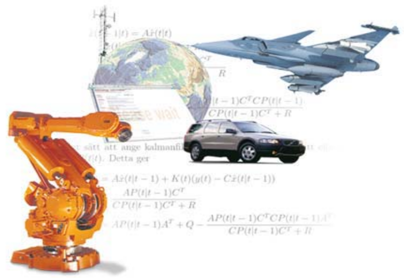
Lecture 5 – Visual SLAM and Sensor Fusion Using Camera Images



Thomas Schön,
Division of Automatic Control,
Department of Electrical Engineering,
Linköping University.

Email: schon@isy.liu.se

The Kalman filter is the sequential solution to a weighted least squares problem

Content – Lecture 5

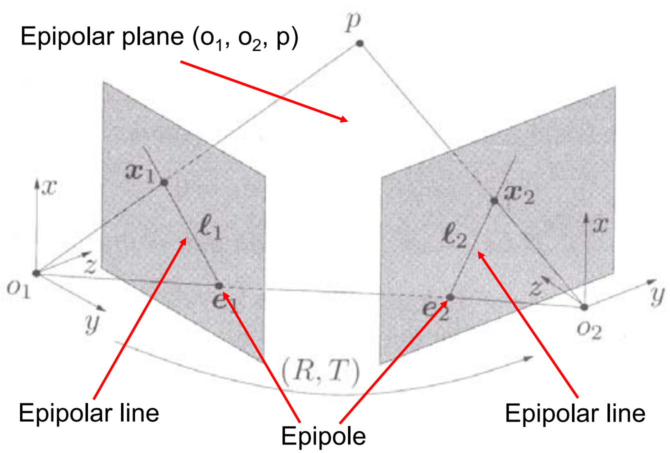
1. Summary of Lecture 4
2. State Estimation as an Optimization Problem
3. Linear Regression as an Optimization Problem
4. Probabilistic Graphical Models
5. Modeling SLAM and VO Problems
6. Case Study – Visual SLAM ("FrameSLAM")
7. Sensor Fusion using Cameras

Personal Reflection: What we currently see in the SLAM community is very much what we previously saw in the control community when MPC came around.

Summary – Lecture 4 (Epipolar Geometry)

Epipolar geometry = the geometry of two views

Epipolar plane (o_1, o_2, p)



We derived the epipolar constraint

$$x_2^T \hat{T} R x_1 = 0$$

Summary – Lecture 4 (Reconstruction)

Reconstruction tries to recover a model of the 3D scene from multiple images.

The eight-point algorithm – good insight into the underlying geometry.

$$\tilde{x}_1^j = x_1^j + w_1^j, \quad \tilde{x}_2^j = x_2^j + w_2^j, \qquad w_1^j = \begin{pmatrix} w_{11}^j \\ w_{12}^j \\ 0 \end{pmatrix} \quad w_2^j = \begin{pmatrix} w_{21}^j \\ w_{22}^j \\ 0 \end{pmatrix}$$

measurements ($\tilde{x}$), ideal image coordinate (model) (x), noise (w)

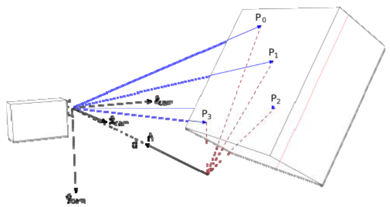
$$\arg\max_{x} \sum_{i,j} \log p\left((\tilde{x}_i^j - x_i^j) | x, R, T\right)$$

Reconstruction is a (typically) high dimensional **parameter estimation** problem.

Summary – Lecture 4 (Homography)

A **homography** is an invertible transformation from a projective plane to a projective plane that maps straight lines to straight lines.

X_1, X_2 - coordinates of p relative to camera 1 and 2, respectively



$$X_2 = RX_1 + T\frac{1}{d}n^TX_1 = \underbrace{(R + \frac{1}{d}Tn^T)}_{H}X_1$$

$$X_2 = HX_1$$

Common applications include autonomous helicopter landing, road geometry estimation.

State Estimation as an Optimization Problem

For a nonlinear state-space model with additive Gaussian noise,

$$x_{t+1} = f(x_t) + w_t, \qquad w_t \sim \mathcal{N}(0, Q)$$
$$y_T = h(x_t) + e_t, \qquad e_t \sim \mathcal{N}(0, R).$$

the *maximum a posteriori* (MAP) estimate

$$\hat{x}_{1:t} = \arg\max_{x_{1:t}} p(x_{1:t}|y_{1:t}) = \arg\max_{x_{1:t}} p(y_{1:t}|x_{1:t})p(x_{1:t})$$

is given by

$$\hat{x}_{1:t} = \arg\min_{x_{1:t}} \left(\|x_1 - \bar{x}_1\|^2_{P_1^{-1}} + \sum_{i=2}^{t} \|x_i - f(x_{i-1})\|^2_{Q^{-1}} + \sum_{i=1}^{t} \|y_i - h(x_i)\|^2_{R^{-1}} \right)$$

i.e., a nonlinear least-squares problem

Linear Gaussian Model – Kalman Filter Connection

The MAP estimate for a linear, Gaussian model is found by solving

$$\hat{x}_{1:t} = \arg\min_{x_{1:t}} \left(\|x_1 - \bar{x}_1\|^2_{P_1^{-1}} + \sum_{i=2}^{t} \|x_i - Ax_{i-1}\|^2_{Q^{-1}} + \sum_{i=1}^{t} \|y_i - Cx_i\|^2_{R^{-1}} \right)$$

which is a quadratic program (QP). It can be shown that the Kalman filter is the sequential solution to the above problem.

The Kalman filter can be interpreted as the sequential solution to a weighted least-squares problem.

Arbitrary convex constraints can be added to the above problem and standard software applies for finding the global optimum.

Clear relation to MPC

Linear Regression

$$y_t = \varphi_t^T\theta + e_t, \quad t = 1, \ldots, N$$
$$Y = \Phi^T\theta + E$$

Most common estimators are in the form

$$\hat{x}_{1:t} = \arg\min_{\theta} \|Y - \Phi^T\theta\|_2^2 + \lambda\|\theta\|_p^p$$

	Cost function	Estimator	Prior
Maximum likelihood	$\|Y - \Phi^T\theta\|_2^2$	ML	-
Ridge regression, regularized least squares	$\|Y - \Phi^T\theta\|_2^2 + \lambda\|\theta\|_2^2$	MAP	$\mathcal{N}(0, \sigma^2/\lambda)$
LASSO	$\|Y - \Phi^T\theta\|_2^2 + \lambda\|\theta\|_1$	MAP	$\mathcal{L}(0, 2\sigma^2/\lambda)$

More details are available in the lecture notes.

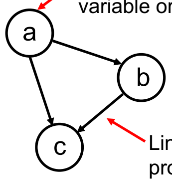
Probabilistic Graphical Models

Probabilistic graphical models use graph theory to represent conditional independencies between a set of random variables.

Why bother, we already have equations

- It is often a **VERY** good idea to view the same problem using **different perspectives**.
- Graphs provide a simple way the **visualize the structure** of a probabilistic model.
- Insights about the **model properties**, such as conditional independence, can be obtained from inspection of the graph.
- Inference can be carried out directly in the graph.

Probabilistic Graphical Models



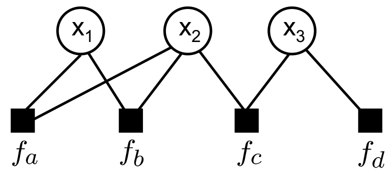
Node (aka vertex), represents a random variable or a group of random variables

Link (aka edge and arc) describes the probabilistic relation between the variables

$$p(a, b, c) = p(c|a, b)p(b|a)p(a)$$

Three different types of graphs

- Bayesian networks (aka Belief network), a directed graph
- Markov random fields (aka Markov network), an undirected graph
- Factor graph

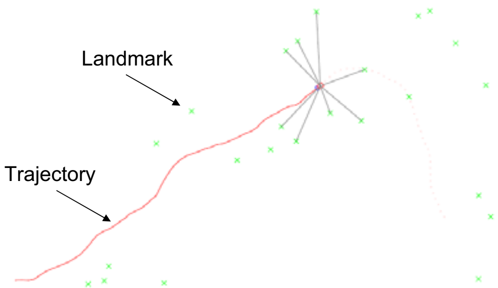


$$p(x) = f_a(x_1, x_2)f_b(x_1, x_2)f_c(x_2, x_3)f_d(x_3)$$

SLAM in Terms of Graphs

Example borrowed from

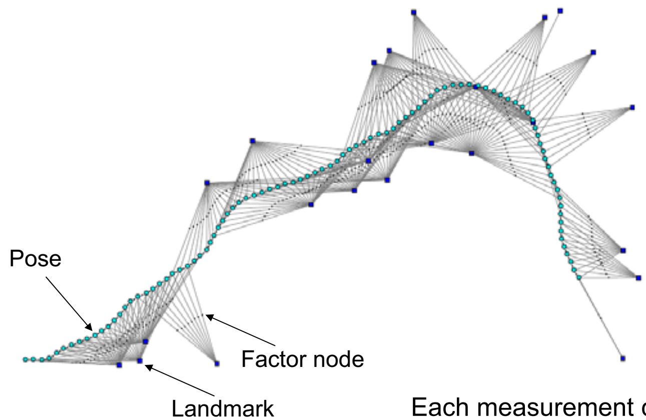
Dellaert, F and Kaess, K. **Square Root SAM: Simultaneous Localization and Mapping via Square Root Information Smoothing**, *International Journal of Robotics Research, 25(12):1181-1203, 2006.*



Synthetic example with 24 landmarks, with a robot acquiring 422 bearing and range measurements along a trajectory consisting of 95 poses.

SLAM in Terms of Graphs

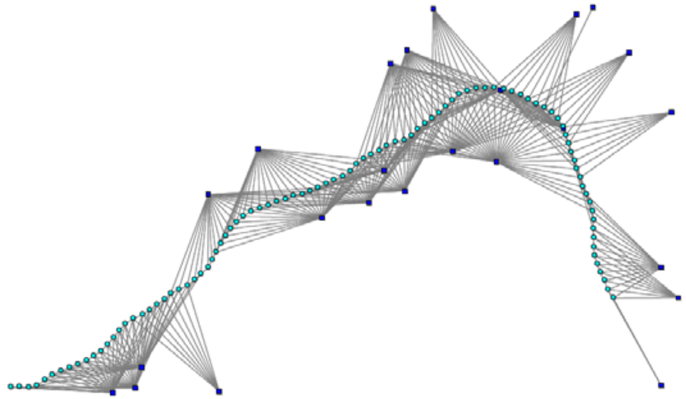
Same example illustrated in terms of a **factor graph**



Each measurement corresponds to a factor node

SLAM in Terms of Graphs

Same example illustrated in terms of a **Markov random field**



Case Study – FrameSLAM

Konolige, K. and Agrawal, M. **FrameSLAM: from Bundle Adjustment to Realtime Visual Mappping**, *IEEE Transactions on Robotics*, 24(5):1066-1077, 2008.

Builds on the ideas we have talked about to far, inspired by bundle adjustment, they consider the ML problem

$$\arg\max_{x_{1:t}} p(y_{1:t}|x_{1:t})$$

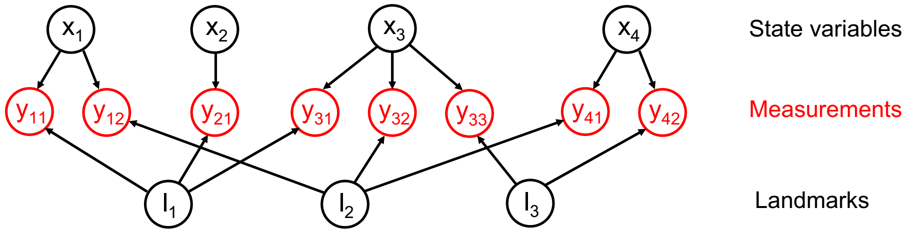
rather than considering the MAP problem

$$\arg\max_{x_{1:t}} p(x_{1:t}|y_{1:t})$$

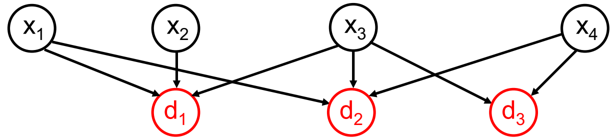
which implies that they do not account for any dynamics.

The problem is modeled as a large **Bayesian network** consisting of **all camera poses and all landmarks**. This huge network is then **reduced** into smaller versions that are solved, so called **skeletons**.

Reducing the Problem Size – “Skeleton” Graphs

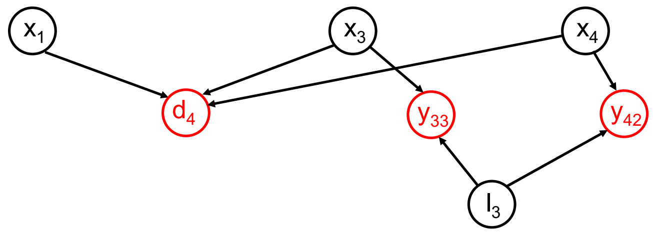


Note that there are no links between the states, implies no dynamics (ML, not MAP).



Here, all the features have been marginalized, introducing artificial measurement between the poses, d_1, d_2 and d_3.

Reducing the Problem Size – “Skeleton” Graphs



Here, we have marginalized the second pose and all the features but one, introducing artificial measurements between the remaining poses, d_4.

Reducing the problem size corresponds to **marginalization**.

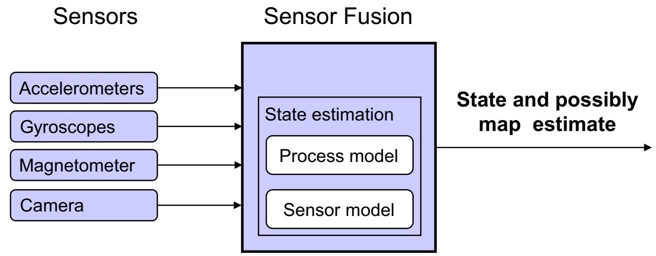
Map represenation: Skeleton = a reduced graph, with relative pose information.

Reflection

Engineering is the art of making the correct approximations at the right time

and the SLAM problem clearly illustrates just that.

Sensor Fusion Using Images



$$y_{tj} = h(x_t, l_{c_{tj}}) + e_{tj}, \quad e_{tj} \sim \mathcal{N}(0, R)$$

The landmarks can be part of the estimation problem (SLAM) or known in advance (e.g. MATRIS project)

Sensor Fusion Using Images

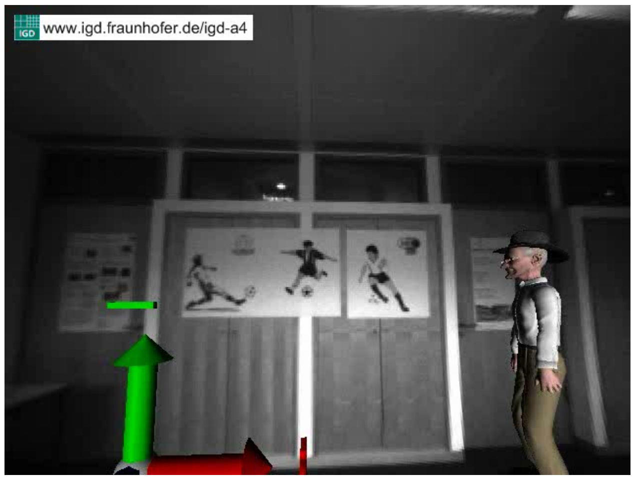
Here are two alternatives:

1. Include the vision measurements and the associated infrastructure in a standard sensor fusion filter.

   This will be done and understood during HW3 for a UAV application, where measurements consists in 3D accelerometers, 3D gyroscopes, barometer and camera images.

2. Make use of the optimization problem we derived and "simply" include the new measurements.

An Example of Alternative 1 – the MATRIS Project



Alternative 2 – Very Brief

$$\underset{x_{1:t}, l_{1:N}}{\arg\min}\; V(x_{1:t}, l_{1:N})$$

where

$$V(x_{1:t}, l_{1:N}) = \|x_1 - \bar{x}_1\|^2_{\bar{P}_1^{-1}} + \sum_{i=2}^{t} \|x_i - f(x_{i-1})\|^2_{Q^{-1}} + \sum_{i=1}^{t}\sum_{j=1}^{M_i} \|y_{ij} - h(x_i, l_{c_{ij}})\|^2_{R^{-1}}$$

$$y_t^{\omega} = \omega_t + \delta_t^{\omega} + e_t^{\omega},$$
$$y_t^{a} = R(\ddot{c}_t - g) + \delta_t^{a} + e_t^{a}.$$

Include the measurements into the cost function, by adding

$$\sum_{i=2}^{t} \|y_i^{\omega} - \omega_t - \delta_t^{\omega}\|^2_{R^{-1}} + \sum_{i=2}^{t} \|y_i^{a} - R(\ddot{c}_t - g) - \delta_t^{a}\|^2_{R^{-1}}$$

or by including the measurements as inputs in the process model.

Course Goal – Did we Reach It?

The **aim of this course** is to describe how we can **pose and solve** various **estimation problems** based on camera images and how cameras can be used together with other sensors.

- Rigid body motion
- Camera models
- Camera calibration
- Feature extraction
- Feature tracking
- Epipolar geometry
- Sensor fusion using cameras
- Industrial applications

Course evaluation forms will be sent out during next week

Guest Lecture from C3 on Friday!!

C3 The Third Dimension

CTO Petter Torle will give an overview of C3 technology

www.c3technologies.com

Date: Friday, January 30

Time: 13.15

Place: Here

Welcome!!!