Module IN 2018

Introduction to Augmented Reality

Prof. Gudrun Klinker

Markerless Optical Tracking and Feature Detection

SS 2018

...

Literature

- *Parallel Tracking and Mapping for Small AR Workspaces*, G. Klein and D. Murray, ISMAR 2007.
- *Parallel Tracking and Mapping on a Camera Phone*, G. Klein and D. Murray, ISMAR 2009.

- *MonoSLAM: Real-Time Single Camera SLAM*, A.J. Davison, I. Reid, N. Molton and O. Stasse, IEEE Trans. PAMI 2007 29(6): 1052-1067 (2007).

Agenda

1. Motivation
2. Introduction to „Simultaneous Localization-And-Mapping“ (SLAM)
3. PTAM: System Overview
4. Feature Map
5. Tracking
6. Mapping
7. Results

1. Motivation

Typical situation in Augmented Reality

- Mobile user
  - Local (inside-out) tracker (here: camera in user's hand or on HMD)
  - Fast, erratic motions (esp. rotations)
- Unprepared environment:
  - No markers
  - No 3D scene description (3D model)
  - No external (outside-in) observing system
- ~~Even worse:~~
  - ~~Moving (changing) objects~~
  - ~~Changing illumination~~
  - ~~Very large area~~
- Need for „Extensible Tracking“

Agenda

1. Motivation
2. Introduction to „Simultaneous Localization-And-Mapping“ (SLAM)
3. PTAM: System Overview
4. Feature Map
5. Tracking
6. Mapping
7. Results

2. SLAM

Similar situation in robotics (mobile vehicle with camera)

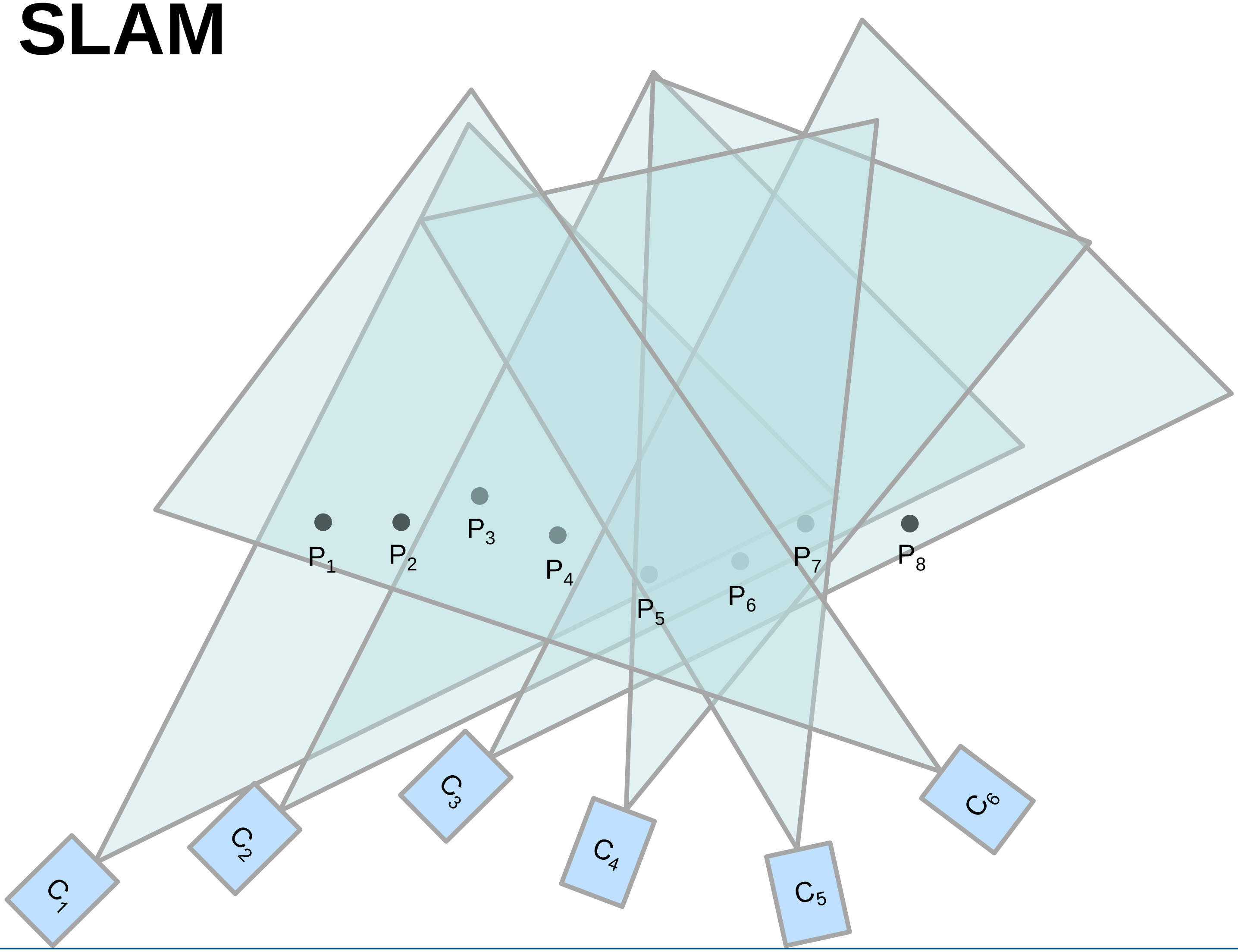
- **SLAM** algorithm (Simultaneous Localization And Map-Building) with a single camera: [Davison 1998] http://www.doc.ic.ac.uk/~ajd/publications.html

2. SLAM

General idea

- Two separate issues
  - Localization
    - By some sensors
    - By a camera – if a scene model (map) is available
  - Map-building: 3D scene reconstruction
    - By generalized stereo
    - By a single, moving camera – if motion is known
- If only a single camera is used and no external scene description is available, both problems need to be solved simultaneously (chicken-and-egg problem).

2. SLAM



2. SLAM

$$\begin{bmatrix} X_{ij}/Z_{ij} \\ Y_{ij}/Z_{ij} \\ 1 \\ 1/Z_{ij} \end{bmatrix} \approx \begin{bmatrix} X_{ij} \\ Y_{ij} \\ Z_{ij} \\ 1 \end{bmatrix} = \begin{bmatrix} f_x & s & c_x & 0 \\ 0 & f_y & c_y & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \bullet \begin{bmatrix} r_{11} & r_{12} & r_{13} & t_x \\ r_{21} & r_{22} & r_{23} & t_y \\ r_{31} & r_{32} & r_{33} & t_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \bullet \begin{bmatrix} x_i \\ y_i \\ z_i \\ 1 \end{bmatrix}$$

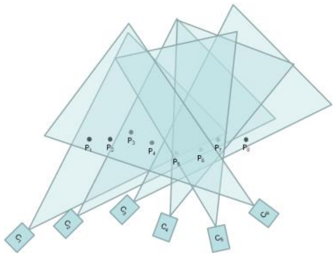


Image formation equation: $\overrightarrow{P_{ij}} = K_j[R_j|t_j]\overrightarrow{p_i}$

Unknowns

- Position (x_i,y_i,z_i) of every point $\overrightarrow{p_i}$, with i ε {0..M-1} → 3M
- Camera pose $C_j = [R_j|t_j]$ with j ε {0..N-1} → 6N
  = f(tx_j,ty_j,tz_j,rx_j,ry_j,rz_j)
  (assume: intrinsic parameters K_j to be known)

→ 3M + 6N

Givens (for points that are seen in all images)

- Image position $\overrightarrow{P_{ij}}$ = (X_{ij},Y_{ij}) of Point $\overrightarrow{p_i}$ in camera image C_j → 2MN

Required points and images for an over-determined system of equations

$$2MN \geq 3M + 6N$$

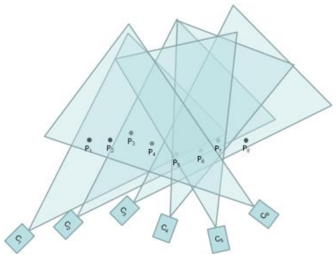
$$N \geq \frac{3M}{2M-6}$$

M points	N images
4	6
5	4
6	3
7	3

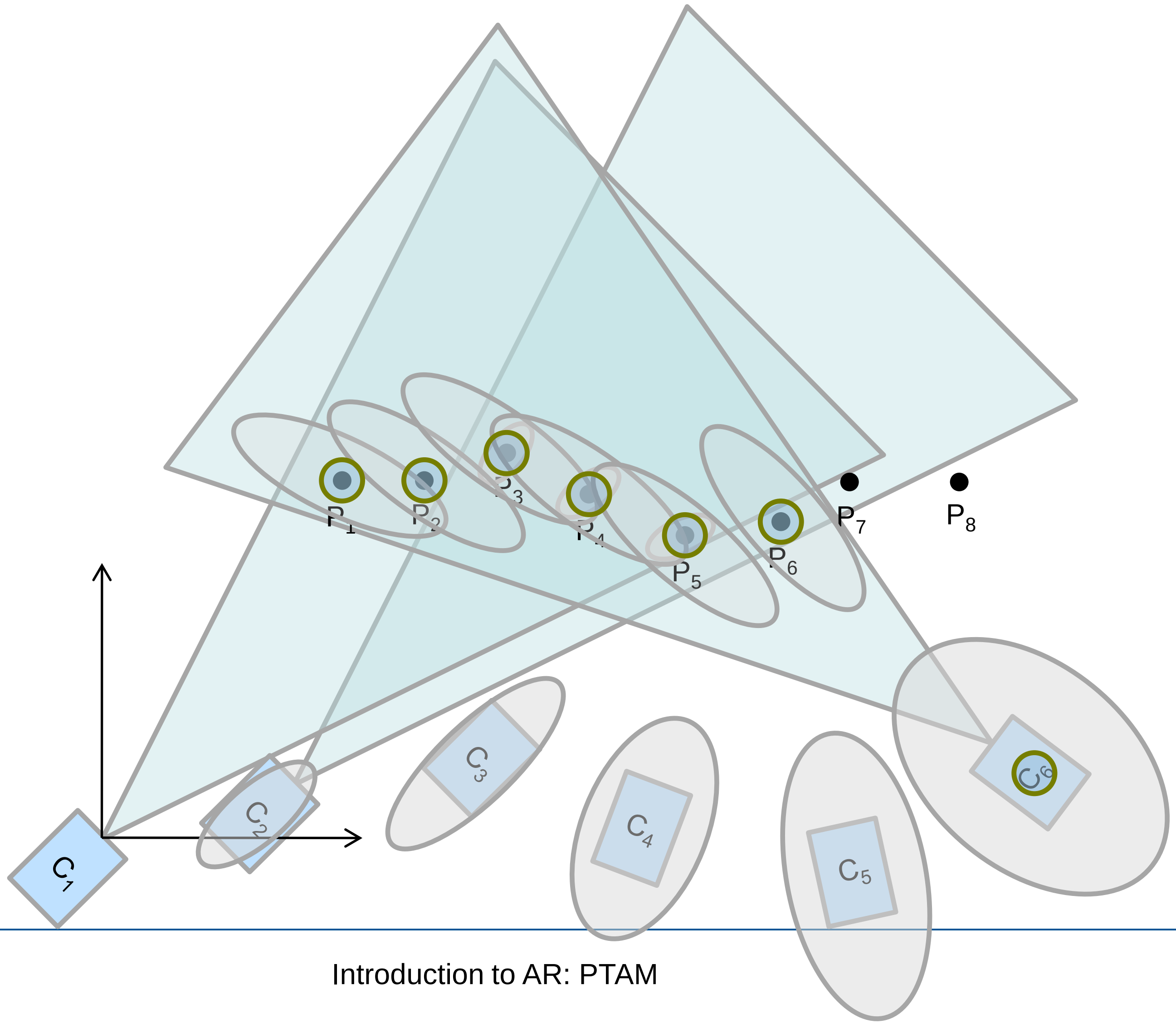
Also known as „Structure from Motion“ problem

2. SLAM

- Offline solutions
  - Structure from motion (computer vision)
  - Bundle adjustment (photogrammetry)
- Online solution: SLAM
  - Incremental map-building
  - Maintenance of a „system state“ that changes over time (Kalman)
  - Explicit models of measurement uncertainty and process noise (both for camera and object poses)



2. SLAM



Agenda

1. Motivation
2. Introduction to „Simultaneous Localization-And-Mapping“ (SLAM)
3. PTAM: System Overview
4. Feature Map
5. Tracking
6. Mapping
7. Results

3. PTAM: System Overview

Parallel Tracking and Mapping

Georg Klein and David Murray (Oxford University).

http://www.robots.ox.ac.uk/~gk/

- Paper: *Parallel Tracking and Mapping for Small AR Workspaces*, G. Klein and D. Murray, ISMAR 2007.
- Free source code.
- Winner of first ISMAR Tracking Contest 2008
  http://ismar08.org/wiki/doku.php?id=program-competition
  (Using Robert Castle's multiple map approach).
- Paper: *Parallel Tracking and Mapping on a Camera Phone*, G. Klein and D. Murray, ISMAR 2009.
  http://www.youtube.com/watch?v=pBI5HwitBX4

3. PTAM: System Overview

Requirements for AR-related tracking and 3D reconstruction

- Fast
- Accurate
- Robust

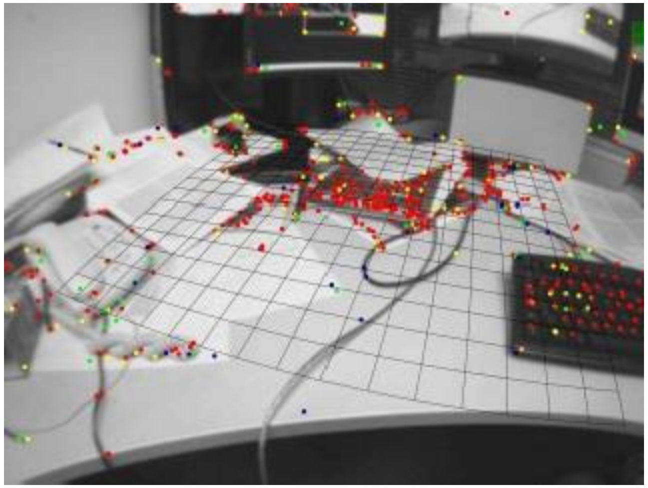
Tracking a hand-held camera is more difficult than tracking a robot because

- Robots often have odometry
- Robots can be arbitrarily slow

Problems with SLAM

- Potential data association errors (wrong matches) due to high speeds
  - Not robust enough because tracking is tied to map-building (too slow for tracking, too fast for high-quality map-building)
- Does not exploit dual-core facilities
- Sparse map of high-quality features vs. dense map of low-quality features (key frames)

3. PTAM: System Overview



Assumptions

- Mostly static scene
  (Not many moving or deformable objects)
- Small
  (User spends most time in the same space)

Main concepts

- Tracking and Mapping separated (two parallel threads)
- Mapping based on keyframes (processed using offline techniques: bundle adjustment)
- Map densely initialized from a stereo pair (5-point algorithm)
- New points initialized via epipolar search
- Thousands of points

Agenda

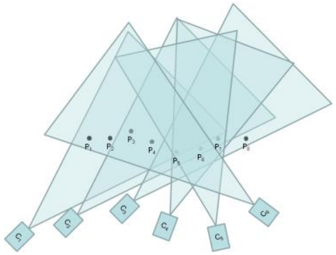
1. Motivation
2. Introduction to „Simultaneous Localization-And-Mapping“ (SLAM)
3. PTAM: System Overview
4. Feature Map
5. Tracking
6. Mapping
7. Results

4. Feature Map

- 4.1 Knowledge Representation
- 4.2 Image Pyramids
- 4.3 Keyframes

4.1 Feature Representation

- M point features $p_{jw} = (x_{jw},y_{jw},z_{jw},1)$ (in some world coordinate system W)
  - Assume: locally planar surface patch with normal $\mathbf{n}_{jw}$
  - Associated with 8x8 pixel patch in some keyframe at some level
- N keyframes (specially selected images of the moving camera)
  - Associated local coordinate system K_i
  - 4-level image pyramid



- Typical feature map contains:
  - M=2000..6000 points
  - N=40..120 keyframes

4. Feature Map

4.1 Knowledge Representation

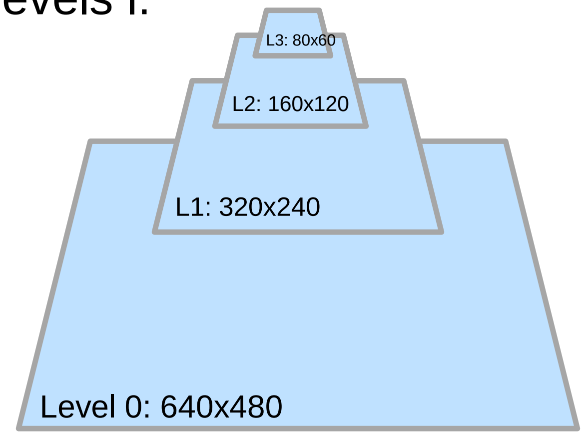
→ 4.2 Image Pyramids

4.3 Keyframes

4.2 Image Pyramids

- Pyramid of recursively smoothed and reduced representations of the original image
  - Requires less than twice the amount of space
  - Fast to compute
  - Increases robustness
  - Maintains accuracy
- Exponentially reduced image sizes at increasing levels l:

$$w_l = \frac{w}{2^l} \qquad h_l = \frac{h}{2^l}$$



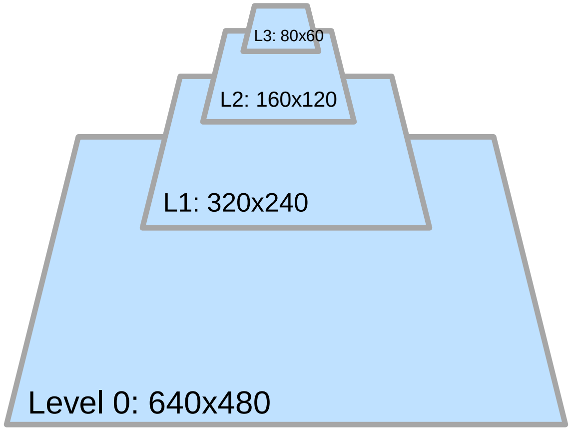
Level 0: 640 x 480 pixels
Level 1: 320 x 240 pixels
Level 2: 160 x 120 pixels
Level 3 : 80 x   60 pixels

4.2 Image Pyramids

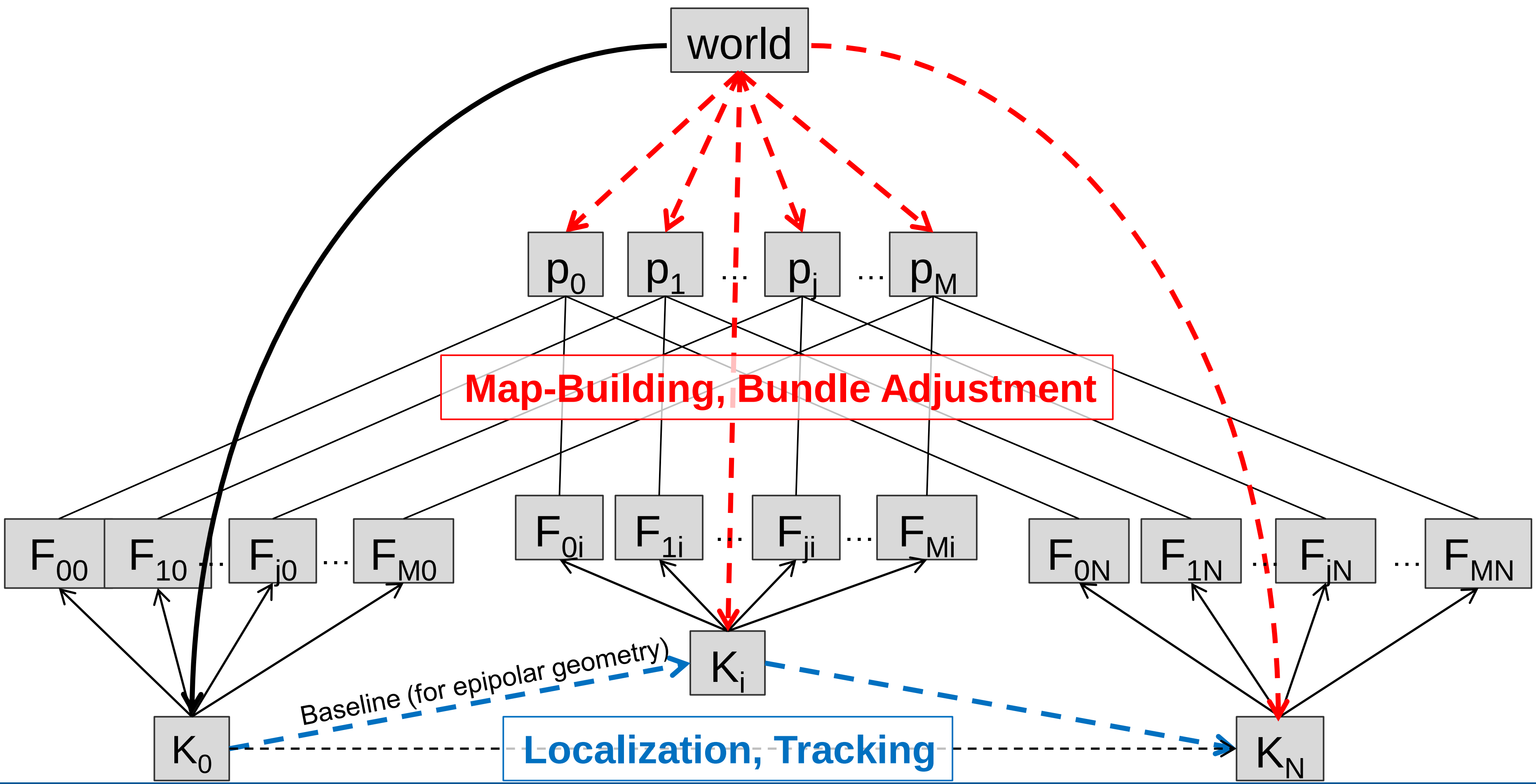
- 8x8 patches on all levels, representing areas of increasing sizes at level 0

8x8 patch at level 3:	64x64 pixels at level 0
8x8 patch at level 2:	32x32 pixels at level 0
8x8 patch at level 1:	16x16 pixels at level 0
8x8 patch at level 0:	8x 8 pixels at level 0

- Patch at coarse level = approximate description of large image area
  - Overview: only dominant image content (LF)
  - For fast wide-area search (correlation) (Redetection after fast camera movements)
- Patch a fine-grained level = precise description of small image area
  - Detail: very specific image data (HF)
  - For high-precision local-area search



4.2 SRG (Spatial Relationship Graph)



4. Feature Map

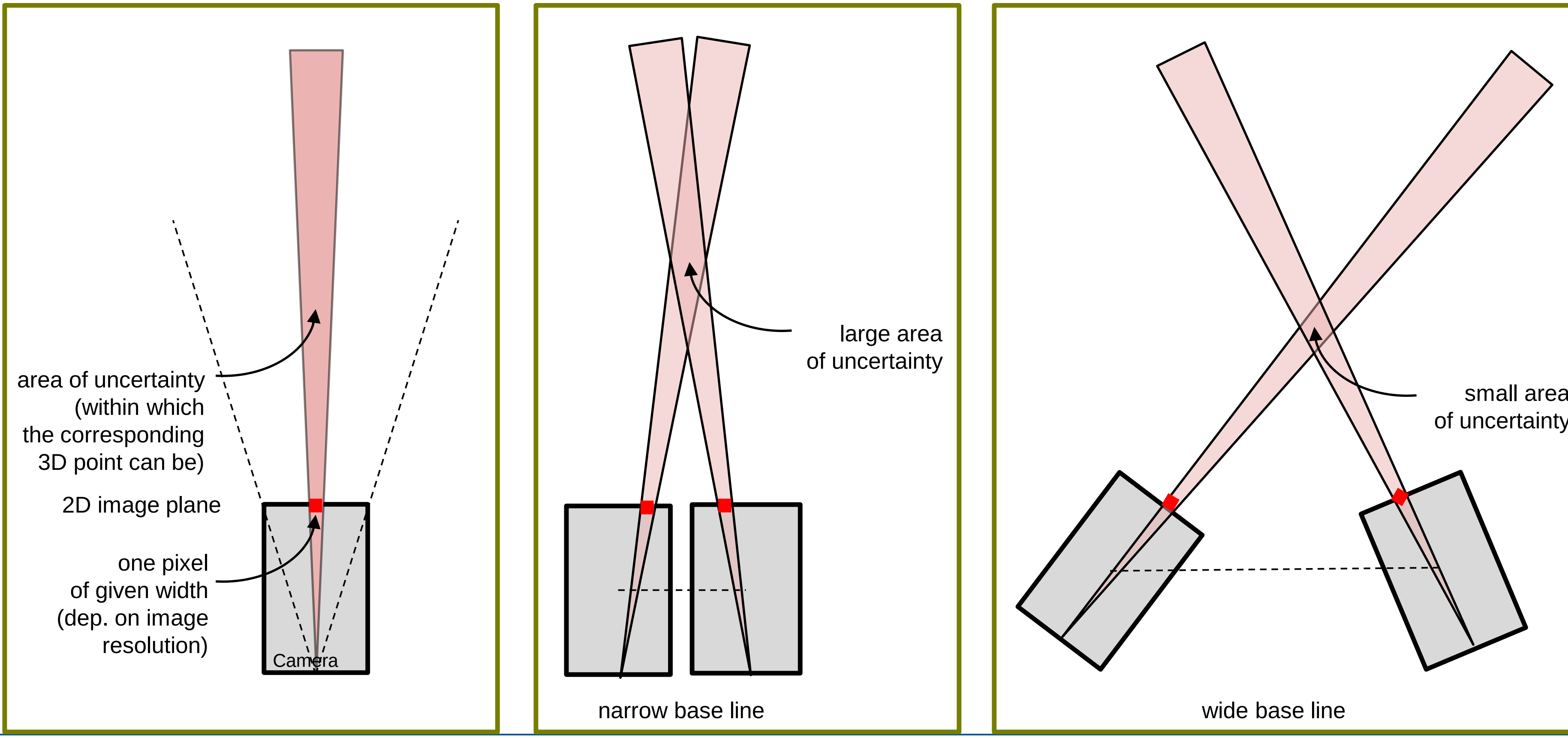
4.1 Knowledge Representation

4.2 Image Pyramids

→ 4.3 Keyframes

4.3 Keyframes

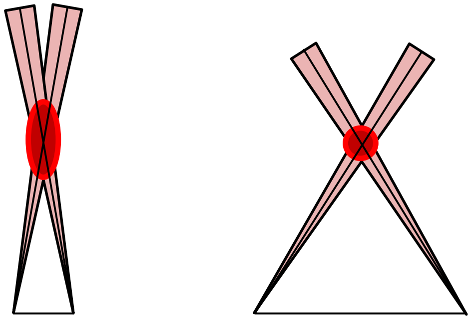
Wide baseline (distance) between cameras (views) important



4.3 Keyframes

Wide baseline (distance) between cameras (views) important

- The wider the baseline the better the depth estimation



narrow base line wide base line

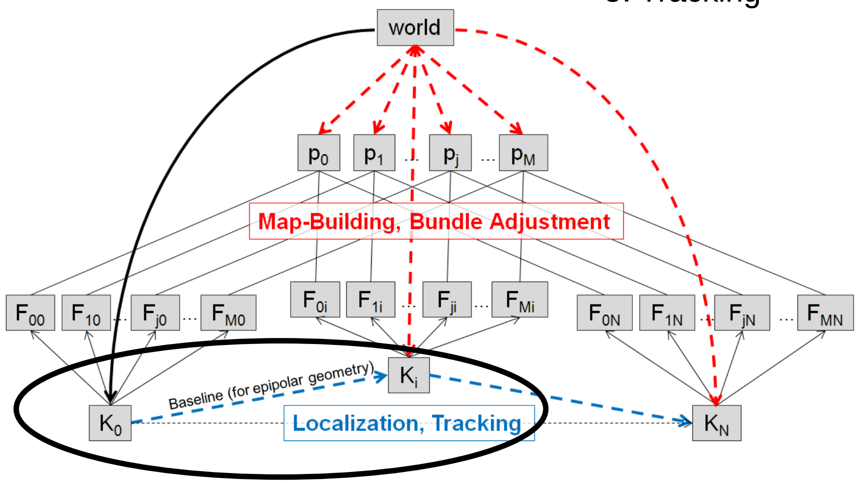
- Skip images from nearly the same viewpoint
  - Poor accuracy
  - Waste of space
  - Waste of time

Agenda

1. Motivation
2. Introduction to „Simultaneous Localization-And-Mapping“ (SLAM)
3. PTAM: System Overview
4. Feature Map
5. Tracking
6. Mapping
7. Results

5. Tracking

5.1 Overview



Assumptions

- Given: a set of 3D points, P_j, with associated 2D image features F_j in various keyframes of images {0..i-1} and across various levels l

Key steps of the algorithm

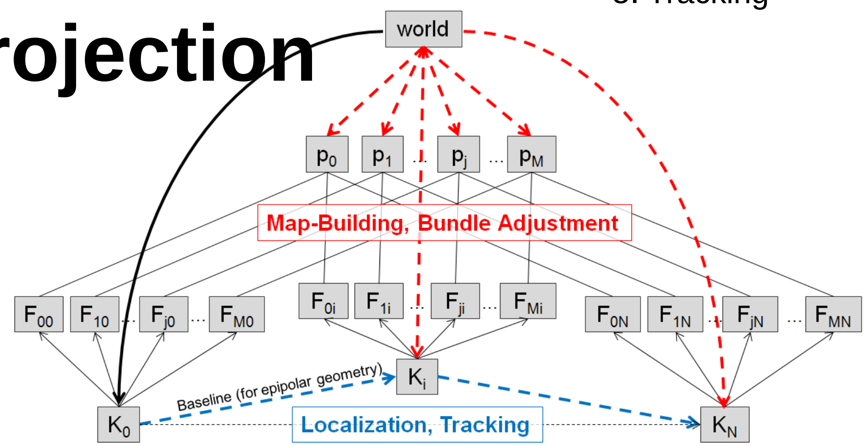
1. Acquire new frame I_i from a hand-held camera, generate a prior pose estimate K_i^-
2. Project map points into the image, according to K_i^-
3. Search for a small number (50) of the coarsest-scale features
4. Update camera pose from coarse matches
5. Reproject larger number (1000) of points into the image, search for their refined location
6. Compute final pose estimate K_i

5.2 Image Acquisition

- Frame grabbing
- Convert to black & white
- Generate image pyramid
- Determine interesting points (corners) on each pyramid level

- Compute prior estimate of camera pose $\mathrm{K}_{\mathrm{i}}^{-}$
  - Decaying velocity model
    (Lacking new measurements, the estimated camera slows down)

5.3 Camera Pose and Projection



- Transfer 3D points p_j from world coordinate system to estimated camera coordinate system K_i^- using transformation matrix $\mathbf{E}_{\mathbf{cw}}^-$

$$\mathbf{p}_{\mathbf{jc}} = \mathbf{E}_{\mathbf{cw}}^- \mathbf{p}_{\mathbf{jw}}$$

- Project points from 3D camera frame into image using given intrinsic parameters, focal length (f_u,f_v), principal point (u_0,v_0), and radial distortion r‘:

$$\begin{pmatrix} u_j \\ v_j \end{pmatrix} = CamProj(\mathbf{p}_{\mathbf{jc}})$$

$$CamProj(\mathbf{p}_{\mathbf{jc}}) = \begin{pmatrix} u_0 \\ v_0 \end{pmatrix} + \begin{bmatrix} f_u & 0 \\ 0 & f_v \end{bmatrix} \frac{r'}{r} \begin{pmatrix} x_{jc} / z_{jc} \\ y_{jc} / z_{jc} \end{pmatrix}$$

5.4 Patch Search

- Fixed-range image search around the predicted image location (circular region with fixed radius) of a point $\mathbf{p}_j$
  - Warp the 8x8 image patch according to the orientation of its normal $\mathbf{n}_j$ and the current viewpoint orientation (affine warp)
  - Determine appropriate pyramid level for search from the projected patch size
  - Generate an 8x8 search template from the source level, using the warp and bilinear interpolation
    - Subtract the average intensity to discount illumination changes
  - Calculate matching quality at all interesting points (corners) in the search area (quality criterion: SSD)
  - If best score is below threshold, a match has been found
- For high pyramid levels, refine matching position to sub-pixel precision (iterative error minimization) (computationally expensive)

5.5 Pose Update

- For each of S successful observations (found patches)
  - Patch location at $(\hat{u}, \hat{v})^T$ with measurement noise $\sigma^2 = 2^{2l}$
- Iterative minimzation of a robust objective function of the reprojection error vector $\mathbf{e_j}$ for a motion matrix **M**

$$\mathbf{e_j} = \begin{pmatrix} \hat{u}_j \\ \hat{v}_j \end{pmatrix} - CamProj(\mathbf{M}\mathbf{p_{jc}})$$

5.6 Two-Stage Coarse-to-Fine Tracking

Two consecutive phases of patch searching and pose update

- Phase 1:
  - Initial coarse search (50 map points) at highest pyramid level
    - Over a large search radius
    - With subpixel refinement
  - Pose update
- Phase 2:
  - Re-projection of up to 1000 of the remaining potentially visible image patches
    - Small search region
    - Refinements only for patches at highest pyramid level
  - Final frame pose from both coarse and fine image measurements

5.7 Tracking Quality, Failure Recovery

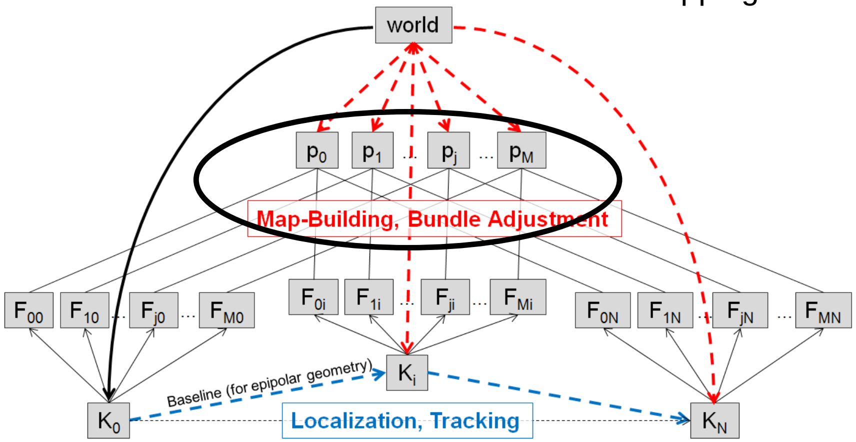
- Estimation of tracking quality as the fraction of successful feature observations
- If quality is below threshold, tracking continues, but no new keyframes are taken (in order not to degrade the 3D scene model)
- If quality is below a second threshold for more than a few frames, tracking is considered lost.
  → Tracking recovery is initiated.

Agenda

1. Motivation
2. Introduction to „Simultaneous Localization-And-Mapping“ (SLAM)
3. PTAM: System Overview
4. Feature Map
5. Tracking
6. Mapping
7. Results

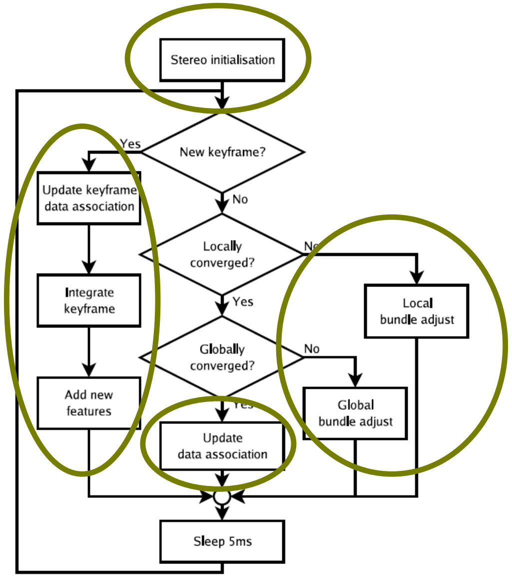
6. Mapping

6.1 Overview



Assumptions

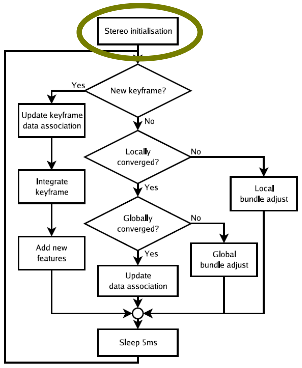
- Given: A set of previous keyframes $K_0..K_{i-1}$ with 2D image features F_j in various levels l, associated 3D points, P_j.



Key aspects of the algorithm:

1. Stereo Initialization
2. Keyframe insertion (and epipolar search)
3. Bundle adjustment
4. Data association refinement

6. Mapping

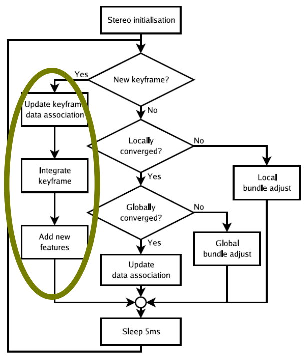


6.2 Stereo Initialization

At startup

- User-initiated initialization with two images (key frames)
  - User positions the camera above the work space and presses a key
    → First image (keyframe) is captured
    - 1000 image patches (interesting points) are automatically selected in the lowest pyramid level
  - User moves the camera by some amount (e.g., 10 cm) and presses a key
    → Second image (keyframe) is captured
    - During the camera motion, the features were already tracked
    - Second keyframe has patches related to first keyframe
- Relative pose estimation and 3D reconstruction using stereo computer vision
- Refinement through bundle adjustment
- Scaled to metric units by assuming that the camera moved by approx. 10 cm.

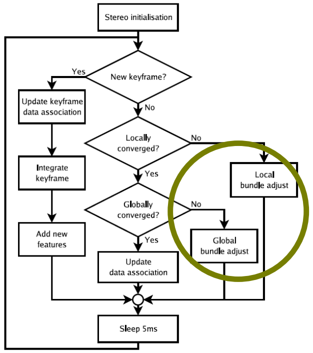
6. Mapping



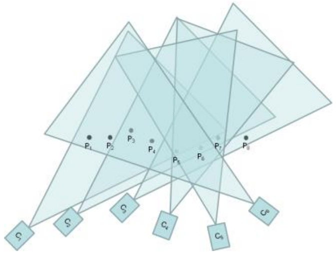
6.3 Keyframe Insertion

- New keyframes are added whenever:
  - Tracking quality is good
  - At least 20 frames since last keyframe have passed
  - New keyframe has a minimum distance to all previous keyframes (minimum distance threshold depends on mean depth of observed features)
- Existing feature points that may have been ignored by the tracking system are updated
- Interesting (salient) new features are computed in each pyramid level, filtered and added to the feature map.
- The depth of the new features is determined by mapping them to the closest already existing keyframe.

6.4 Bundle Adjustment

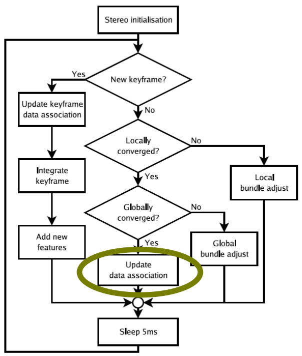


- Global bundle adjustment is very time-consuming; its time demand is dominated by the number of keyframes (currently not feasible in real-time for more than 50 keyframes).
- Local bundle adjustment:
  – Adjust only a small set X of keyframes (the current plus the four closest neighbors)
  – The current set of map points Z consists of all those points that have been detected in any of those five keyframes.
  – Extend the set of keyframes to the set Y of all frames in which the map points of Z have been seen (yet, the pose of these additional keyframes is not part of the adjustment process).



6. Mapping

6.5 Data Association Refinement



- „Luxury routine“: executed when there is spare time (on the non-tracking thread)
- Make new measurements in old keyframes:
  Measure newly created map features from new frames also in older frames
  - Measure newly created map features in older keyframes
  - Re-measure outlier measurements
- Issues addressed
  - Problems with repetitive features (patterns)

6.6 General Remarks

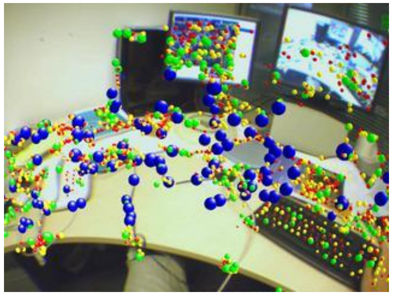
Some aspects of the current implementation could be improved

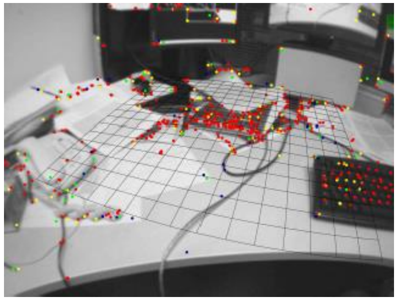
- Simple set of heuristics to remove outliers from the map
- Patches are initialized with a normal vector parallel to the viewing axis
- Two-stage approach can lead to increased tracking jitter

Agenda

1. Motivation
2. Introduction to „Simultaneous Localization-And-Mapping“ (SLAM)
3. PTAM: System Overview
4. Feature Map
5. Tracking
6. Mapping
7. Results

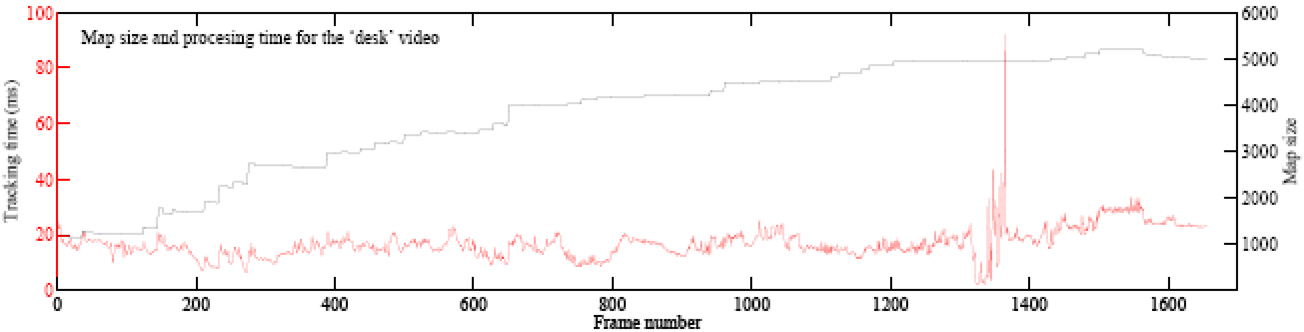
7. Results





Video of live demonstration

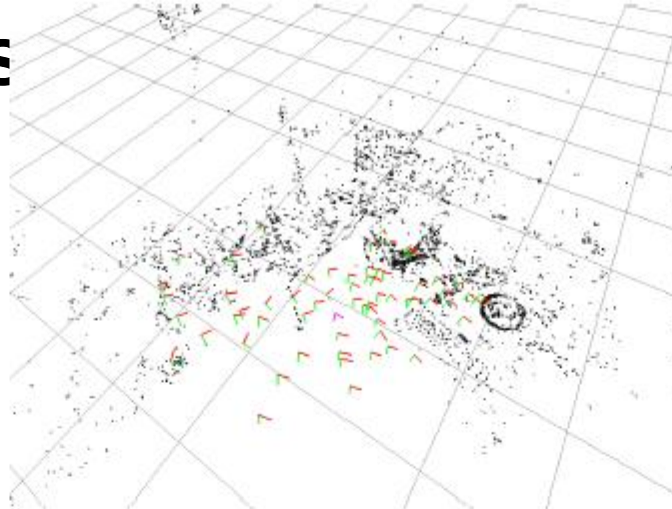
- Cluttered desk with immediate surroundings
- Over 1656 frames (55.2 seconds), 57 keyframes, 4997 point features
- Distribution across pyramid levels: L0: 51%, L1: 33%, L2: 9%, L3: 7%

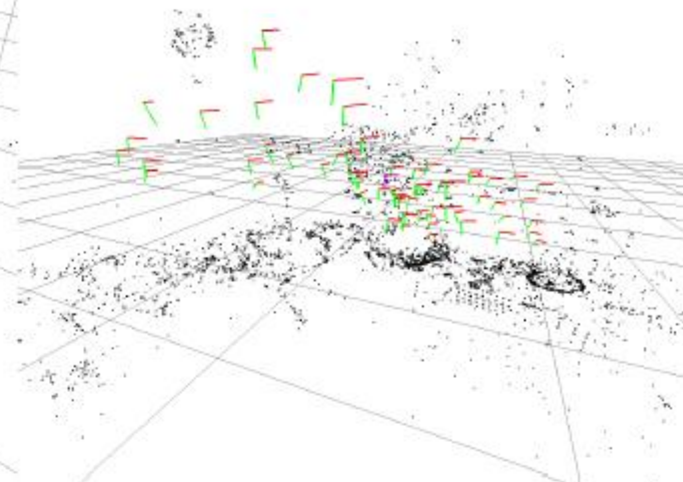


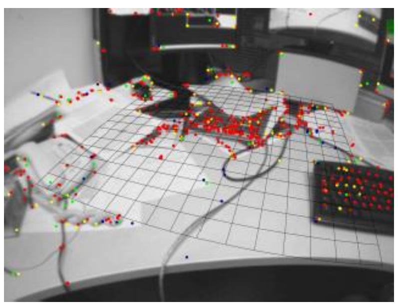
Demos:

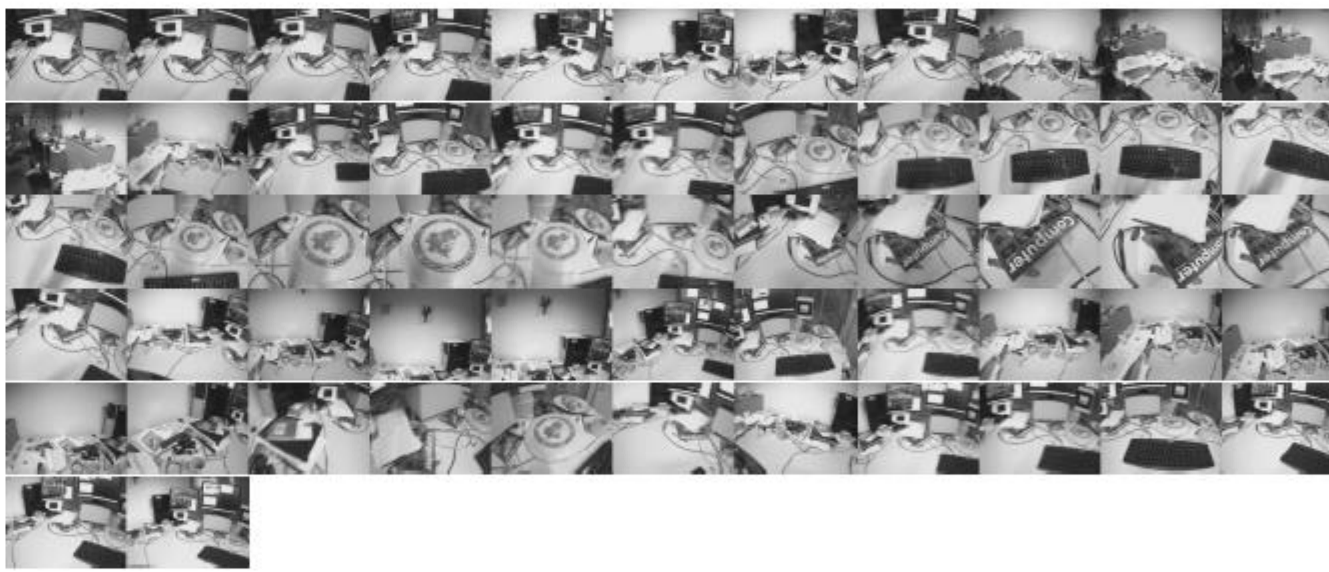
- Scale change
- Lens simulation: camera as input device, camera-centered lens to scorch the environment
- Ewok rampage: camera used to aim Darth Vader‘s laser pistol. Movement is controlled with the keyboard.

7. Results









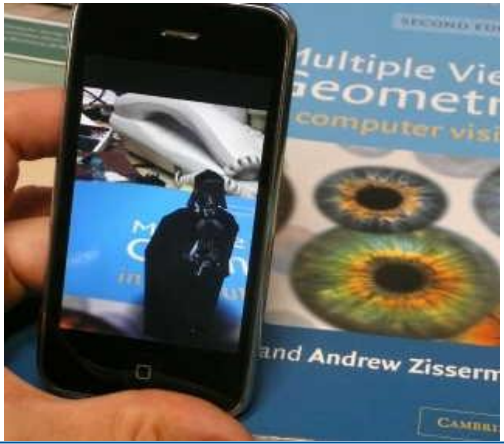
7. Results

More demos:

- PTAM at ISMAR 2007 in Nara, Japan

- Tracking Contest 2008

- PTAM on iPhone, ISMAR 2009
http://www.youtube.com/watch?v=pBI5HwitBX4



Thank you!