COLLEGE OF ENGINEERING
ELECTRICAL ENGINEERING & COMPUTER SCIENCE
UNIVERSITY OF MICHIGAN

Fitting

EECS 598-08 Fall 2014
Foundations of Computer Vision

Instructor: Jason Corso (jjcorso)
web.eecs.umich.edu/~jjcorso/t/598F14

Readings: FP 10; SZ 4.3, 5.1

Date: 10/8/14

Plan

- Problem Formulation
- Least Squares Methods
- RANSAC
- Hough Transform
- Multi-model Fitting
- Expectation-Maximization
- Examples of Uses of Fitting

What is Fitting?

Goals:

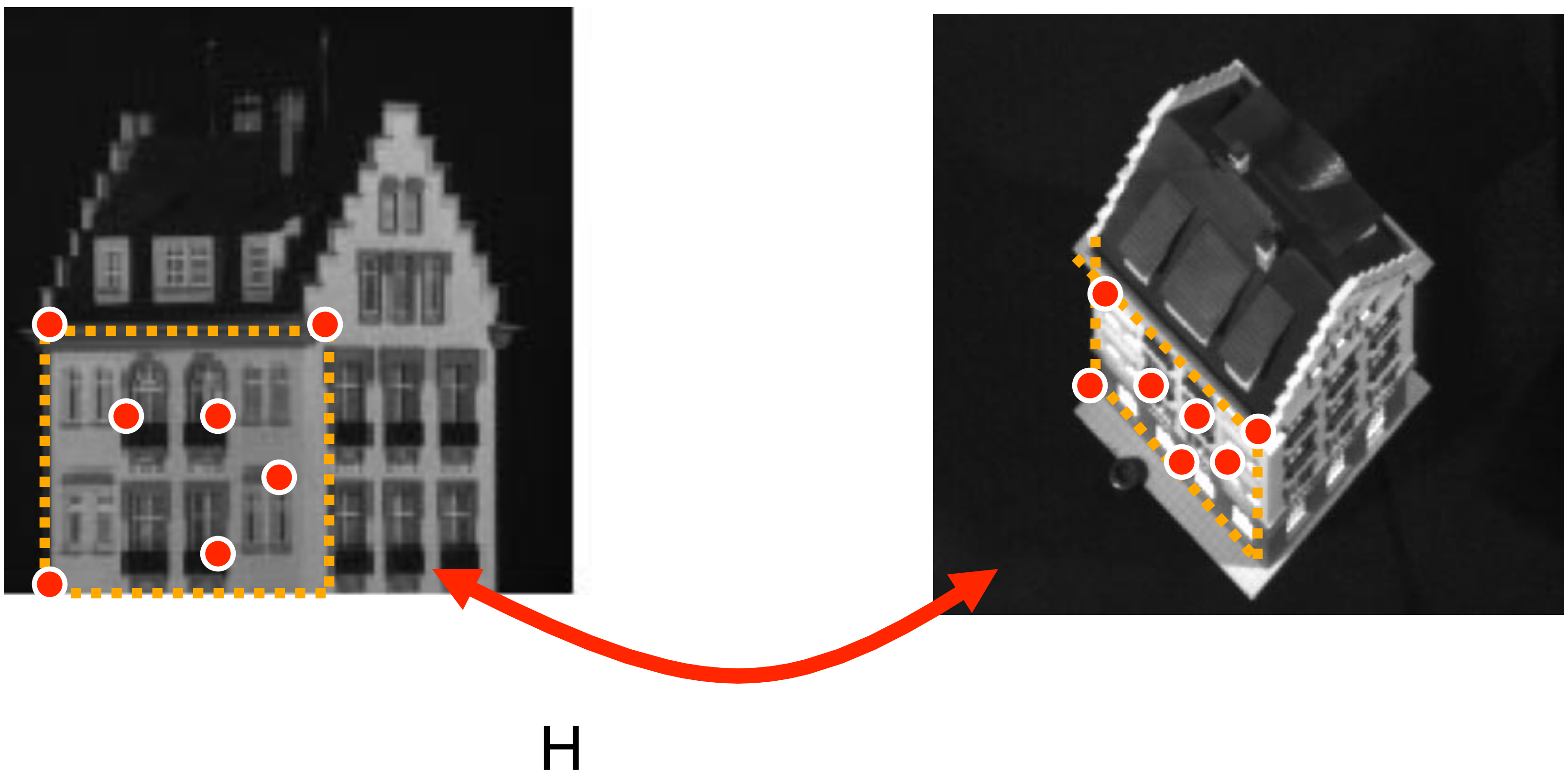
- Choose a parametric model to fit a certain quantity from data
- Estimate model parameters

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

Example: fitting lines
(for computing vanishing points)

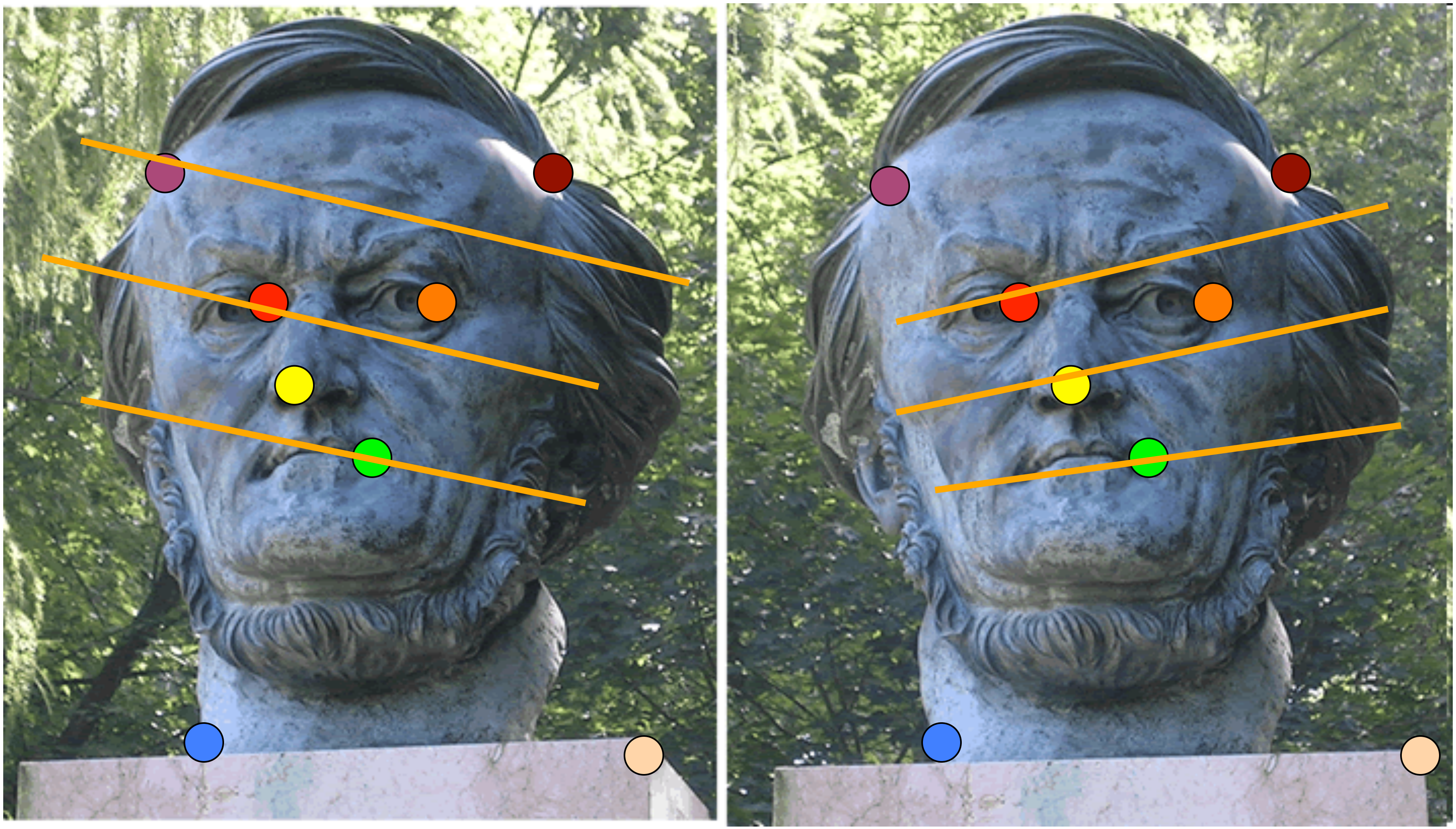
Source: S. Savarese slides.

Example: Estimating an homographic transformation



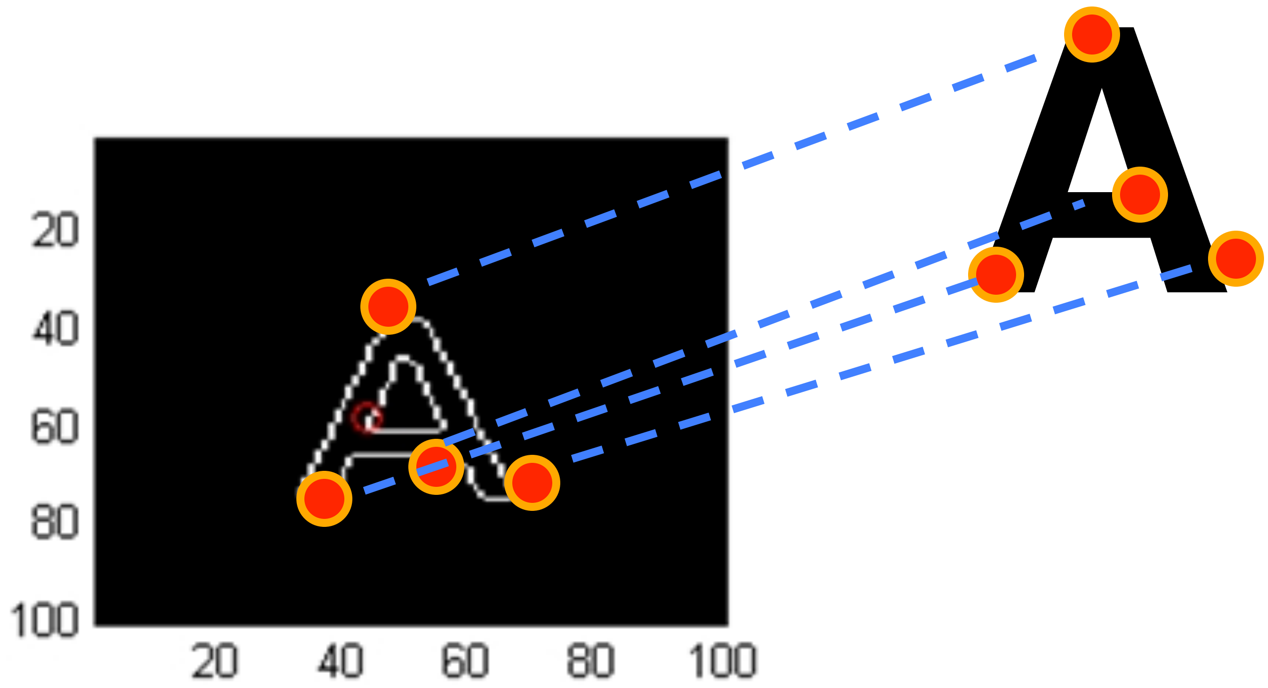
Source: S. Savarese slides.

Example: Estimating F



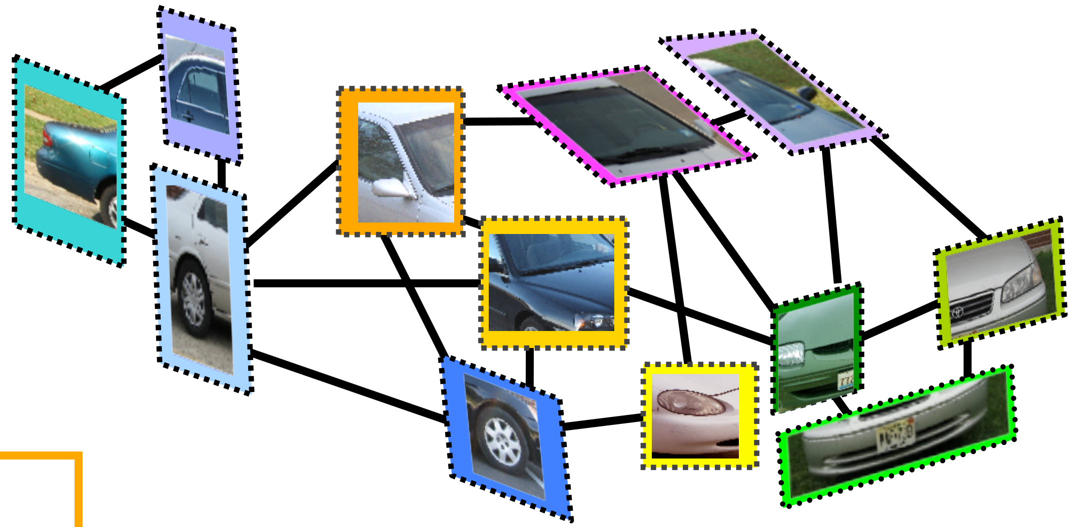
Source: S. Savarese slides.

Example: fitting a 2D shape template



Source: S. Savarese slides.

Example: fitting a 3D object model



Source: S. Savarese slides.

Fitting

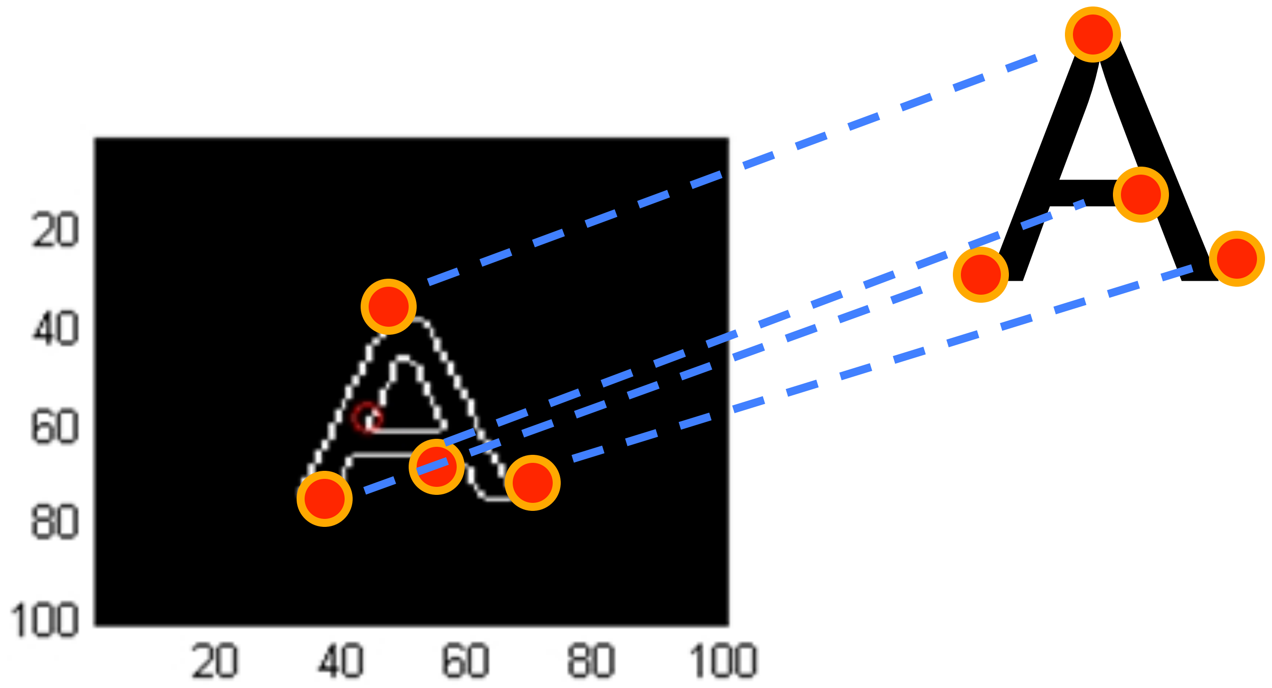
- Critical issues:
  - Noisy data
  - Outliers
  - Missing data

Source: S. Savarese slides.

Critical issues: noisy data

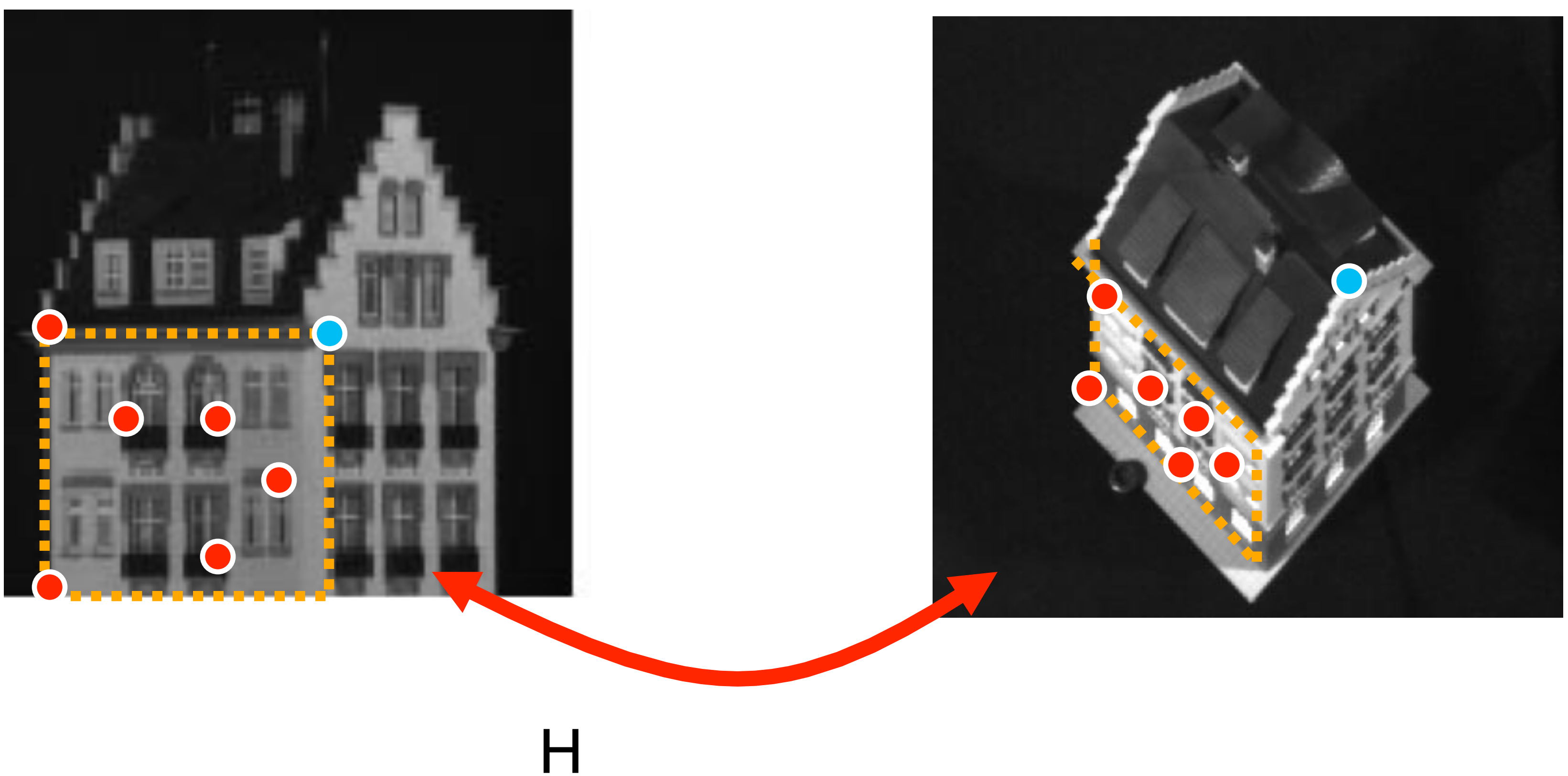
Source: S. Savarese slides.

Critical issues: noisy data (intra-class variability)



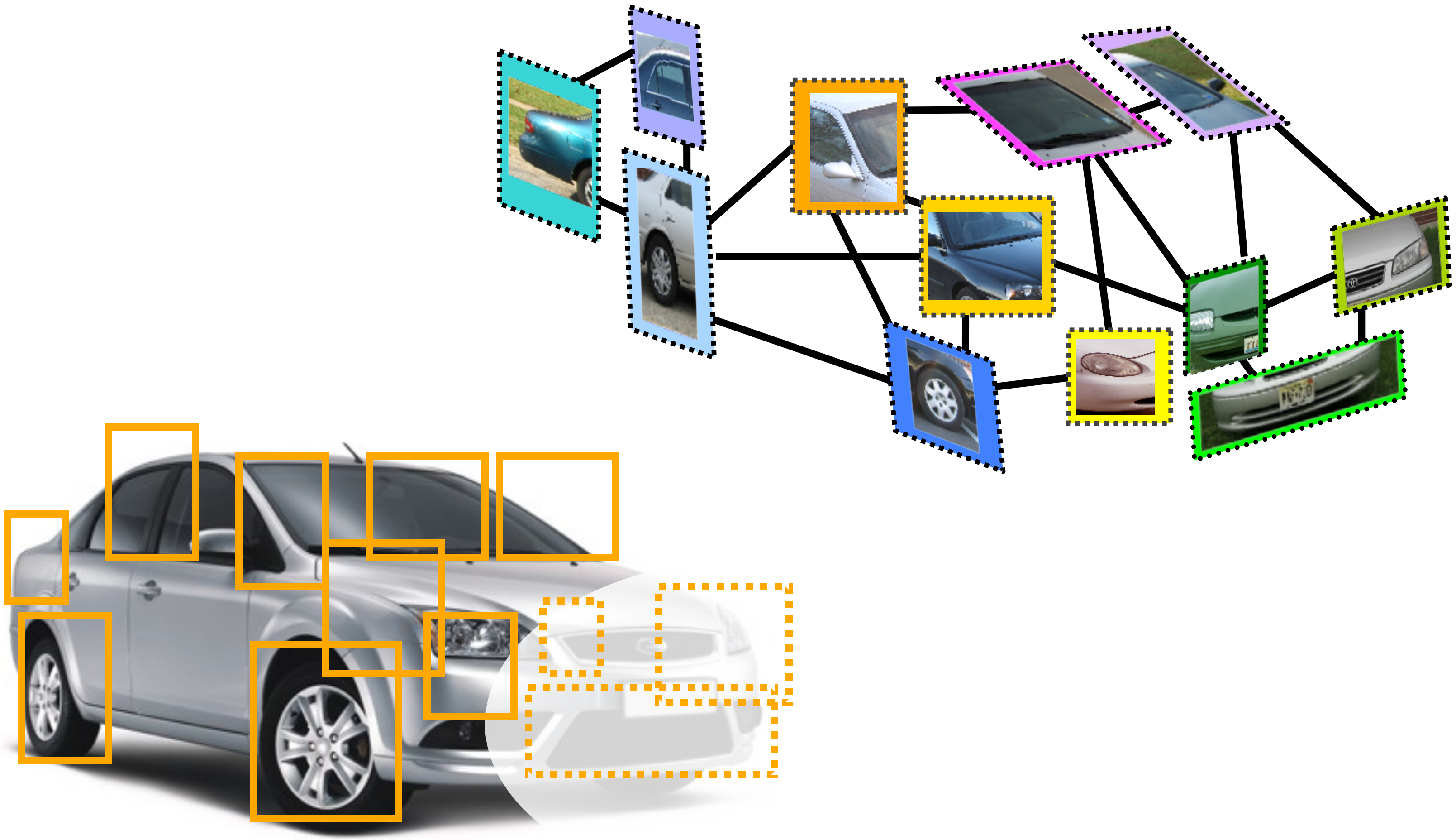
Source: S. Savarese slides.

Critical issues: outliers



Source: S. Savarese slides.

Critical issues: missing data (occlusions)



Source: S. Savarese slides.

Fitting

Goal: Choose a parametric model to fit a certain quantity from data

Techniques:

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

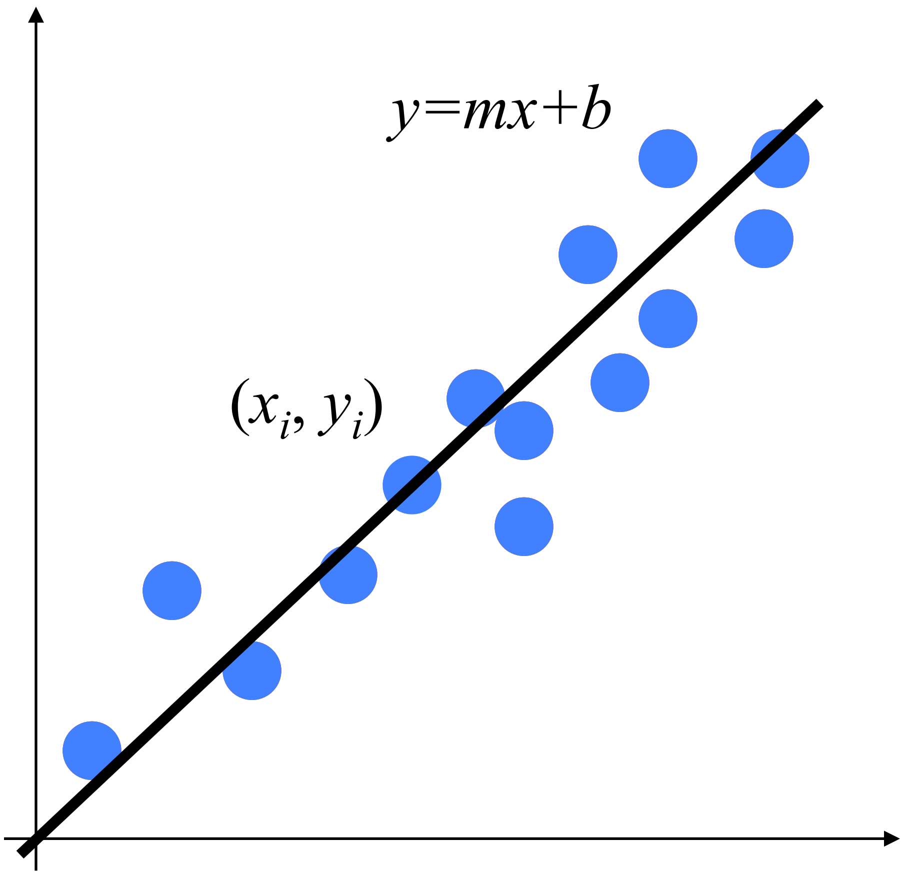
Source: S. Savarese slides.

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

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



Least squares methods

- fitting a line -

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

Source: S. Savarese slides.

Least squares methods

- fitting a line -

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

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

$$= (Y - XB)^T (Y - XB) = Y^T Y - 2(XB)^T Y + (XB)^T (XB)$$

Find (m, b) that minimize E

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

$$X^T XB = X^T Y$$

Normal equation

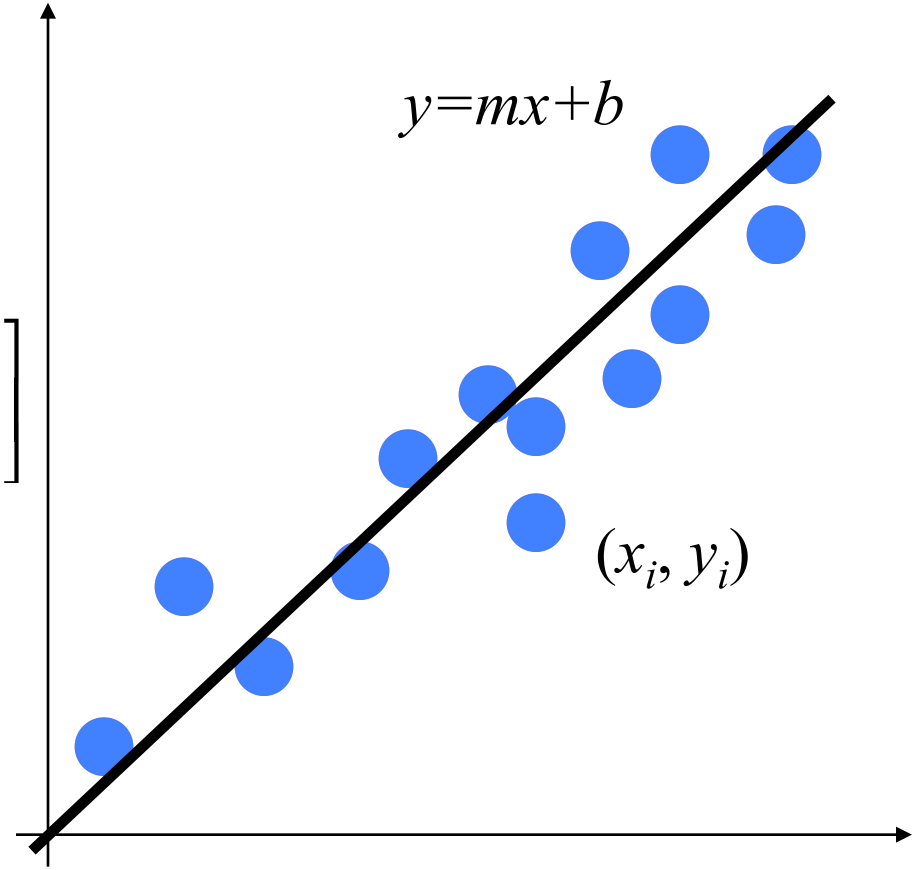
$$B = \left(X^T X\right)^{-1} X^T Y$$

Source: S. Savarese slides.

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

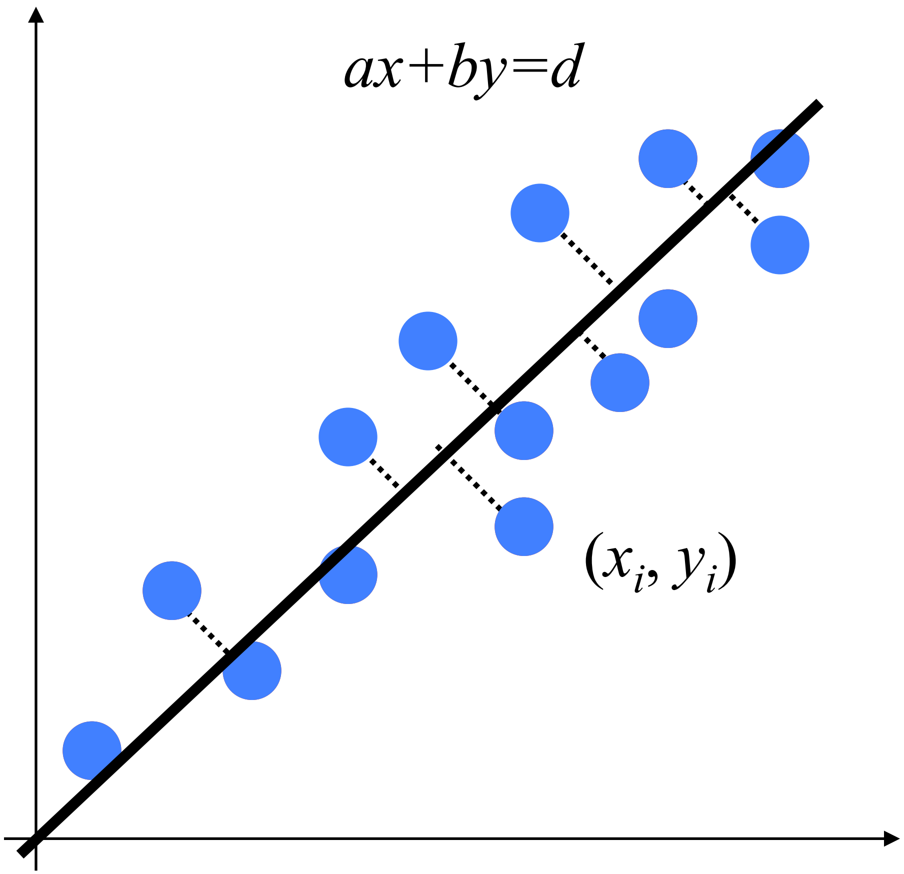
- Fails completely for vertical lines

Source: S. Savarese slides.

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

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



$$U N = 0$$

data　model parameters

Source: S. Savarese slides.

Least squares methods

- fitting a line -

$$\mathrm{A\,h} = 0$$

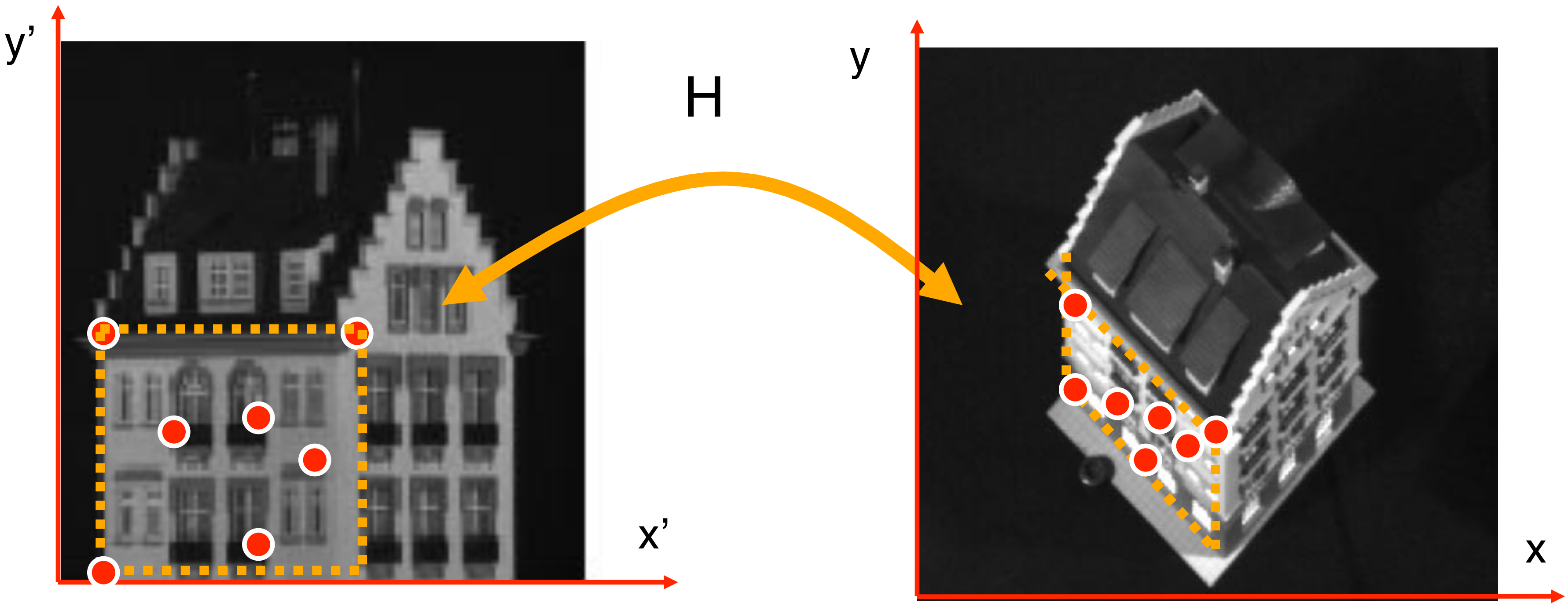
$$\text{Minimize } \|\mathrm{A\,h}\| \quad \text{subject to} \quad \|\mathrm{h}\| = 1$$

$$A = UDV^T$$

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

Source: S. Savarese slides.

Least squares methods
– fitting an homography –

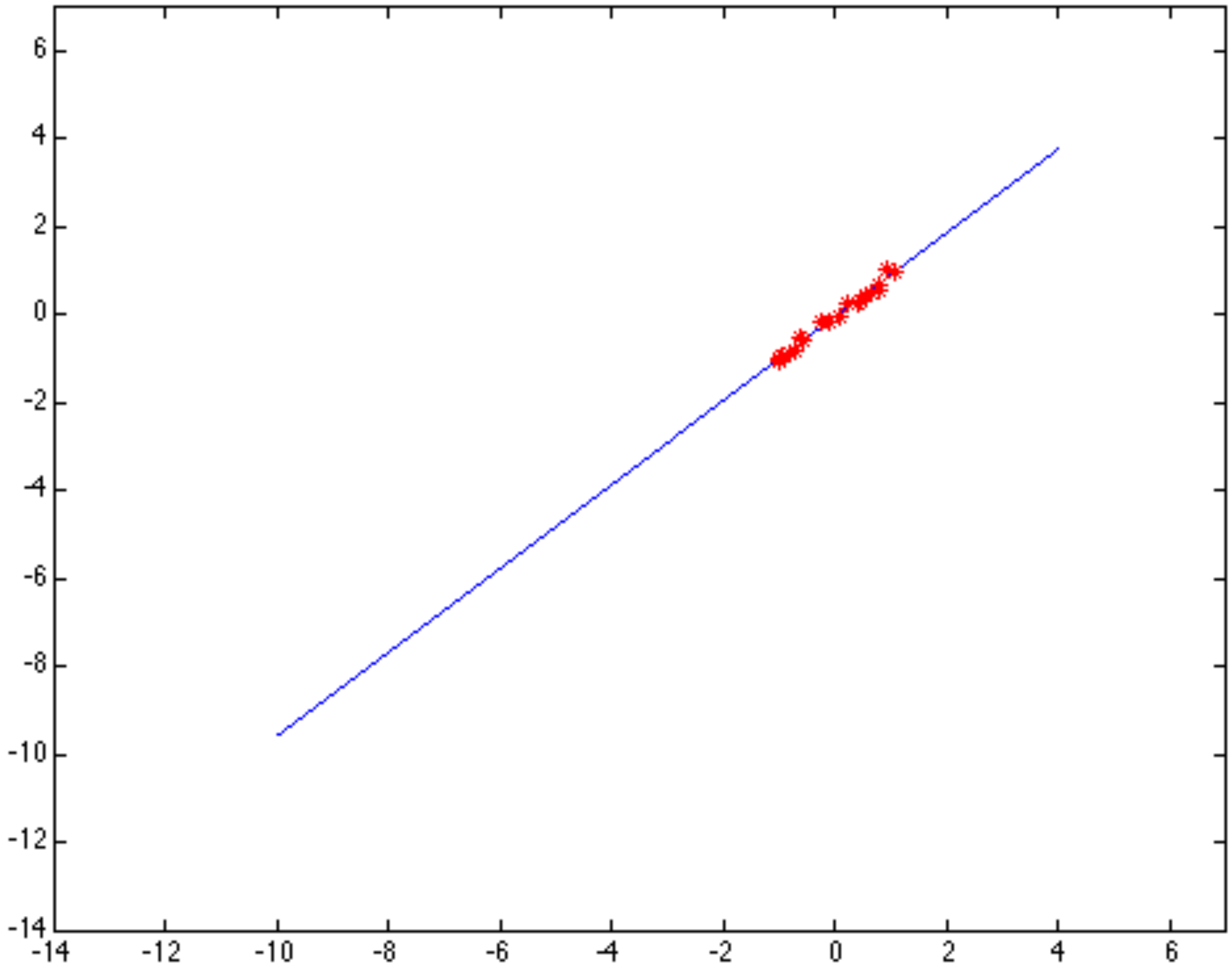


$$UN = 0$$

data model parameters

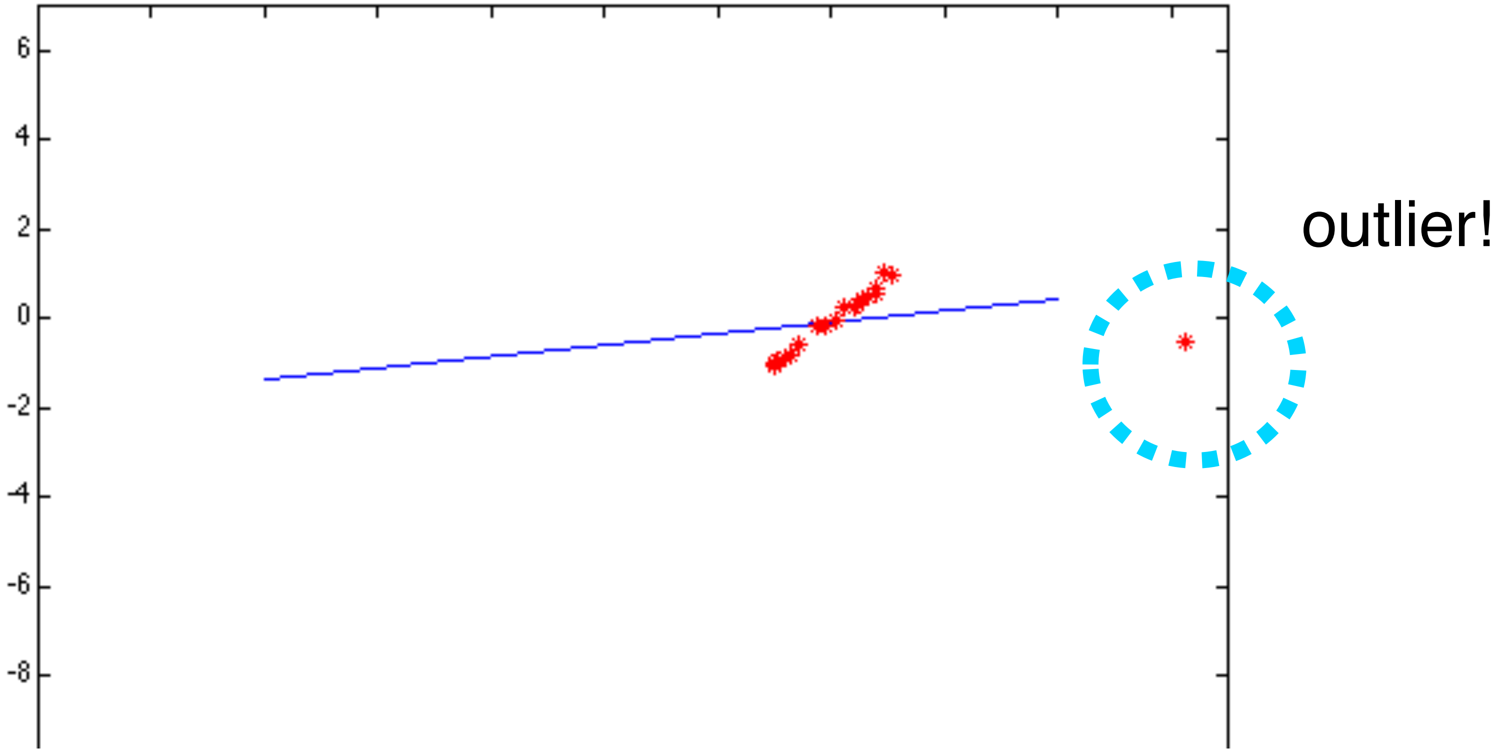
Source: S. Savarese slides.

Least squares: Robustness to noise



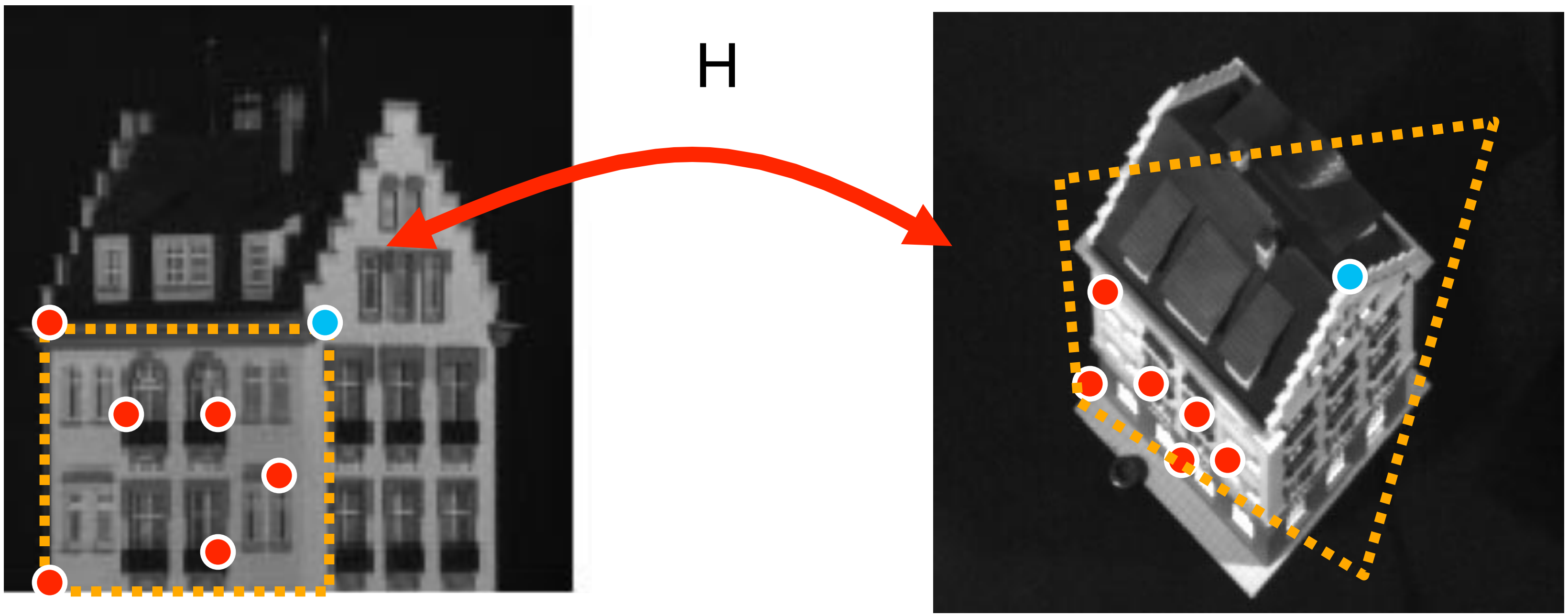
Source: S. Savarese slides.

Least squares: Robustness to noise



Source: S. Savarese slides.

Critical issues: outliers



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

Source: S. Savarese slides.

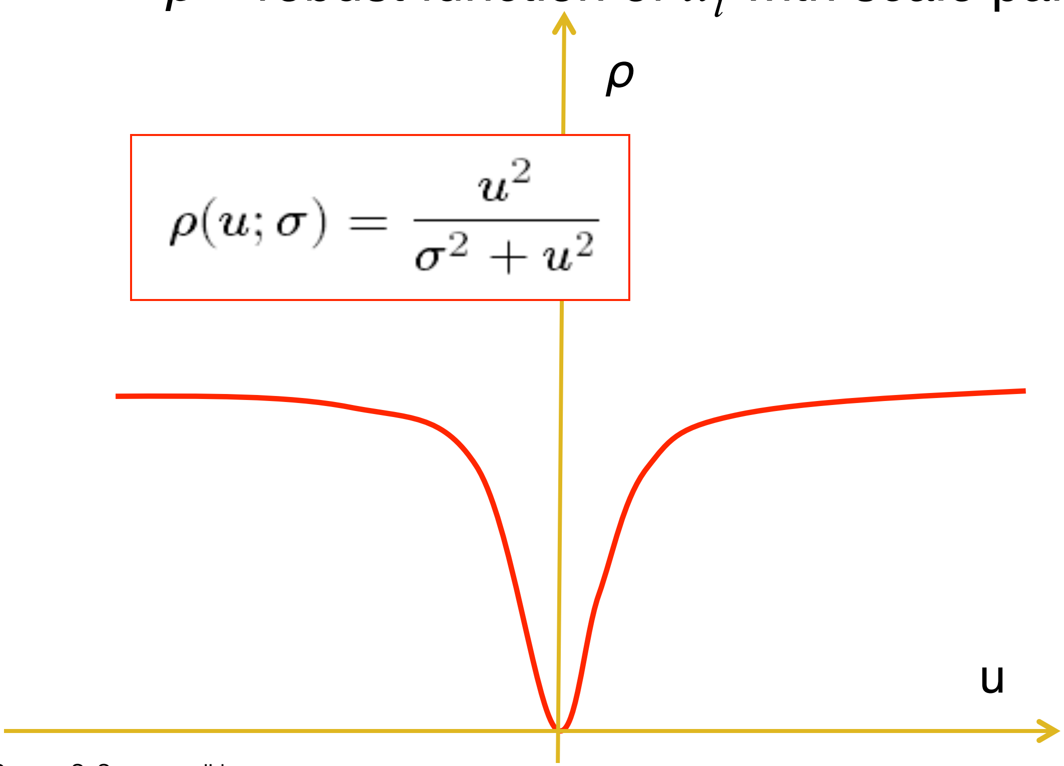
Least squares: Robust estimators

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

We minimize $E = \sum_i \rho(u_i;\sigma)$ $\quad u_i = ax_i + by_i - d$

- u_i = error (residual) of ith point w.r.t. model parameters β = (a,b,d)
- ρ = robust function of u_i with scale parameter σ

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



The robust function ρ

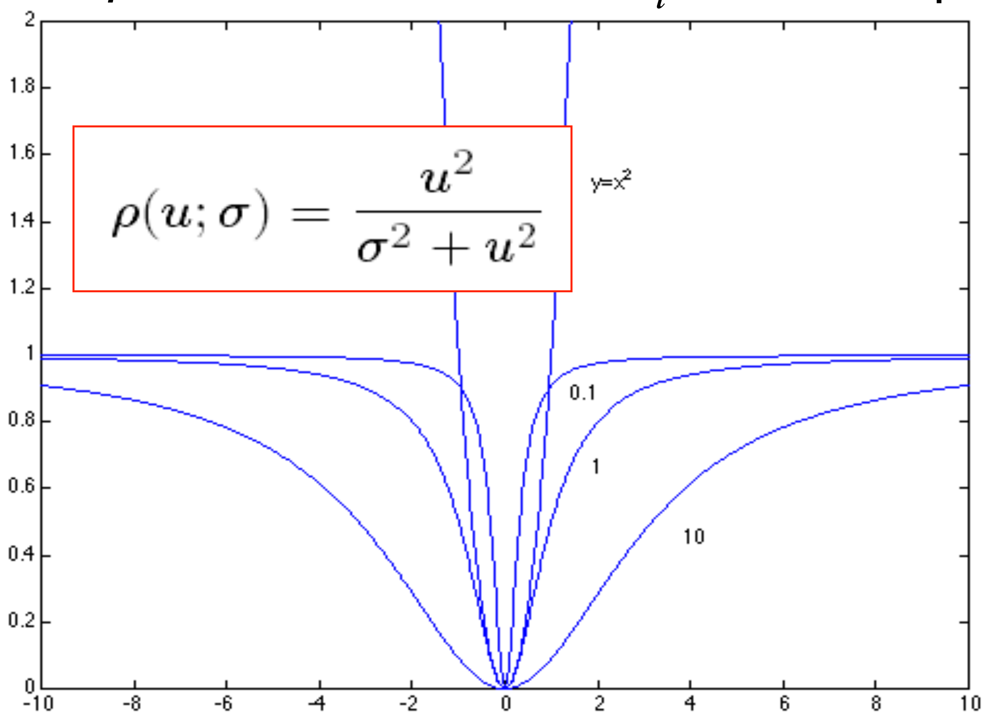
- Favors a configuration with small residuals
- Penalizes large residuals

Least squares: Robust estimators

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

We minimize $E = \sum_i \rho(u_i ; \sigma)$ $\quad u_i = ax_i + by_i - d$

- u_i = error (residual) of i^th point w.r.t. model parameters β = (a,b,d)
  - ρ = robust function of u_i with scale parameter σ

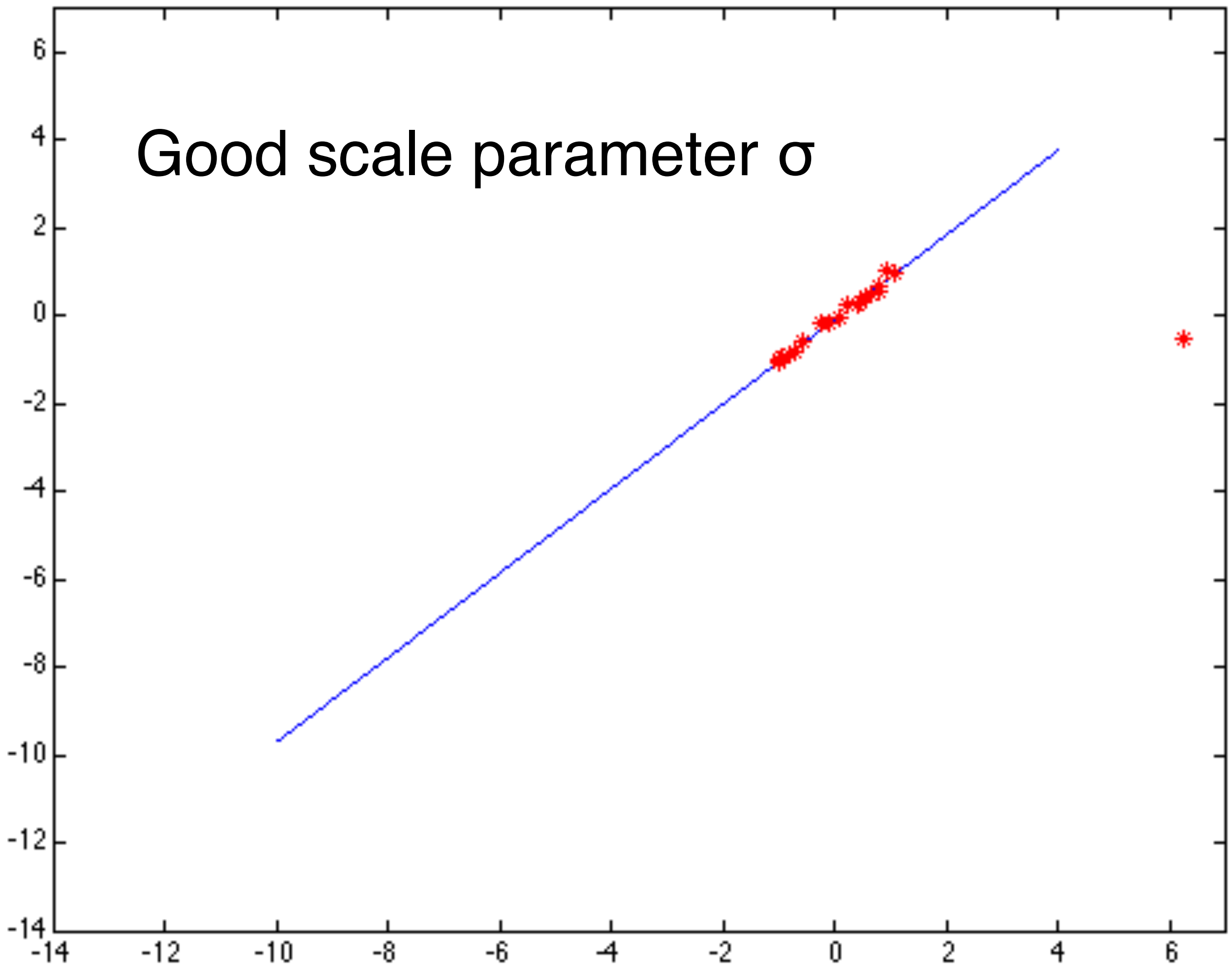


The robust function ρ

- Favors a configuration with small residuals
- Penalizes large residuals

Source: S. Savarese slides.

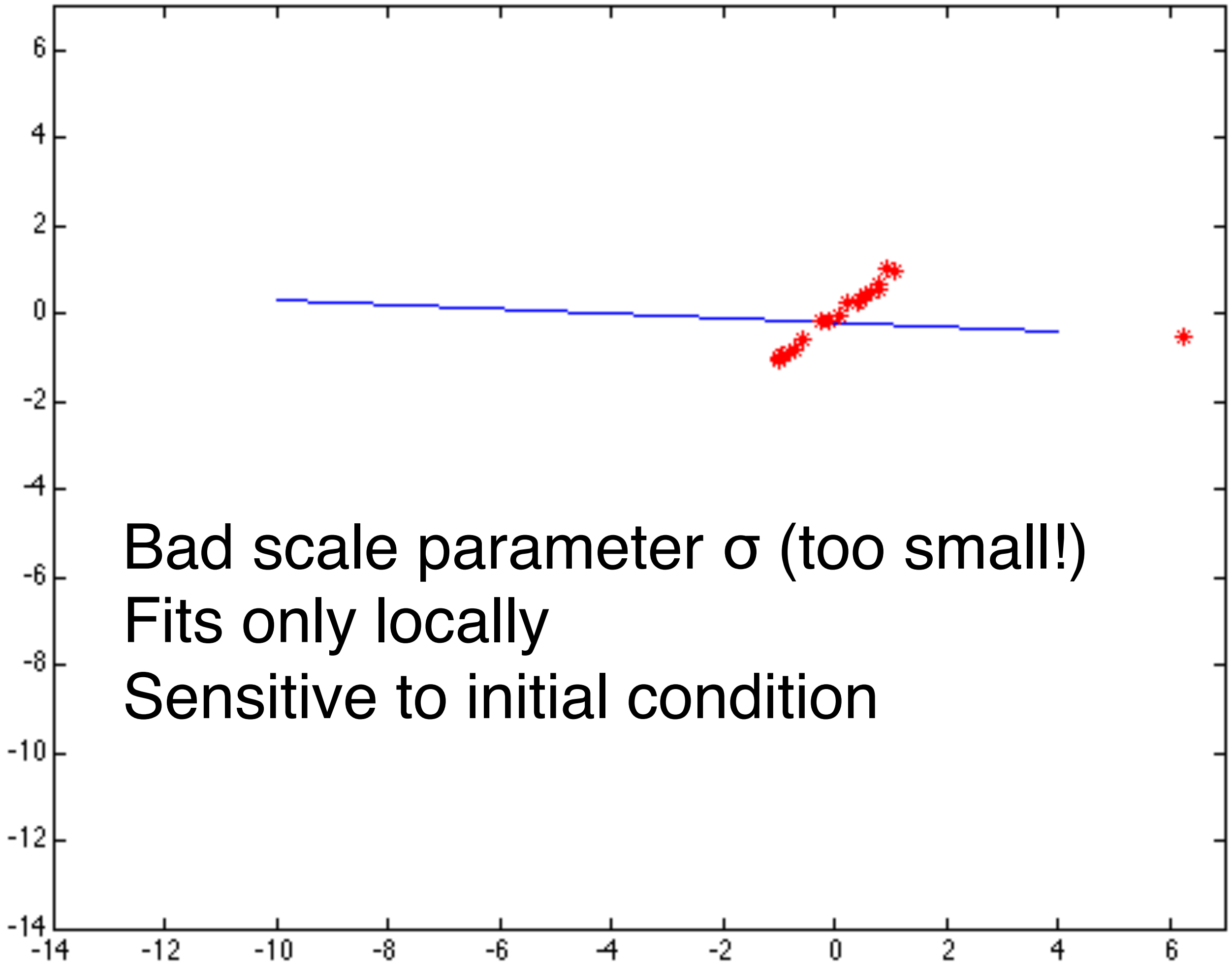
Least squares: Robust estimators



The effect of the outlier is eliminated

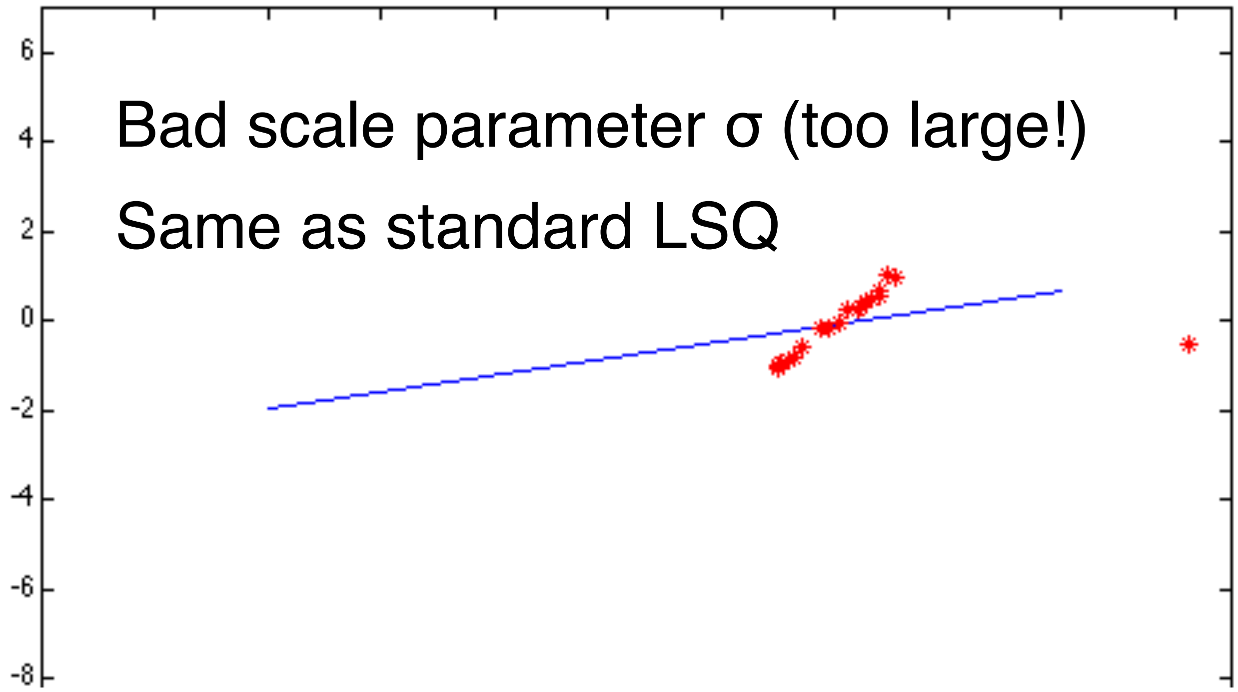
Source: S. Savarese slides.

Least squares: Robust estimators



Source: S. Savarese slides.

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

Source: S. Savarese slides.

Fitting

Goal: Choose a parametric model to fit a certain quantity from data

Techniques:

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

Source: S. Savarese slides.

Basic philosophy
(voting scheme)

- Data elements are used to vote for one (or multiple) models

- Robust to outliers and missing data

- Assumption 1: Noisy features will not vote consistently for any single model ("few" outliers)

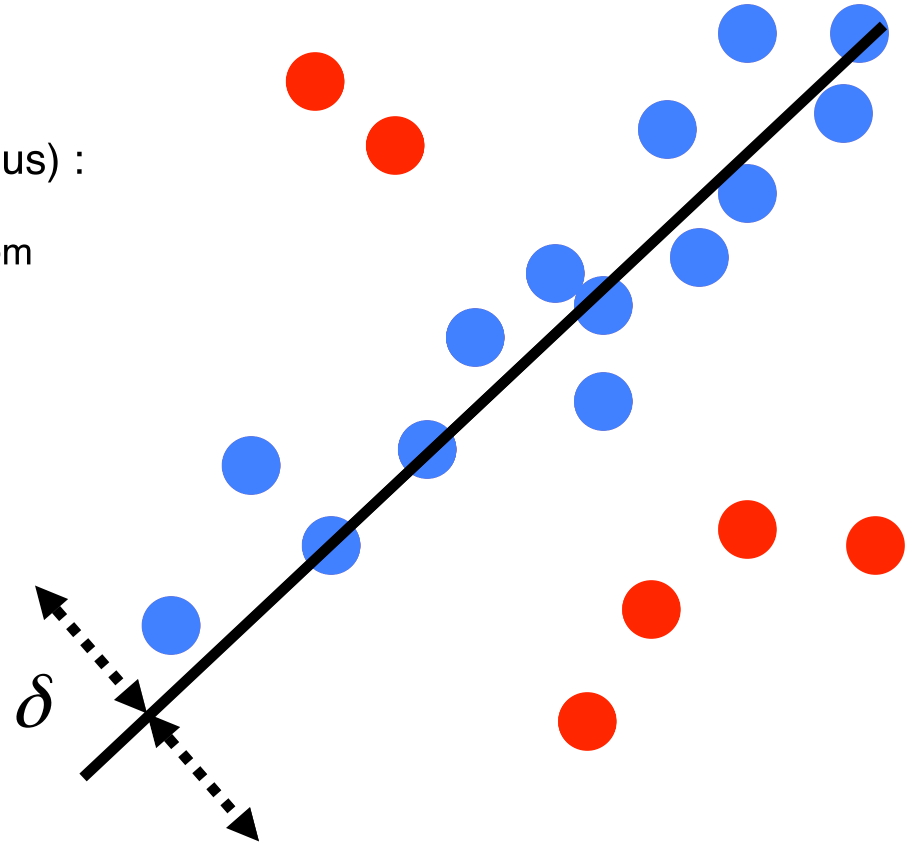
- Assumption 2: there are enough features to agree on a good model ("few" missing data)

Source: S. Savarese slides.

RANSAC

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

Fischler & Bolles in '81.



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

such that:

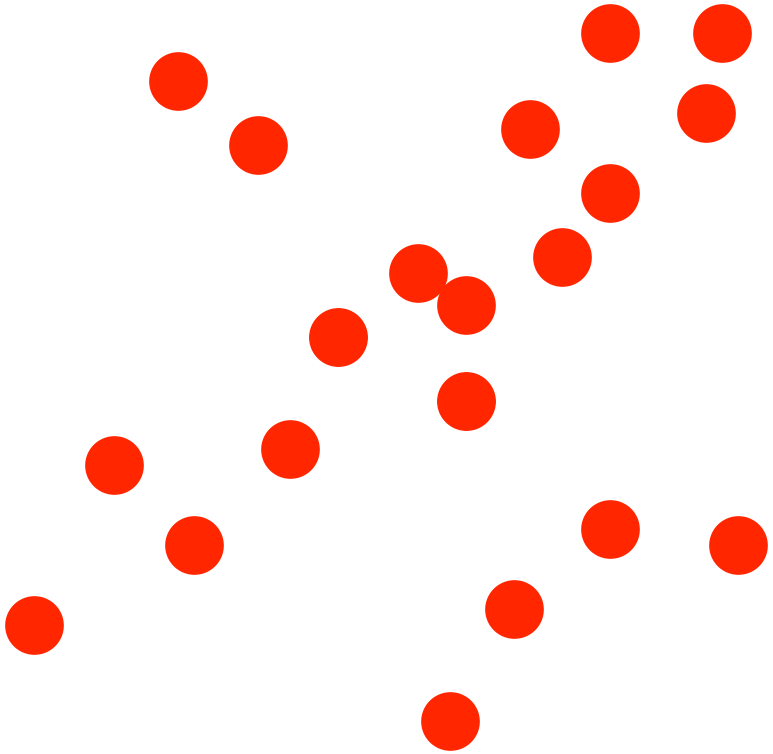
$$f(P, \beta) < \delta$$

Model parameters

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

Source: S. Savarese slides.

RANSAC



Sample set = set of points in 2D

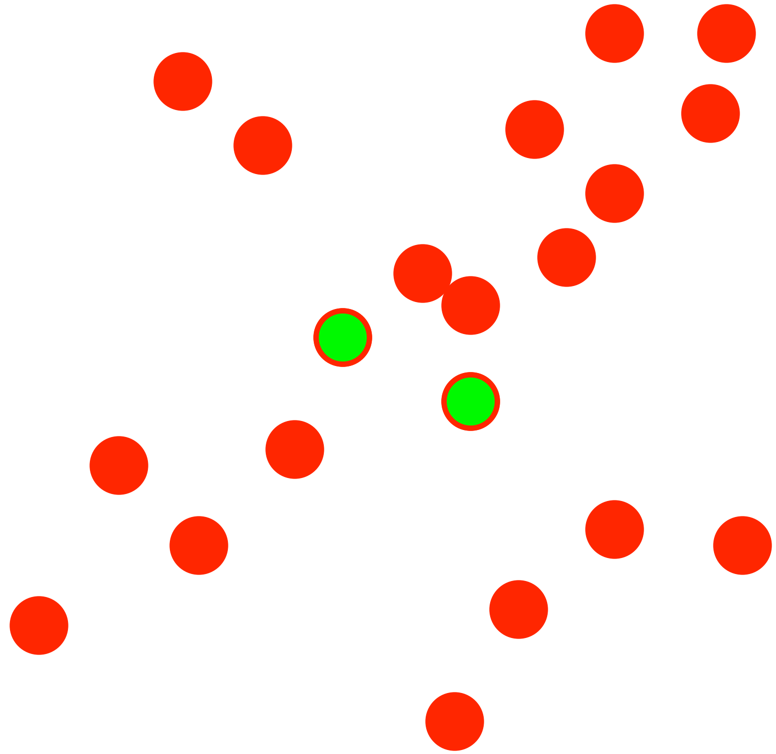
Algorithm:

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

Source: S. Savarese slides.

RANSAC



Sample set = set of points in 2D

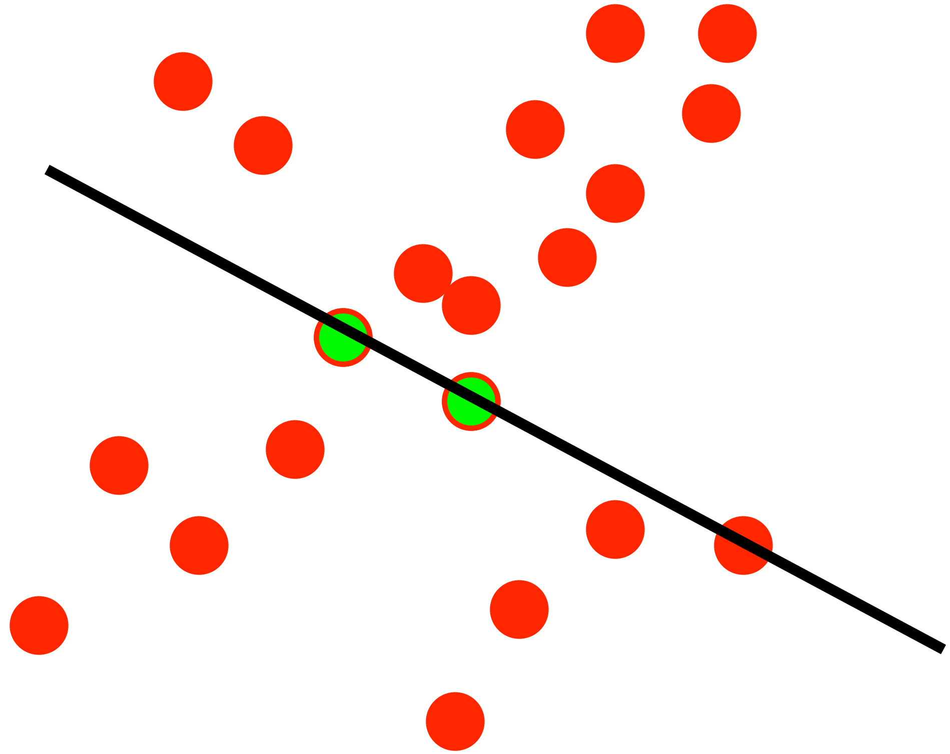
Algorithm:

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

Source: S. Savarese slides.

RANSAC



Sample set = set of points in 2D

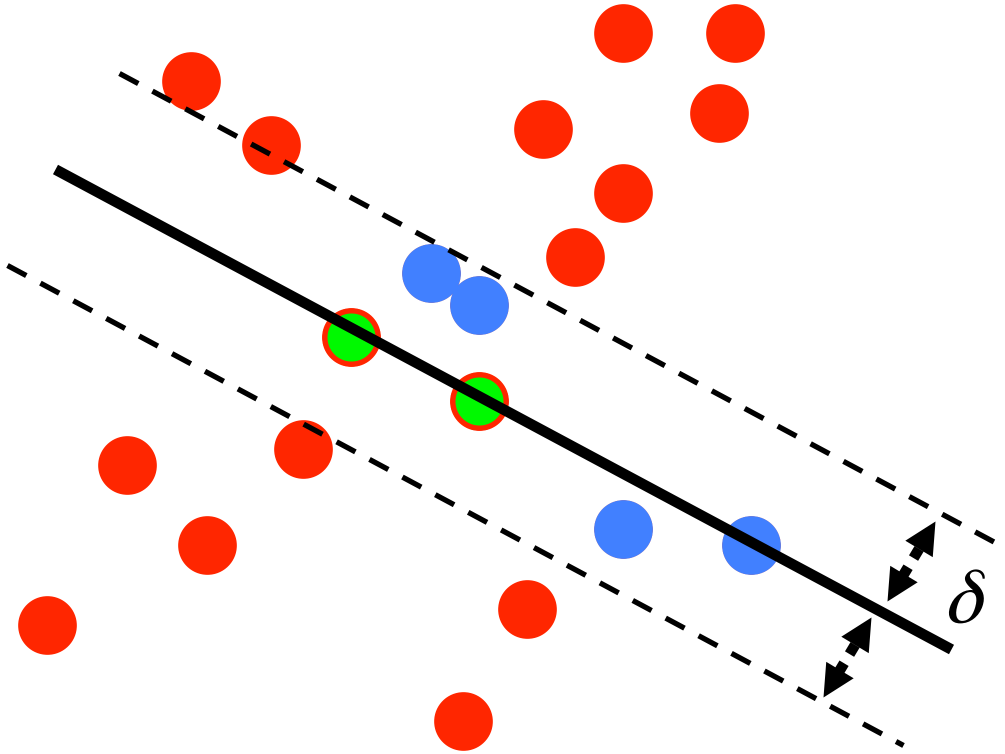
Algorithm:

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

Source: S. Savarese slides.

RANSAC



Sample set = set of points in 2D

$|O| = 14$

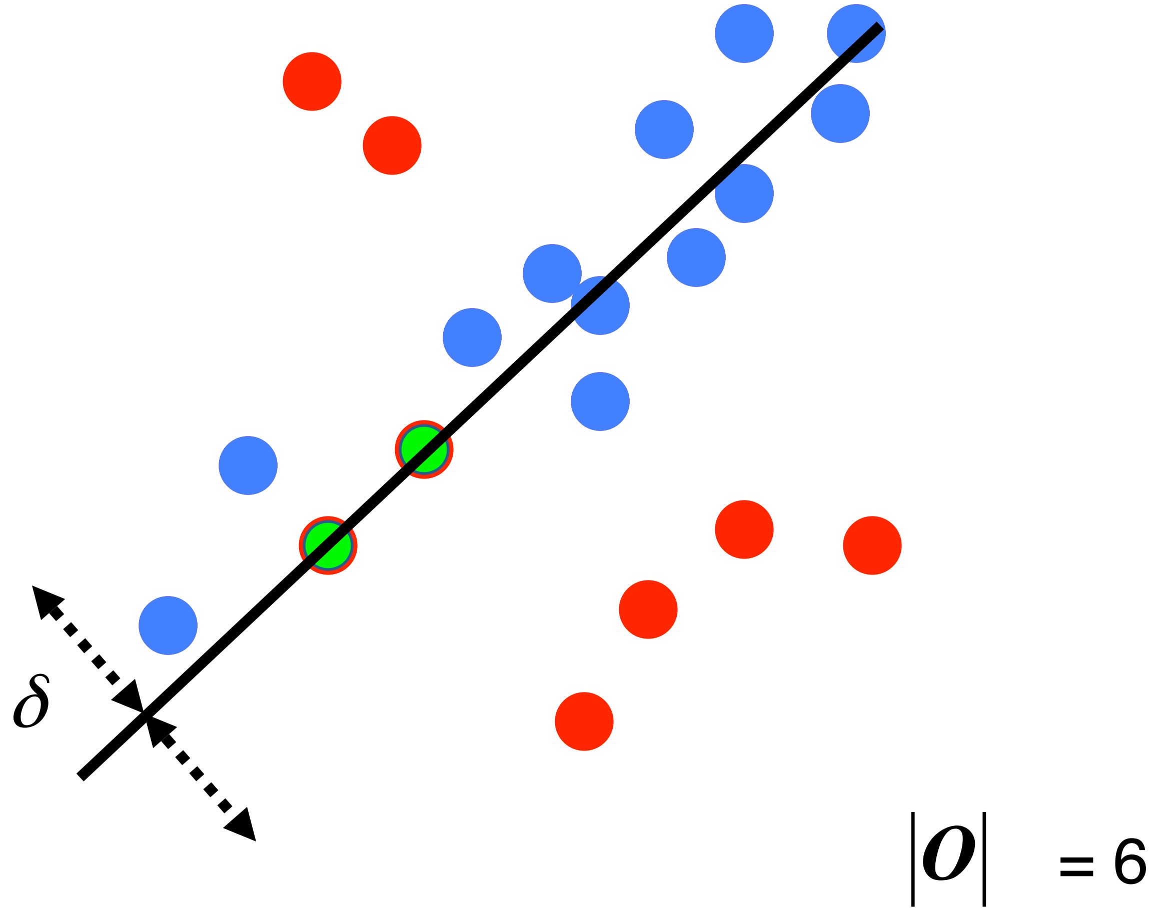
Algorithm:

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

Source: S. Savarese slides.

RANSAC



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

Source: S. Savarese slides.

How many samples?

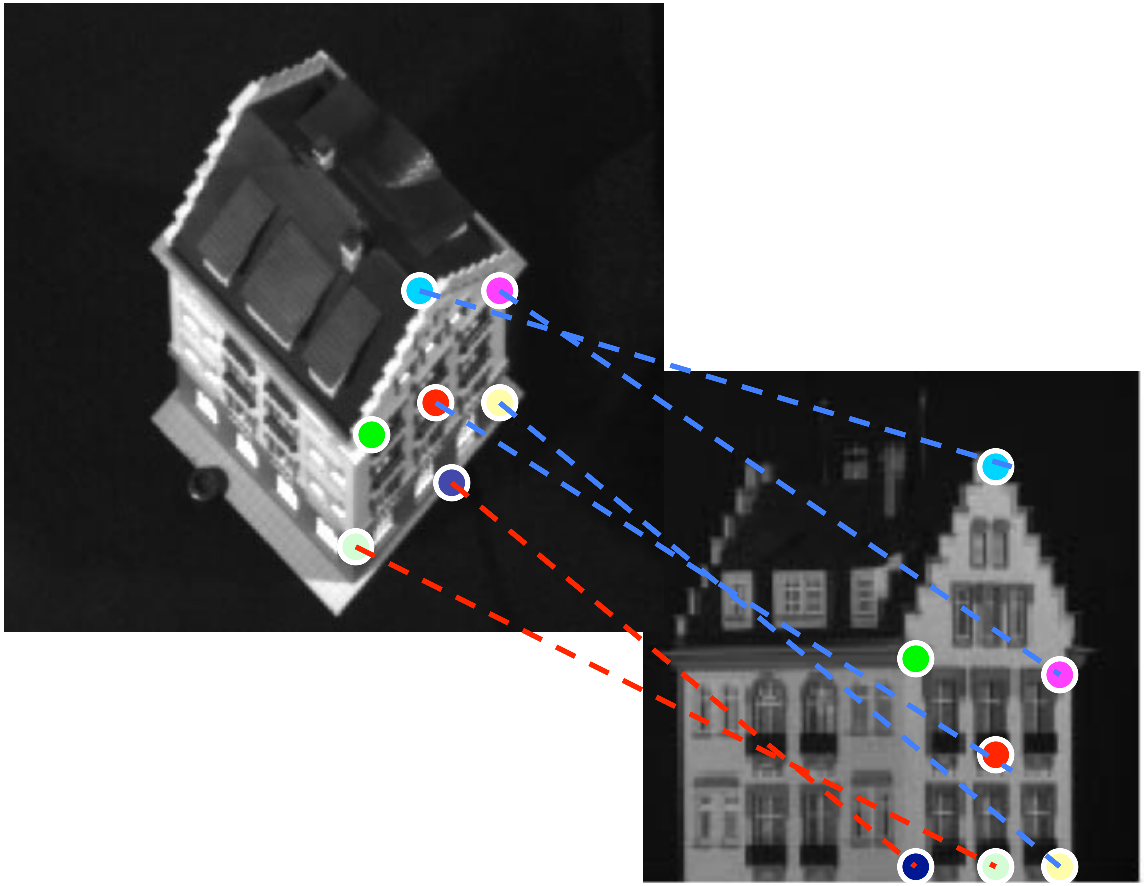
- Number of samples *S*
  - p = probability at least one random sample is valid (free from outliers)
  - *e* = outlier ratio (1-p)
  - P is total probability of success after S trials
  - Likelihood in one trial that all s samples are inliers is p^s
  - s = minimum number needed to fit the model
- Likelihood that S such trials will all fail is $1 - P = (1 - p^s)^S$
- Hence the required number of minimum trials is

	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

$$S = \frac{\log(1-P)}{\log(1-p^s)}$$

Source: S. Savarese slides.

Estimating H by RANSAC



•H → 8 DOF
•Need 4 correspondences

Sample set = set of matches between 2 images

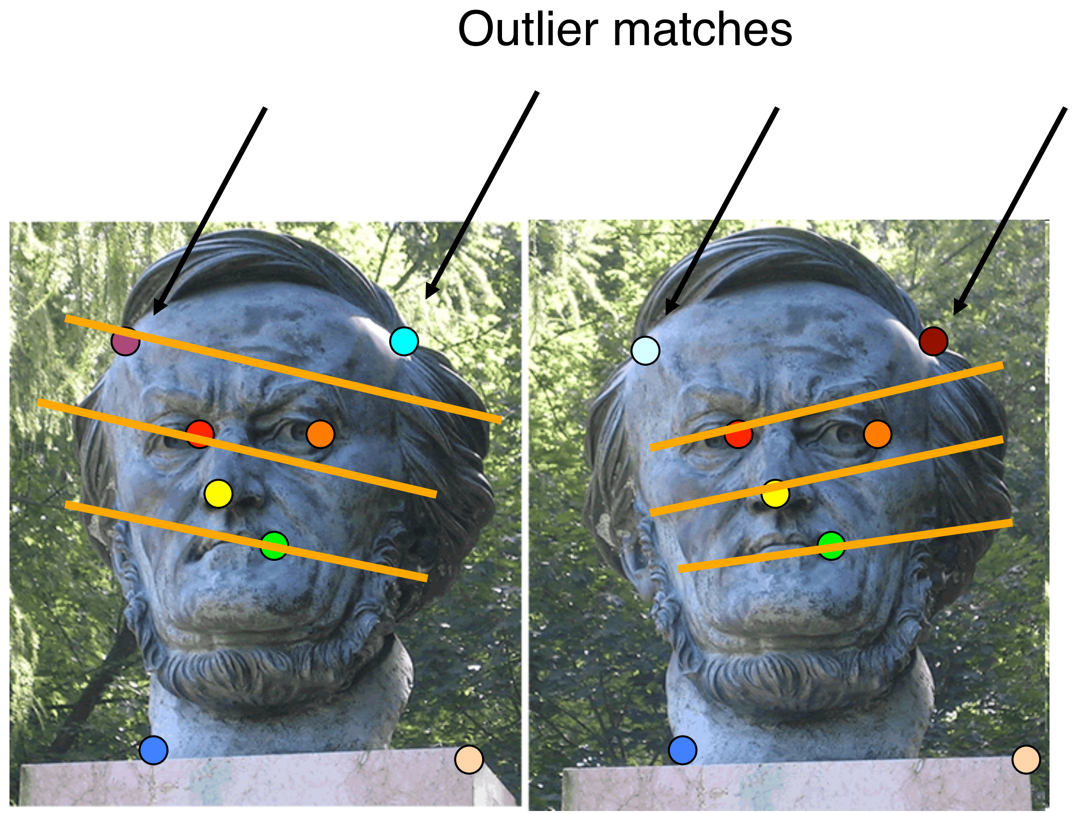
Algorithm:

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

Source: S. Savarese slides.

Estimating F by RANSAC



•F → 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

Source: S. Savarese slides.

RANSAC Conclusions

Good:

- Simple and easily implementable
- Successful in different contexts

Bad:

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

Source: S. Savarese slides.

Fitting

Goal: Choose a parametric model to fit a certain quantity from data

Techniques:

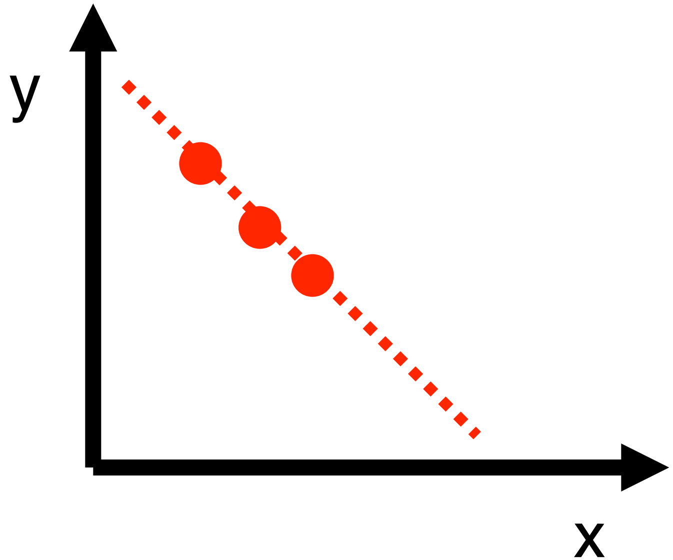
- Least square methods
- RANSAC
- Hough transform

Source: S. Savarese slides.

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 points, find the curve or line that explains the data points best

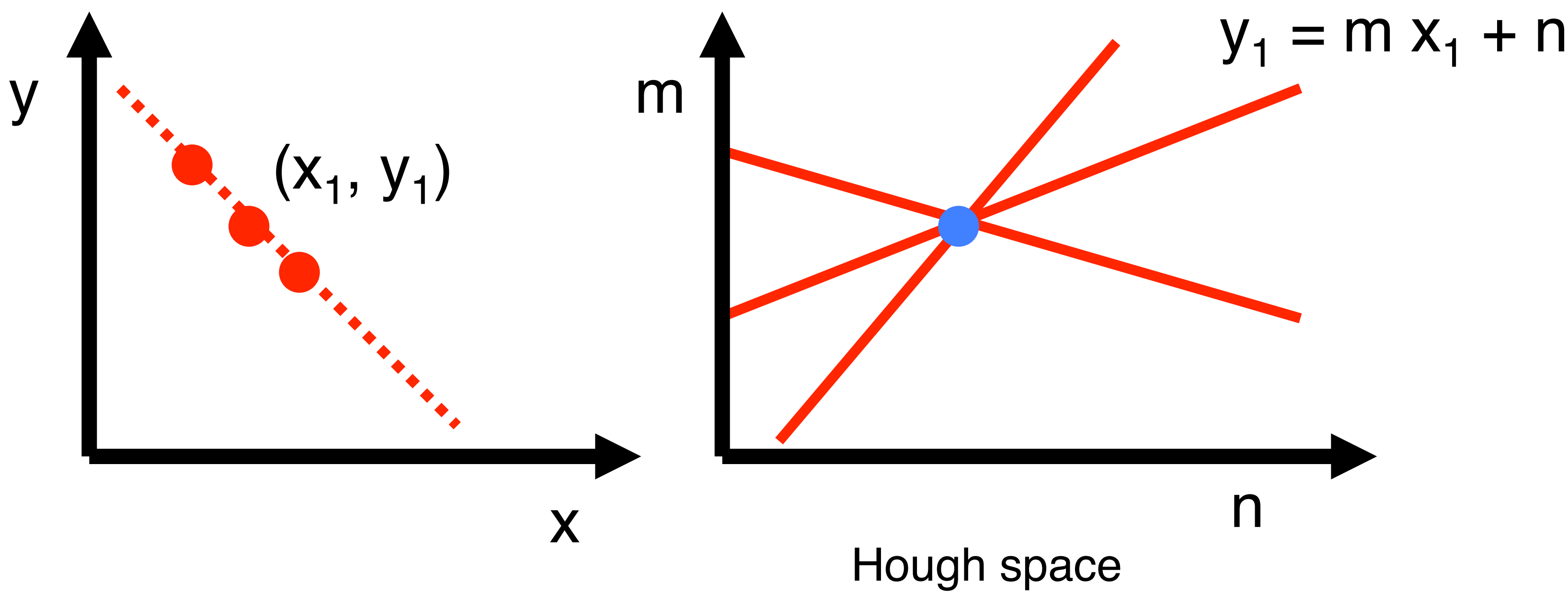


Source: S. Savarese slides.

Hough transform

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

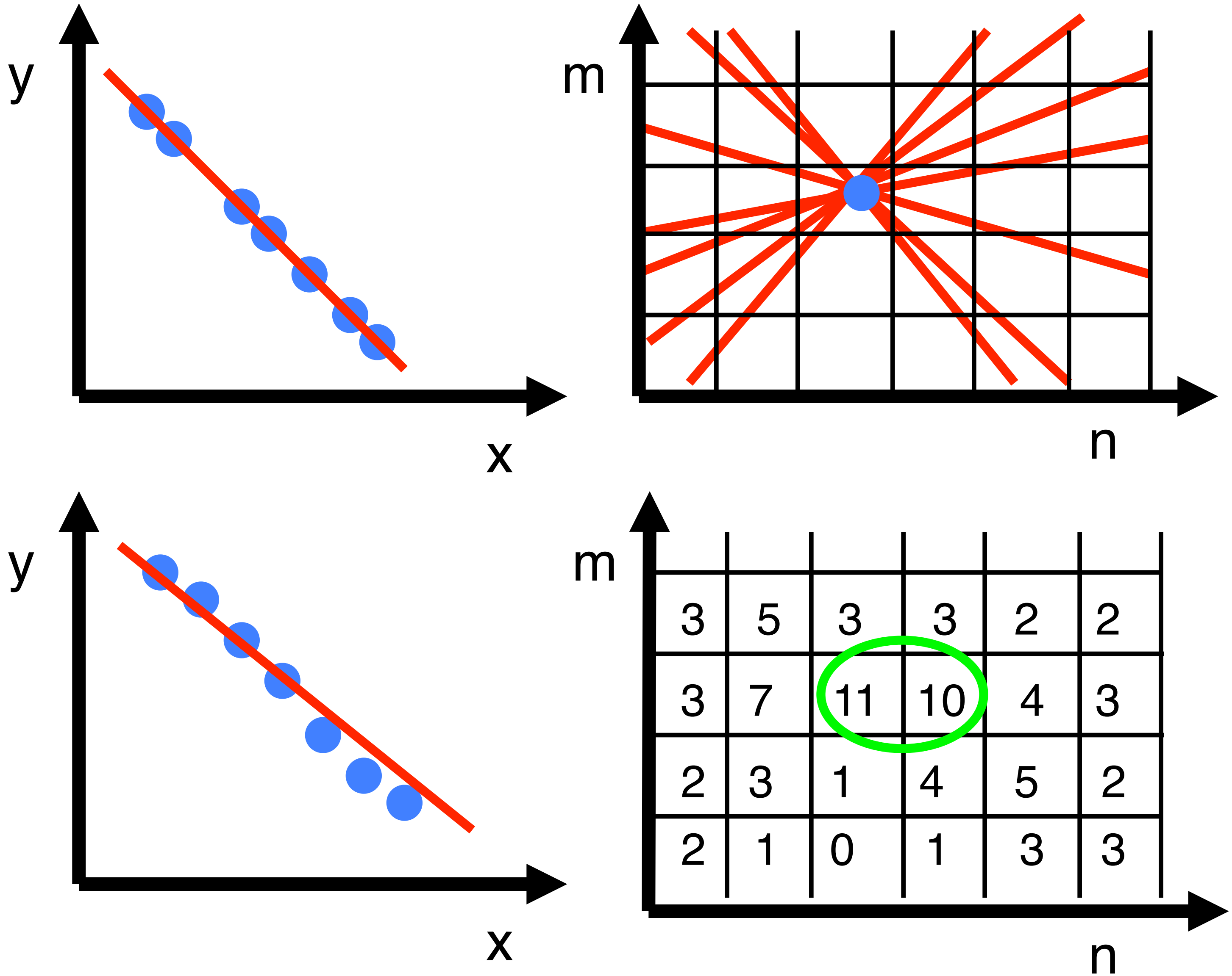
Given a set of points, find the curve or line that explains the data points best



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

Source: S. Savarese slides.

Hough transform



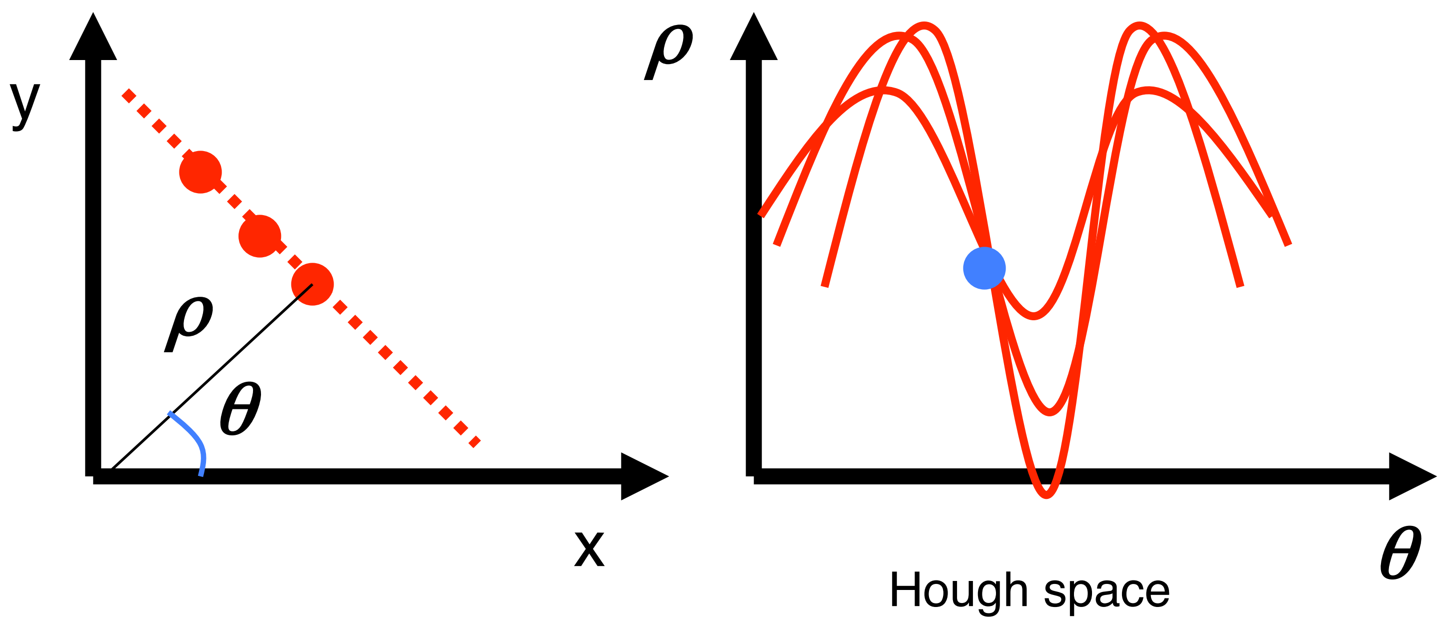
Source: S. Savarese slides.

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…

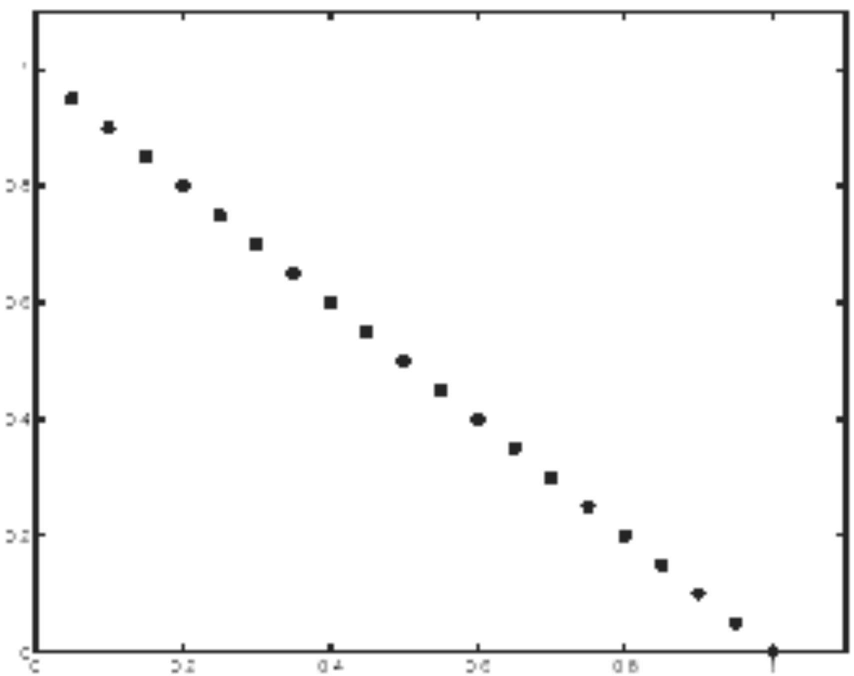
- Use a polar representation for the parameter space



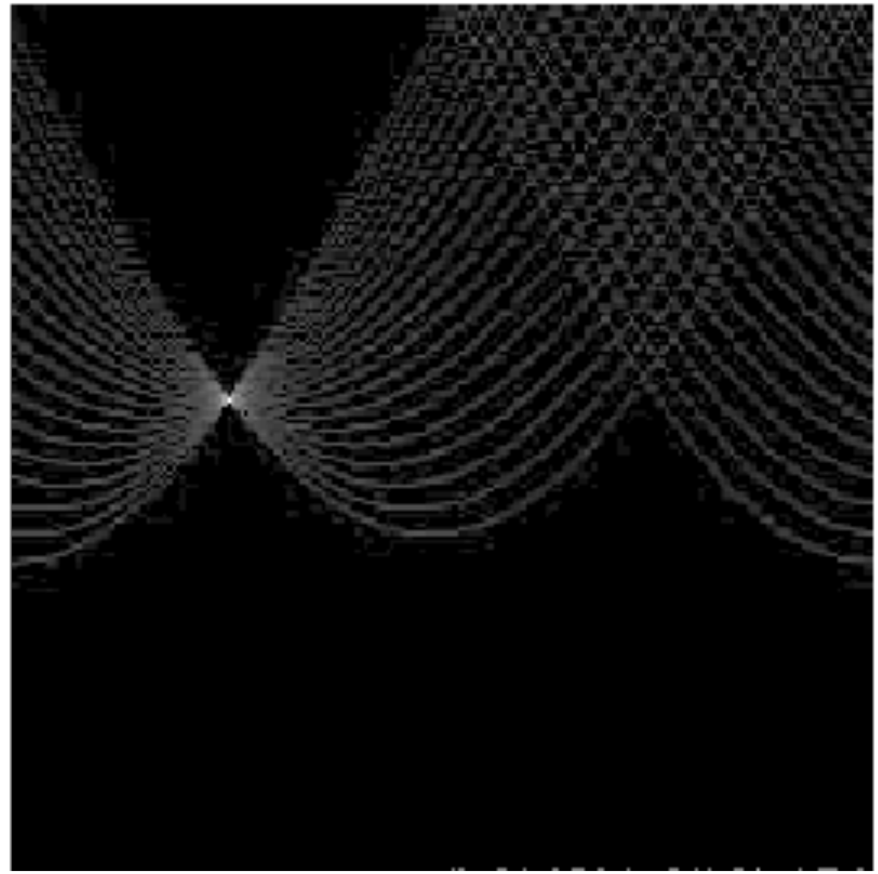
$$x\cos\theta + y\sin\theta = \rho$$

Source: S. Savarese slides.

Hough transform - experiments



features

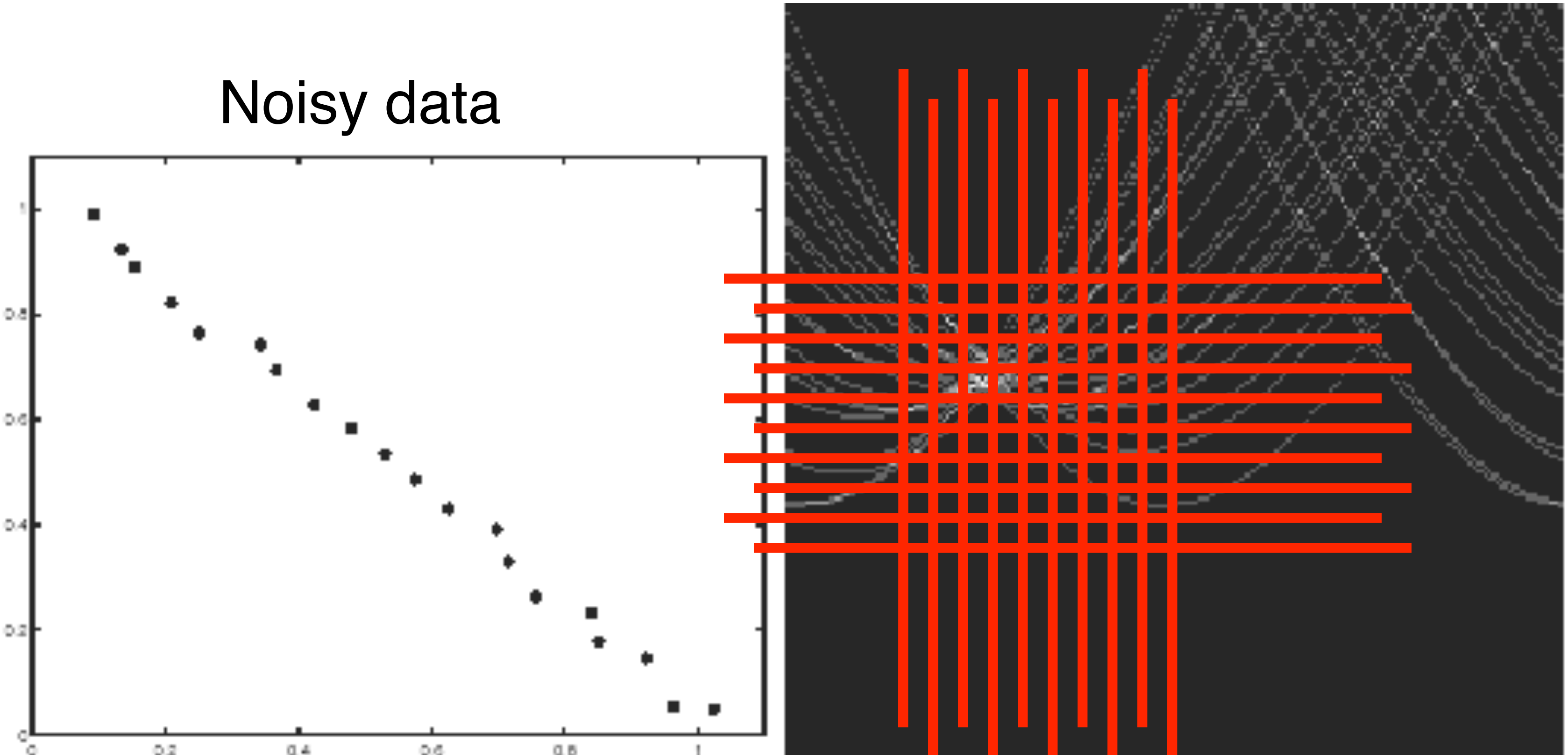


votes

Source: S. Savarese slides.

Hough transform - experiments

Noisy data

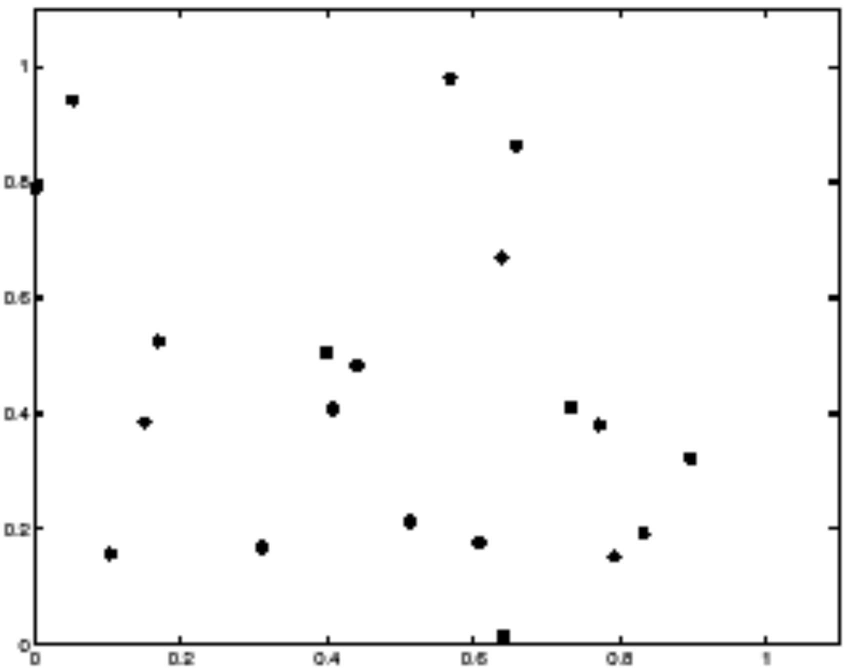


features votes

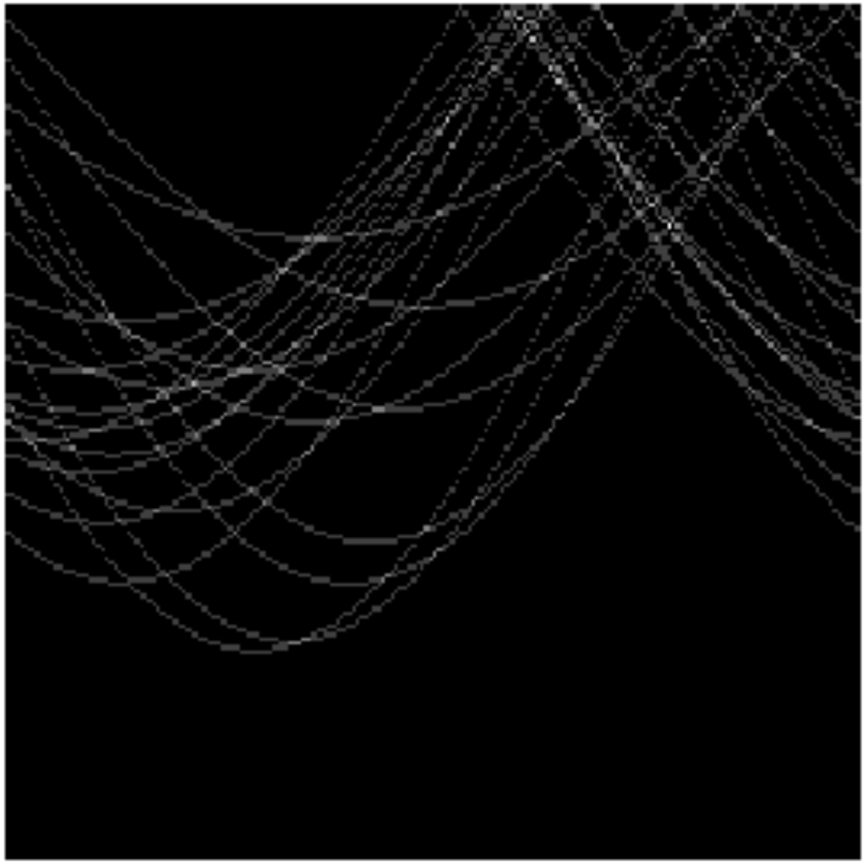
How to compute the intersection point?
IDEA: introduce a grid a count intersection points in each cell
Issue: Grid size needs to be adjusted…

Source: S. Savarese slides.

Hough transform - experiments



features



votes

Issue: spurious peaks due to uniform noise

Source: S. Savarese slides.

Hough transform - conclusions

Good:

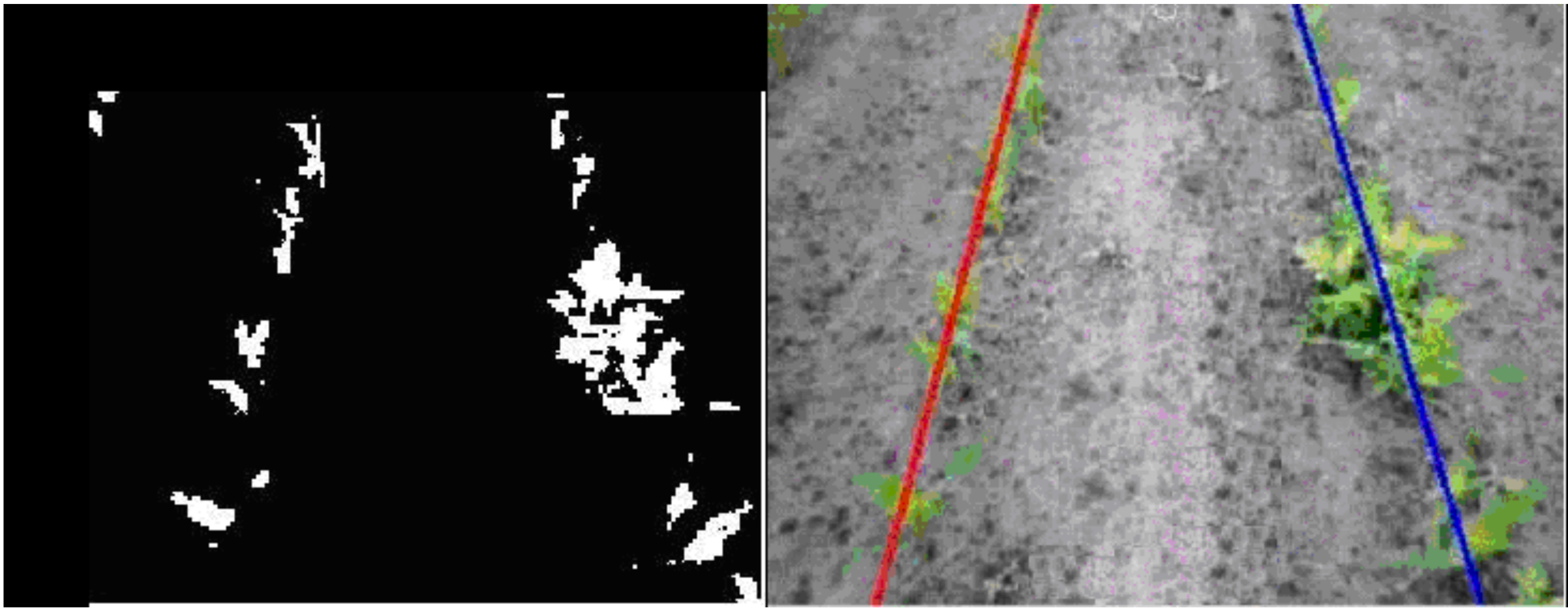
- All points are processed independently, so can cope with occlusion/outliers
- Some robustness to noise: noise points unlikely to contribute consistently to any single bin

Bad:

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

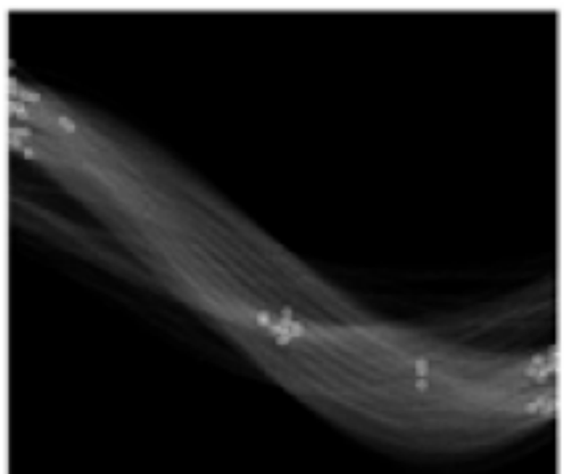
Source: S. Savarese slides.

Hough transform - experiments



Courtesy of TKK Automation Technology Laboratory

Source: S. Savarese slides.



Credit slide: *K. Grauman*

Source: S. Savarese slides.

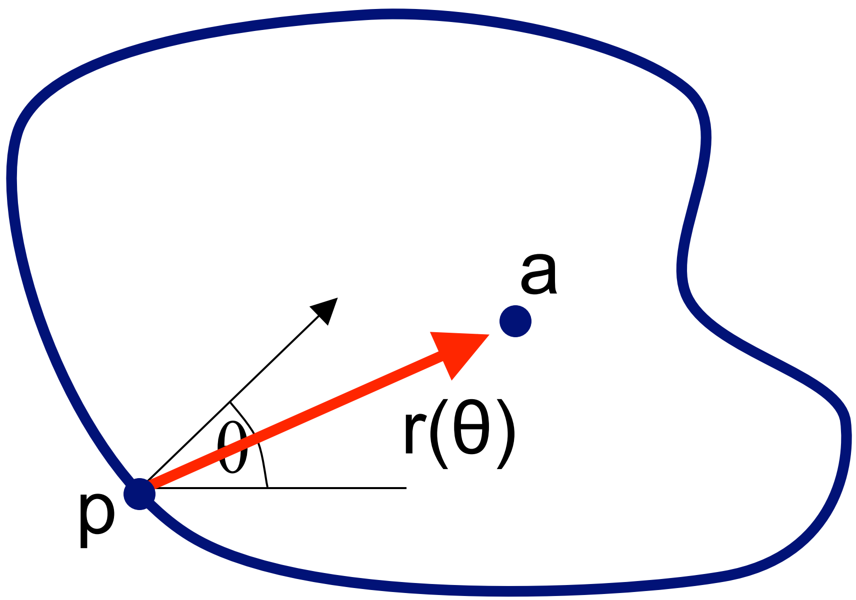
Generalized Hough transform

[more on forthcoming lectures]

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

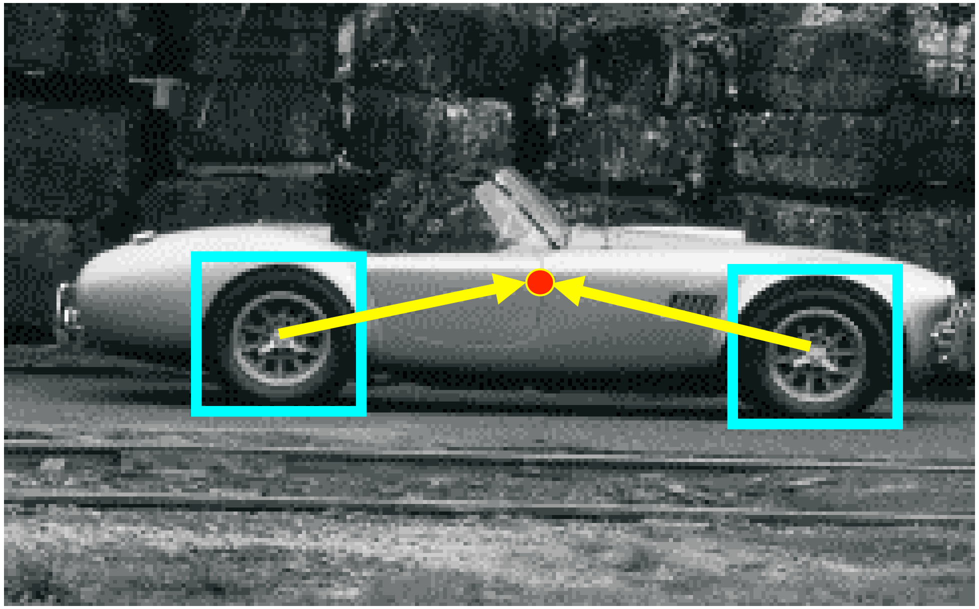
- Measurements: orientation theta, location of p
- Each measurement casts a vote in the Hough space: $p + r(\theta)$



Source: S. Savarese slides.

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



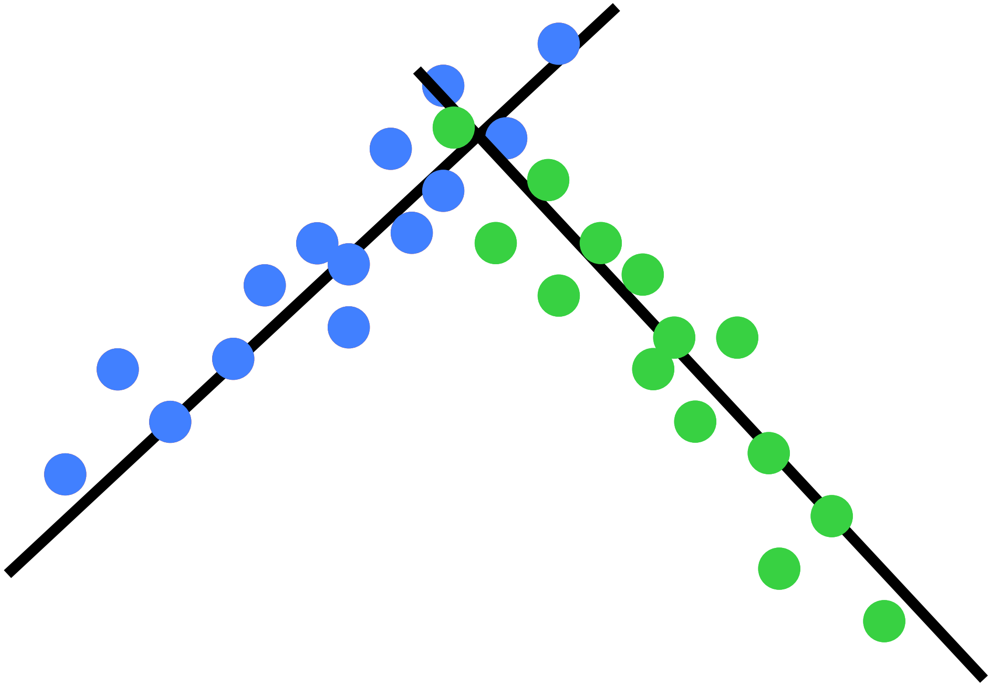
Source: S. Savarese slides.

Plan

- Problem Formulation
- Least Squares Methods
- RANSAC
- Hough Transform
- Multi-model Fitting
- Expectation-Maximization
- Examples of Uses of Fitting

Source: S. Savarese slides.

Fitting multiple models



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

Source: S. Savarese slides.

Incremental line fitting

Scan data point sequentially (using locality constraints)

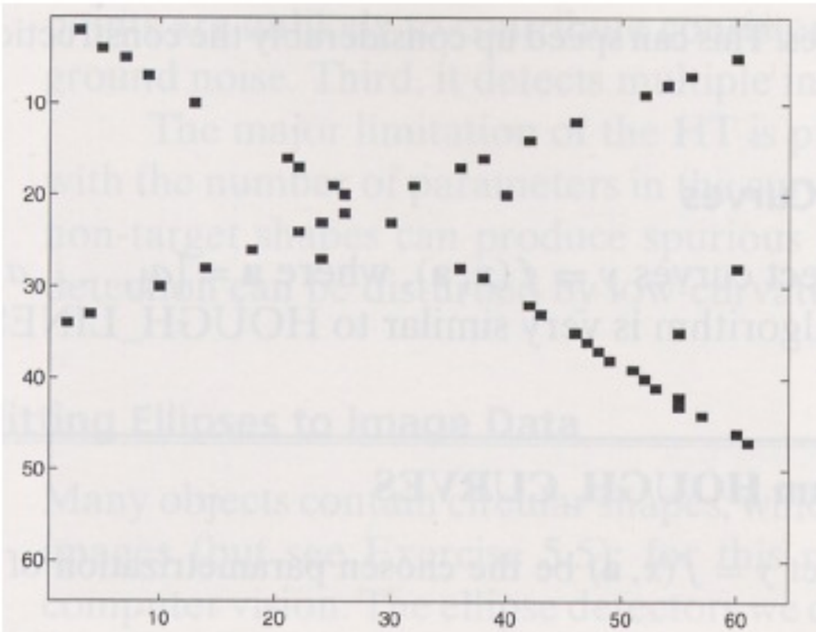
Perform following loop:

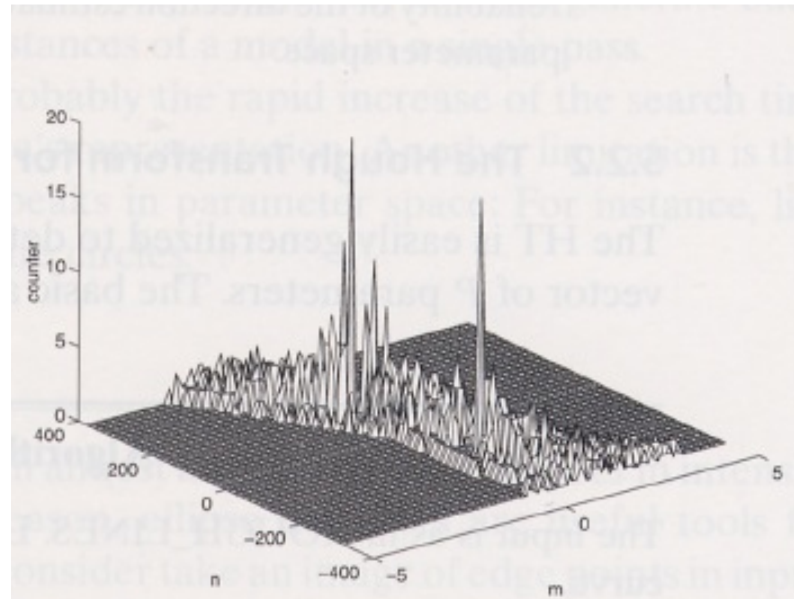
1. Select N point and fit line to N points
2. Compute residual R_N
3. Add a new point, re-fit line and re-compute R_{N+1}
4. Continue while line fitting residual is small enough,

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

Source: S. Savarese slides.

Hough transform





Courtesy of unknown

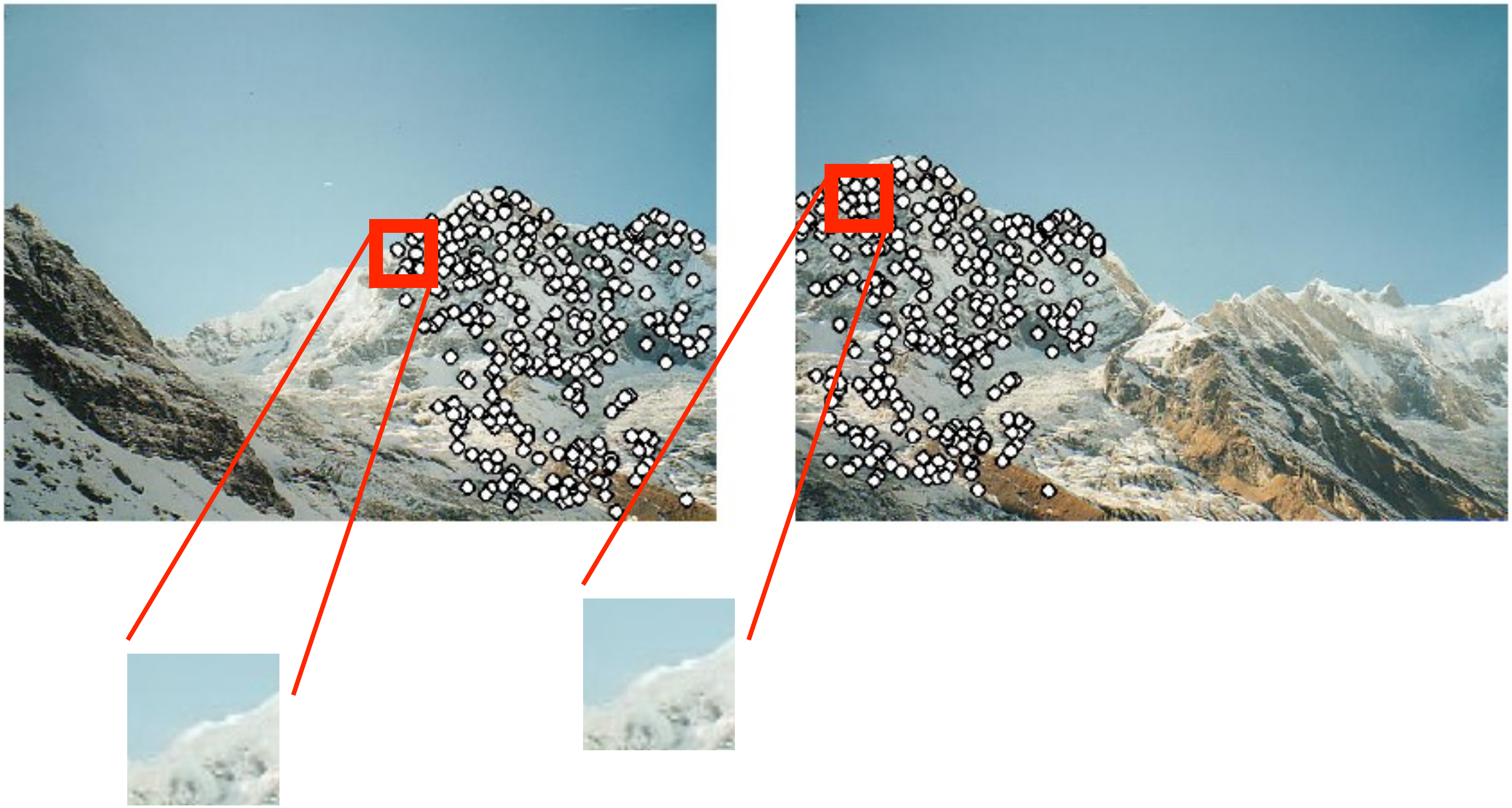
Same cons and pros as before…

Source: S. Savarese slides.

Plan

- Problem Formulation
- Least Squares Methods
- RANSAC
- Hough Transform
- Multi-model Fitting
- Expectation-Maximization
- Examples of Uses of Fitting

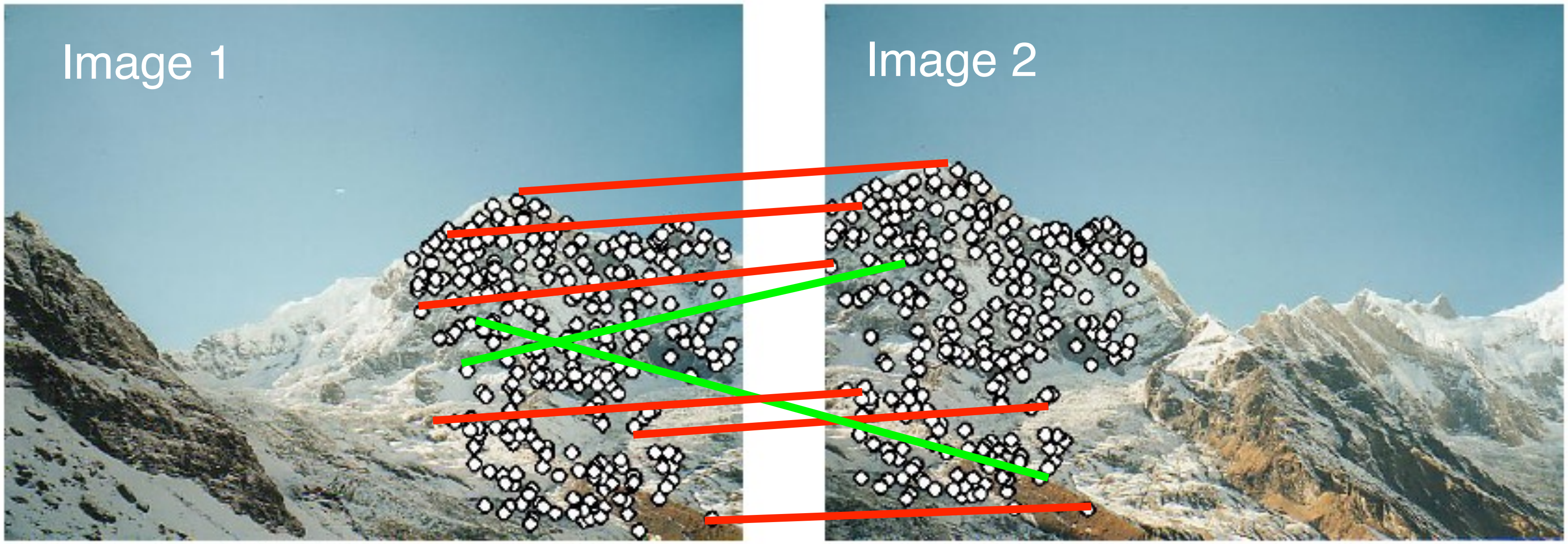
Fitting helps matching!



Feature are matched (for instance, based on correlation)

Source: S. Savarese slides.

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)!

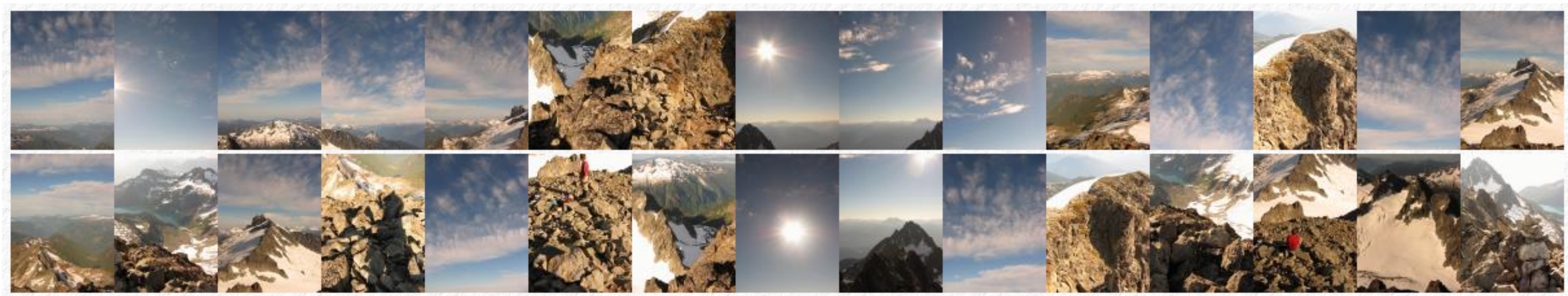
Source: S. Savarese slides.

Fitting helps matching!

Source: S. Savarese slides.

Recognising Panoramas

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



Source: S. Savarese slides

Fitting helps matching!

Images courtesy of Brandon Lloyd

Source: S. Savarese slides.

Source: S. Savarese slides.

Next Lecture: Moving on to Motion Module

- Readings: FP 10.6; SZ 8; TV 8
  - (TV is Trucco and Verri, which is not a required book.)

Least squares methods

- fitting a line -

$$Ax = b$$

- More equations than unknowns

- Look for solution which minimizes $||Ax-b|| = (Ax-b)^T(Ax-b)$
- Solve $\frac{\partial (Ax-b)^T (Ax-b)}{\partial x_i} = 0$

- LS solution

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

Source: S. Savarese slides.

Least squares methods

- fitting a line -

Solving $x = (A^t A)^{-1} A^t b$

$$A^+ = (A^t A)^{-1} A^t$$ = pseudo-inverse of A

$$A = U \sum V^t$$ = SVD decomposition of A

$$A^{-1} = V \sum^{-1} U$$

$$A^+ = V \sum^+ U$$

with $\sum^+$ equal to $\sum^{-1}$ for all nonzero singular values and zero otherwise

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:

$$\mathrm{A\,h} = 0$$

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

Source: S. Savarese slides.