Introduction to Mobile Robotics

Basics of LSQ Estimation, Geometric Feature Extraction

Wolfram Burgard, Cyrill Stachniss,
Maren Bennewitz, Kai Arras

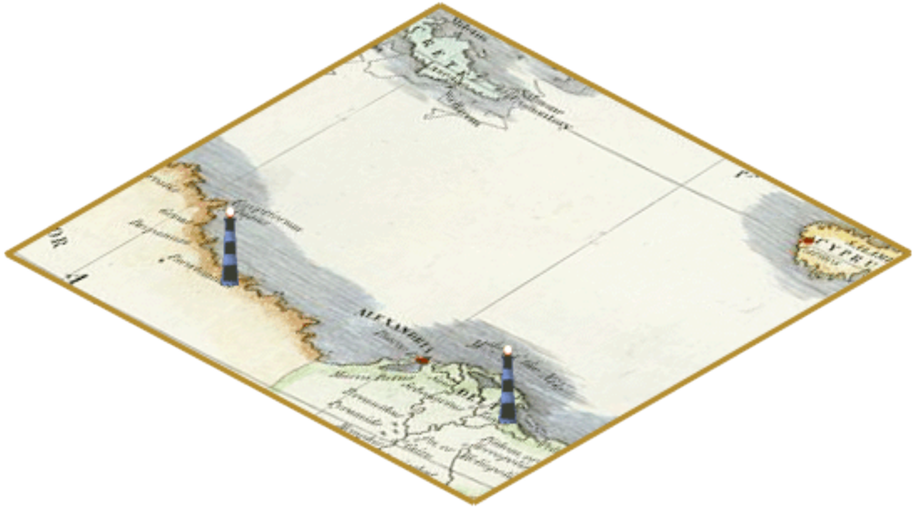
Slides by Kai Arras

Last update: June 2010

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

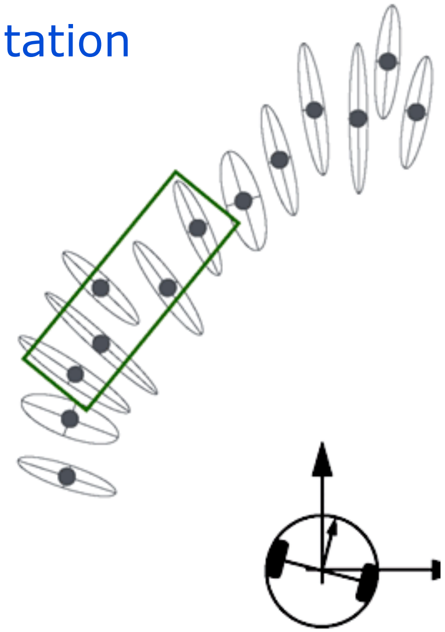
– Not general

Geometric Feature Extraction

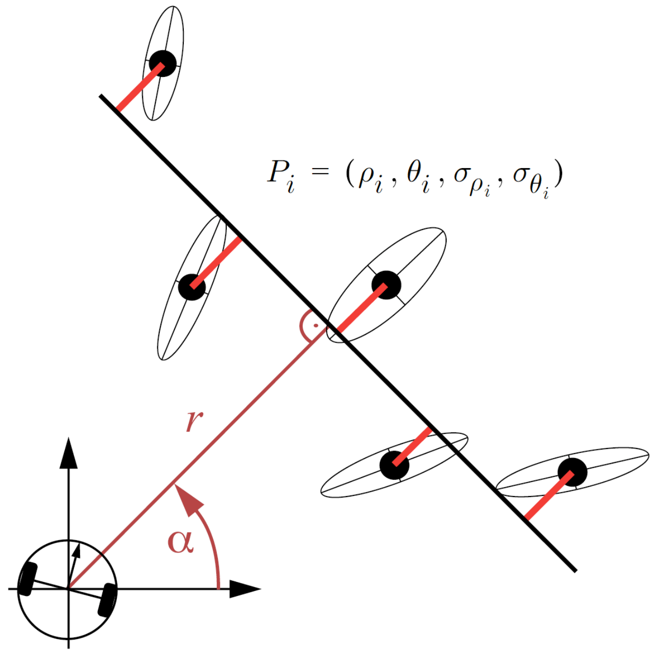
Can be subdivided into two subproblems:

- **Segmentation:** *Which* points contribute?
- **Fitting:** *How* do the points contribute?

Segmentation

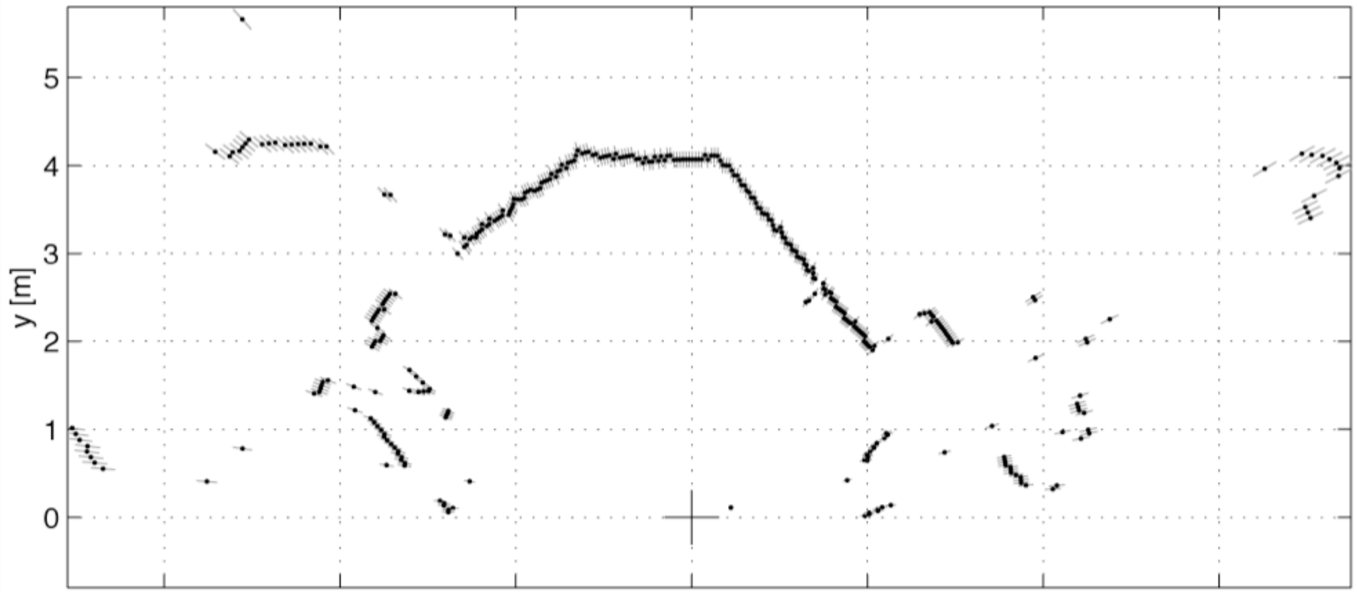


Fitting

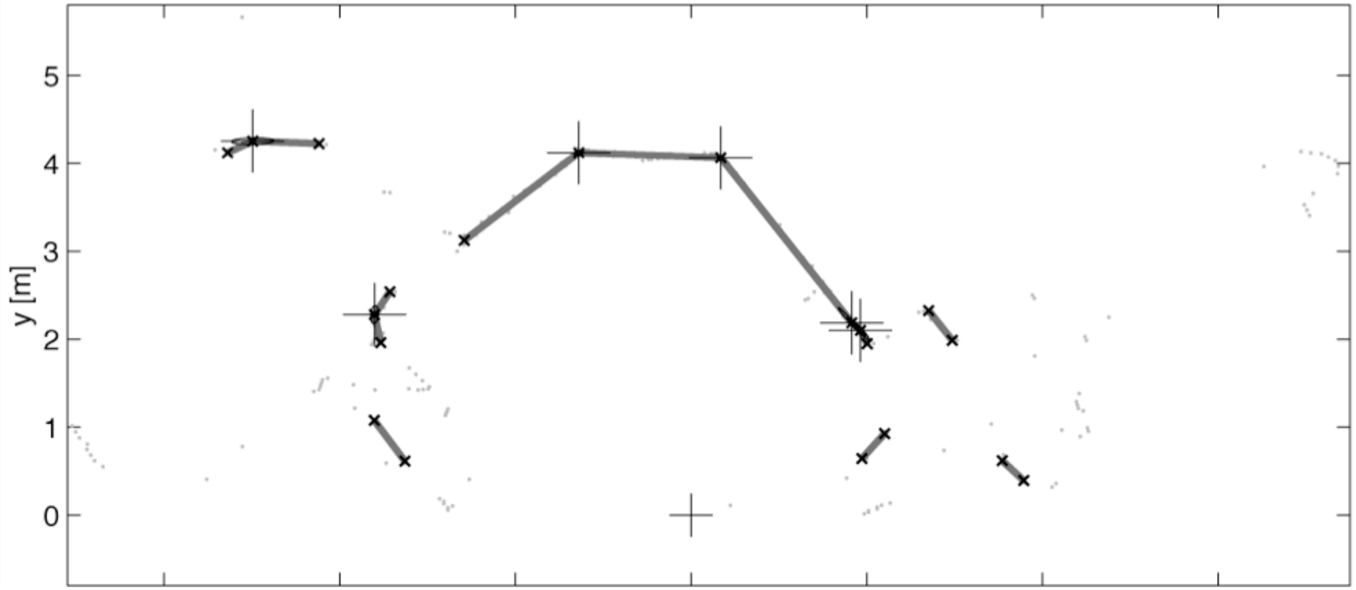


Example: Local Map with Lines

Raw
range data



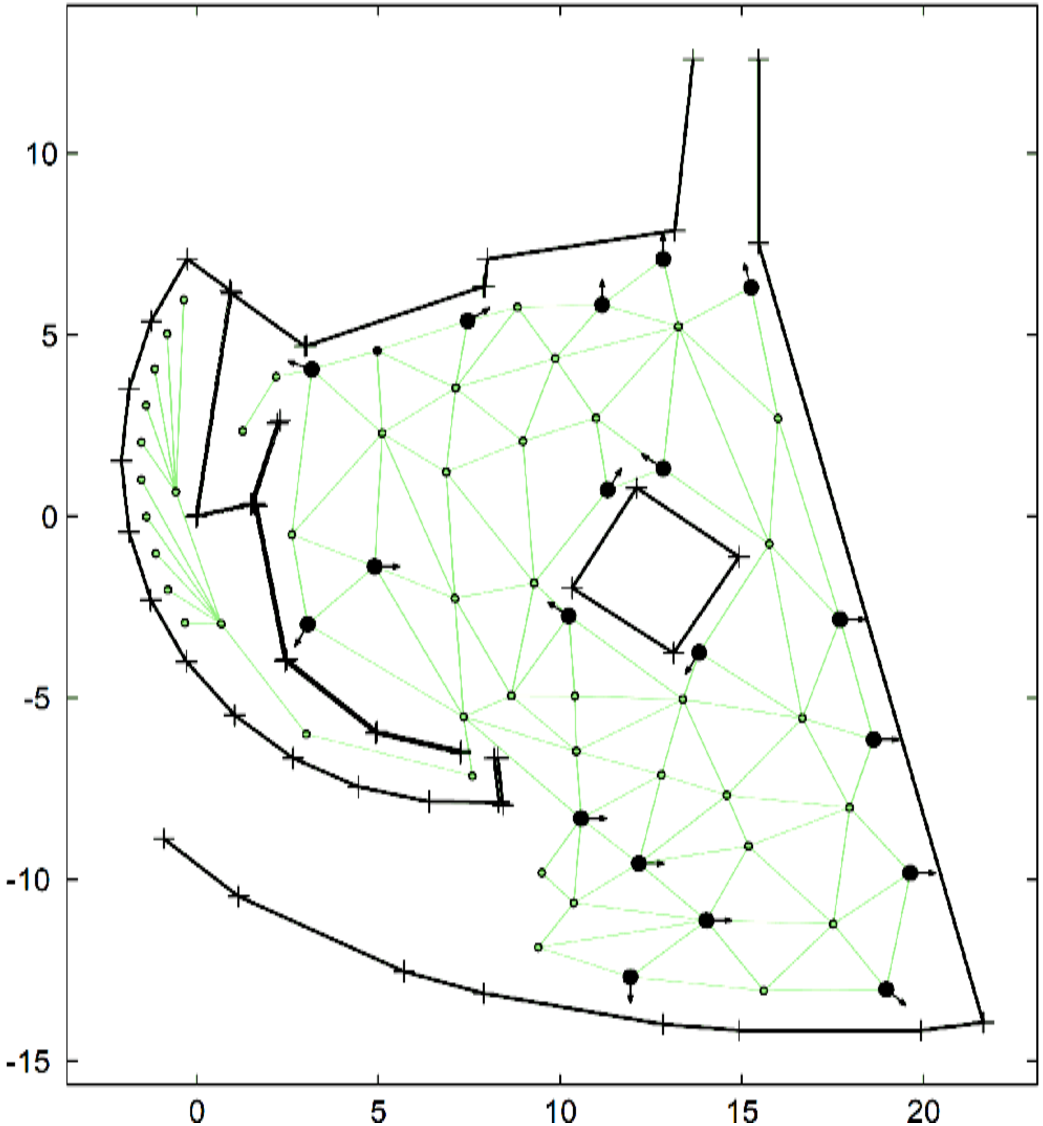
Line
segments



Example: Global Map with Lines

Expo.02 map

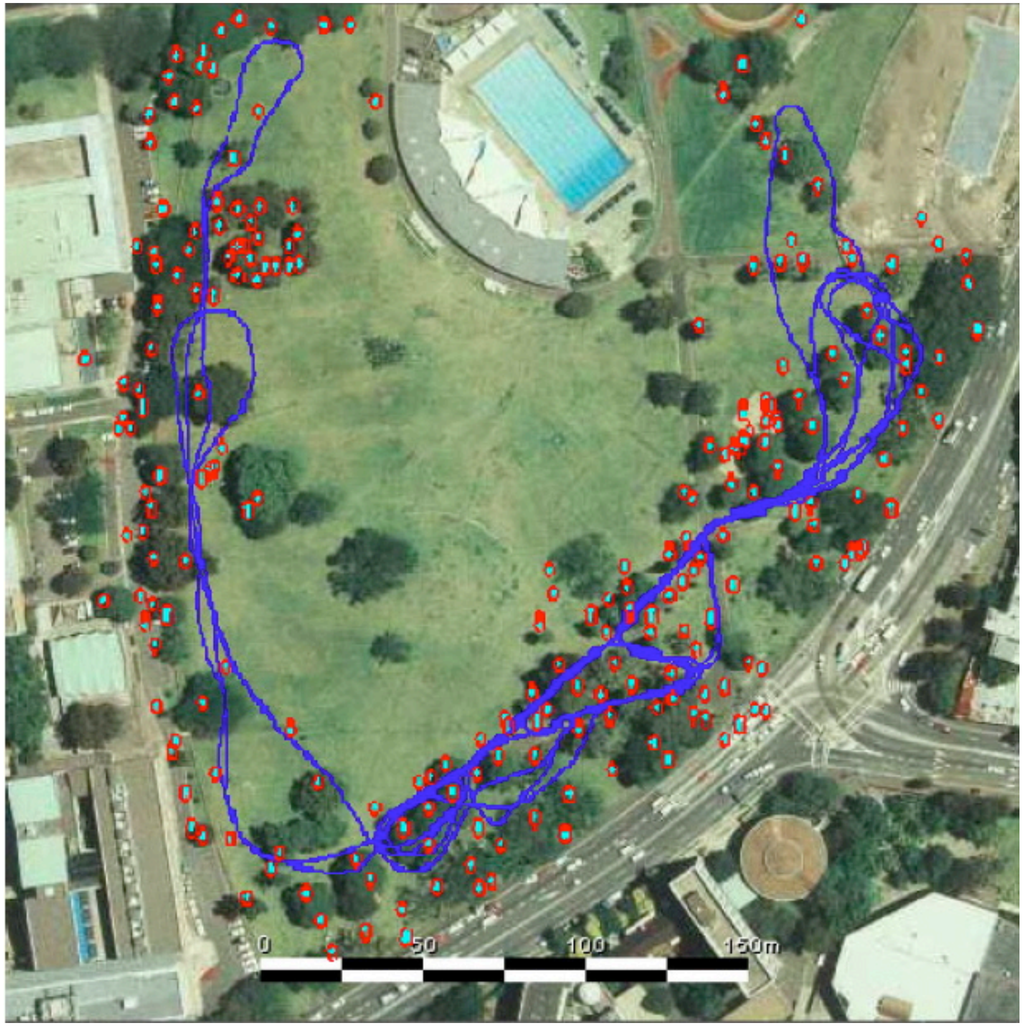
- 315 m^2
- 44 Segments
- 8 kbytes
- 26 bytes / m^2
- Localization accuracy ~1cm



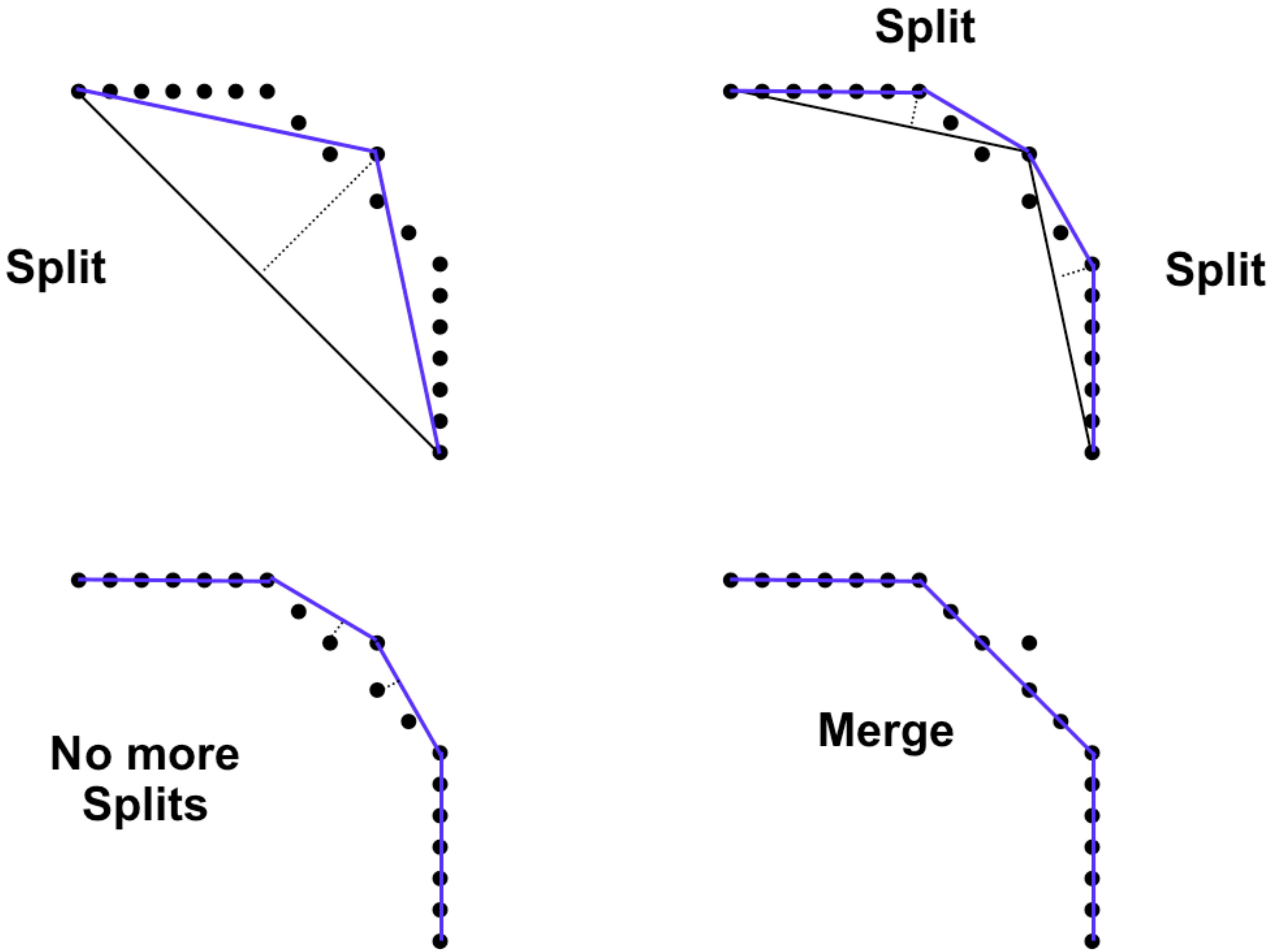
Example: Global Map w. Circles

Victoria Park, Sydney

- Trees



Split and Merge



Picture by J. Tardos

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

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

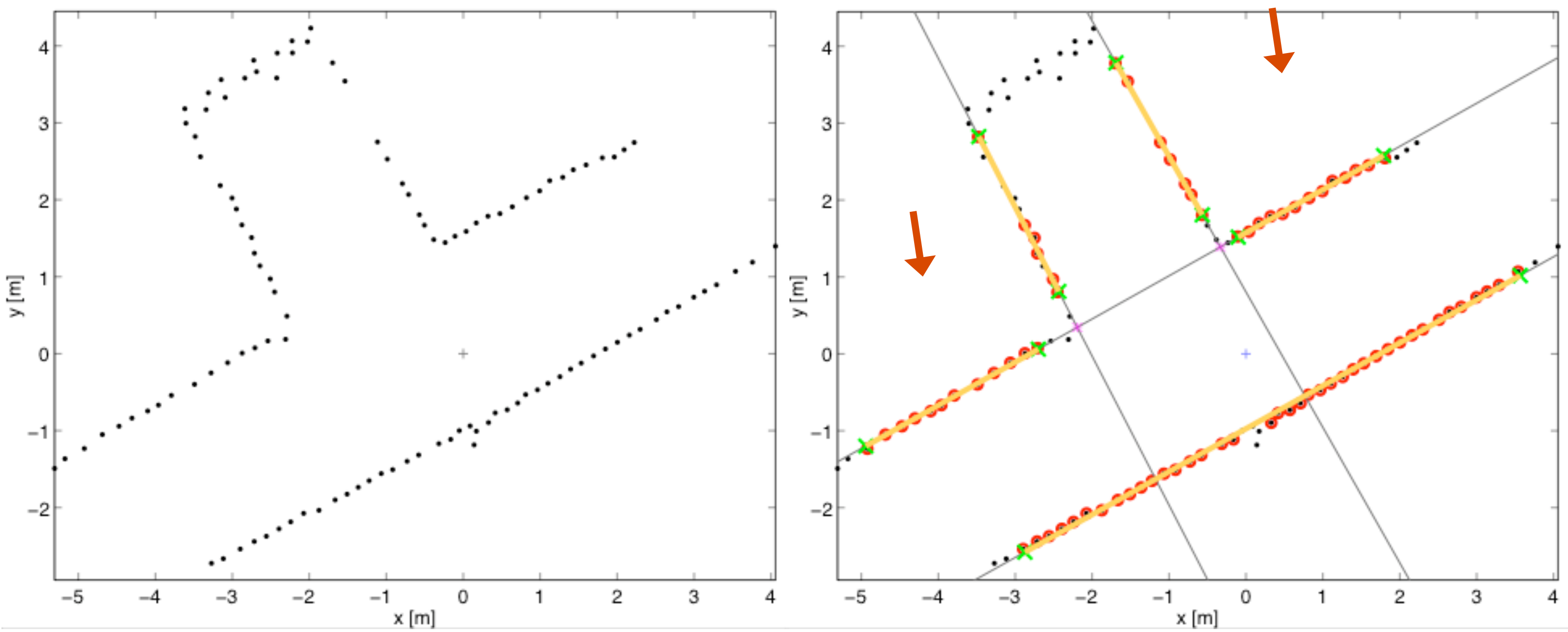
P_S, P_E, P_B :
start-, end-,
break-point

Split only if the break point provides a "better interpretation" in terms of the error sum

[Castellanos 1998]

Split and Merge: Improvements

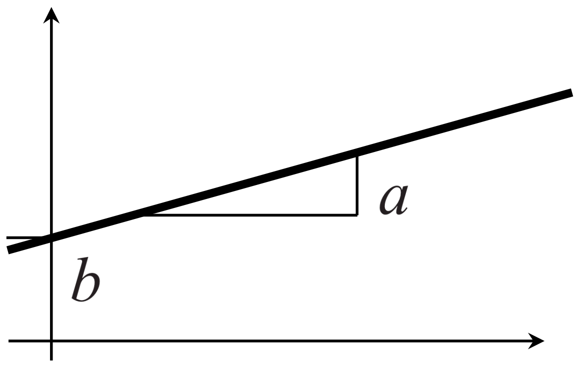
- Merge **non-consecutive** segments as a post-processing step



Line Representation

Choice of the line representation matters!

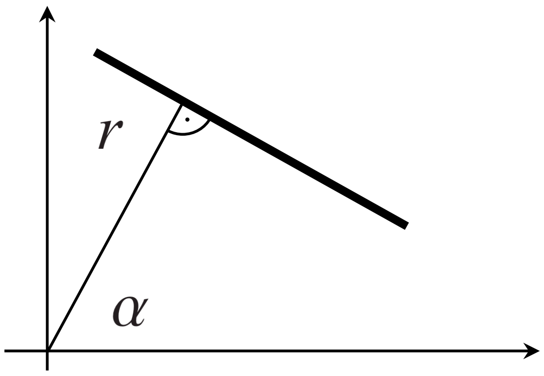
Intercept-Slope



$$y = a x + b$$

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

Hessian model



$$x \cos\alpha + y \sin\alpha - r = 0$$

$$C = \begin{bmatrix} \sigma_\alpha^2 & \sigma_{\alpha r} \\ \sigma_{r\alpha} & \sigma_r^2 \end{bmatrix}$$

Each model has advantages and drawbacks

Fit Expressions

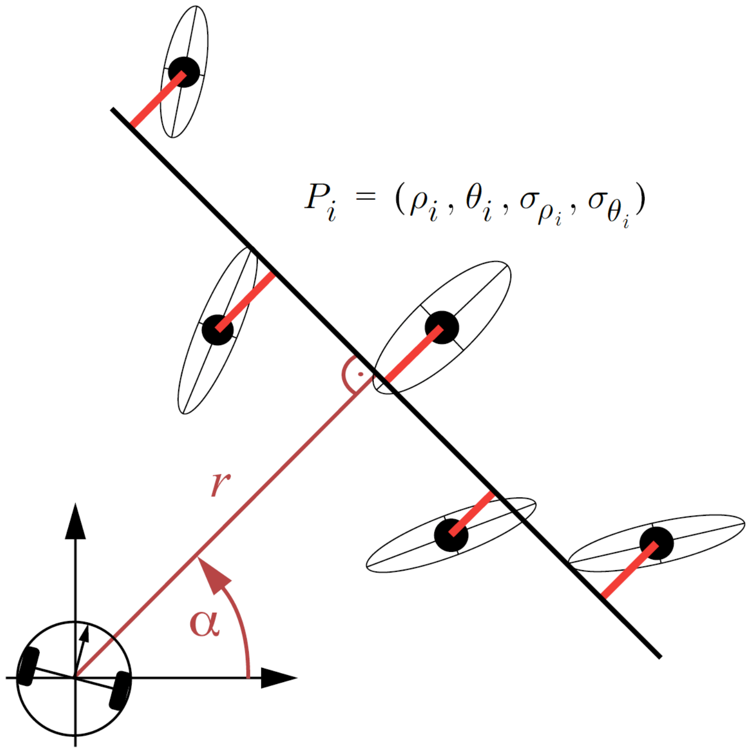
Given:

A set of n points in polar coordinates

Wanted:

Line parameters

$\alpha\,,\,r$



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

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

[Arras 1997]

LSQ Estimation

Regression, Least Squares-Fitting

$$\epsilon_i = x_i \cos\alpha + y_i \sin\alpha - r$$

$$S = \sum_{i=1}^{n} {\epsilon_i}^2$$

Solve the non-linear equation system

$$\frac{\partial S}{\partial \alpha} = 0 \qquad \frac{\partial S}{\partial r} = 0$$

Solution (for points in Cartesian coordinates):

→ 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

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

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

Parametrization trick

Circle Extraction

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

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

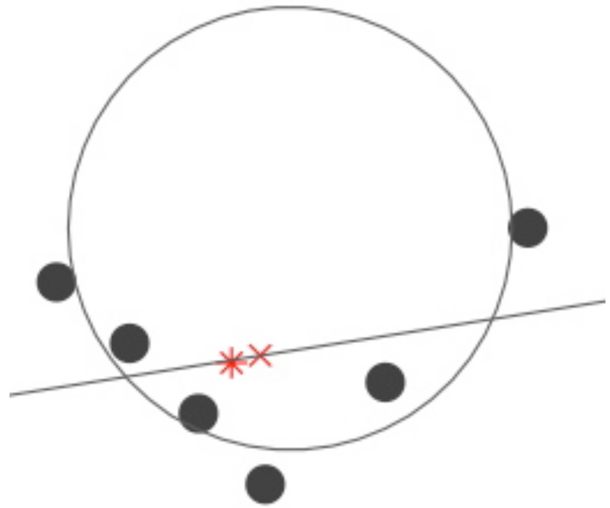
with vector of unknowns

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

Solution via **Pseudo-Inverse**

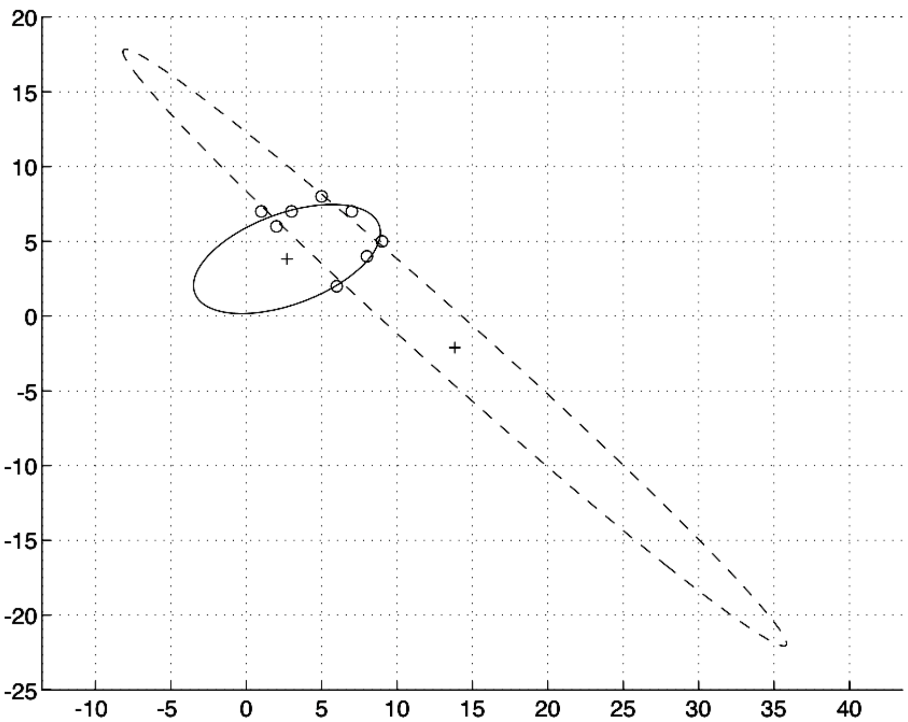
$$x = (A^T A)^{-1} A^T \cdot b$$

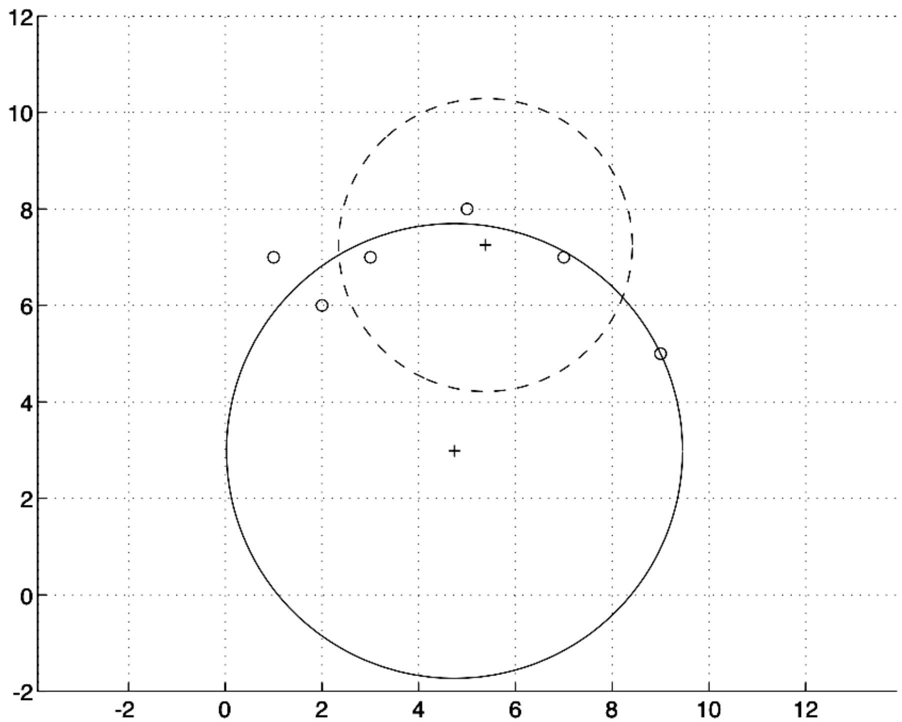
(assuming that *A* has full rank)



Fitting Curves to Points

Attention: Always know the errors that you minimize!





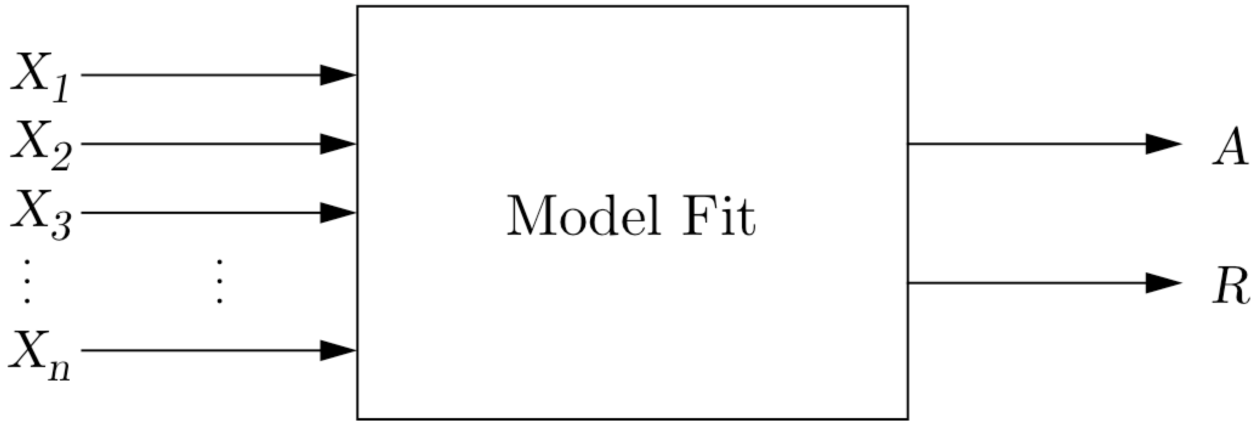
Algebraic versus **geometric** fit solutions [Gander 1994]

LSQ Estimation: Uncertainties?

How does the **input uncertainty** propagate over the fit expressions to the **output**?

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

A, R : Gaussian output random variables



Example: Line Extraction

Wanted: Parameter Covariance Matrix

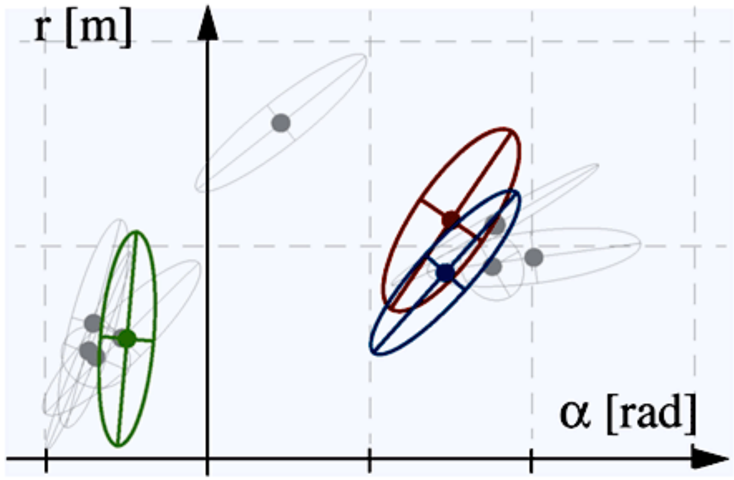
$$C_{AR} = \begin{bmatrix} \sigma_A^2 & \sigma_{AR} \\ \sigma_{AR} & \sigma_R^2 \end{bmatrix}$$

$$C_X = \begin{bmatrix} \sigma_{\rho_1}^2 & 0 & \dots & 0 \\ 0 & \sigma_{\rho_2}^2 & \dots & 0 \\ : & : & & : \\ 0 & 0 & \dots & \sigma_{\rho_n}^2 \end{bmatrix}$$

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***