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
\(x_2^T R x_1 = 0\)

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.

\[
\begin{align*}
\bar{x}_1 &= x_1^T + w_1^T \\
\bar{x}_2 &= x_2^T + w_2^T \\
w_1^T &= \begin{pmatrix} w_1^{x_1} \\
w_1^{x_2} \\
0 \end{pmatrix} \\
w_2^T &= \begin{pmatrix} w_2^{x_1} \\
w_2^{x_2} \\
0 \end{pmatrix}
\end{align*}
\]

\[
\arg\max_x \sum_{i,j} \log p((\bar{x}_i - x_i)|x, H, I)
\]

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

\[
X_2 = RX_1 + \frac{1}{d} R^T X_1 = (R + \frac{1}{d} R^T) X_1
\]

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,

\[
x_{t+1} = f(x_t) + w_t, \quad w_t \sim \mathcal{N}(0, Q)
\]
\[
y_t = h(x_t) + e_t, \quad e_t \sim \mathcal{N}(0, R).
\]

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

\[
\hat{x}_{1:t} = \arg \min_{x_{1:t}} \left( \|x_t - \bar{x}_t \|^2_{P_t} + \sum_{i=2}^t \|x_t - A x_{t-1}\|^2_{Q_{t-1}} + \sum_{i=1}^t \|y_i - h(x_t)\|^2_{R^{-1}} \right)
\]

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

\[
\hat{x}_{1:t} = \arg \min_{\bar{x}_{1:t}} \left( \|x_t - \bar{x}_t \|^2_{P_t} + \sum_{i=2}^t \|x_t - A x_{t-1}\|^2_{Q_{t-1}} + \sum_{i=1}^t \|y_i - C x_t\|^2_{R^{-1}} \right)
\]

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.

Linear Regression

The most common estimators are in the form

\[
y_t = \varphi^T \theta + e_t, \quad t = 1, \ldots, N
\]
\[
Y = \Phi^T \theta + E
\]

Most common estimators are in the form

\[
\hat{\theta}_{1:t} = \arg \min_{\theta} \|Y - \Phi^T \theta \|^2_{2} + \lambda \|\theta\|^2_{P}
\]

Maximum likelihood

Ridge regression, regularized least squares

LASSO

<table>
<thead>
<tr>
<th>Cost function</th>
<th>Estimator</th>
<th>Prior</th>
</tr>
</thead>
<tbody>
<tr>
<td>|Y - \Phi^T \theta|^2_{2}</td>
<td>ML</td>
<td>-</td>
</tr>
<tr>
<td>|Y - \Phi^T \theta|^2_{2} + \lambda |\theta|^2_{P}</td>
<td>MAP</td>
<td>( N(0, \sigma^2/\lambda) )</td>
</tr>
<tr>
<td>|Y - \Phi^T \theta|^2_{2} + \lambda |\theta|_{1}</td>
<td>MAP</td>
<td>( L^{2}(0, 2\sigma^2/\lambda) )</td>
</tr>
</tbody>
</table>

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.

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

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

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

Three different types of graphs

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

SLAM in Terms of Graphs

Example borrowed from


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 Markov random field

Case Study – FrameSLAM


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

$$\arg \max_{x_{1:2}} p(y_{1:2}|x_{1:2})$$

rather than considering the MAP problem

$$\arg \max_{x_{1:2}} p(x_{1:2}|y_{1:2})$$

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).

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 representation: 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

Sensor Fusion

- Accelerometers
- Gyrosopes
- Magnetometer
- Camera

State estimation
- Process model
- Sensor model

State and possibly map estimate

\[ y_{tj} = h(x_t, l_{Cij}) + r_{tj}, \quad r_{tj} \sim N(0, R) \]

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

www.lpg.fraunhofer.de/lpg.at
Alternative 2 – Very Brief

\[ \arg \min_{x_{1:N}} V(x_{1:N}) \]

where

\[ y_t^u = \omega_t + \delta_t^u + e_t^u, \]
\[ y_t^g = R(\tilde{v}_t - y) + \delta_t^g + e_t^g. \]

\[ V(x_{1:N}) = \|x_1 - \tilde{x}_1\|^2_{F_1} + \sum_{i=2}^{t} \|x_i - f(x_{i-1})\|^2_{Q_{i-1}} \]
\[ + \sum_{i=1}^{t} \sum_{j=1}^{M_i} \|y_{ij} - h(x_i, l_{ij})\|^2_{R_{ij}} \]

Include the measurements into the cost function, by adding

\[ \sum_{i=2}^{t} \|y_t^u - \omega_t - \delta_t^u\|^2_{R_{i-1}} + \sum_{i=2}^{t} \|y_t^g - R(\tilde{v}_t - g) - \delta_t^g\|^2_{R_{i-1}} \]

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!!

CTO Petter Torle will give an overview of C3 technology

www.c3technologies.com

Date: Friday, January 30
Time: 13.15
Place: Here

Welcome!!!