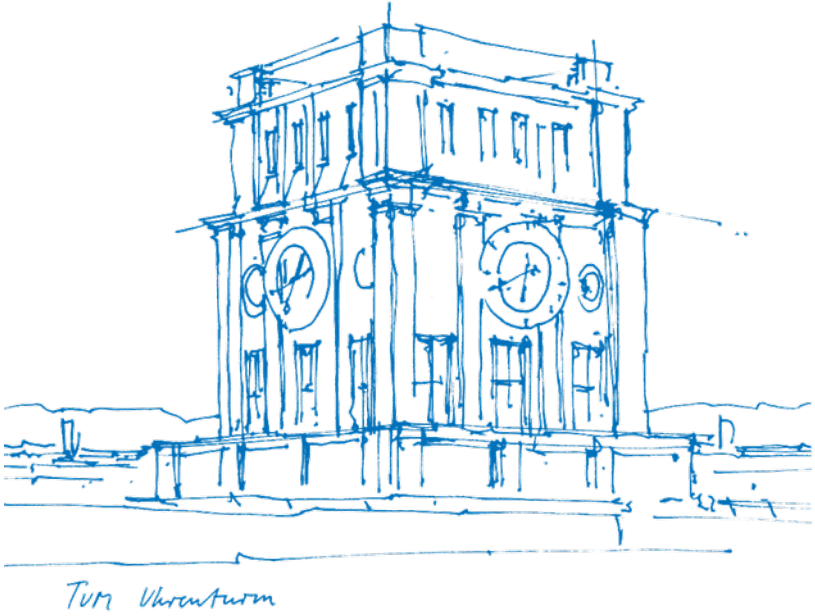
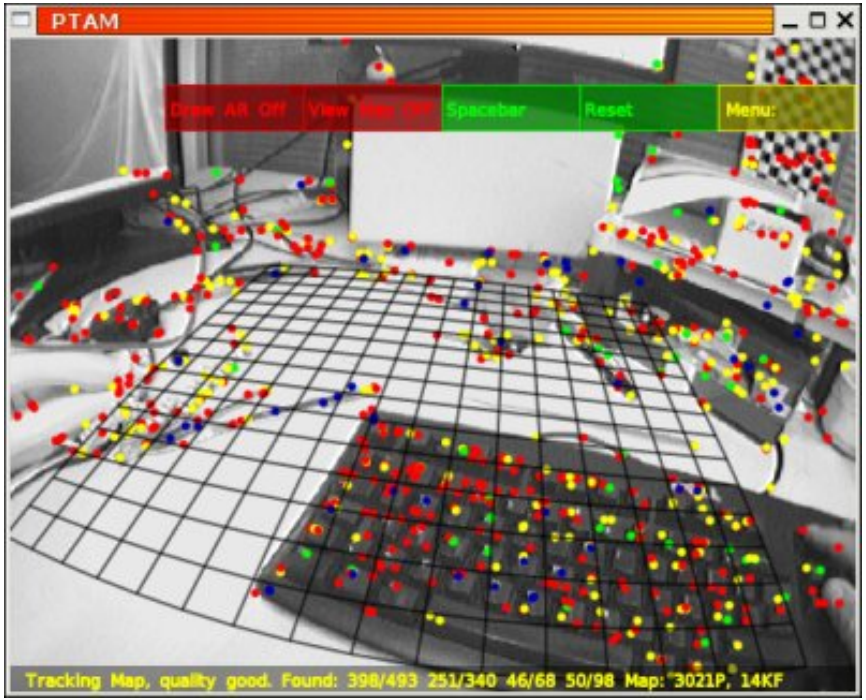
Parallel Tracking and Mapping

Martin Wudenka
Technical University of Munich
Department of Informatics
Computer Vision Group
Munich, 15-16/04/2020



Parallel Tracking and Mapping (PTAM) in a Nutshell



PTAM is a SLAM framework designed for small workplaces with augmented reality applications in mind.

- Georg Klein and David Murray
- Active Vision Laboratory, University of Oxford
- ISMAR 2007 conference paper
- keyframe based
- pointcloud map
- no prior scene knowledge required

Agenda

1 Primer

2 Genealogy of Ideas

3 Methodology

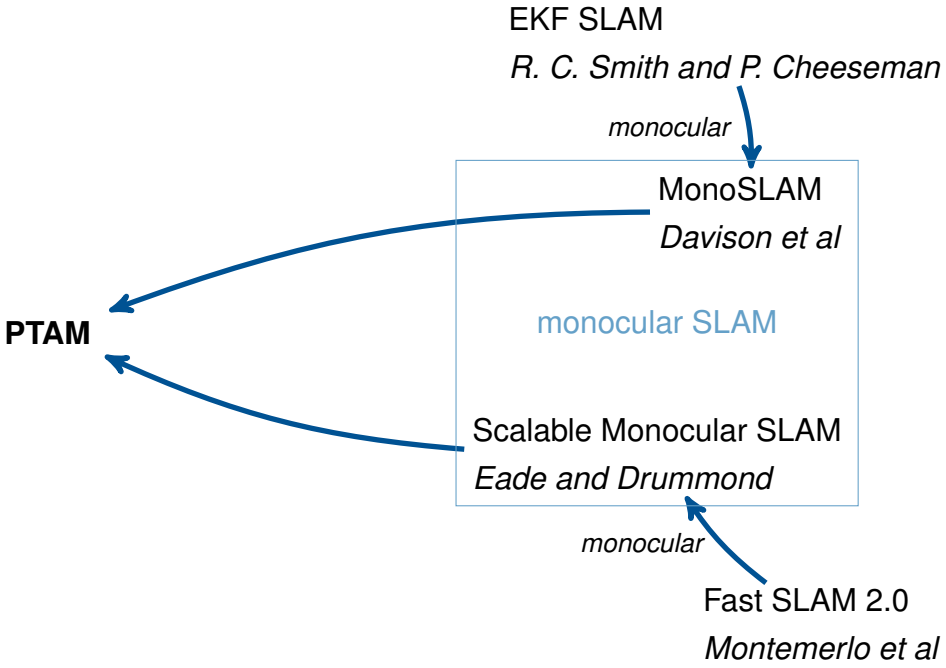
4 Results

5 Summary

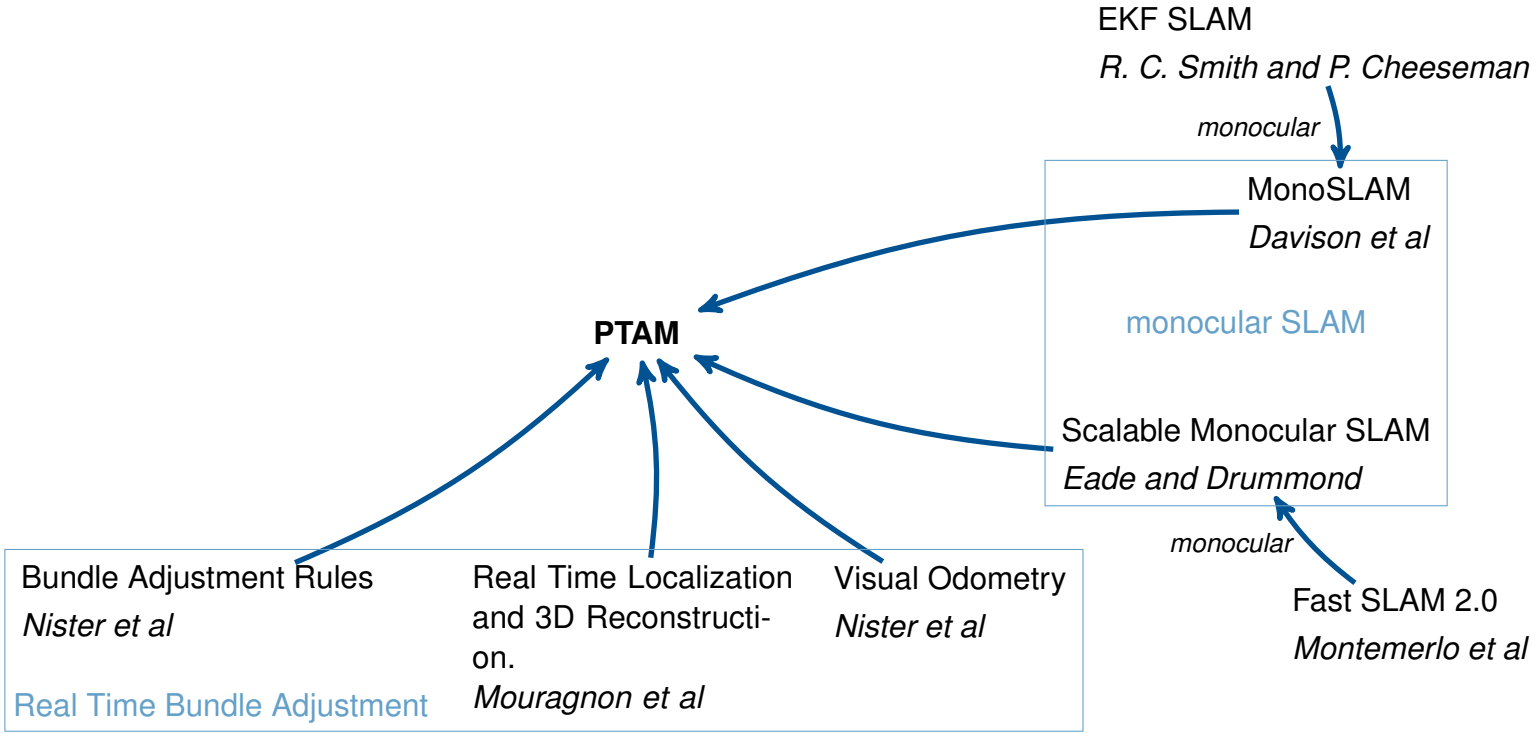
6 Discussion

Genealogy of Ideas

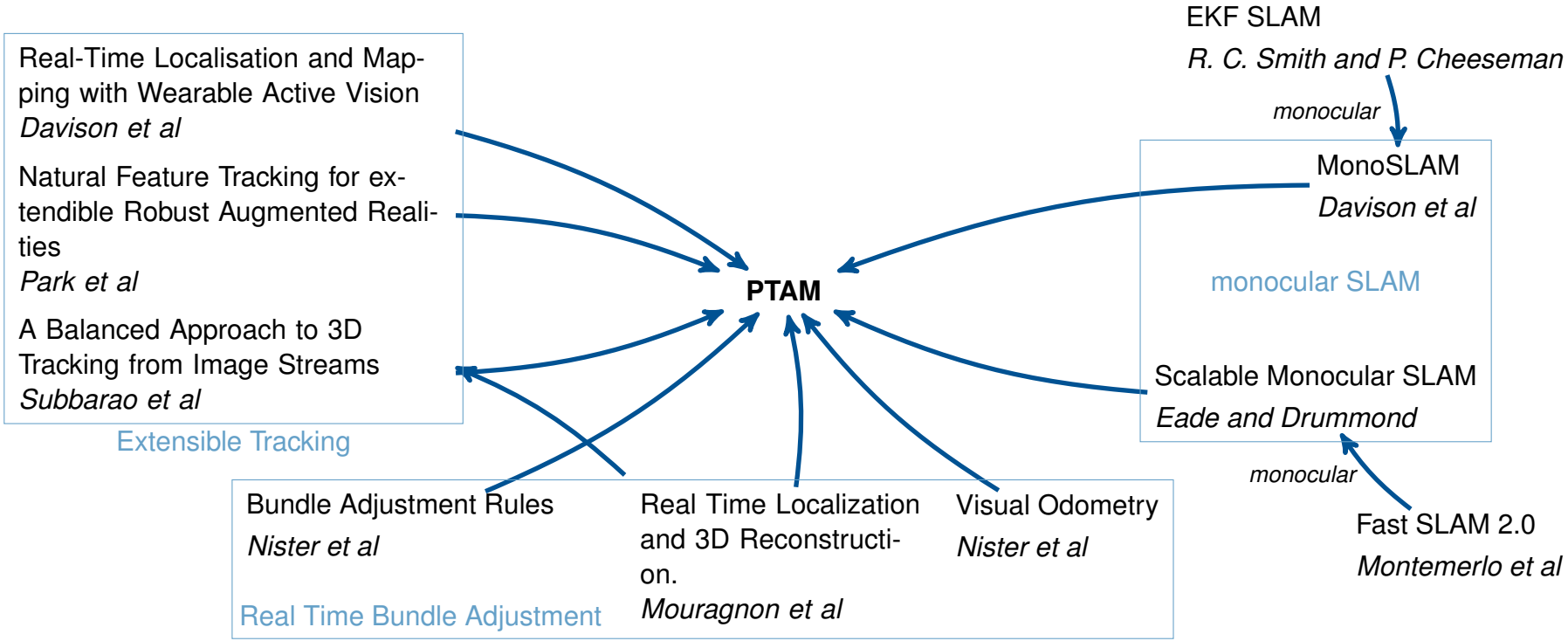
Genealogy of Ideas



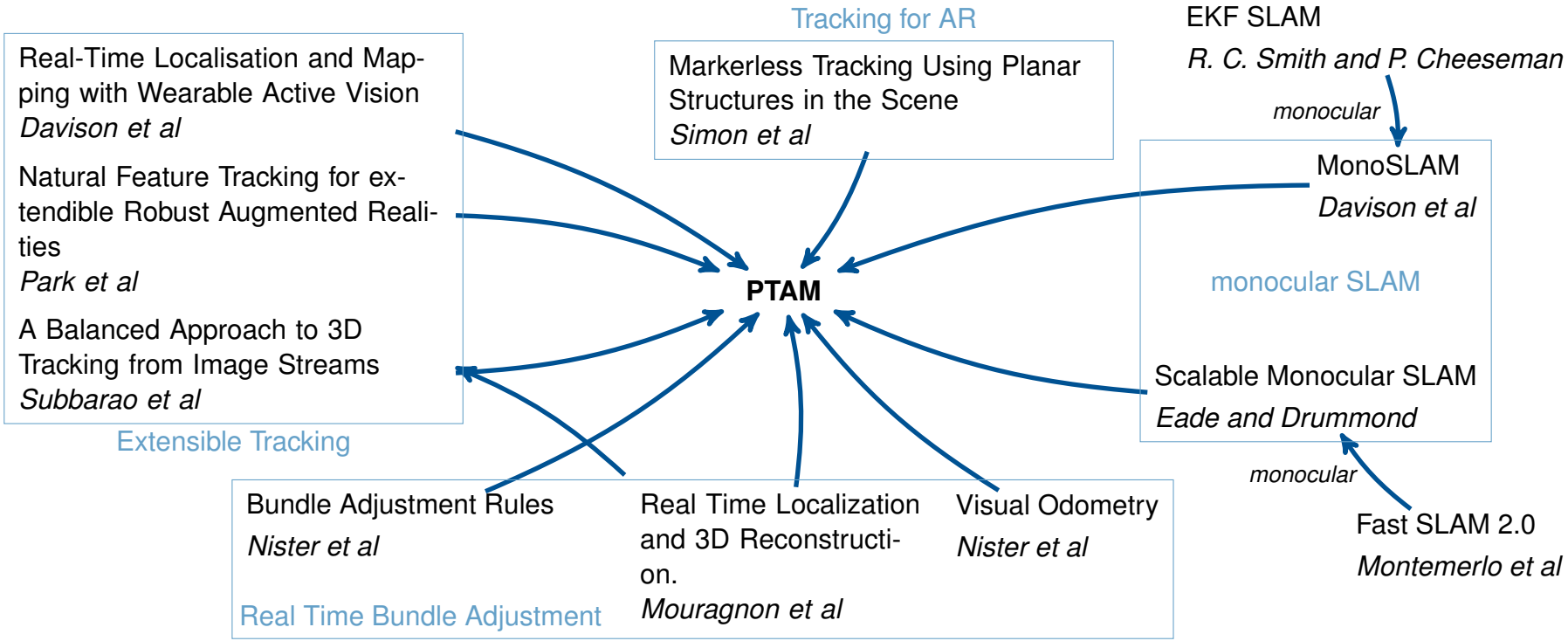
Genealogy of Ideas



Genealogy of Ideas

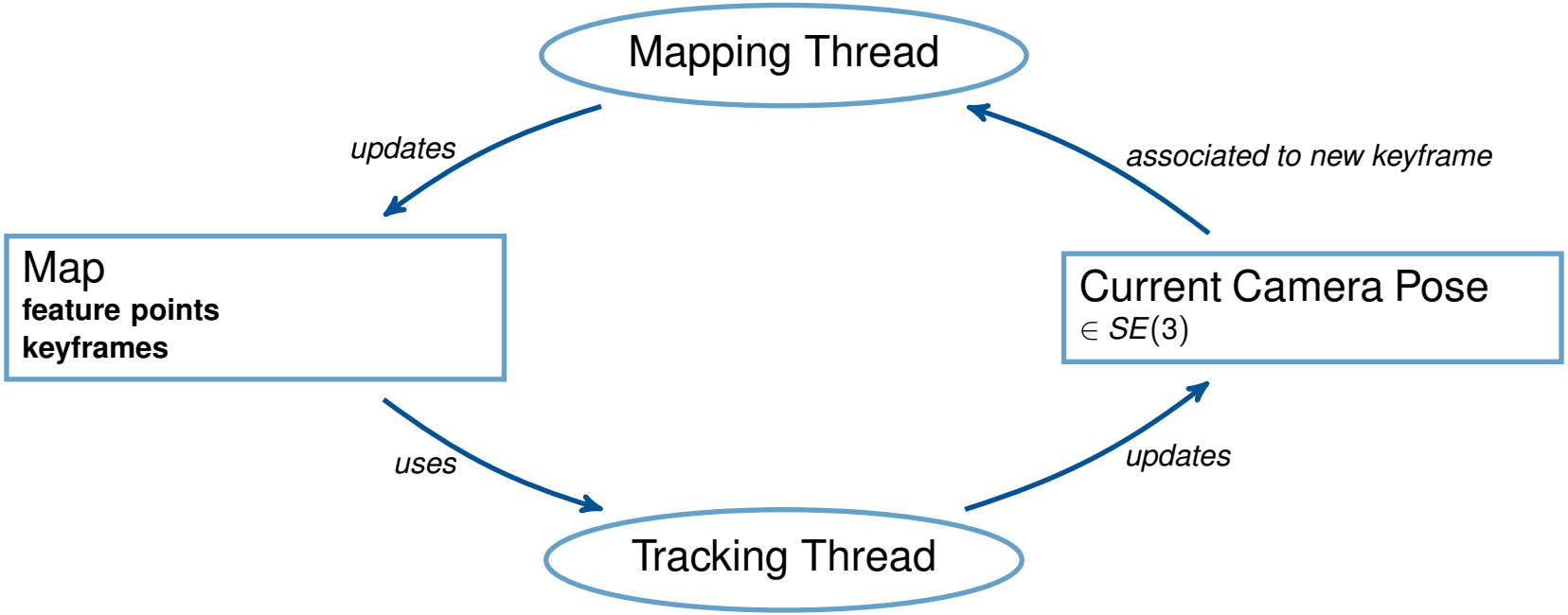


Genealogy of Ideas

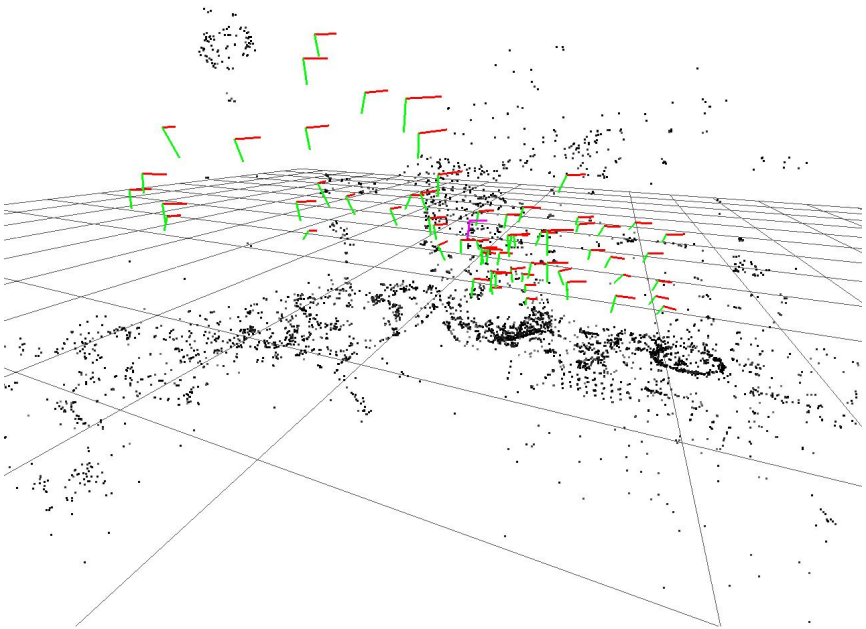


Methodology

System Architecture



Map



Feature Points

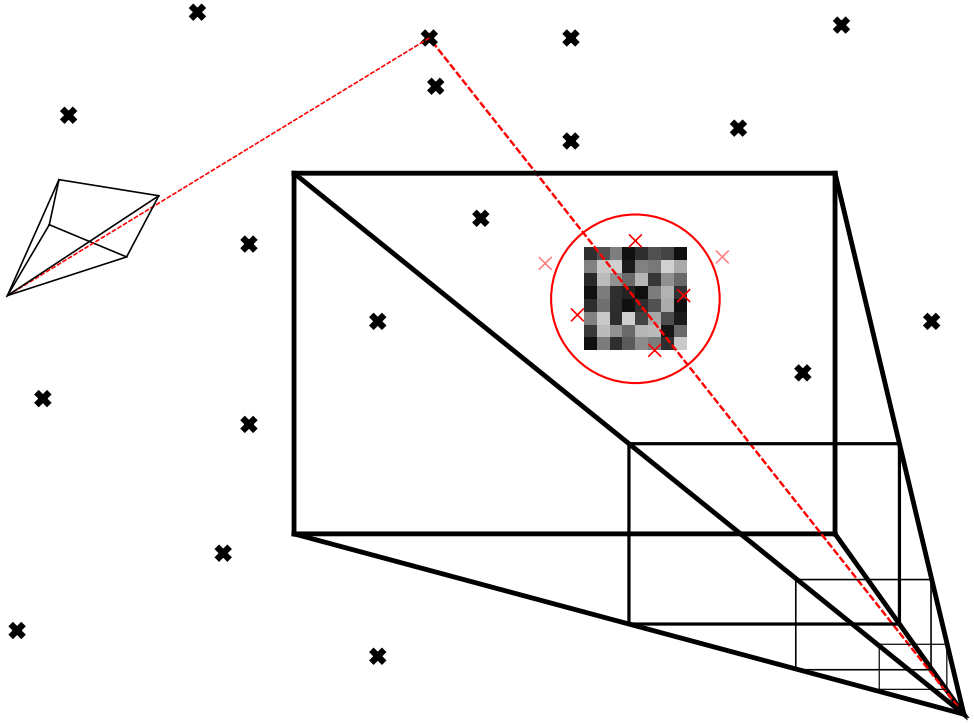
- 2000 to 6000
- 3D coordinates
- normal unit vector
- reference to keyframe pyramid level

Keyframes

- 40 to 120
- greyscale
- 4 level pyramid (640x480 to 80x60)
- camera coordinate system transformation

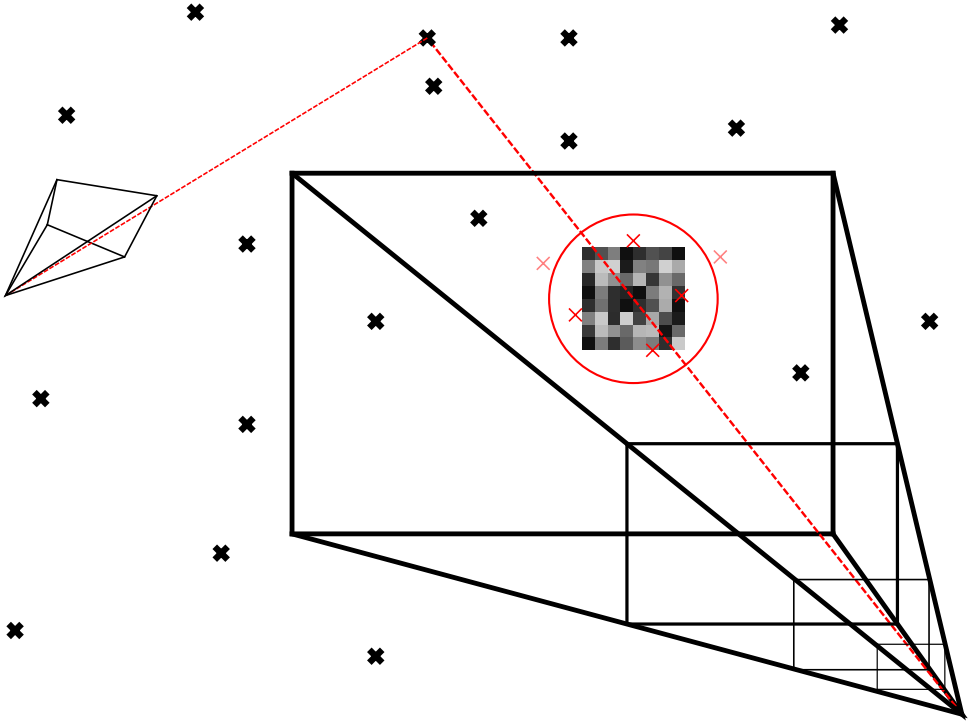
Tracking

1. Prior pose estimate
2. Image preparation
3. Projection of map points
4. Search of map points
5. Coarse and fine pose update



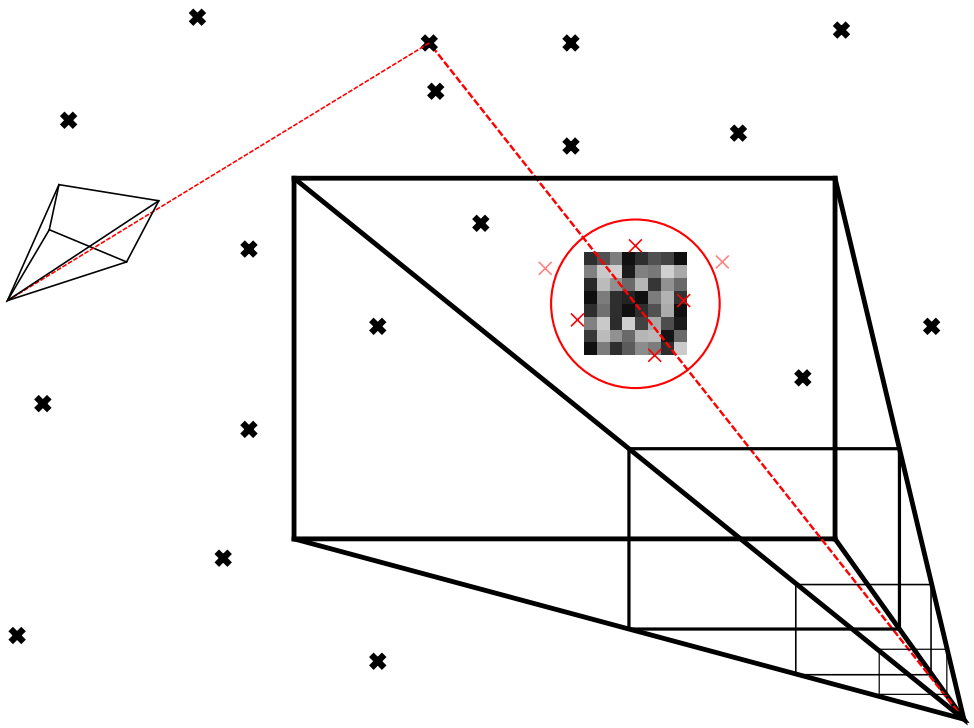
Tracking

1. Prior pose estimate
   - **Decaing velocity model**
2. Image preparation
3. Projection of map points
4. Search of map points
5. Coarse and fine pose update



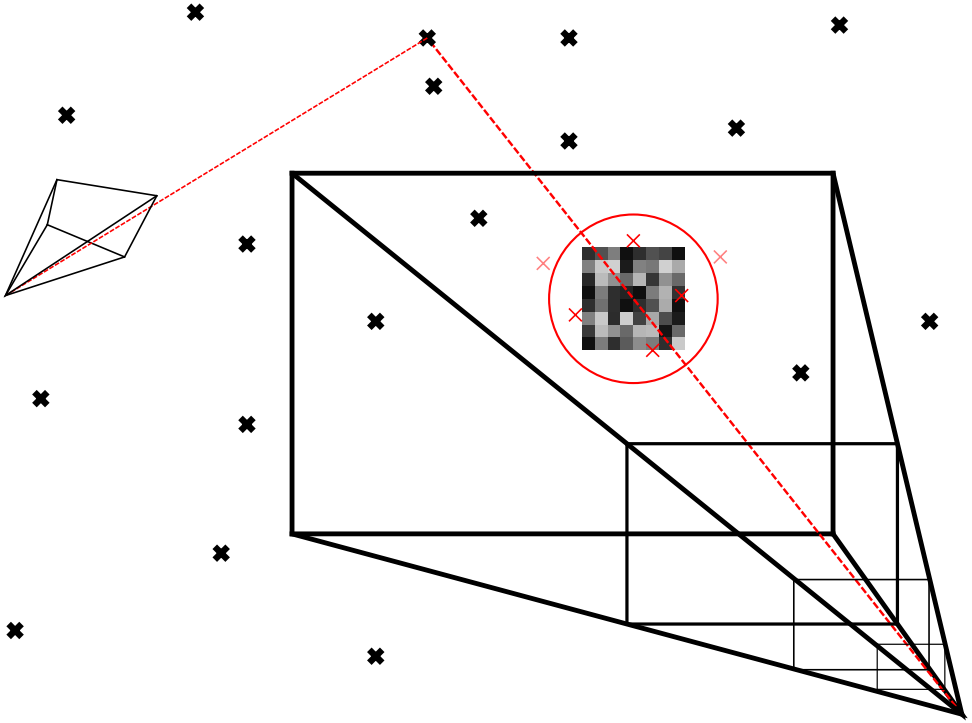
Tracking

1. Prior pose estimate
2. Image preparation
   - **Image pyramid**
   - **FAST-10 features**
3. Projection of map points
4. Search of map points
5. Coarse and fine pose update



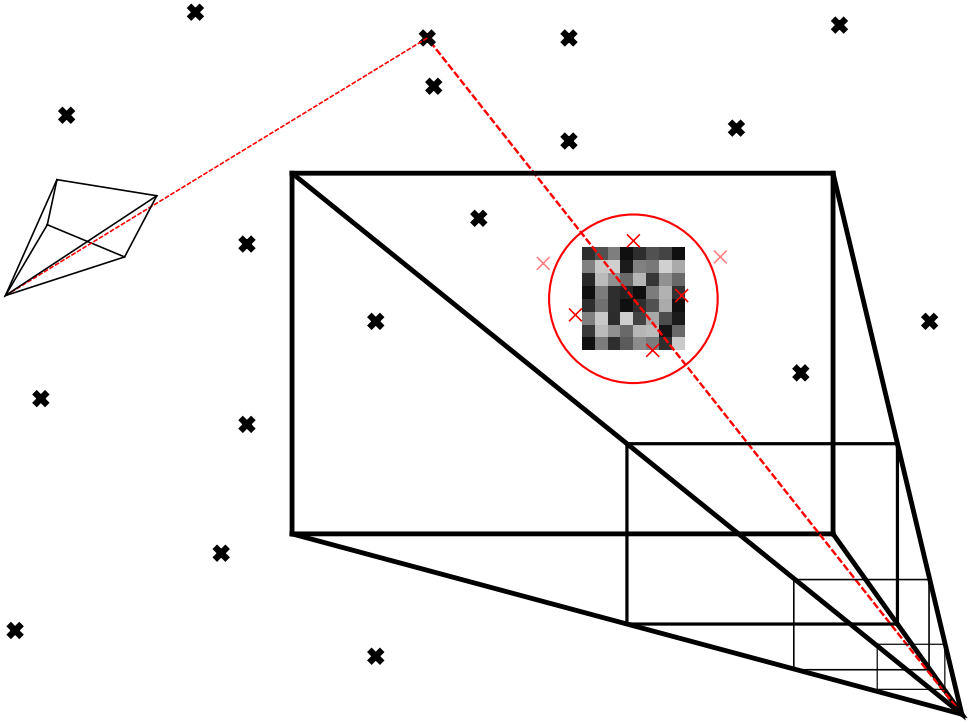
Tracking

1. Prior pose estimate
2. Image preparation
3. Projection of map points
   - **Pin-hole camera**
   - **Arctan calibration model**
   - **Warping**
   - **Bilinear interpolation**
   - **Patches of 8x8 pixels**
4. Search of map points
5. Coarse and fine pose update



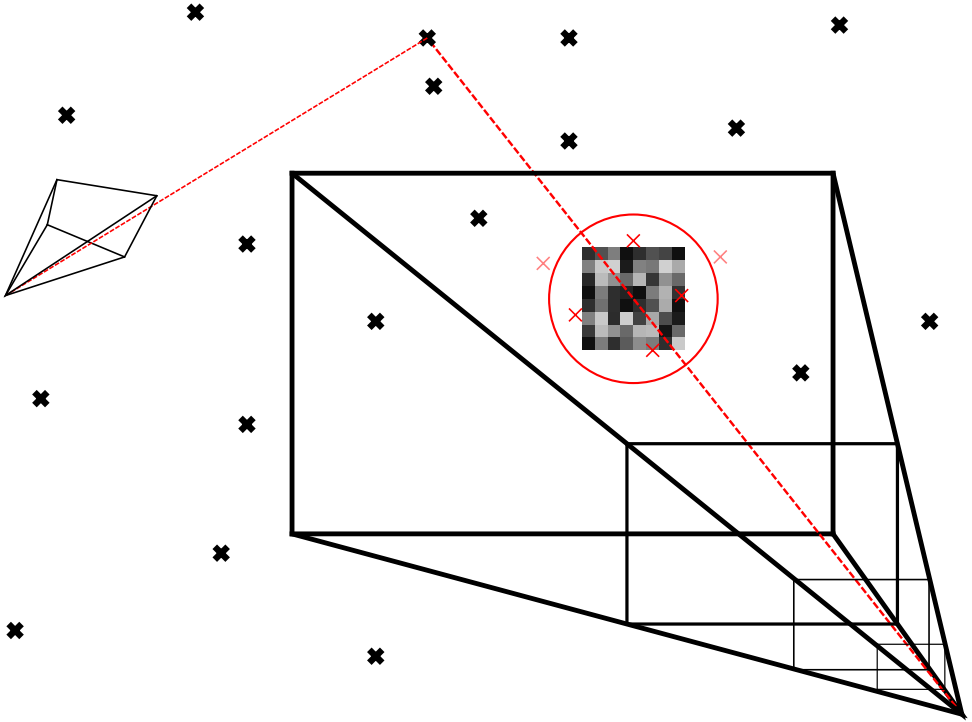
Tracking

1. Prior pose estimate
2. Image preparation
3. Projection of map points
4. Search of map points
   - **Pyramid level desided by warping matrix**
   - **Over FAST features around projection**
   - **Invariant to lighting changes**
   - **SSD**
   - **Sometimes sub-pixel accurcy**
5. Coarse and fine pose update



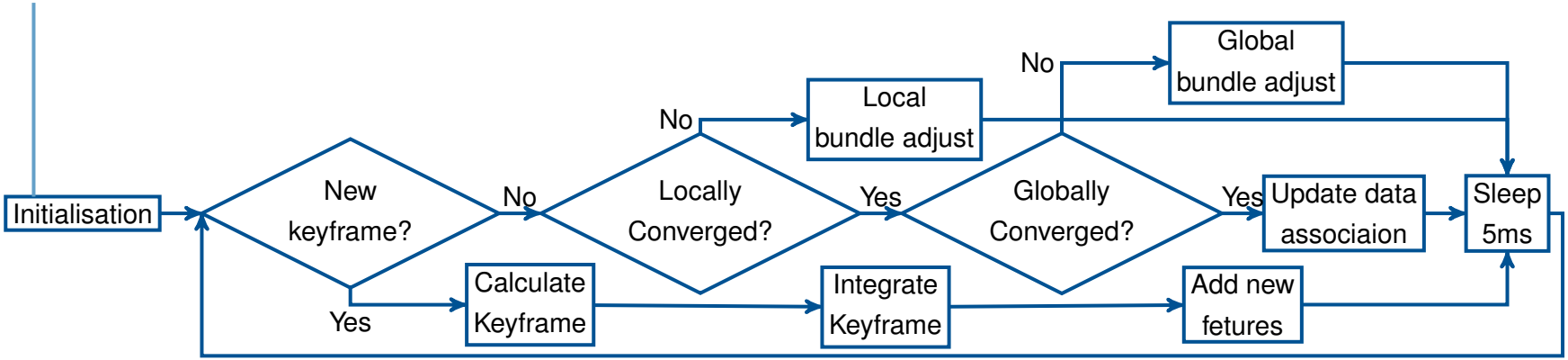
Tracking

1. Prior pose estimate
2. Image preparation
3. Projection of map points
4. Search of map points
5. Coarse and fine pose update
   - **50 coarse points**
   - **1000 fine points**
   - **Bundle adjustment**
   - **Tukey biweight objective function**
   - **10 iterations of Levenberg-Marquardt**



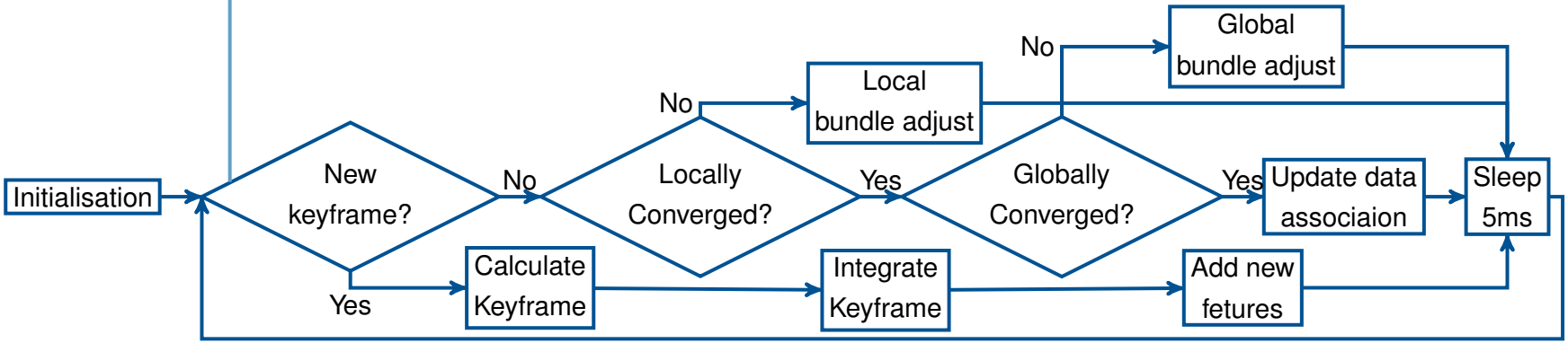
Mapping

- user cooperative
- five-point stereo or Faugeras-Lustman
- RANSAC
- 2 keyframes
- 1000 features
- distance between 2KF is "10cm"



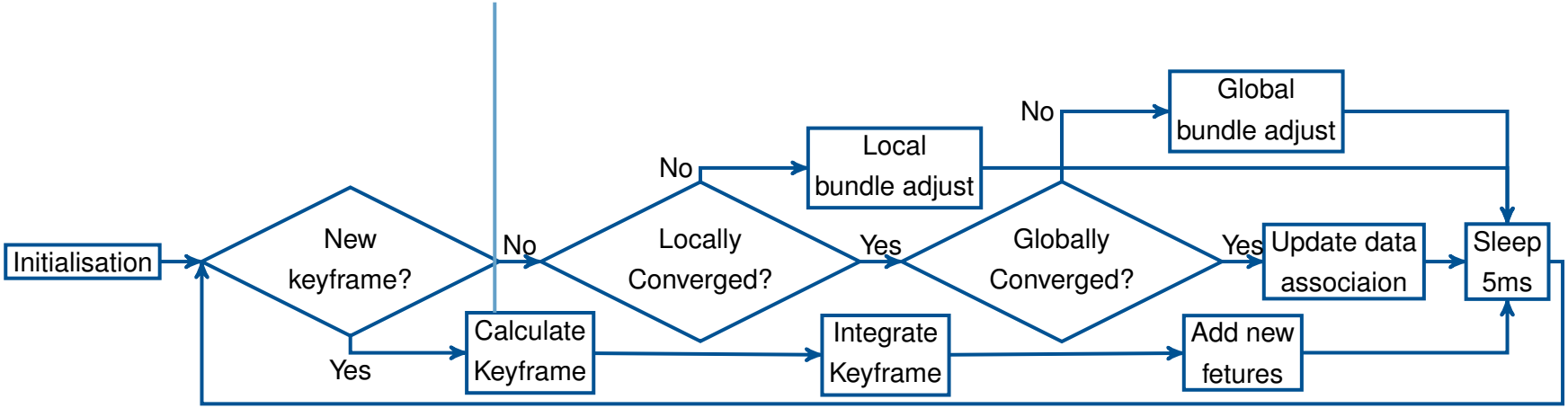
Mapping

- good tracking
- frames since last KF $\geq$ 20
- distance away to closest keyframe



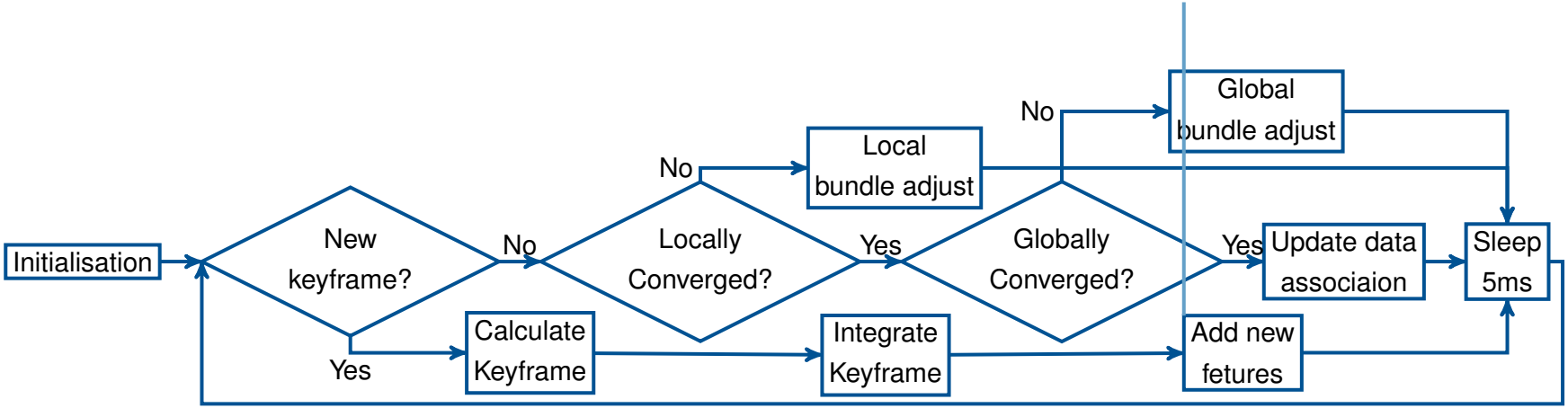
Mapping

- improve pose with more features
- non-maxima suppr. of features



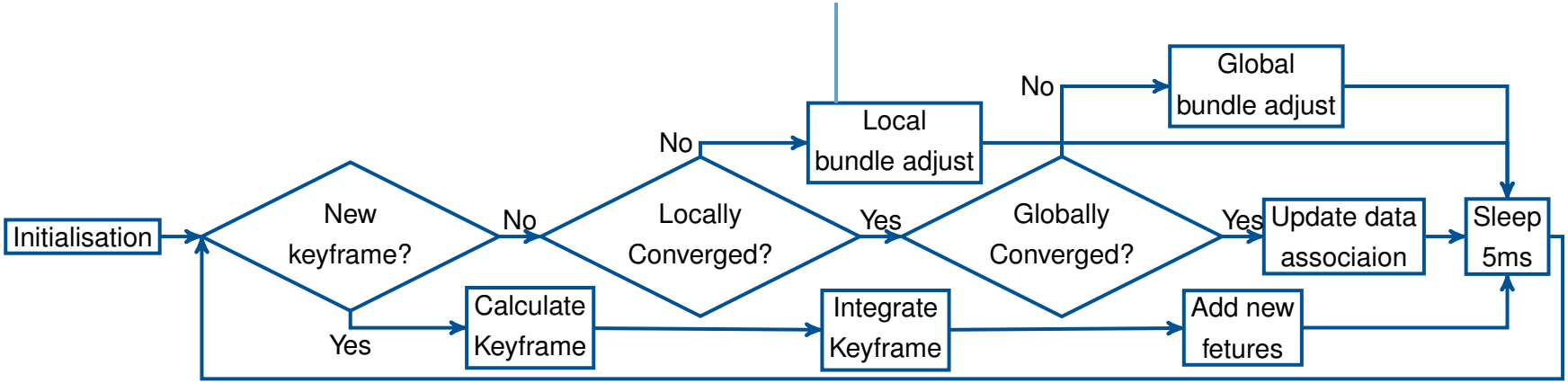
Mapping

- epipolar search with closest KF
- no warping
- only same pyramid levels
- zero mean SSD



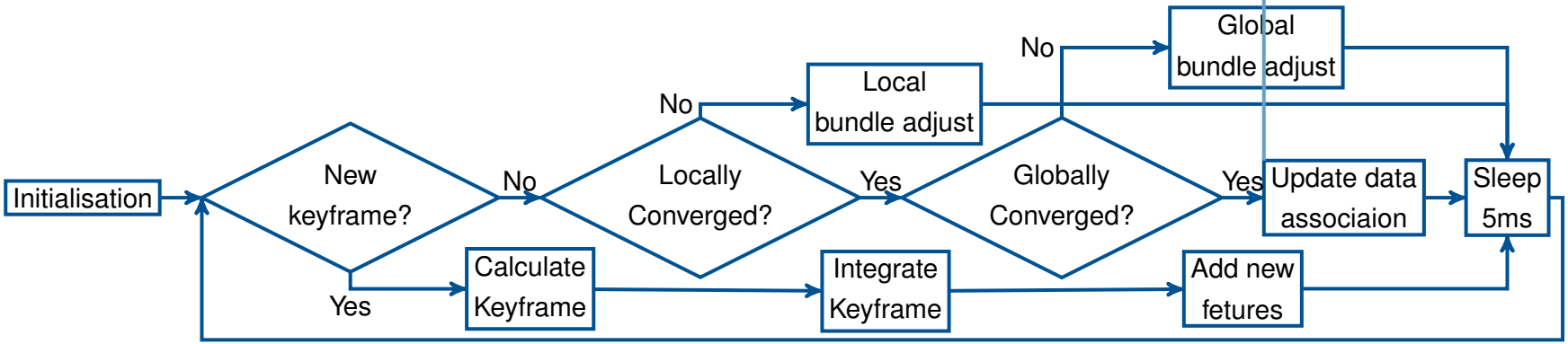
Mapping

- Levenberg-Marquardt
- Tukey biweight objective function
- adjusts map point positions and KF poses
- 5 KF to adjust
- + all points visible
- + all KF the points are visible in



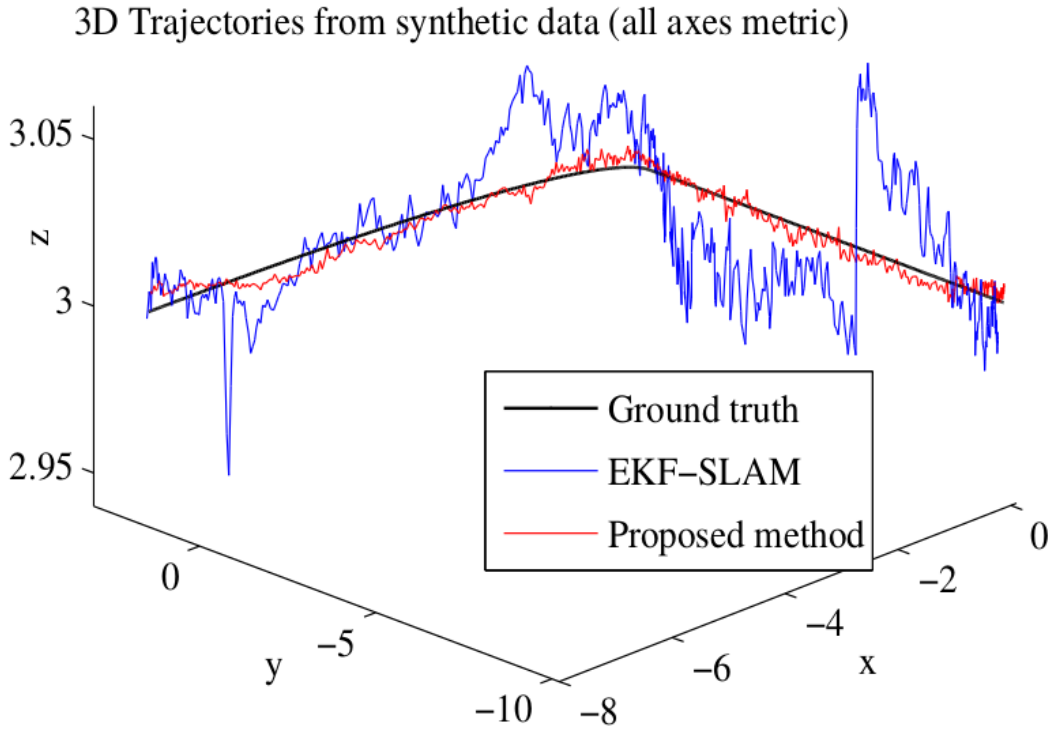
Mapping

- new features in old KF
- re-measure outliers



Results

Comparison with EKF-SLAM



Application: TUM AR Drone

Follow-Up Papers

Improving the Agility of Keyframe-Based SLAM

Georg Klein and David Murray

Active Vision Laboratory, University of Oxford, UK
{gk,dwm}@robots.ox.ac.uk

Abstract. The ability to localise a camera moving in a previously unknown environment is desirable for a wide range of applications. In computer vision this problem is studied as monocular SLAM. Recent years have seen improvements to the usability and scalability of monocular SLAM systems to the point that they may soon find uses outside of laboratory conditions. However, the robustness of these systems to rapid camera motions (we refer to this quality as agility) still lags behind that of tracking systems which use known object models. In this paper we attempt to remedy this. We present two approaches to improving the agility of a keyframe-based SLAM system: Firstly, we add edge features to the map and exploit their resilience to motion blur to improve tracking under fast motion. Secondly, we implement a very simple inter-frame rotation estimator to aid tracking when the camera is rapidly panning – and demonstrate that this method also en-

Parallel Tracking and Mapping on a Camera Phone

Georg Klein* David Murray†

Active Vision Laboratory
Department of Engineering Science
University of Oxford

Abstract

Camera phones are a promising platform for hand-held augmented reality. As their computational resources grow, they are becoming increasingly suitable for visual tracking tasks. At the same time, they still offer considerable challenges: Their cameras offer a narrow field-of-view not best suitable for robust tracking; images are often received at less than 15Hz; long exposure times result in significant motion blur; and finally, a rolling shutter causes severe smearing effects. This paper describes an attempt to implement a keyframe-based SLAM system on a camera phone (specifically, the Apple iPhone 3G). We describe a series of adaptations to the Parallel Tracking and Mapping system to mitigate the impact of the device's imaging deficiencies. Early results demonstrate a system capable of generating and augmenting small maps, albeit with reduced accuracy and robustness compared to SLAM on a PC.

1 Introduction

Modern mobile phones have become a compelling platform for mobile Augmented Reality. They contain all the equipment typically required for video-see-though AR: A computer with graphics acceleration, a camera, and a display. Even though the computational power available on phones still lags far behind desktop PCs, some researchers have already demonstrated that it is sufficient to perform real-time visual tracking [8, 10, 14, 15]. In particular, we are encouraged by [15]: Wagner *et al.* demonstrate that by carefully

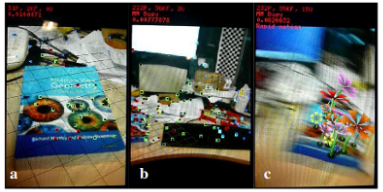


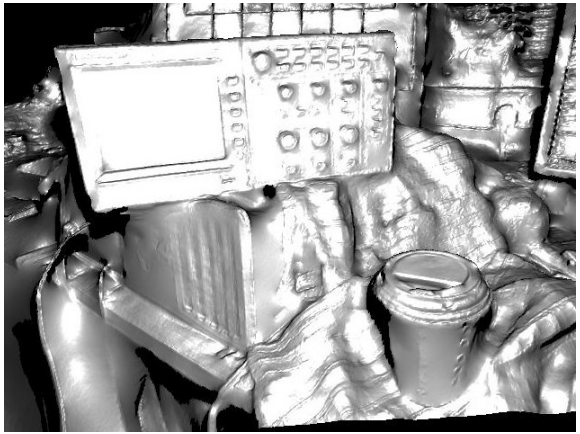
Figure 1: PTAM running on an iPhone 3G. A map is started on a book (a), expanded to circa 220 points (b), and then used to insert virtual flowers into the scene (c).

camera, which differs in three ways from the cameras commonly used for SLAM. It operates at a low (<15Hz) frame-rate; it has a rolling shutter; and most importantly, it has a narrow field-of-view. Each of these effects individually would be problematic, but when all three are present they reinforce one another and conspire to produce video images poorly suited to 3D reconstruction.

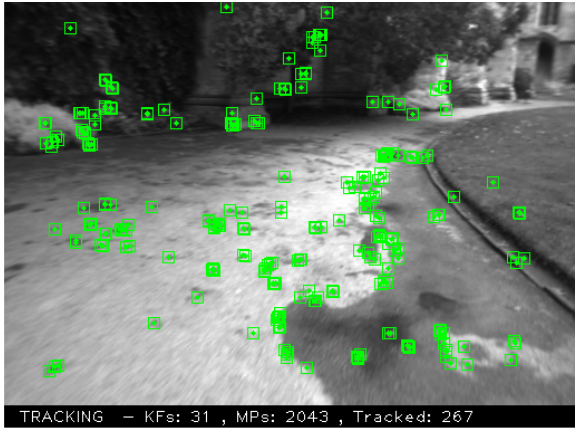
After a brief review of related mobile tracking work in Section 2, this paper describes a series of changes to the PTAM algorithm which mitigate the impact of the above imaging and processing lim-

Following Research

DTAM *2011*



ORB-SLAM *2015*



S-PTAM *2017*

Summary

Significance

[PTAM] was the first work to introduce the idea of splitting camera tracking and mapping in parallel threads, and demonstrated to be successful for real-time augmented reality applications in small environments.

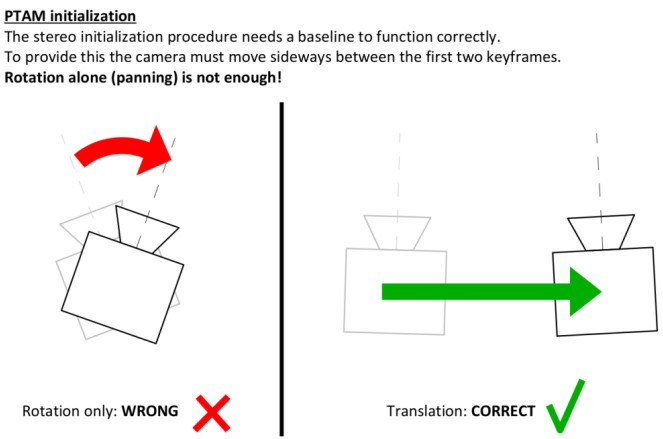
ORB-SLAM authors

Pros and Cons

+ parallelization
+ tracking quality
+ dynamic activities of mapping thread
+ can recover lost tracking
+ constant tracking rate

Pros and Cons

- \+ parallelization
- \+ tracking quality
- \+ dynamic activities of mapping thread
- \+ can recover lost tracking
- \+ constant tracking rate



- \- initialization
- \- assumptions: static, textured, not self-occluding
- \- no absolute scale
- \- prone to blur
- \- normal vectors of features not optimized
- \- no loop closure
- \- wrong keyframes corrupt map
- \- no map pruning
- \- no access to feature covariance

Discussion

Summary

[PTAM] was the first work to introduce the idea of splitting camera tracking and mapping in parallel threads, and demonstrated to be successful for realtime augmented reality applications in small environments.

ORB-SLAM authors

+ parallelization
+ tracking quality
+ dynamic activities of mapping thread
+ can recover lost tracking
+ constant tracking rate

- assumptions: static, textured, not self-occluding
- initialization
- no absolute scale
- prone to blur
- wrong keyframes corrupt map
- normal vectors of features not optimized
- no loop closure
- no map pruning
- no access to feature covariance