CEE598 - Visual Sensing for Civil Infrastructure Eng. & Mgmt.

Session 16 – Fitting and Matching

Mani Golparvar-Fard
Department of Civil and Environmental Engineering
3129D, Newmark Civil Engineering Lab
e-mail: mgolpar@illinois.edu

Applications of Model Fitting

- http://www.youtube.com/watch?v=mwzgdbyx8RU
- http://www.youtube.com/watch?v=4DFPg910pvE&feature=related

Paris 26 Giga pixels project

- http://www.youtube.com/user/autopano#p/u/0/GabKQla9Qsw
- http://www.youtube.com/watch?v=PypARCJHJ9I&feature=related

Paris 26 Gigapixels project

- http://www.paris-26-gigapixels.com/HDView/index-en.php

Outline

- Fitting and Matching Methods

- Problem formulation
- Least square methods
- RANSAC
- Hough transforms
- Multi-model fitting
- Fitting helps matching!

- Reading: [HZ] Chapters: 4, 11
  [FP] Chapter: 16

Some slides of this lectures are courtesy of prof S. Savarese, prof. J. Ponce, prof FF Li, prof S. Lazebnik & prof. M. Hebert

Fitting

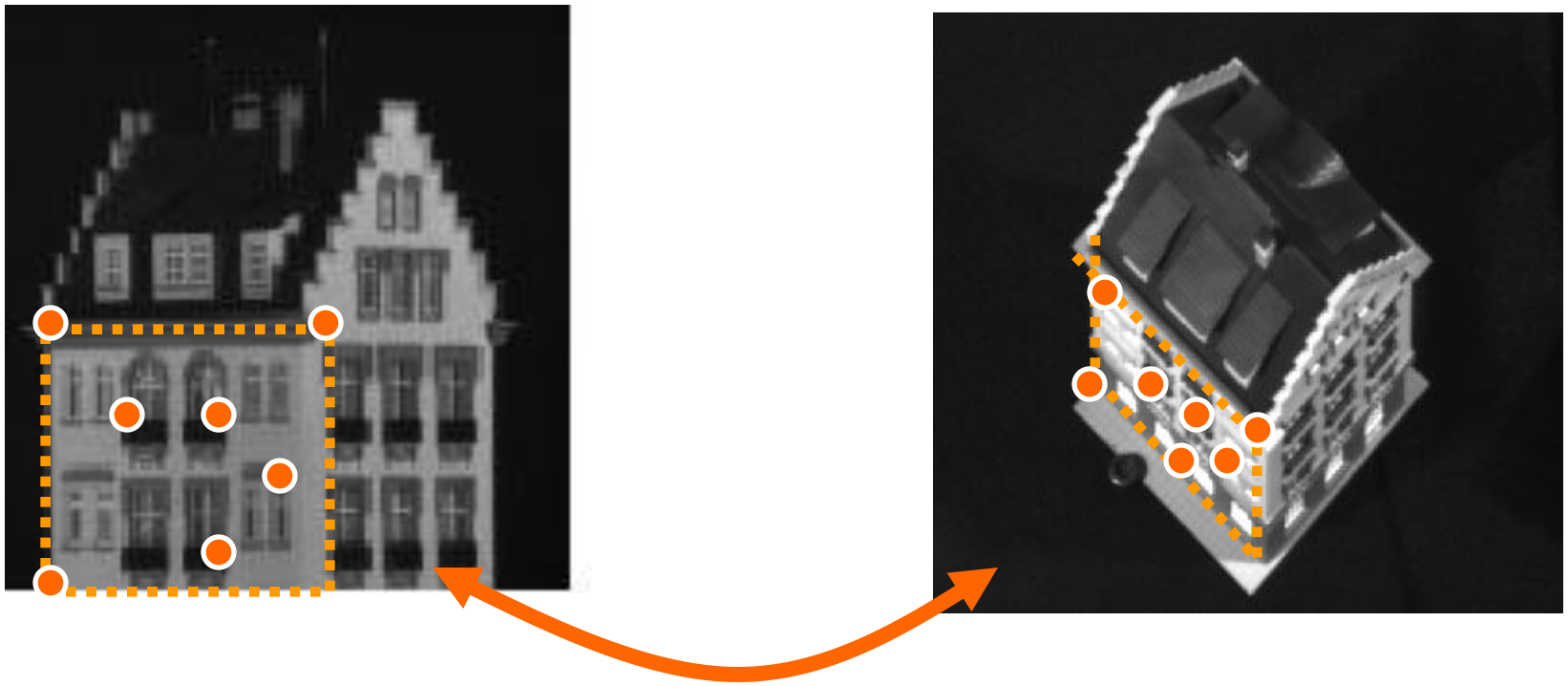
- Goal:

Choose a parametric model to fit a certain quantity from data

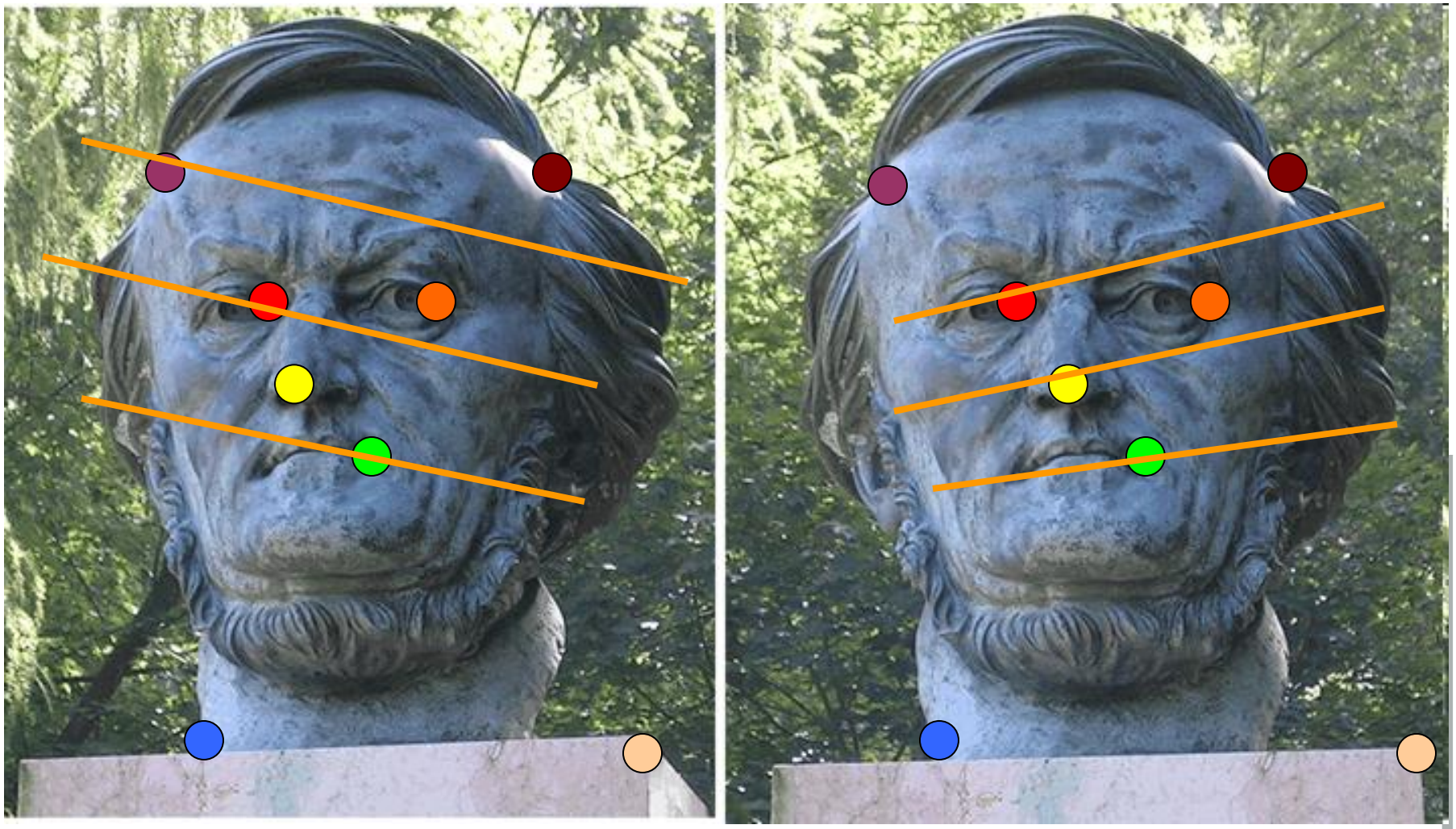
- Lines
- Curves
- Homographic transformation
- Fundamental matrix
- Shape model

Example: Computing vanishing points

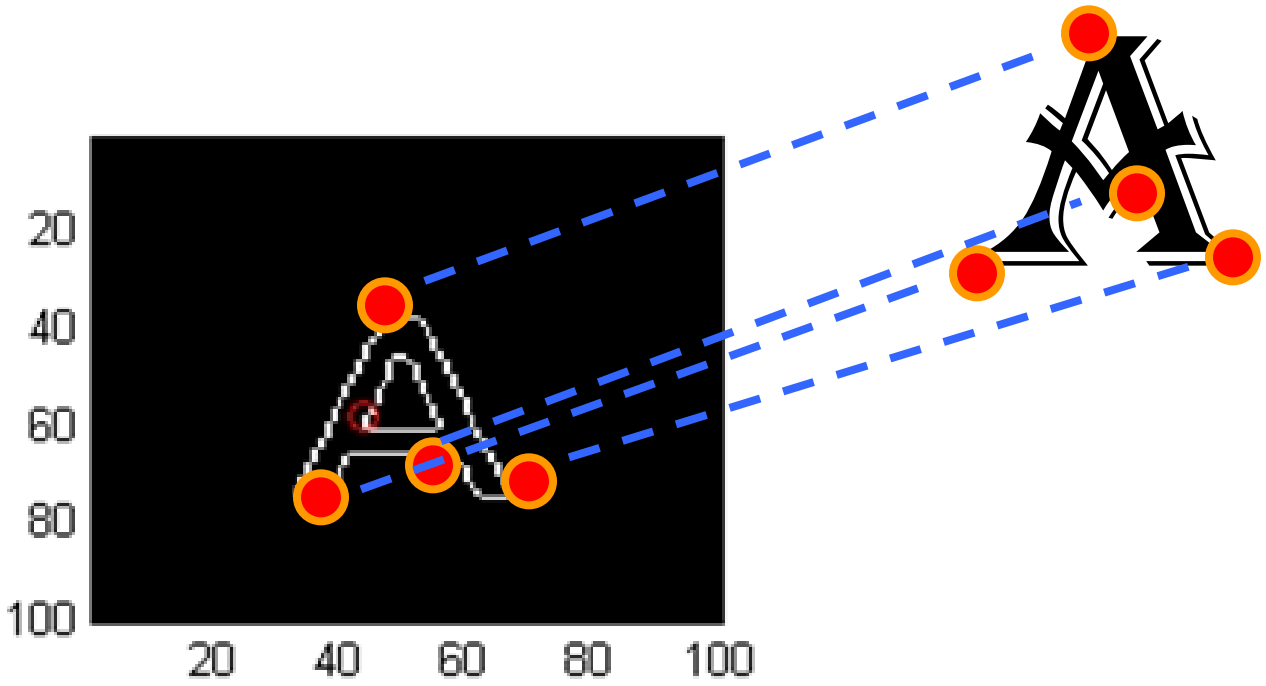
Example: Estimating an homographic transformation



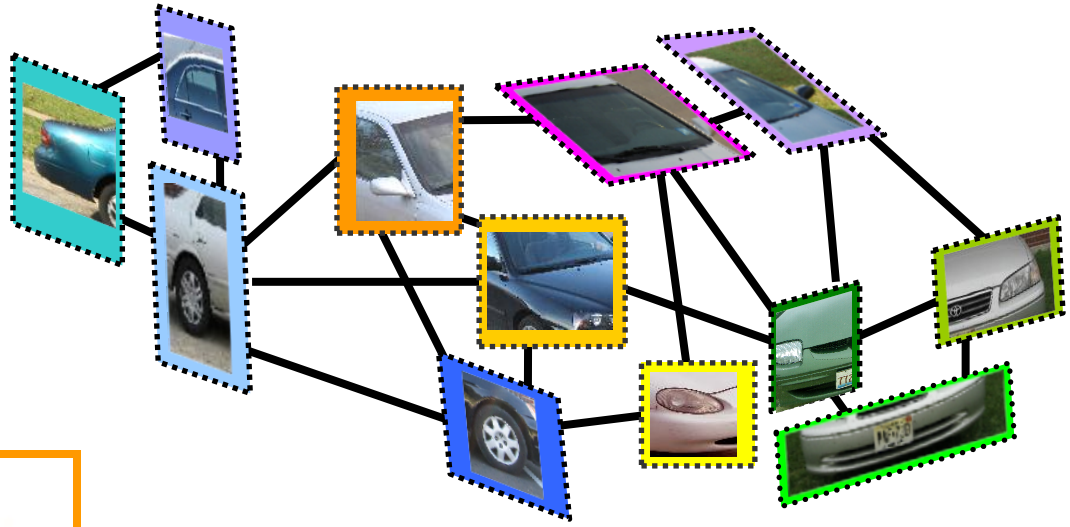
Example: Estimating F

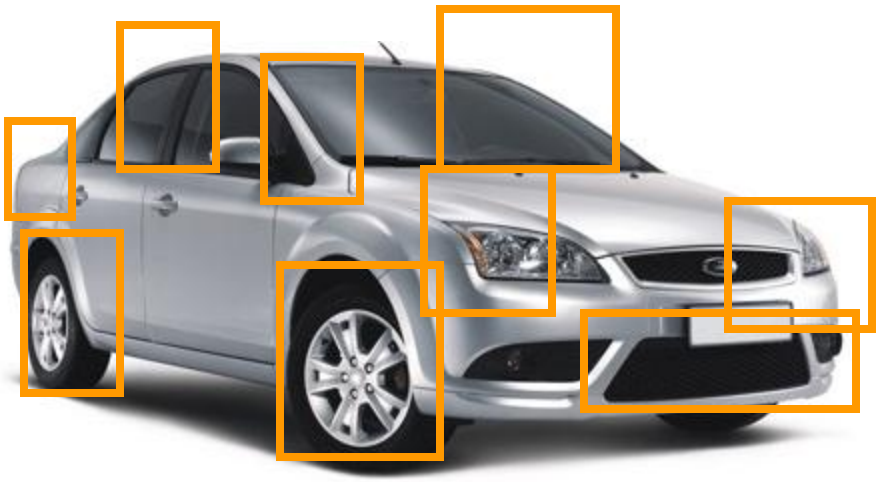


Example: fitting an 2D shape template



Example: fitting a 3D object model





Fitting

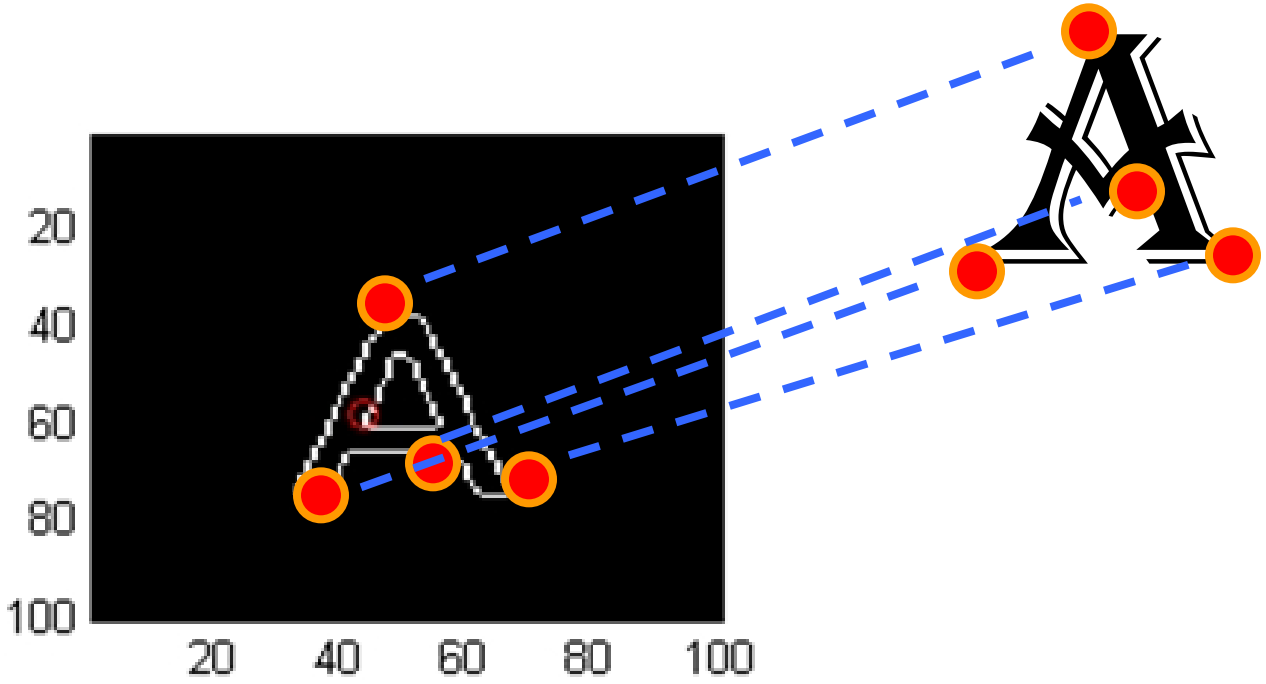
- Goal:

Choose a parametric model to fit a certain quantity from data

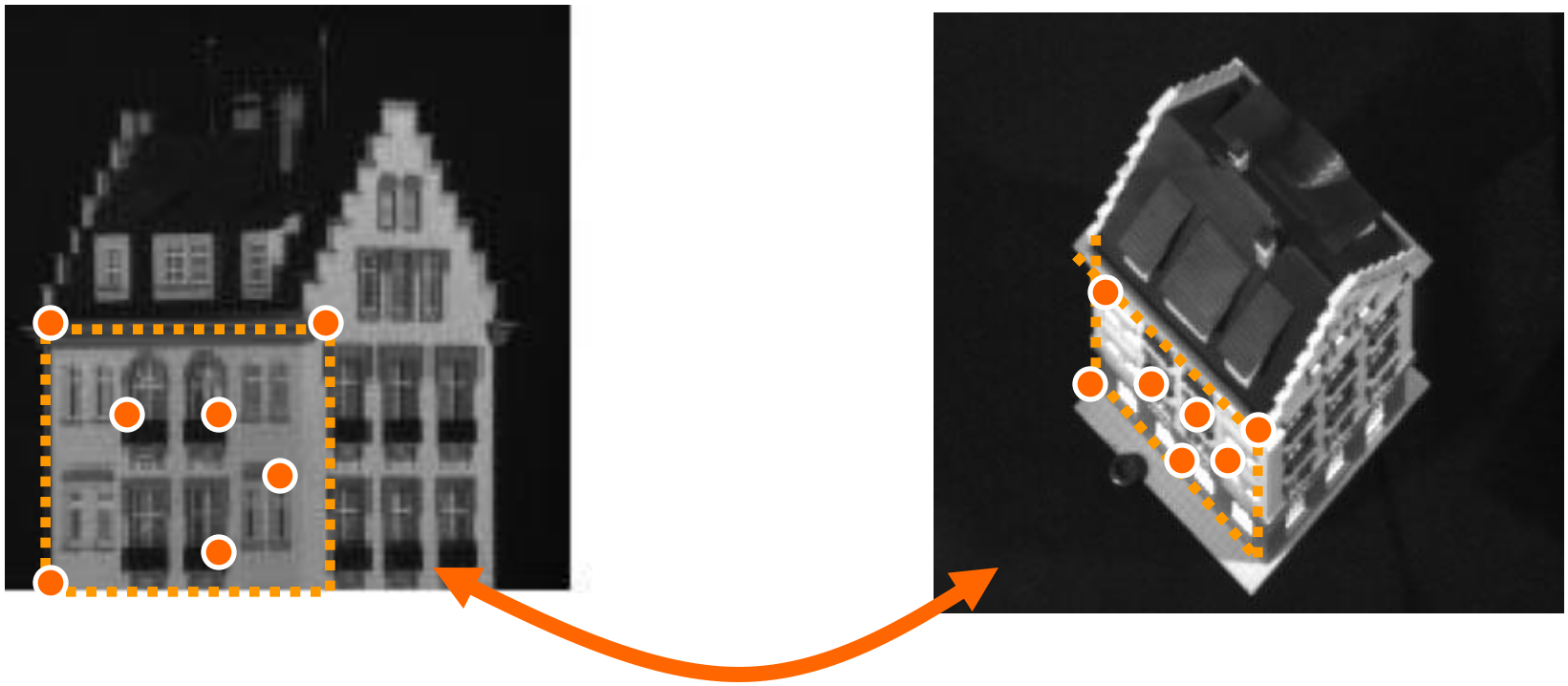
- Critical issues:
  - noisy data
  - outliers
  - missing data

Critical Issues: Noisy Data

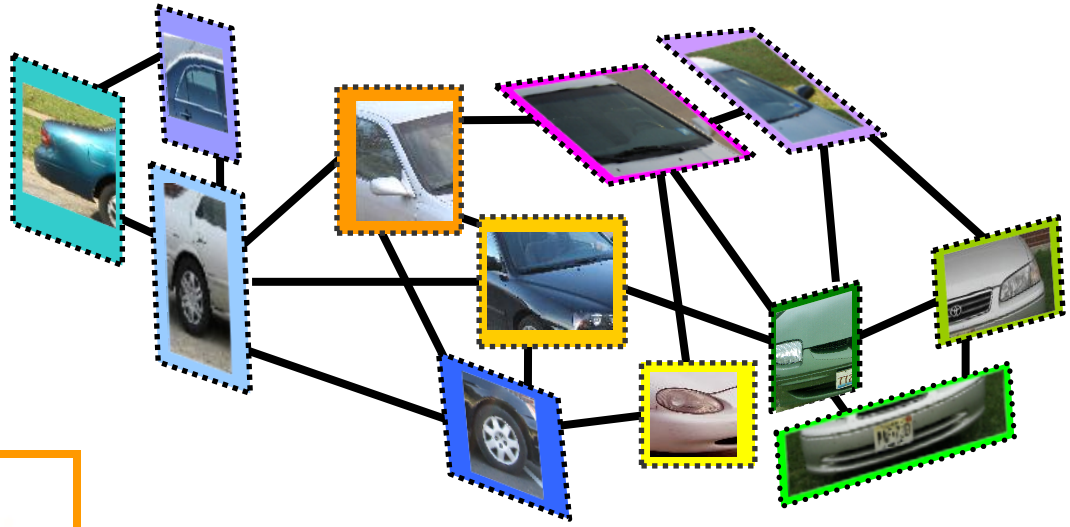
Critical issues: noisy data (intra-class variability)

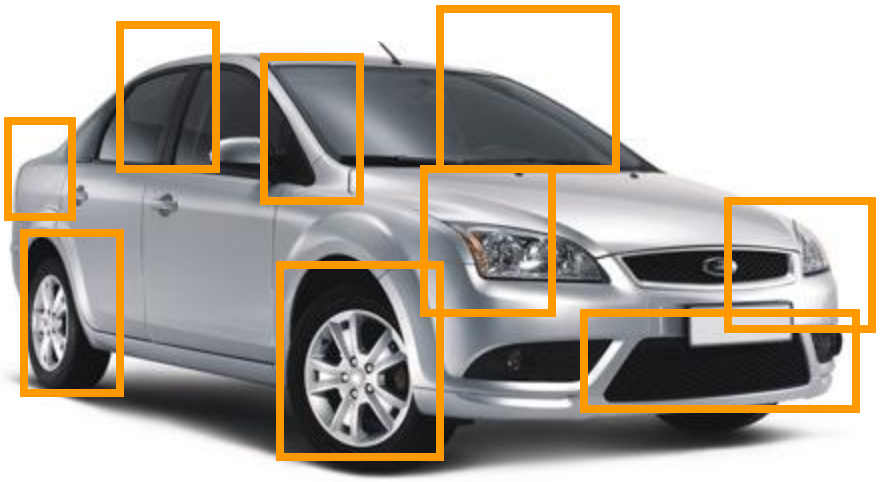


Critical Issues: Outliers



Critical Issues: Missing Data (Occlusions)





Fitting

- Goal:

Choose a parametric model to fit a certain quantity from data

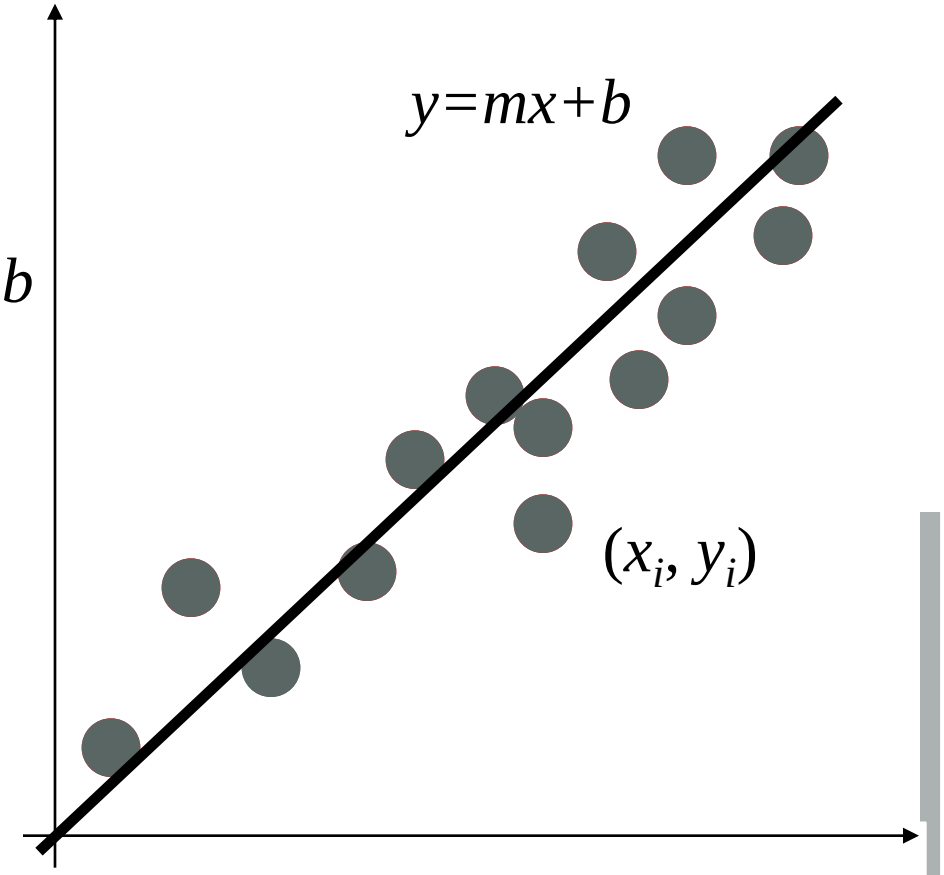
- Techniques:
  - Least square methods
  - RANSAC
  - Hough transform
  - EM (Expectation Maximization)

Least squares methods

- fitting a line -

- Data: $(x_1, y_1), \ldots, (x_n, y_n)$
- Line equation: $y_i = m x_i + b$
- Find (m, b) to minimize

$$E = \sum_{i=1}^{n} (y_i - m x_i - b)^2$$



Least squares methods

- fitting a line -

$$E = \sum_{i=1}^{n} (y_i - mx_i - b)^2$$

$$\mathrm{E} = \sum_{\mathrm{i}=1}^{\mathrm{n}} \left(\mathrm{y_i} - \begin{bmatrix} \mathrm{x_i} & 1 \end{bmatrix} \begin{bmatrix} \mathrm{m} \\ \mathrm{b} \end{bmatrix} \right)^2 = \left\| \begin{bmatrix} \mathrm{y_1} \\ \vdots \\ \mathrm{y_n} \end{bmatrix} - \begin{bmatrix} \mathrm{x_1} & 1 \\ \vdots & \vdots \\ \mathrm{x_n} & 1 \end{bmatrix} \begin{bmatrix} \mathrm{m} \\ \mathrm{b} \end{bmatrix} \right\|^2 = \left\| \mathrm{Y} - \mathrm{XB} \right\|^2$$

$$= (\mathrm{Y} - \mathrm{XB})^{\mathrm{T}} (\mathrm{Y} - \mathrm{XB}) = \mathrm{Y}^{\mathrm{T}}\mathrm{Y} - 2(\mathrm{XB})^{\mathrm{T}}\mathrm{Y} + (\mathrm{XB})^{\mathrm{T}}(\mathrm{XB})$$

$$\frac{dE}{dB} = 2X^T XB - 2X^T Y = 0$$

$$\mathrm{X}^{\mathrm{T}}\mathrm{XB} = \mathrm{X}^{\mathrm{T}}\mathrm{Y}$$

Normal equation

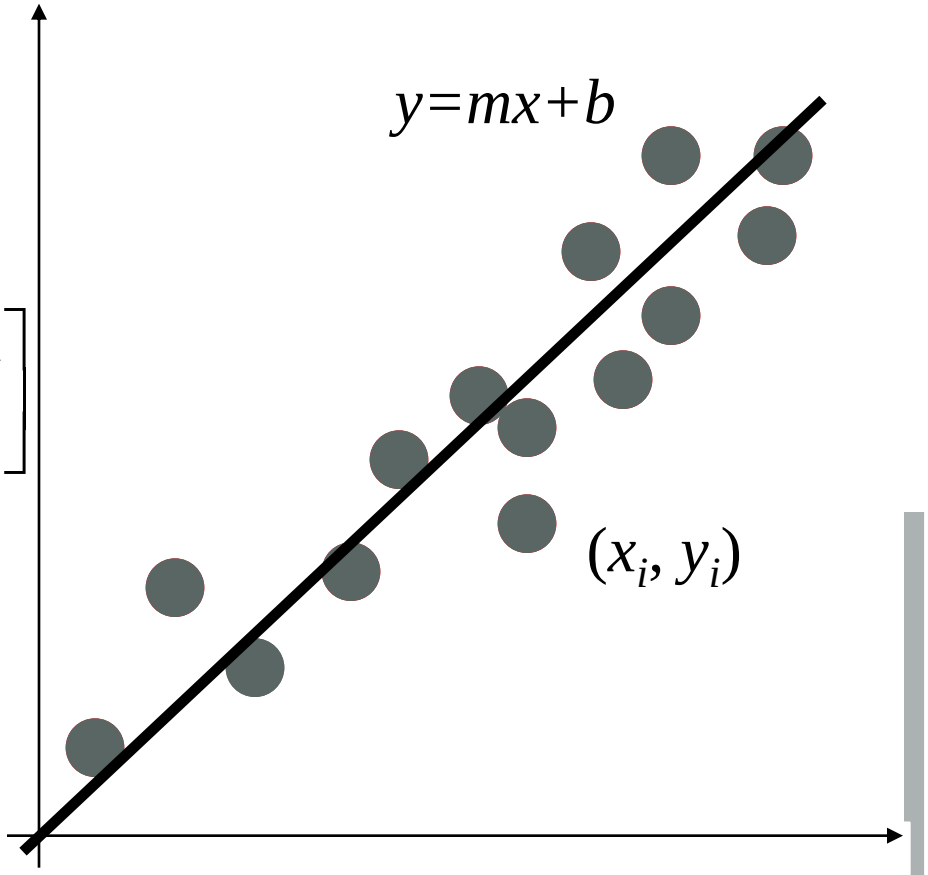
$$\mathrm{B} = \left(\mathrm{X}^{\mathrm{T}}\mathrm{X}\right)^{-1}\mathrm{X}^{\mathrm{T}}\mathrm{Y}$$

Least squares methods

- fitting a line -

$$E = \sum_{i=1}^{n} (y_i - mx_i - b)^2$$

$$B = \left(X^T X\right)^{-1} X^T Y \qquad B = \begin{bmatrix} m \\ b \end{bmatrix}$$



Limitations

- Not rotation-invariant
- Fails completely for vertical lines

Least squares methods

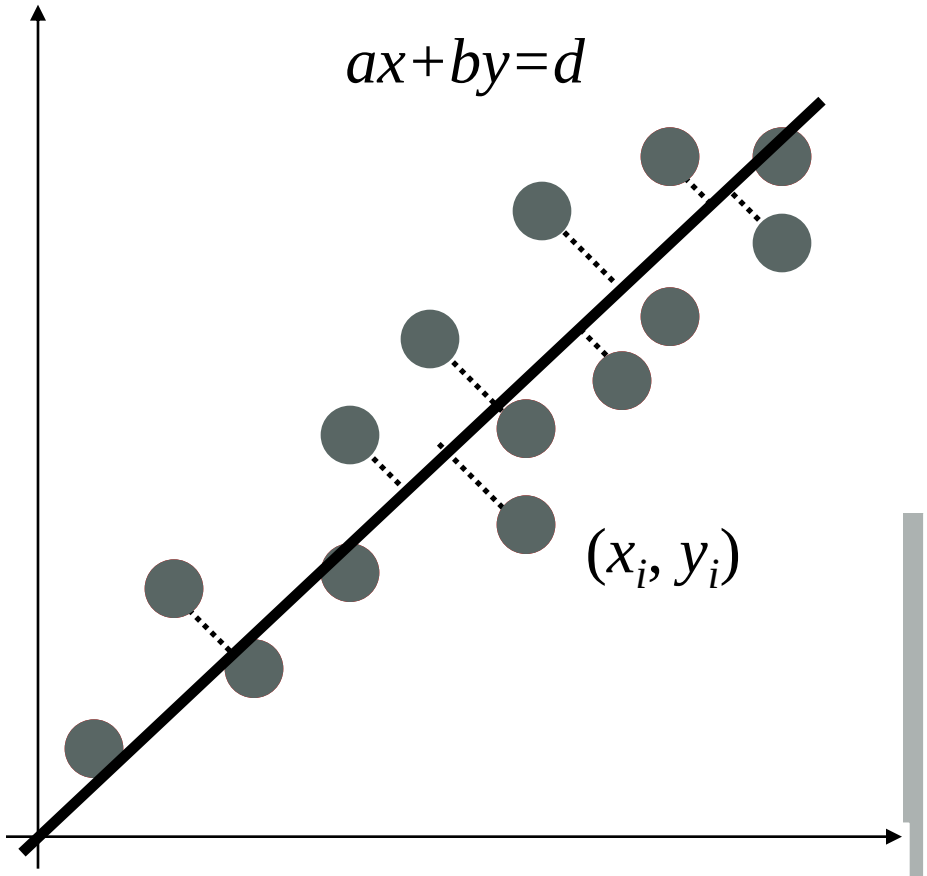
- fitting a line -

- Distance between point (x_n, y_n) and line $ax+by=d$
- Find (a, b, d) to minimize the sum of squared perpendicular distances

$$E = \sum_{i=1}^{n} (ax_i + by_i - d)^2$$

$$(\mathrm{U}^{\mathrm{T}}\mathrm{U})\,\mathrm{N} = 0$$

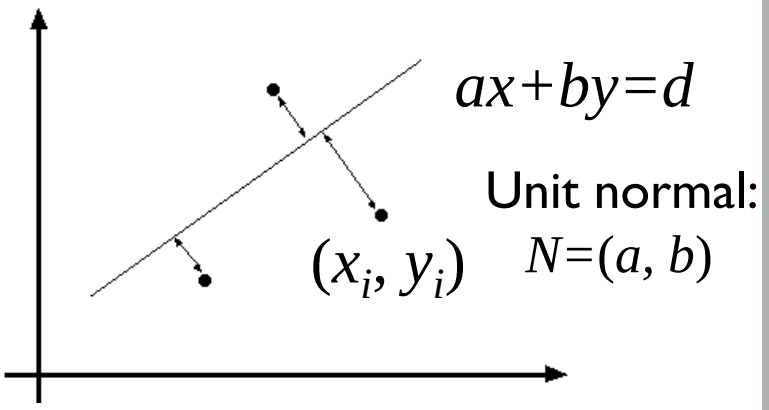
data model parameters



Least squares methods

- Distance between point (x_i, y_i) and line $ax+by=d$ $(a^2+b^2=1)$: $|ax_i + by_i - d|$
- Find (a, b, d) to minimize the sum of squared perpendicular distances

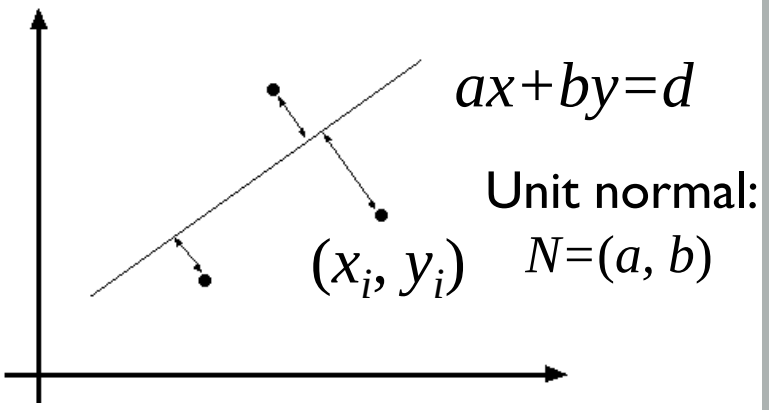
$$E = \sum_{i=1}^{n} (ax_i + by_i - d)^2$$



Slide: S. Lazebnik

Least squares methods

- Distance between point (x_i, y_i) and line $ax+by=d$ $(a^2+b^2=1)$: $|ax_i + by_i - d|$
- Find (a, b, d) to minimize the sum of squared perpendicular distances



$$E = \sum_{i=1}^{n} (ax_i + by_i - d)^2$$

$$\frac{\partial E}{\partial d} = \sum_{i=1}^{n} -2(ax_i + by_i - d) = 0$$

$$d = \frac{a}{n}\sum_{i=1}^{n} x_i + \frac{b}{n}\sum_{i=1}^{n} x_i = a\bar{x} + b\bar{y}$$

$$E = \sum_{i=1}^{n} (a(x_i - \bar{x}) + b(y_i - \bar{y}))^2 = \left\| \begin{bmatrix} x_1 - \bar{x} & y_1 - \bar{y} \\ \vdots & \vdots \\ x_n - \bar{x} & y_n - \bar{y} \end{bmatrix} \begin{bmatrix} a \\ b \end{bmatrix} \right\|^2 = (UN)^T (UN)$$

$$\frac{dE}{dN} = 2(U^T U)N = 0$$

Solution to $(U^TU)N = 0$, subject to $||N||^2 = 1$: eigenvector of U^TU associated with the smallest eigenvalue (least squares solution to *homogeneous linear system* $UN = 0$)

Slide: S. Lazebnik

Total least squares

$$U = \begin{bmatrix} x_1 - \bar{x} & y_1 - \bar{y} \\ \vdots & \vdots \\ x_n - \bar{x} & y_n - \bar{y} \end{bmatrix} \qquad U^T U = \begin{bmatrix} \sum_{i=1}^{n} (x_i - \bar{x})^2 & \sum_{i=1}^{n} (x_i - \bar{x})(y_i - \bar{y}) \\ \sum_{i=1}^{n} (x_i - \bar{x})(y_i - \bar{y}) & \sum_{i=1}^{n} (y_i - \bar{y})^2 \end{bmatrix}$$

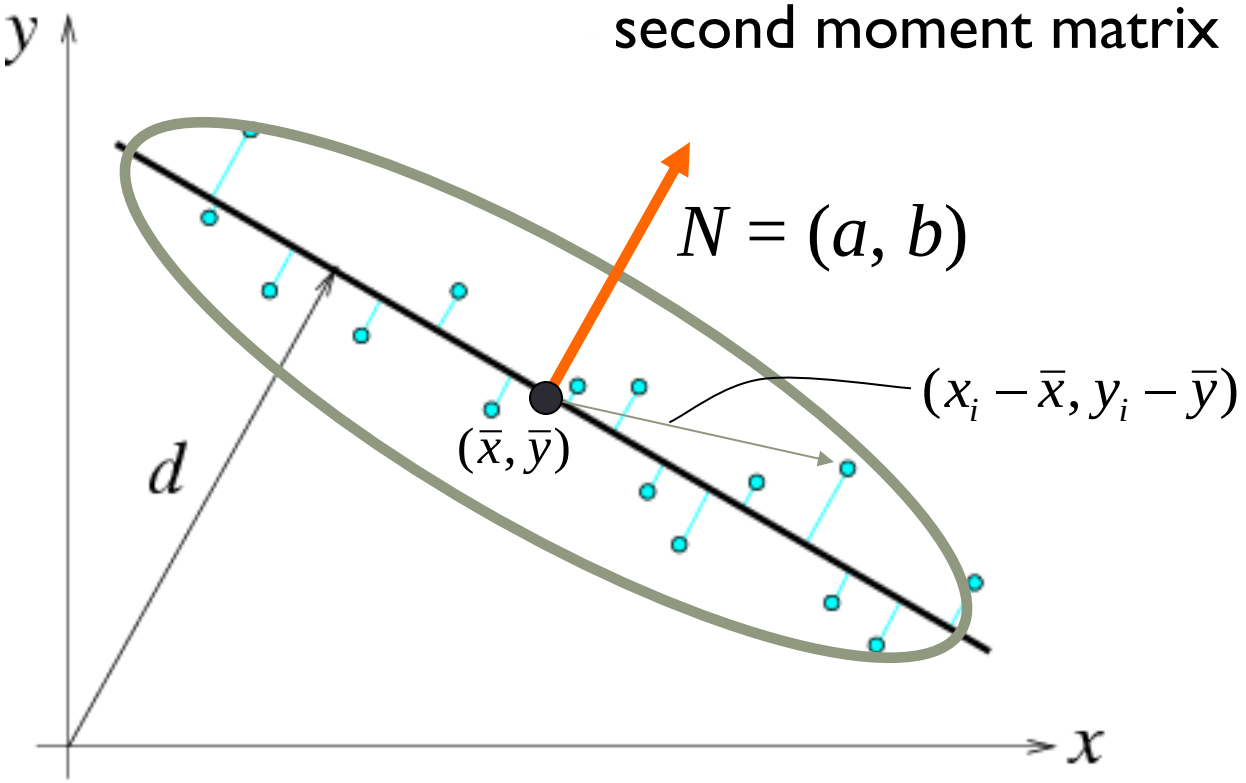
second moment matrix

Slide: S. Lazebnik

Total least squares

$$U = \begin{bmatrix} x_1 - \bar{x} & y_1 - \bar{y} \\ \vdots & \vdots \\ x_n - \bar{x} & y_n - \bar{y} \end{bmatrix} \qquad U^T U = \begin{bmatrix} \sum_{i=1}^{n} (x_i - \bar{x})^2 & \sum_{i=1}^{n} (x_i - \bar{x})(y_i - \bar{y}) \\ \sum_{i=1}^{n} (x_i - \bar{x})(y_i - \bar{y}) & \sum_{i=1}^{n} (y_i - \bar{y})^2 \end{bmatrix}$$

second moment matrix



Slide: S. Lazebnik

Least squares methods

- fitting a line -

$$A\,h = 0$$

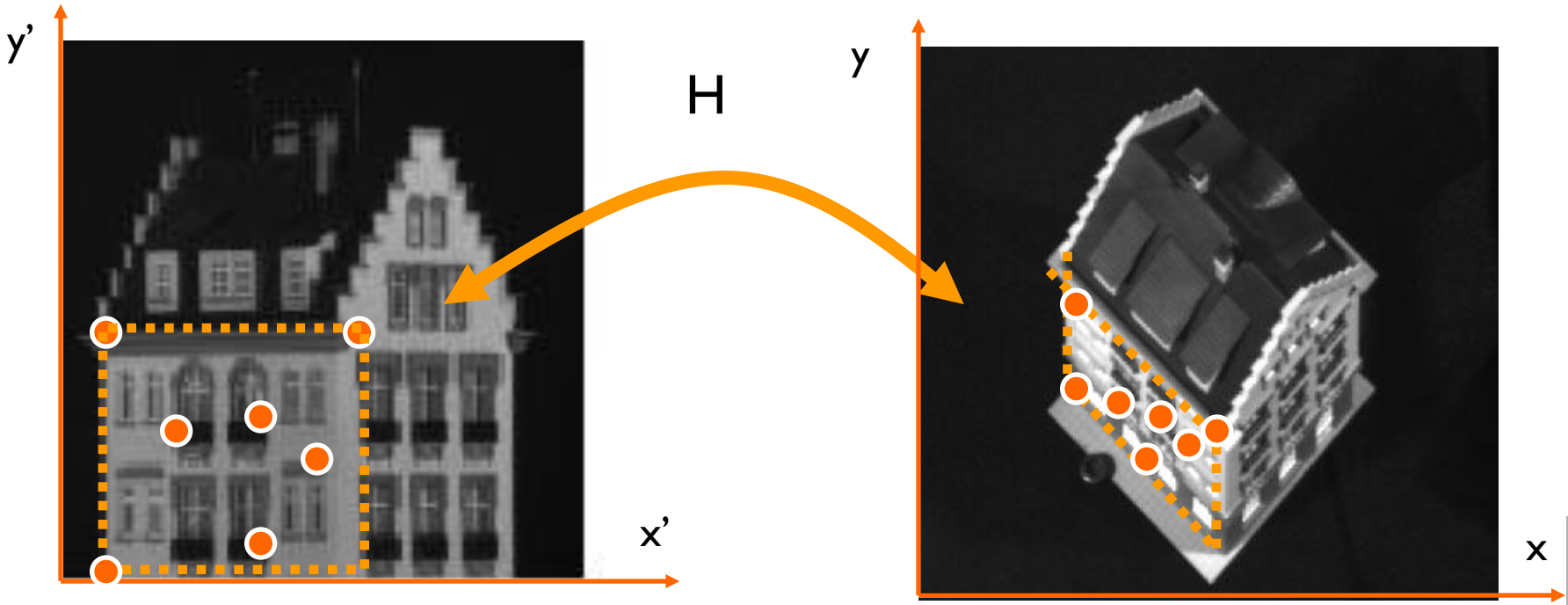
$$\text{Minimize} \| A\,h \| \quad \text{subject to} \quad \| h \| = 1$$

$$A = UDV^T$$

$$h = \text{last column of } V$$

Least squares methods

- fitting an homography -



$$\begin{bmatrix} wx' \\ wy' \\ w \end{bmatrix} = \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

Least squares methods

- fitting an homography -

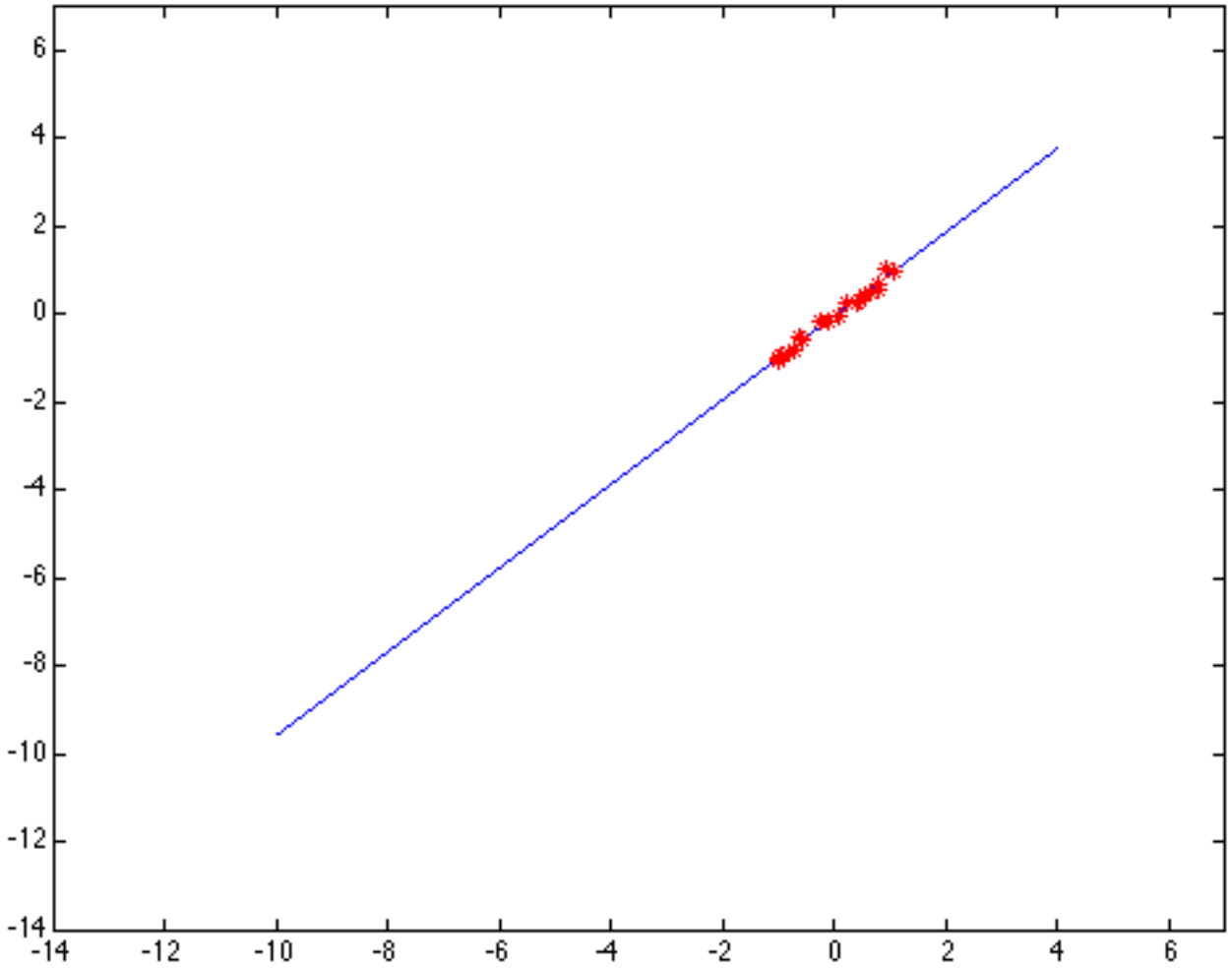
$$
\begin{aligned}
h_{11}x + h_{12}y + h_{13} - h_{31}xx' - h_{32}yx' - x' &= 0 \\
h_{21}x + h_{22}y + h_{23} - h_{31}xy' - h_{32}yy' - y' &= 0
\end{aligned}
$$

From n>=4 corresponding points:

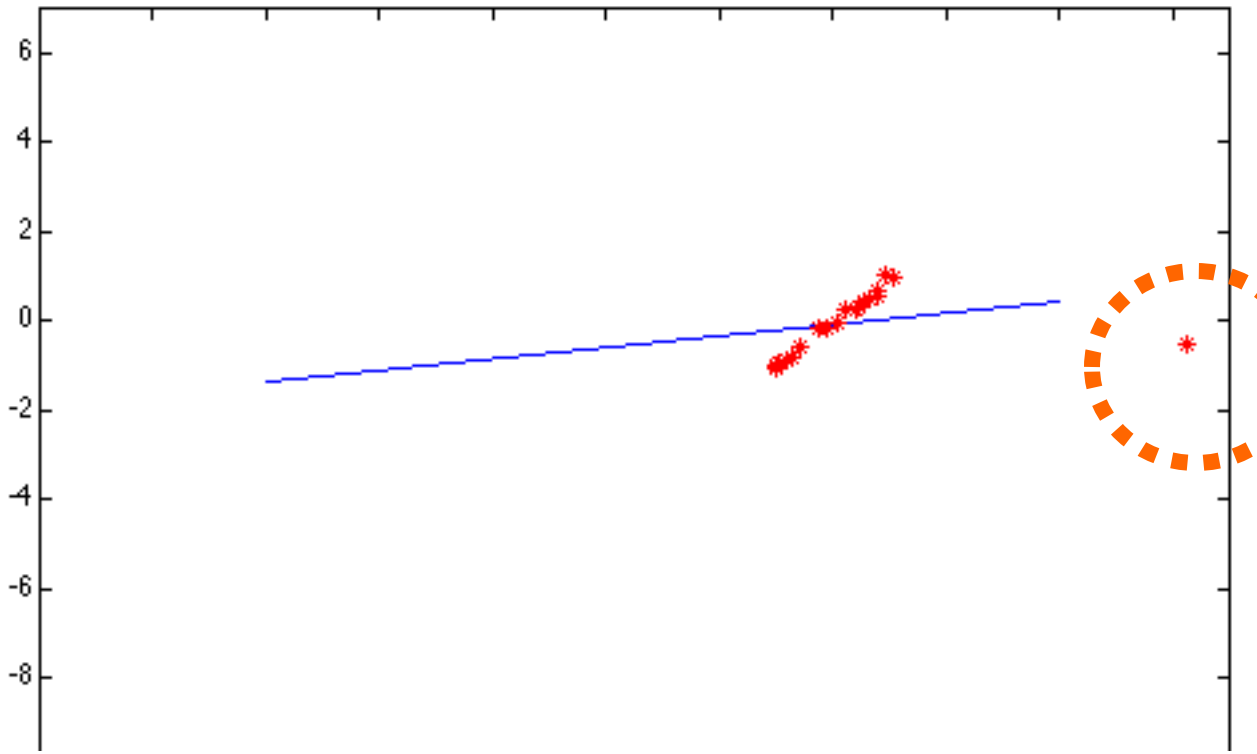
$$A\,h = 0$$

$$
\begin{pmatrix}
x_1 & y_1 & 1 & 0 & 0 & 0 & -x_1x'_1 & -y_1x'_1 & -x'_1 \\
0 & 0 & 0 & x_1 & y_1 & 1 & -x_1y'_1 & -y_1y'_1 & -y'_1 \\
x_2 & y_2 & 1 & 0 & 0 & 0 & -x_2x'_2 & -y_2x'_2 & -x'_2 \\
0 & 0 & 0 & x_2 & y_2 & 1 & -x_2y'_2 & -y_2y'_2 & -y'_2 \\
\vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\
x_n & y_n & 1 & 0 & 0 & 0 & -x_nx'_n & -y_nx'_n & -x'_n \\
0 & 0 & 0 & x_n & y_n & 1 & -x_ny'_n & -y_ny'_n & -y'_n
\end{pmatrix}
\begin{bmatrix}
h_{1,1} \\ h_{1,2} \\ \vdots \\ h_{3,3}
\end{bmatrix}
= 0
$$

Least squares: Robustness to noise

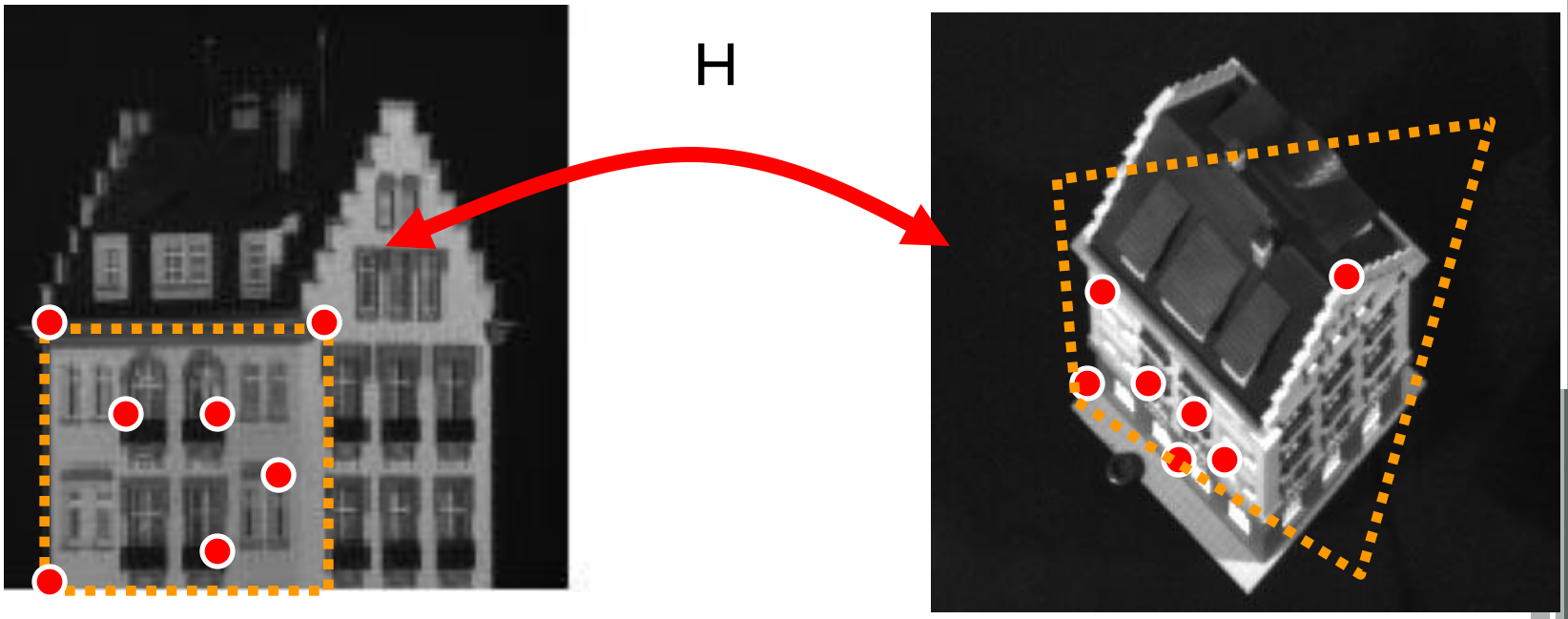


Least squares: Robustness to noise



Problem: squared error heavily penalizes outliers

Critical issues: outliers



CONCLUSION: Least square is not robust w.r.t. outliers

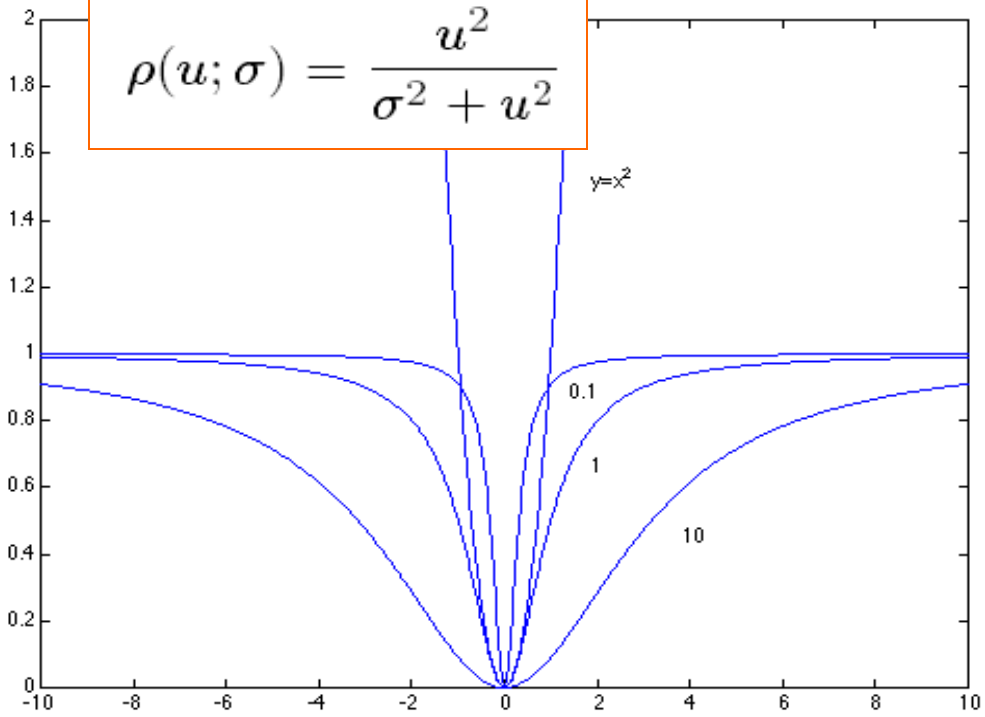
Least squares: Robust estimators

- General approach:

minimize $\sum_i \rho(u_i(x_i, \theta); \sigma)$ $\quad u = \sum_{i=1}^{n} (y_i - mx_i - b)^2$

- $u_i(x_i, \theta)$ – residual of ith point w.r.t. model parameters ϑ
  ρ – robust function with scale parameter σ

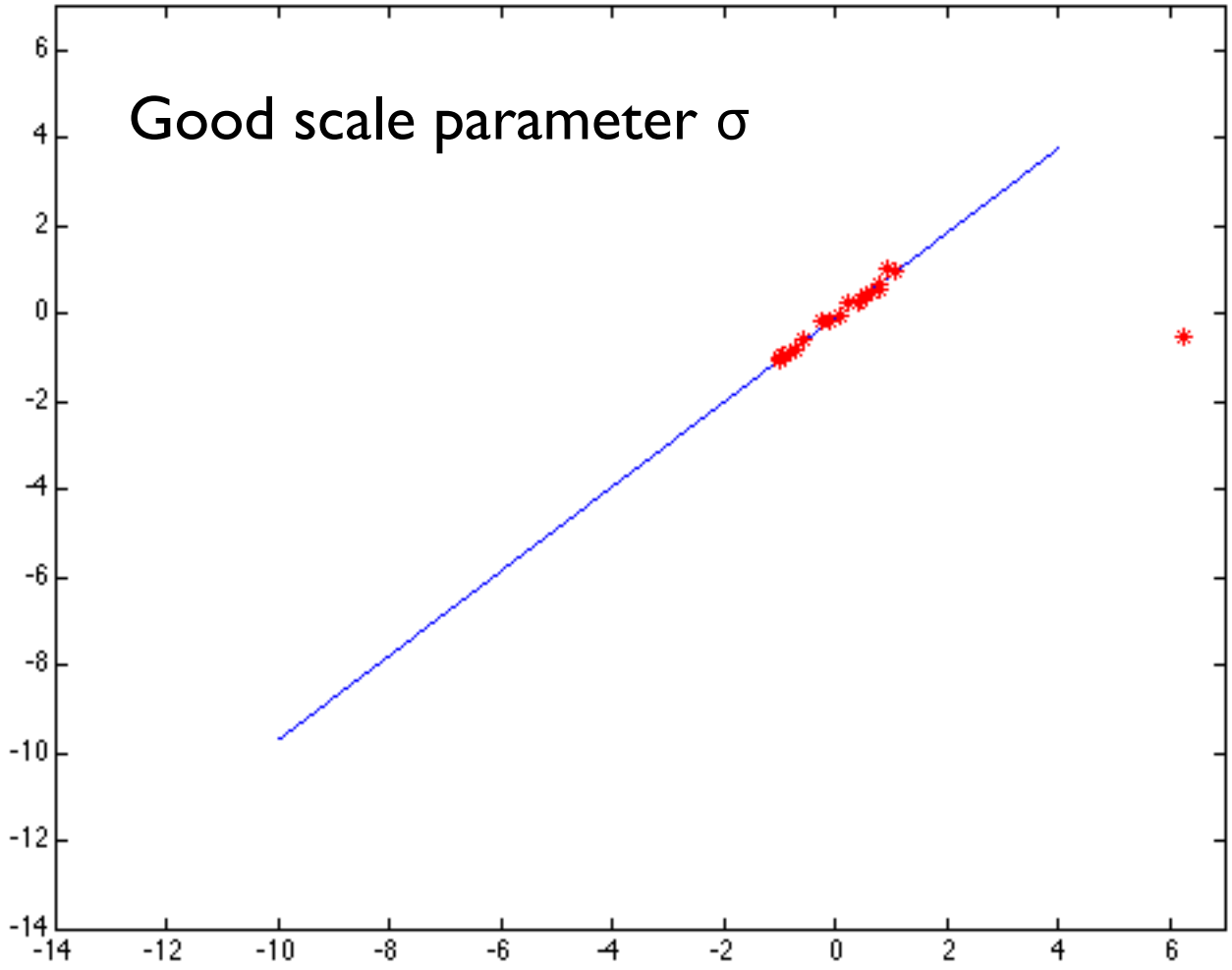
$$\rho(u;\sigma) = \frac{u^2}{\sigma^2 + u^2}$$



The robust function ρ

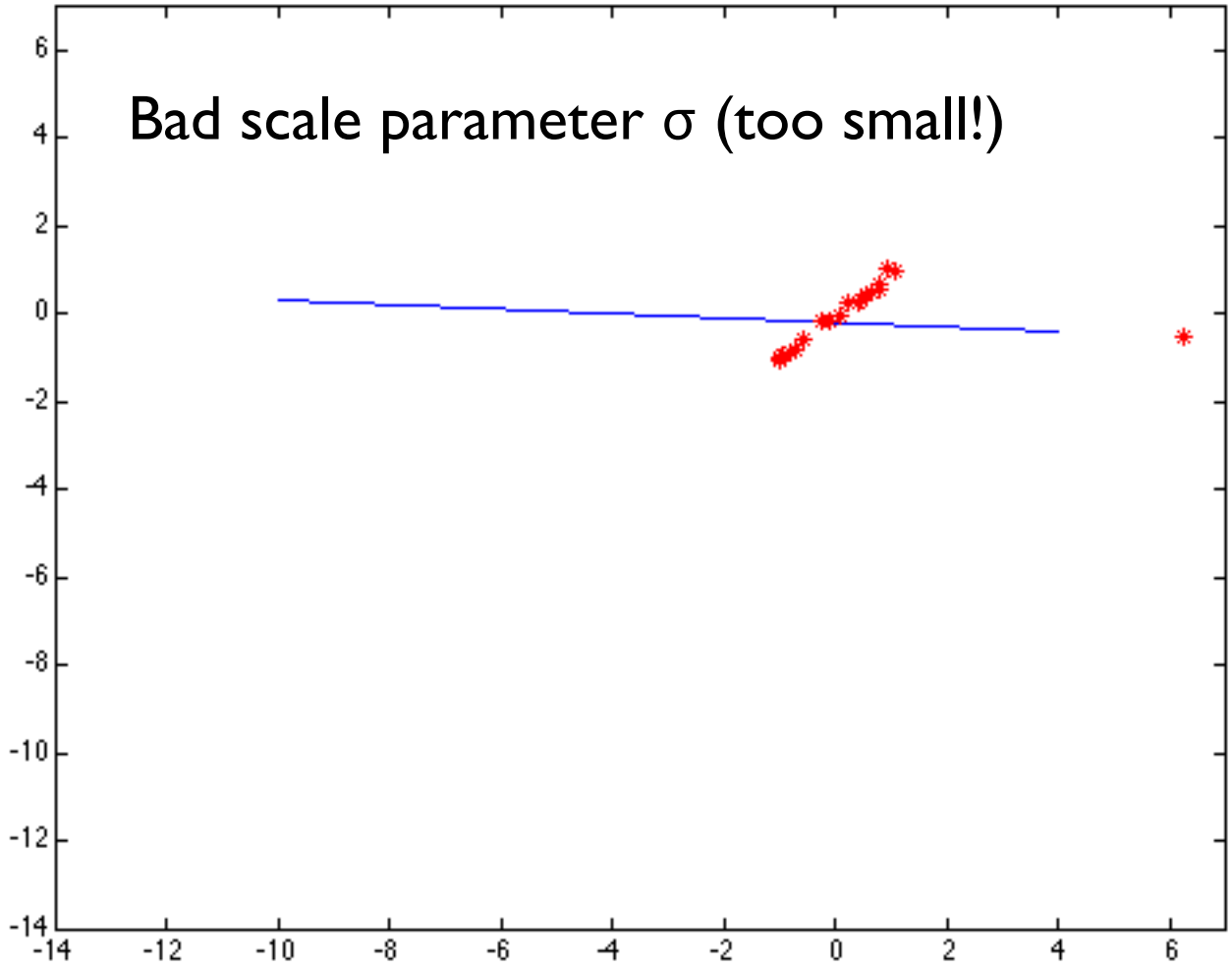
- favors a configuration with small residuals
- Penalizes large residuals

Least squares: Robust estimators

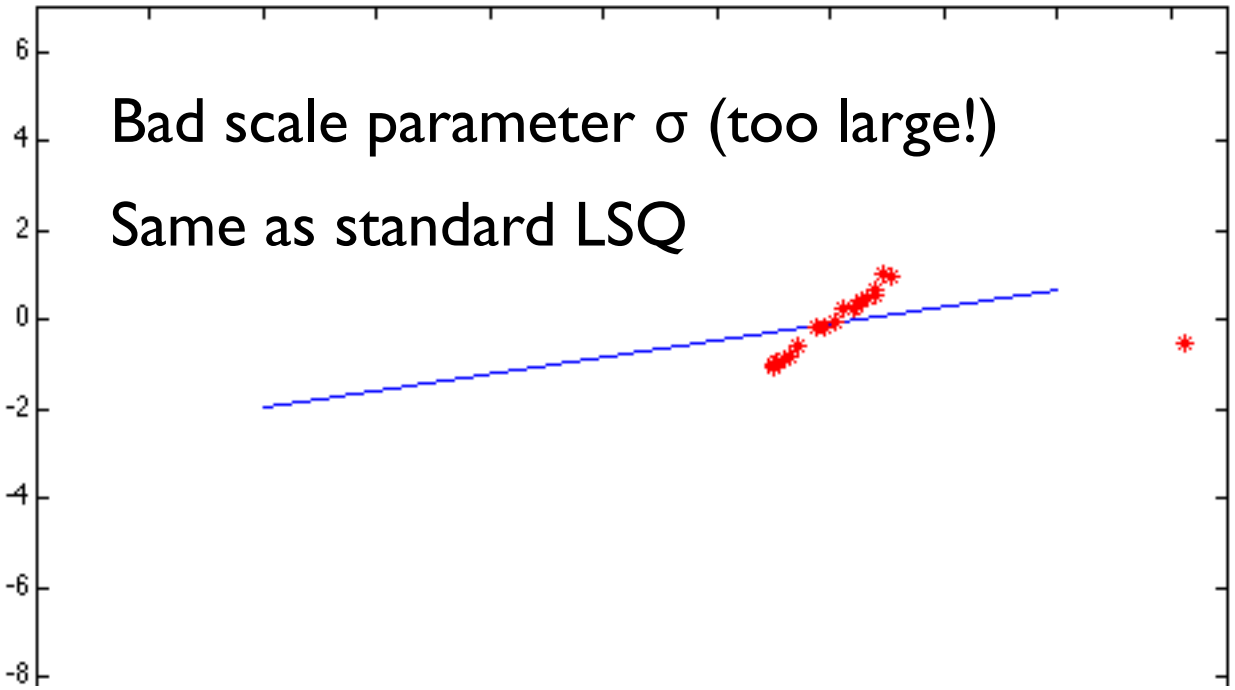


The effect of the outlier is eliminated

Least squares: Robust estimators



Least squares: Robust estimators



•CONCLUSION: Robust estimator useful if prior info about the distribution of points is known

• Robust fitting is a nonlinear optimization problem (iterative solution)

• Least squares solution provides good initial condition

Fitting

- Goal:

Choose a parametric model to fit a certain quantity from data

- Techniques:
  - Least square methods
  - RANSAC
  - Hough transform
  - EM (Expectation Maximization)

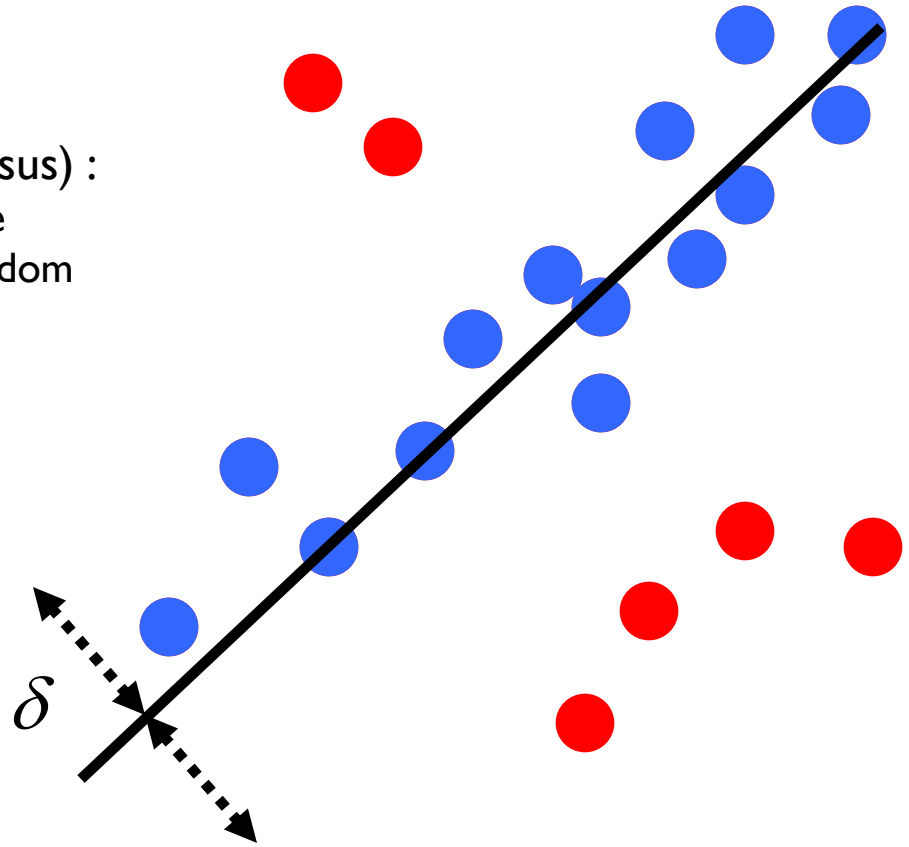
Basic philosophy
(voting scheme)

- Data elements are used to vote for one (or multiple) models
- Robust to outliers and missing data
- Assumption1: Noise features will **not** vote **consistently** for any single model ("few" outliers)
- Assumption2: there are enough features to agree on a good model ("few" missing data)

RANSAC

(RANdom SAmple Consensus) :
Learning technique to estimate parameters of a model by random sampling of observed data

Fischler & Bolles in '81.



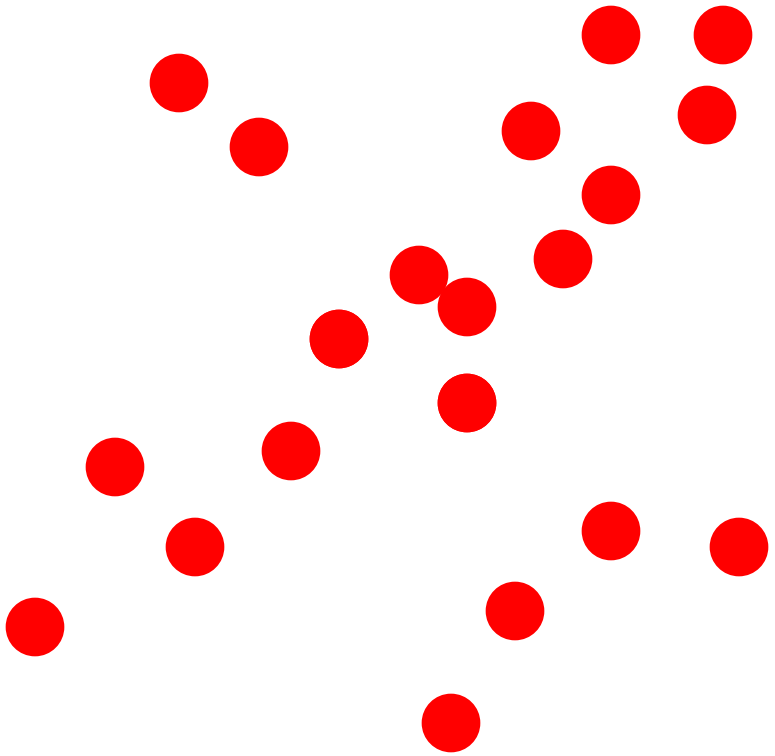
$$\pi : \boldsymbol{I} \rightarrow \{\boldsymbol{P}, \boldsymbol{O}\} \qquad \min_{\pi} |\boldsymbol{O}|$$

such that:

$$\boldsymbol{f}(\boldsymbol{P}, \beta) < \delta \qquad \boldsymbol{f}(\boldsymbol{P}, \beta) = \left\| \beta - \left(\boldsymbol{P}^{\boldsymbol{T}} \boldsymbol{P}\right)^{-1} \boldsymbol{P}^{\boldsymbol{T}} \right\|$$

Model parameters

RANSAC



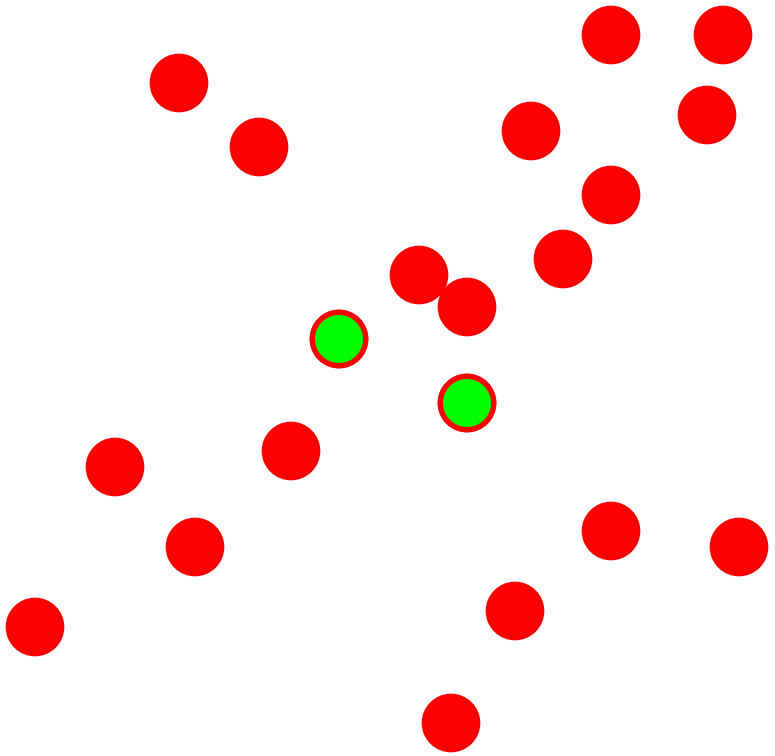
Sample set = set of points in 2D

Algorithm:

1. Select random sample of minimum required size to fit model
2. Compute a putative model from sample set
3. Compute the set of inliers to this model from whole data set

Repeat 1-3 until model with the most inliers over all samples is found

RANSAC



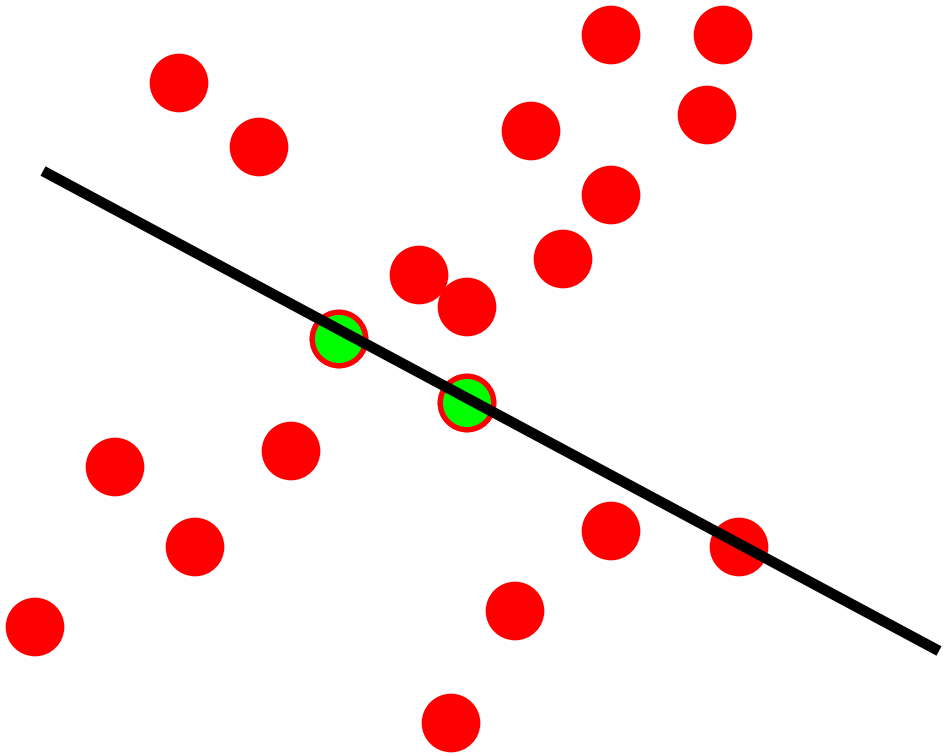
Sample set = set of points in 2D

Algorithm:

1. Select random sample of minimum required size to fit model [?] =[2]
2. Compute a putative model from sample set
3. Compute the set of inliers to this model from whole data set

Repeat 1-3 until model with the most inliers over all samples is found

RANSAC



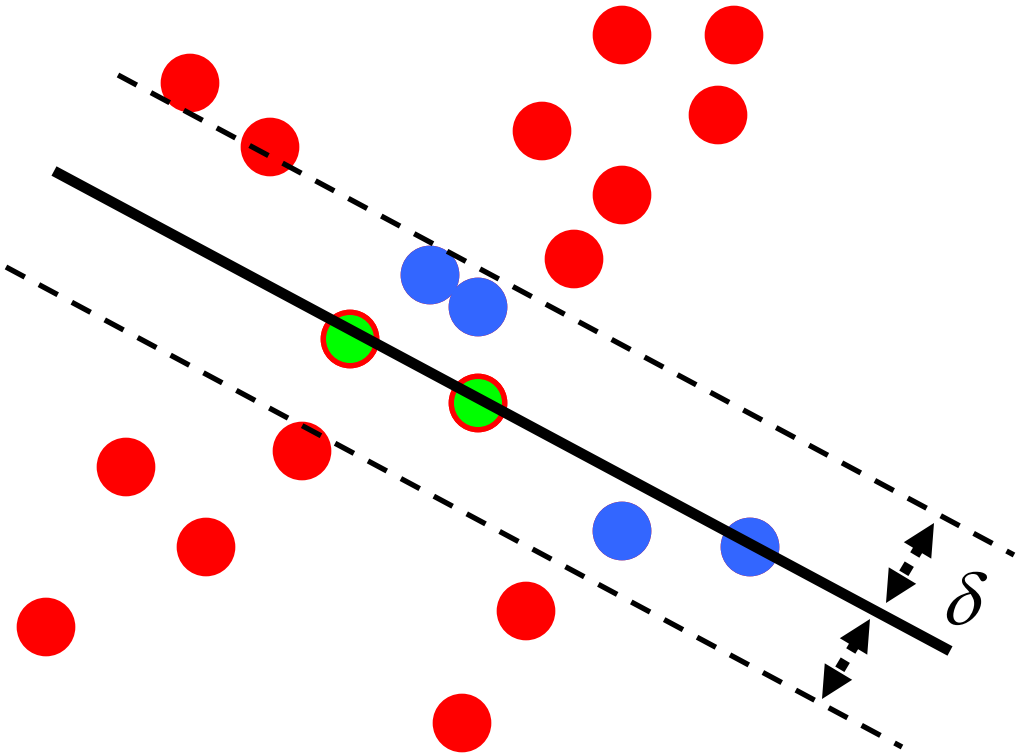
Sample set = set of points in 2D

Algorithm:

1. Select random sample of minimum required size to fit model [?] =[2]
2. Compute a putative model from sample set
3. Compute the set of inliers to this model from whole data set

Repeat 1-3 until model with the most inliers over all samples is found

RANSAC



Sample set = set of points in 2D

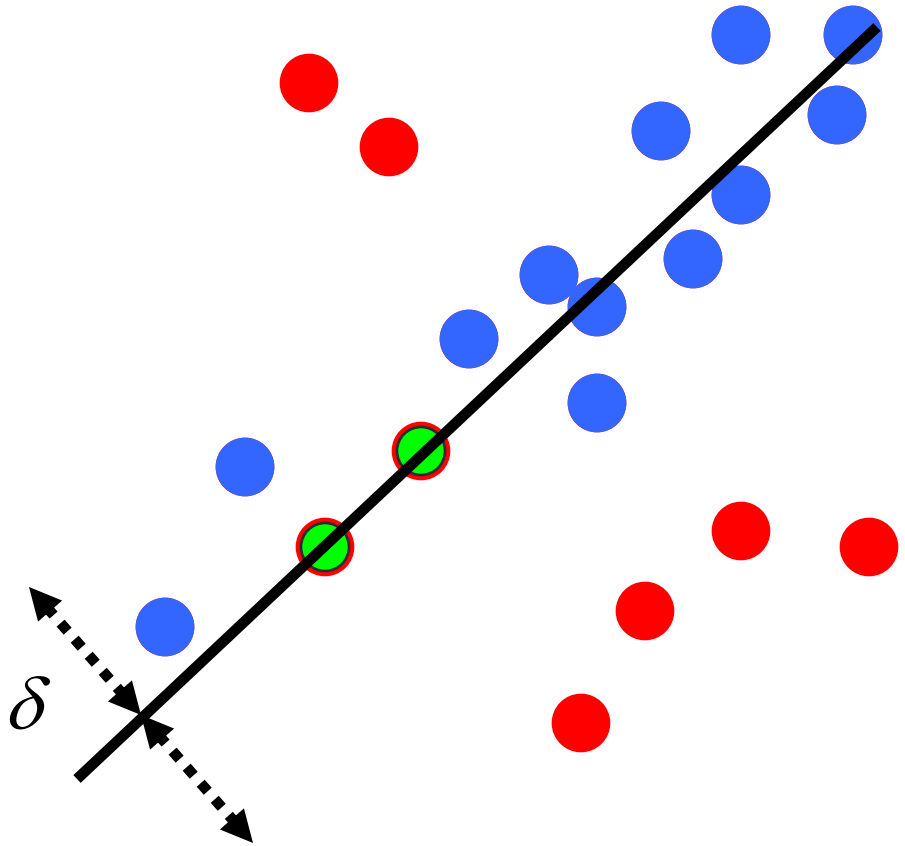
$|O| = 14$

Algorithm:

1. Select random sample of minimum required size to fit model [?] =[2]
2. Compute a putative model from sample set
3. Compute the set of inliers to this model from whole data set

Repeat 1-3 until model with the most inliers over all samples is found

RANSAC



$|O| = 6$

Algorithm:

1. Select random sample of minimum required size to fit model [?]
2. Compute a putative model from sample set
3. Compute the set of inliers to this model from whole data set

Repeat 1-3 until model with the most inliers over all samples is found

How many samples?

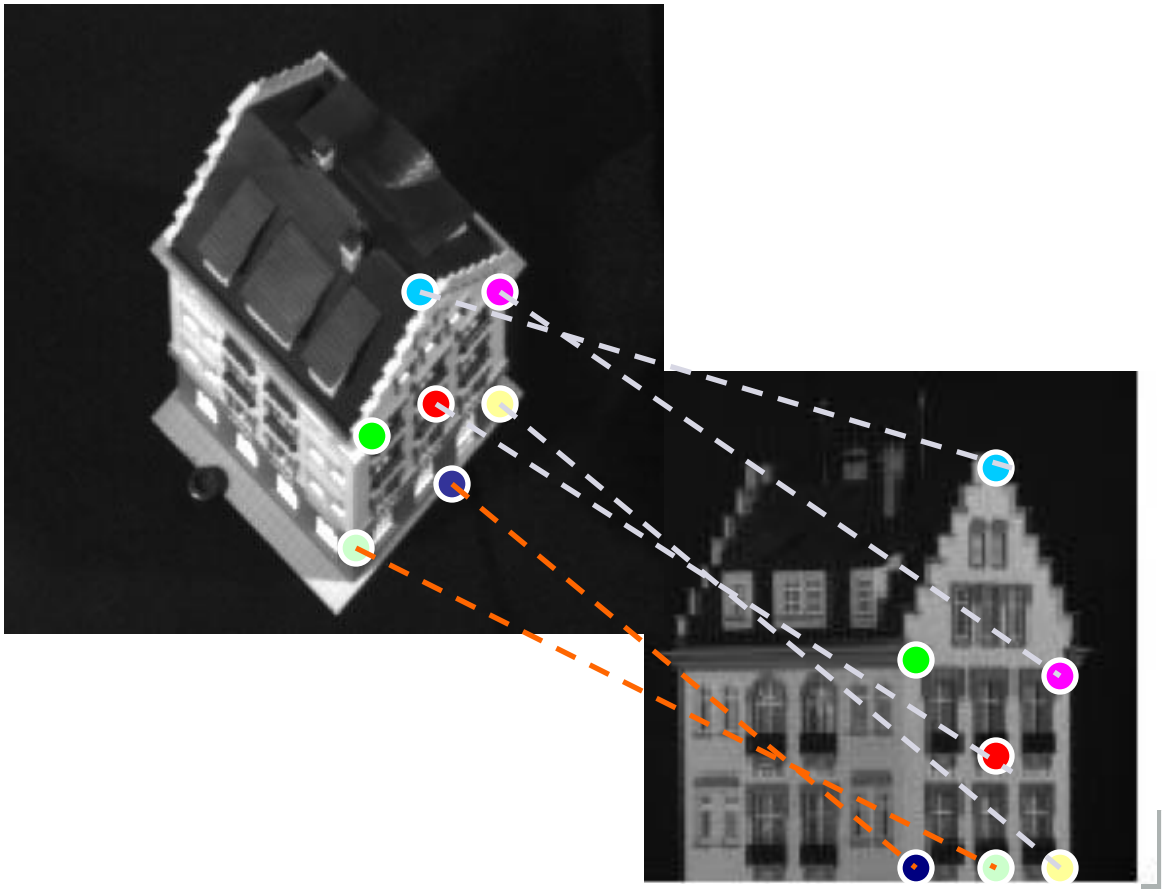
- Number of samples *N*
  - Choose *N* so that, with probability *p*, at least one random sample is free from outliers (e.g. *p*=0.99) (outlier ratio: e)
- Initial number of points *s*
  - Typically minimum number needed to fit the model
- Distance threshold δ
  - Choose δ so probability for inlier is *p* (e.g. 0.95)
  - Zero-mean Gaussian noise with std. dev. σ: $t^2=3.84\sigma^2$

$$N = \log(1-p)/\log\left(1-(1-e)^s\right)$$

	proportion of outliers *e*						
s	5%	10%	20%	25%	30%	40%	50%
2	2	3	5	6	7	11	17
3	3	4	7	9	11	19	35
4	3	5	9	13	17	34	72
5	4	6	12	17	26	57	146
6	4	7	16	24	37	97	293
7	4	8	20	33	54	163	588
8	5	9	26	44	78	272	1177

Source: M. Pollefeys

Estimating H by RANSAC



- H $\rightarrow$ 8 DOF
- Need 4 correspondences

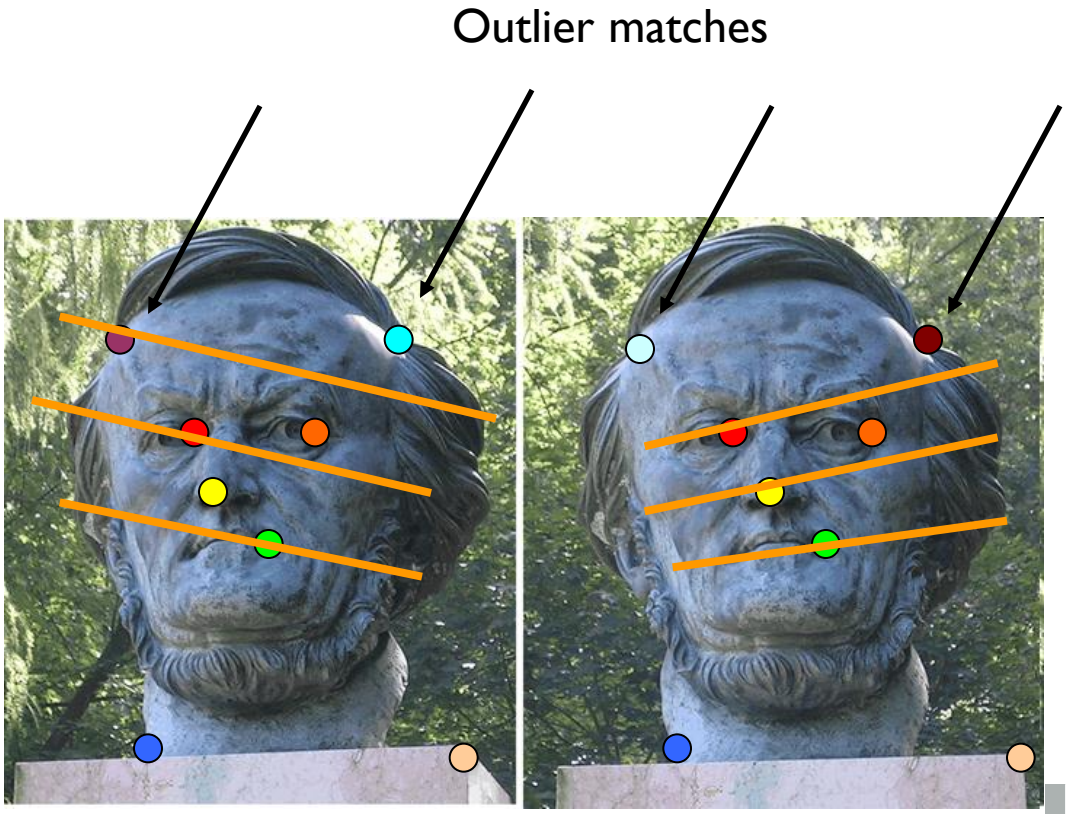
Sample set = set of matches between 2 images

Algorithm:

1. Select a random sample of minimum required size [?]
2. Compute a putative model from these
3. Compute the set of inliers to this model from whole sample space

Repeat 1-3 until model with the most inliers over all samples is found

Estimating F by RANSAC



- $F \rightarrow 7$ DOF
- Need 7 (8) correspondences

Sample set = set of matches between 2 images

Algorithm:

1. Select a random sample of minimum required size [?]
2. Compute a putative model from these
3. Compute the set of inliers to this model from whole sample space

Repeat 1-3 until model with the most inliers over all samples is found

RANSAC - conclusions

Good:

- Simple and easily implementable
- Successful in different contexts

Bad:

- Many parameters to tune
- Trade-off accuracy-vs-time
- Cannot be use if ratio inliers/outliers is too small

Fitting

- Goal:

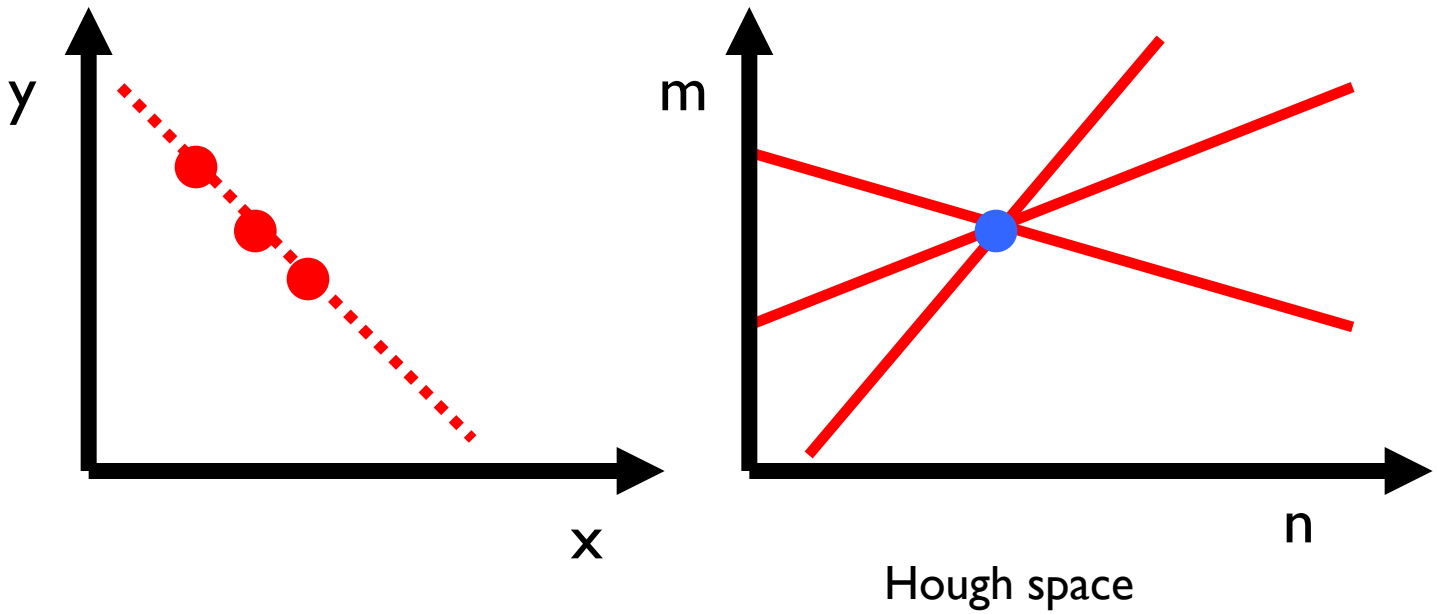
Choose a parametric model to fit a certain quantity from data

- Techniques:
  - Least square methods
  - RANSAC
  - Hough transform
  - EM (Expectation Maximization)

Hough Transform

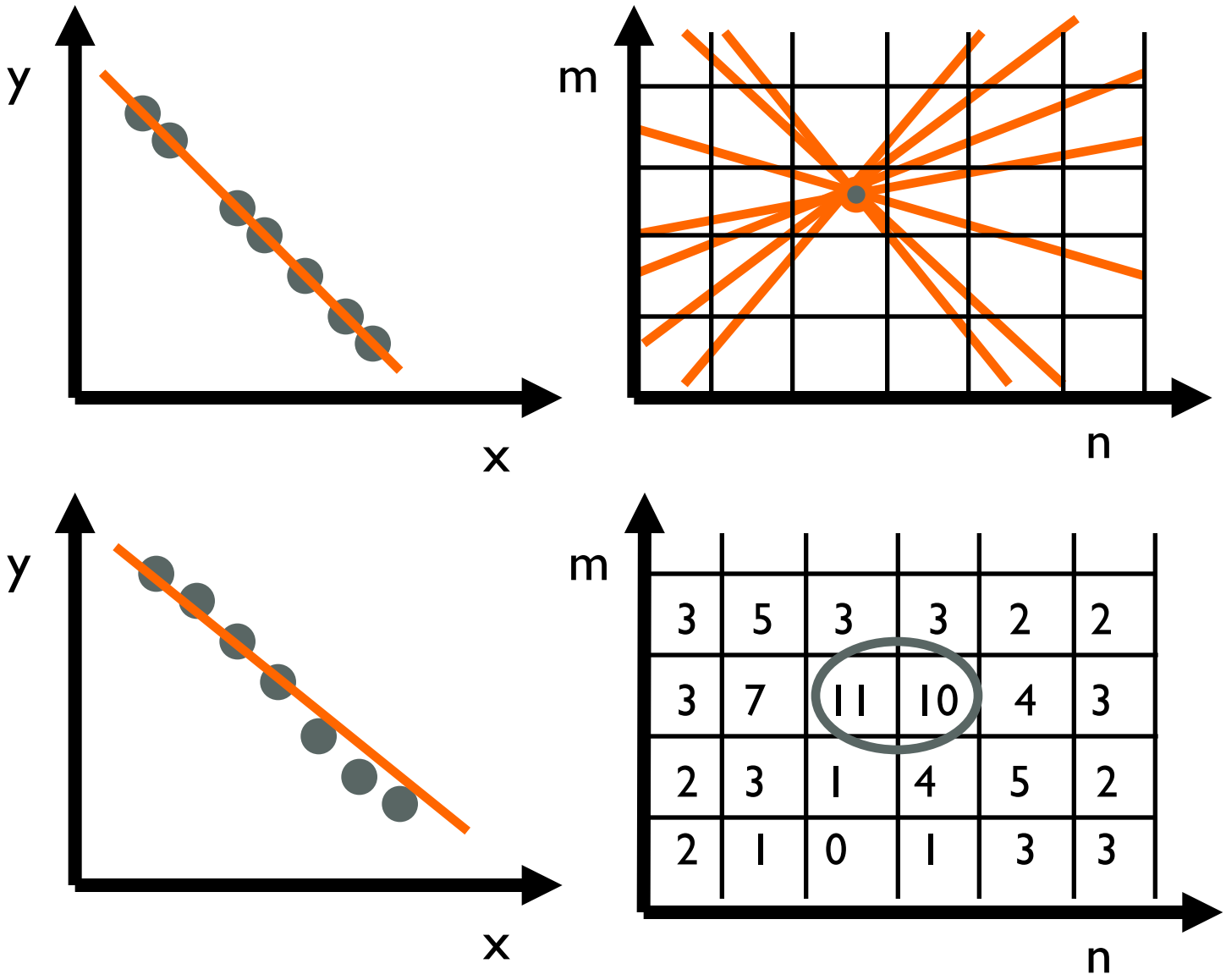
P.V.C. Hough, *Machine Analysis of Bubble Chamber Pictures,* Proc. Int. Conf. High Energy Accelerators and Instrumentation, 1959

Given a set of , find the curve or line that explains the data points best



$$y = m\,x + n$$

Hough Transform

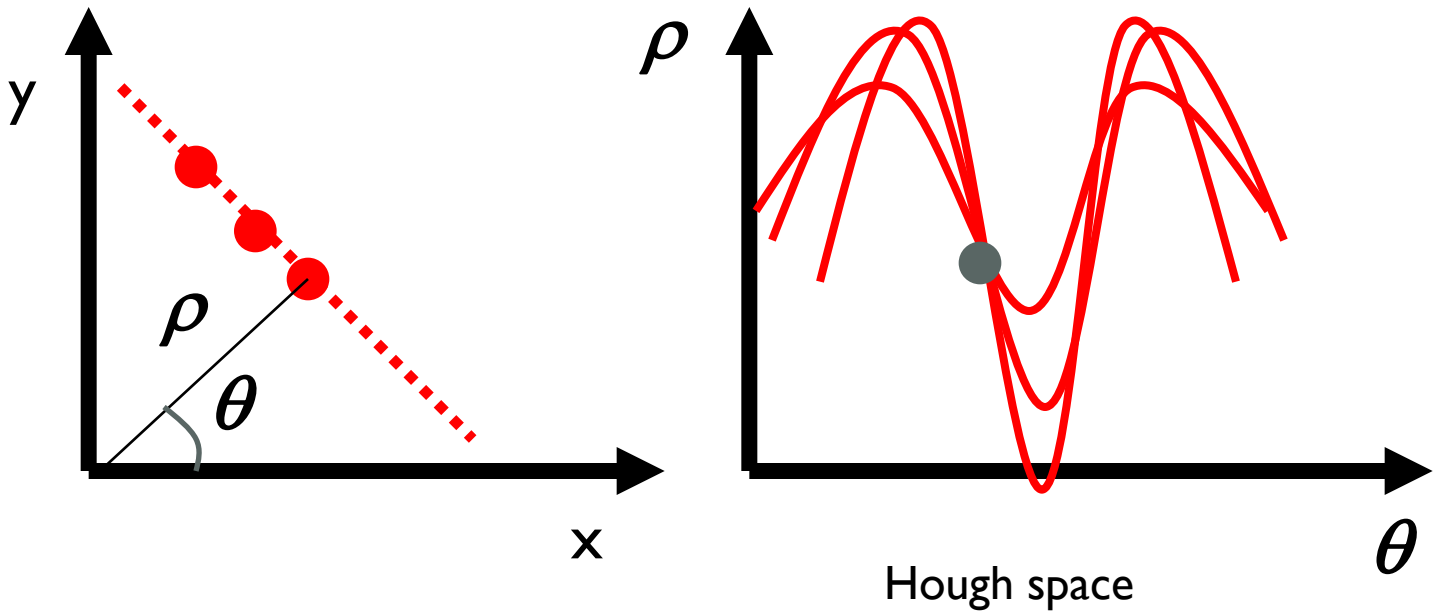


Hough Transform

P.V.C. Hough, *Machine Analysis of Bubble Chamber Pictures,* Proc. Int. Conf. High Energy Accelerators and Instrumentation, 1959

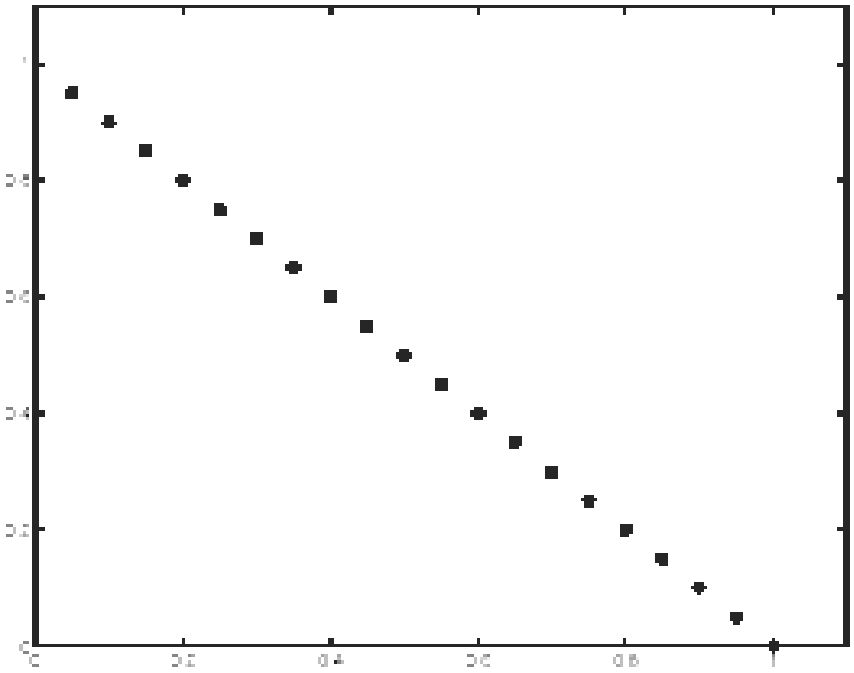
Issue : parameter space [m,n] is unbounded…

- Use a polar representation for the parameter space

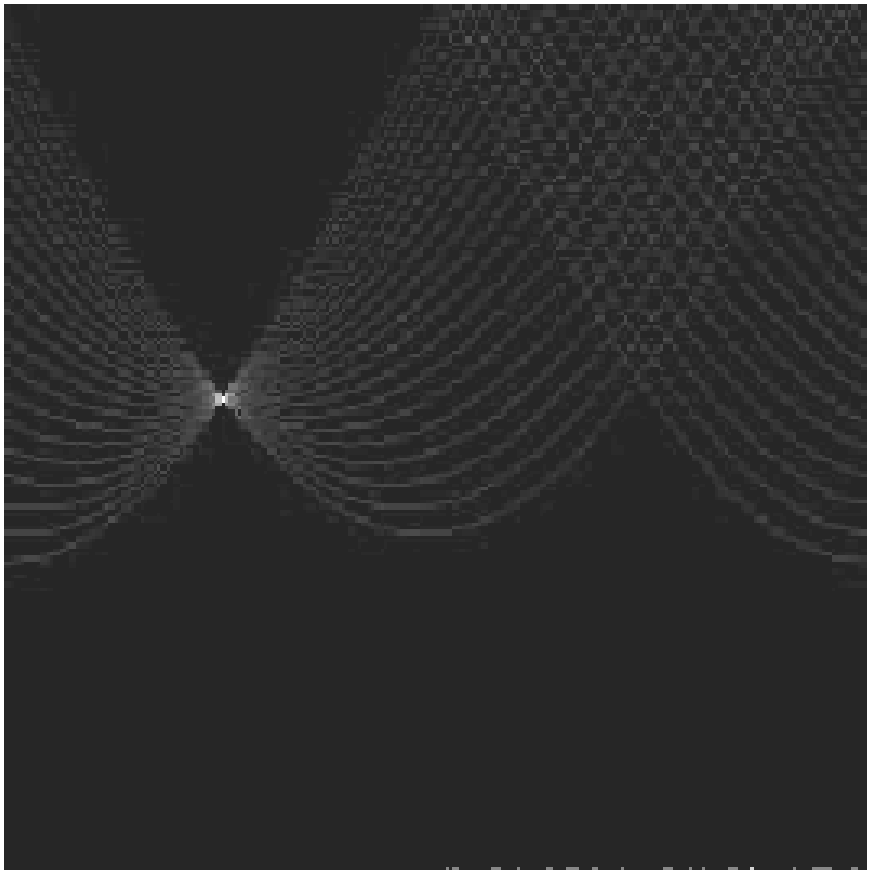


$$x\cos\theta + y\sin\theta = \rho$$

Hough Transform - Experiments



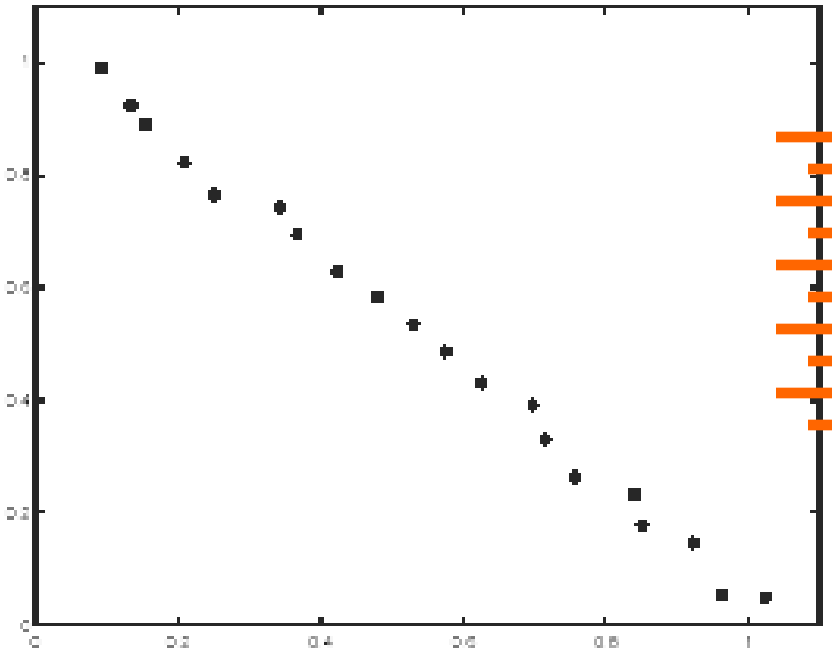
features



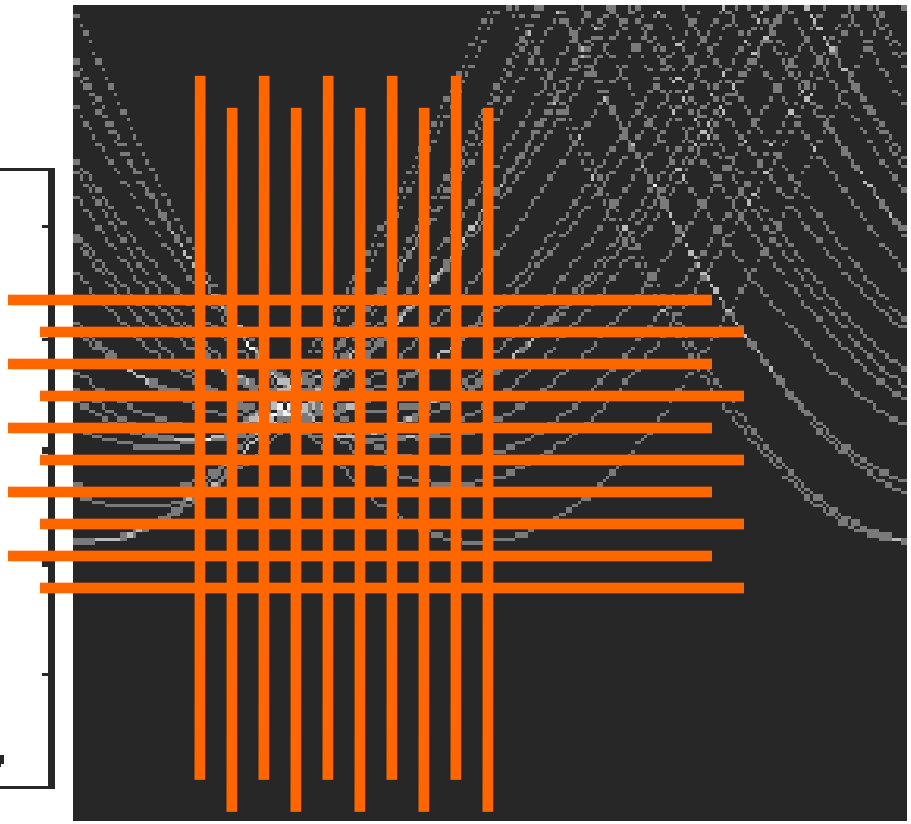
votes

Hough Transform - Experiments

Noisy data



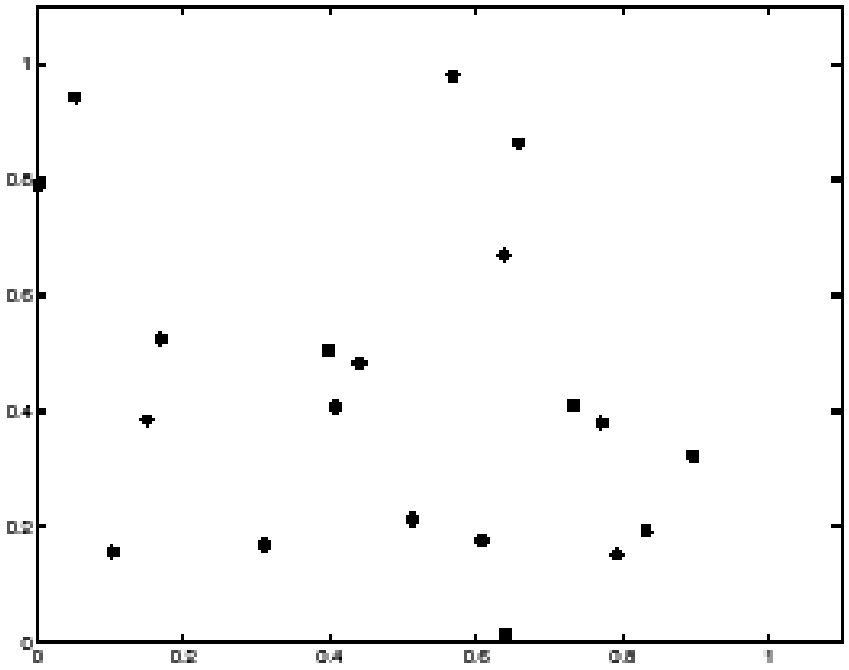
features



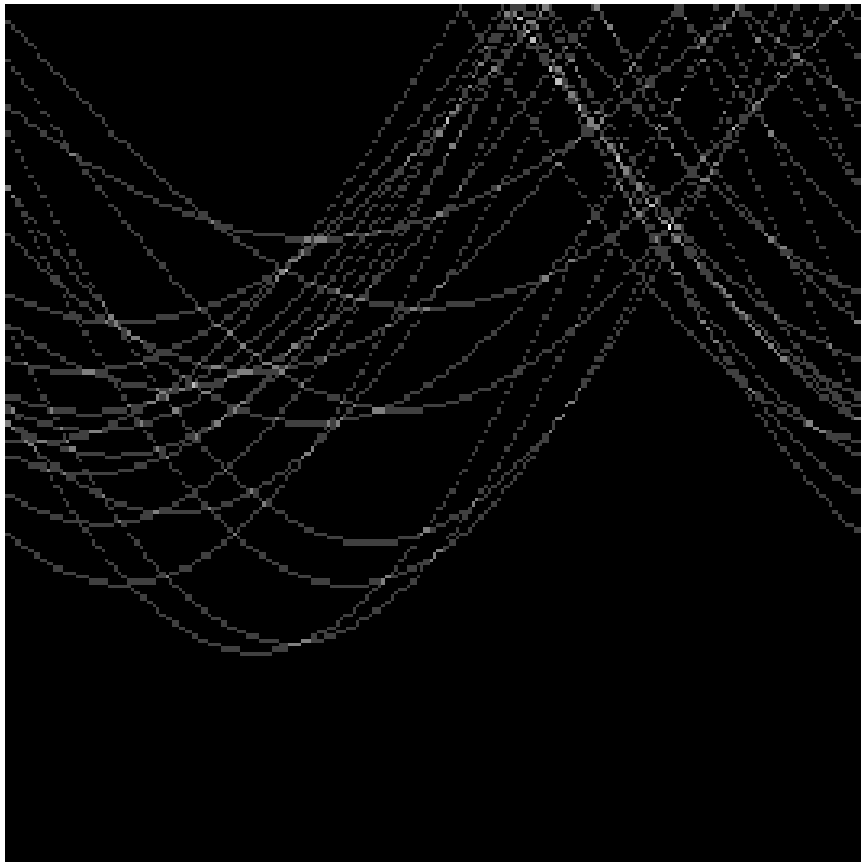
votes

Issue: Grid size needs to be adjusted…

Hough Transform - Experiments



features



votes

Issue: spurious peaks due to uniform noise

Hough transform - conclusions

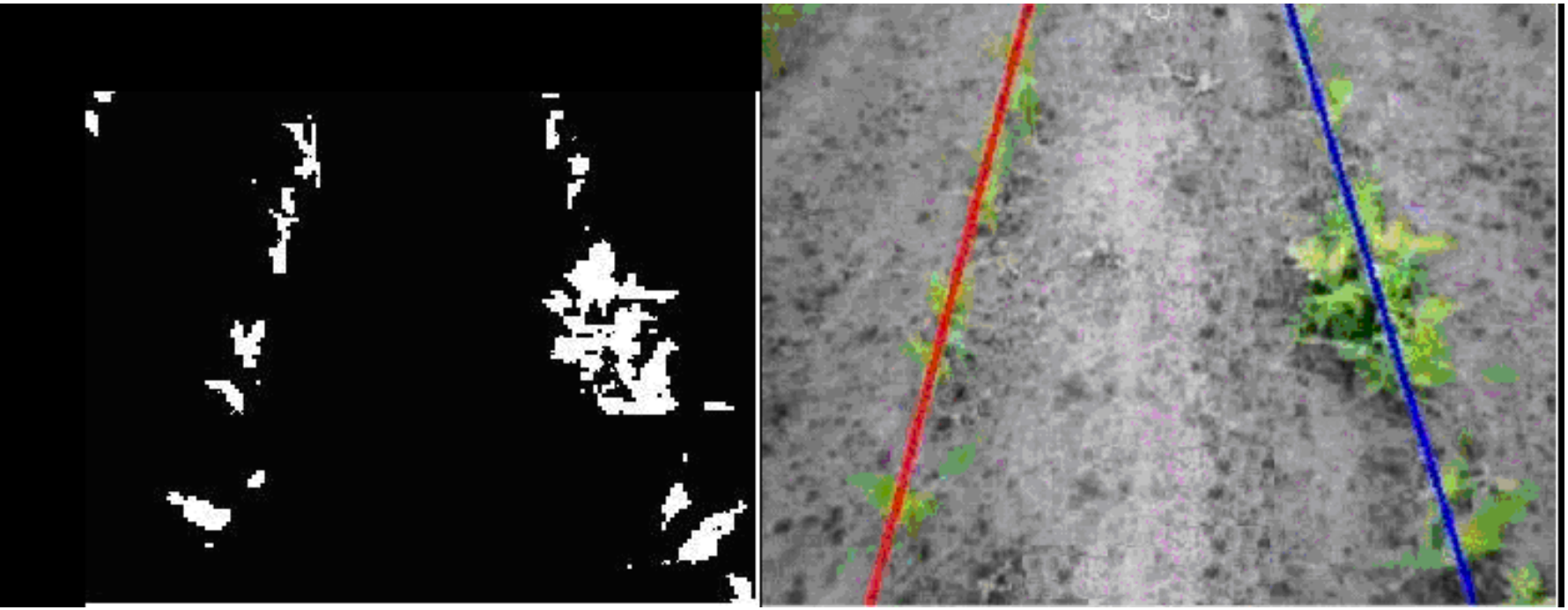
Good:

- All points are processed independently, so can cope with occlusion/outliers
- Some robustness to noise: noise points unlikely to contribute consistently to any single bin

Bad:

- Spurious peaks due to uniform noise
- Trade-off noise-grid size (hard to find sweet point)

Hough Transform - Experiments



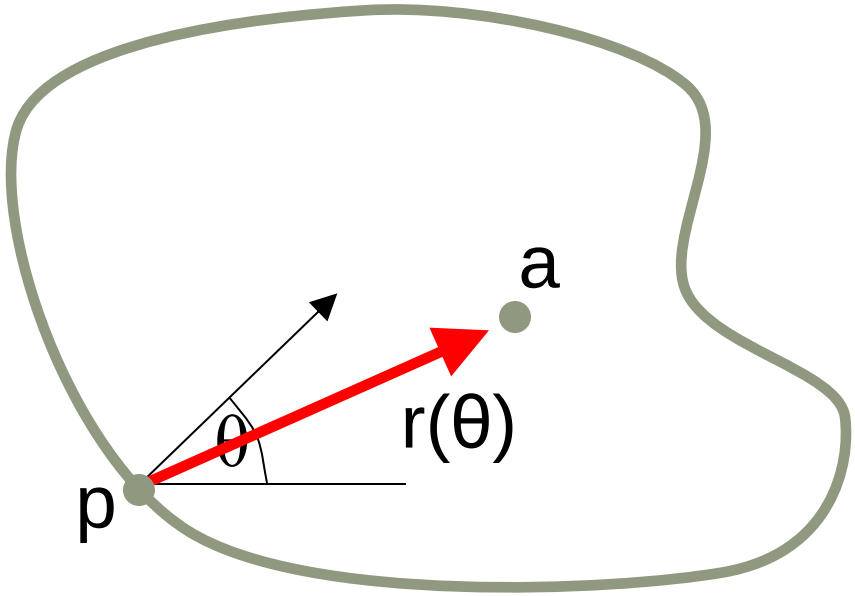
Courtesy of TKK Automation Technology Laboratory

Generalized Hough transform

D. Ballard, Generalizing the Hough Transform to Detect Arbitrary Shapes, Pattern Recognition 13(2), 1981

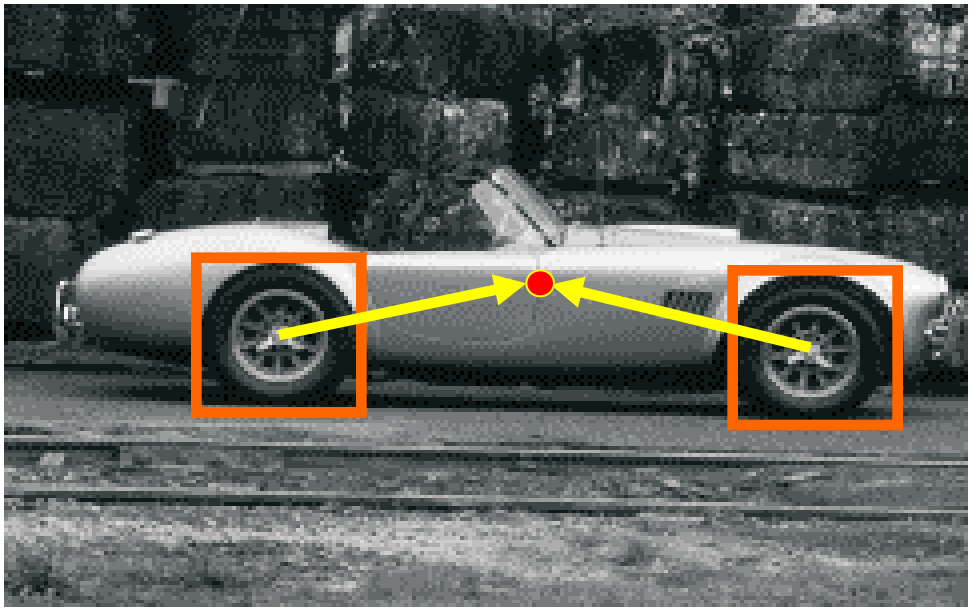
- Identify a shape model by measuring the location of its parts and shape centroid

- Measurements: orientation theta, location of p
- Each measurement casts a vote in the Hough space: $p + r(\theta)$

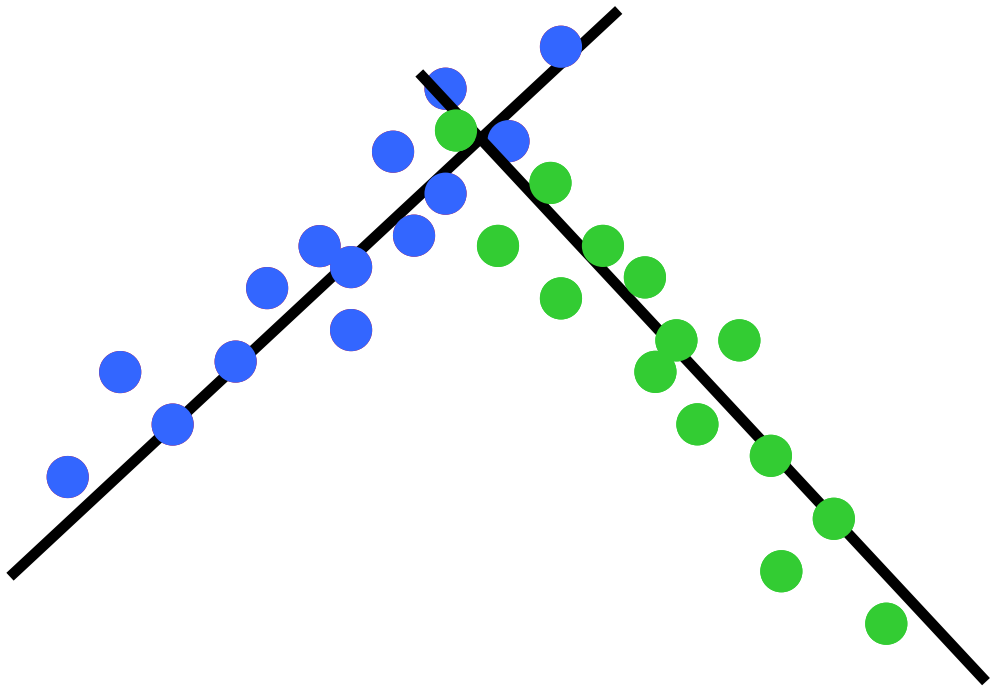


Generalized Hough transform

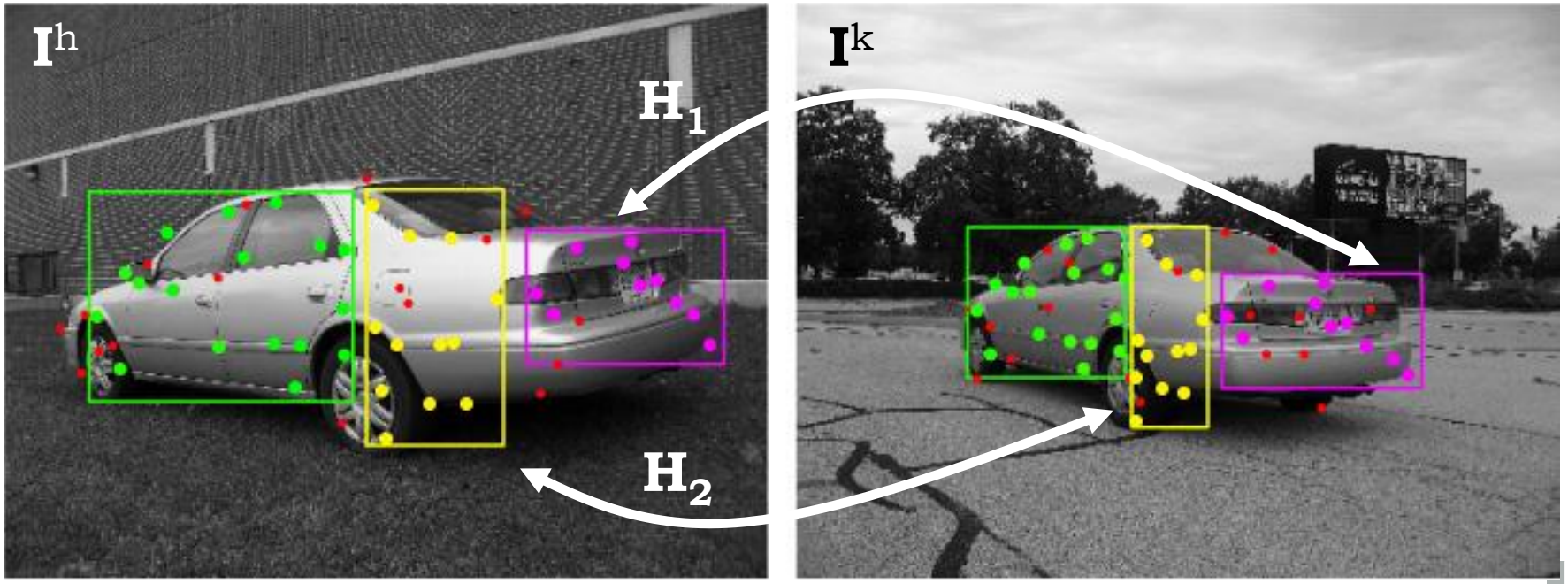
B. Leibe, A. Leonardis, and B. Schiele, Combined Object Categorization and Segmentation with an Implicit Shape Model, ECCV Workshop on Statistical Learning in Computer Vision 2004



Fitting multiple models



Fitting multiple models



- Incremental fitting
- E.M. (probabilistic fitting)
- Hough transform

Incremental line fitting

Scan data point sequentially (using locality constraints)

Perform following loop:

1. Select N point and fit line to N points
2. Compute residual R_N
3. Add a new point, re-fit line and re-compute R_{N+1}
4. Continue while line fitting residual is small enough,

- When residual exceeds a threshold, start fitting new model (line)

Hough transform

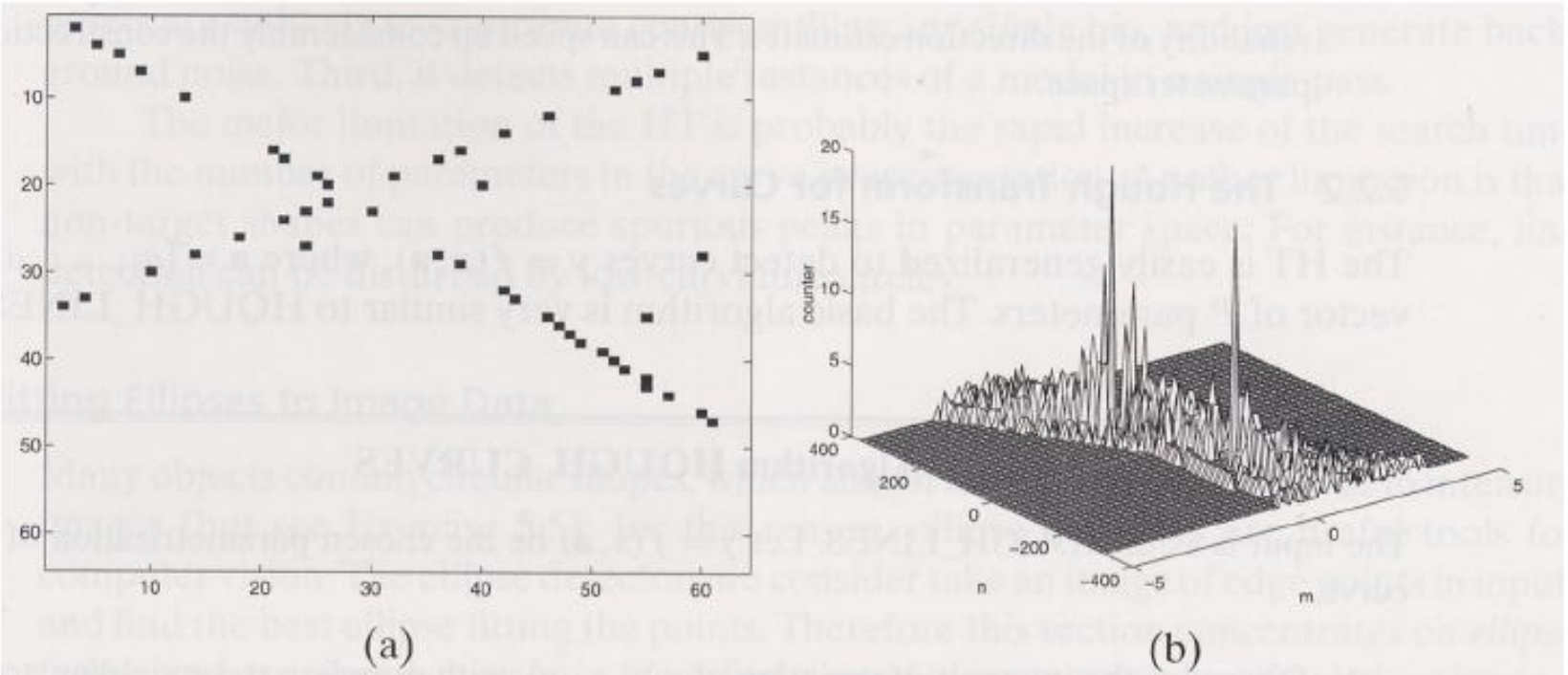


Figure 5.2 (a) An image containing two lines, sampled irregularly, and several random points. (b) Plot of the counters in the corresponding parameter space (how many points contribute to each cell (m, n)). Notice that the main peaks are obvious, but there are many secondary peaks.

Courtesy of unknown

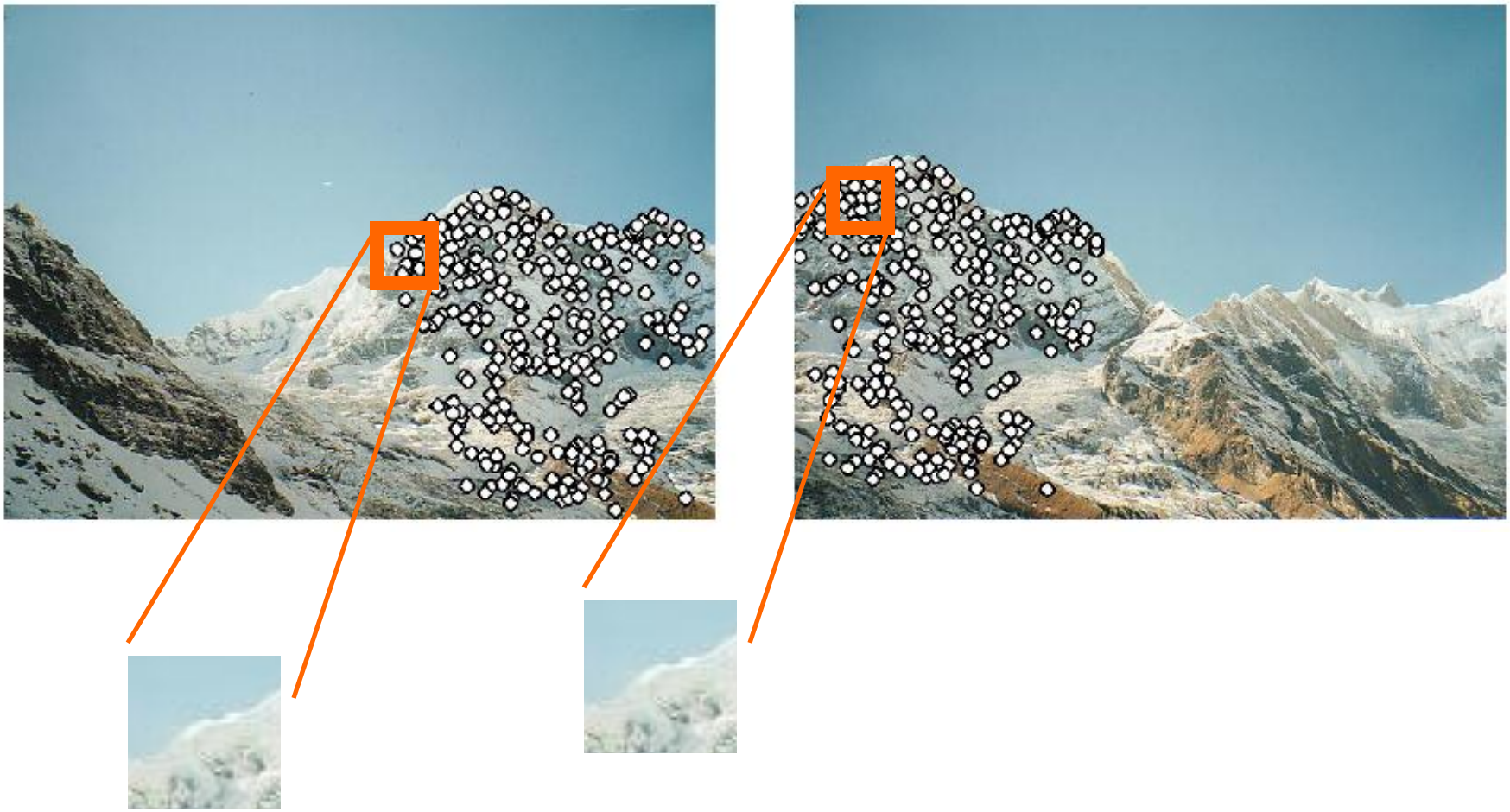
Same cons and pros as before…

Fitting helps matching!

Suppose we need to align these 2 images

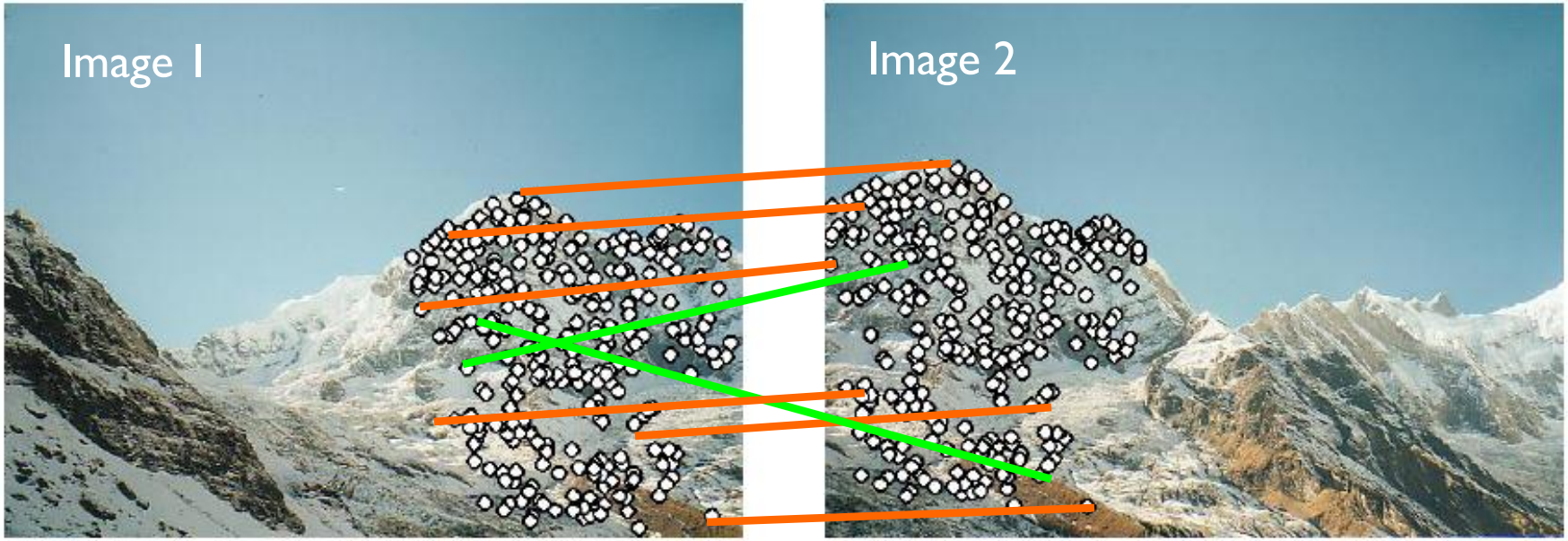
Images courtesy of M. Brown

Fitting helps matching!



Feature are matched (for instance, based on correlation)

Fitting helps matching!



Matches bases on appearance only

Red: good matches

Green: bad matches

Idea:

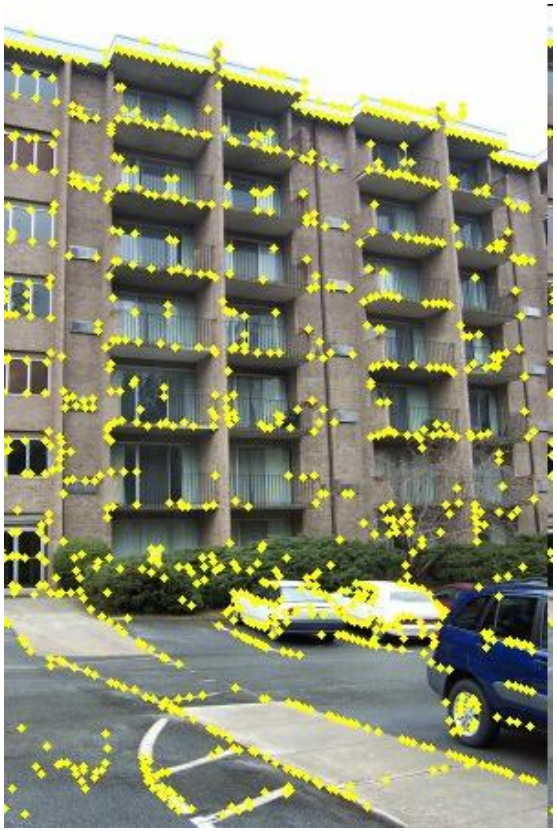
- Fitting an homography H (by RANSAC) mapping features from images 1 to 2
- Bad matches will be labeled as outliers (hence rejected)!

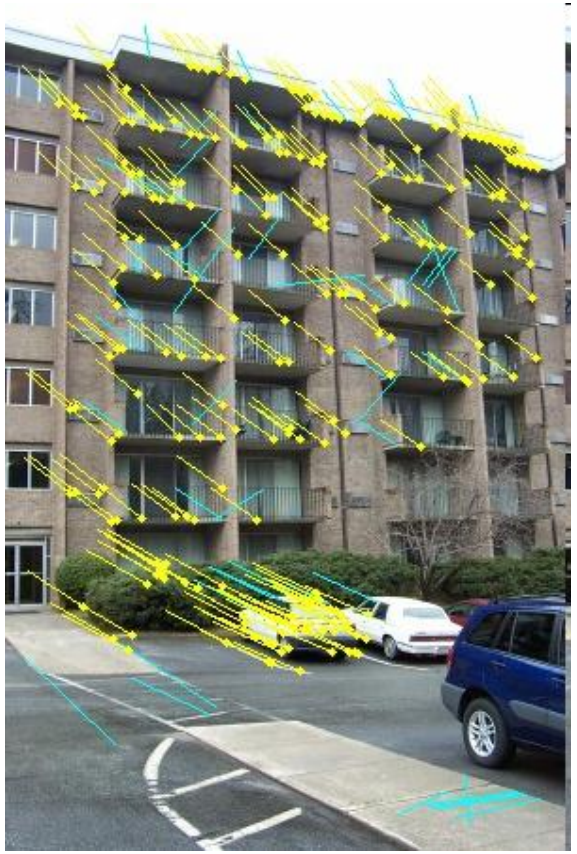
Fitting helps matching!

Recognising Panoramas

M. Brown and D. G. Lowe. Recognising Panoramas. In Proceedings of the *9th International Conference on Computer Vision – ICCV2003*

Fitting helps matching!





Images courtesy of Brandon Lloyd