times \mathbf{R}\hat{\mathbf{f}}_0\right)\right| = \left|\hat{\mathbf{f}}_1^\top \mathbf{E}\hat{\mathbf{f}}_0\right|$ → The magnitude of $\hat{f}_0$ and $\hat{f}_1$ is one.

2. Normalized Epipolar Error



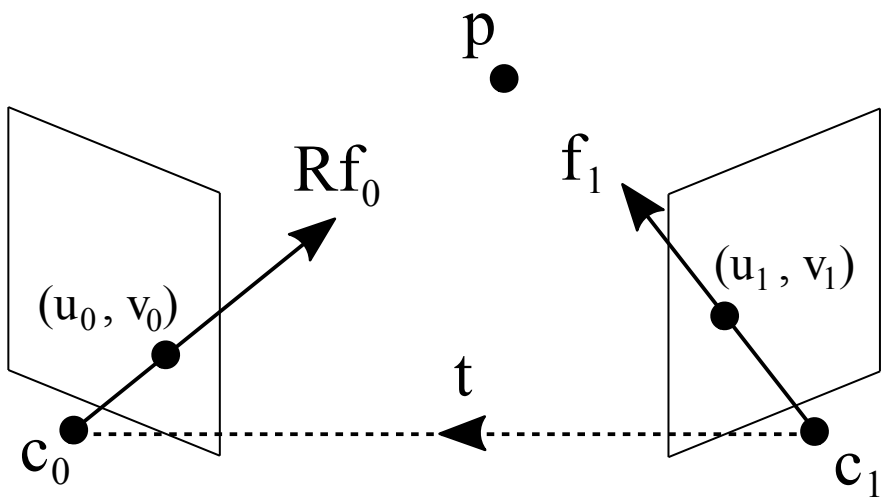
Epipolar error

$$e = \left| \mathbf{f}_1^\top \mathbf{E} \mathbf{f}_0 \right|$$

Normalized epipolar error

$$\widehat{e} = \left| \widehat{\mathbf{f}}_1^\top \mathbf{E} \widehat{\mathbf{f}}_0 \right|$$

2. Normalized Epipolar Error



Epipolar error

$$e = \left| \mathbf{f}_1^\top \mathbf{E} \mathbf{f}_0 \right|$$

Purely algebraic quantity.

Normalized epipolar error

$$\widehat{e} = \left| \widehat{\mathbf{f}}_1^\top \mathbf{E} \widehat{\mathbf{f}}_0 \right|$$

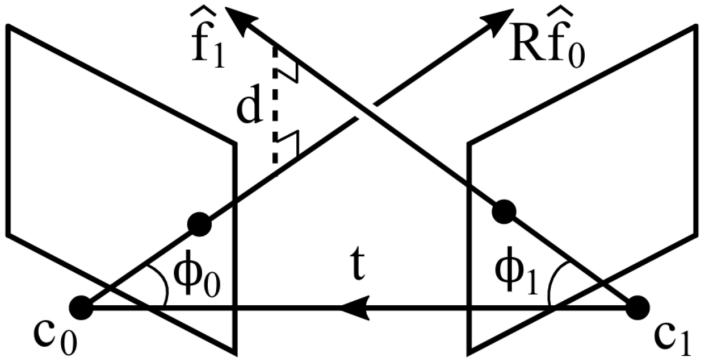
It has geometric interpretations.

2. Normalized Epipolar Error

- Interpretation 1: Relation to the distance between the backprojected rays.

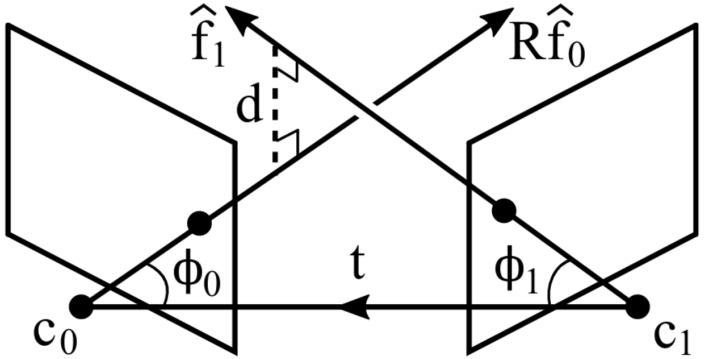
2. Normalized Epipolar Error

- Interpretation 1: Relation to the distance between the backprojected rays.



2. Normalized Epipolar Error

- Interpretation 1: Relation to the distance between the backprojected rays.

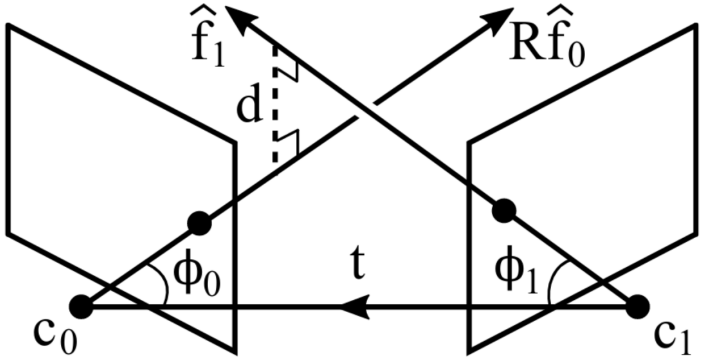


$$\widehat{e} = \frac{\sin(\beta)}{\|\mathbf{t}\|} d.$$

where $\beta := \angle(\mathbf{R}\widehat{\mathbf{f}}_0, \widehat{\mathbf{f}}_1) \in [0, \pi/2].$

2. Normalized Epipolar Error

- Interpretation 1: Relation to the distance between the backprojected rays.



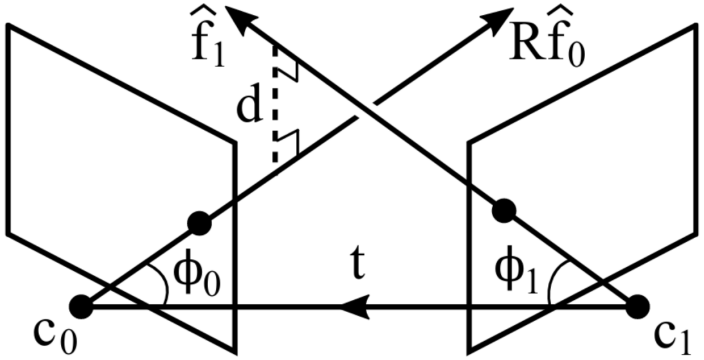
$$\widehat{e} = \frac{\sin(\beta)}{\|\mathbf{t}\|} d.$$

where $\beta := \angle(\mathbf{R}\widehat{\mathbf{f}}_0, \widehat{\mathbf{f}}_1) \in [0, \pi/2].$

2. Interpretation 2: Relation to the L_1 - optimal angular reprojection error.

2. Normalized Epipolar Error

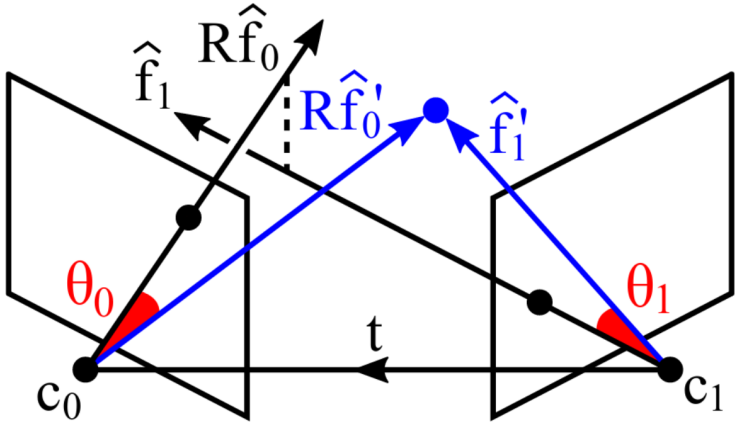
- Interpretation 1: Relation to the distance between the backprojected rays.



$$\boxed{\widehat{e} = \frac{\sin(\beta)}{\|\mathbf{t}\|} d.}$$

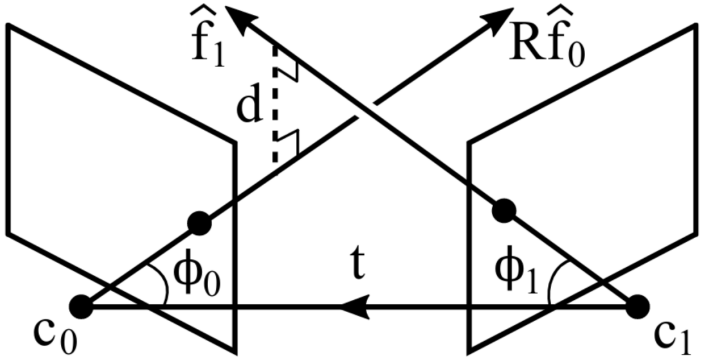
where $\beta := \angle(\mathbf{R}\widehat{\mathbf{f}}_0, \widehat{\mathbf{f}}_1) \in [0, \pi/2].$

2. Interpretation 2: Relation to the L_1 - optimal angular reprojection error.



2. Normalized Epipolar Error

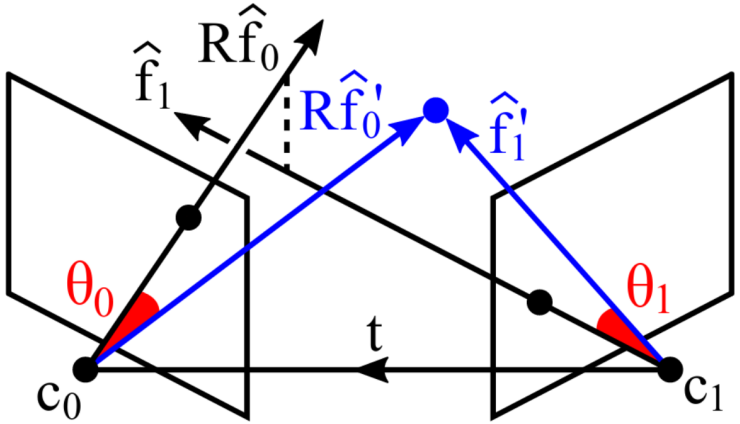
- Interpretation 1: Relation to the distance between the backprojected rays.



$$\widehat{e} = \frac{\sin(\beta)}{\|\mathbf{t}\|} d.$$

where $\beta := \angle(\mathbf{R}\widehat{\mathbf{f}}_0, \widehat{\mathbf{f}}_1) \in [0, \pi/2]$.

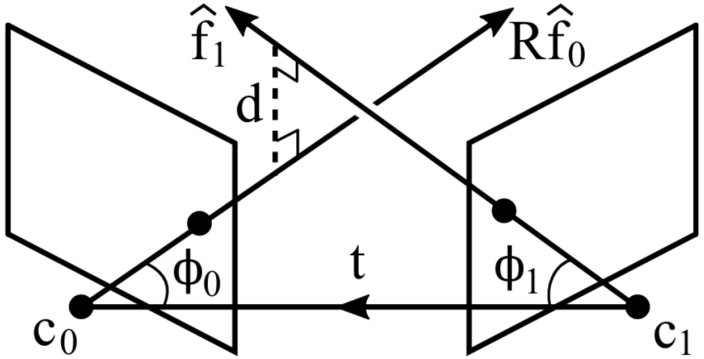
2. Interpretation 2: Relation to the L_1 - optimal angular reprojection error.



$\theta^*_{L1} = \min(\theta_0 + \theta_1)$ that make the rays intersect.

2. Normalized Epipolar Error

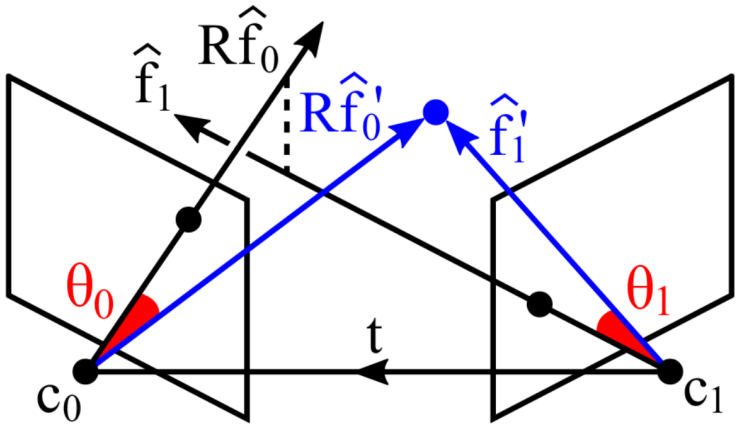
- Interpretation 1: Relation to the distance between the backprojected rays.



$$\hat{e} = \frac{\sin(\beta)}{\|\mathbf{t}\|} d.$$

where $\beta := \angle(\mathbf{R}\hat{\mathbf{f}}_0, \hat{\mathbf{f}}_1) \in [0, \pi/2].$

2. Interpretation 2: Relation to the L_1 - optimal angular reprojection error.



$\theta^*_{L1} = \min(\theta_0 + \theta_1)$ that make the rays intersect.

$$\hat{e} = \sin\left(\max(\phi_0, \phi_1)\right) \sin(\theta^*_{L1}).$$

where $\phi_0 := \angle(\mathbf{R}\hat{\mathbf{f}}_0, \hat{\mathbf{t}}) \in [0, \pi/2],$

$\phi_1 := \angle(\hat{\mathbf{f}}_1, \hat{\mathbf{t}}) \in [0, \pi/2].$

2. Normalized Epipolar Error

For the derivations and more interpretations, check our arXiv preprint "Geometric Interpretations of the Normalized Epipolar Error" (2020).

3. Two-View Rotation-Only Optimization (Kneip & Lynen 2013)

3. Two-View Rotation-Only Optimization (Kneip & Lynen 2013)

- Goal: Find the rotation and translation between camera j and k that minimize the sum of squares of the normalized epipolar errors.

3. Two-View Rotation-Only Optimization (Kneip & Lynen 2013)

- Goal: Find the rotation and translation between camera j and k that minimize the sum of squares of the normalized epipolar errors.

→ $$\underset{\mathbf{R}_{jk},\widehat{\mathbf{t}}_{jk}}{\text{argmin}} \sum_{i=1}^{m} (\hat{e}_i)^2_{(j,k)}$$

3. Two-View Rotation-Only Optimization (Kneip & Lynen 2013)

- Goal: Find the rotation and translation between camera j and k that minimize the sum of squares of the normalized epipolar errors.

→ $$\underset{\mathbf{R}_{jk},\widehat{\mathbf{t}}_{jk}}{\operatorname{argmin}} \sum_{i=1}^{m} (\hat{e}_i)^2_{(j,k)} = \sum_{i=1}^{m} \left| \widehat{\mathbf{t}}_{jk} \cdot \left(\left(\widehat{\mathbf{f}}_i \right)_j \times \mathbf{R}_{jk} \left(\widehat{\mathbf{f}}_i \right)_k \right) \right|^2$$

3. Two-View Rotation-Only Optimization (Kneip & Lynen 2013)

- Goal: Find the rotation and translation between camera j and k that minimize the sum of squares of the normalized epipolar errors.

$$\rightarrow \quad \underset{\mathbf{R}_{jk},\hat{\mathbf{t}}_{jk}}{\operatorname{argmin}} \sum_{i=1}^{m} (\hat{e}_i)^2_{(j,k)} = \sum_{i=1}^{m} \left| \hat{\mathbf{t}}_{jk} \cdot \left(\left(\hat{\mathbf{f}}_i \right)_j \times \mathbf{R}_{jk} \left(\hat{\mathbf{f}}_i \right)_k \right) \right|^2$$

$$= \hat{\mathbf{t}}_{jk}^{\top} \left[\sum_{i=1}^{m} \left(\left(\hat{\mathbf{f}}_i \right)_j \times \mathbf{R}_{jk} \left(\hat{\mathbf{f}}_i \right)_k \right) \left(\left(\hat{\mathbf{f}}_i \right)_j \times \mathbf{R}_{jk} \left(\hat{\mathbf{f}}_i \right)_k \right)^{\top} \right] \hat{\mathbf{t}}_{jk}$$

3. Two-View Rotation-Only Optimization (Kneip & Lynen 2013)

- Goal: Find the rotation and translation between camera j and k that minimize the sum of squares of the normalized epipolar errors.

$$\rightarrow \quad \underset{\mathbf{R}_{jk},\hat{\mathbf{t}}_{jk}}{\operatorname{argmin}} \sum_{i=1}^{m} (\hat{e}_i)^2_{(j,k)} = \sum_{i=1}^{m} \left| \hat{\mathbf{t}}_{jk} \cdot \left(\left(\hat{\mathbf{f}}_i \right)_j \times \mathbf{R}_{jk} \left(\hat{\mathbf{f}}_i \right)_k \right) \right|^2$$

$$= \hat{\mathbf{t}}_{jk}^{\top} \left[\sum_{i=1}^{m} \left(\left(\hat{\mathbf{f}}_i \right)_j \times \mathbf{R}_{jk} \left(\hat{\mathbf{f}}_i \right)_k \right) \left(\left(\hat{\mathbf{f}}_i \right)_j \times \mathbf{R}_{jk} \left(\hat{\mathbf{f}}_i \right)_k \right)^{\top} \right] \hat{\mathbf{t}}_{jk}$$

$$= \hat{\mathbf{t}}_{jk}^{\top} \mathbf{M}_{jk} \hat{\mathbf{t}}_{jk}$$

3. Two-View Rotation-Only Optimization (Kneip & Lynen 2013)

- Goal: Find the rotation and translation between camera j and k that minimize the sum of squares of the normalized epipolar errors.

→ $$\underset{\mathbf{R}_{jk},\hat{\mathbf{t}}_{jk}}{\operatorname{argmin}} \sum_{i=1}^{m} (\hat{e}_i)^2_{(j,k)} = \sum_{i=1}^{m} \left| \hat{\mathbf{t}}_{jk} \cdot \left((\hat{\mathbf{f}}_i)_j \times \mathbf{R}_{jk} (\hat{\mathbf{f}}_i)_k \right) \right|^2$$

$$= \hat{\mathbf{t}}_{jk}^{\top} \left[\sum_{i=1}^{m} \left((\hat{\mathbf{f}}_i)_j \times \mathbf{R}_{jk} (\hat{\mathbf{f}}_i)_k \right) \left((\hat{\mathbf{f}}_i)_j \times \mathbf{R}_{jk} (\hat{\mathbf{f}}_i)_k \right)^{\top} \right] \hat{\mathbf{t}}_{jk}$$

$$= \hat{\mathbf{t}}_{jk}^{\top} \mathbf{M}_{jk} \hat{\mathbf{t}}_{jk}$$ (where $\mathbf{M}_{jk}$ is a positive semidefinite matrix that depends only on $\mathbf{R}_{jk}$)

$$\underset{\mathbf{R}_{jk},\widehat{\mathbf{t}}_{jk}}{\operatorname{argmin}} \sum_{i=1}^{m} (\hat{e}_i)^2_{(j,k)} = \underset{\mathbf{R}_{jk},\widehat{\mathbf{t}}_{jk}}{\operatorname{argmin}} \widehat{\mathbf{t}}_{jk}^{\top} \mathbf{M}_{jk} \widehat{\mathbf{t}}_{jk} \ (\mathbf{M}_{jk} \geq 0 \text{ depends only on } \mathbf{R}_{jk})$$

$$\underset{\mathbf{R}_{jk},\widehat{\mathbf{t}}_{jk}}{\operatorname{argmin}} \sum_{i=1}^{m} (\hat{e}_i)^2_{(j,k)} = \underset{\mathbf{R}_{jk},\widehat{\mathbf{t}}_{jk}}{\operatorname{argmin}} \widehat{\mathbf{t}}_{jk}^{\top} \mathbf{M}_{jk} \widehat{\mathbf{t}}_{jk} \ (\mathbf{M}_{jk} \geq 0 \text{ depends only on } \mathbf{R}_{jk})$$

Linear algebra tells us that its minimum is equal to the smallest eigenvalue (λ) of $\mathbf{M}_{jk}$, and this is achieved when $\widehat{\mathbf{t}}_{jk}$ is the eigenvector corresponding to λ!

$$\underset{\mathbf{R}_{jk},\widehat{\mathbf{t}}_{jk}}{\operatorname{argmin}} \sum_{i=1}^{m} (\hat{e}_i)^2_{(j,k)} = \underset{\mathbf{R}_{jk},\widehat{\mathbf{t}}_{jk}}{\operatorname{argmin}} \widehat{\mathbf{t}}_{jk}^{\top} \mathbf{M}_{jk} \widehat{\mathbf{t}}_{jk} \ (\mathbf{M}_{jk} \geq 0 \text{ depends only on } \mathbf{R}_{jk})$$

Linear algebra tells us that its minimum is equal to the smallest eigenvalue (λ) of $\mathbf{M}_{jk}$, and this is achieved when $\widehat{\mathbf{t}}_{jk}$ is the eigenvector corresponding to λ!

→ Minimize $\sum_{i=1}^{m} (\hat{e}_i)^2_{(j,k)}$ = Minimize the smallest eigenvalue of $\mathbf{M}_{jk}$

$$\underset{\mathbf{R}_{jk},\,\widehat{\mathbf{t}}_{jk}}{\operatorname{argmin}} \sum_{i=1}^{m} (\hat{e}_i)^2_{(j,k)} = \underset{\mathbf{R}_{jk},\,\widehat{\mathbf{t}}_{jk}}{\operatorname{argmin}} \widehat{\mathbf{t}}_{jk}^{\top} \mathbf{M}_{jk} \widehat{\mathbf{t}}_{jk} \; (\mathbf{M}_{jk} \geq 0 \text{ depends only on } \mathbf{R}_{jk})$$

Linear algebra tells us that its minimum is equal to the smallest eigenvalue (λ) of $\mathbf{M}_{jk}$, and this is achieved when $\widehat{\mathbf{t}}_{jk}$ is the eigenvector corresponding to λ!

→ Minimize $\sum_{i=1}^{m} (\hat{e}_i)^2_{(j,k)}$ = Minimize the smallest eigenvalue of $\mathbf{M}_{jk}$

Recall $\mathbf{M}_{jk} = \sum_{i=1}^{m} \left(\left(\widehat{\mathbf{f}}_i \right)_j \times \mathbf{R}_{jk} \left(\widehat{\mathbf{f}}_i \right)_k \right) \left(\left(\widehat{\mathbf{f}}_i \right)_j \times \mathbf{R}_{jk} \left(\widehat{\mathbf{f}}_i \right)_k \right)^{\top}$ depends only on $\mathbf{R}_{jk}$.

$$\underset{\mathbf{R}_{jk},\,\widehat{\mathbf{t}}_{jk}}{\operatorname{argmin}} \sum_{i=1}^{m} (\hat{e}_i)^2_{(j,k)} = \underset{\mathbf{R}_{jk},\,\widehat{\mathbf{t}}_{jk}}{\operatorname{argmin}} \widehat{\mathbf{t}}_{jk}^{\top} \mathbf{M}_{jk} \widehat{\mathbf{t}}_{jk} \; (\mathbf{M}_{jk} \geq 0 \text{ depends only on } \mathbf{R}_{jk})$$

Linear algebra tells us that its minimum is equal to the smallest eigenvalue (λ) of $\mathbf{M}_{jk}$, and this is achieved when $\widehat{\mathbf{t}}_{jk}$ is the eigenvector corresponding to λ!

→ Minimize $\sum_{i=1}^{m} (\hat{e}_i)^2_{(j,k)}$ = Minimize the smallest eigenvalue of $\mathbf{M}_{jk}$

Recall $\mathbf{M}_{jk} = \sum_{i=1}^{m} \left(\left(\widehat{\mathbf{f}}_i \right)_j \times \mathbf{R}_{jk} \left(\widehat{\mathbf{f}}_i \right)_k \right) \left(\left(\widehat{\mathbf{f}}_i \right)_j \times \mathbf{R}_{jk} \left(\widehat{\mathbf{f}}_i \right)_k \right)^{\top}$ depends only on $\mathbf{R}_{jk}$.

So, now the problem is: Find $\mathbf{R}_{jk}$ that minimizes λ of $\mathbf{M}_{jk}$.

$$\underset{\mathbf{R}_{jk},\,\widehat{\mathbf{t}}_{jk}}{\operatorname{argmin}} \sum_{i=1}^{m} (\hat{e}_i)^2_{(j,k)} = \underset{\mathbf{R}_{jk},\,\widehat{\mathbf{t}}_{jk}}{\operatorname{argmin}} \widehat{\mathbf{t}}_{jk}^{\top} \mathbf{M}_{jk} \widehat{\mathbf{t}}_{jk} \; (\mathbf{M}_{jk} \geq 0 \text{ depends only on } \mathbf{R}_{jk})$$

Linear algebra tells us that its minimum is equal to the smallest eigenvalue (λ) of $\mathbf{M}_{jk}$, and this is achieved when $\widehat{\mathbf{t}}_{jk}$ is the eigenvector corresponding to λ!

→ Minimize $\sum_{i=1}^{m} (\hat{e}_i)^2_{(j,k)}$ = Minimize the smallest eigenvalue of $\mathbf{M}_{jk}$

Recall $\mathbf{M}_{jk} = \sum_{i=1}^{m} \left(\left(\widehat{\mathbf{f}}_i \right)_j \times \mathbf{R}_{jk} \left(\widehat{\mathbf{f}}_i \right)_k \right) \left(\left(\widehat{\mathbf{f}}_i \right)_j \times \mathbf{R}_{jk} \left(\widehat{\mathbf{f}}_i \right)_k \right)^{\top}$ depends only on $\mathbf{R}_{jk}$.

So, now the problem is: Find $\mathbf{R}_{jk}$ that minimizes λ of $\mathbf{M}_{jk}$.

→ $\underset{\mathbf{R}_{jk}}{\operatorname{argmin}} \lambda_{\mathbf{M}}(\mathbf{R}_{jk})$

$$\underset{\mathbf{R}_{jk},\,\widehat{\mathbf{t}}_{jk}}{\operatorname{argmin}} \sum_{i=1}^{m} (\hat{e}_i)^2_{(j,k)} = \underset{\mathbf{R}_{jk},\,\widehat{\mathbf{t}}_{jk}}{\operatorname{argmin}} \widehat{\mathbf{t}}_{jk}^{\top} \mathbf{M}_{jk} \widehat{\mathbf{t}}_{jk} \; (\mathbf{M}_{jk} \geq 0 \text{ depends only on } \mathbf{R}_{jk})$$

Linear algebra tells us that its minimum is equal to the smallest eigenvalue (λ) of $\mathbf{M}_{jk}$, and this is achieved when $\widehat{\mathbf{t}}_{jk}$ is the eigenvector corresponding to λ!

→ Minimize $\sum_{i=1}^{m} (\hat{e}_i)^2_{(j,k)}$ = Minimize the smallest eigenvalue of $\mathbf{M}_{jk}$

Recall $\mathbf{M}_{jk} = \sum_{i=1}^{m} \left((\widehat{\mathbf{f}}_i)_j \times \mathbf{R}_{jk} (\widehat{\mathbf{f}}_i)_k \right) \left((\widehat{\mathbf{f}}_i)_j \times \mathbf{R}_{jk} (\widehat{\mathbf{f}}_i)_k \right)^{\top}$ depends only on $\mathbf{R}_{jk}$.

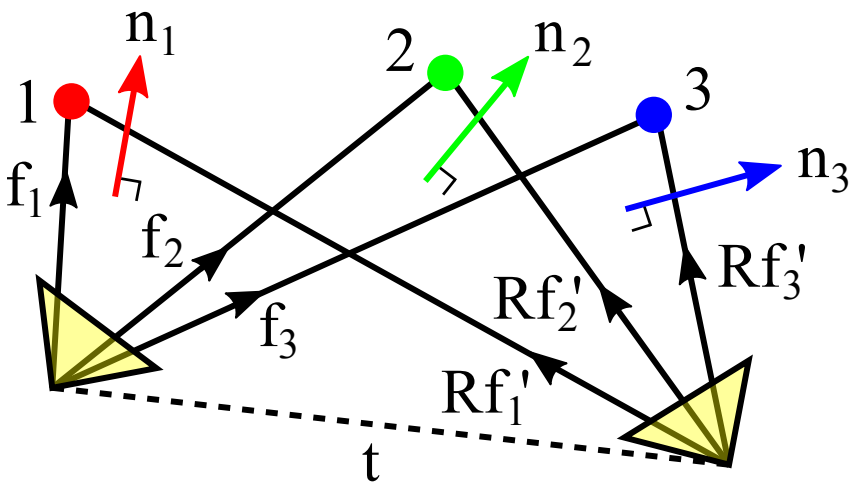
So, now the problem is: Find $\mathbf{R}_{jk}$ that minimizes λ of $\mathbf{M}_{jk}$.

→ $\underset{\mathbf{R}_{jk}}{\operatorname{argmin}} \lambda_{\mathbf{M}}(\mathbf{R}_{jk})$

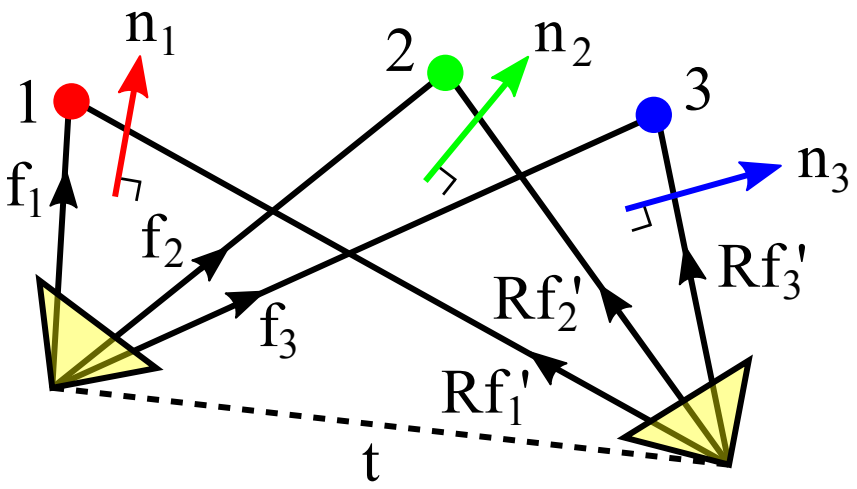
This transforms a relative pose problem into a rotation-only problem!

$$\underset{\mathbf{R}_{jk}}{\operatorname{argmin}}\, \lambda_{\mathbf{M}}(\mathbf{R}_{jk}) \longrightarrow$$ But what does it mean geometrically?

$\underset{\mathbf{R}_{jk}}{\operatorname{argmin}} \lambda_{\mathbf{M}}(\mathbf{R}_{jk})$ ⟶ But what does it mean geometrically?

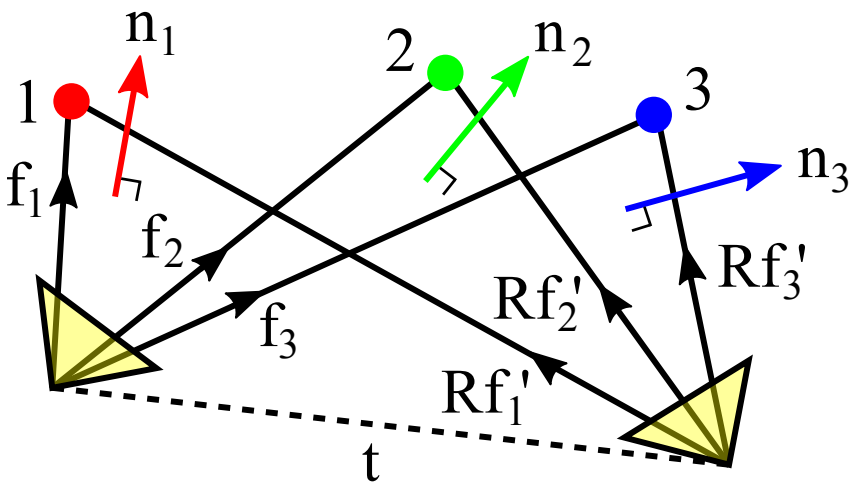


$$\underset{\mathbf{R}_{jk}}{\operatorname{argmin}}\, \lambda_{\mathbf{M}}(\mathbf{R}_{jk}) \longrightarrow$$ But what does it mean geometrically?



If all measurements were accurate, then the normals of the epipolar planes (n_i) would be coplanar, because they are all perpendicular to the translation vector (t).

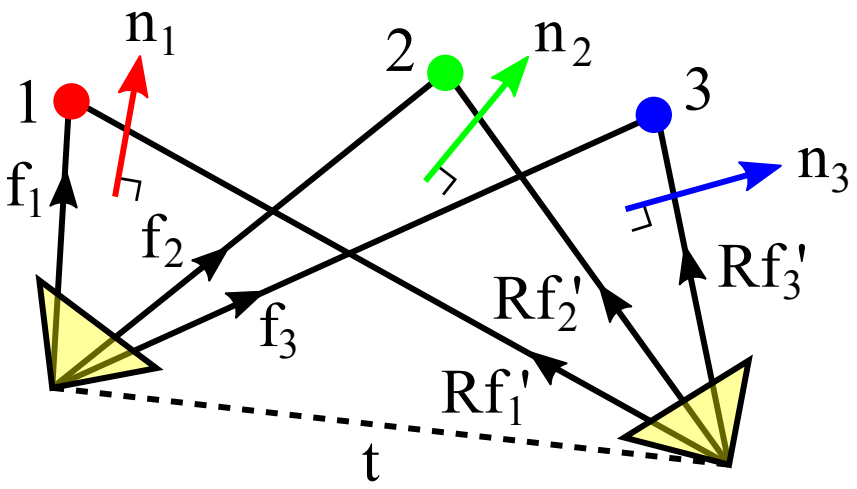
$$\underset{\mathbf{R}_{jk}}{\operatorname{argmin}}\, \lambda_{\mathbf{M}}(\mathbf{R}_{jk}) \longrightarrow$$ But what does it mean geometrically?



If all measurements were accurate, then the normals of the epipolar planes (n_i) would be coplanar, because they are all perpendicular to the translation vector (t).

In practice, this does not happen due to noise.

$\underset{\mathbf{R}_{jk}}{\operatorname{argmin}} \lambda_{\mathbf{M}}(\mathbf{R}_{jk})$ ⟶ But what does it mean geometrically?

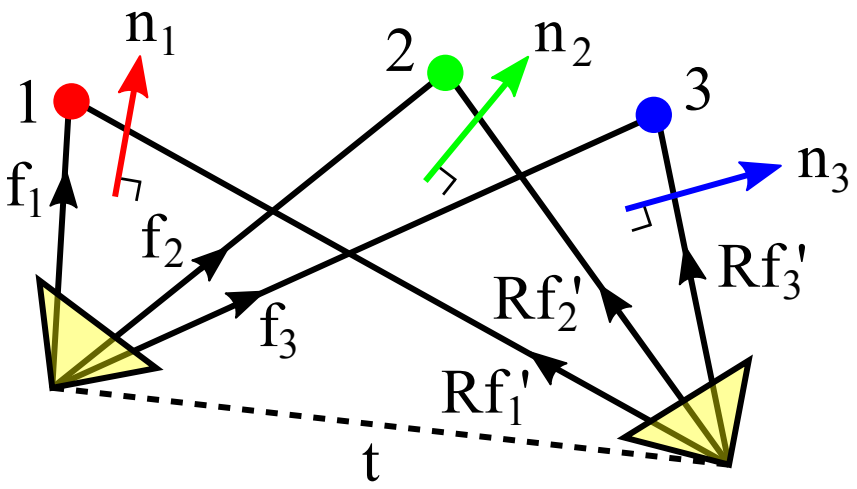


If all measurements were accurate, then the normals of the epipolar planes (n_i) would be coplanar, because they are all perpendicular to the translation vector (t).

In practice, this does not happen due to noise.

So, instead, we find R that makes the normals as coplanar as possible.

$\underset{\mathbf{R}_{jk}}{\operatorname{argmin}} \lambda_{\mathbf{M}}(\mathbf{R}_{jk})$ ⟶ But what does it mean geometrically?



If all measurements were accurate, then the normals of the epipolar planes (n_i) would be coplanar, because they are all perpendicular to the translation vector (t).

In practice, this does not happen due to noise.

So, instead, we find R that makes the normals as coplanar as possible. This leads to the eigenvalue minimization problem above, where

$$M = \sum_i n_i n_i^T = \left(\hat{f}_i \times R\hat{f}_i'\right)\left(\hat{f}_i \times R\hat{f}_i'\right)^T$$

4. Rotation-Only Bundle Adjustment

$$\underset{\mathbf{R}_{jk}}{\operatorname{argmin}}\, \lambda_{\mathbf{M}}(\mathbf{R}_{jk})$$

Using this formulation, Kneip & Lynen (2013) estimated the rotation, independently of the translation.

4. Rotation-Only Bundle Adjustment

$$\underset{\mathbf{R}_{jk}}{\operatorname{argmin}} \lambda_{\mathbf{M}}(\mathbf{R}_{jk})$$

Using this formulation, Kneip & Lynen (2013) estimated the rotation, independently of the translation.

However, this is limited to only two views (camera j and k).

4. Rotation-Only Bundle Adjustment

$$\underset{\mathbf{R}_{jk}}{\operatorname{argmin}} \lambda_{\mathbf{M}}(\mathbf{R}_{jk})$$

Using this formulation, Kneip & Lynen (2013) estimated the rotation, independently of the translation.

However, this is limited to only two views (camera j and k).

In our work, we extend their idea to multiple views.

4. Rotation-Only Bundle Adjustment

$$\underset{\mathbf{R}_{jk}}{\operatorname{argmin}} \lambda_{\mathbf{M}}(\mathbf{R}_{jk})$$

Using this formulation, Kneip & Lynen (2013) estimated the rotation, independently of the translation.

However, this is limited to only two views (camera j and k).

In our work, we extend their idea to multiple views.

1. $\underset{\mathbf{R}_1,\cdots,\mathbf{R}_n}{\operatorname{argmin}} \sum_{(j,k)\in\mathcal{E}} \sqrt{\lambda_{\mathbf{M}}(\mathbf{R}_{jk})}$ where $\mathcal{E}$ is the set of all camera pairs observing more than 10 points in common.
2. Solve it using the Adam optimizer (Kingma & Ba, 2015).

5. Results

- Comparing the two approaches:

Rotation Averaging	vs	Rotation Averaging + Rotation-Only Bundle Adjustment
"RA"		"RA + ROBA"

5. Results

- Comparing the two approaches:

Rotation Averaging	vs	Rotation Averaging + Rotation-Only Bundle Adjustment
"RA"		"RA + ROBA"

For rotation averaging, we used the state-of-the-art method by Chatterjee & Govindu, 2018.

5. Results

5. Results

(1) $$\underset{\mathbf{R}_1,\cdots,\mathbf{R}_n}{\operatorname{argmin}} \sum_{(j,k)\in\mathcal{E}} \sqrt{\lambda_{\mathbf{M}}(\mathbf{R}_{jk})}$$

5. Results

(1) $\underset{\mathbf{R}_1,\cdots,\mathbf{R}_n}{\operatorname{argmin}} \sum_{(j,k)\in\mathcal{E}} \sqrt{\lambda_{\mathbf{M}}(\mathbf{R}_{jk})}$ ⟶ Square-rooting improves the results!

5. Results

(1) $\underset{\mathbf{R}_1,\cdots,\mathbf{R}_n}{\operatorname{argmin}} \sum_{(j,k)\in\mathcal{E}} \sqrt{\lambda_{\mathbf{M}}(\mathbf{R}_{jk})}$ $\longrightarrow$

Square-rooting improves the results!

It works as a robust cost function, i.e.,

$L_{0.5}$-norm, downweighting large residuals.

5. Results

(1) $\underset{\mathbf{R}_1,\cdots,\mathbf{R}_n}{\operatorname{argmin}} \sum_{(j,k)\in\mathcal{E}} \sqrt{\lambda_{\mathbf{M}}(\mathbf{R}_{jk})}$ ⟶ Square-rooting improves the results!

It works as a robust cost function, i.e.,

$L_{0.5}$-norm, downweighting large residuals.

(2) We tried Gauss-Newton and Levenberg-Marquardt method.

5. Results

(1) $\underset{\mathbf{R}_1,\cdots,\mathbf{R}_n}{\operatorname{argmin}} \sum_{(j,k)\in\mathcal{E}} \sqrt{\lambda_{\mathbf{M}}(\mathbf{R}_{jk})}$ ⟶ Square-rooting improves the results!

It works as a robust cost function, i.e.,

$L_{0.5}$-norm, downweighting large residuals.

(2) We tried Gauss-Newton and Levenberg-Marquardt method.

But they often got stuck at local minima near the initial seed.

5. Results

(1) $\underset{\mathbf{R}_1,\cdots,\mathbf{R}_n}{\operatorname{argmin}} \sum_{(j,k)\in\mathcal{E}} \sqrt{\lambda_{\mathbf{M}}(\mathbf{R}_{jk})}$ ⟶ Square-rooting improves the results!

It works as a robust cost function, i.e.,

$L_{0.5}$-norm, downweighting large residuals.

(2) We tried Gauss-Newton and Levenberg-Marquardt method.

But they often got stuck at local minima near the initial seed.

The Adam optimizer was able to find a better solution.

5. Results

(1) $\underset{\mathbf{R}_1,\cdots,\mathbf{R}_n}{\operatorname{argmin}} \sum_{(j,k)\in\mathcal{E}} \sqrt{\lambda_{\mathbf{M}}(\mathbf{R}_{jk})}$ ⟶ Square-rooting improves the results!

It works as a robust cost function, i.e.,

$L_{0.5}$-norm, downweighting large residuals.

(2) We tried Gauss-Newton and Levenberg-Marquardt method.

But they often got stuck at local minima near the initial seed.

The Adam optimizer was able to find a better solution.

Our speculation: It could escape the weak local minima more easily

by taking into account the previous gradients (momentum).

5. Results

(1) $\underset{\mathbf{R}_1,\cdots,\mathbf{R}_n}{\operatorname{argmin}} \sum_{(j,k)\in\mathcal{E}} \sqrt{\lambda_{\mathbf{M}}(\mathbf{R}_{jk})}$ ⟶ Square-rooting improves the results!

It works as a robust cost function, i.e.,

$L_{0.5}$-norm, downweighting large residuals.

(2) We tried Gauss-Newton and Levenberg-Marquardt method.

But they often got stuck at local minima near the initial seed.

The Adam optimizer was able to find a better solution.

Our speculation: It could escape the weak local minima more easily

by taking into account the previous gradients (momentum).

→ Not reported in the paper, as we found this during the preliminary phase...

5. Results

(1) $\underset{\mathbf{R}_1,\cdots,\mathbf{R}_n}{\operatorname{argmin}} \sum_{(j,k)\in\mathcal{E}} \sqrt{\lambda_{\mathbf{M}}(\mathbf{R}_{jk})}$ ⟶ Square-rooting improves the results!
It works as a robust cost function, i.e.,
$L_{0.5}$-norm, downweighting large residuals.

(2) We tried Gauss-Newton and Levenberg-Marquardt method.
But they often got stuck at local minima near the initial seed.
The Adam optimizer was able to find a better solution.
Our speculation: It could escape the weak local minima more easily
by taking into account the previous gradients (momentum).
→ Not reported in the paper, as we found this during the preliminary phase...

(3) The Adam optimizer uses the first-order gradient, and we use some approximation to speed up the computation.

5. Results

(1) $\underset{\mathbf{R}_1,\cdots,\mathbf{R}_n}{\operatorname{argmin}} \sum_{(j,k)\in\mathcal{E}} \sqrt{\lambda_{\mathbf{M}}(\mathbf{R}_{jk})}$ ⟶ Square-rooting improves the results!
It works as a robust cost function, i.e.,
$L_{0.5}$-norm, downweighting large residuals.

(2) We tried Gauss-Newton and Levenberg-Marquardt method.
But they often got stuck at local minima near the initial seed.
The Adam optimizer was able to find a better solution.
Our speculation: It could escape the weak local minima more easily
by taking into account the previous gradients (momentum).
→ Not reported in the paper, as we found this during the preliminary phase...

(3) The Adam optimizer uses the first-order gradient, and we use some approximation to speed up the computation. → Slightly less accuracte, but much faster (×1.8).

5. Results

- Synthetic data

5. Results

- Synthetic data
- We tested 12 different configurations.

5. Results

- Synthetic data
- We tested 12 different configurations.
- For each configuration, we ran 100 independent simulations.

5. Results

- Synthetic data
- We tested 12 different configurations.
- For each configuration, we ran 100 independent simulations.
- In **1200 out of 1200** simulations, ROBA improved the results.

5. Results

- Synthetic data
- We tested 12 different configurations.
- For each configuration, we ran 100 independent simulations.
- In **1200 out of 1200** simulations, ROBA improved the results.
- On the right, we present the median rotation errors (in deg).

	RA	RA + ROBA
Baseline	2.31	**1.26**
More points	1.78	**1.05**
Fewer views	1.08	**0.33**
More views	4.35	**3.73**
Closer points	5.74	**3.99**
Farther points	0.69	**0.36**
Less noise	2.30	**1.17**
More noise	2.31	**1.48**
Planar scene	1.74	**0.80**
Pure rotations	0.045	**0.026**
Pure + Planar	0.045	**0.027**
Mixed rotations	0.17	**0.069**

5. Results

- Synthetic data
- We tested 12 different configurations.
- For each configuration, we ran 100 independent simulations.
- In **1200 out of 1200** simulations, ROBA improved the results.
- On the right, we present the median rotation errors (in deg).
- The more densely connected the graph, the better our method works.

	RA	RA + ROBA
Baseline	2.31	**1.26**
More points	1.78	**1.05**
Fewer views	1.08	**0.33**
More views	4.35	**3.73**
Closer points	5.74	**3.99**
Farther points	0.69	**0.36**
Less noise	2.30	**1.17**
More noise	2.31	**1.48**
Planar scene	1.74	**0.80**
Pure rotations	0.045	**0.026**
Pure + Planar	0.045	**0.027**
Mixed rotations	0.17	**0.069**

5. Results

- Synthetic data
- We tested 12 different configurations.
- For each configuration, we ran 100 independent simulations.
- In **1200 out of 1200** simulations, ROBA improved the results.
- On the right, we present the median rotation errors (in deg).
- The more densely connected the graph, the better our method works.
- It is robust to challenging configurations such as planar scenes and pure rotations.

	RA	RA + ROBA
Baseline	2.31	**1.26**
More points	1.78	**1.05**
Fewer views	1.08	**0.33**
More views	4.35	**3.73**
Closer points	5.74	**3.99**
Farther points	0.69	**0.36**
Less noise	2.30	**1.17**
More noise	2.31	**1.48**
Planar scene	1.74	**0.80**
Pure rotations	0.045	**0.026**
Pure + Planar	0.045	**0.027**
Mixed rotations	0.17	**0.069**

5. Results

- Real data (Wilson & Snavely 2014)

5. Results

- Real data (Wilson & Snavely 2014)
- In all 15 datasets, ROBA improved the results.

5. Results

- Real data (Wilson & Snavely 2014)
- In all 15 datasets, ROBA improved the results.
- The rotation errors (in deg) are shown in black, and the computation times (in sec) are given in blue.

	RA	RA + ROBA
ALM	4.08 (16)	**2.35** (267)
ELS	2.10 (1)	**1.06** (48)
GDM	6.05 (3)	**2.43** (87)
MDR	6.20 (2)	**4.42** (61)
MND	1.46 (4)	**0.82** (142)
NTD	2.08 (14)	**1.27** (291)
NYC	2.87 (1)	**1.03** (49)
PDP	3.86 (1)	**2.18** (61)
PIC	4.14 (220)	**1.58** (899)
ROF	2.94 (6)	**2.18** (186)
TOL	3.83 (1)	**1.15** (64)
TFG	3.40 (553)	**2.76** (2194)
USQ	5.59 (1)	**3.26** (62)
VNC	6.12 (15)	**4.96** (289)
YKM	3.67 (1)	**1.66** (74)

6. Limitations and Future Work

6. Limitations and Future Work

- Our current method assumes that there are no outliers in the input.

6. Limitations and Future Work

- Our current method assumes that there are no outliers in the input.

  Specifically, we assumed that

  (1) the outliers in the point matches were removed by a robust pose estimator,

6. Limitations and Future Work

- Our current method assumes that there are no outliers in the input.

  Specifically, we assumed that

  (1) the outliers in the point matches were removed by a robust pose estimator,

  (2) the outliers in the relative rotations were handled by robust rotation averaging.

6. Limitations and Future Work

- Our current method assumes that there are no outliers in the input.

  Specifically, we assumed that

  (1) the outliers in the point matches were removed by a robust pose estimator,

  (2) the outliers in the relative rotations were handled by robust rotation averaging.

  One could enhance the robustness against outliers by incorporating a robust cost function (e.g., Huber loss).

6. Limitations and Future Work

- Our current method assumes that there are no outliers in the input.

  Specifically, we assumed that

  (1) the outliers in the point matches were removed by a robust pose estimator,

  (2) the outliers in the relative rotations were handled by robust rotation averaging.

  One could enhance the robustness against outliers by incorporating a robust cost function (e.g., Huber loss).

- We have not evaluated the resultant impact on the full bundle adjustment.

6. Limitations and Future Work

- Our current method assumes that there are no outliers in the input.

  Specifically, we assumed that

  (1) the outliers in the point matches were removed by a robust pose estimator,

  (2) the outliers in the relative rotations were handled by robust rotation averaging.

  One could enhance the robustness against outliers by incorporating a robust cost function (e.g., Huber loss).

- We have not evaluated the resultant impact on the full bundle adjustment.

- We have not thoroughly explored different optimization methods.

7. Conclusions

7. Conclusions

- We proposed rotation-only bundle adjustment, a novel method for estimating the global rotations of multiple views, independently of the translations and the scene structure.

7. Conclusions

- We proposed rotation-only bundle adjustment, a novel method for estimating the global rotations of multiple views, independently of the translations and the scene structure.

- We formulated the optimization problem by extending the two-view rotation-only method of Kneip & Lynen (2013) to multiple views.

7. Conclusions

- We proposed rotation-only bundle adjustment, a novel method for estimating the global rotations of multiple views, independently of the translations and the scene structure.

- We formulated the optimization problem by extending the two-view rotation-only method of Kneip & Lynen (2013) to multiple views.

- We solved the problem using the Adam optimizer, and showed that it consistently improves the accuracy when used after rotation averaging.