Real Time Simultaneous Localization And Mapping with Single Camera (Mono-SLAM)

Andrew J. Davidson

Vibot

uB
UNIVERSITÉ DE BOURGOGNE

Master in Computer Vision

By:

Deogratias
Vamshi

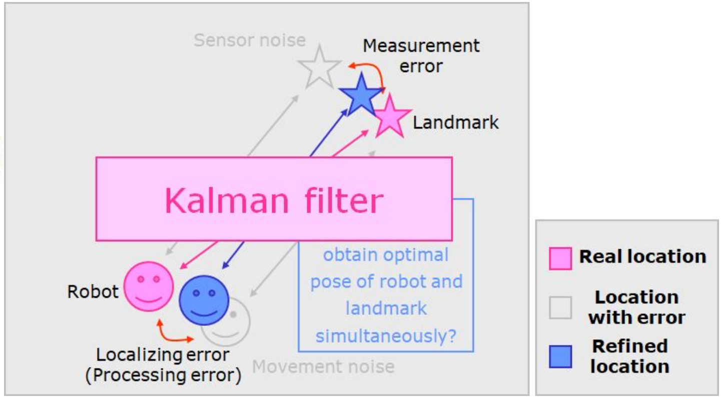
Contents

- Basics - Structure from motion
    - What is MonoSLAM?
    - Overview
- Details of implementation
    - Initialization
    - Prediction
    - Tracking
    - Correction
- Results
    - Raw data
    - Video annotation explanation
    - Video results and Limitations
- Conclusion

Kalman filter

- What is a Kalman filter?
  - Mathematical power tool
  - Optimal recursive data processing algorithm
    - Noise effect minimization

- Applications
  - Tracking (head, hands etc.)
  - Lip motion from video sequences of speakers
  - Fitting spline
  - Navigation
  - Lot’s of computer vision problem



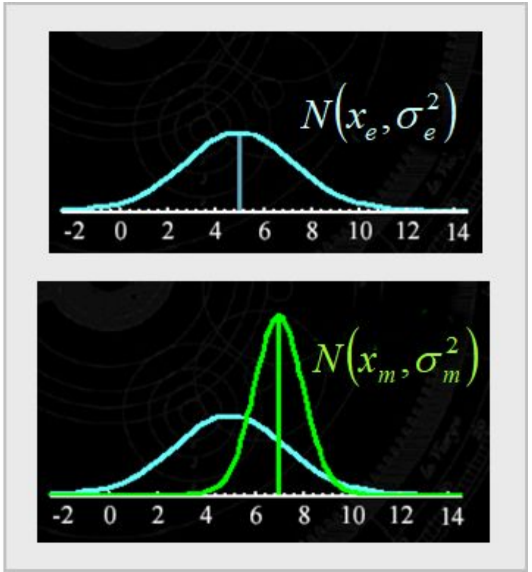
Kalman filter

- Example (Simple Gaussian form)
  - Assumption
    - All error form Gaussian noise
  - Estimated value

$$x_e, \sigma_e^2$$

  - Measurement value

$$x_m, \sigma_m^2$$



Kalman filter

- Example (Simple Gaussian form)
  - Optimal variance

$$\boxed{\frac{1}{\sigma^2} = \frac{1}{\sigma_e^2} + \frac{1}{\sigma_m^2}}$$

  - Optimal value

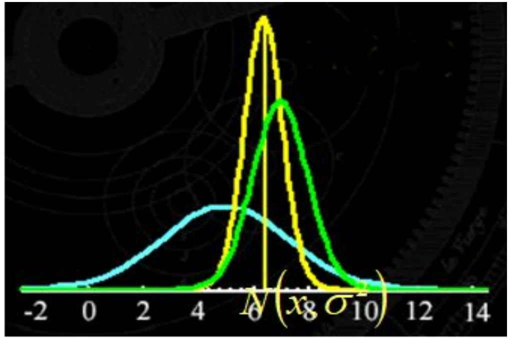
$$x = \left[\frac{\sigma_m^2}{\sigma_e^2 + \sigma_m^2}\right] x_e + \left[\frac{\sigma_e^2}{\sigma_e^2 + \sigma_m^2}\right] x_m$$

$$x = x_e + \boxed{\left[\frac{\sigma_e^2}{\sigma_e^2 + \sigma_m^2}\right]}\boxed{(x_m - x_e)}$$

$$\boxed{x = x_e + K\,i}$$

Innovation

Kalman gain



SLAM

Simultaneously Localization and map building system

EKF(Extended Kalman filter)-based framework

If we have the solution to the SLAM problem…

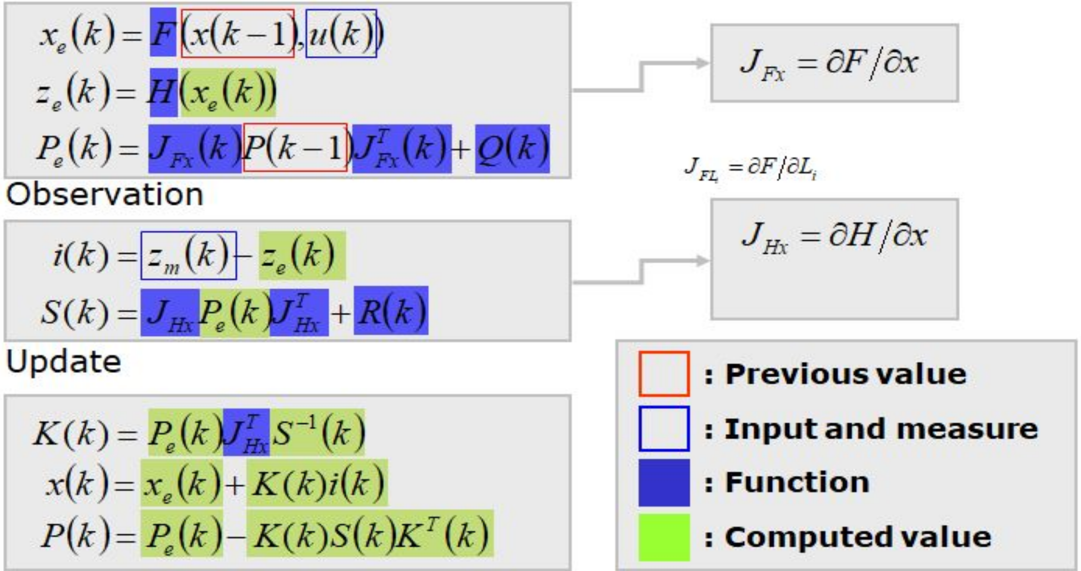
1. Allow robots to operate in an environment without a priori knowledge of a map

2. Open up a vast range of potential application for autonomous vehicles & robot

3. Research over the last decade has shown that SLAM is indeed possible

SLAM

- Kalman filter and SLAM problem
  - Extended Kalman filter form for SLAM



Prediction

$$x_e(k) = F(x(k-1), u(k))$$

$$z_e(k) = H(x_e(k))$$

$$P_e(k) = J_{Fx}(k) P(k-1) J_{Fx}^T(k) + Q(k)$$

$$J_{Fx} = \partial F / \partial x$$

$$J_{FL_i} = \partial F / \partial L_i$$

Observation

$$i(k) = z_m(k) - z_e(k)$$

$$S(k) = J_{Hx} P_e(k) J_{Hx}^T + R(k)$$

$$J_{Hx} = \partial H / \partial x$$

Update

$$K(k) = P_e(k) J_{Hx}^T S^{-1}(k)$$

$$x(k) = x_e(k) + K(k) i(k)$$

$$P(k) = P_e(k) - K(k) S(k) K^T(k)$$

□ : **Previous value**

□ : **Input and measure**

■ : **Function**

🟩 : **Computed value**

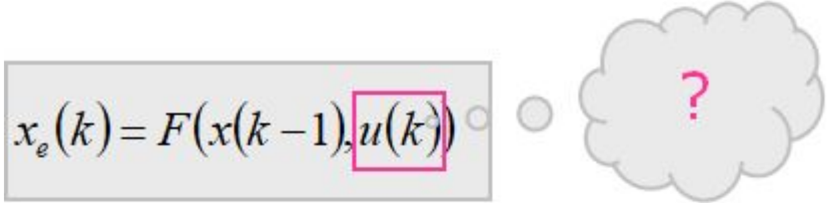
Mono-SLAM

What is Mono SLAM?

EKF-SLAM framework (EKF : Extended Kalman Filter)

Single camera

Unknown user input

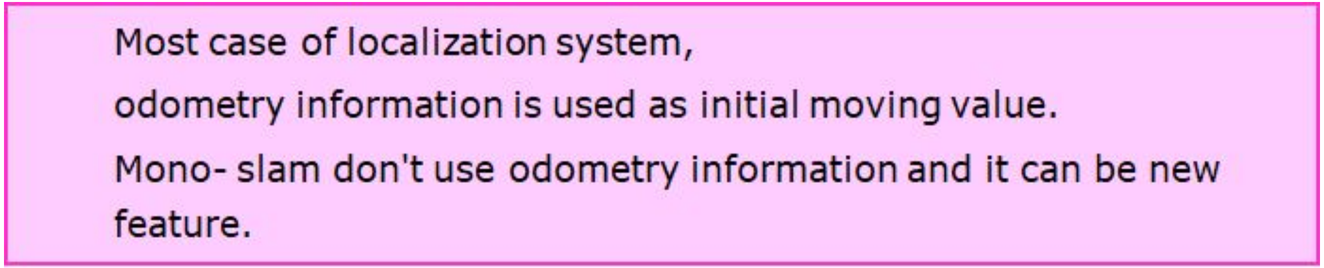


$$x_e(k) = F(x(k-1), \boxed{u(k)})$$

User input

Known control input

Encoder information of robot or vehicle (odometry)



Most case of localization system,

odometry information is used as initial moving value.

Mono- slam don't use odometry information and it can be new feature.

Overview

To get an idea on how MonoSLAM is working, we can roughly introduce it through a few remarks

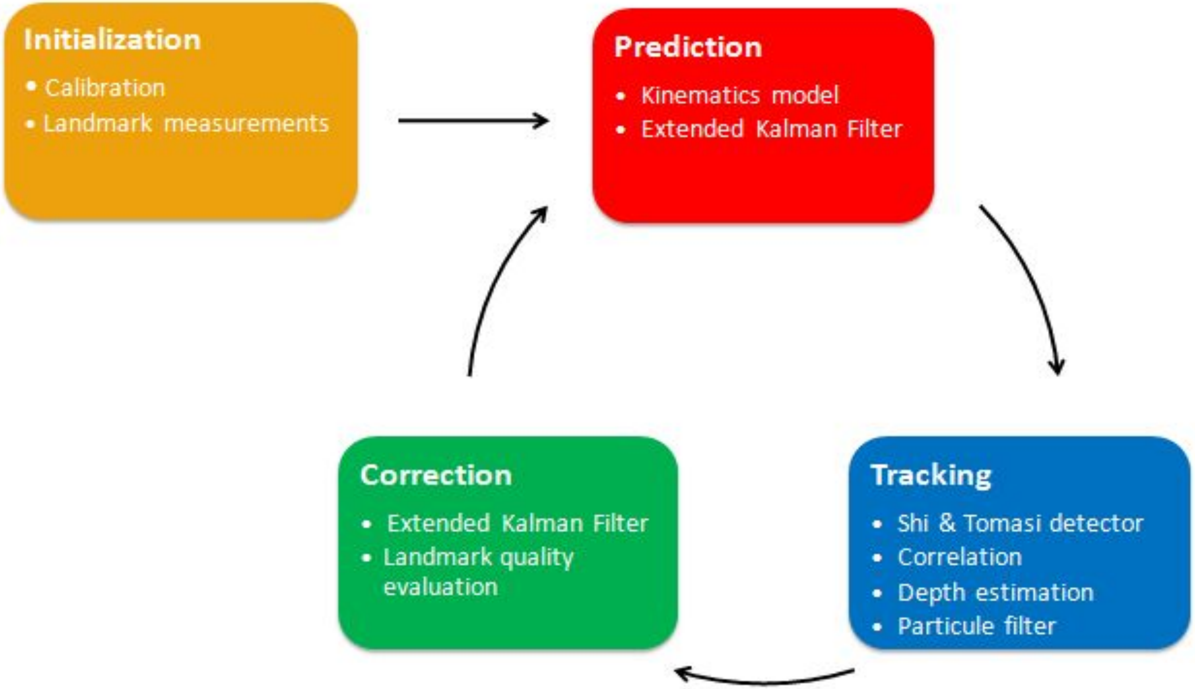
- SLAM in a 3D environment → Need of 3D landmarks;
- Camera-based method → Use of visual 2D interest points;
- Camera gives 2D interest points → No depth information;
- How to get depth? → By using camera motion;
- How to estimate camera motion? → By using 3D landmarks!

Overview

So it leads to the following conclusions:

- 3D landmark = 2D interest point + depth;
- A few initial landmarks must be given to the algorithm (otherwise camera motion cannot be estimated at the beginning, and no landmark can be placed)
- After an initialization phase, the MonoSLAM algorithm is an iterative process in which:
  - 2D interest points must evolve into 3D landmarks when the estimate of their depth is good enough;
  - 3D landmarks contibute to estimation of the state (camera and map).

Mono-SLAM Architecture



Initialization

- Calibration
- Landmark measurements

Prediction

- Kinematics model
- Extended Kalman Filter

Tracking

- Shi & Tomasi detector
- Correlation
- Depth estimation
- Particule filter

Correction

- Extended Kalman Filter
- Landmark quality evaluation

Details of Implementation

- The camera and map state is estimated through an **Extented Kalman Filter (EKF)**.

- this filter performs a model-based prediction, compares it to the measurements (camera data),

- thus applies a correction on the global state according to the innovation.

Initialization

❏ To use the camera data, it is necessary to know its parameters such as

  - ❏ Resolution,
  - ❏ focal length,
  - ❏ CMOS sensor dimension…
  - ❏ Calibration for fine estimation.

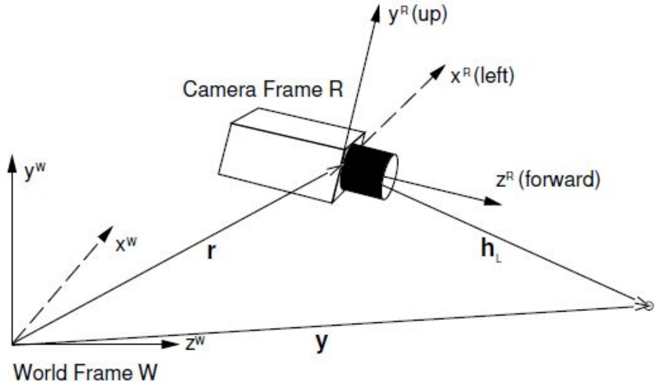
➢ A few initial landmarks have to be given

Prediction

$$f_v = \begin{pmatrix} r^W_{new} \\ q^{WR}_{new} \\ v^W_{new} \\ \omega^W_{new} \end{pmatrix} = \begin{pmatrix} r^W + v^W \Delta t \\ q^{WR} \times q\left(\omega^W \Delta t\right) \\ v^W \\ \omega^W \end{pmatrix}$$

This model allows only constant velocity movement (not very realistic)

we must add a noise as a way to allow for "errors" to be accepted (i.e., changes in the direction, speed, etc)



The camera state is described by its:

- Position (r)
- Orientation (using a quaternion, q)
- Linear speeds (v)
- Angular speeds (ω)

Prediction

$$a \sim \mathcal{N}\left(0, {\sigma_a}^2\right)$$

Angular acceleration is represented by the stochastic variable "α":

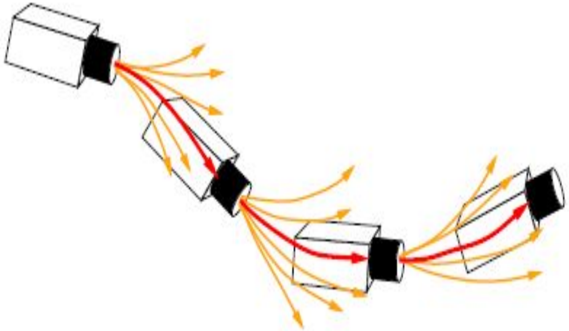
$$\alpha \sim \mathcal{N}\left(0, {\sigma_\alpha}^2\right)$$

Leading to the noise vector "n" (with covariance matrix "Pn"):

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

Introducing this noise, the kinematic model becomes:

$$f_v = \begin{pmatrix} r^W_{new} \\ q^{WR}_{new} \\ v^W_{new} \\ \omega^W_{new} \end{pmatrix} = \begin{pmatrix} r^W + \left(v^W + V^W\right) \Delta t \\ q^{WR} \times q\left(\left(\omega^W + \Omega^W\right) \Delta t\right) \\ v^W + V^W \\ \omega^W + \Omega^W \end{pmatrix}$$



Mono-SLAM

Covariance update

In the EKF, the new state estimate $\mathbf{F}(\mathbf{x},\mathbf{u})$ must be accompanied by the increase in state uncertainty (process noise covariance) for the camera after this motion.

Qv is found via the Jacobian calculation

$$\boxed{Q_v = \left(\frac{\partial \mathbf{F}}{\partial \mathbf{n}}\right) P_n \left(\frac{\partial \mathbf{F}}{\partial \mathbf{n}}\right)^T}$$

$$\boxed{P_{new}(k) = J_{\mathbf{Fx}}(k) P(k-1) J_{\mathbf{Fx}}^T(k) + Q_v(k)}$$

P_{new} : new state uncertainty

P : previous state uncertainty

P_n : covariance of noise vector $\mathbf{n}$

Mono-SLAM

Covariance of noise vector

-> The rate of growth of uncertainty in this motion model -> the size of P_n,

-> setting these parameters to small or large values -> smoothness of the motion

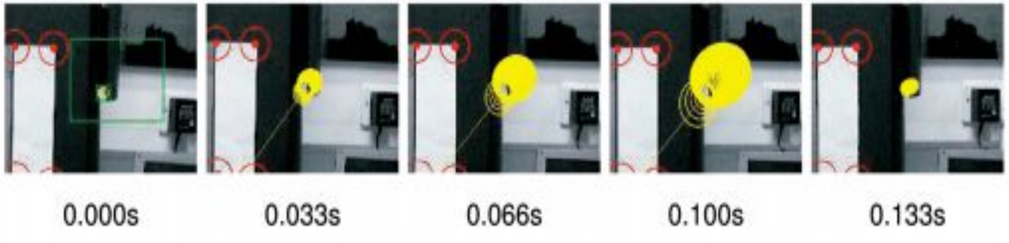
Small P_n

- We expect a very smooth motion with small accelerations, well placed to track motion but unable to cope with sudden rapid movements

High P_n

- The uncertainty in the system increases significantly at each time step.
- This can be cope with rapid accelerations.

Tracking



1. Interest Point detection indentifies landmarks

2. Use Shi and Tomasi detector to define salient points

3. 15 X15 pixel patches or 9 X 9

4. **normalization to increase the robostness**

5. Since (MONOCULAR) = No depth

6. Particle filter need to recover depth info

$$X = \begin{pmatrix} \langle I_x^2 \rangle & \langle I_{xy}^2 \rangle \\ \langle I_{xy}^2 \rangle & \langle I_y^2 \rangle \end{pmatrix}$$

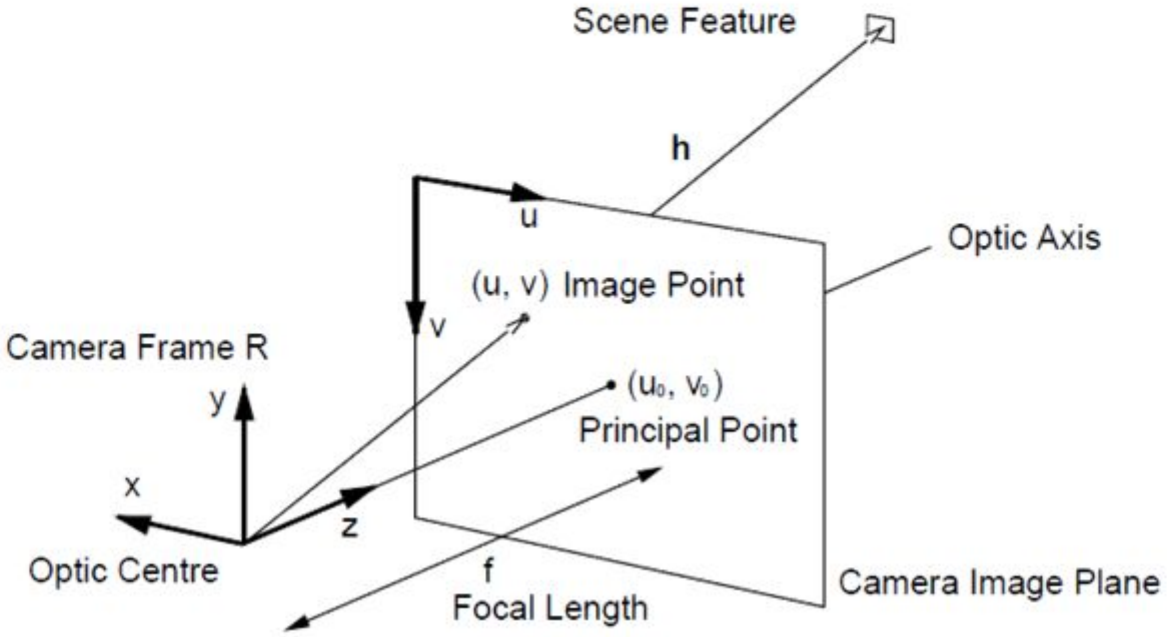
Correction Step

$$h_L^R = R^{RW}\left(y_i^W - r^W\right)$$

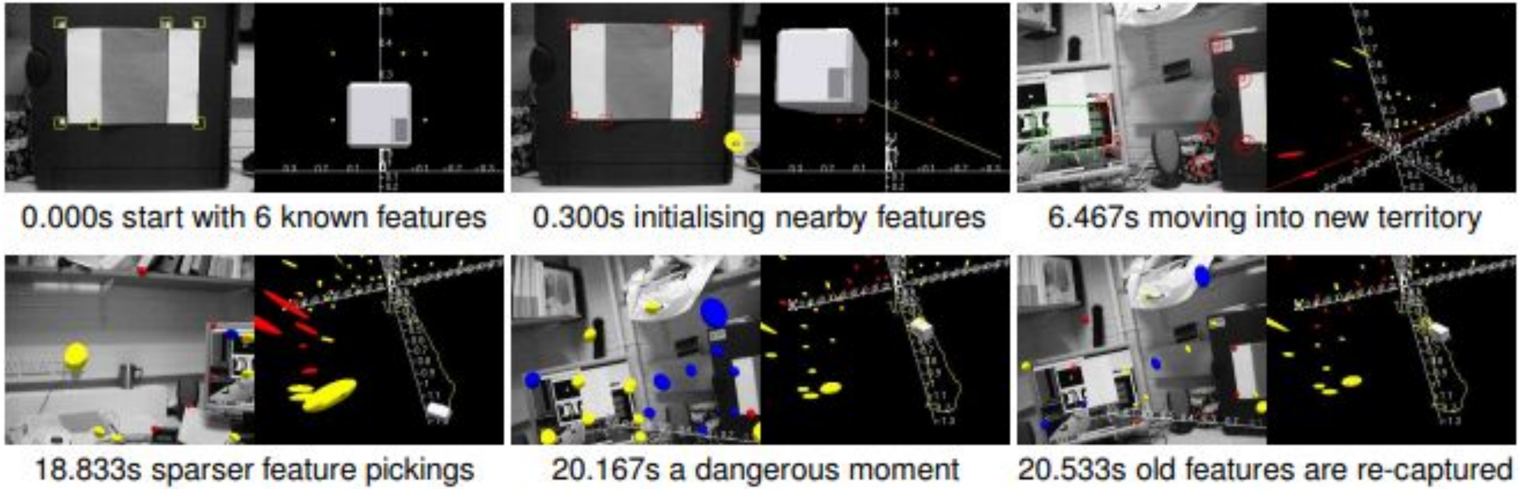
$$h_i = \begin{pmatrix} u \\ v \end{pmatrix} = \begin{pmatrix} u_0 - f k_u \frac{h_{Lx}^R}{h_{Lz}^R} \\ v_0 - f k_v \frac{h_{Ly}^R}{h_{Lz}^R} \end{pmatrix}$$

$$\frac{\partial h}{\partial X_v}, \frac{\partial h}{\partial Y_i}$$

EKF is very likely to be lost if the landmark assotiation is bad. Therefore, unrealiable matching must not be used.



Results



0.000s start with 6 known features

0.300s initialising nearby features

6.467s moving into new territory

18.833s sparser feature pickings

20.167s a dangerous moment

20.533s old features are re-captured

Conclusion

We have described a principled, Bayesian, top-down approach to sequential Simultaneous Localisation and Mapping or Structure from Motion which takes account of the extra sources of information often neglected in batch methods to push performance past the real-time barrier, and demonstrated robust performance in an indoor scene.

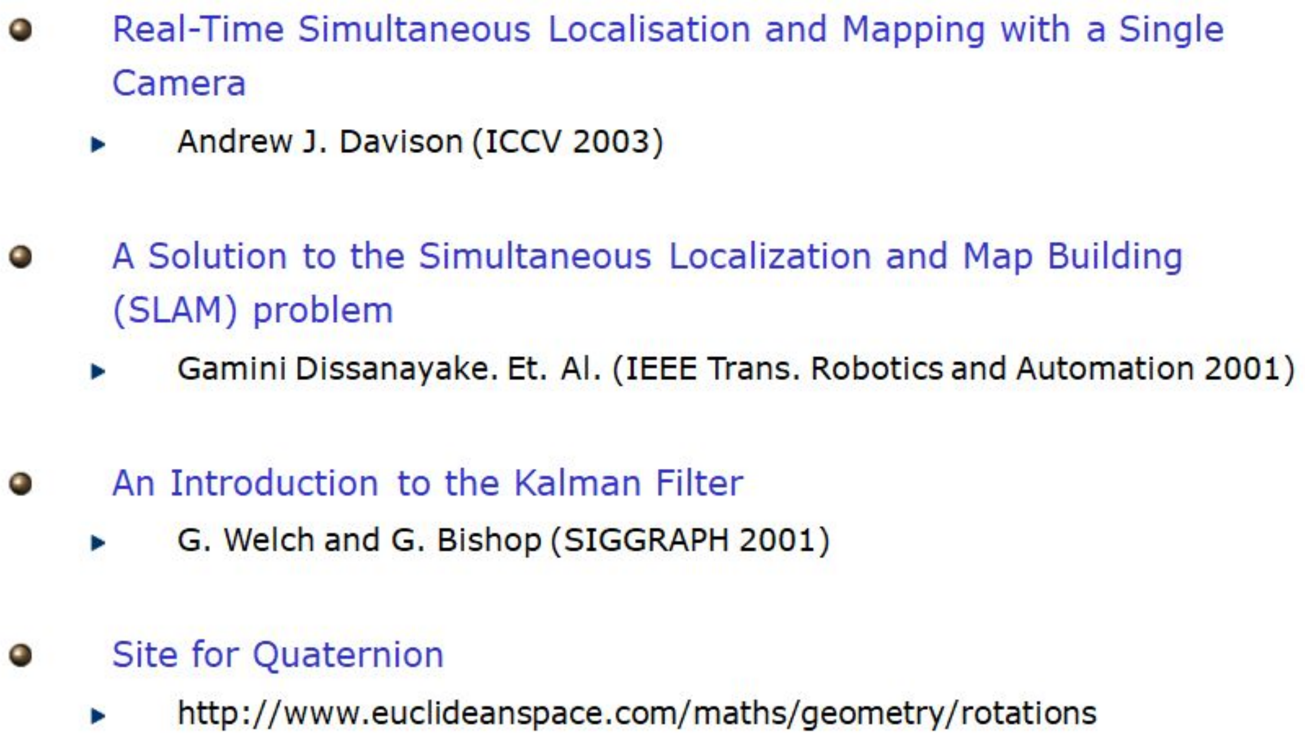
But It has limitations ...

Limitation

From a theoritical point of view, there are several limits to this approach:

- EKF is more prone to divergence due to a rough linearization (first order only).
- Tuning manually the noise parameters leads to difficult recovering for the filter from a false landmarks. To help with this issue, we use many landmarks and we remove the "bad" landmarks.
- particule filter can be very slow to converge without good discriminant movements
- This approch is based on visual interest points, and therefore is not possible in uniform environments.
- The auto-focus changes the focal length automatically. Calibration was consequently not meaningful.

References



- Real-Time Simultaneous Localisation and Mapping with a Single Camera
  - Andrew J. Davison (ICCV 2003)
- A Solution to the Simultaneous Localization and Map Building (SLAM) problem
  - Gamini Dissanayake. Et. Al. (IEEE Trans. Robotics and Automation 2001)
- An Introduction to the Kalman Filter
  - G. Welch and G. Bishop (SIGGRAPH 2001)
- Site for Quaternion
  - http://www.euclideanspace.com/maths/geometry/rotations