# Parallel Tracking and Mapping for Small AR Workspaces

Georg Klein and David Murray
Active Vision Lab, Oxford

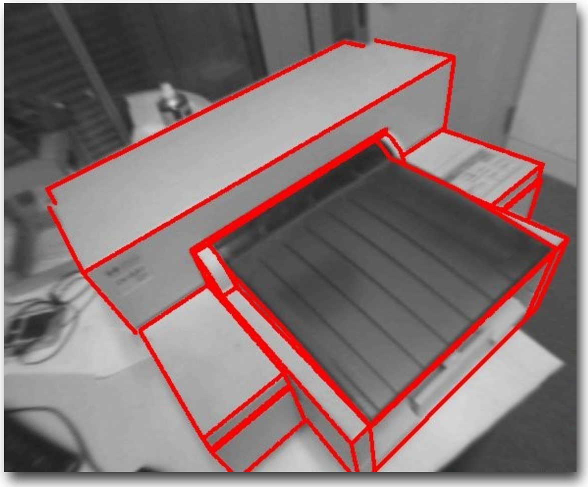
This is a PDF of the slides of the talk given at ISMAR 2007

# Aim

- AR with a hand-held camera
- Visual Tracking provides registration

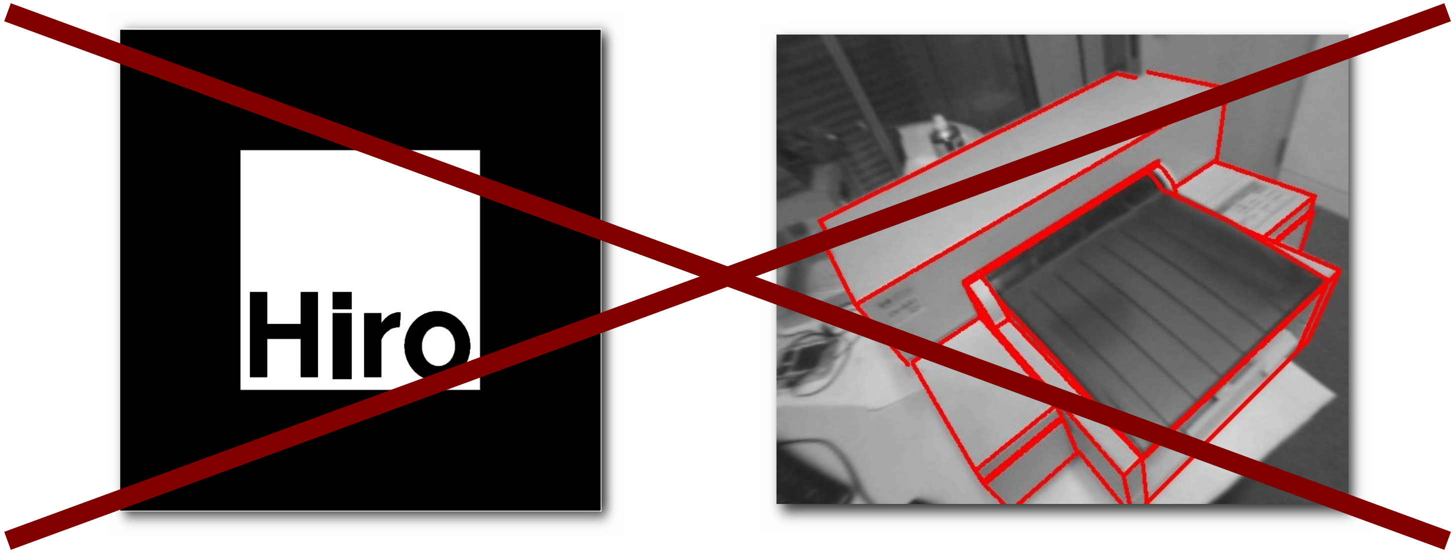
# Aim

- AR with a hand-held camera
- Visual Tracking provides registration



# Aim

- AR with a hand-held camera
- Visual Tracking provides registration
- Track *without* prior model of world



# Aim

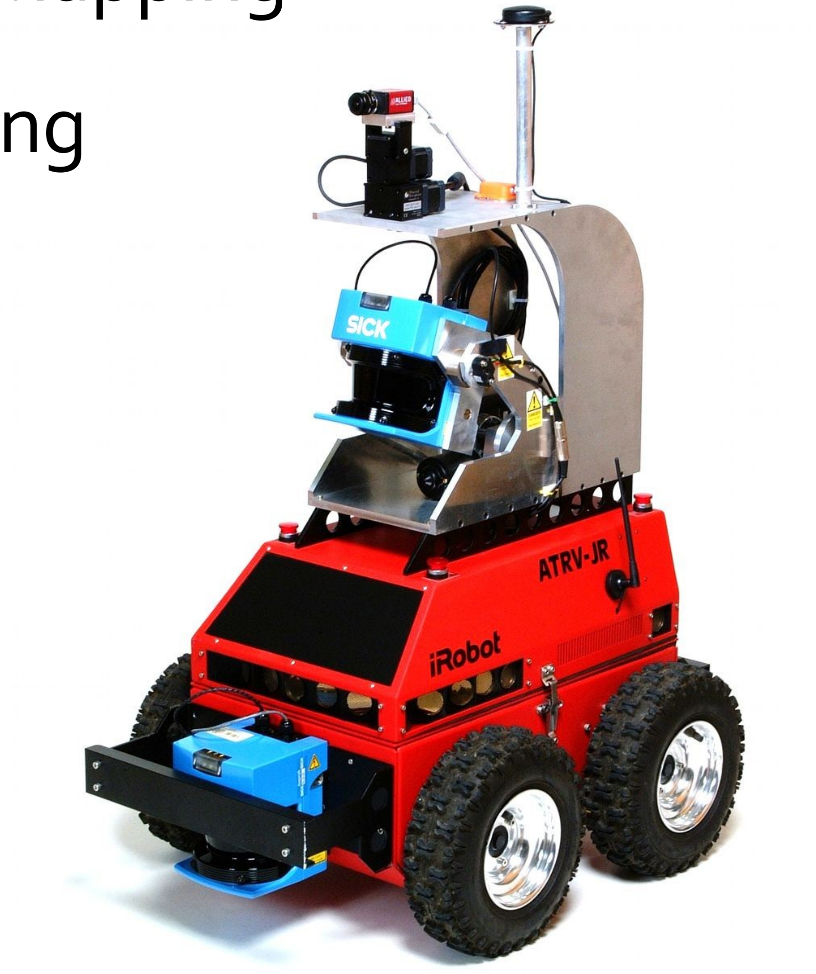
- AR with a hand-held camera
- Visual Tracking provides registration
- Track *without* prior model of world
- Challenges:
  - Speed
  - Accuracy
  - Robustness
  - Interaction with real world

# Existing attempts: SLAM

- Simultaneous Localisation and Mapping
- Well-established in robotics (using a rich array of sensors)

# Existing attempts: SLAM

- Simultaneous Localisation and Mapping
- Well-established in robotics (using a rich array of sensors)



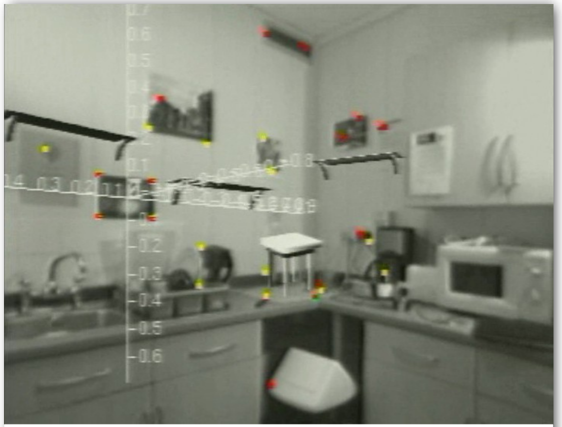
Courtesy of Oxford Mobile Robotics Group

# Existing attempts: SLAM

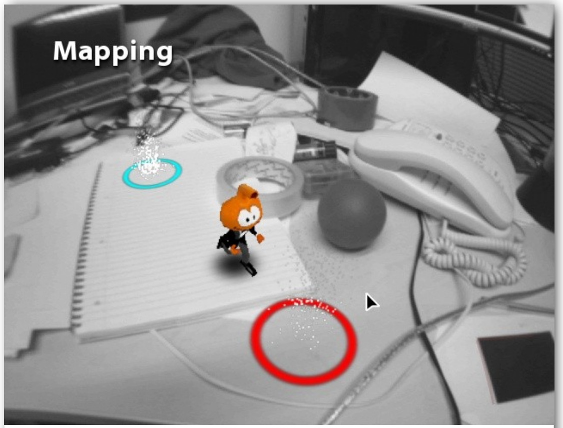
- Simultaneous Localisation and Mapping
- Well-established in robotics (using a rich array of sensors)
- Demonstrated with a single hand-held camera by Davison 2003

Courtesy of Oxford Mobile Robotics Group

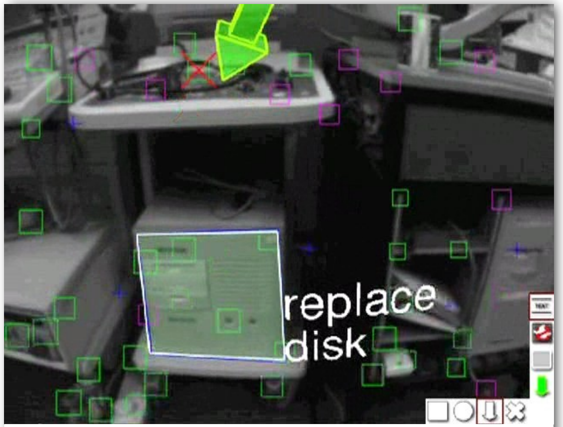
# SLAM applied to AR



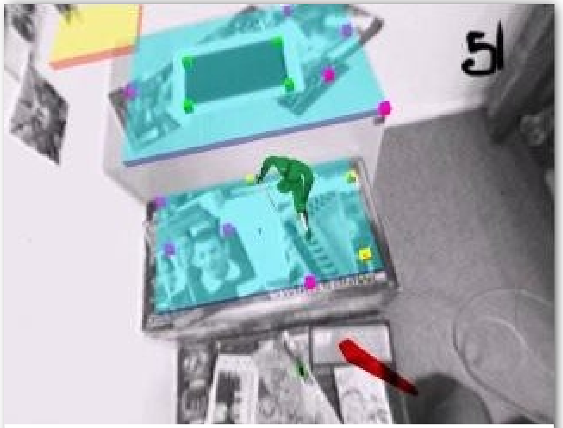
Davison *et al* 2004



Williams *et al*  ICCV 2007

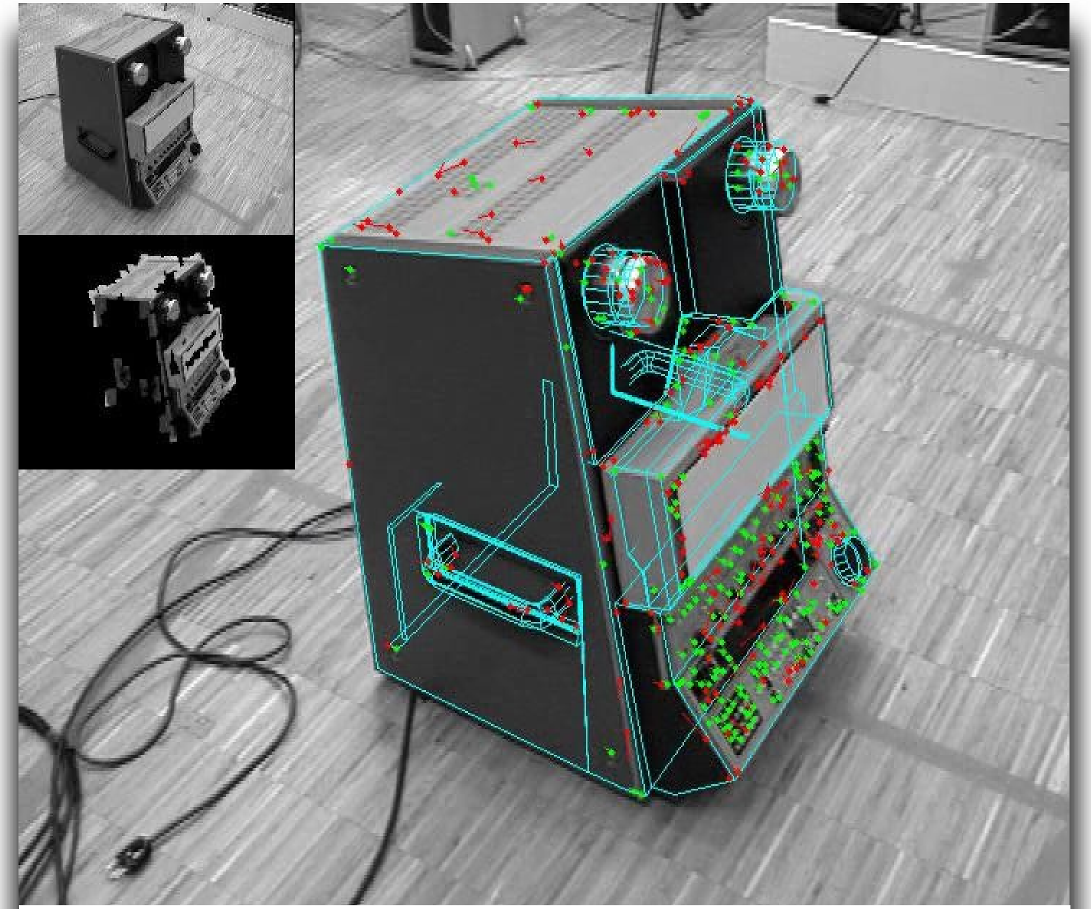


Reitmayr *et al*  ISMAR 2007



Checklov *et al*  ISMAR 2007

# Model-based tracking vs SLAM

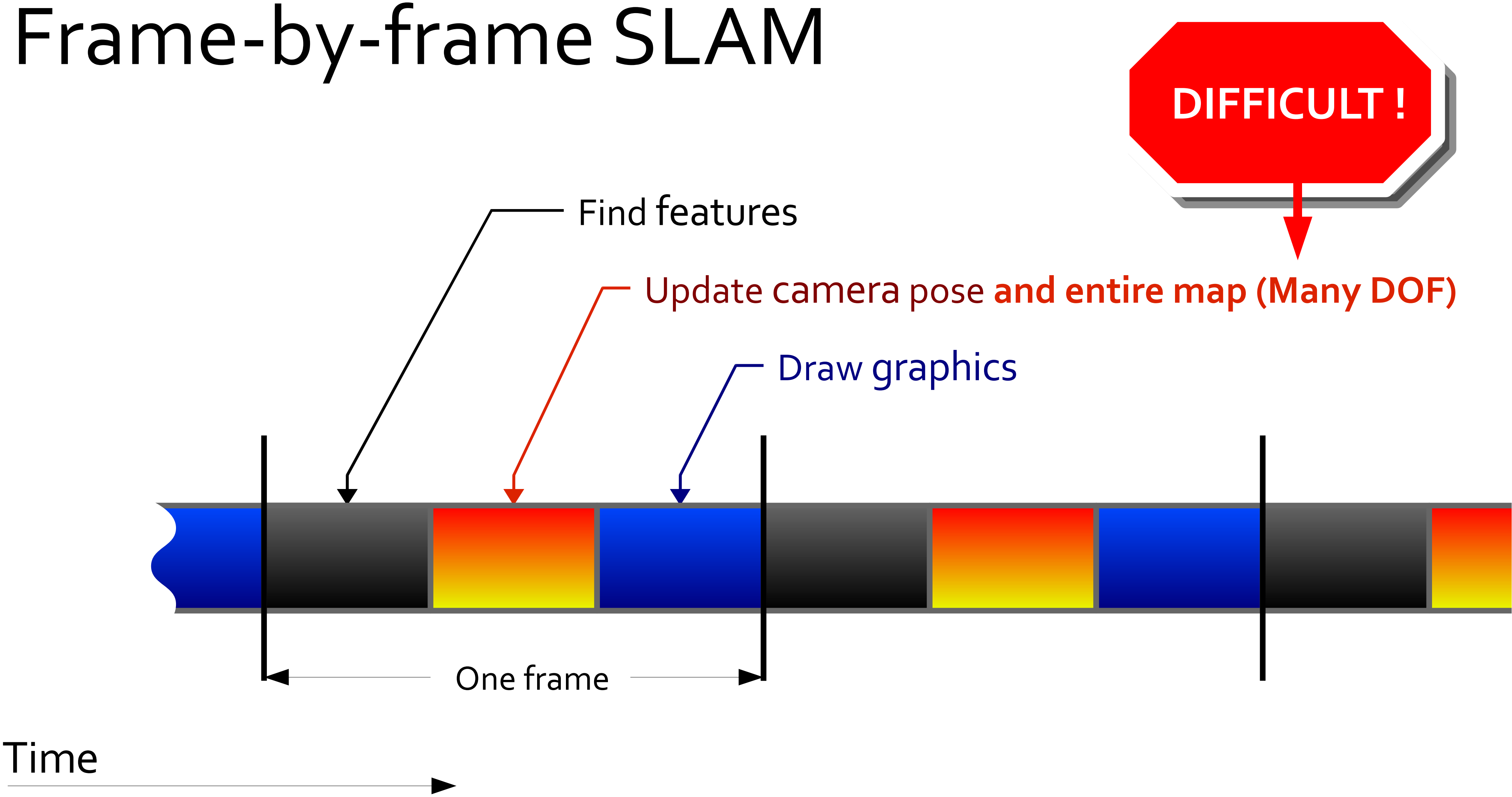


Lepetit, Vachetti & Fua ISMAR 2003

# Model-based tracking vs SLAM

- Model-based tracking is
  - More robust
  - More accurate
- Why?
  - SLAM fundamentally harder?

# Frame-by-frame SLAM



DIFFICULT !

Find features

Update camera pose **and entire map (Many DOF)**

Draw graphics

One frame

Time

# Frame-by-frame SLAM

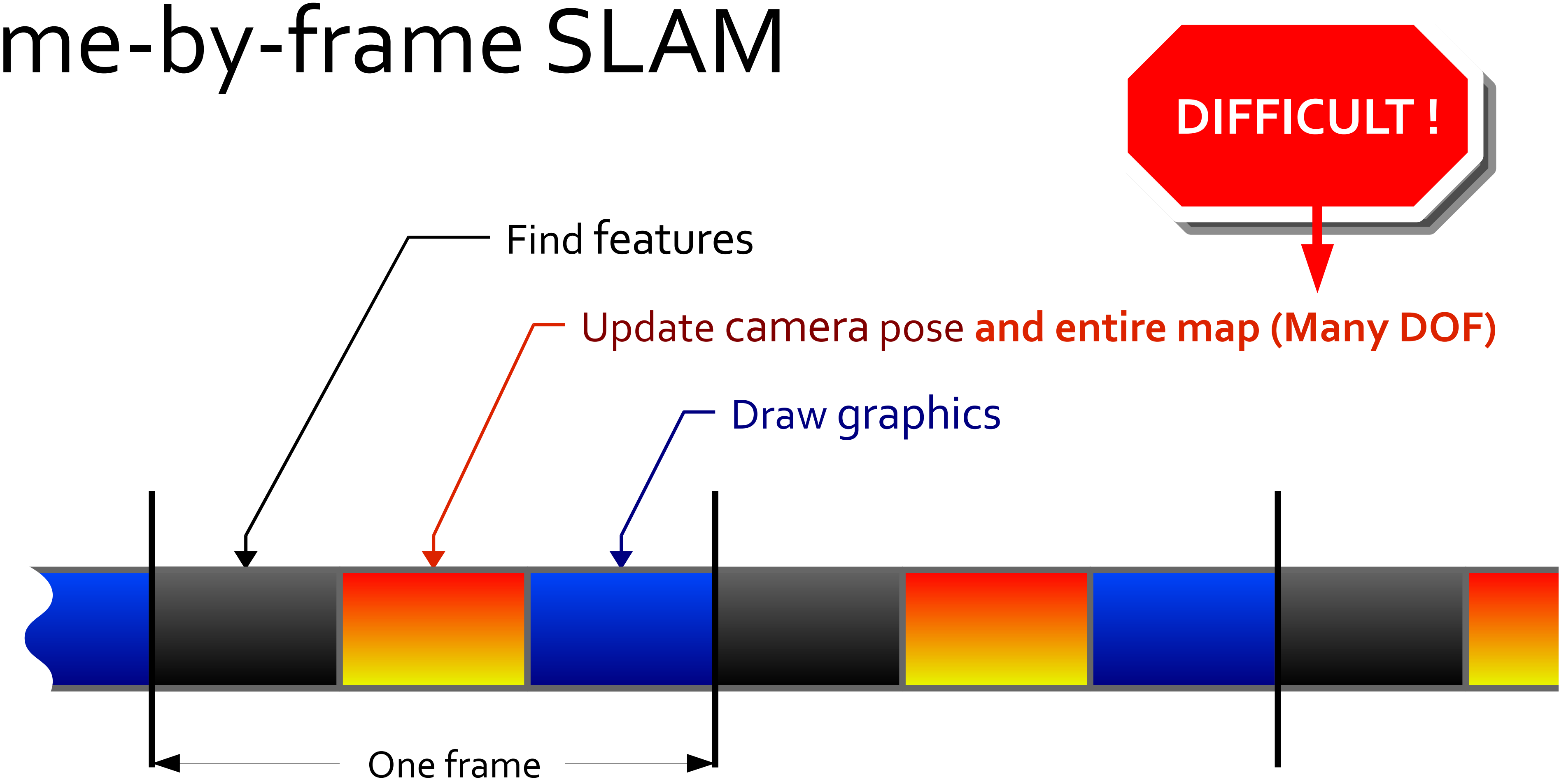
- Updating entire map every frame is expensive
- Mandates “sparse map of high-quality features”

- A. Davison

# Our approach

- Use dense map (of low-quality features)
- Don't update the map every frame: **Keyframes**
- Split the tracking and mapping into two threads

# Frame-by-frame SLAM



DIFFICULT !

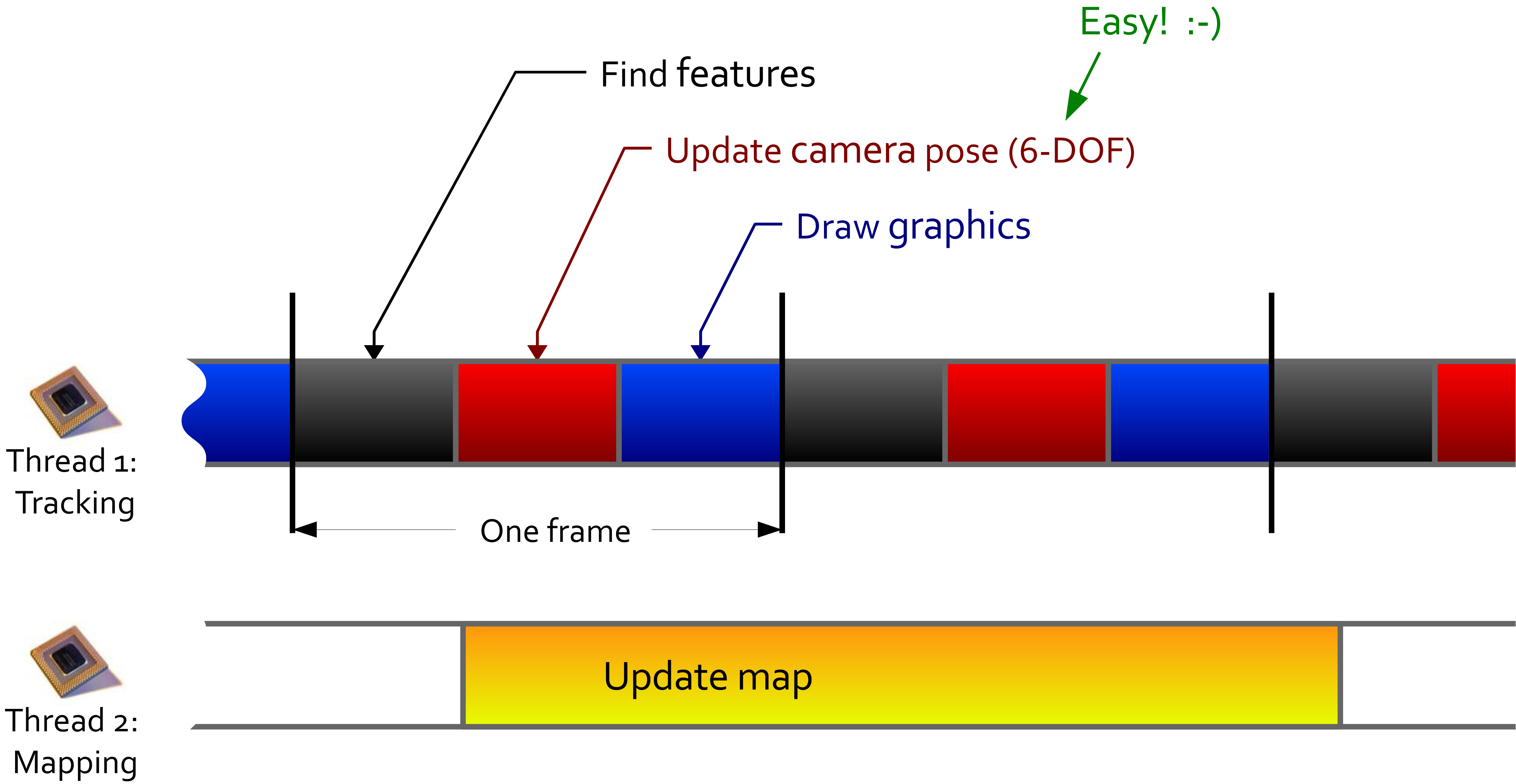
Find features

Update camera pose **and entire map (Many DOF)**

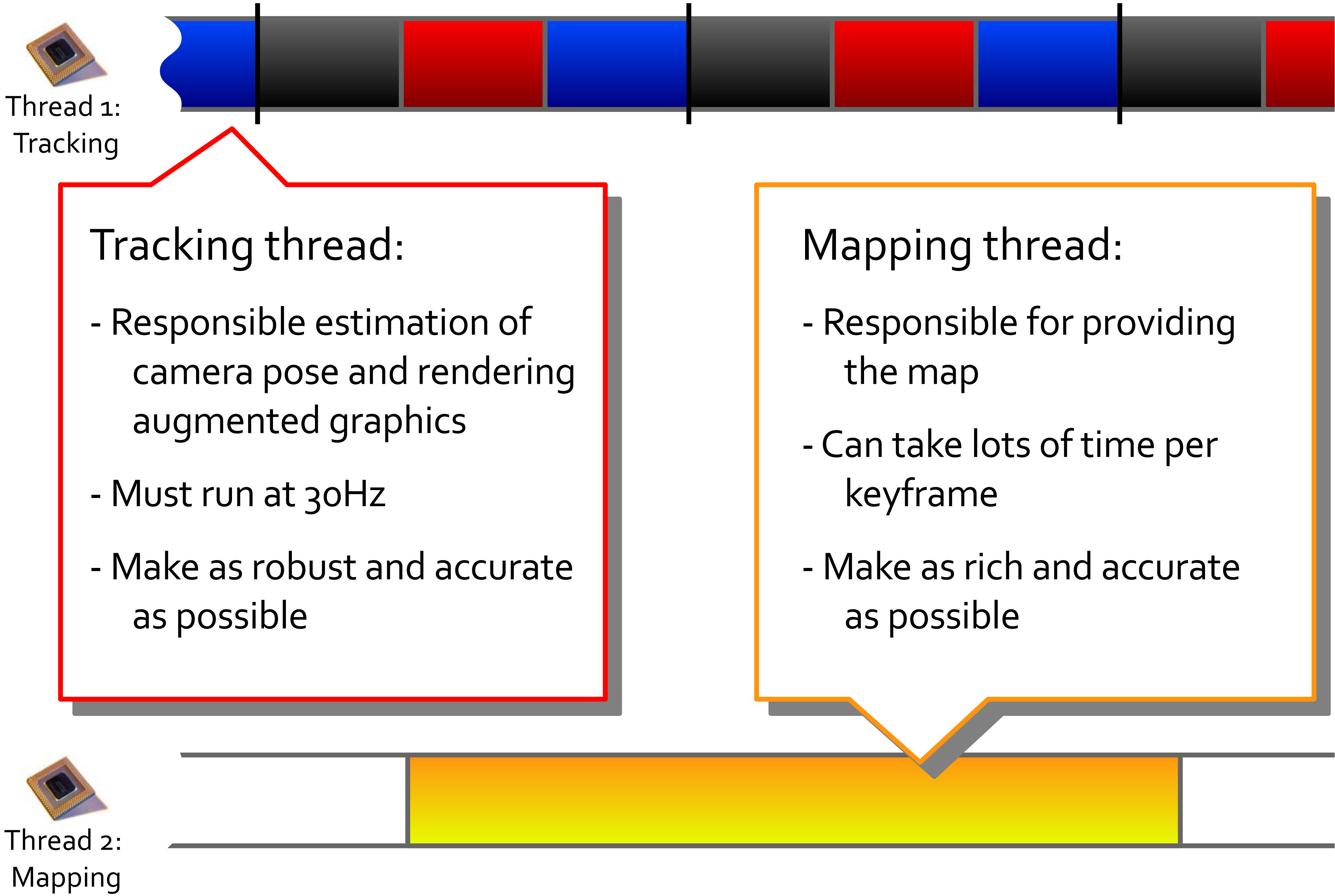
Draw graphics

One frame

# Parallel Tracking and Mapping



# Parallel Tracking and Mapping for Small AR Workspaces



## Tracking thread:

- Responsible estimation of camera pose and rendering augmented graphics

- Must run at 30Hz

- Make as robust and accurate as possible

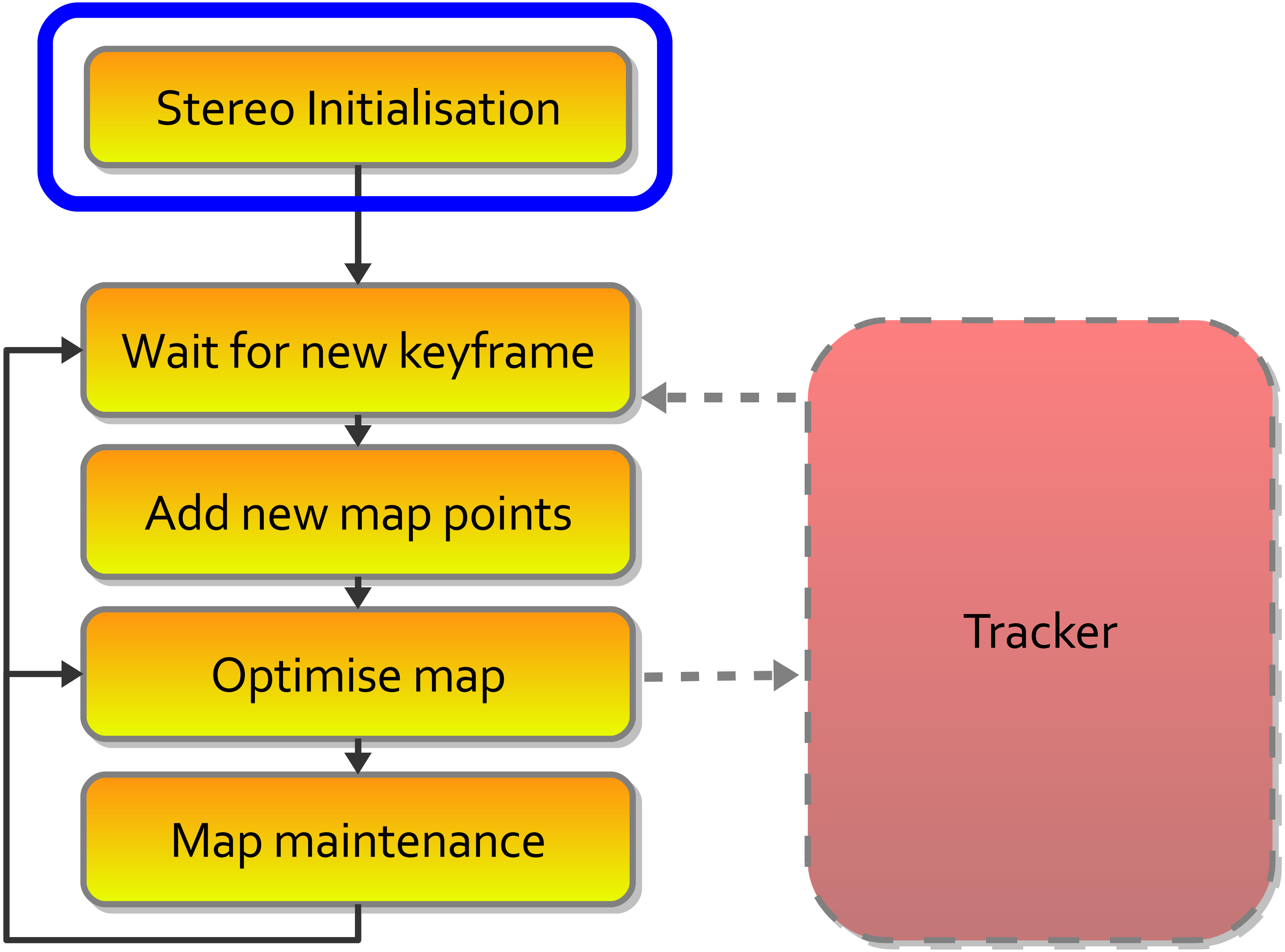
## Mapping thread:

- Responsible for providing the map

- Can take lots of time per keyframe

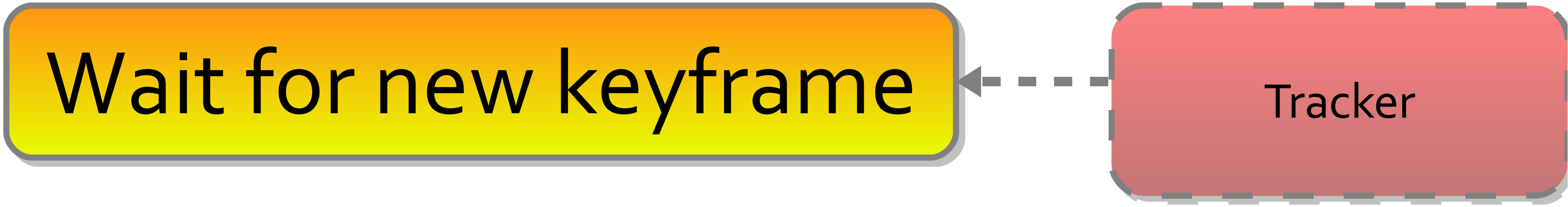
- Make as rich and accurate as possible

# Mapping thread



# Stereo Initialisation

- Use five-point-pose algorithm (Stewenius et al '06)
- Requires a pair of frames and feature correspondences
- Provides initial map
- User input required:
  - Two clicks for two keyframes
  - Smooth motion for feature correspondence



Wait for new keyframe

Tracker

- Keyframes are only added if:
  - There is a baseline to the other keyframes
  - Tracking quality is good
- When a keyframe is added:
  - The mapping thread stops whatever it is doing
  - All points in the map are measured in the keyframe
  - New map points are found and added to the map

# Add new map points

- Want as many map points as possible
- Check **all** maximal FAST corners in the keyframe:
  - Check Shi-Tomasi score
  - Check if already in map
- Epipolar search in a neighbouring keyframe
- Triangulate matches and add to map
- Repeat in four image pyramid levels

# Optimise map

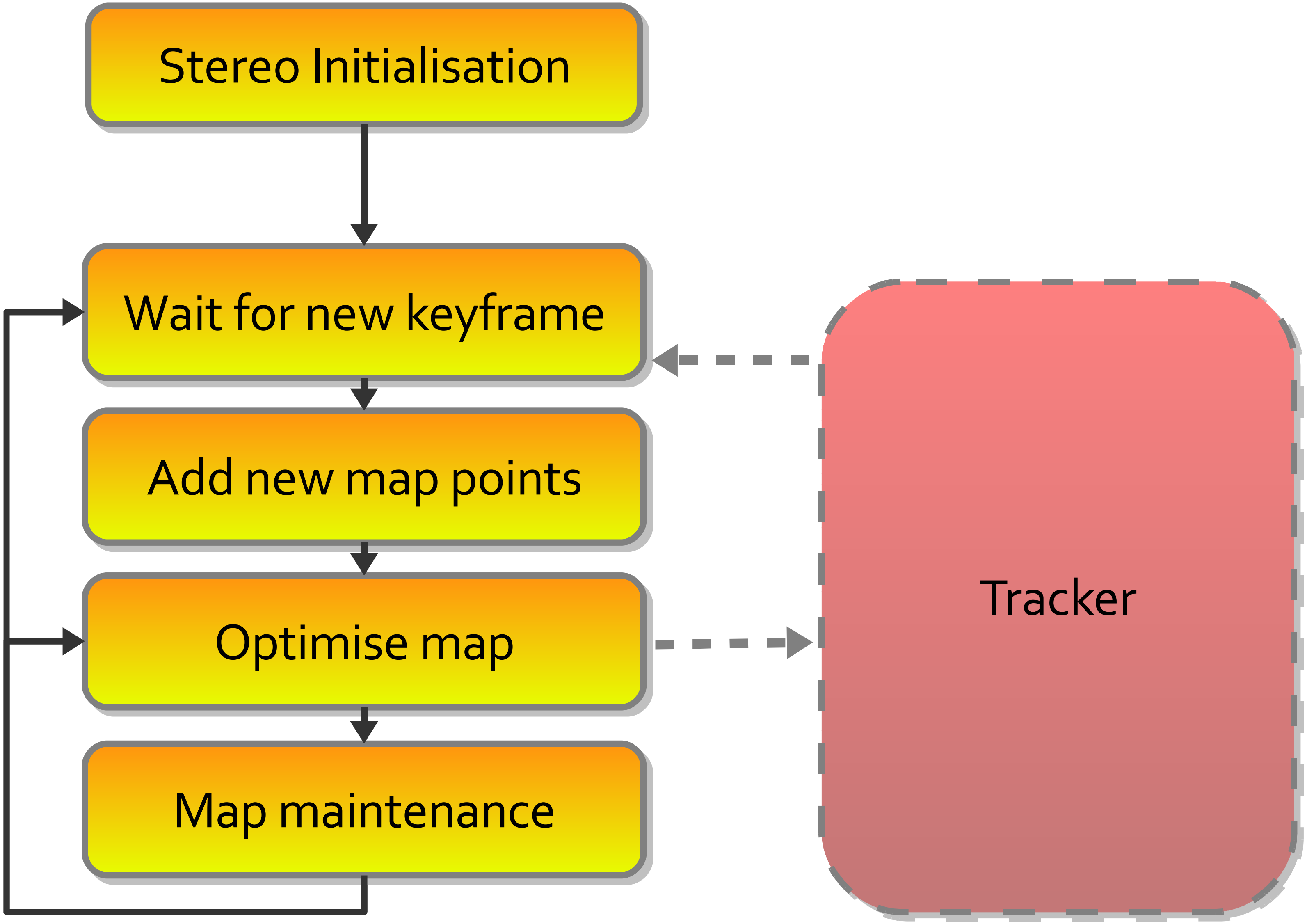
- Use batch SFM method: **Bundle Adjustment***
- Adjusts map point positions and keyframe poses
- Minimises reprojection error of all points in all keyframes (or use only last N keyframes)
- Cubic complexity with keyframes, linear with map points
- Compatible with M-estimators (we use Tukey)

* - According to Engels, Stewenius and Nister, this **rules**

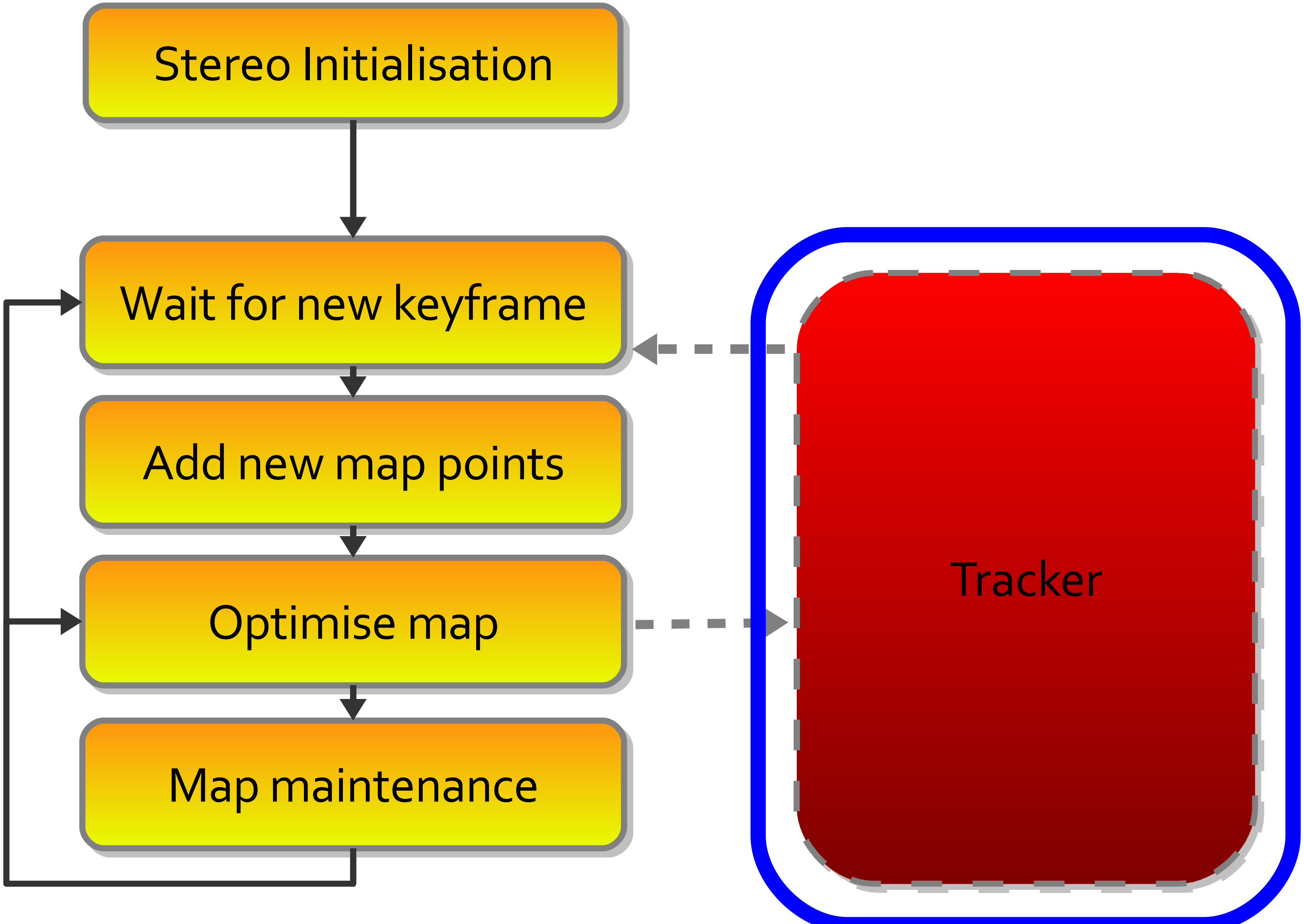
# Map Maintenance

- When camera is not exploring, mapping thread has idle time – use this to improve the map
- Data association in bundle adjustment is reversible
- Re-attempt outlier measurements
- Try to measure new map features in all old keyframes

# Mapping thread



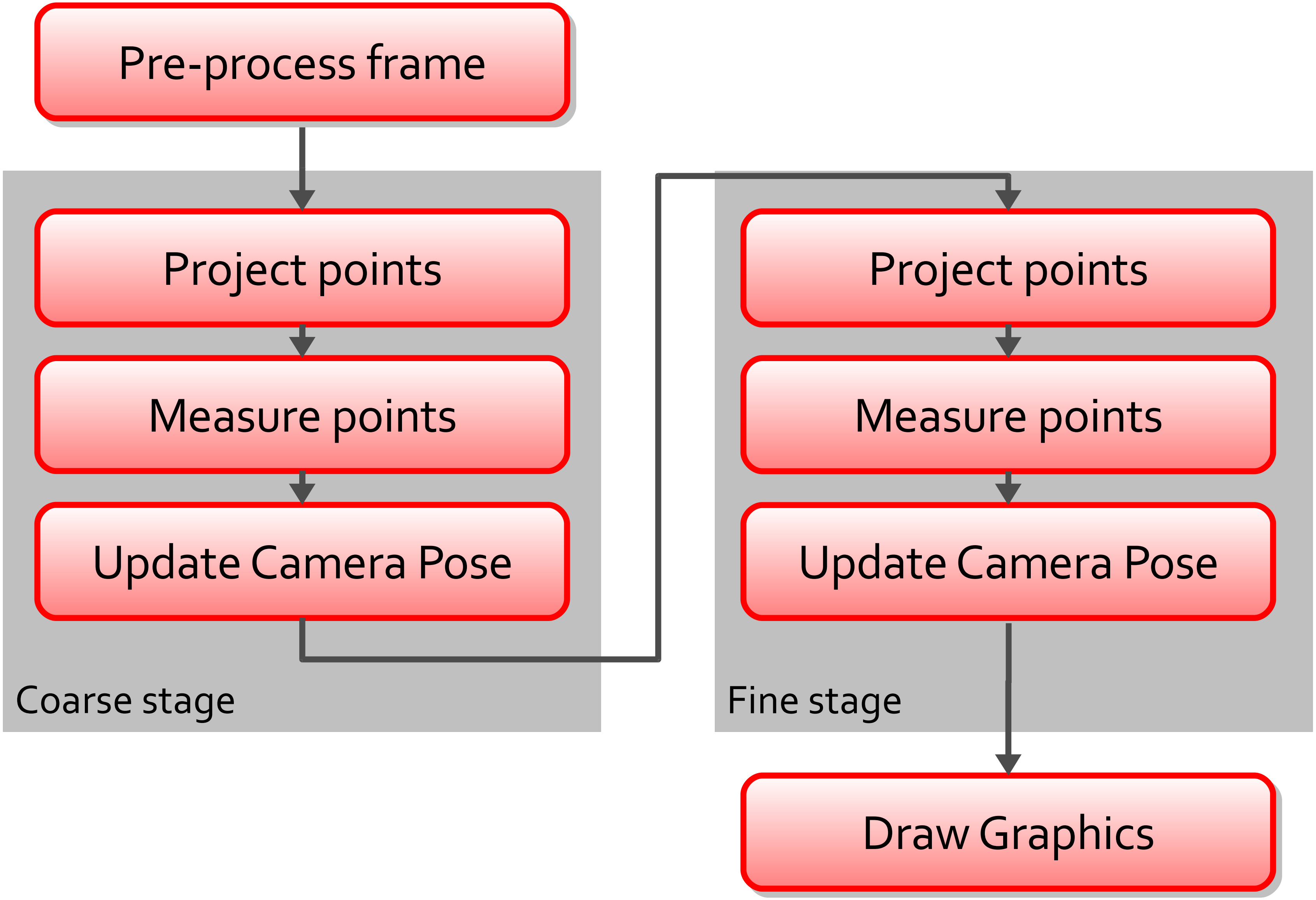
# Tracking thread



# Tracking thread

- Responsible estimation of camera pose and rendering augmented graphics
- Must run at 30Hz
- Make as robust and accurate as possible
- Track/render loop with two tracking stages

# Tracking thread



Pre-process frame

Coarse stage

Project points

Measure points

Update Camera Pose

Fine stage

Project points

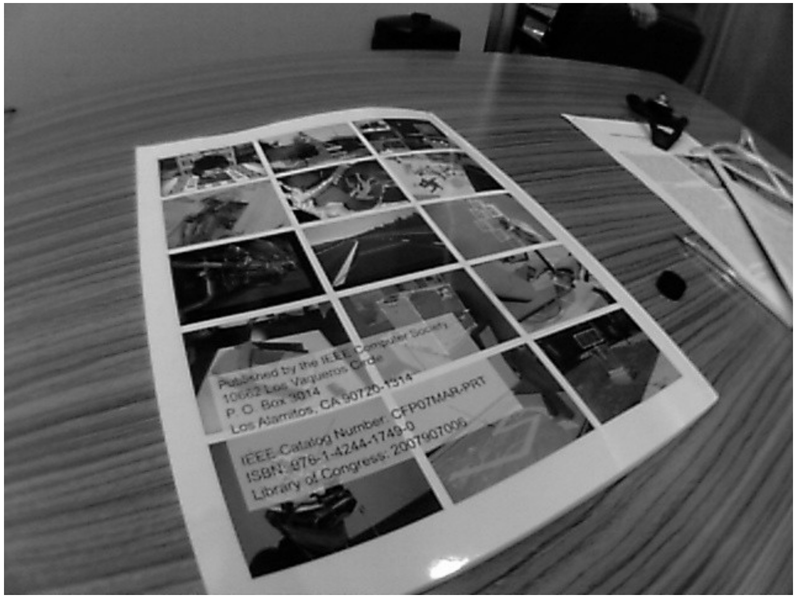
Measure points

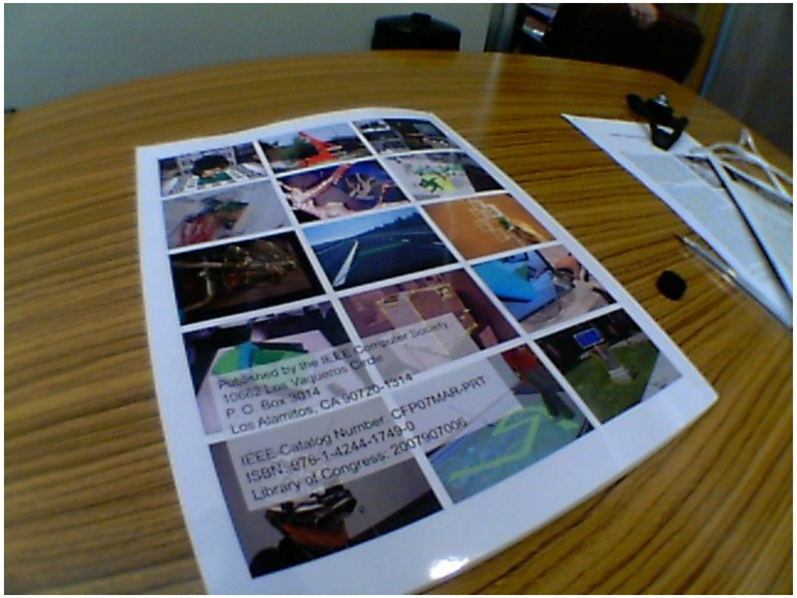
Update Camera Pose

Draw Graphics

# Pre-process frame

- Make mono and RGB version of image


Published by the IEEE Computer Society
10662 Los Vaqueros Circle
P. O. Box 3014
Los Alamitos, CA 90720-1314
IEEE Catalog Number: CFP07MAR-PRT
ISBN: 978-1-4244-1749-0
Library of Congress: 2007907006


Published by the IEEE Computer Society
10662 Los Vaqueros Circle
P. O. Box 3014
Los Alamitos, CA 90720-1314
IEEE Catalog Number: CFP07MAR-PRT
ISBN: 978-1-4244-1749-0
Library of Congress: 2007907006

# Pre-process frame

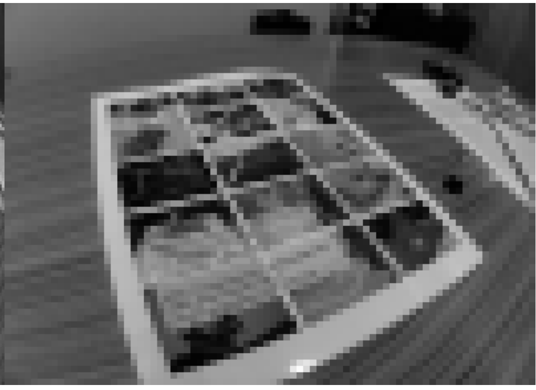
- Make mono and RGB version of image
- Make four pyramid levels

640x480

320x240

160x120

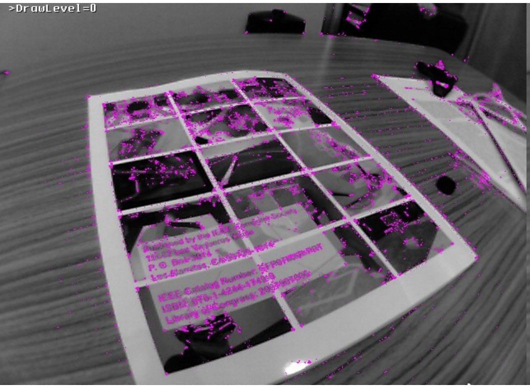
80x60



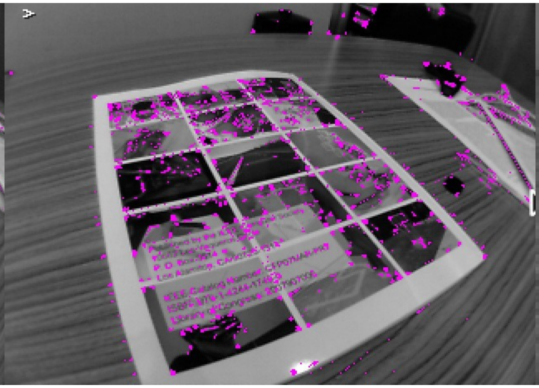
# Pre-process frame

- Make mono and RGB version of image
- Make four pyramid levels
- Detect FAST corners

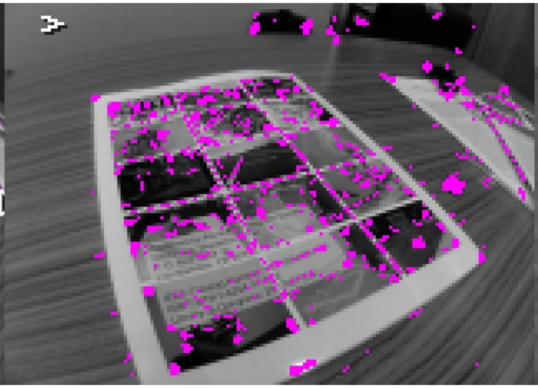
640x480



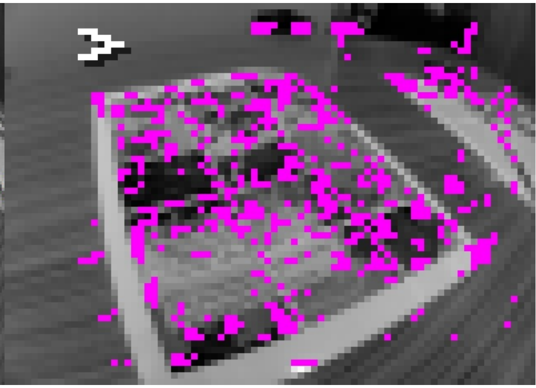
320x240



160x120



80x60



# Project Points

- Use motion model to update camera pose
- Project all map points into image to see which are visible, and at what pyramid level
- Choose subset to measure
  - ~50 biggest features for coarse stage
  - 1000 randomly selected for fine stage

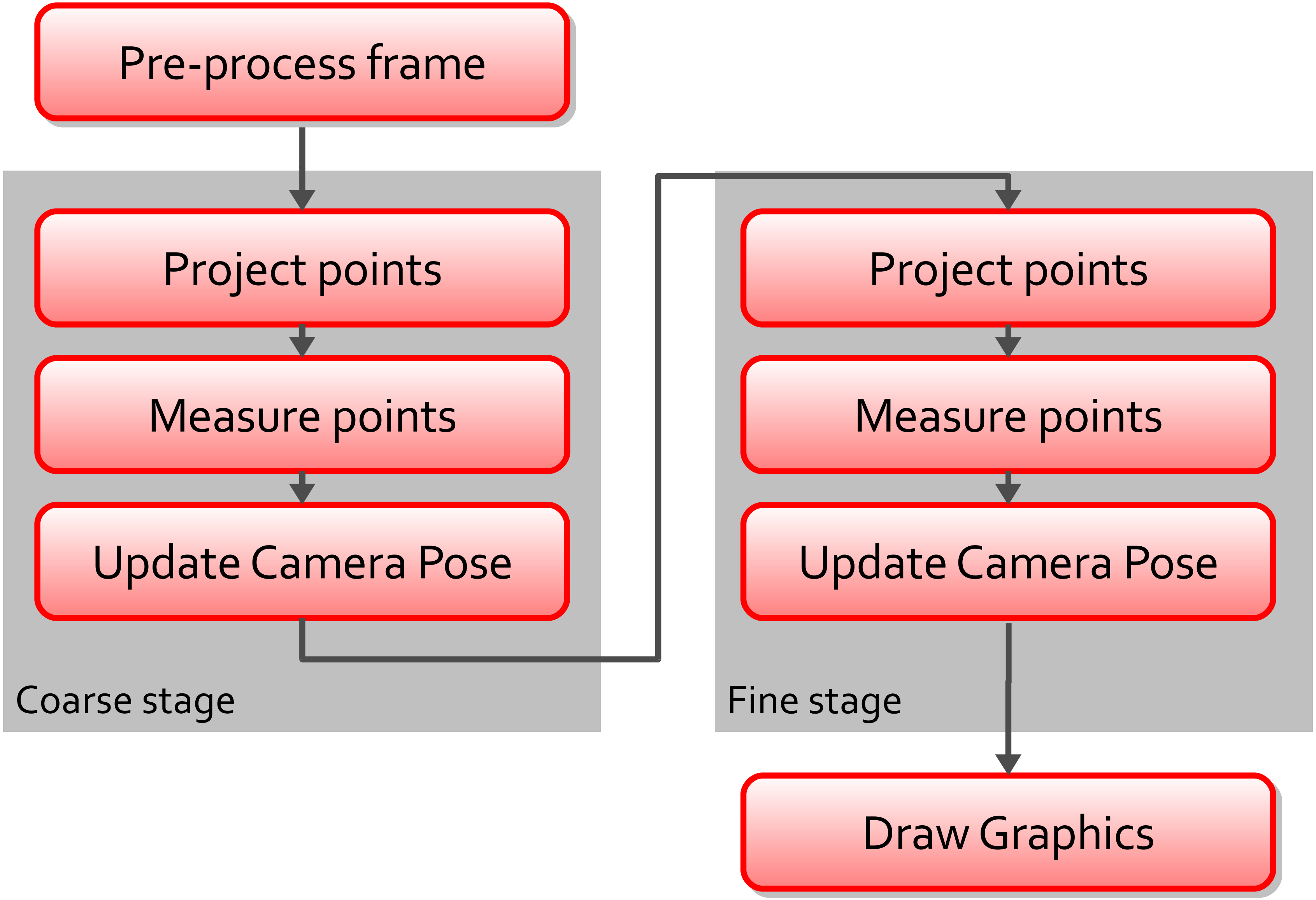
# Measure Points

- Generate 8x8 matching template (warped from source keyframe)
- Search a fixed radius around projected position
  - Use zero-mean SSD
  - Only search at FAST corner points
- Up to 10 inverse composition iterations for sub-pixel position (for some patches)
- Typically find 60-70% of patches

# Update camera pose

- 6-DOF problem
- 10 IRWLS iterations
- Tukey M-Estimator

# Tracking thread



Pre-process frame

Project points

Measure points

Update Camera Pose

Coarse stage

Project points

Measure points

Update Camera Pose

Fine stage

Draw Graphics

# Draw graphics

- What can we draw in an unknown scene?
  - Assume single plane visible at start
  - Run VR simulation on the plane
- Radial distortion
- Want proper blending

# Parallel Tracking and Mapping for Small AR  Workspaces

Input Image 640x480

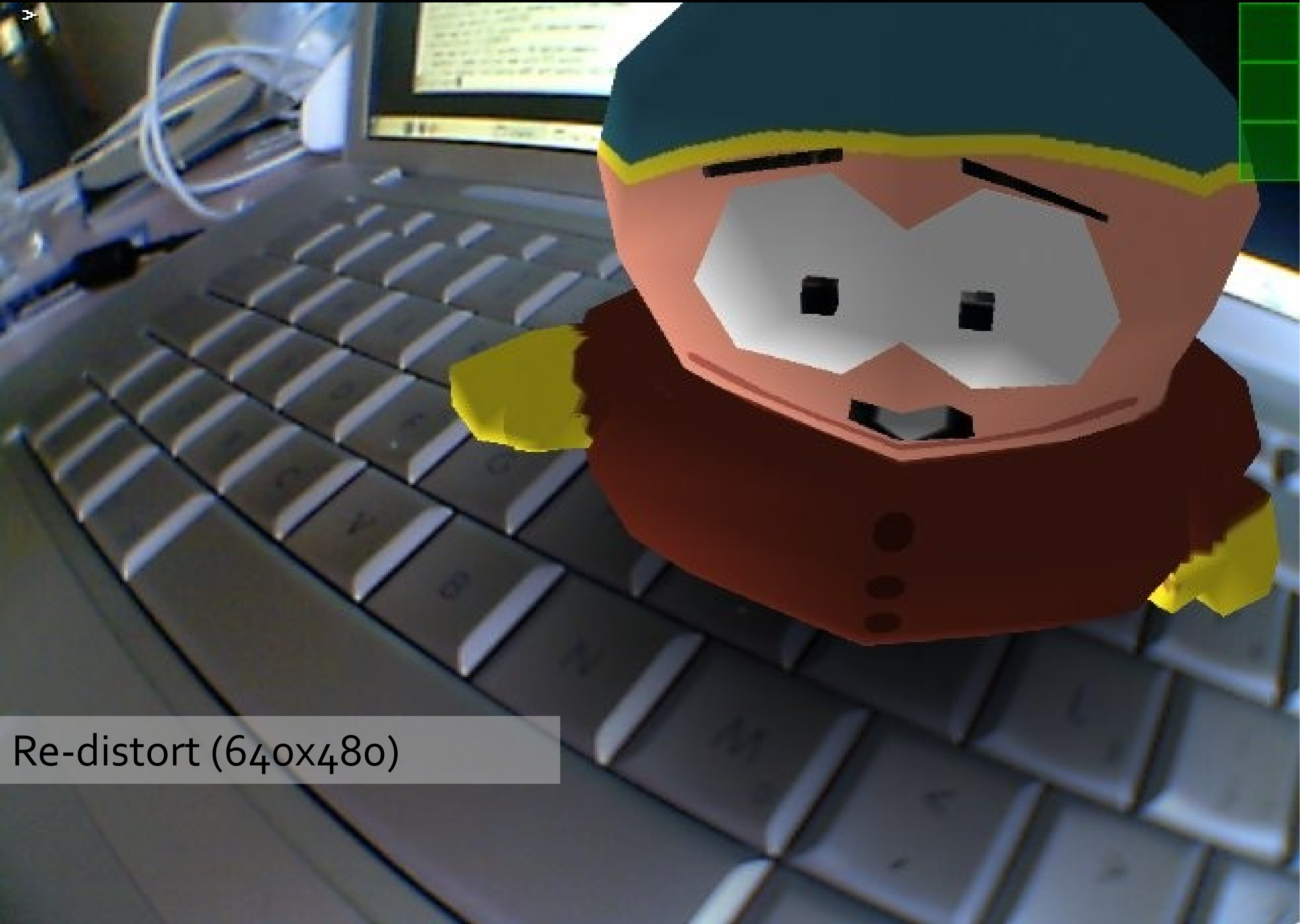
# Parallel Tracking and Mapping for Small AR Workspaces

Undistort 1600x1200

# Parallel Tracking and Mapping for Small AR Workspaces

Render 1600x1200

# Parallel Tracking and Mapping for Small AR  Workspaces



Re-distort (640x480)

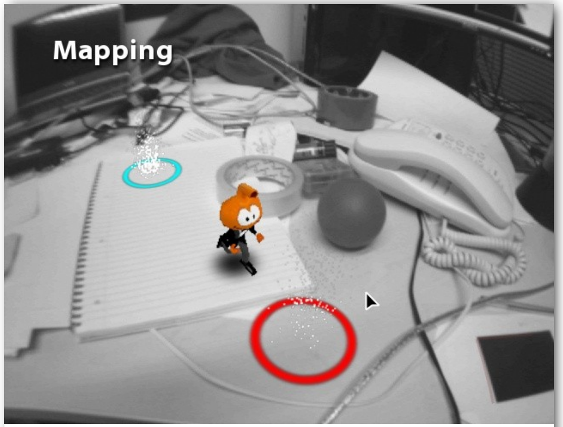
# Tracking Quality Monitoring

- Heuristic check based on fraction of found measurements
- Three quality levels: Good, Poor, Lost
- Only add to map on `Good'
- Stop tracking and relocalise on `Lost'

# Results

- Is it any good?
  - Yes

# Comparison to EKF-SLAM



EKF-SLAM ICCV 2007

ISMAR 2007

- More accurate
- More robust
- Faster tracking

# Results Video

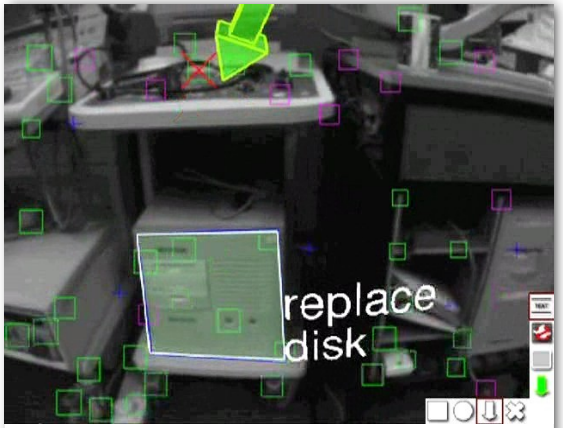
ISMAR 2007

# Areas in need of improvement

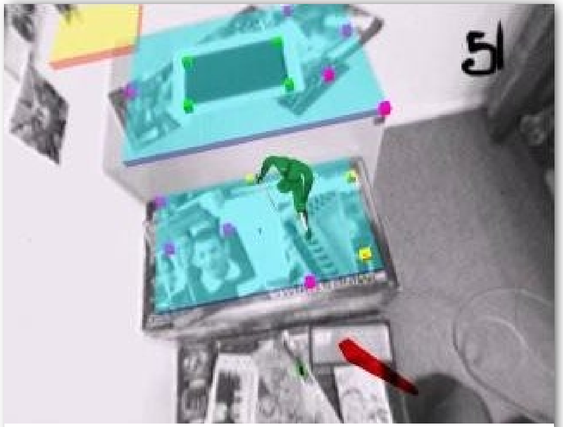
- Outlier management
- Still brittle in some scenarios
  - Fine repeated texture
  - Stereo initialisation
- **No occlusion reasoning**
- **Point cloud is inadequate for AR**

- **No occlusion reasoning**
- **Point cloud is inadequate for AR**

- User interaction?
- Automatic primitive detection?



Reitmayr *et al* ISMAR 2007



Checklov *et al* ISMAR 2007

- **Live dense reconstruction?**

# Parallel Tracking and Mapping for Small AR Workspaces

Georg Klein and David Murray
Active Vision Lab, Oxford