Introduction to Mobile Robotics

Basics of LSQ Estimation
Line and Circle Extraction

Kai O. Arras
Feature Extraction: Motivation

**Landmarks** for:
- Localization
- SLAM
- Scene analysis

Examples:
- **Lines, corners, clusters:** good for indoor
- **Circles, rocks, plants:** good for outdoor
Features: Properties

A feature/landmark is a **physical object** which is

- **static**
- **perceptible**
- (at least locally) **unique**

Abstraction from the raw data...

- **type** (range, image, vibration, etc.)
- **amount** (sparse or dense)
- **origin** (different sensors, map)

+ Compact, efficient, accurate, scales well, semantics
  – Not general
Feature Extraction

Can be subdivided into two subproblems:

- **Segmentation:** *Which* points contribute?
- **Fitting:** *How* do the points contribute?
Example: Local Map with Lines

Raw range data

Line segments
Example: Global Map with Lines

**Expo.02 map**
- 315 m²
- 44 Segments
- 8 kbytes
- 26 bytes / m²
- Localization accuracy ~1cm
Example: Global Map w. Circles

Victoria Park, Sydney

- Trees
Split and Merge

Split

No more Splits

Merge
Split and Merge

Algorithm

Split
• Obtain the line passing by the two extreme points
• Find the most distant point to the line
• If distance > threshold, split and repeat with the left and right point sets

Merge
• If two consecutive segments are close/collinear enough, obtain the common line and find the most distant point
• If distance <= threshold, merge both segments
Split and Merge: Improvements

- Residual analysis before split

\[ \sum_{i = P_S} P_E d_i^2 > \sum_{i = P_S} P_B d_i^2 + \sum_{i = P_B} P_E d_i^2 \]

Split only if “interpretation gets better” in terms of error sums

[Castellanos 1998]
Split and Merge: Improvements

- Merge **non-consecutive** segments as a post-processing step
Line Representation

Choice of the line representation matters!

Intercept-Slope

\[ y = ax + b \]

\[ C = \begin{bmatrix} \sigma_a^2 & \sigma_{ab} \\ \sigma_{ba} & \sigma_b^2 \end{bmatrix} \]

Hessian model

\[ x \cos \alpha + y \sin \alpha - r = 0 \]

\[ C = \begin{bmatrix} \sigma_\alpha^2 & \sigma_{ar} \\ \sigma_{ra} & \sigma_r^2 \end{bmatrix} \]

Each model has advantages and drawbacks
Fit Expressions

Given:
A set of $n$ points in polar coordinates

Wanted:
Line parameters $\alpha, r$

\[
\tan 2\alpha = \frac{2 \sum w_i \sum_{i < j} w_i w_j \rho_i \rho_j \sin(\theta_i + \theta_j) + \frac{1}{\sum w_i} \sum (w_i - \sum w_i) w_i \rho_i^2 \sin 2\theta_i}{2 \sum w_i \sum_{i < j} w_i w_j \rho_i \rho_j \cos(\theta_i + \theta_j) + \frac{1}{\sum w_i} \sum (w_i - \sum w_i) w_i \rho_i^2 \cos 2\theta_i}
\]

\[
r = \frac{\sum w_i \rho_i \cos(\theta_i - \alpha)}{\sum w_i}
\]

[Arras 1997]
LSQ Estimation

Regression, Least Squares-Fitting

\[ \varepsilon_i = x_i \cos \alpha + y_i \sin \alpha - r \]

\[ S = \sum_{i=1}^{n} \varepsilon_i^2 \]

Solve the non-linear equation system

\[ \frac{\partial S}{\partial \alpha} = 0 \quad \frac{\partial S}{\partial r} = 0 \]

Solution (for points in Cartesian coordinates):
\[ \rightarrow \text{Solution on blackboard} \]
Circle Extraction

Can be formulated as a linear regression problem

Given $n$ points $\mathcal{P} = \{P_i\}_{i=1}^n$ with $P_i = (x_i, y_i)^T$

Circle equation: $(x_i - x_c)^2 + (y_i - y_c)^2 = r_c^2$

Develop circle equation

$$x_i^2 - 2x_ix_c + x_c^2 + y_i^2 - 2y_iy_c + y_c^2 = r_c^2$$

$$(-2x_i - 2y_i 1) \begin{pmatrix} x_c \\ y_c \\ x_c^2 + y_c^2 - r_c^2 \end{pmatrix} = (-x_i^2 - y_i^2)$$
Circle Extraction

Leads to **overdetermined** equation system $A \cdot x = b$

$$A = \begin{pmatrix}
-2x_1 & -2y_1 & 1 \\
-2x_2 & -2y_2 & 1 \\
\vdots & \vdots & \vdots \\
-2x_n & -2y_n & 1 \\
\end{pmatrix} \quad b = \begin{pmatrix}
-x_1^2 - y_1^2 \\
-x_2^2 - y_2^2 \\
\vdots \\
-x_n^2 - y_n^2 \\
\end{pmatrix}$$

with vector of unknowns

$$x = (x_c \quad y_c \quad x_c^2 + y_c^2 - r_c^2)^T$$

Solution via **Pseudo-Inverse**

$$x = (A^T A)^{-1} A^T \cdot b$$

(assuming that $A$ has full rank)
Fitting Curves to Points

**Attention:** Always know the errors that you minimize!

Algebraic versus geometric fit solutions [Gander 1994]
LSQ Estimation: Uncertainties?

How does the \textbf{input uncertainty} propagate over the fit expressions to the \textbf{output}?

$X_1, \ldots, X_n$: Gaussian input random variables

$A, R$: Gaussian output random variables
Example: Line Extraction

**Wanted:** Parameter Covariance Matrix

\[ C_{AR} = \begin{bmatrix} \sigma_A^2 & \sigma_{AR} \\ \sigma_{AR} & \sigma_R^2 \end{bmatrix} \]

Simplified sensor model:
all \( \sigma_{\theta_i}^2 = 0 \), independence

\[ C_{AR} = F_X C_X F_X^T \]

Result: Gaussians in the parameter space
Line Extraction in Real Time

- Robot *Pygmalion*
  EPFL, Lausanne
- CPU: PowerPC 604e at 300 MHz
  Sensor: 2 SICK LMS
- Line Extraction Times: ~25 *ms*
Derivations (1/4)

Result: Line Fit Cartesian Coordinates

(only for $r$, $\alpha$ more complicated...)

\[
\frac{\partial S}{\partial r} = 0
\]
\[
\iff \frac{\partial}{\partial r} \{ \sum \epsilon_i^2 \} = \sum \frac{\partial}{\partial r} \{ \epsilon_i^2 \} = 2 \sum \epsilon_i \frac{\partial}{\partial r} \{ \epsilon_i \}
\]
\[
\iff 2 \sum (x_i \cos \alpha + y_i \sin \alpha - r)(-1) = 0
\]
\[
\iff \sum (x_i \cos \alpha + y_i \sin \alpha - r) = 0
\]
\[
\iff \sum x_i \cos \alpha + \sum y_i \sin \alpha - nr = 0
\]
\[
\iff r = 1/n \sum x_i \cos \alpha + 1/n \sum y_i \sin \alpha
\]
\[
\iff r = \bar{x} \cos \alpha + \bar{y} \sin \alpha
\]