Semester Project

-

Map Fusion for Collaborative UAV SLAM

-

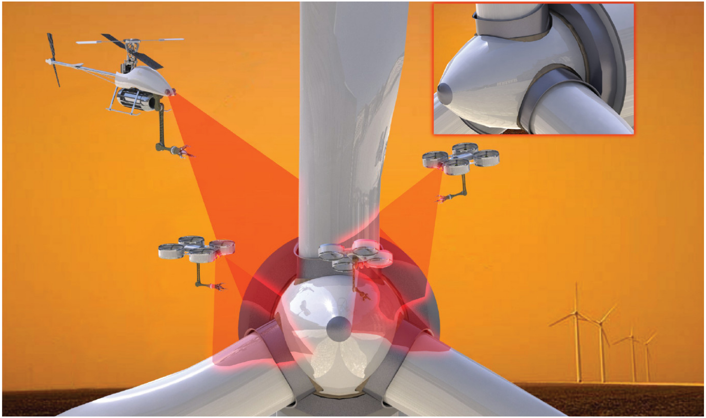
ETH zürich
V4RL VISION FOR ROBOTICS LAB
Autonomous Systems Lab

Contents

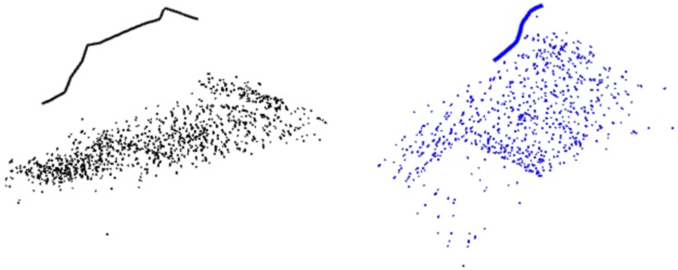
Acronyms

SLAM	Simultaneous Localisation and Mapping.
UAV	Unmanned Aerial Vehicle.
KF	KeyFrame.
KFM	KeyFrame Match.
BA	Bundle Adjustment.
PGO	Pose Graph Optimization.
LM	Levenberg-Marquardt.
DL	Powell's dog leg.

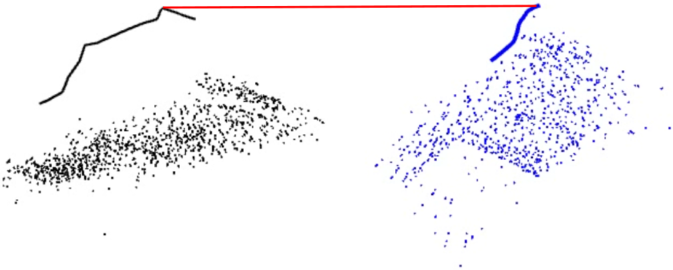
Introduction



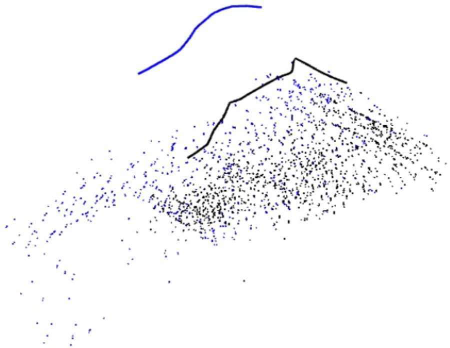
Introduction



Introduction



Introduction



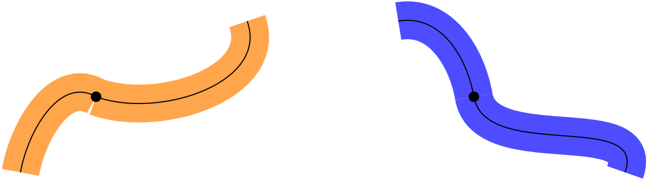
Introduction - What is a KeyFrame Match (KFM)?

KeyFrames (KFs): The most “representative” poses

Introduction - What is a KeyFrame Match (KFM)?

KeyFrames (KFs): The most “representative” poses

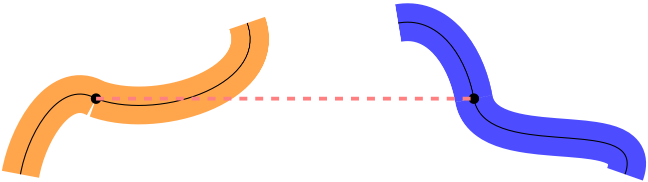
Two clients each with own landmarks and KeyFrames (KFs)



Introduction - What is a KeyFrame Match (KFM)?

KeyFrames (KFs): The most “representative” poses

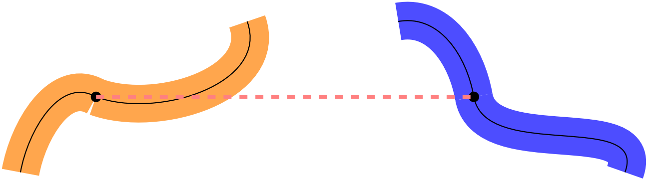
KeyFrame Match (KFM): Two KeyFrames (KFs) observing the same location



Introduction - What is a KeyFrame Match (KFM)?

KeyFrames (KFs): The most "representative" poses

KeyFrame Match (KFM): Two KeyFrames (KFs) observing the same location → Can obtain transformation

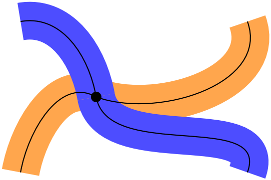


Introduction - What is a KeyFrame Match (KFM)?

KeyFrames (KFs): The most “representative” poses

KeyFrame Match (KFM): Two KeyFrames (KFs) observing the same location

With the transformation → maps can be aligned

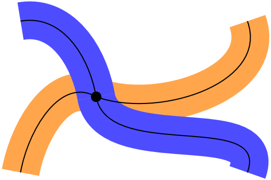


Introduction - What is a KeyFrame Match (KFM)?

KeyFrames (KFs): The most "representative" poses

A KeyFrame Match (KFM) contains:

- Two KeyFrames (KFs) (One per map)
- The transformation ($T \in \mathrm{Sim}(3)$) between them



Motivation

- A multi agent SLAM system based on ORB-SLAM2 should be extended

[Mur-Artal and Tardos, 2016]

Motivation

- A multi agent SLAM system based on ORB-SLAM2 should be extended
- So far, as soon as a KeyFrame Match (KFM) was detected, maps were merged

[Mur-Artal and Tardos, 2016]

Motivation

- A multi agent SLAM system based on ORB-SLAM2 should be extended
- So far, as soon as a KeyFrame Match (KFM) was detected, maps were merged
- Using multiple KFMs to guarantee no false map merging

[Mur-Artal and Tardos, 2016]

Motivation

- A multi agent SLAM system based on ORB-SLAM2 should be extended
- So far, as soon as a KeyFrame Match (KFM) was detected, maps were merged
- Using multiple KFMs to guarantee no false map merging
- Using multiple KFMs to obtain an optimal map alignment

[Mur-Artal and Tardos, 2016]

Map merging - Old approach

Old approach:

- As soon as a KFM was detected, maps were merged

Map merging - New approach

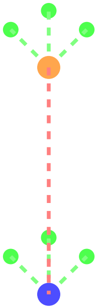
Find $n(=3)$ KeyFrame Matches (KFMs)

KeyFrames of a Match
Skipped KeyFrames
Map Points (Landmarks)

Map merging - New approach

Find $n(=3)$ KeyFrame Matches (KFMs),
Skip $m(=5)$ KeyFrames (KFs)

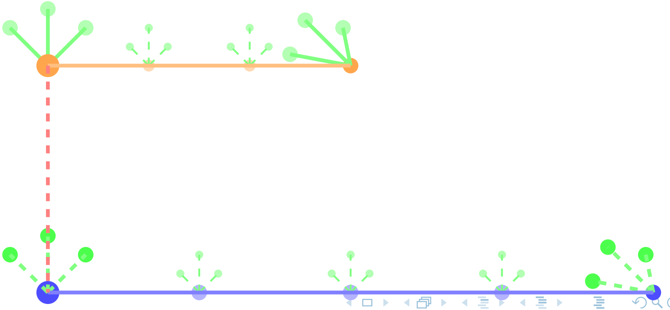
KeyFrames of a Match
Skipped KeyFrames
Map Points (Landmarks)



Map merging - New approach

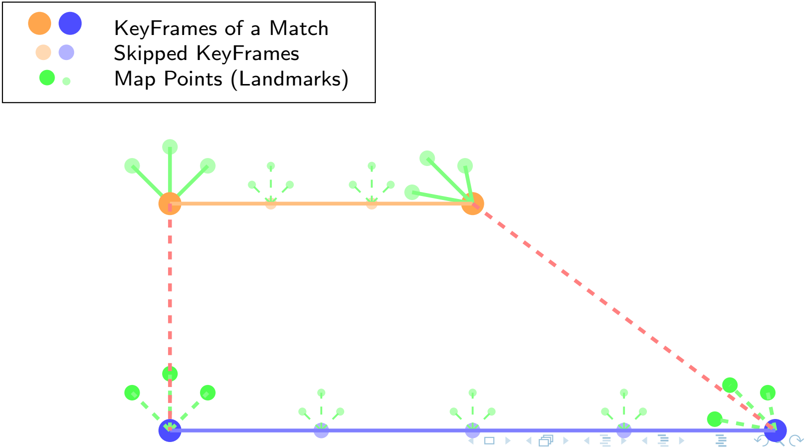
Find $n(=3)$ KeyFrame Matches (KFMs),
Skip $m(=5)$ KeyFrames (KFs)

KeyFrames of a Match
Skipped KeyFrames
Map Points (Landmarks)



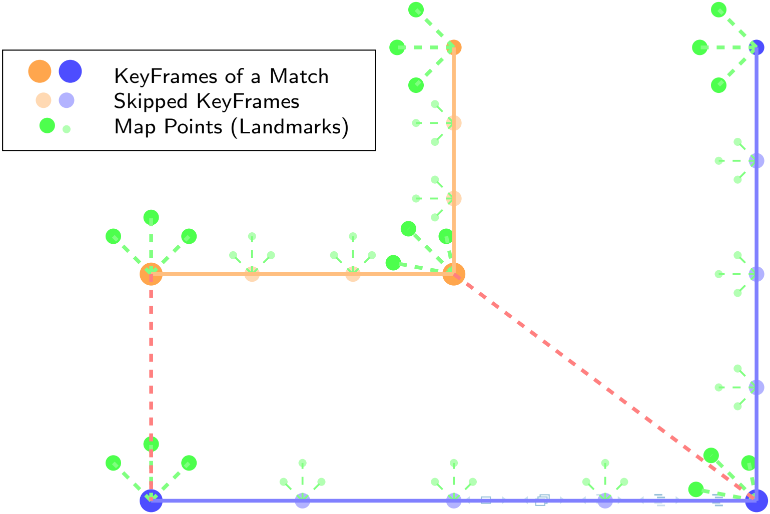
Map merging - New approach

Find $n(=3)$ KeyFrame Matches (KFMs),
Skip $m(=5)$ KeyFrames (KFs)



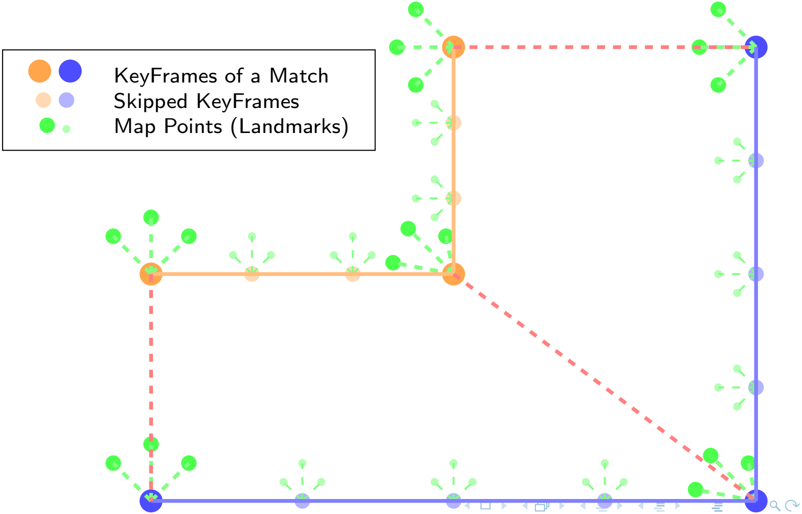
Map merging - New approach

Find $n(=3)$ KeyFrame Matches (KFMs), Skip $m(=5)$ KFs



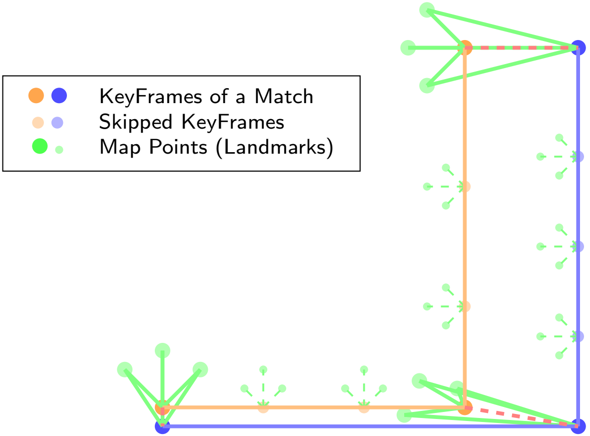
Map merging - New approach

Find $n(=3)$ KeyFrame Matches (KFMs)



Map merging - New approach

Merge the two maps and fuse map points



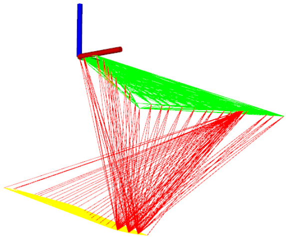
Map merging - Results - skipping of KeyFrame (KF)

Co-visibility graph

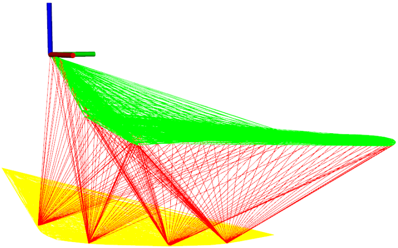
Connections/Edges between KeyFrames (KFs) which observe the same map points (landmarks)

Map merging - Results - skipping of KF

green: Covisibility graph of first map
yellow: Covisibility graph of second map
red: Covisibility between the KFMs

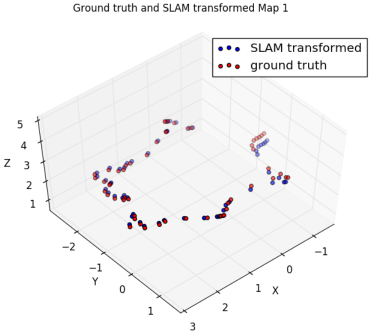


(a) 1 KF skipped after a KFM was found

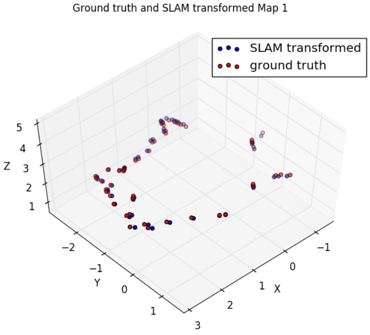


(b) 10 KF skipped after a KFM was found

Map merging - Results - Reduction of drift



(a) original approach



(b) new approach

Reduction of the error from rmse $= 0.13\text{m}$ to rmse $= 0.10\text{m}$

Map merging - Results

# KFMs	# KFs skip	rmse
1	0	0.1311
5	20	0.0912
10	5	0.1236
10	10	0.0961

Table: Error (rmse) of different settings (KFMs and KF skipps).

Culling - Remove redundant KF

Motivation

Perform KeyFrame (KF) culling to remove redundant information as bundle adjustment complexity grows with the number of KFs

Culling - Remove redundant KF

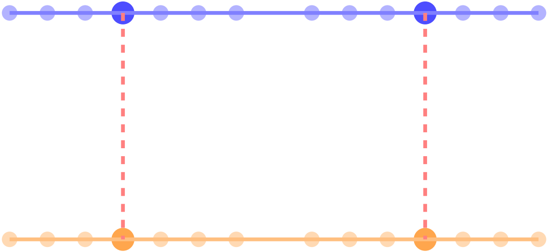
- Remove redundant KFs before map merging

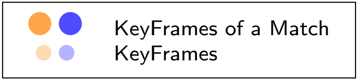
Culling - Remove redundant KF

- Remove redundant KFs before map merging
- Performs culling for every KFM separately

Culling - Remove redundant KF

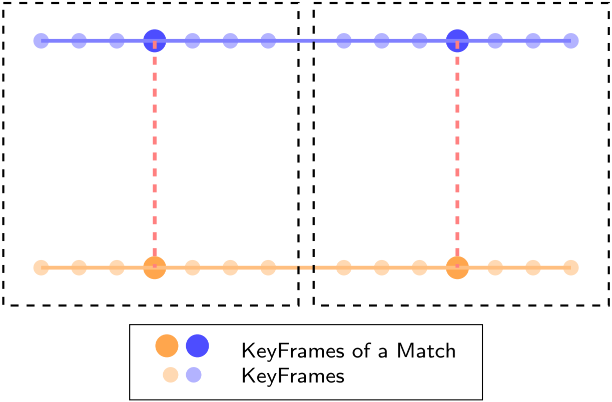
- Remove redundant KFs before map merging
- Performs culling for every KFM separately





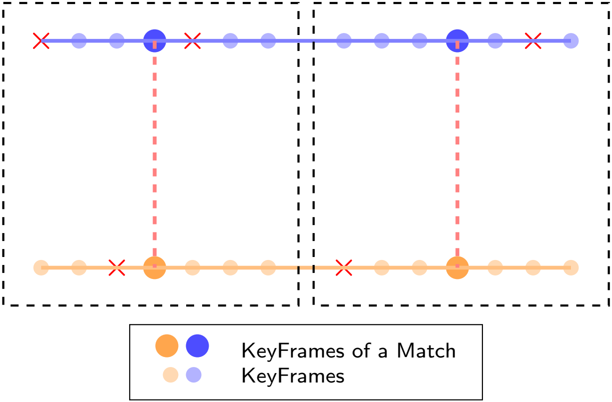
Culling - Remove redundant KF

- Remove redundant KFs before map merging
- Performs culling for every KFM separately



Culling - Remove redundant KF

- Remove redundant KFs before map merging
- Performs culling for every KFM separately



Culling - Results

Pose Graph Optimization (PGO)
Bundle Adjustment (BA)
v / e: numbers of verteces / edges of the graph to optimize

Culling	# KFMs	# KFs skip	PGO v / e	BA v / e
No	1	0	59 / 754	2308 / 22193
Yes	1	0	36 / 283	1893 / 13222
No	10	10	150 / 1949	4806 / 49765
Yes	10	10	93 / 657	4165 / 30453

Table: Time measurements of Pose Graph Optimization (PGO) and Bundle Adjustment (BA) without and with culling.

Culling - Results

Culling removes $\approx 13\%$ of the KeyFrames (KFs)

Pose Graph Optimization (PGO)
Bundle Adjustment (BA)

Culling - Results

Culling removes $\approx 13\%$ of the KeyFrames (KFs)

Culling	# KFMs	# KFs skipped	PGO [ms]	BA [ms]
No	10	10	532.28	3659.48
Yes	10	10	178.83	1098.37

Table: Time measurements of PGO and BA without and with culling.

Pose Graph Optimization (PGO)
Bundle Adjustment (BA)

Culling - Results

Culling removes $\approx 13\%$ of the KeyFrames (KFs)

Culling	# KFMs	# KFs skipped	PGO [ms]	BA [ms]
No	10	10	532.28	3659.48
Yes	10	10	178.83	1098.37

Table: Time measurements of PGO and BA without and with culling.

Performance increases significantly when culling is enabled

Pose Graph Optimization (PGO)
Bundle Adjustment (BA)

Culling - Results

Culling	# KFMs	# KFs skipped	*rmse* [m]
No	1	0	0.1311
Yes	1	0	0.2187
No	10	10	0.0961
Yes	10	10	0.0965

Table: *rmse* without and with culling.

Culling - Results

Culling	# KFMs	# KFs skipped	*rmse* [m]
No	1	0	0.1311
Yes	1	0	0.2187
No	10	10	0.0961
Yes	10	10	0.0965

Table: *rmse* without and with culling.

Accuracy gets worse if not enough information is available.
No problem with multiple KeyFrame Matches (KFMs).

Optimization - Idea

Considerable computational benefits can be gained by substituting the Levenberg-Marquardt (LM) algorithm in the implementation of Bundle Adjustment (BA) with a variant of Powell's dog leg (DL) non-linear least squares technique [Lourakis and Argyros, 2005]

Optimization - Idea

Considerable computational benefits can be gained by substituting the Levenberg-Marquardt (LM) algorithm in the implementation of Bundle Adjustment (BA) with a variant of Powell's dog leg (DL) non-linear least squares technique [Lourakis and Argyros, 2005]

DL optimizer handles trust region differently

Optimization - LM

The Levenberg-Marquardt (LM) solves iteratively

$$(\mathbf{J}^T\mathbf{J} + \lambda\mathbf{I})\delta = \mathbf{J}^T\boldsymbol{\epsilon}, \text{ where } \boldsymbol{\epsilon} = [\mathbf{y} - \mathbf{f}(\boldsymbol{\beta})]$$

Optimization - LM

The LM solves iteratively

$$(\mathbf{J}^T\mathbf{J} + \lambda\mathbf{I})\delta = \mathbf{J}^T\boldsymbol{\epsilon}, \text{ where } \boldsymbol{\epsilon} = [\mathbf{y} - \mathbf{f}(\boldsymbol{\beta})]$$

- With a small λ LM becomes a Gauss-Newton method

Optimization - LM

The LM solves iteratively

$$(\mathbf{J}^T\mathbf{J} + \lambda\mathbf{I})\delta = \mathbf{J}^T\boldsymbol{\epsilon}, \text{ where } \boldsymbol{\epsilon} = [\mathbf{y} - \mathbf{f}(\boldsymbol{\beta})]$$

- With a small λ LM becomes a Gauss-Newton method
- With a big λ LM behaves like a Gradient-descent method

Optimization - LM

The LM solves iteratively

$$(\mathbf{J}^T\mathbf{J} + \lambda\mathbf{I})\delta = \mathbf{J}^T\boldsymbol{\epsilon}, \text{ where } \boldsymbol{\epsilon} = [\mathbf{y} - \mathbf{f}(\boldsymbol{\beta})]$$

- With a small λ LM becomes a Gauss-Newton method
- With a big λ LM behaves like a Gradient-descent method
- If an update doesn't reduce the error, λ will be increased and the equation must be solved again

Optimization - DL

The Powell's dog leg (DL) solves iteratively

$$\min_{\delta} 2(\frac{1}{2}\boldsymbol{\epsilon}^T\boldsymbol{\epsilon} - (\mathbf{J}\boldsymbol{\epsilon})^T\delta + \frac{1}{2}\delta^T\mathbf{J}^T\mathbf{J}\delta), \text{ subjected to } ||\delta|| \leq \Delta$$

For $\kappa \in [0, 2]$, the dog leg trajectory is defined as

$$\delta(\kappa) = \begin{cases} \kappa\delta_{gd} & 0 \leq \kappa \leq 1 \\ \delta_{gd} + (\kappa - 1)(\delta_{gn} - \delta_{gd}) & 1 \leq \kappa \leq 2 \end{cases}$$

Optimization - DL

With

$$\delta_{gd} = \frac{\mathbf{g}^T\mathbf{g}}{\mathbf{g}^T\mathbf{J}^T\mathbf{J}\mathbf{g}}\mathbf{g}$$

and δ_{gn} the solution of

$$\mathbf{J}^T\mathbf{J}\delta_{gn} = \mathbf{g}$$

the dog leg trajectory looks like

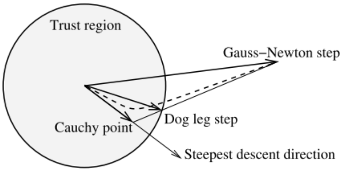


Figure from [Lourakis and Argyros, 2005]

Optimization - DL

- Once the Gauss-Newton step has been determined, the DL algorithm can solve the subproblem for various Δ without resolving an equation

Optimization - DL

- Once the Gauss-Newton step has been determined, the DL algorithm can solve the subproblem for various Δ without resolving an equation
- Reducing the number of times the Gauss-Newton step has to be determined is crucial for the overall performance of the minimization process

Optimization - DL

- Once the Gauss-Newton step has been determined, the DL algorithm can solve the subproblem for various Δ without resolving an equation
- Reducing the number of times the Gauss-Newton step has to be determined is crucial for the overall performance of the minimization process
- For the mentioned reasons the DL algorithm requires less computational effort compared to the LM algorithm

Optimization - Approach

- Tried Pose Graph Optimization (PGO) and Bundle Adjustment (BA) with the Powell's dog leg (DL) optimizer.

Optimization - Approach

- Tried Pose Graph Optimization (PGO) and Bundle Adjustment (BA) with the Powell's dog leg (DL) optimizer.
- PGO: Slightly worse timing using the DL optimizer

Optimization - Approach

- Tried Pose Graph Optimization (PGO) and Bundle Adjustment (BA) with the Powell's dog leg (DL) optimizer.
- PGO: Slightly worse timing using the DL optimizer
- BA: Better timing using the DL optimizer

Optimization - Approach

- Tried Pose Graph Optimization (PGO) and Bundle Adjustment (BA) with the Powell's dog leg (DL) optimizer.
- PGO: Slightly worse timing using the DL optimizer
- BA: Better timing using the DL optimizer

Conclusion

LM optimizer for PGO and DL optimizer for BA

Optimization - Results

Opt.	# KFMs	# KFs skipped	PGO [ms]	BA [ms]
LM/LM	10	10	178.83	1098.37
LM/DL	10	10	178.70	383.54

Table: Time measurements of LM and DL optimizer.

Pose Graph Optimization (PGO)
Bundle Adjustment (BA)

Optimization - Results

Opt.	# KFMs	# KFs skipped	PGO [ms]	BA [ms]
LM/LM	10	10	178.83	1098.37
LM/DL	10	10	178.70	383.54

Table: Time measurements of LM and DL optimizer.

Accuracy stays the same while the performance is increased

Pose Graph Optimization (PGO)
Bundle Adjustment (BA)

Conclusion

- Multiple KFMs approach increases accuracy

Conclusion

- Multiple KFMs approach increases accuracy
- Skipping of KFs spreads KFMs over a bigger area

Conclusion

Higher accuracy

The use of KFMs from a bigger area serves PGO and BA with more information $\rightarrow$ higher accuracy

Conclusion

Higher accuracy

The use of KFMs from a bigger area serves PGO and BA with more information $\rightarrow$ higher accuracy

- Culling removes redundant KFs $\rightarrow$ improved timing

Conclusion

Higher accuracy

The use of KFMs from a bigger area serves PGO and BA with more information → higher accuracy

- Culling removes redundant KFs → improved timing
- Using DL optimizer for the BA also improves timing

Conclusion

Higher accuracy

The use of KFMs from a bigger area serves PGO and BA with more information $\rightarrow$ higher accuracy

Better timing

Culling and the use of the DL optimizer improves timing

Outlook & Limitation

Outlook:

- Heuristic for best map alignment
- Extend area for KF culling

Limitation:

- # of KFMs and # of skips depends on data set

Outlook & Limitation

Outlook:

- Heuristic for best map alignment
- Extend area for KF culling

Limitation:

- # of KFMs and # of skips depends on data set

Outlook & Limitation

Outlook:

- Heuristic for best map alignment
- Extend area for KF culling

Limitation:

- # of KFMs and # of skips depends on data set

Outlook & Limitation

Outlook:

- Heuristic for best map alignment
- Extend area for KF culling

Limitation:

- # of KFMs and # of skips depends on data set

Outlook & Limitation

Outlook:

- Heuristic for best map alignment
- Extend area for KF culling

Limitation:

- # of KFMs and # of skips depends on data set

Outlook & Limitation

Outlook:

- Heuristic for best map alignment
- Extend area for KF culling

Limitation:

- # of KFMs and # of skips depends on data set

Q&A

References

Lourakis, M. I. A. and Argyros, A. A. (2005).
Is Levenberg-Marquardt the most efficient optimization algorithm for implementing bundle adjustment?
In *Proceedings of the IEEE International Conference on Computer Vision*, volume II, pages 1526–1531.

Mur-Artal, R. and Tardos, J. D. (2016).
ORB-SLAM2: an Open-Source SLAM System for Monocular, Stereo and RGB-D Cameras.