MonoSLAM

- This talk will present the work by Davison et. al. published in PAMI 2007. The following work and images are al obtained from the published PAMI paper [1]

What is SLAM?

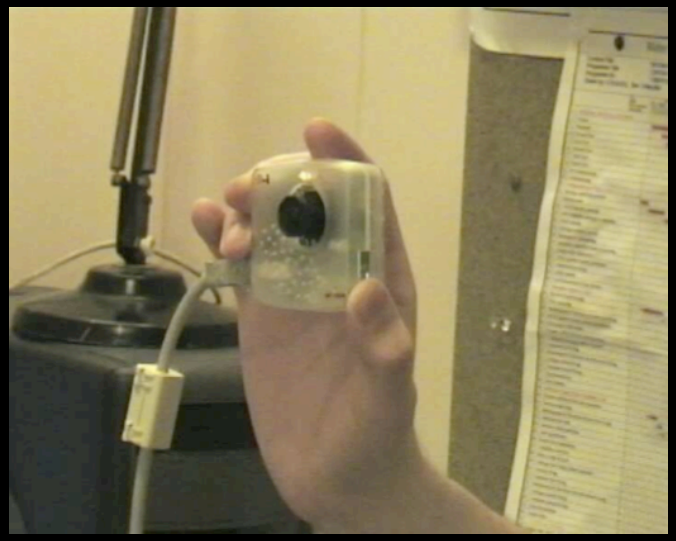
- Simultaneous Location And Mapping
  - build map within unknown environment
  - keep track of current position
  - geometrically accurate maps
  - sensors: laser range finders, sonar, camera, etc...

Uses

- UAV
- AUV
- planetary rovers
- domestic robots

Introduction

- single lens camera
- recover 3D trajectory in real time
- first to use single uncontrolled camera



- main idea: on-line creation of sparse, persistent set of landmarks in probabilistic framework

Contributions

- active approach to mapping and measurement
- use of general camera motion model
- monocular feature initialization
- feature orientation estimation

Past Research

- camera images too difficult to turn into maps in real time
- concentrate on SFM
  - analyze image sequence
  - reconstruct scene structure and camera trajectory
  - offline

Related Work

- Harris and Pike
  - visual maps from single cam
  - showed 3D maps of features
  - real-time
  - limitation: ignored correlation between mapped visual features

SLAM

- Localization and mapping are coupled
- MonoSLAM: concentrate on localization
  - high frame rate required
  - where it is
  - sparse map

MonoSLAM

- Probabilistic 3D map
- Visual landmarks
- System initialization
- Motion modeling & prediction
- Active feature measurement and map update
- Feature initialization
- Map Management
- Feature orientation estimation

Probabilistic 3D Map

- feature-based
- represents state of the camera and points of interest at a given time
- updated by Extended Kalman Filter
- map can be enlarged or reduces
- represented by state vector and covariance matrix
- role: provide real-time localization

Probabilistic 3D Map

- camera state vector
  - r^W - 3D position
  - q^{RW} - orientation
  - v^W - velocity
  - w^R - angular velocity

$$\mathbf{x}_v = \begin{pmatrix} \mathbf{r}^W \\ \mathbf{q}^{WR} \\ \mathbf{v}^W \\ \omega^R \end{pmatrix}$$

Probabilistic 3D Map

- state vector
- covariance matrix

$$\hat{\mathbf{x}} = \begin{pmatrix} \hat{\mathbf{x}}_v \\ \hat{\mathbf{y}}_1 \\ \hat{\mathbf{y}}_2 \\ \vdots \end{pmatrix}, \quad \mathrm{P} = \begin{bmatrix} \mathrm{P}_{xx} & \mathrm{P}_{xy_1} & \mathrm{P}_{xy_2} & \cdots \\ \mathrm{P}_{y_1x} & \mathrm{P}_{y_1y_1} & \mathrm{P}_{y_1y_2} & \cdots \\ \mathrm{P}_{y_2x} & \mathrm{P}_{y_2y_1} & \mathrm{P}_{y_2y_2} & \cdots \\ \vdots & \vdots & \vdots & \end{bmatrix}$$

Probabilistic 3D Map

- map provides localization, not scene description
- sparse set of landmarks
- assume rigid scene
- landmark are stationary world features
- each landmark assumed to correspond to point feature in 3D space
- camera modeled as rigid body with (R, t) parameters
- maintain linear and angular velocity estimates

Probabilistic 3D Map

- Map has order $O(N^2)$
- Algorithm complexity is also $O(N^2)$
  - limits number of features (N) for real time processing
  - experiments: N <= 100
  - enough to span a room

Probabilistic 3D Map

- method uses full covariance EKF vs probabilistic approaches
- feature correlation in off-diagonal entries
- repeatable localization vs exploring new environment
- free-moving camera - various subsets of features visible
- can only be represented by sparse map (state vector, covariance matrix) for complexity reasons

Visual Landmarks

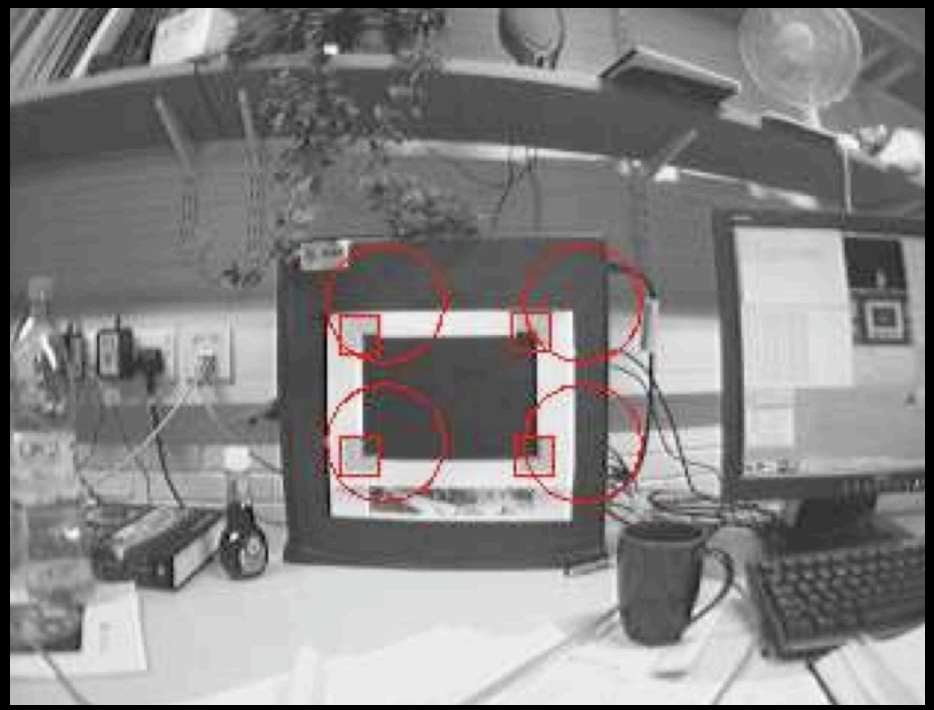
- Features originally detected via Shi and Tomasi's detection operator
- Large image patches (11 x 11)
- Goal: identify same landmarks under any camera motion

Visual Landmarks

- Assume each landmark lies on a planar surface
- Set surface normal parallel to optical axis
- Each feature stored as oriented planar texture (3D location, including depth)
- Matching from new camera position
  - project patch from 3D to image plane (warped)
  - projective warp, shearing/perspective distortion

System Initialization

- Startup aided using some prior info
- Known target is placed in front of camera
- Provides several features with known position/ appearance



System Initialization

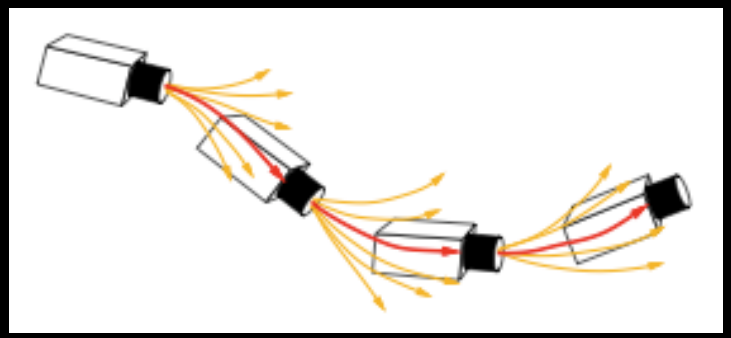
- Reasons for init target:
  - no way of measuring position estimation
  - having features allows normal cycle to begin (predict - measure - update)
- Init features inserted into map
- Cam held at set distance from target
- Target features define world coordinate frame for SLAM

Motion Modeling

- State vector updated in 2 ways:
  - prediction - camera moves in blind interval between image capture
  - update - after making measurements on visual features

Motion Modeling

- Camera motion model
  - ‘smooth’ motion
  - at each position, predict most likely path and alternatives with small deviations



Motion Modeling

- ‘constant velocity, constant angular velocity’ model
  - in a time step, expect undetermined accelerations with a Gaussian profile
  - imposing smoothness on expected motion
  - very large accelerations unlikely
  - gives entire system robustness (sparse features)

Motion Modeling

- Assume at each time step, unknown acceleration a^W and angular acceleration α^R occur:

$$\mathbf{n} = \begin{pmatrix} \mathbf{V}^W \\ \Omega^R \end{pmatrix} = \begin{pmatrix} \mathbf{a}^W \Delta t \\ \alpha^R \Delta t \end{pmatrix}$$

- State vector is updated to be:

$$\mathbf{f}_v = \begin{pmatrix} \mathbf{r}^W_{new} \\ \mathbf{q}^{WR}_{new} \\ \mathbf{v}^W_{new} \\ \omega^R_{new} \end{pmatrix} = \begin{pmatrix} \mathbf{r}^W + (\mathbf{v}^W + \mathbf{V}^W)\Delta t \\ \mathbf{q}^{WR} \times \mathbf{q}((\omega^R + \Omega^R)\Delta t) \\ \mathbf{v}^W + \mathbf{V}^W \\ \omega^R + \Omega^R \end{pmatrix}$$

Motion Modeling

- EKF requires increase in state uncertainty Q_v for camera after its motion
- Q_v calculated from Jacobian as:

$$\mathtt{Q}_v = \frac{\partial \mathbf{f}_v}{\partial \mathbf{n}} \mathtt{P}_n \frac{\partial \mathbf{f}_v}{\partial \mathbf{n}}^\top$$

where P_n is the covariance of noise vector **n**

Motion Modeling

- Pn determines rate of growth of uncertainty in motion
- Small P_n
  - expect smooth motion with small accelerations
  - cannot handle sudden movements
- Large P_n
  - uncertainty increases, many good measurements must be made
  - can handle fast accelerations

Active Features

- Idea: predict image position of each feature before measuring
- Given camera & feature position, feature position relative to the camera is:

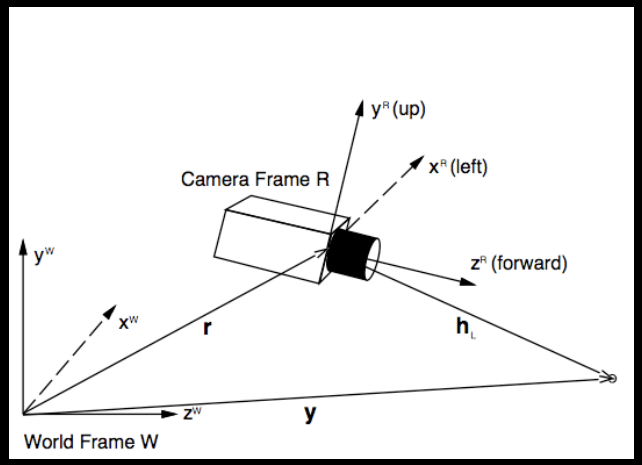
$$\mathbf{h}_L^R = \mathtt{R}^{RW}(\mathbf{y}_i^W - \mathbf{r}^W)$$

- Feature position (perspective cam) on the image is given by:

$$\mathbf{h}_i = \begin{pmatrix} u \\ v \end{pmatrix} = \begin{pmatrix} u_0 - f k_u \frac{h_{Lx}^R}{h_{Lz}^R} \\ v_0 - f k_v \frac{h_{Ly}^R}{h_{Lz}^R} \end{pmatrix}$$

Active Features

- Wide-angle vs high resolution cameras
- 100° FOV
- Camera/map estimates are better constrained when simultaneous features are observed from different viewing angles



Active Features

- Warp perspective coordinates with radial distortion to get final predicted image position (u_d, v_d)

$$
\begin{aligned}
u_d - u_0 &= \frac{u - u_0}{\sqrt{1 + 2K_1 r^2}} \\
v_d - v_0 &= \frac{v - v_0}{\sqrt{1 + 2K_1 r^2}}
\end{aligned}
$$

$$
r = \sqrt{(u - u_0)^2 + (v - v_0)^2}
$$

Active Features

- Jacobians of projection (S_i) are computed
- Allow calculation of uncertainty in feature location prediction

$$\mathrm{S}_i = \frac{\partial \mathbf{u}_{di}}{\partial \mathbf{x}_v} \mathrm{P}_{xx} \frac{\partial \mathbf{u}_{di}}{\partial \mathbf{x}_v}^\top + \frac{\partial \mathbf{u}_{di}}{\partial \mathbf{x}_v} \mathrm{P}_{xy_i} \frac{\partial \mathbf{u}_{di}}{\partial \mathbf{y}_i}^\top + \frac{\partial \mathbf{u}_{di}}{\partial \mathbf{y}_i} \mathrm{P}_{y_i x} \frac{\partial \mathbf{u}_{di}}{\partial \mathbf{x}_v}^\top + \frac{\partial \mathbf{u}_{di}}{\partial \mathbf{y}_i} \mathrm{P}_{y_i y_i} \frac{\partial \mathbf{u}_{di}}{\partial \mathbf{y}_i}^\top + \mathrm{R}$$

- "Constant noise covariance R of measurements is taken to be diagonal with magnitude determined by image resolution"

Active Features

- Si represents the shape of a 2D Gaussian PDF over image coordinates
- Defines an elliptical search window of high probability
- Maximize efficiency and minimize mismatches
- Si also measures information content
  - features with high Si provide more information for camera/feature position

Feature Initialization

- Monocular cameras cannot give 3D location of new features - no depth information
- Tracking is no good - fast motion
  - estimate depth from multiple view triangulation

Feature Initialization

- Insert 3D line from cam position to INF, passing through feature viewing direction
- Line has Gaussian uncertainty in its parameters
- Represented as:

$$\mathbf{y}_{pi} = \begin{pmatrix} \mathbf{r}_i^W \\ \hat{\mathbf{h}}_i^W \end{pmatrix}$$

- Line has all possible locations with 1 degree of freedom - depth

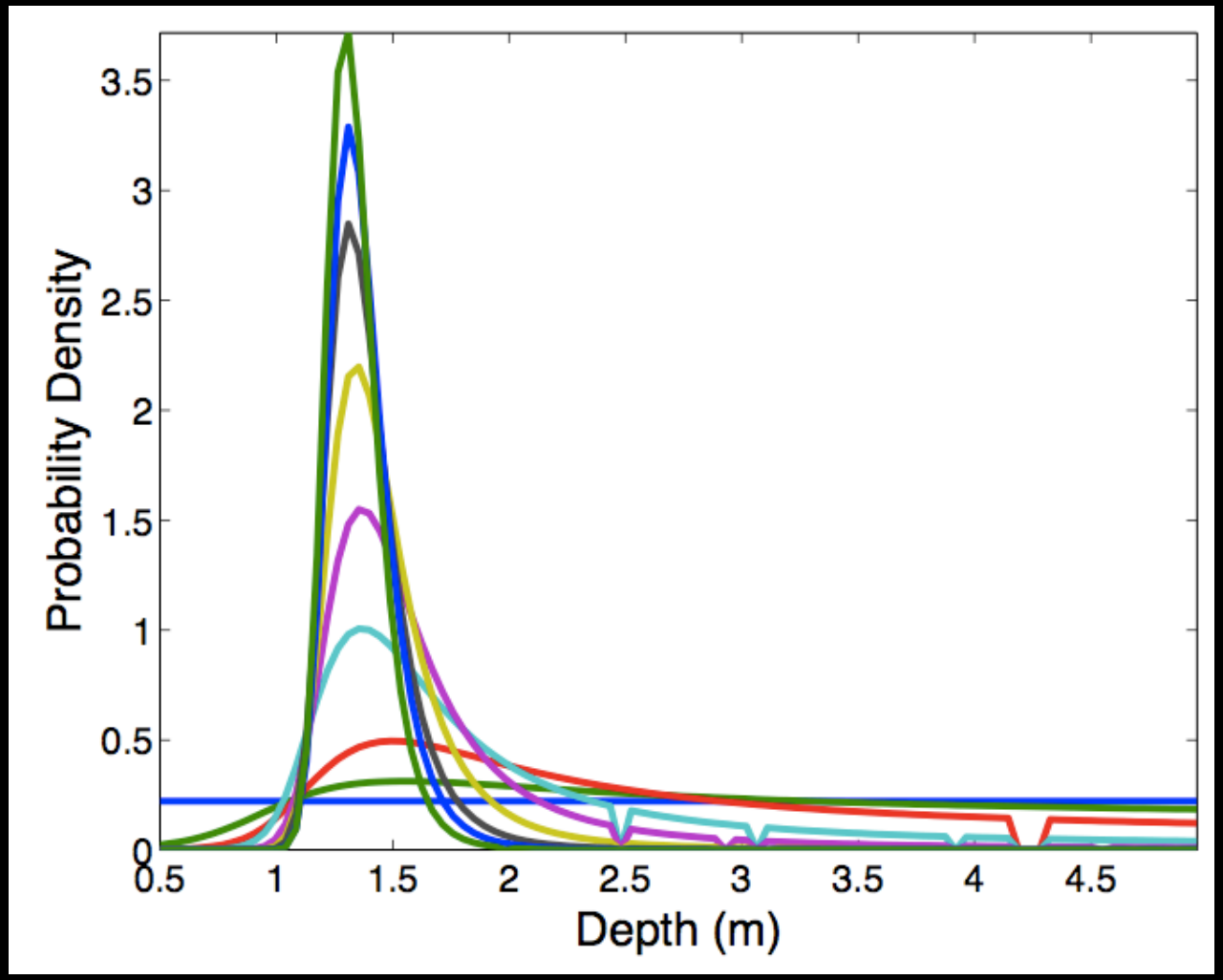
Feature Initialization

- Discrete set of depth hypotheses are uniformly distributed along line
- Next measurements only provide depth
- Each hypothesis is tested at each time step by projecting them into the image
  - generates elliptical search region
  - each hypothesis at depth λ has 3D coordinate $\mathbf{y}_{\lambda i} = \mathbf{r}_i^W + \lambda \hat{\mathbf{h}}_i^W$

Feature Initialization

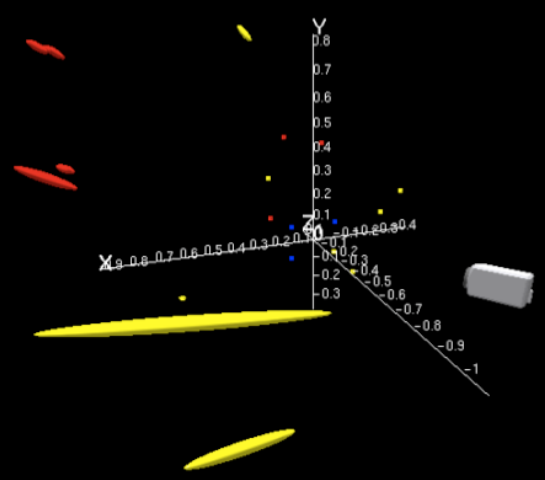
- Make correlation searches for same feature template over set of ellipses (overlap)
- Feature matching produces a likelihood for each, reweighted using Baye's rule:
  - likelihood score is probability indicated by 2D Gaussian PDF in image space
- Small ellipses with small overlap have stronger resolving power (faster)

Feature Initialization



Feature Initialization

- At each step, weakest hypothesis is removed for efficiency
- Method would perform poorly if no parallax is measured (optical axis)



Map Management

- When to add/remove features
- 12 good features provide enough data for SLAM without overburdening CPU
- Add - total features less than threshold
- Shi, Tomasi detector
  - random location (80x60)
  - no overlap
  - not expected to disappear

Map Management

- Delete feature - matching attempts fail when feature should be visible
  - more than 50% failure
  - removes “bad” features
    - occluded
    - on moving objects
    - specular highlights

Map Management

- Natural selection - good feature set
- Problems - mismatch due to similar appearance
  - catastrophic
  - unlikely
    - large templates
      - uniqueness

Feature Orientation

- Use visual measurements to improve orientation for planar surface features
  - obtain surface normals
  - better feature location
  - future goal of building detailed 3D map

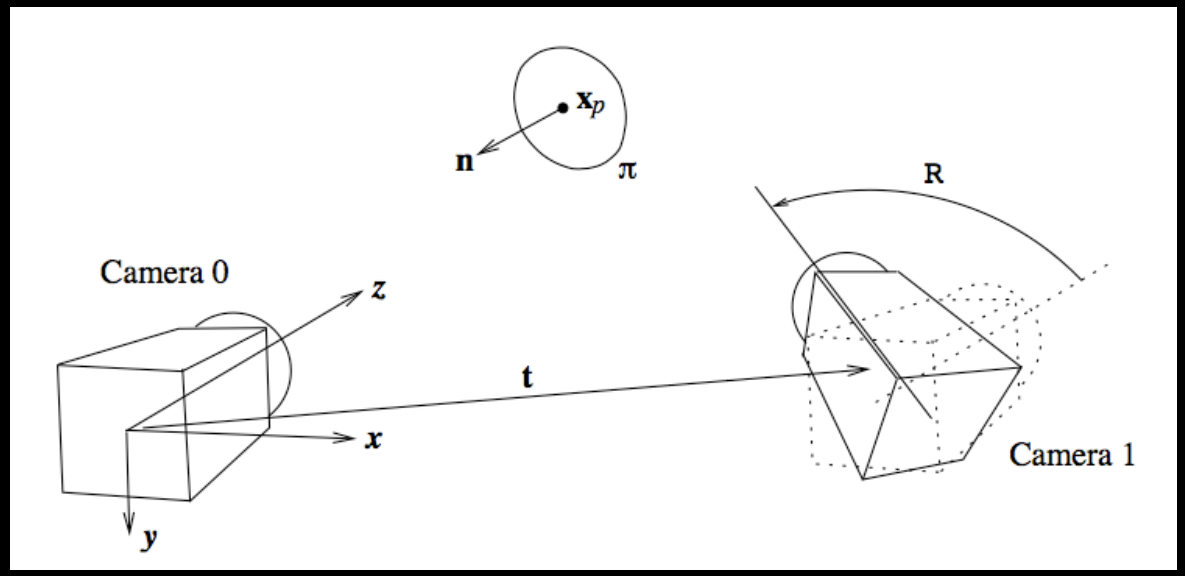
Feature Orientation

- Assume local planar region
- Camera motion - warp initial template
  - depends on
    - initial/current cam position
    - feature 3D center
    - surface orientation
  - allows prediction of warp

Feature Orientation

- Difference between prediction and measurement used to update orientation
- Warp is described by the homography:

$$\mathtt{H} = \mathtt{CR}[\mathbf{n}^T \mathbf{x}_p \mathtt{I} - \mathbf{t}\mathbf{n}^T]\mathtt{C}^{-1}$$



Feature Orientation

- Measuring warp between template and current
  - use gradient descent
  - search through additional parameters
  - efficient - not all degrees of freedom
  - assume change is small
  - find global best fit

Feature Orientation

- Only works with large patches
  - texture
  - low texture - weak estimation
- Motivation - improve range of camera motion
  - long-term landmarks
  - features with weak texture are viewpoint invariant - matching will still work

Method

- Initialize with known target
- Insert into probability map
- Move camera
- Predict location of features
- Measure difference between location and prediction
- Update map parameters
- Update features in map ading/removing features accordingly

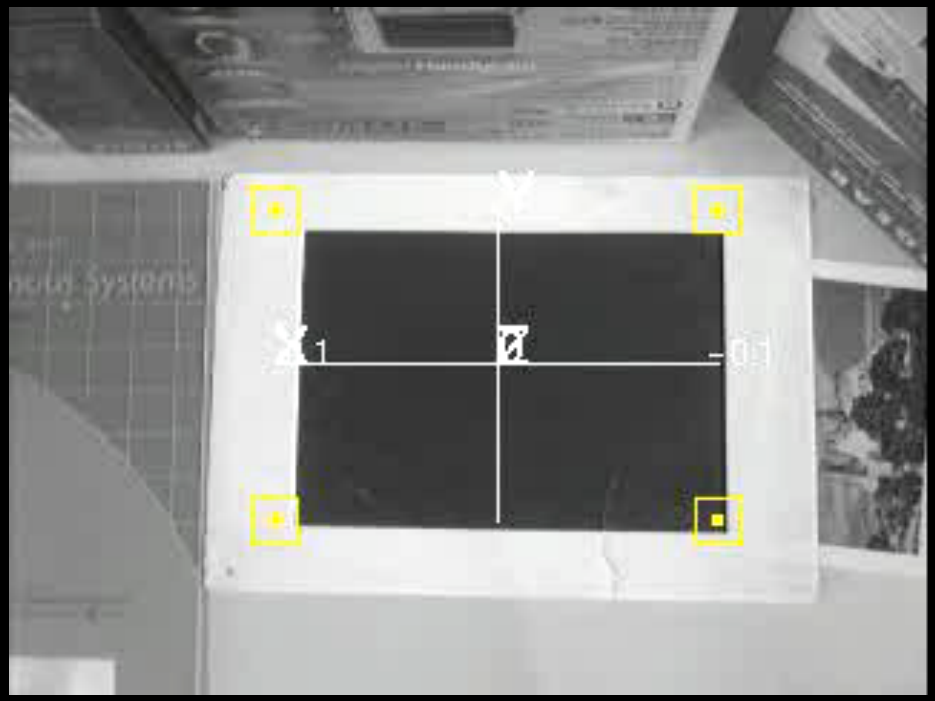
Method

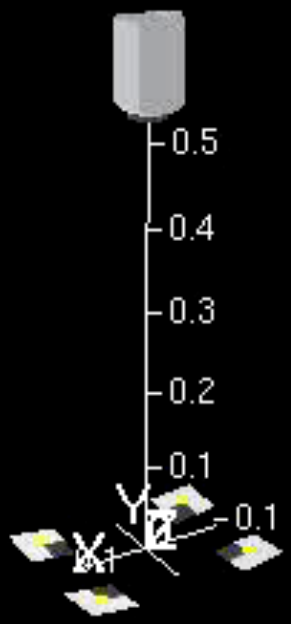
- Pentium M processor, 1.6GHz

Image loading and administration	2ms
Image correlation searches	3ms
Kalman Filter update	5ms
Feature initialization search	4ms
Graphical rendering	5ms
Total	19ms

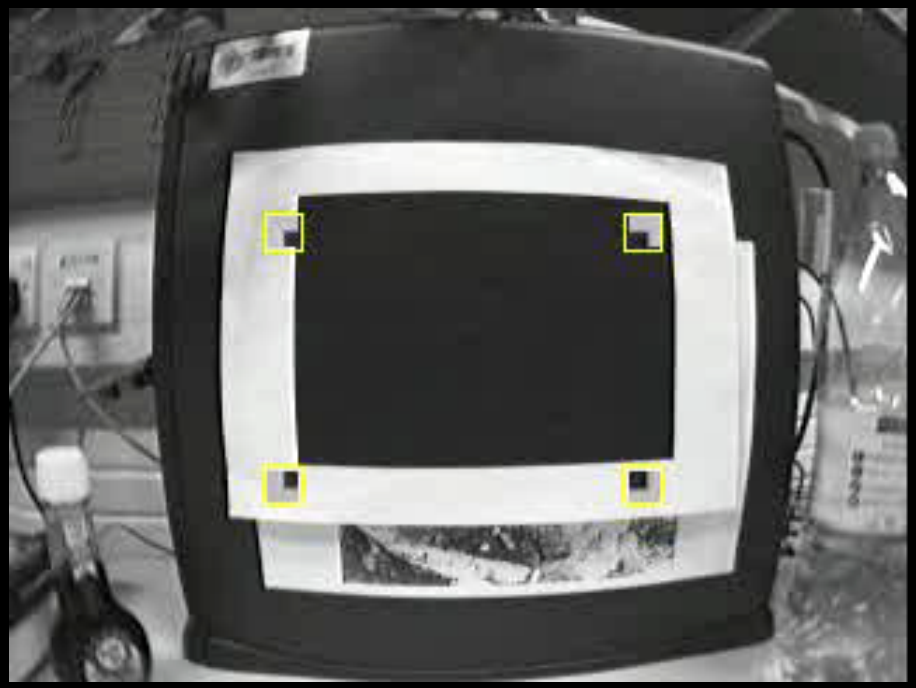
- 30 Hz performance achieved

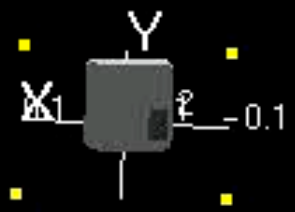
Example



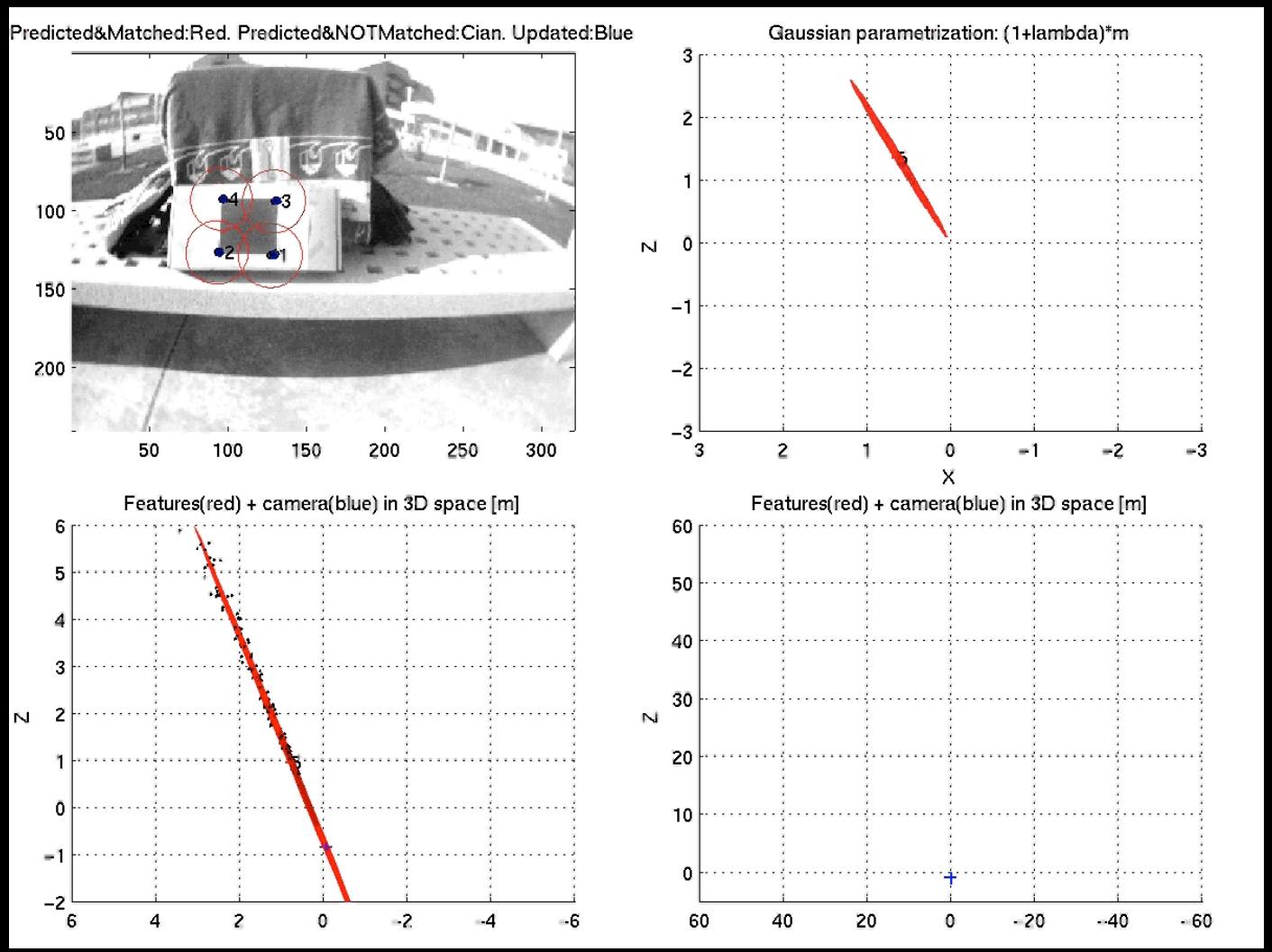


Example

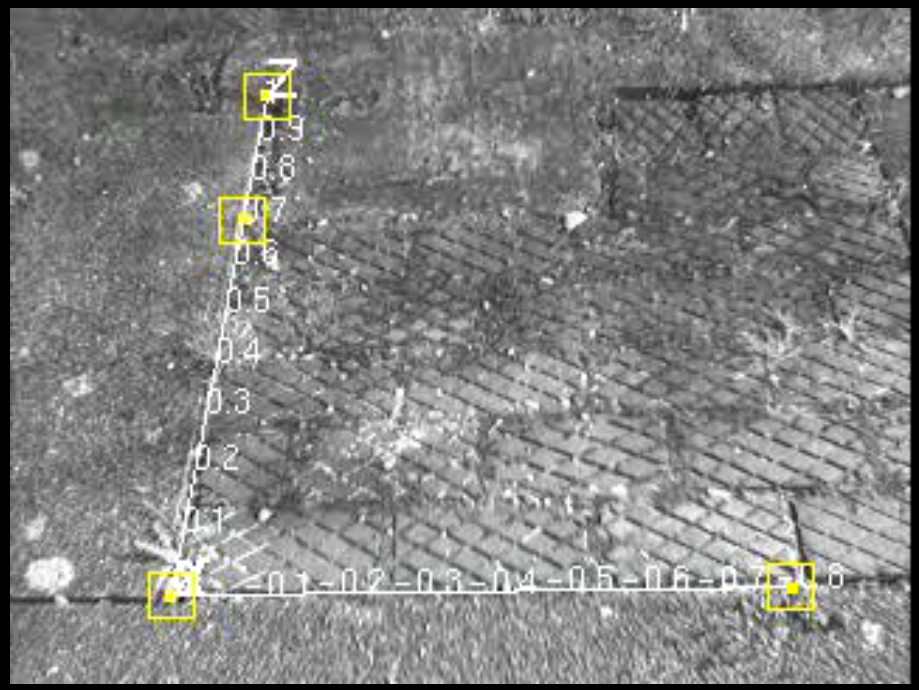


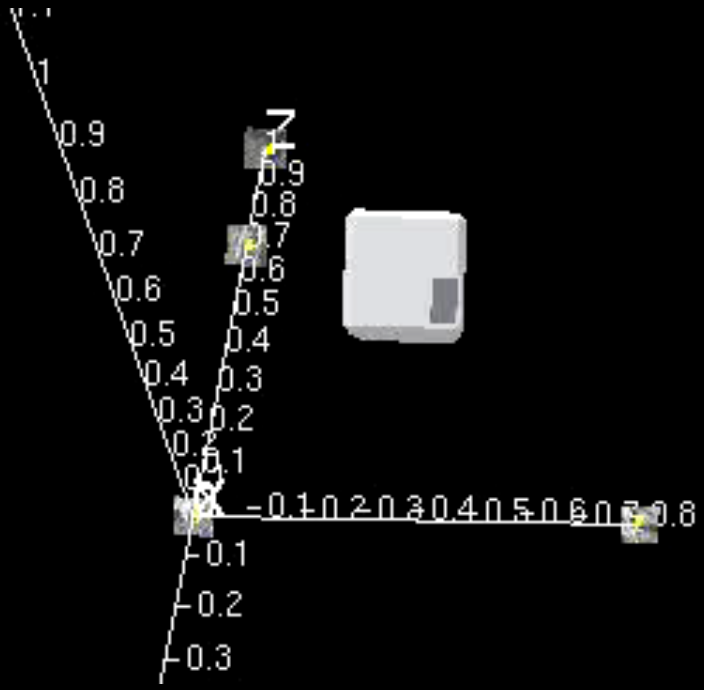


Example


Predicted&Matched:Red. Predicted&NOTMatched:Cian. Updated:Blue
Gaussian parametrization: (1+lambda)*m
Features(red) + camera(blue) in 3D space [m]
Features(red) + camera(blue) in 3D space [m]
X
Z

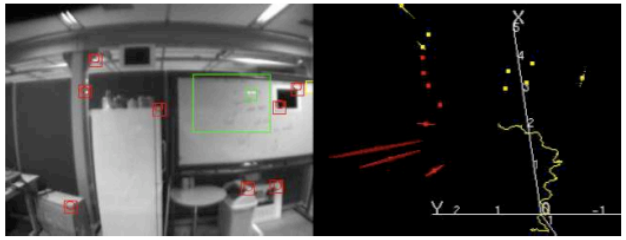
Example



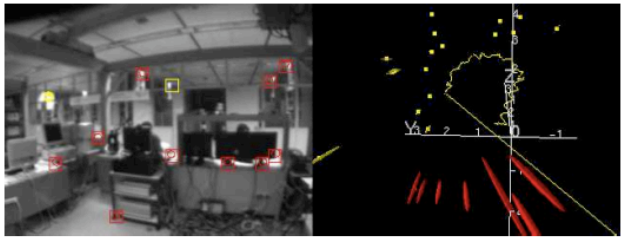


Example

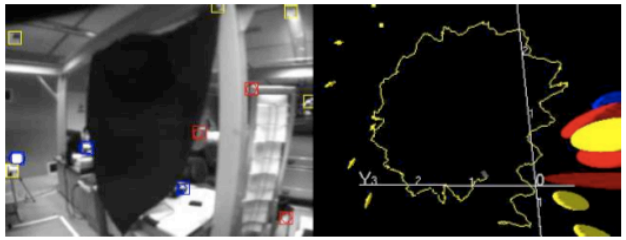
Example



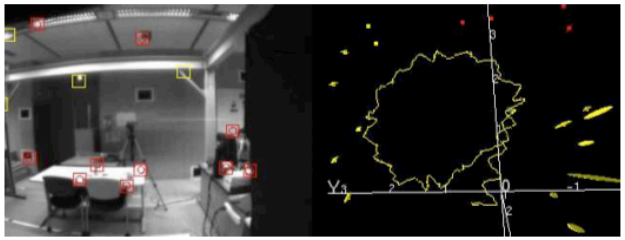
1. Early exploration and first turn



2. Mapping back wall and greater uncertainty

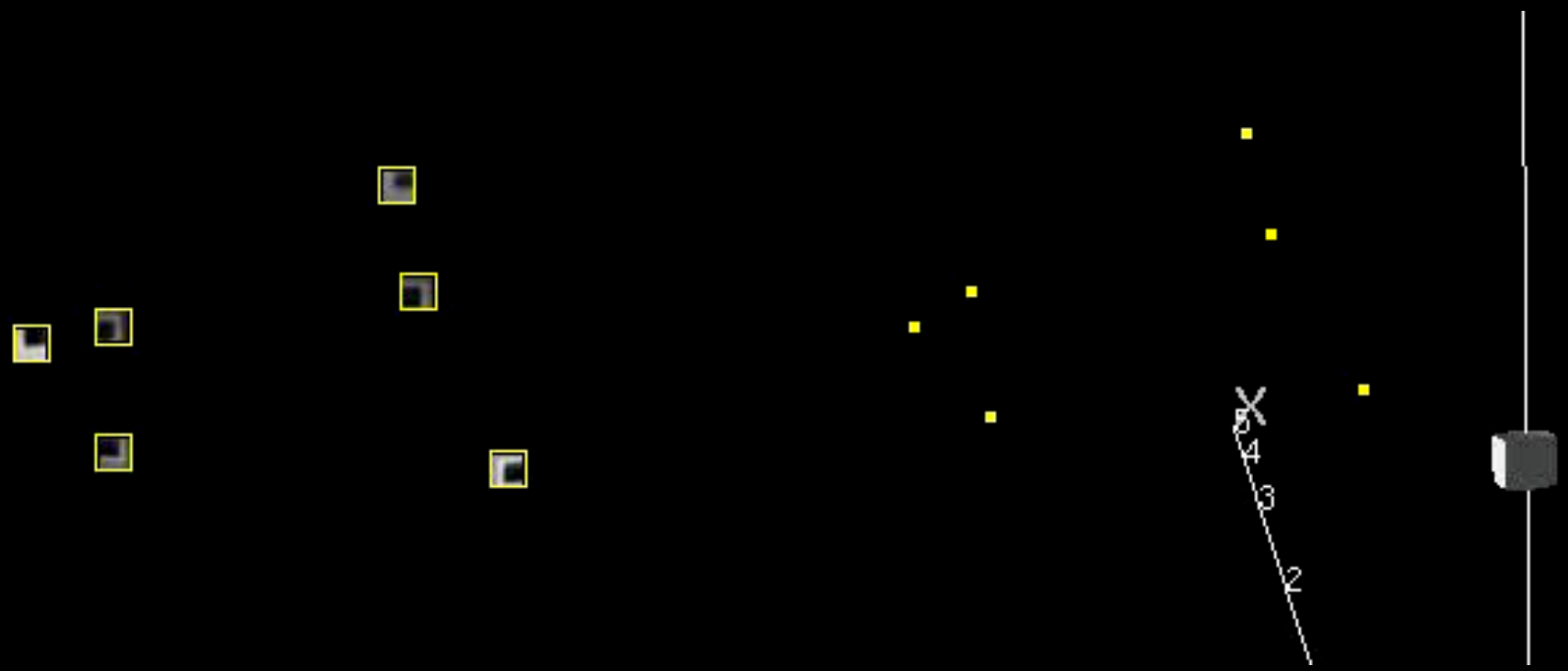


3. Just before loop close; maximum uncertainty



4. End of circle with closed loop and drift corrected

Example



Conclusion

- real-time SLAM
- single monocular camera
- low cost
- MonoSlam
  - probabilistic mapping
  - motion modeling
  - active measurement

References

1. Davidson, Andrew, Ian Reid, Nicholas Molton, Olivier Stasse. "MonoSLAM: Real Time Single Camera SLAM". PAMI, 2007.
2. Soren Riisgaard and Morten Rufus Blas. "SLAM for Dummies". http://ocw.mit.edu/NR/rdonlyres/Aeronautics-and-Astronautics/16-412JSpring-2005/9D8DB59F-24EC-4B75-BA7A-F0916BAB2440/0/1aslam_blas_repo.pdf