Model Fitting

Faisal Qureshi
faisal.qureshi@uoit.ca

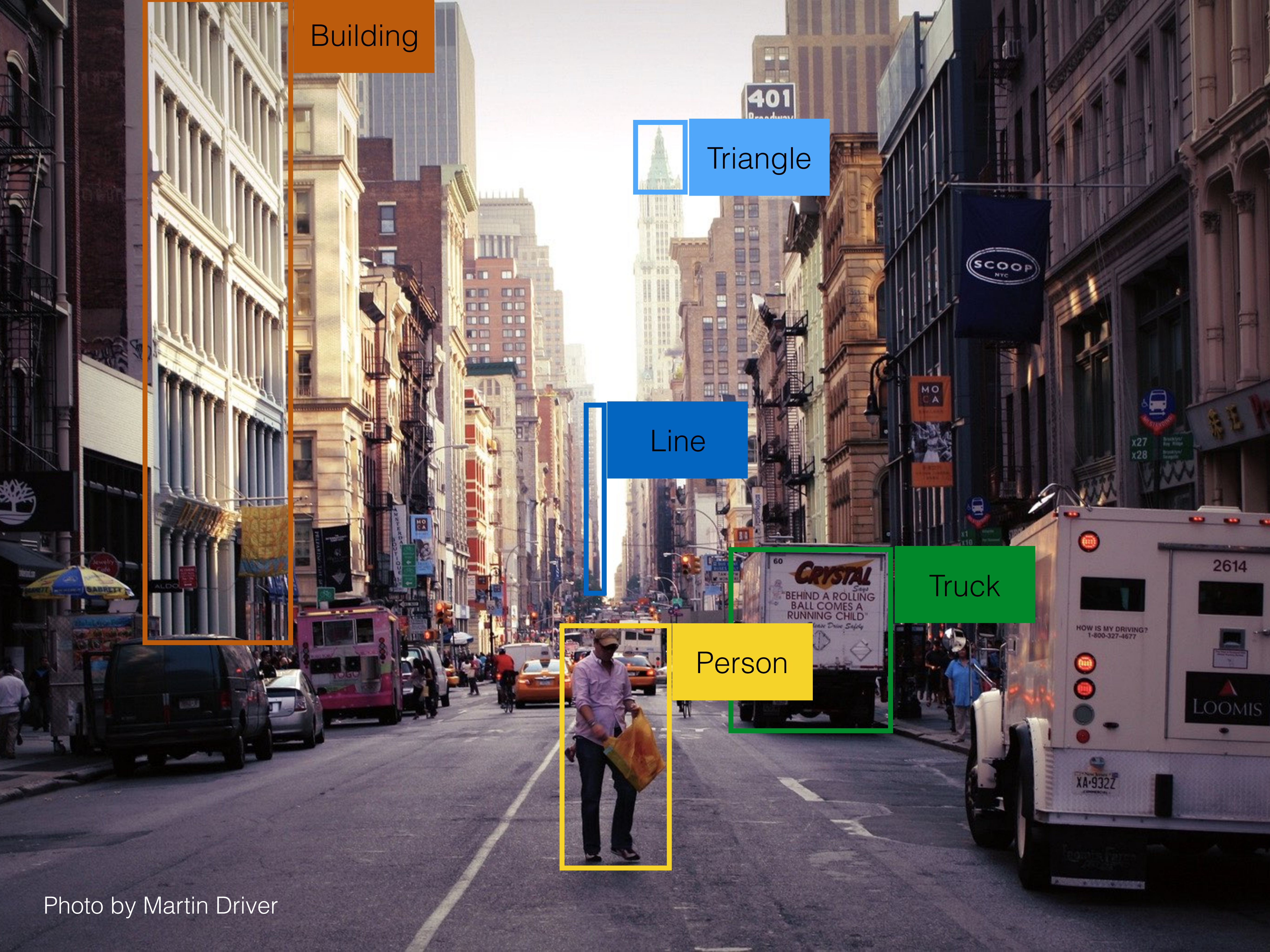


Photo by Martin Driver


Photo by Martin Driver

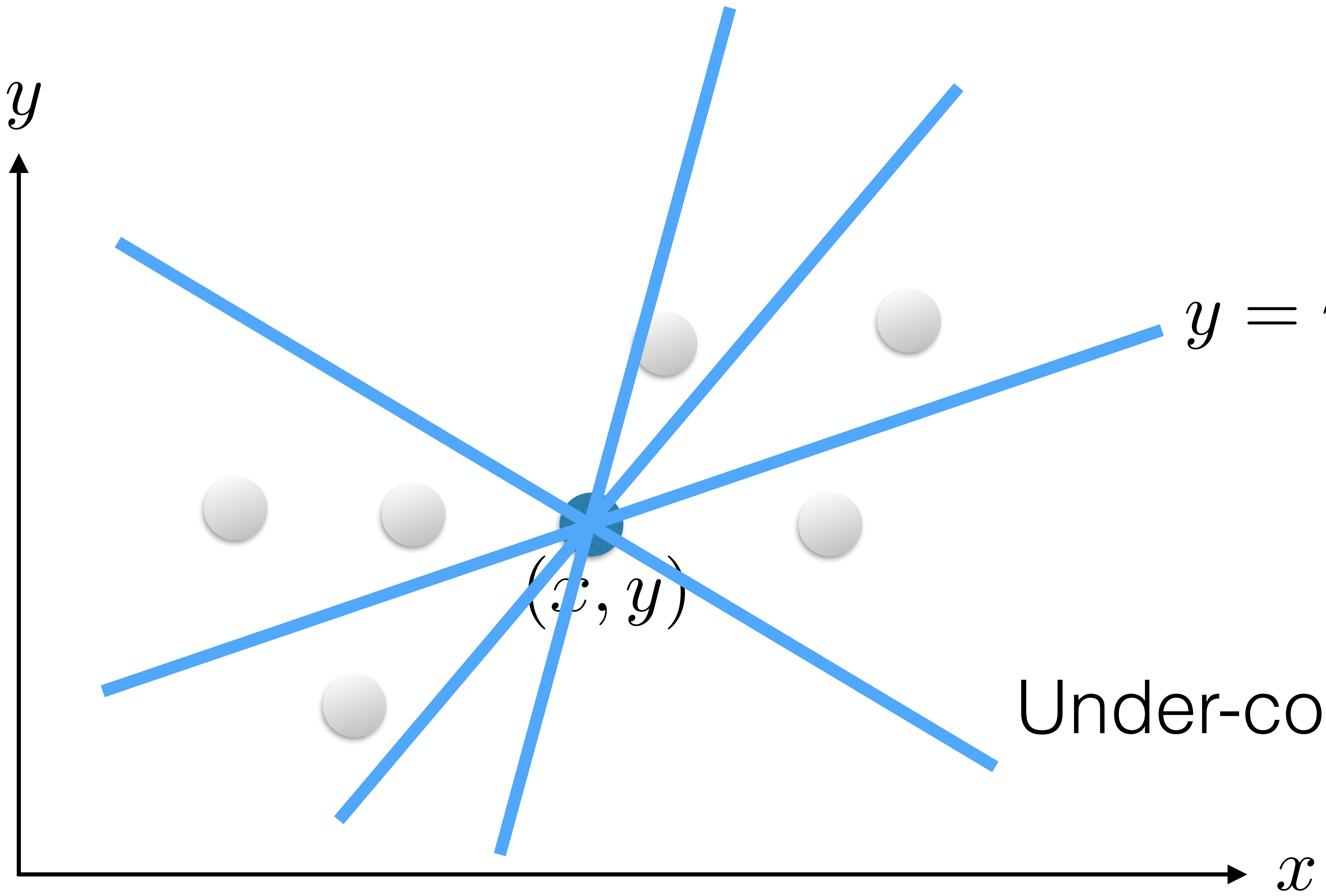
Photo by Martin Driver

Photo by Martin Driver

Model Fitting

Least squares
Robust least squares
RANSAC
Hough Transform
Application: Image Stitching

2D Line Fitting

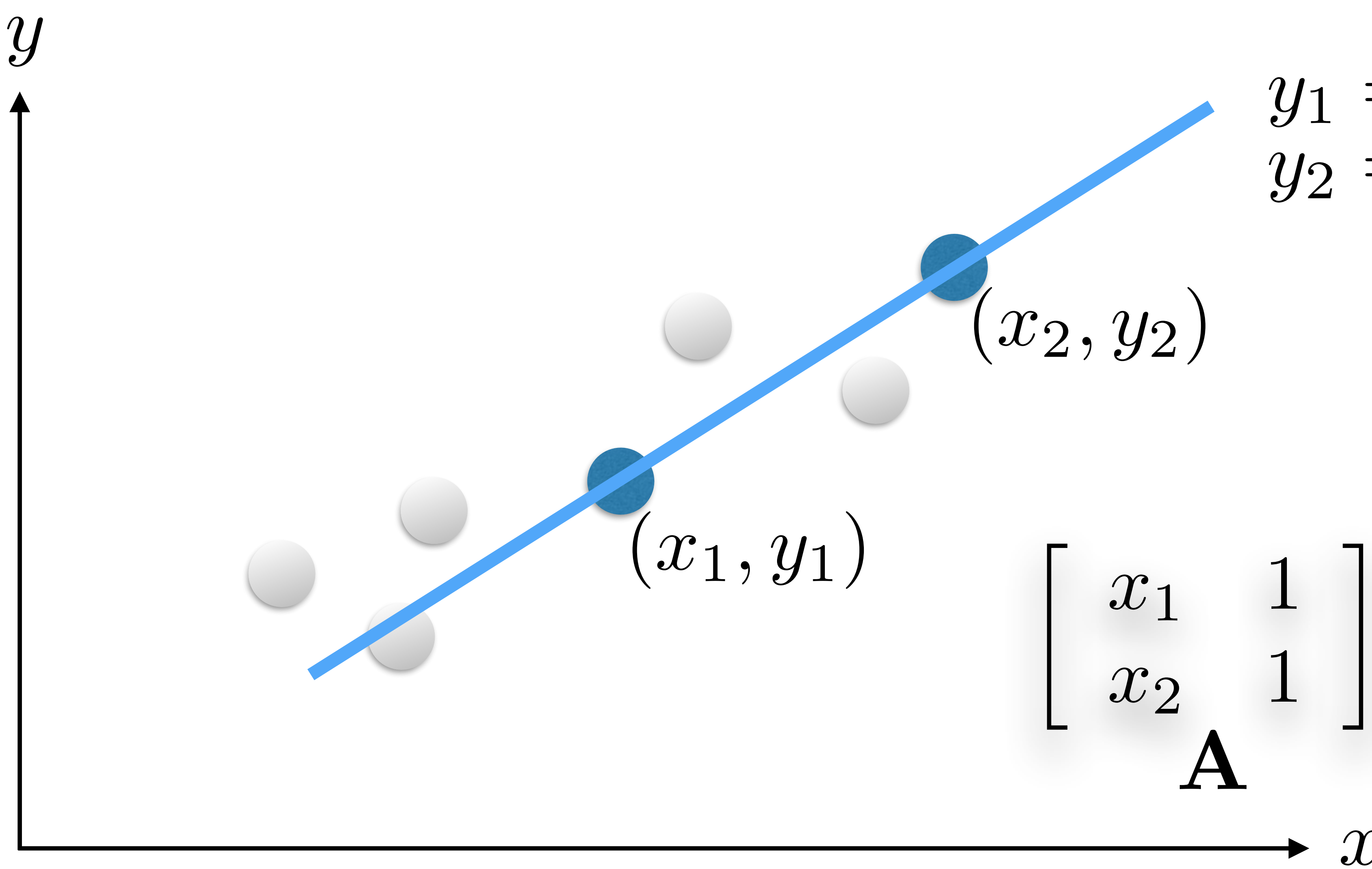


$$y = mx + c$$

Unknowns

Under-constrained system

2D Line Fitting



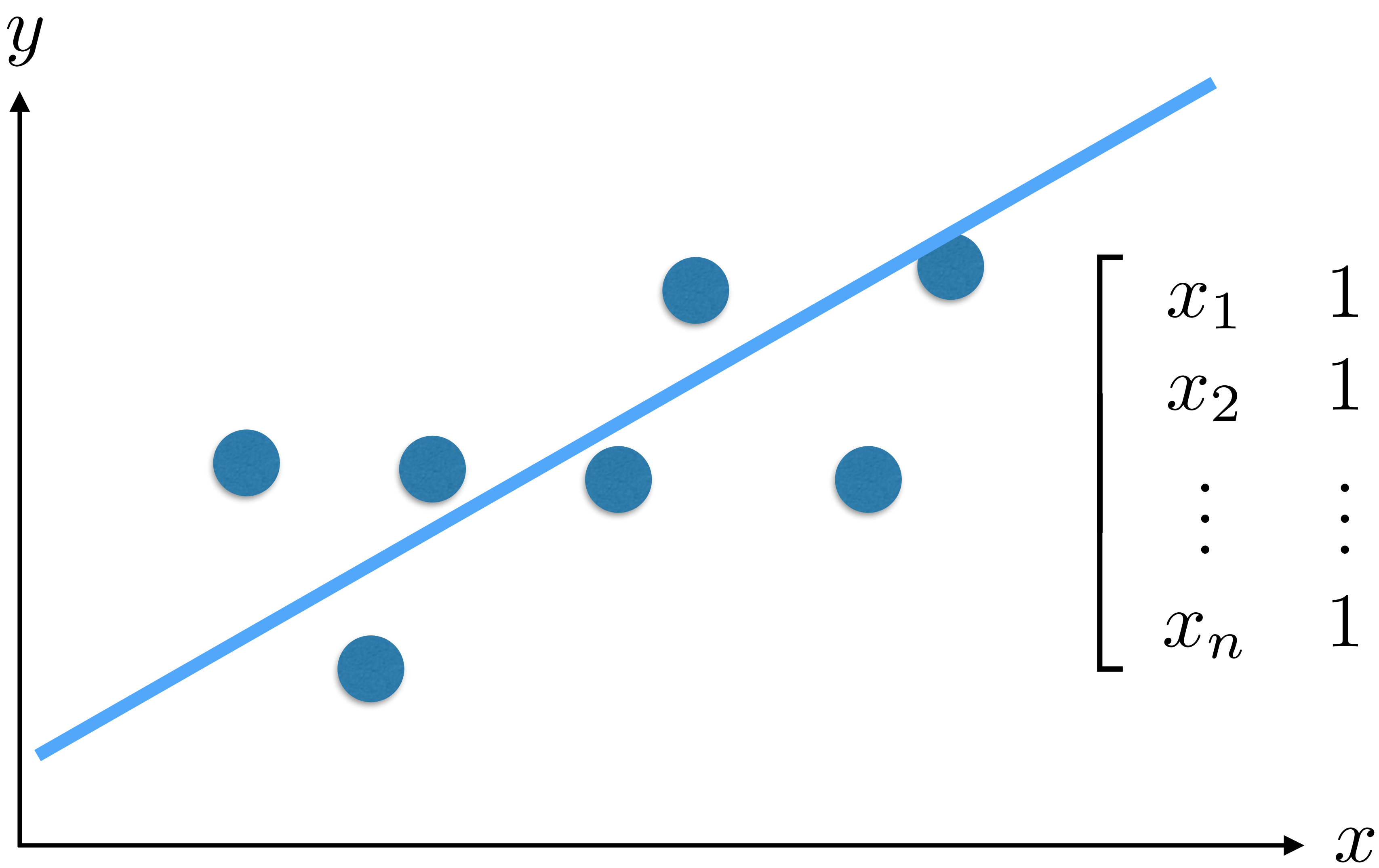
$$y_1 = mx_1 + c$$
$$y_2 = mx_2 + c$$

$$\underset{\mathbf{A}}{\begin{bmatrix} x_1 & 1 \\ x_2 & 1 \end{bmatrix}} \underset{\mathbf{p}}{\begin{bmatrix} m \\ c \end{bmatrix}} = \underset{\mathbf{y}}{\begin{bmatrix} y_1 \\ y_2 \end{bmatrix}}$$

System of linear equations

$$\mathbf{Ap} = \mathbf{y}$$

2D Line Fitting

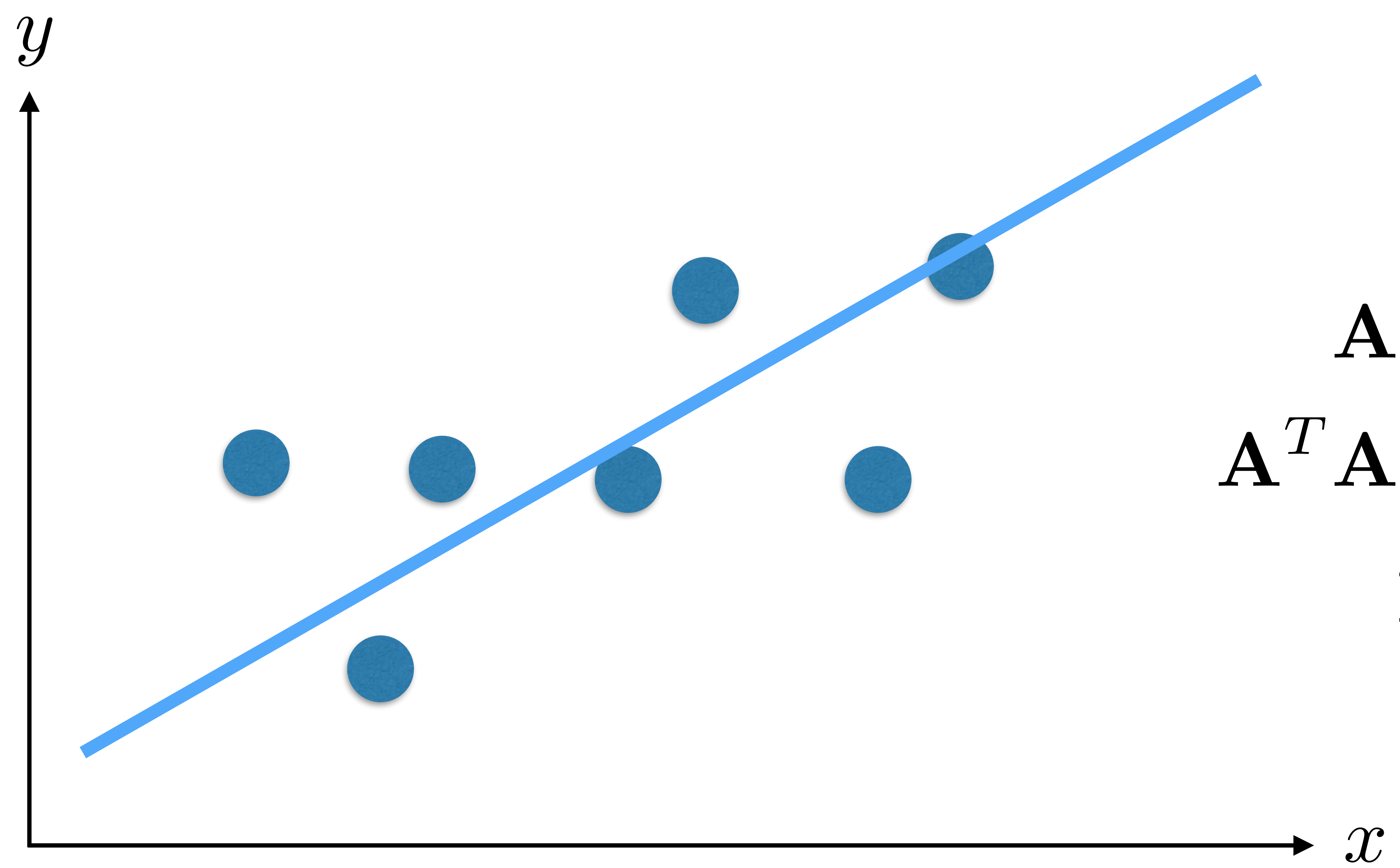


$$\begin{bmatrix} x_1 & 1 \\ x_2 & 1 \\ \vdots & \vdots \\ x_n & 1 \end{bmatrix} \begin{bmatrix} m \\ c \end{bmatrix} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix}$$

$$\mathbf{A}\mathbf{p} = \mathbf{y}$$

Over-constrained system

2D Line Fitting



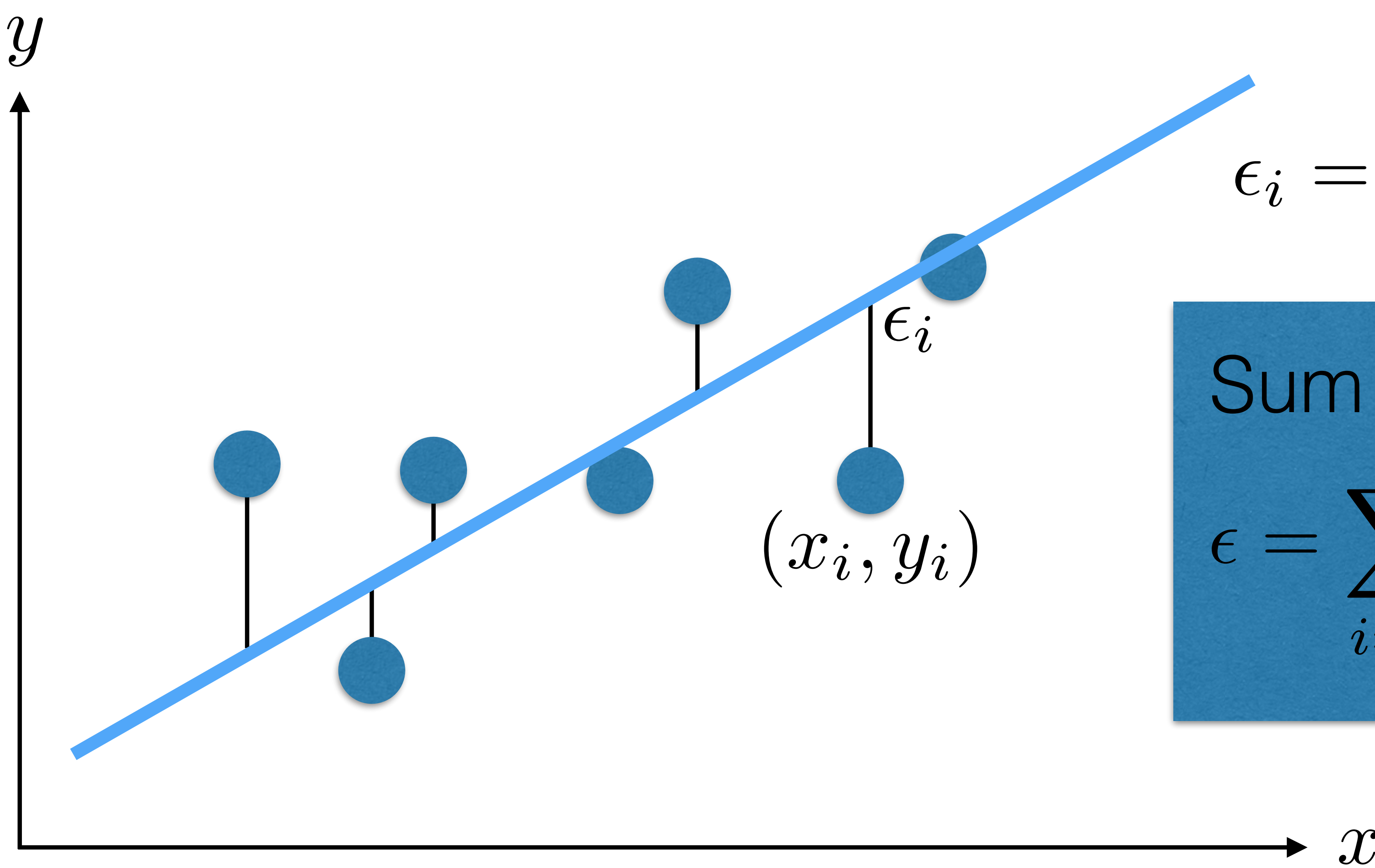
$$\mathbf{A}\mathbf{p} = \mathbf{y}$$

$$\mathbf{A}^T\mathbf{A}\mathbf{p} = \mathbf{A}^T\mathbf{y}$$

$$\mathbf{p} = \boxed{(\mathbf{A}^T\mathbf{A})^{-1}}\mathbf{A}^T\mathbf{y}$$

Pseudo Inverse

Least Squares



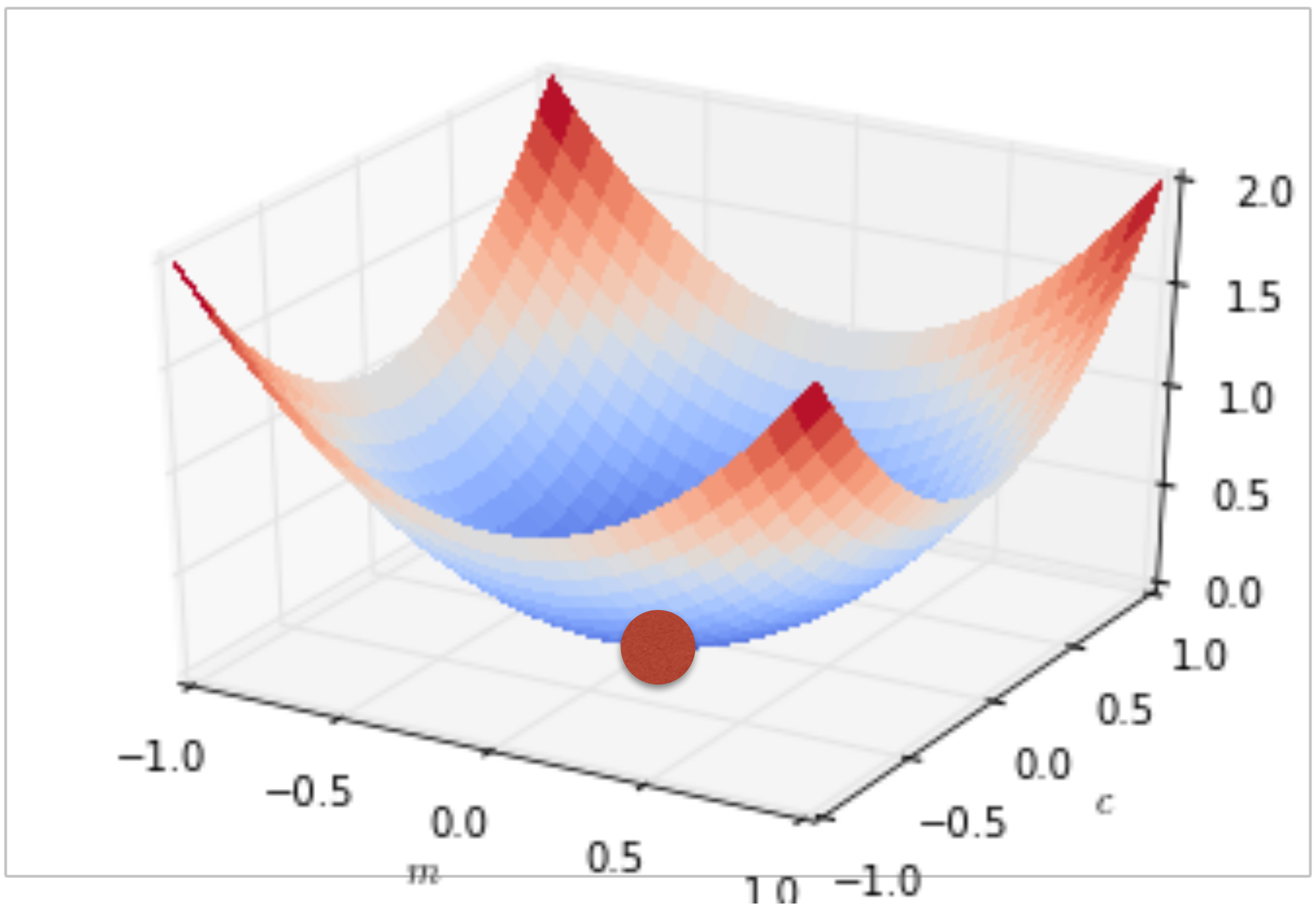
$$\epsilon_i = y_i - mx_i - c$$

Sum of Squared Errors

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

Find the line that minimizes the overall sum of squared errors

Minimize $$\epsilon = \sum_{i=1}^{n} (y_i - mx_i - c)^2$$



Set partial derivatives equal to 0 and solve for m and c

$$\frac{\partial \epsilon}{\partial m} = 0$$

$$\frac{\partial \epsilon}{\partial c} = 0$$

Minimize $\quad \epsilon = \sum_{i=1}^{n} (y_i - mx_i - c)^2$

$$\epsilon = \left\| \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix} - \begin{bmatrix} x_1 & 1 \\ x_2 & 1 \\ \vdots & \vdots \\ x_n & 1 \end{bmatrix} \begin{bmatrix} m \\ c \end{bmatrix} \right\|^2$$

$$= \|\mathbf{y} - \mathbf{A}\mathbf{p}\|^2$$

$$= \mathbf{y}^T\mathbf{y} - 2(\mathbf{A}\mathbf{p})^T\mathbf{y} + (\mathbf{A}\mathbf{p})^T\mathbf{A}\mathbf{p}$$

Minimize $\quad \epsilon = \sum_{i=1}^{n} (y_i - mx_i - c)^2$

$$\epsilon = \mathbf{y}^T\mathbf{y} - 2(\mathbf{A}\mathbf{p})^T\mathbf{y} + (\mathbf{A}\mathbf{p})^T\mathbf{A}\mathbf{p}$$

$$\frac{\partial \epsilon}{\partial \mathbf{p}} = -2\mathbf{A}^T\mathbf{y} + 2\mathbf{A}^T\mathbf{A}\mathbf{p}$$

Set $\frac{\partial \epsilon}{\partial \mathbf{p}}$ to 0 and solve for $\mathbf{p}$

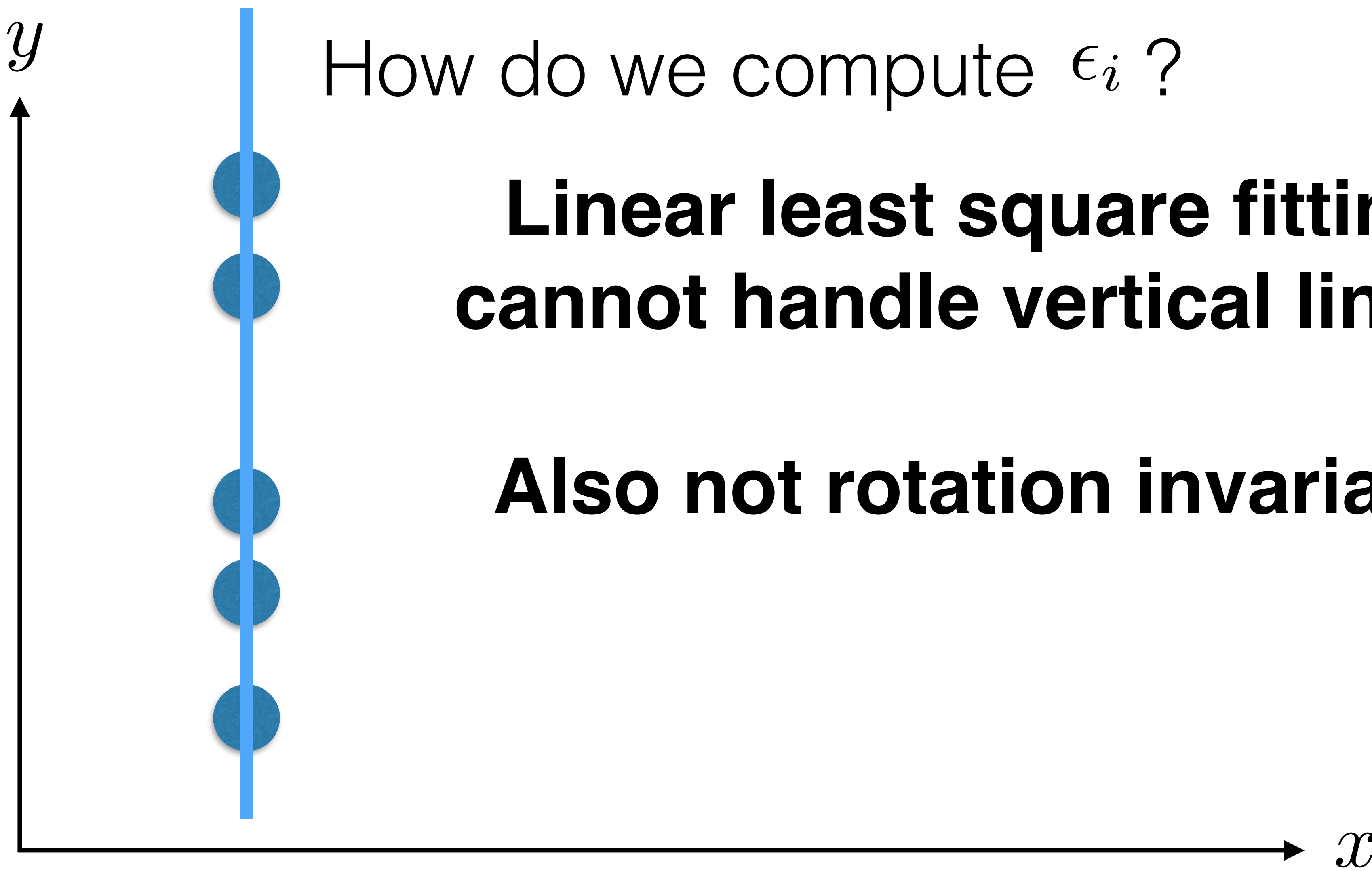
$$-2\mathbf{A}^T\mathbf{y} + 2\mathbf{A}^T\mathbf{A}\mathbf{p} = 0$$

$$\implies \mathbf{A}^T\mathbf{A}\mathbf{p} = \mathbf{A}^T\mathbf{y}$$

$$\implies \mathbf{p} = (\mathbf{A}^T\mathbf{A})^{-1}\mathbf{A}^T\mathbf{y}$$

Same as before. We were performing least square fitting all along.

Least Squares

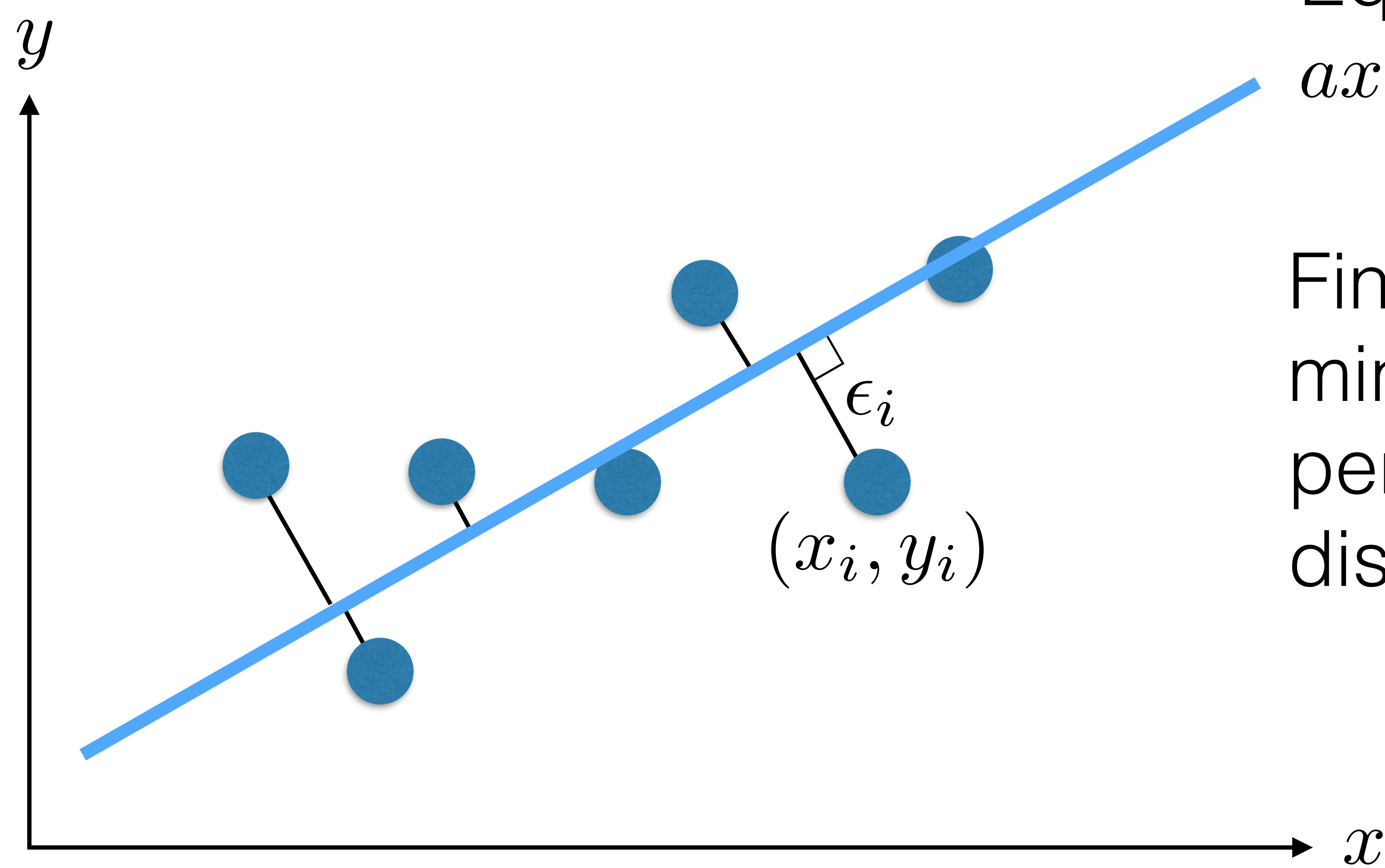


How do we compute ϵ_i ?

Linear least square fitting cannot handle vertical lines!

Also not rotation invariant

Total Least Squares



Eq. of a line:
$ax + by + c = 0$

Find (a, b, c) to minimize the perpendicular distances

(a, b) is the unit normal of the line

Total Least Squares

Minimize $\epsilon = \sum_{i=1}^{n}(ax_i + by_i + c)^2$ s.t. $(a, b)^T(a, b) = 1$

Set $\frac{\partial \epsilon}{\partial c} = 2\sum_{i=1}^{n}(ax_i + by_i + c)$ to 0

$$\sum_{i=1}^{n}(ax_i + by_i + c) = 0$$

$$\implies a\sum_{i=1}^{n} x_i + b\sum_{i=1}^{n} y_i + nc = 0$$

$$\implies c = -\frac{a}{n}\sum_{i=1}^{n} x_i - \frac{b}{n}\sum_{i=1}^{n} y_i = -a\hat{x} - b\hat{y}$$

Total Least Squares

Substituting $c = -\frac{a}{n}\sum_{i=1}^{n} x_i - \frac{b}{n}\sum_{i=1}^{n} y_i$ in ϵ

$$\epsilon = \sum_{i=1}^{n}(ax_i + by_i - a\hat{x} - b\hat{y})^2$$

$$= \sum_{i=1}^{n}(a(x_i - \hat{x}) + b(y_i - \hat{y}))^2$$

$$= \| \begin{bmatrix} x_1 - \hat{x} & y_1 - \hat{y} \\ \vdots & \vdots \\ x_n - \hat{x} & y_n - \hat{y} \end{bmatrix} \begin{bmatrix} a \\ b \end{bmatrix} \|^2 = \|\mathbf{Ap}\|^2$$

Notice not the same
as least squares

Total Least Squares

Solution

$$\mathbf{p}^* = \underset{\mathbf{p}}{\operatorname{argmin}} \|\mathbf{A}\mathbf{p}\|^2 \text{ s.t. } \mathbf{p}^T\mathbf{p} = 1$$

Solution is the eigenvector corresponding to the smallest eigenvalue of $\mathbf{A}^T\mathbf{A}$

Singular Value Decomposition

The Singular Value Decomposition (SVD) of a matrix $\mathbf{A} \in \mathbb{R}^{m \times n}$ is defined as

$$\mathbf{A} = \mathbf{U}\mathbf{D}\mathbf{V}^T$$

such that

- $\mathbf{U}$ is a $m \times n$ matrix with orthogonal columns
- $\mathbf{D}$ is a $n \times n$ diagonal matrix. Its entries are called singular values
- $\mathbf{V}$ is an $n \times n$ orthogonal matrix

Singular Value Decomposition

The Singular Value Decomposition (SVD) of a matrix $\mathbf{A} \in \mathbb{R}^{m \times n}$ is defined as

$$\mathbf{A} = \mathbf{U}\mathbf{D}\mathbf{V}^T$$

such that

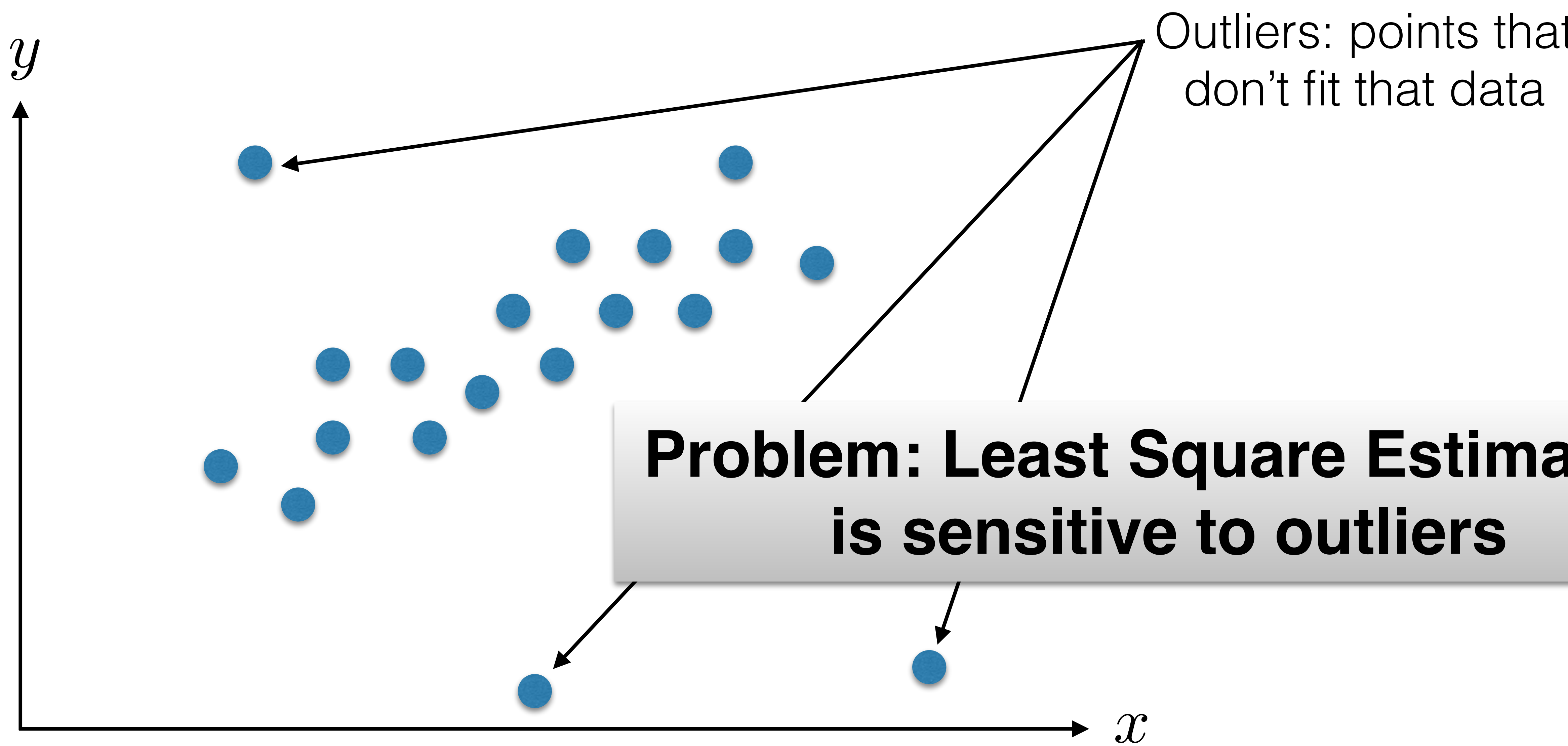
lled

If diagonal entries of $\mathbf{D}$ are sorted as

$$\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_n \geq 0$$

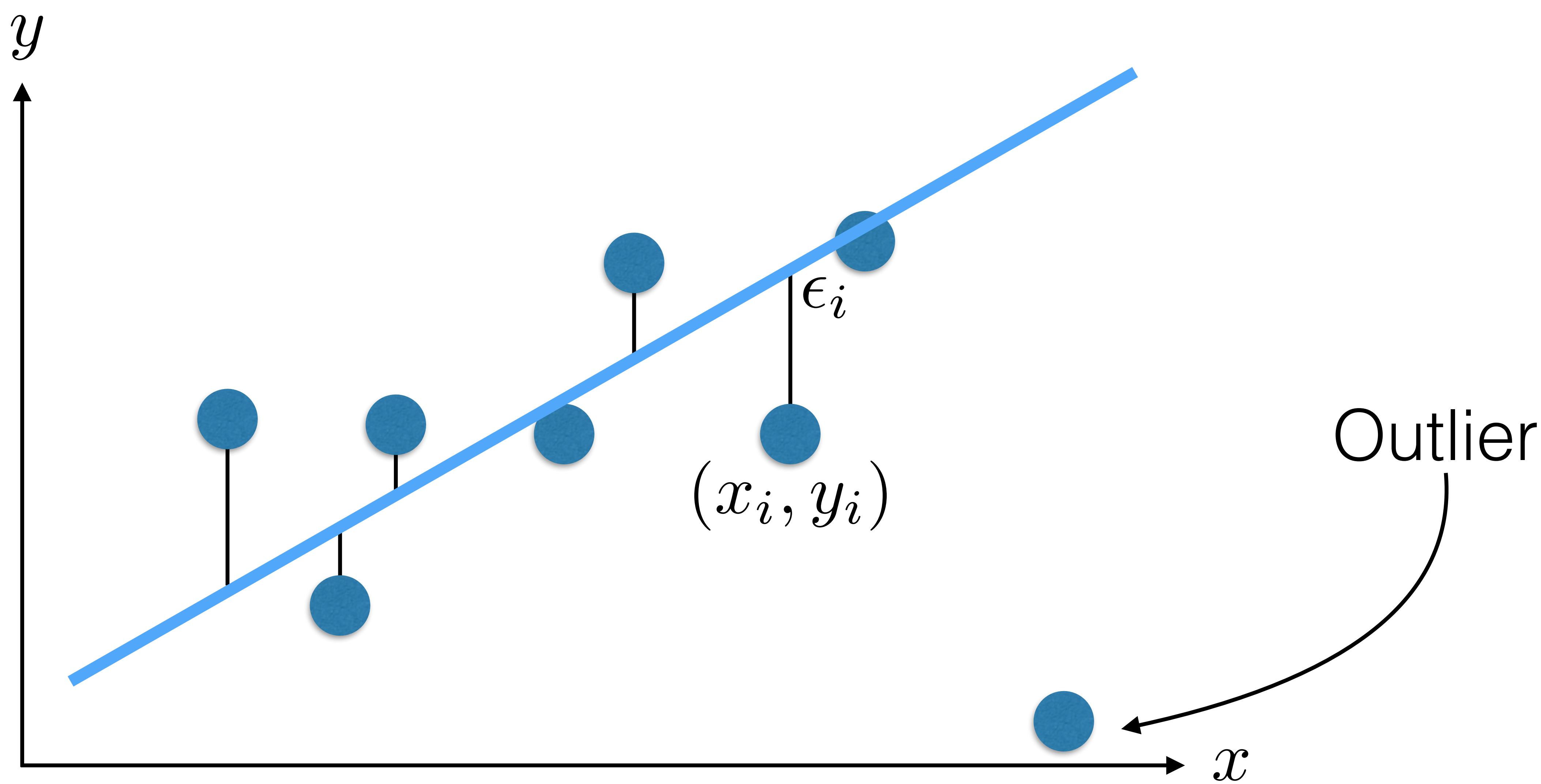
Last column of $\mathbf{V}$ is the smallest eigenvector of $\mathbf{A}^T\mathbf{A}$

Robust Estimation



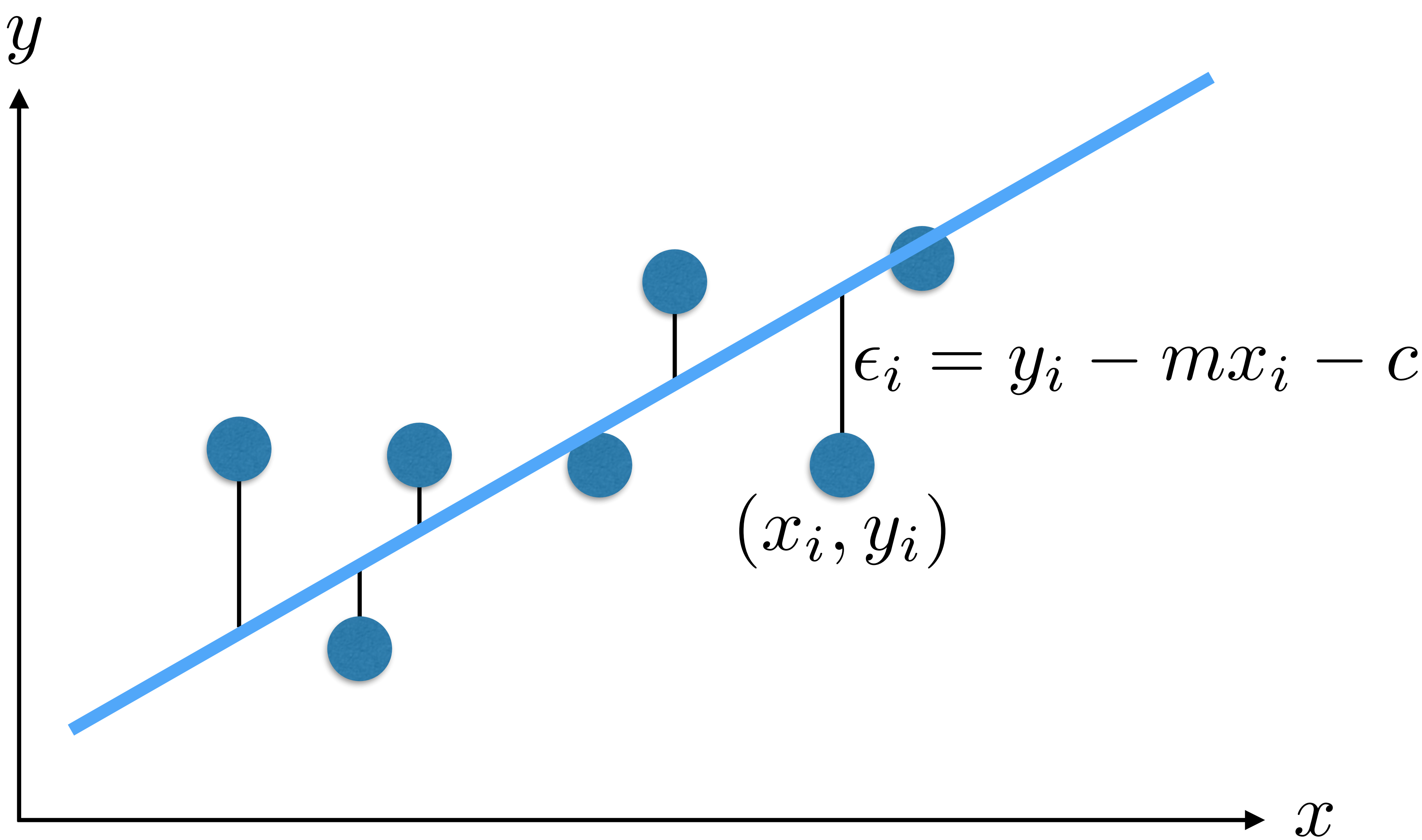
Problem: Least Square Estimation is sensitive to outliers

Least Squares



Solution: Use robust least square fitting, RANSAC

Robust Least Squares



Minimize

$$\sum_{i=1}^{n} \rho(\epsilon_i, \sigma)$$

where

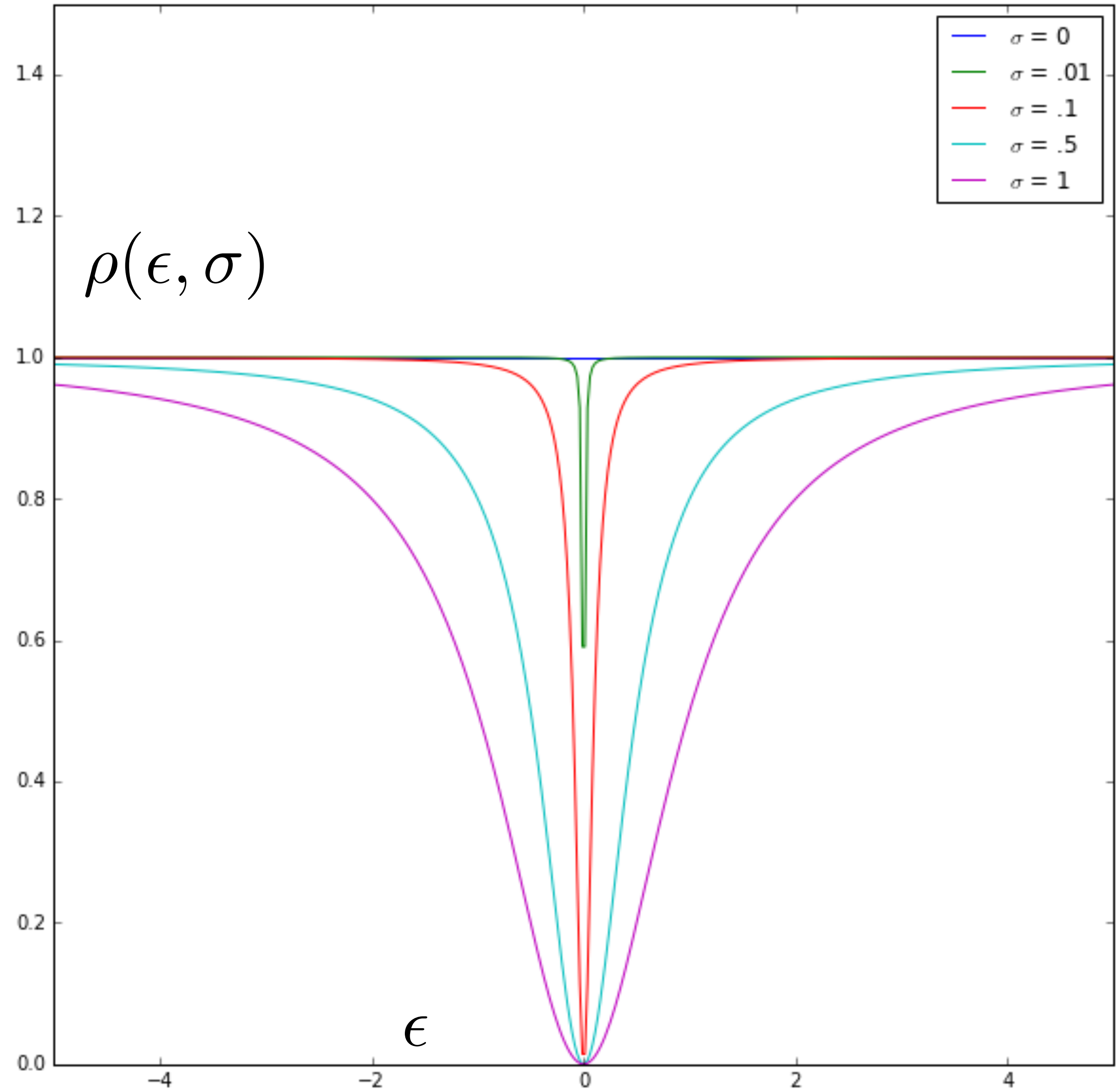
$$\rho(\epsilon, \sigma) = \frac{\epsilon^2}{\epsilon^2 + \sigma^2}$$

- Favors a configuration with small residuals
- Constant penalty for large residuals

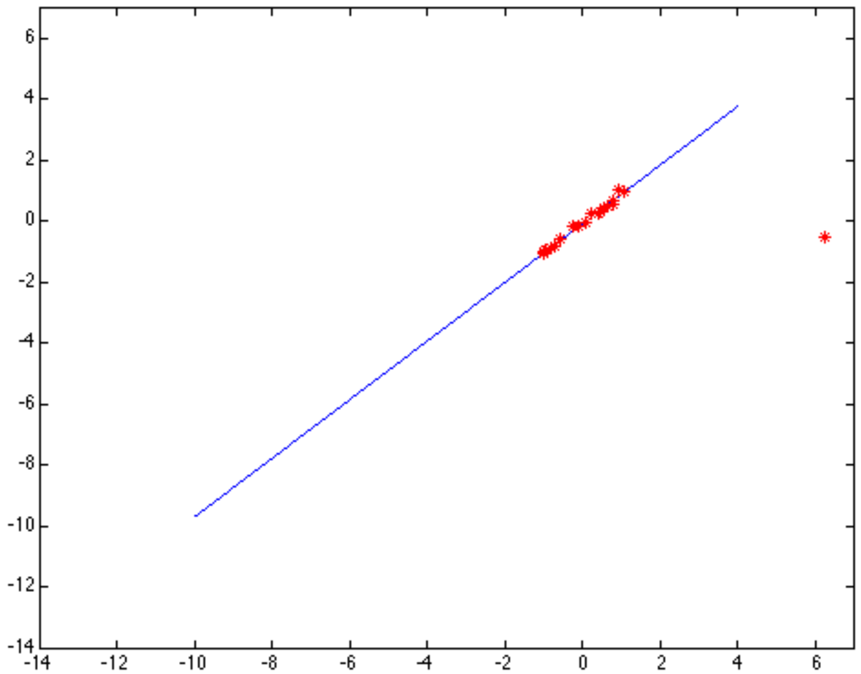
Robust Least Squares

$$\rho(\epsilon, \sigma) = \frac{\epsilon^2}{\epsilon^2 + \sigma^2}$$

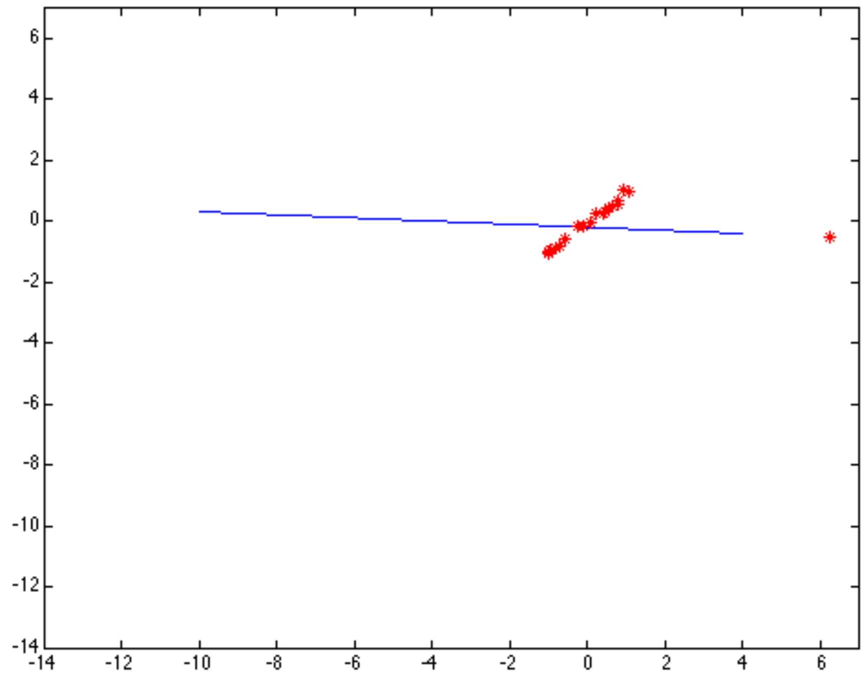
Scale parameter



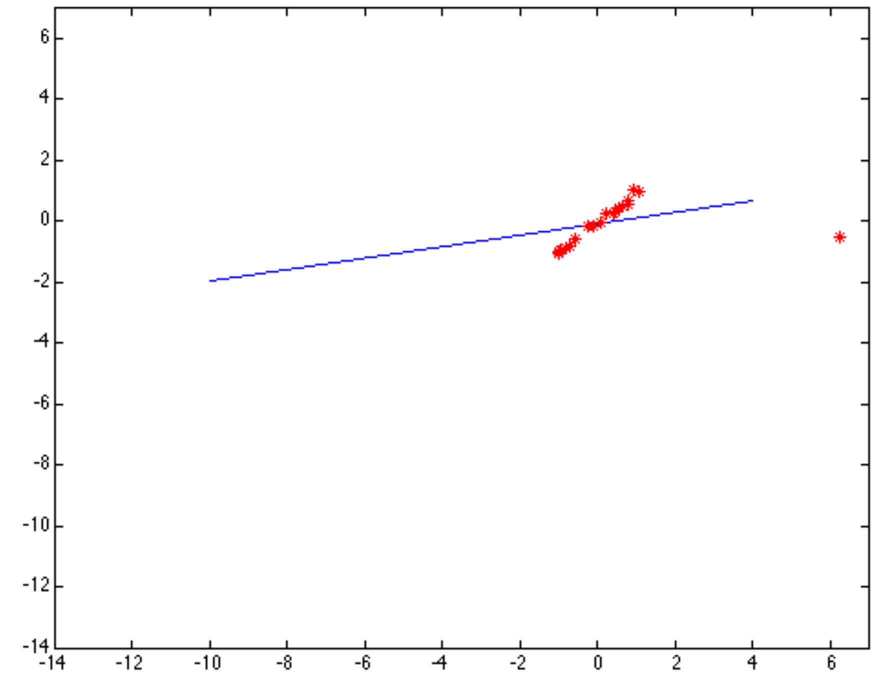
Robust Least Squares



When the scale is selected just right, the effects of outliers are eliminated.



Scale is too small, so the error value is almost the same for each point. This results in a poor fit.

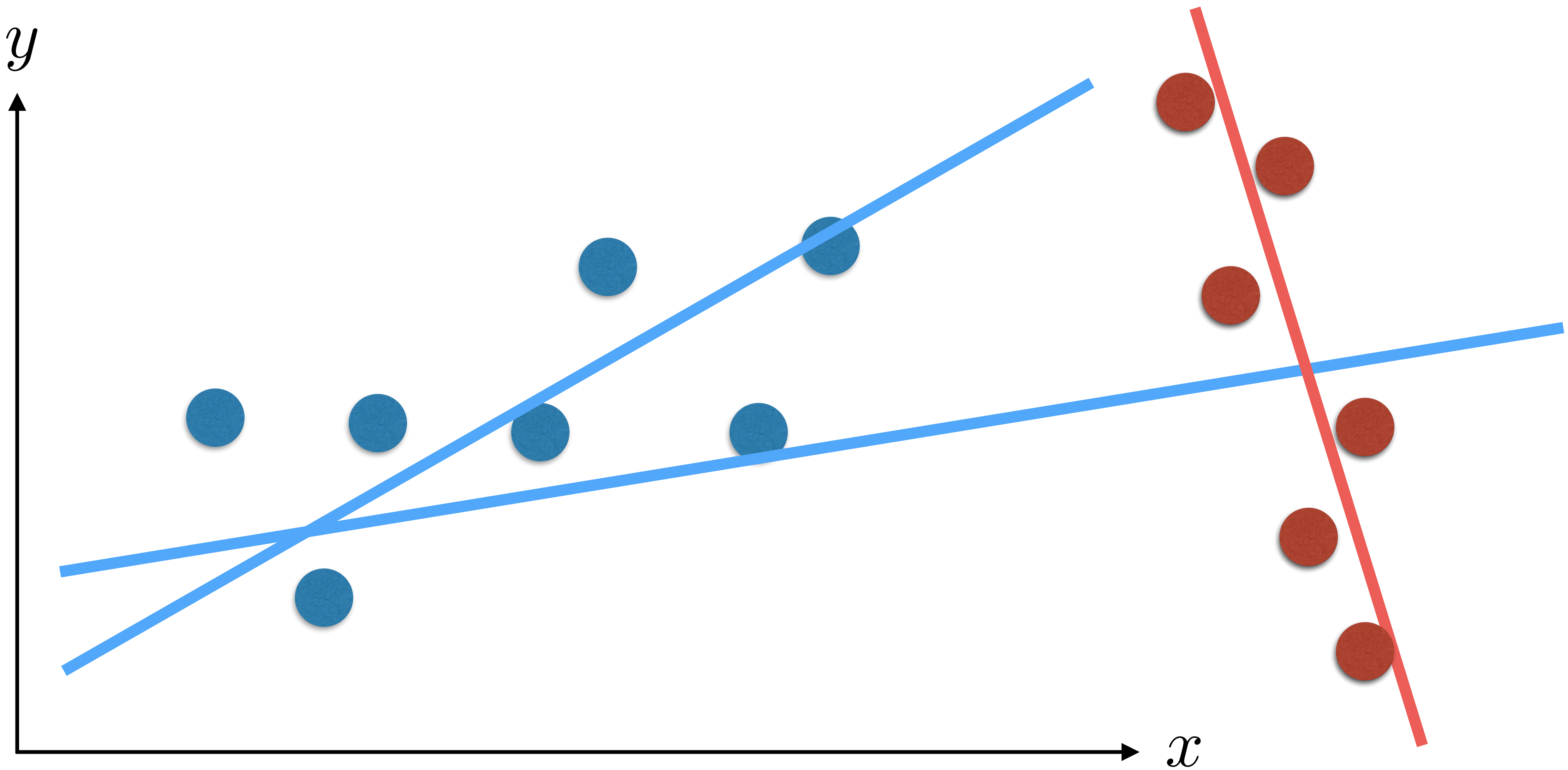


When the scale is too large, the system behaves like a least squares. It is sensitive to outliers

Robust Least Squares

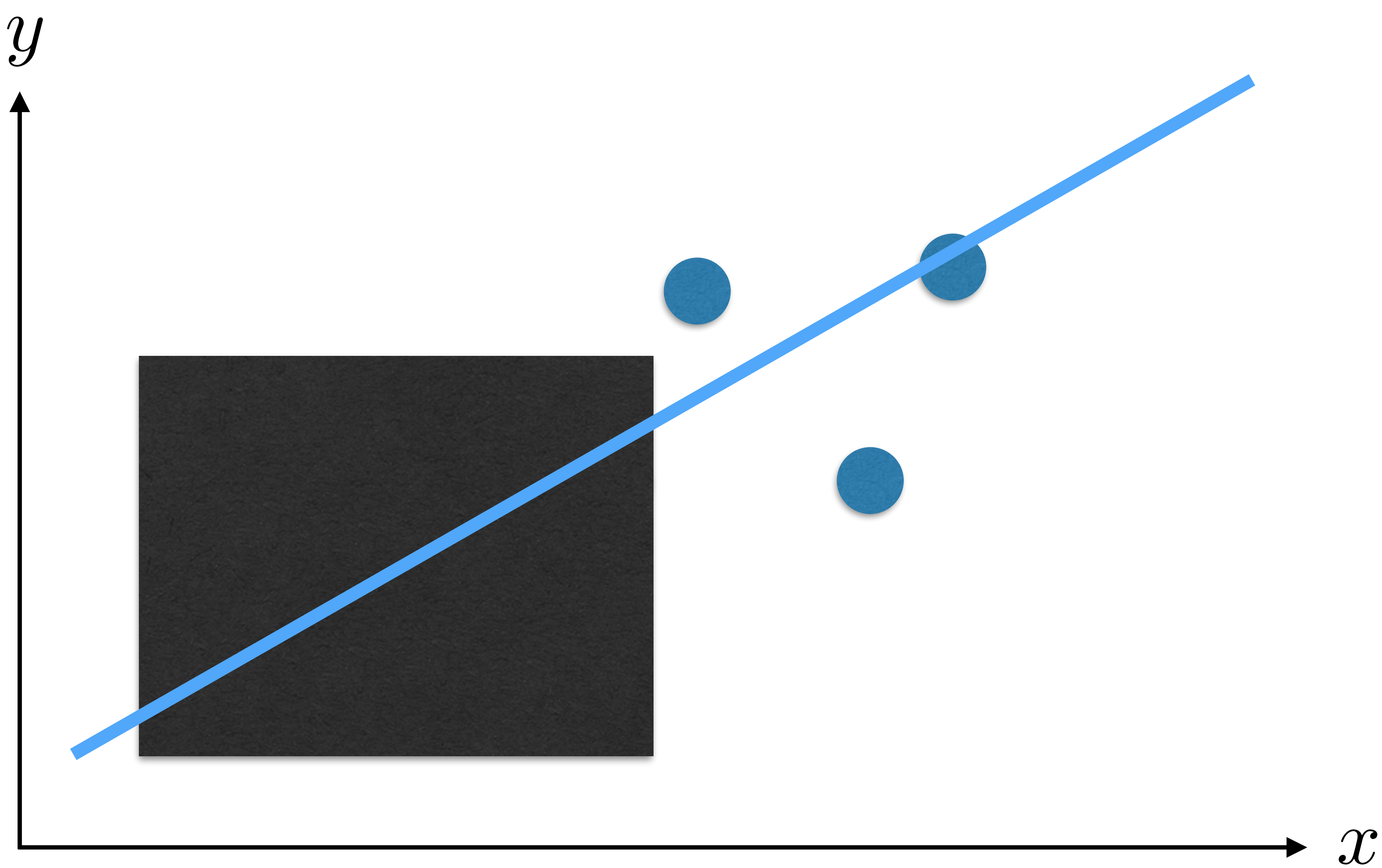
- Robust least square fitting is a non-linear optimization problem, which must be solved iterative.
  - The iterative problem can be initialized using least square fitting.
- Scale parameters can be chosen adaptively as well
  - Scale parameter is typically selected to 1.5 times the median residual

Robust Estimation



Which data points are responsible for which model?

Model Fitting



How can we be sure that all relevant data points are available?

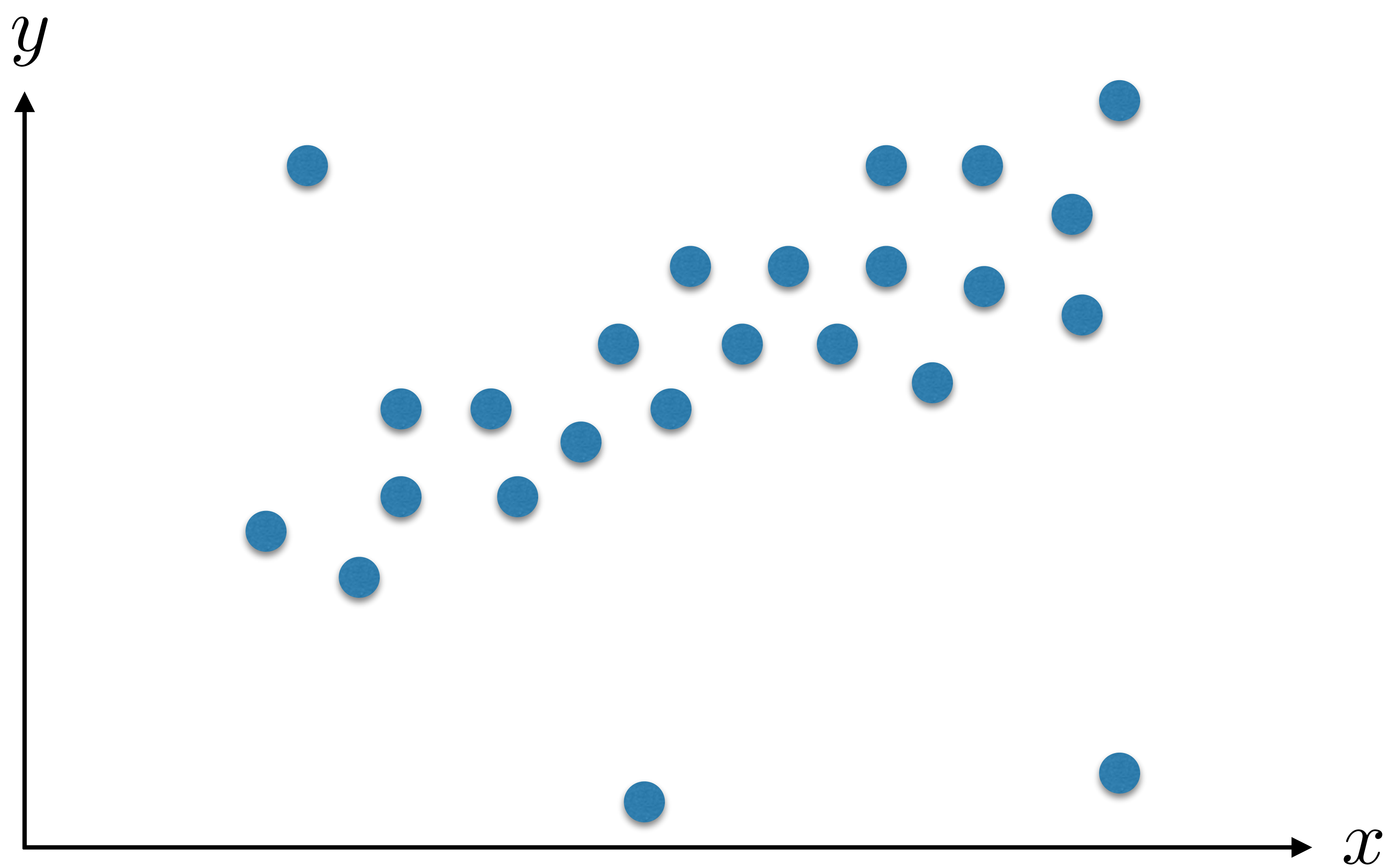
RANSAC

RANSAC

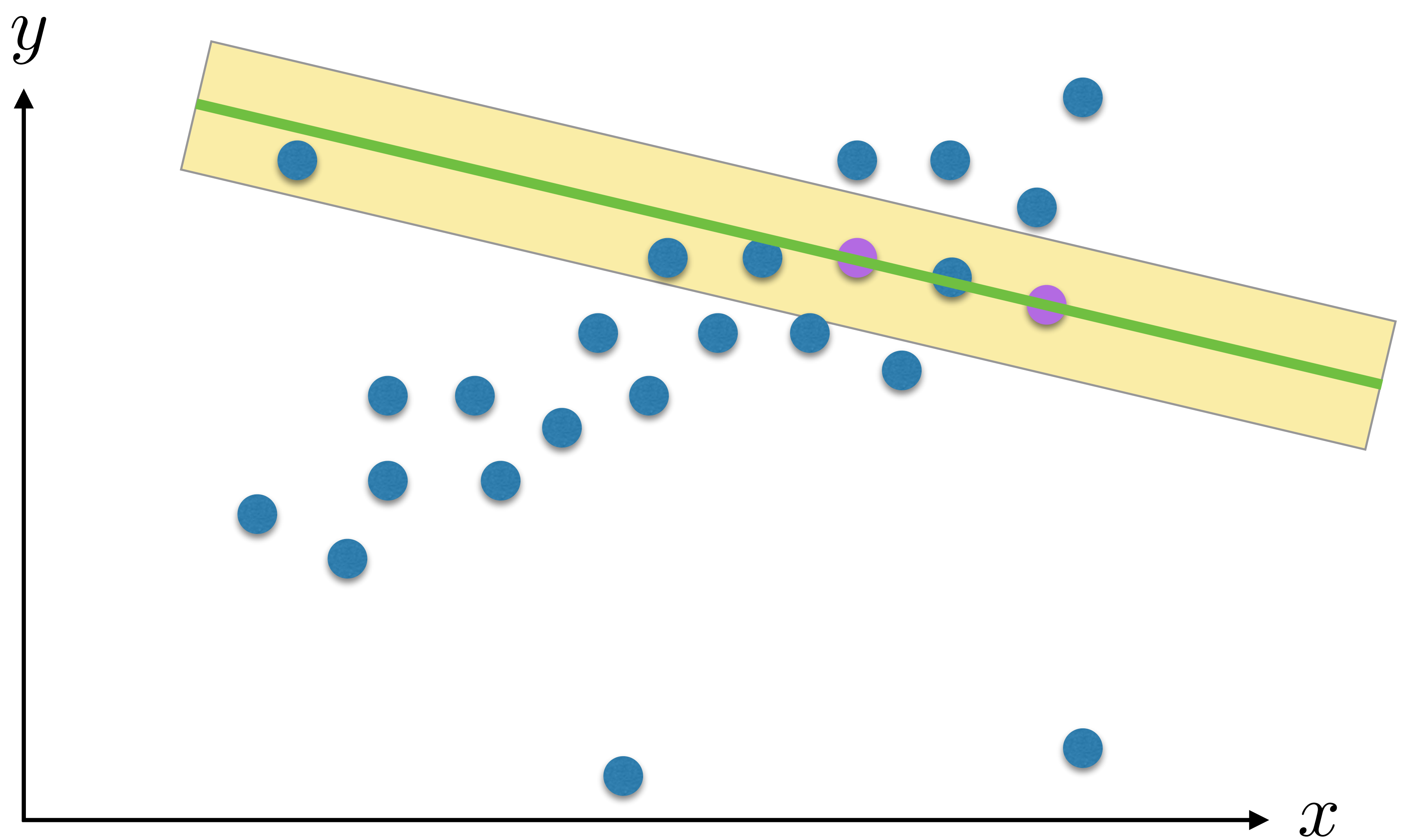
- General approach
  - Classify points as inliers and outlines
  - Estimated model parameters using inliers, ignoring outliers
  - Repeat if necessary

M. A. Fischler and R. C. Bolles (June 1981). "Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography". *Comm. of the ACM* 24: 381--395.

RANSAC

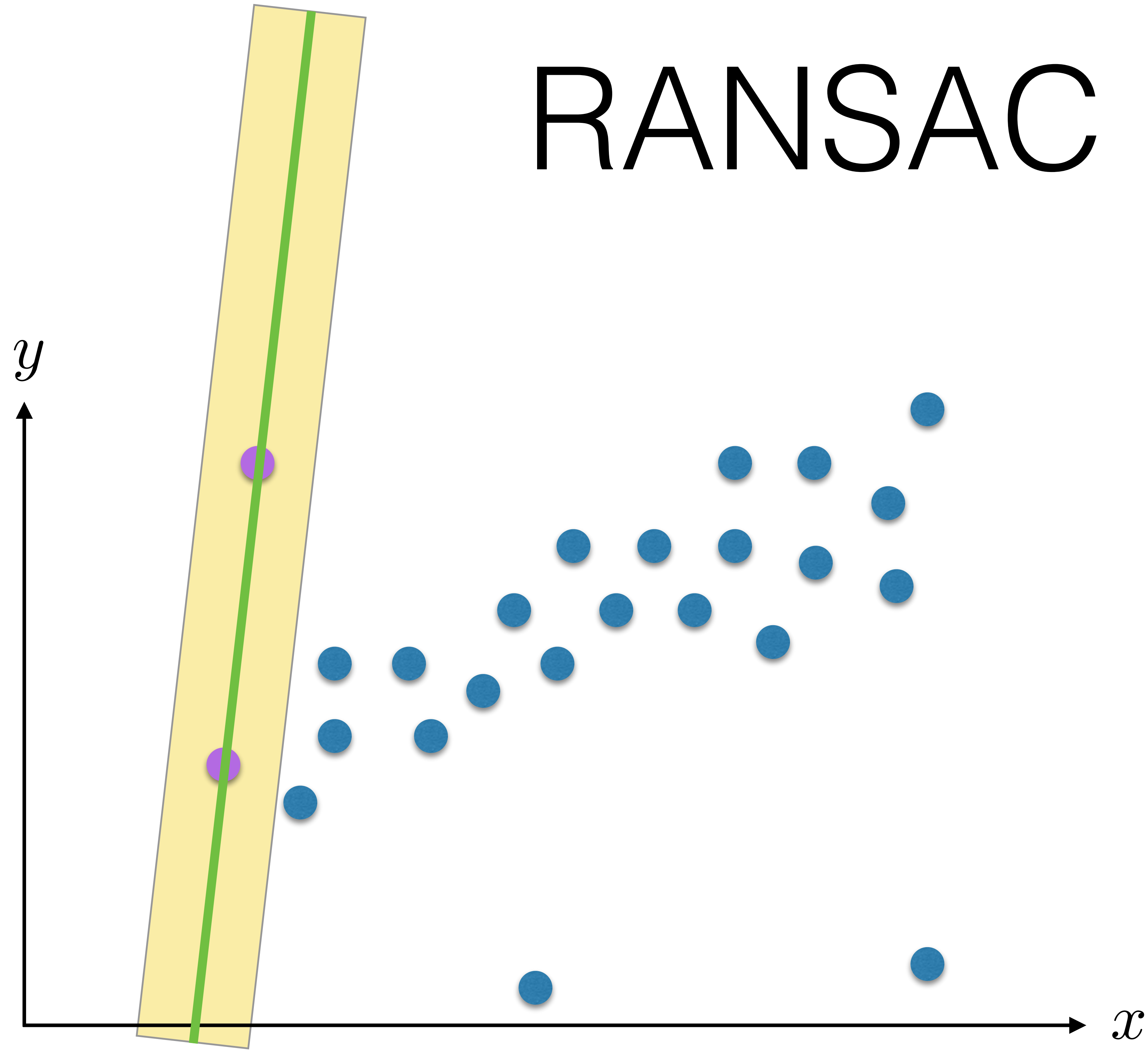


RANSAC



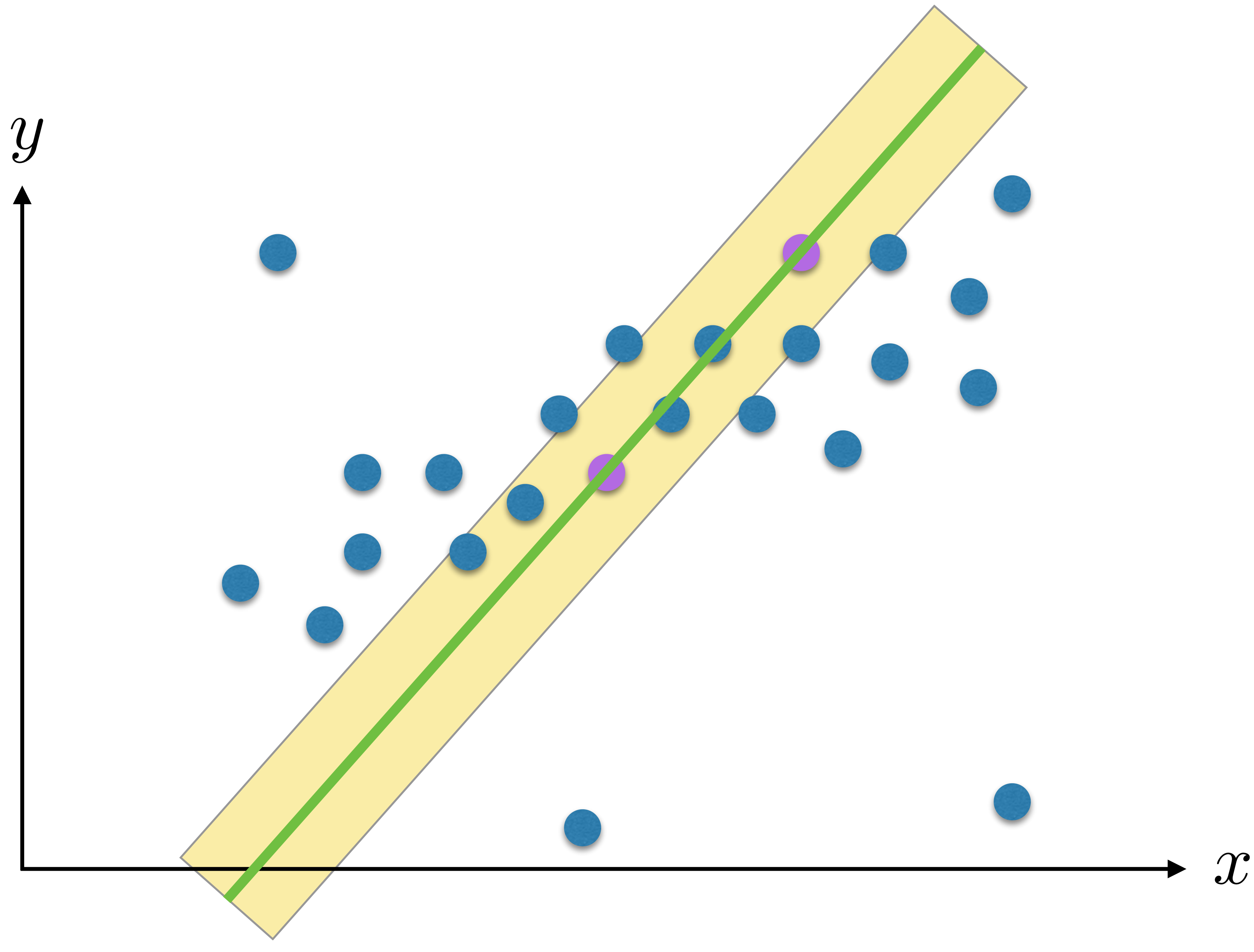
Count = 6

RANSAC



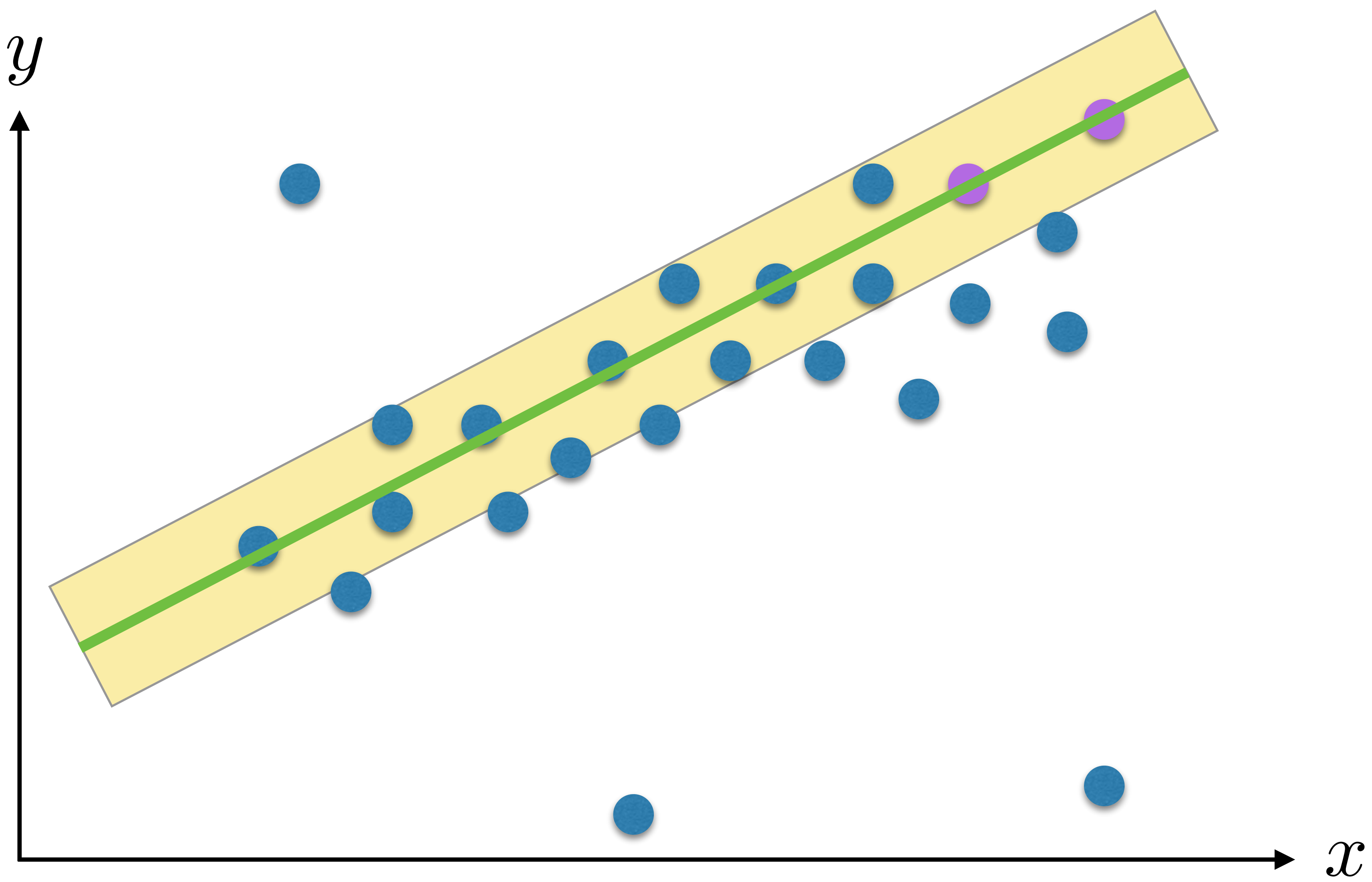
Count = 2

RANSAC



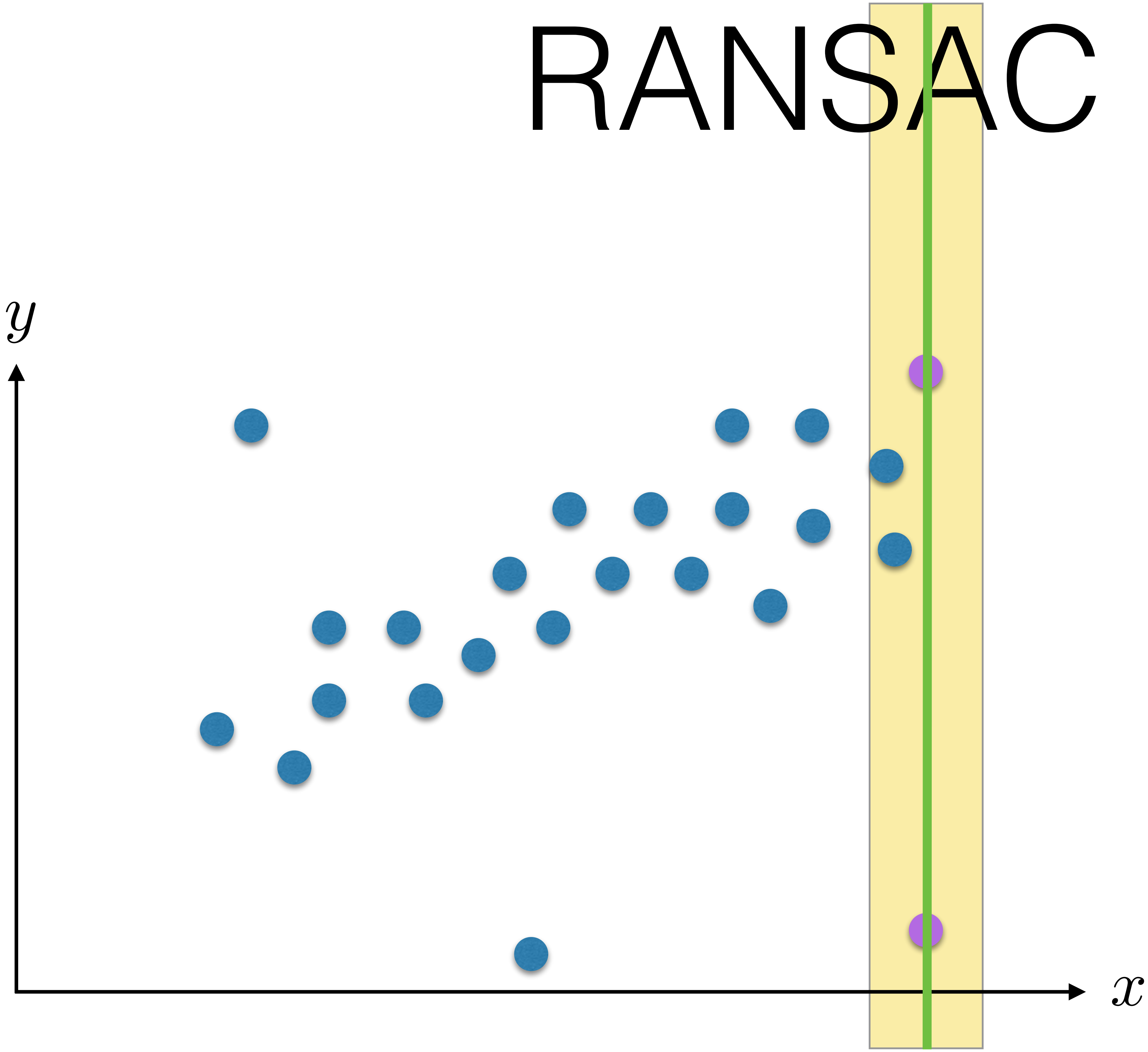
Count = 8

RANSAC



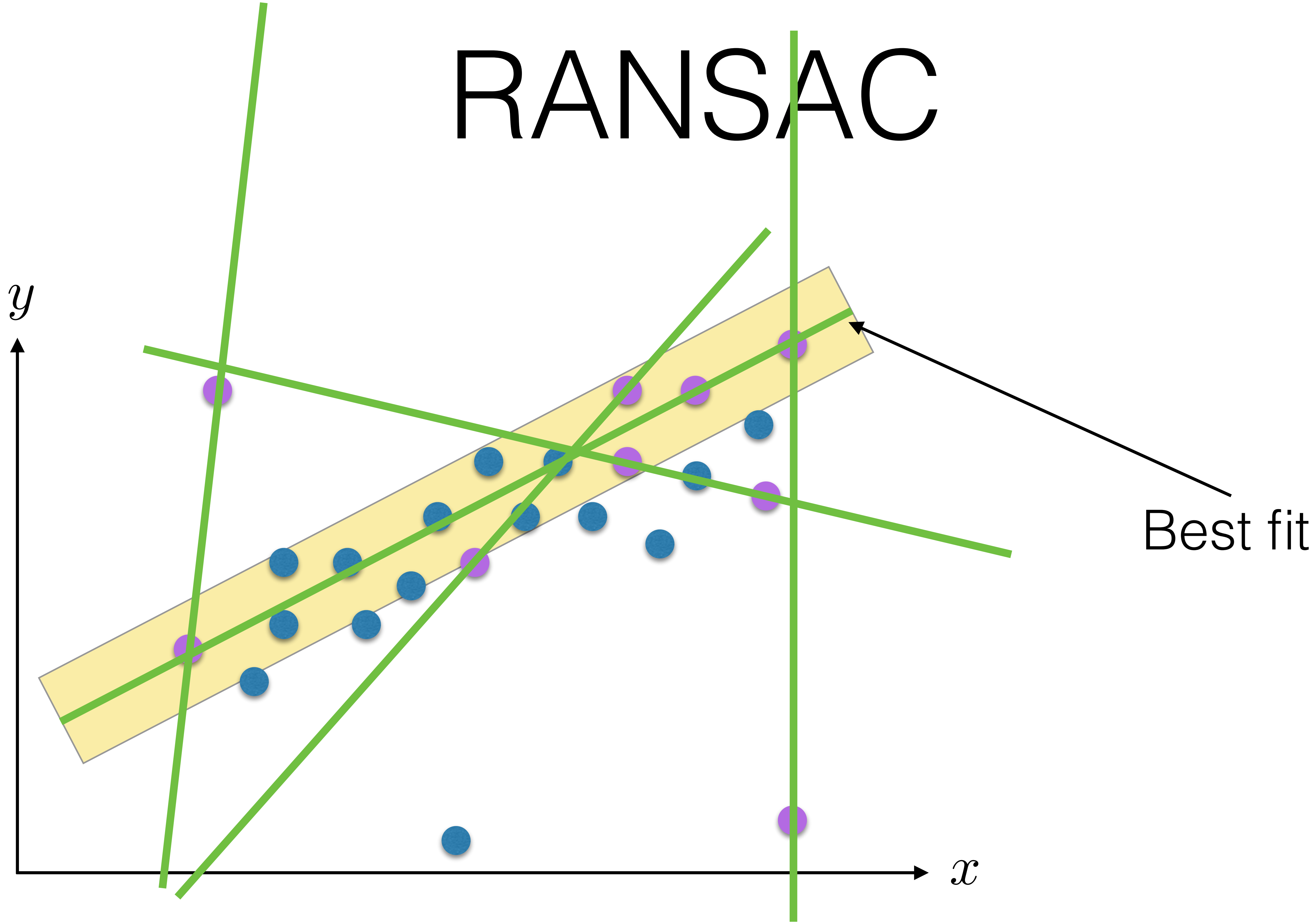
Count = 16

RANSAC



Count = 4

RANSAC



RANSAC

Algorithm 15.4: RANSAC: fitting lines using random sample consensus

Determine:
    s — the smallest number of points required
    N — the number of iterations required
    d — the threshold used to identify a point that fits well
    T — the number of nearby points required
        to assert a model fits well
Until **N** iterations have occurred
    Draw a sample of **s** points from the data
        uniformly and at random
    Fit to that set of **s** points
    For each data point outside the sample
        Test the distance from the point to the line
            against **d** if the distance from the point to the line
            is less than **d** the point is close
    end
    If there are **T** or more points close to the line
        then there is a good fit. Refit the line using all
        these points.
end
Use the best fit from this collection, using the
    fitting error as a criterion

Forsyth and Ponce

RANSAC

- How many iterations N are needed to success probability equal to p?

Probability of sampling an outlier: e

Samples required for each model fitting: s

RANSAC

- Prob. of selecting an inlier: $1 - e$
- Prob. of selecting s inliers in a row: $(1 - e)^s$
- Prob. of getting a bad sample: $1 - (1 - e)^s$
  At least one of those s selections was an outlier
- Prob. of getting N bad samples in a row: $(1 - (1 - e)^s)^N$
- Prob. of getting at least one good sample after N tries: $1 - (1 - (1 - e)^s)^N$

RANSAC

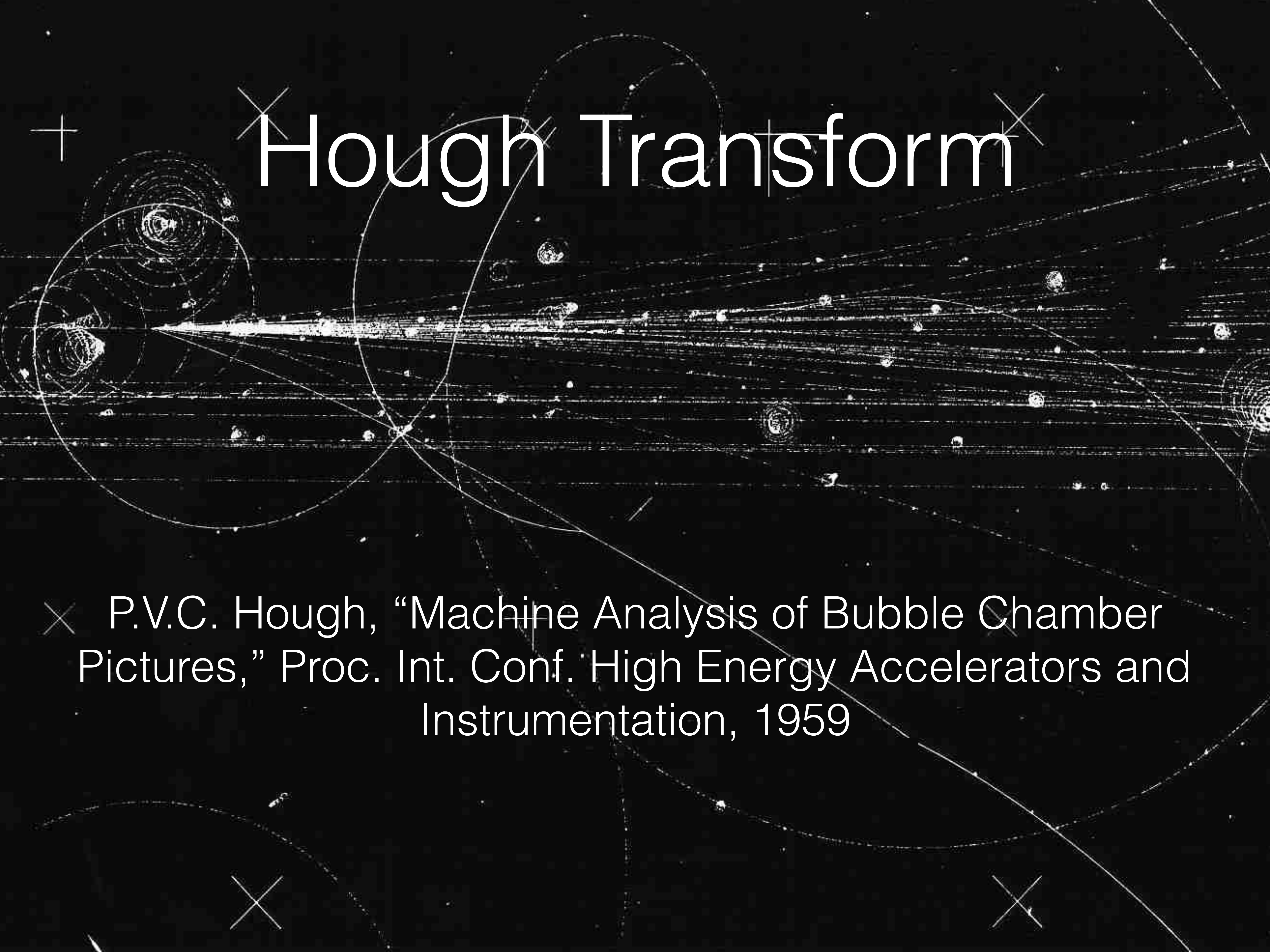
So

$$
\begin{aligned}
& p = 1 - \left(1 - (1 - e)^s\right)^N \\
\implies\ & \left(1 - (1 - e)^s\right)^N = 1 - p \\
\implies\ & \log\left(1 - (1 - e)^s\right)^N = \log(1 - p) \\
\implies\ & N \log\left(1 - (1 - e)^s\right) = \log(1 - p) \\
\implies\ & N = \frac{\log(1 - p)}{\log\left(1 - (1 - e)^s\right)}
\end{aligned}
$$

RANSAC

- Pros
  - Robust to outliers
  - Applicable to large number of parameters (than hough transform approaches)
  - Parameters are easier to choose than hough transform
- Cons
  - Not good for getting multiple fits
    - Computational times grows quickly with fraction of outliers and number of parameters that need to be estimated
- Uses
  - Estimating homography (image stitching)
  - Estimating fundamental matrix (relating two views)

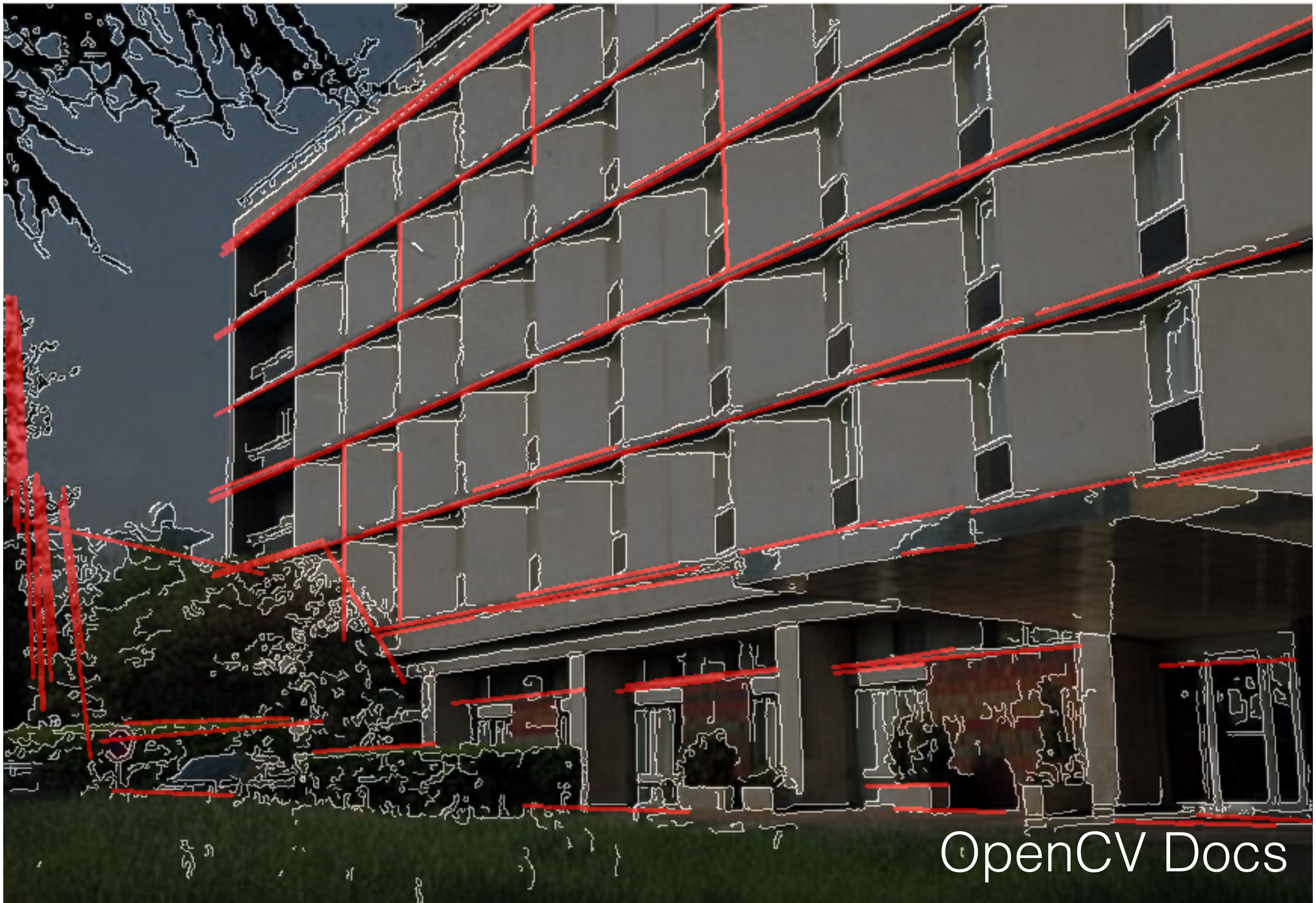
Hough Transform



Hough Transform

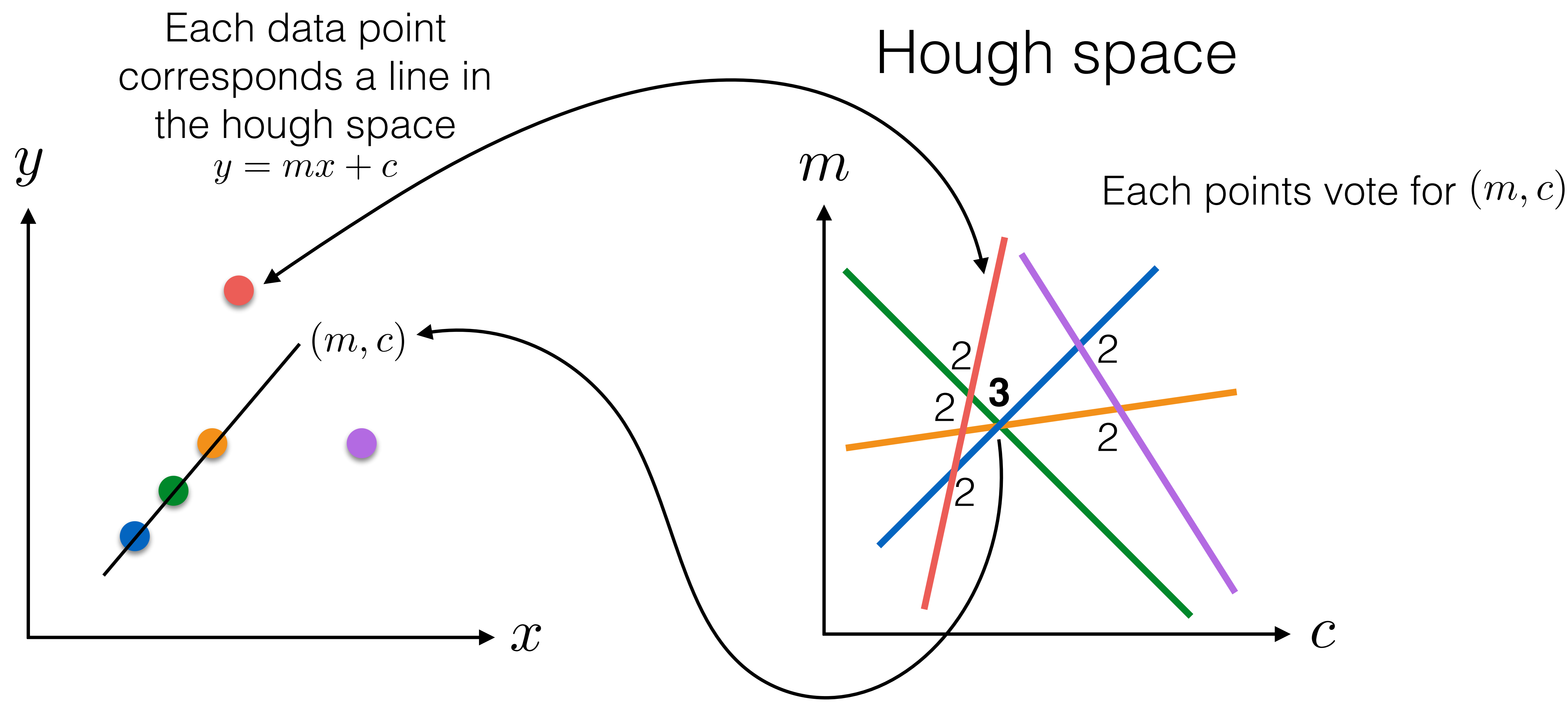
P.V.C. Hough, “Machine Analysis of Bubble Chamber Pictures,” Proc. Int. Conf. High Energy Accelerators and Instrumentation, 1959

Hough Transform

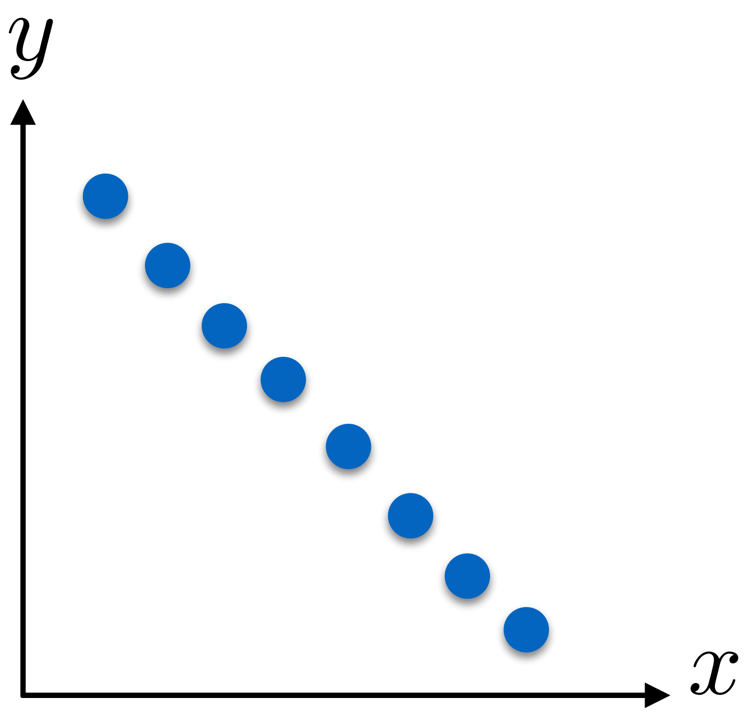


Hough Transform

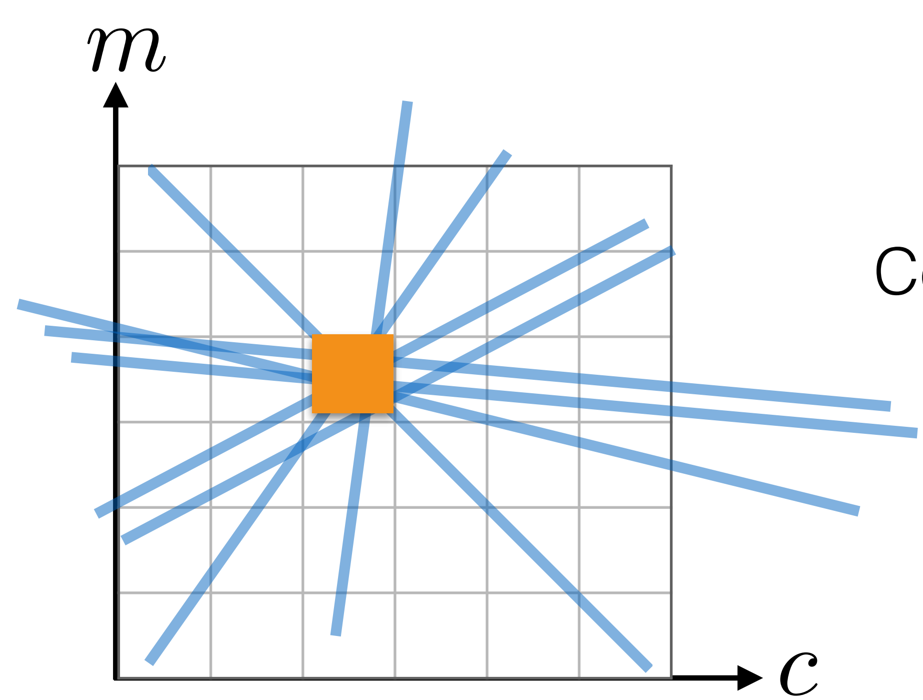
- Given a set of points, find the curve or line that best explains the points



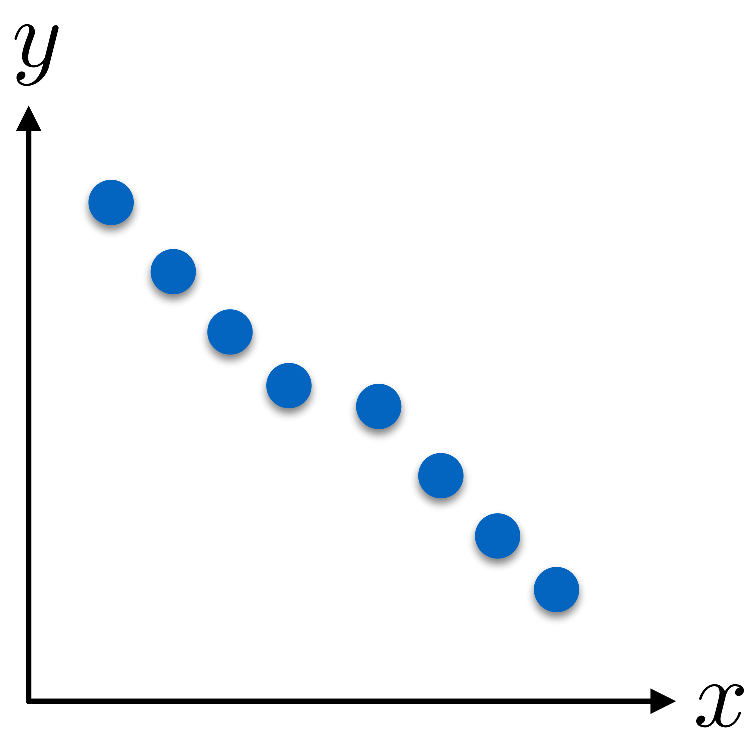
Hough Transform

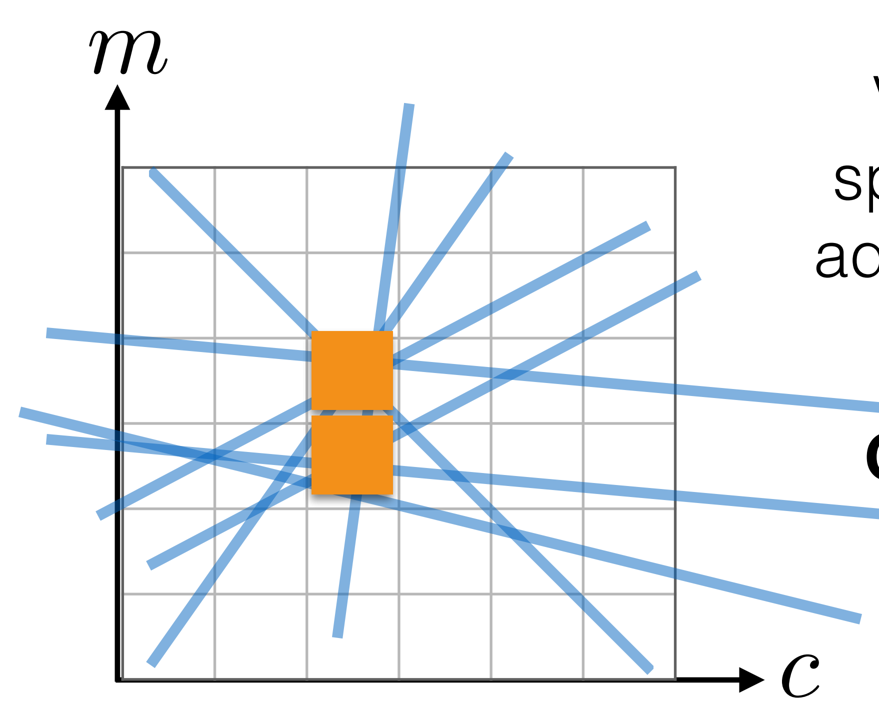


Hough space



Collecting votes

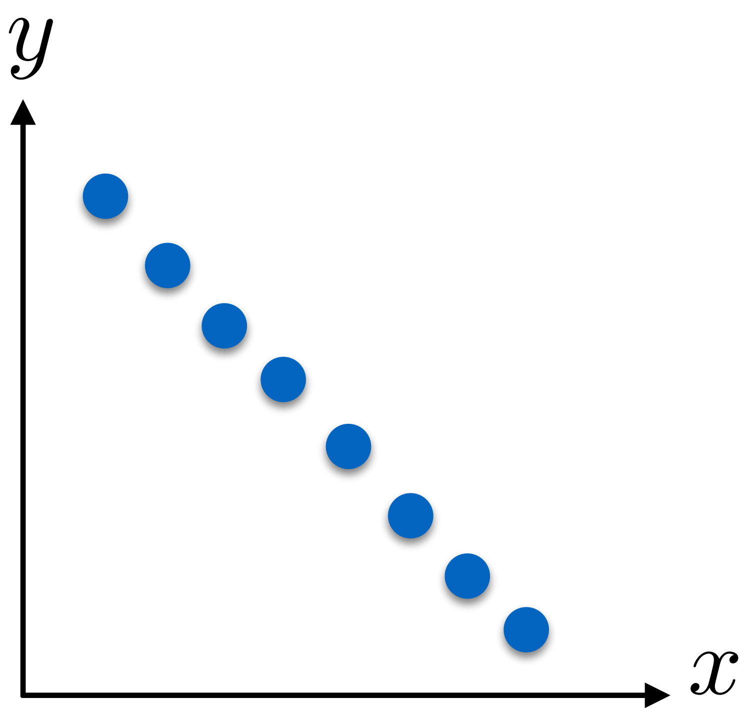


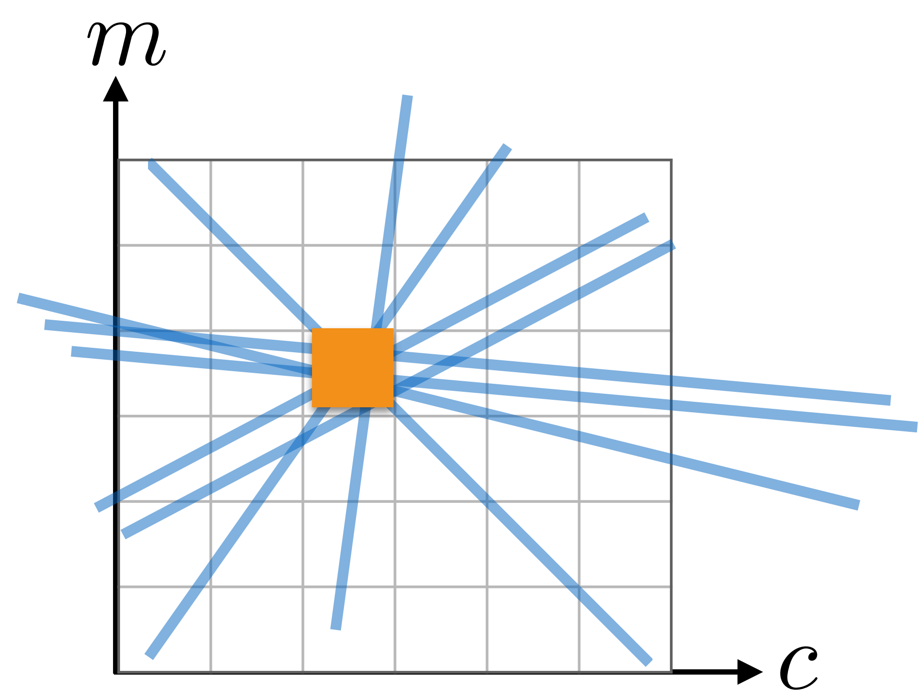


Votes are spread over adjacent grid values

Grid size?

Hough Transform





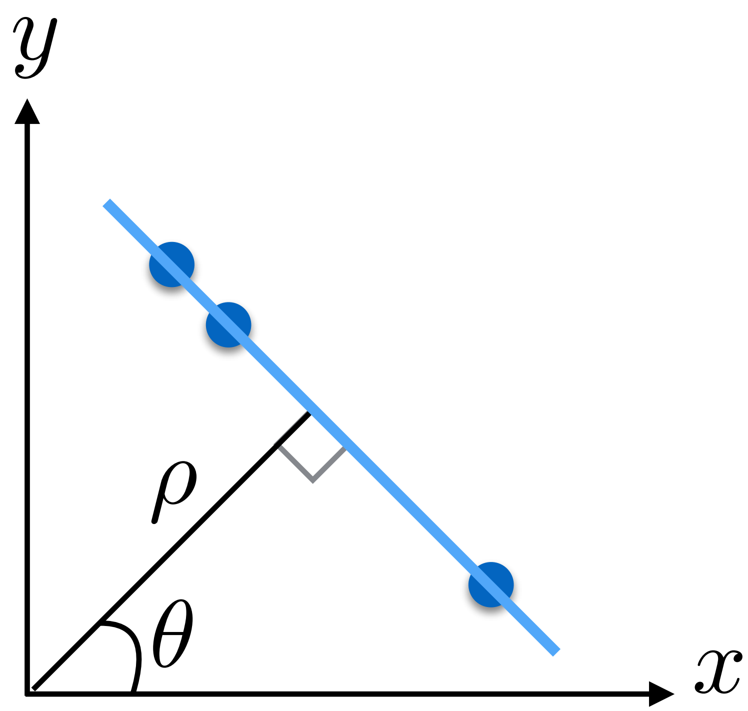
Problem

Parameter space (m, c) is unbounded

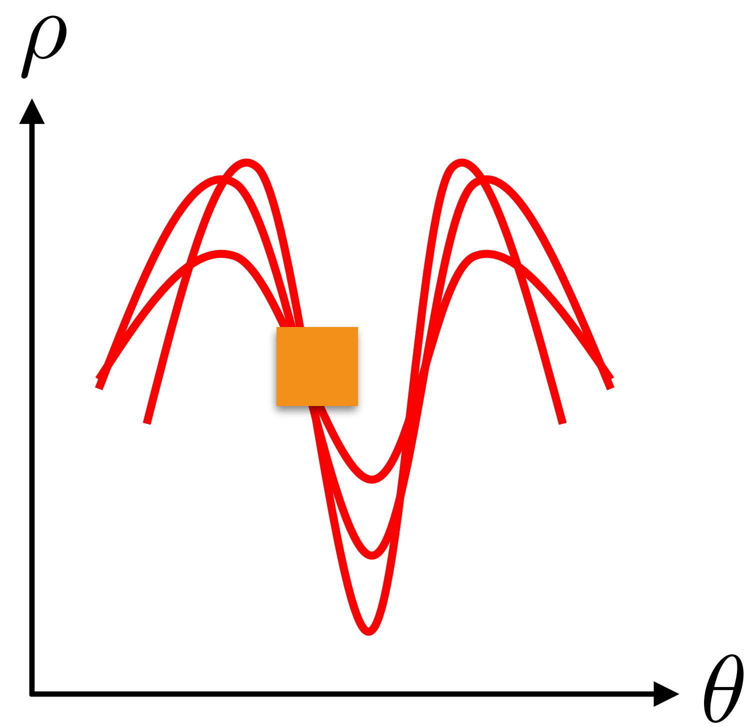
Solution

Use polar representation for parameter space

Polar Representation

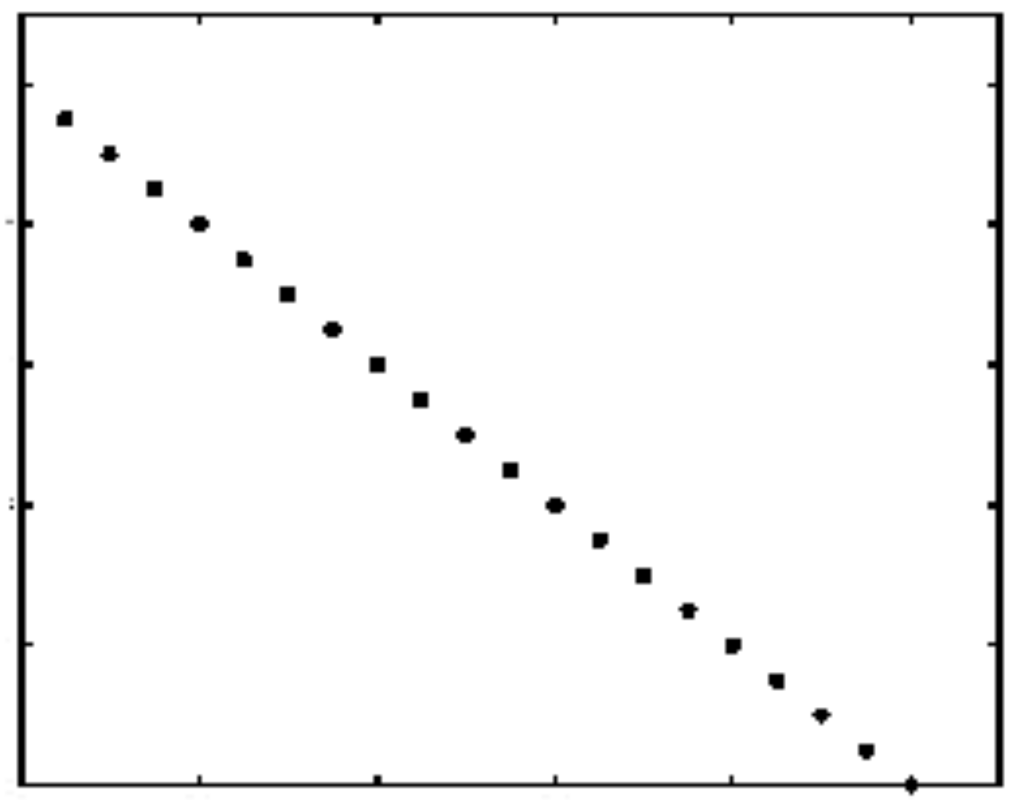


Hough space

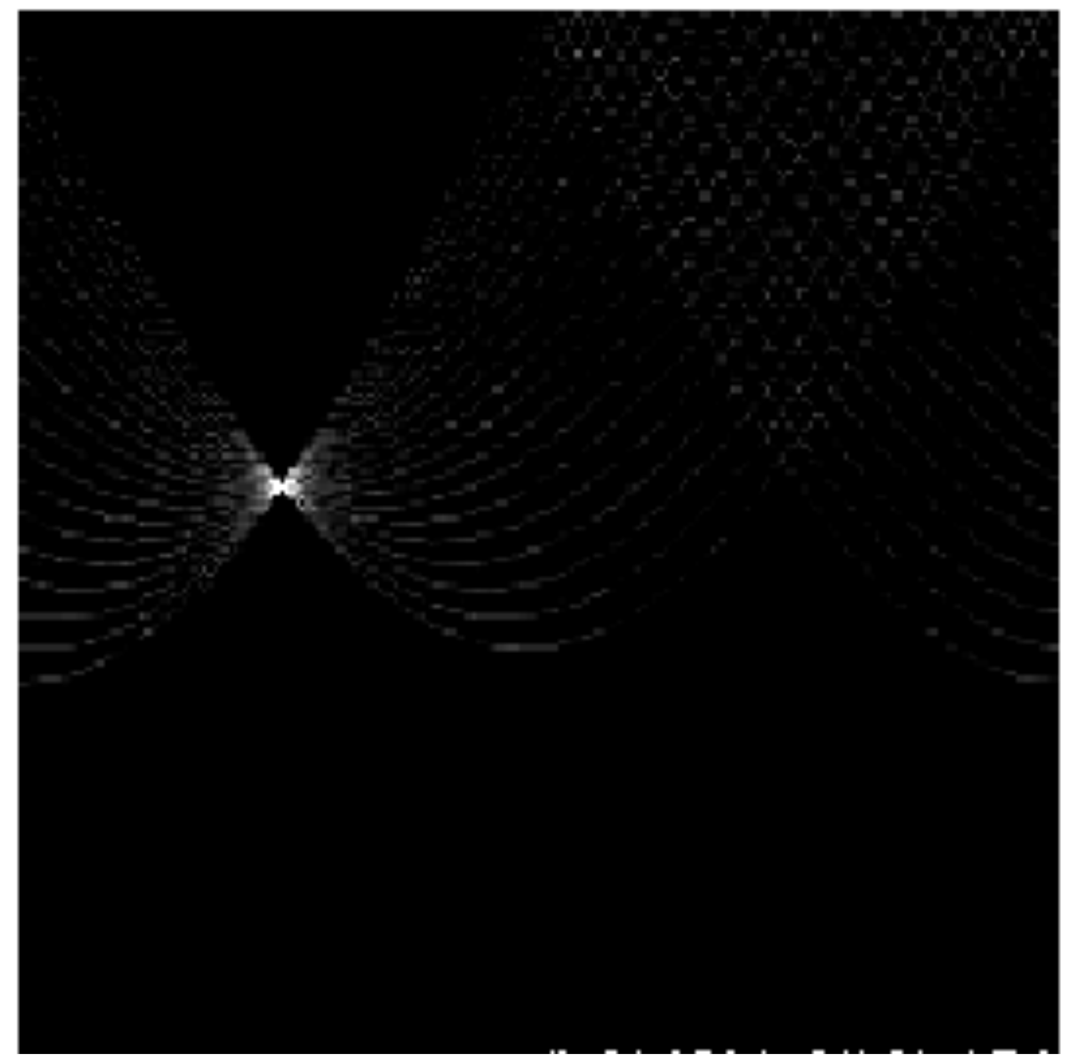


$$x \cos \theta + y \sin \theta = \rho$$

Hough Transform



features

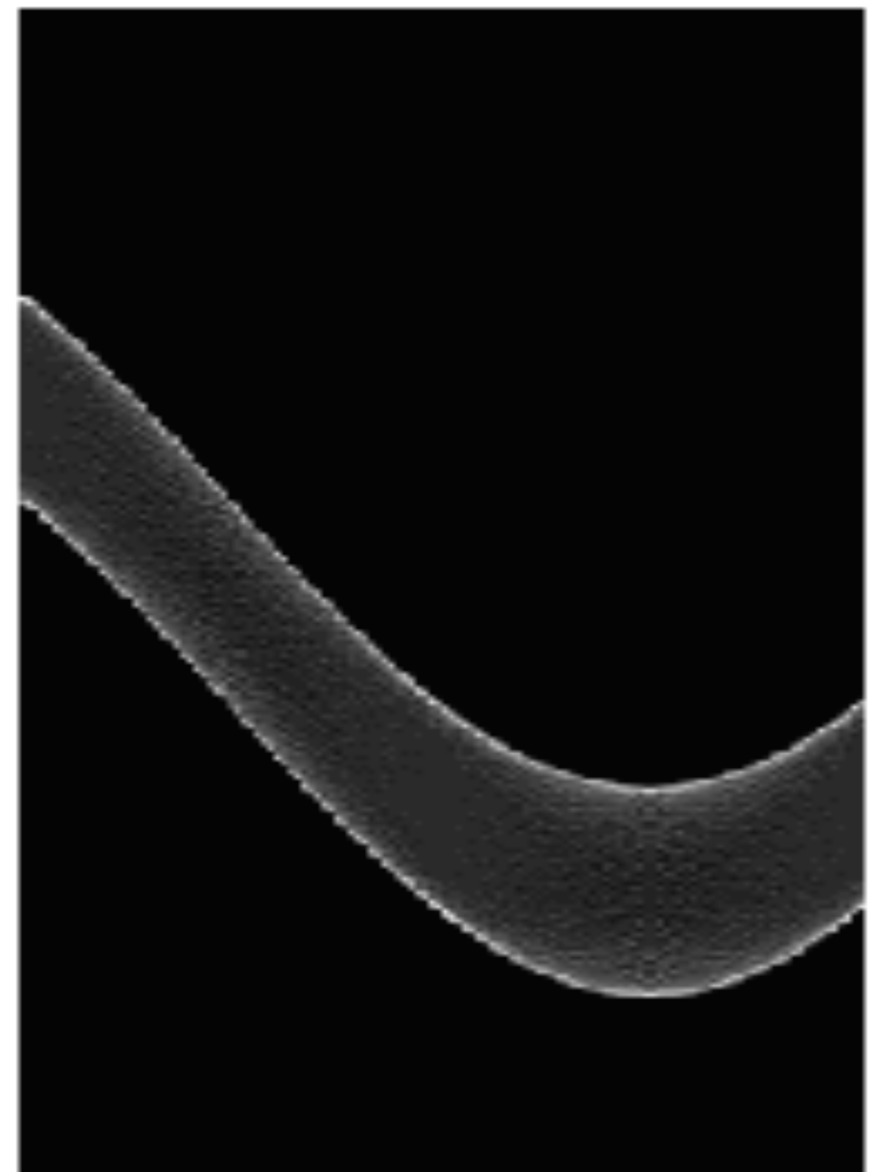


votes (polar representation)

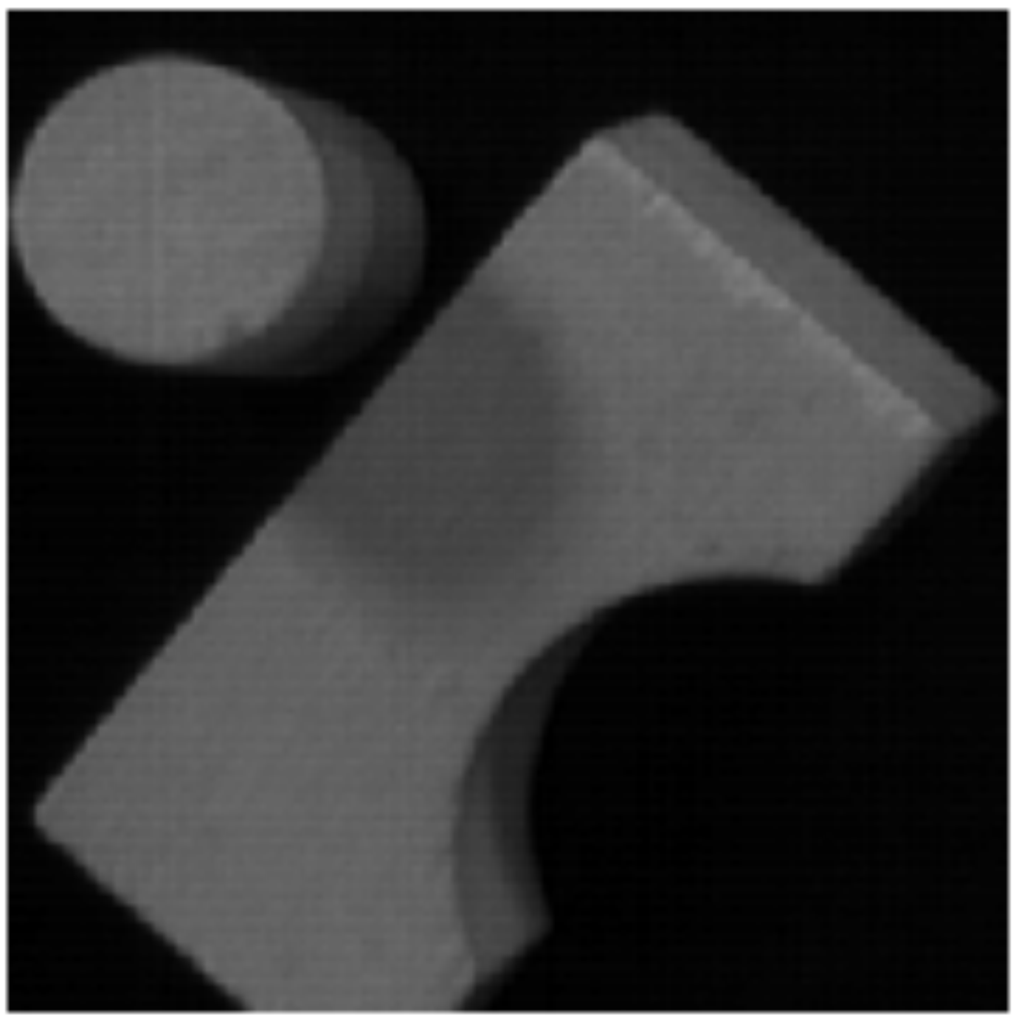
Hough Transform

Squares

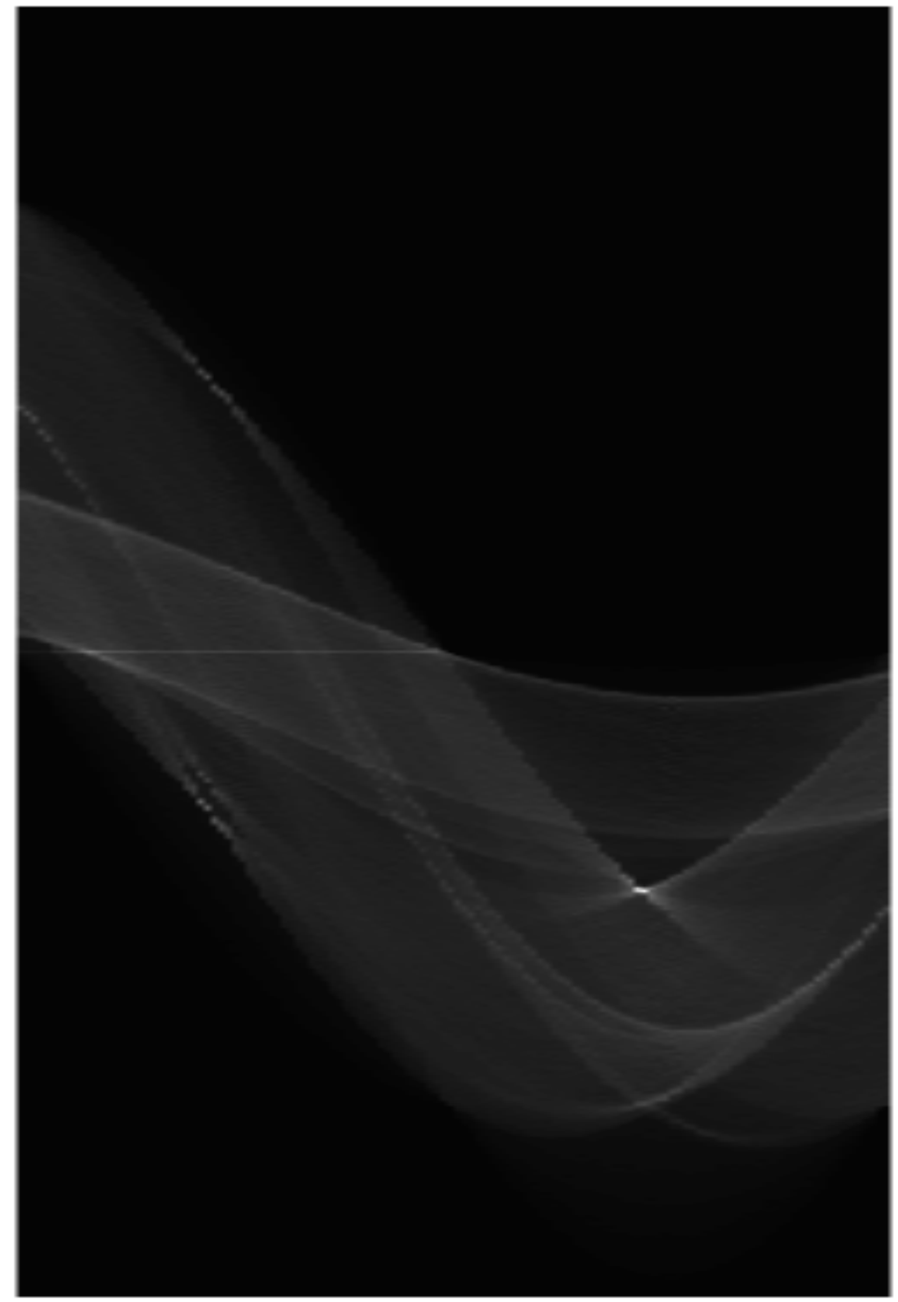
Circle



Hough Transform



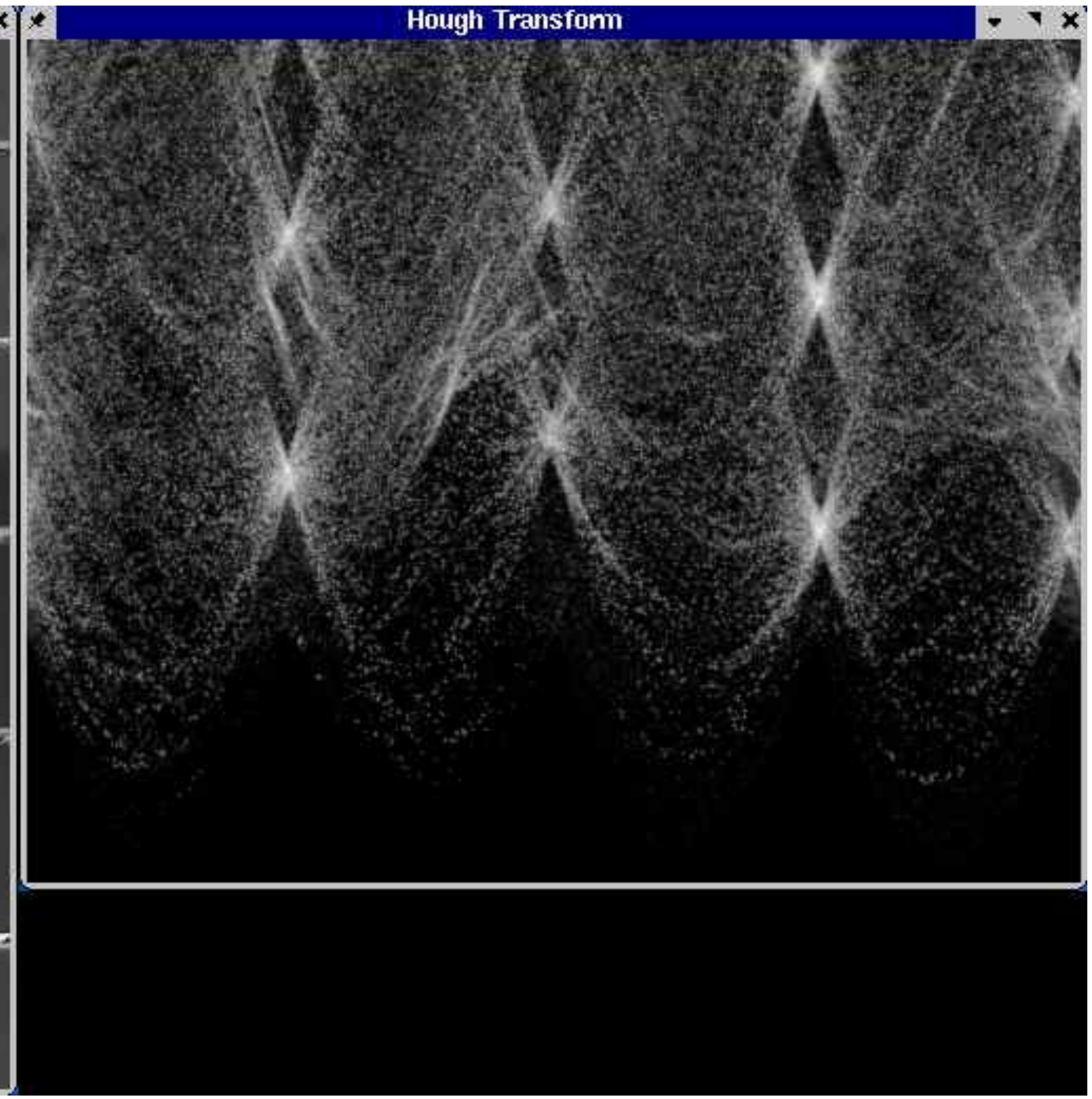
Image



Hough Space

Hough Transform

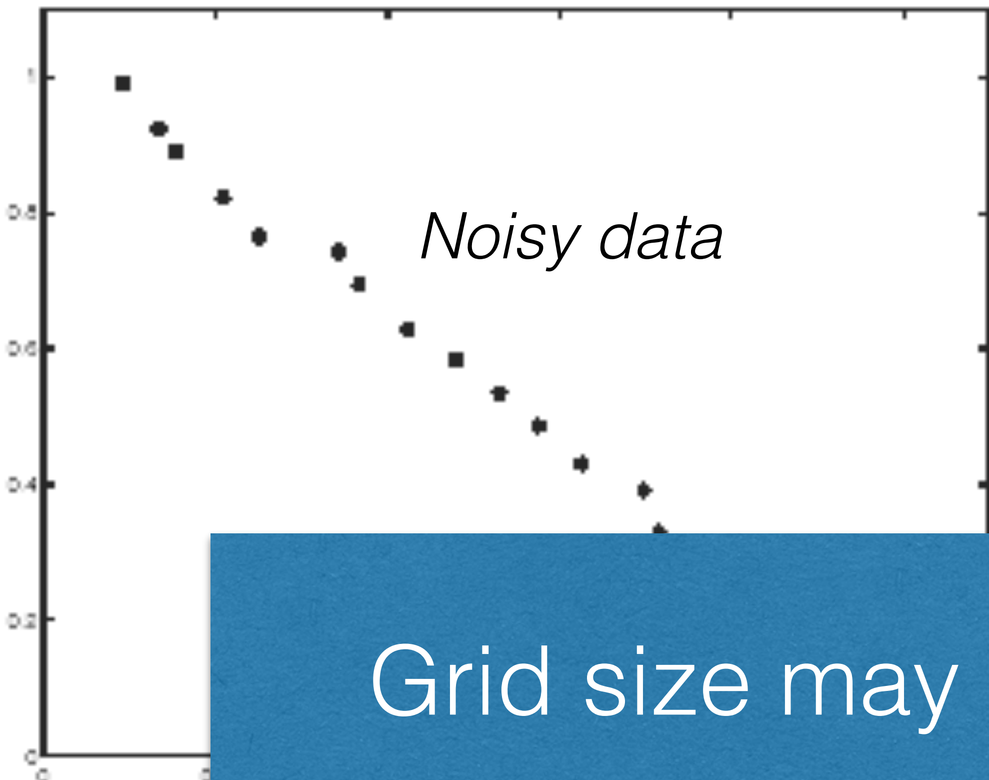
Image

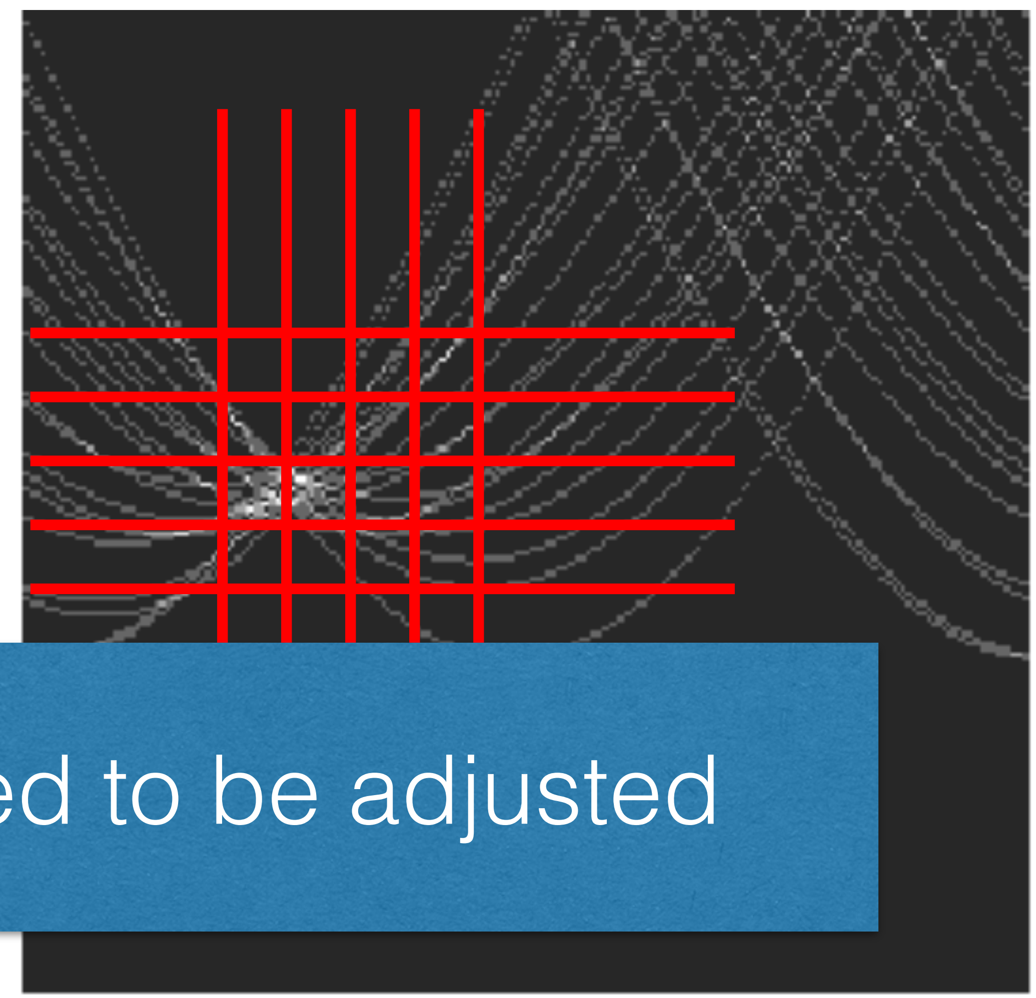


Hough Space

From http://ostatic.com/files/images/ss_hough.jpg

Hough Transform (Grid Size)





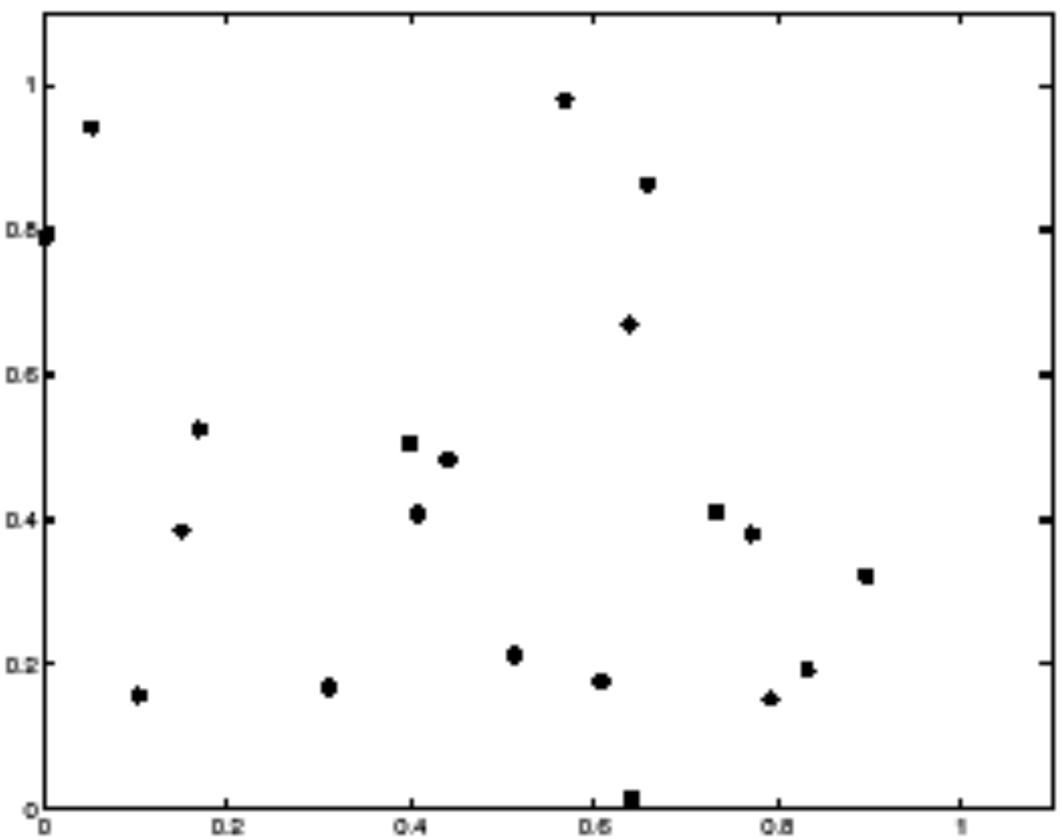
Grid size may need to be adjusted

features

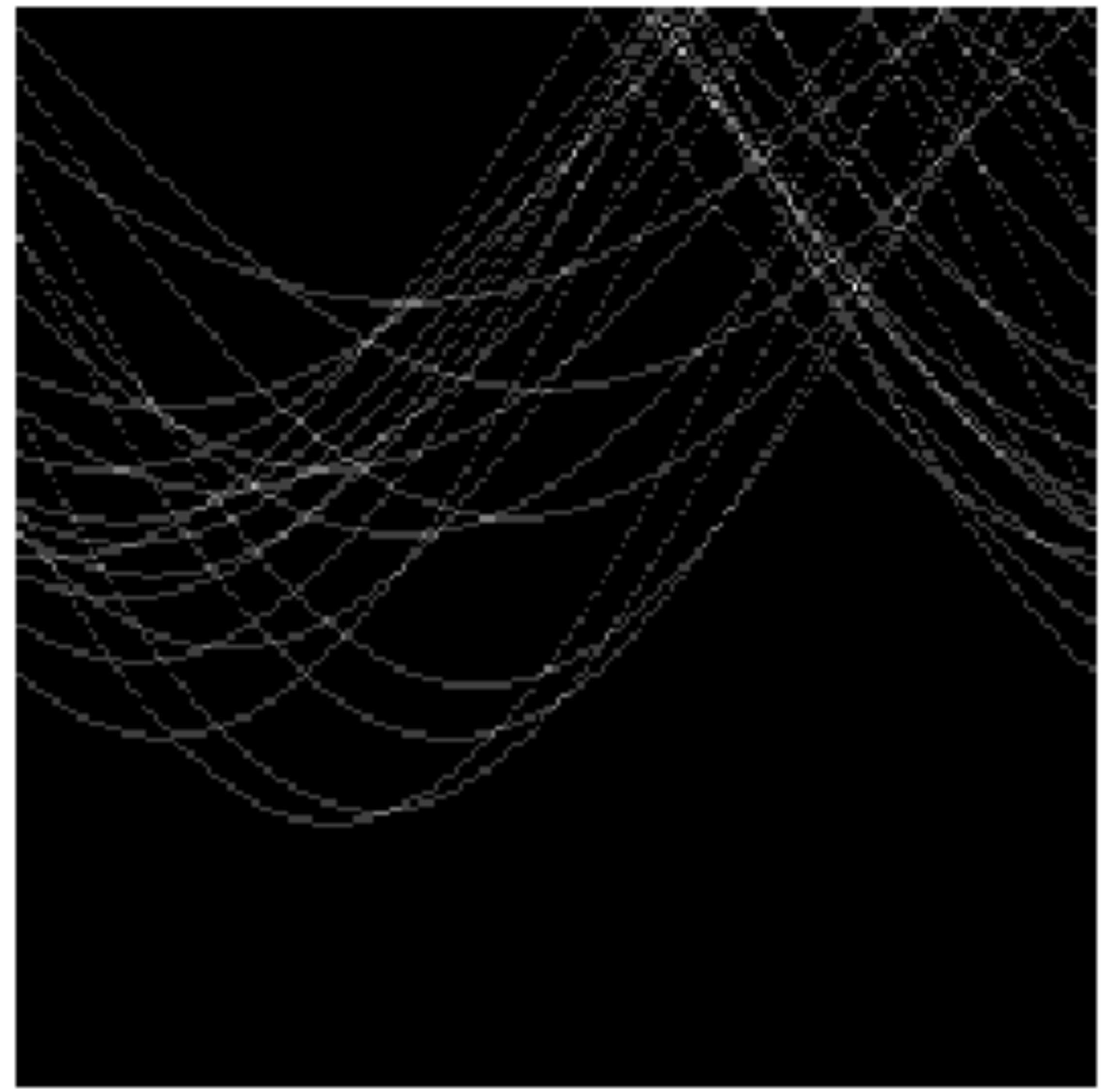
votes in hough space

Slide from S. Savarese

Hough Transform (Noise)



features

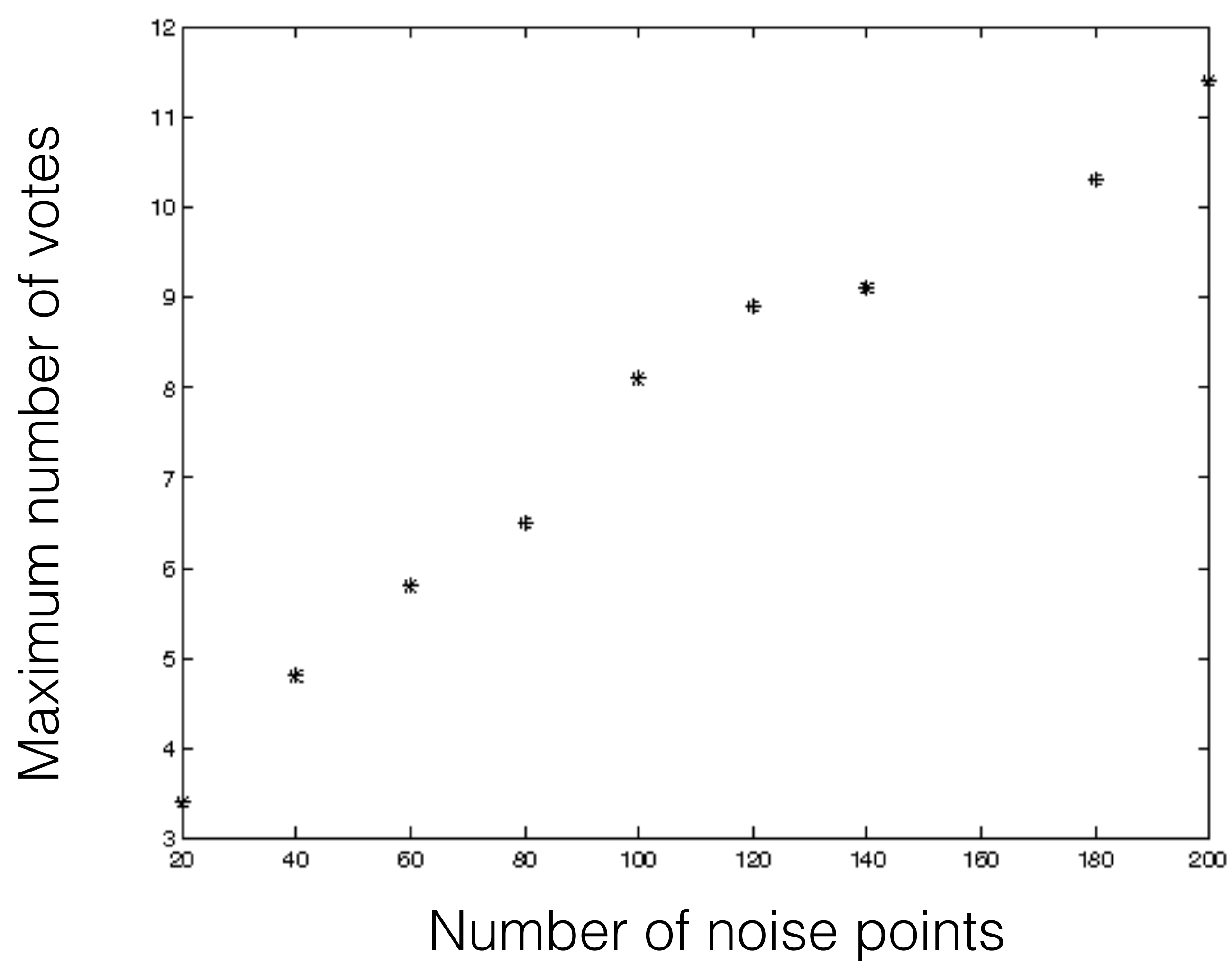


votes in hough space

Uniform noise can lead to spurious peaks in the array

Hough Transform (Noise)

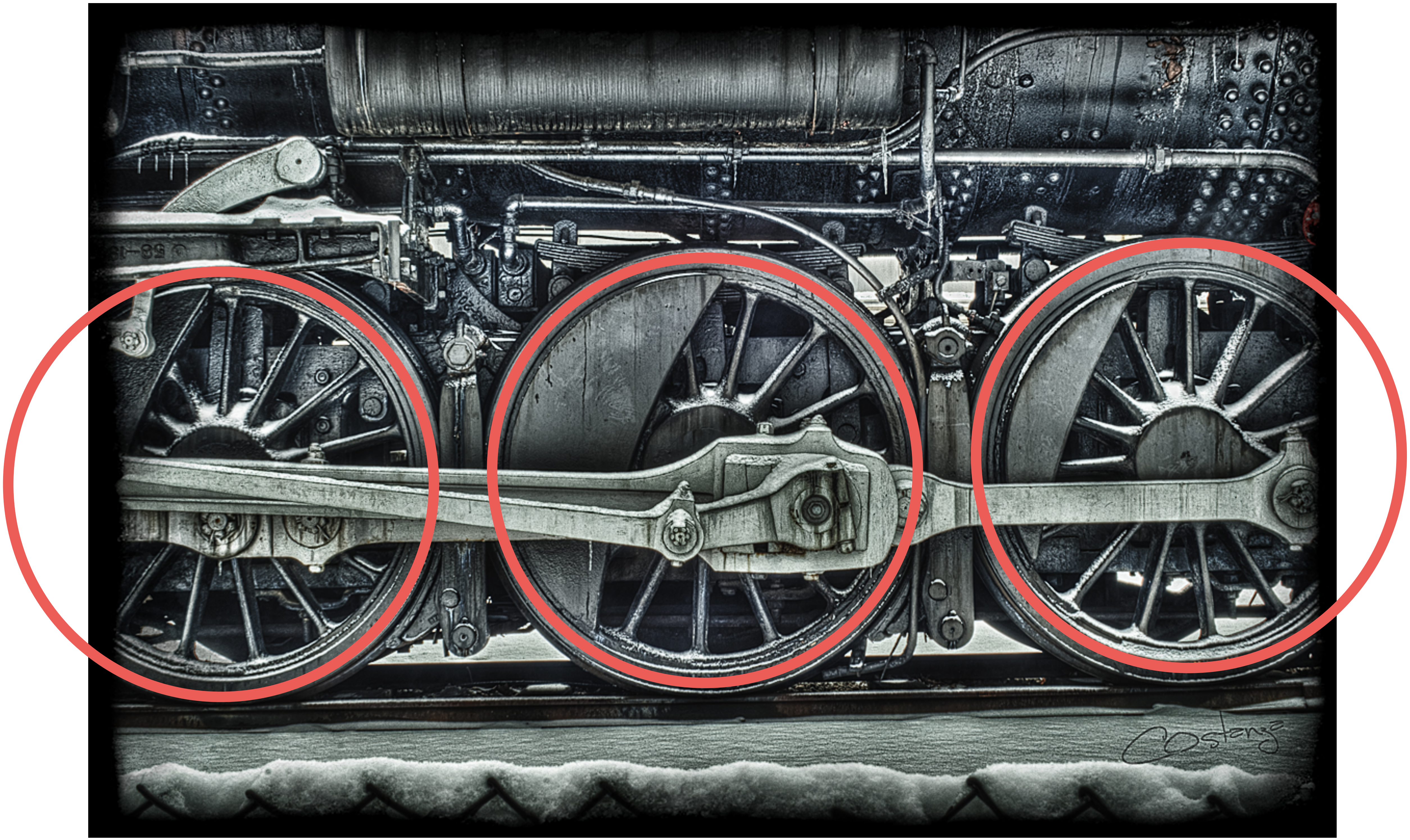
- As the level of uniform noise increases, the maximum number of votes increases too



Hough Transform (Dealing with Noise)

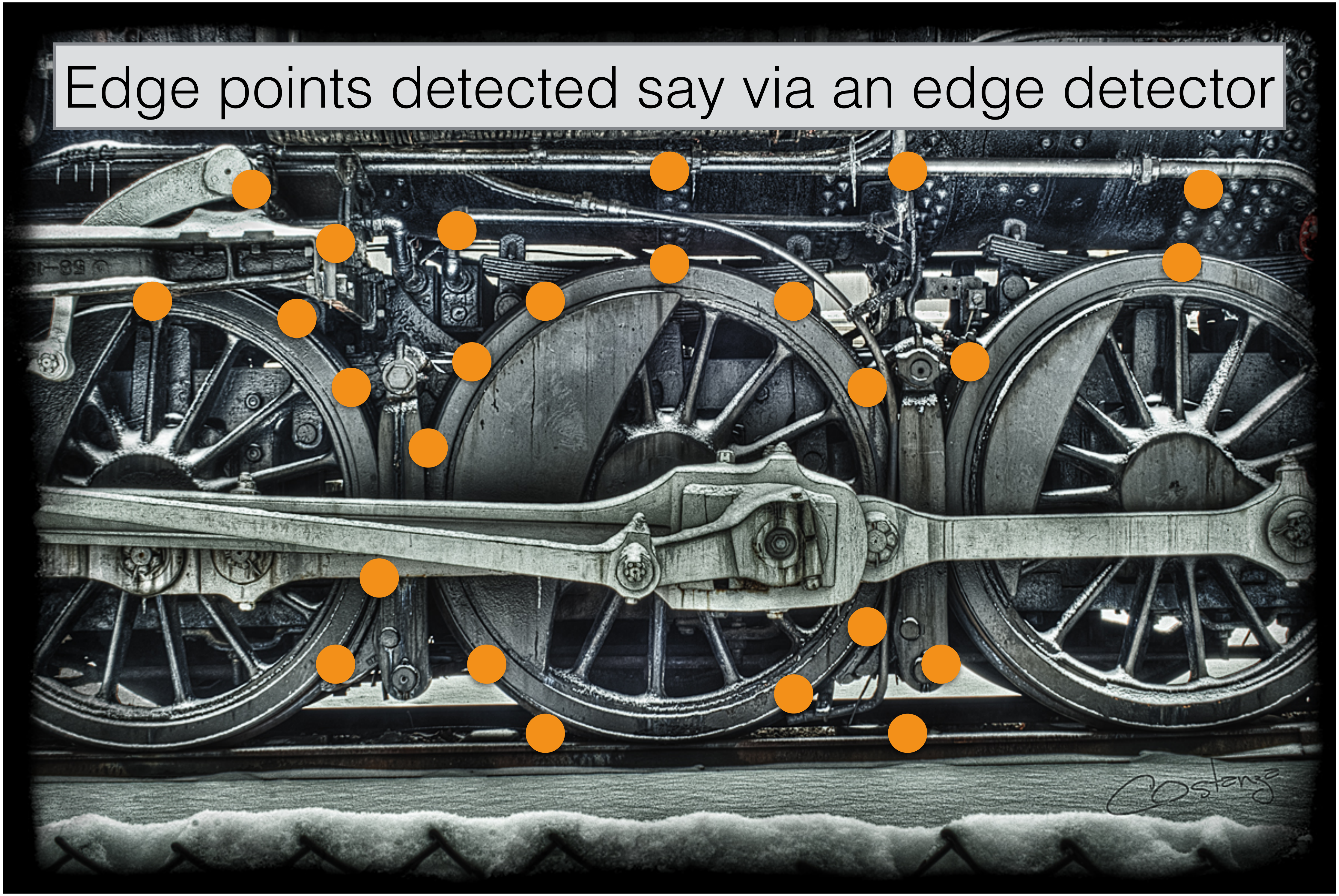
- Choose a good grid size (or discretization)
  - Too small: miss lines because only points that are *exactly* collinear will vote for a single line
  - Too large: many different lines will appear in the same bucket
- Increment neighbouring bins (smoothing accumulator arrays)
- Avoid irrelevant features.
  - Use pixels on *strong* edges (e.g., selected via gradient magnitude)

Hough Transform



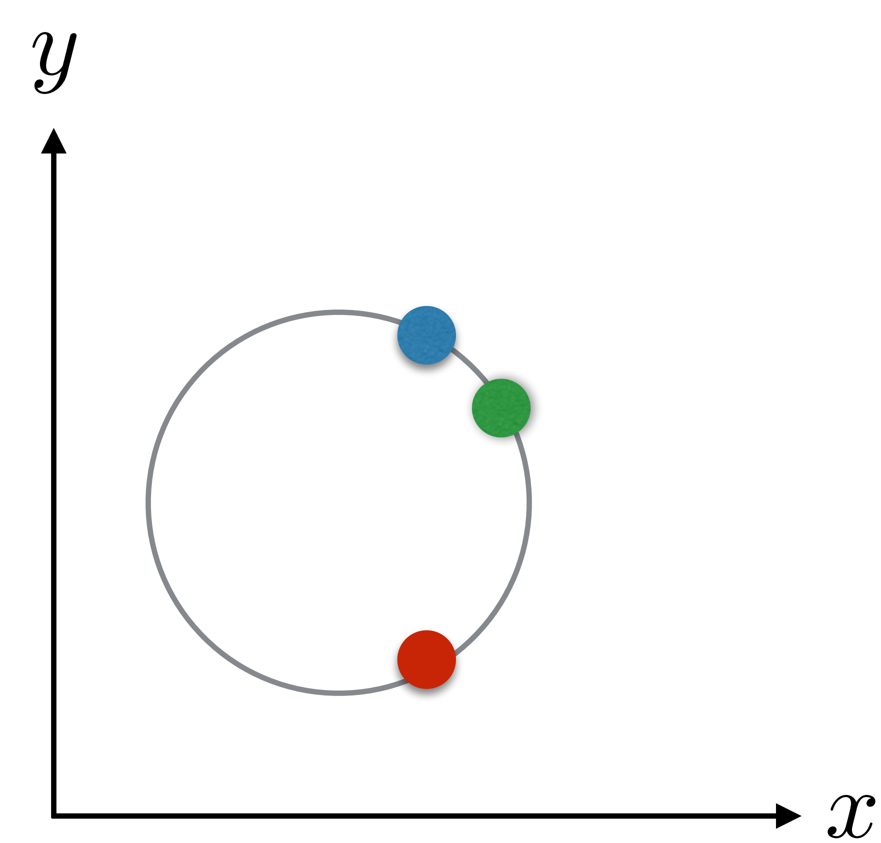
http://mappingignorance.org/2016/03/09/the-sweet-spot-of-wheel-rail-contact-modelling/

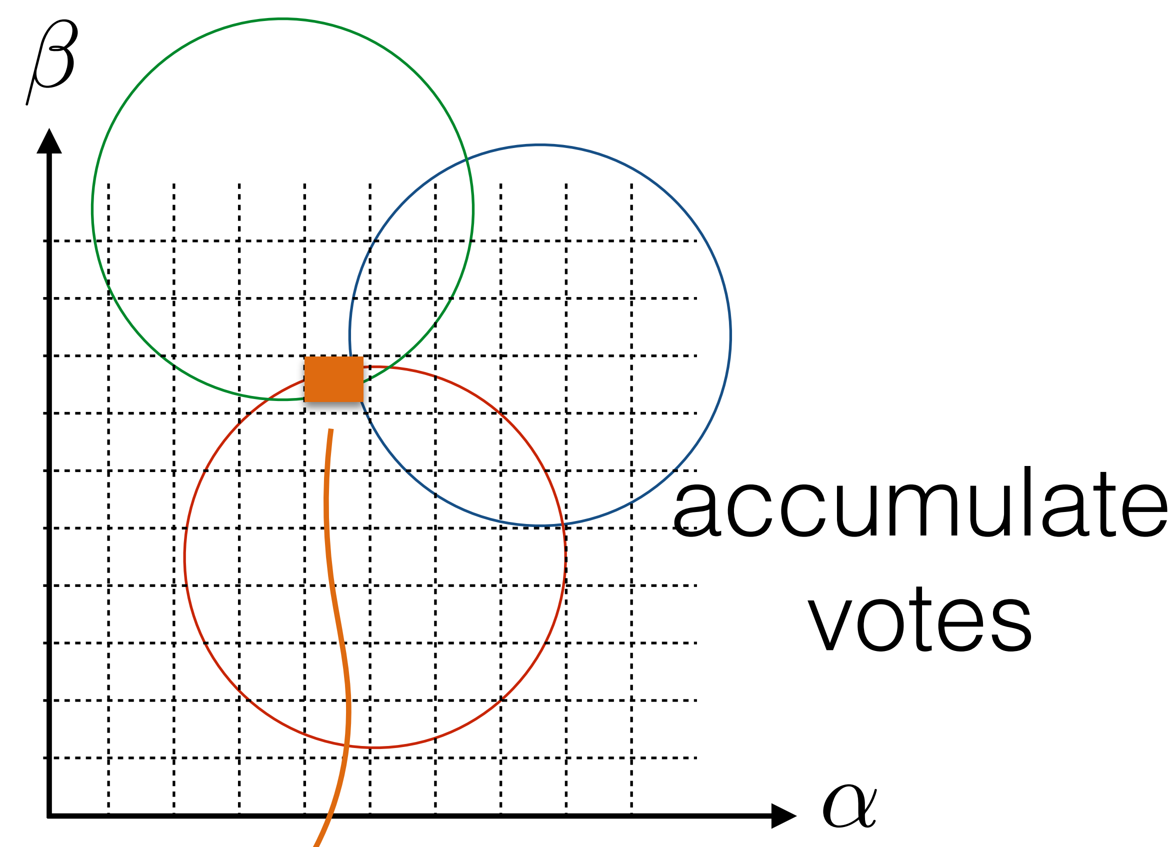
Hough Transform



Hough Transform

Assumption: fixed circle





$$(x-\alpha)^2+(y-\beta)^2=r^2$$

Unknowns

solution

Hough Transform

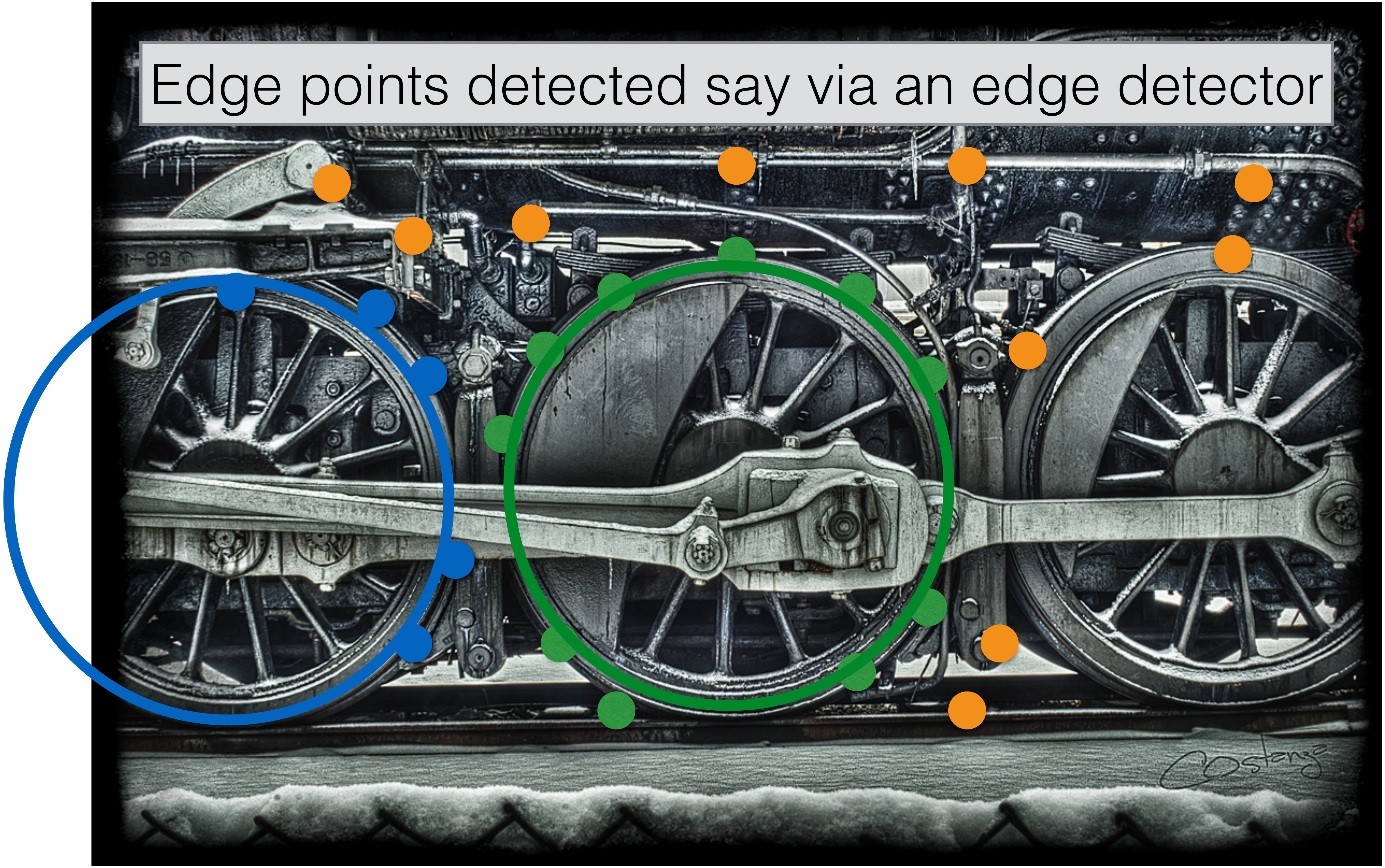
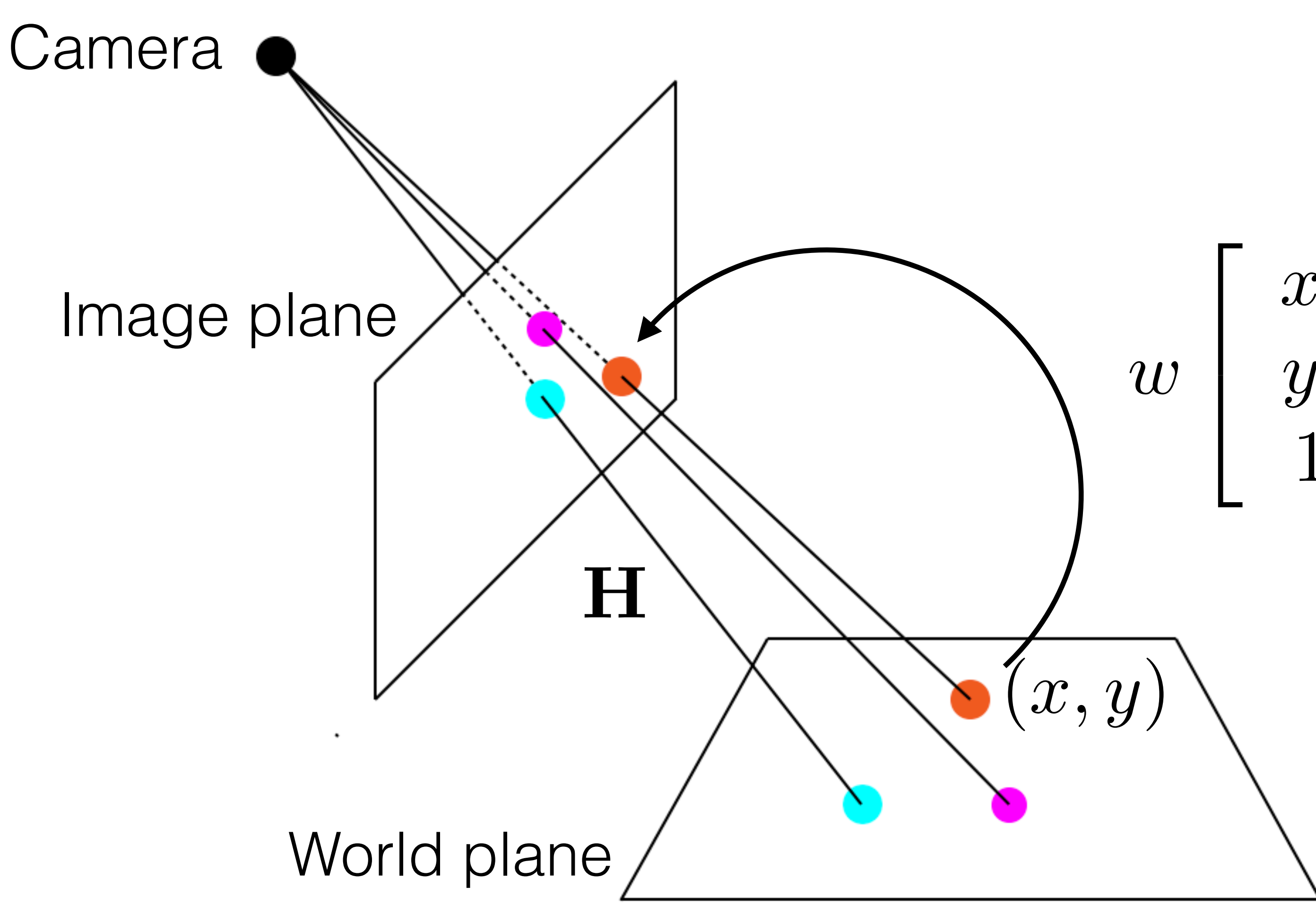


Image Stitching

Homography

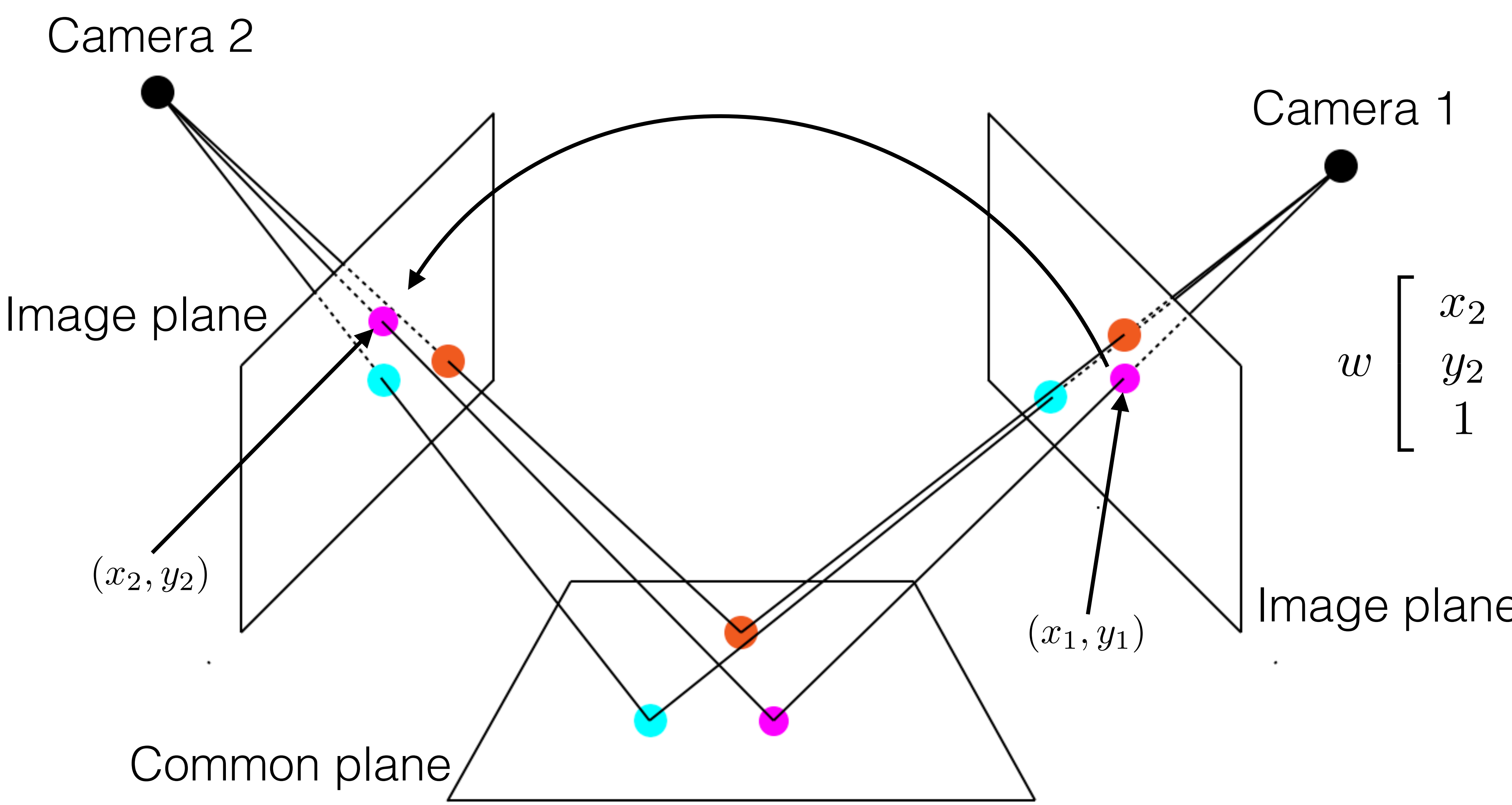
- Projection of a point in a world plane to the image plane. The location of the point and that of its projection is related via *homography*



$$w \begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \mathbf{H} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

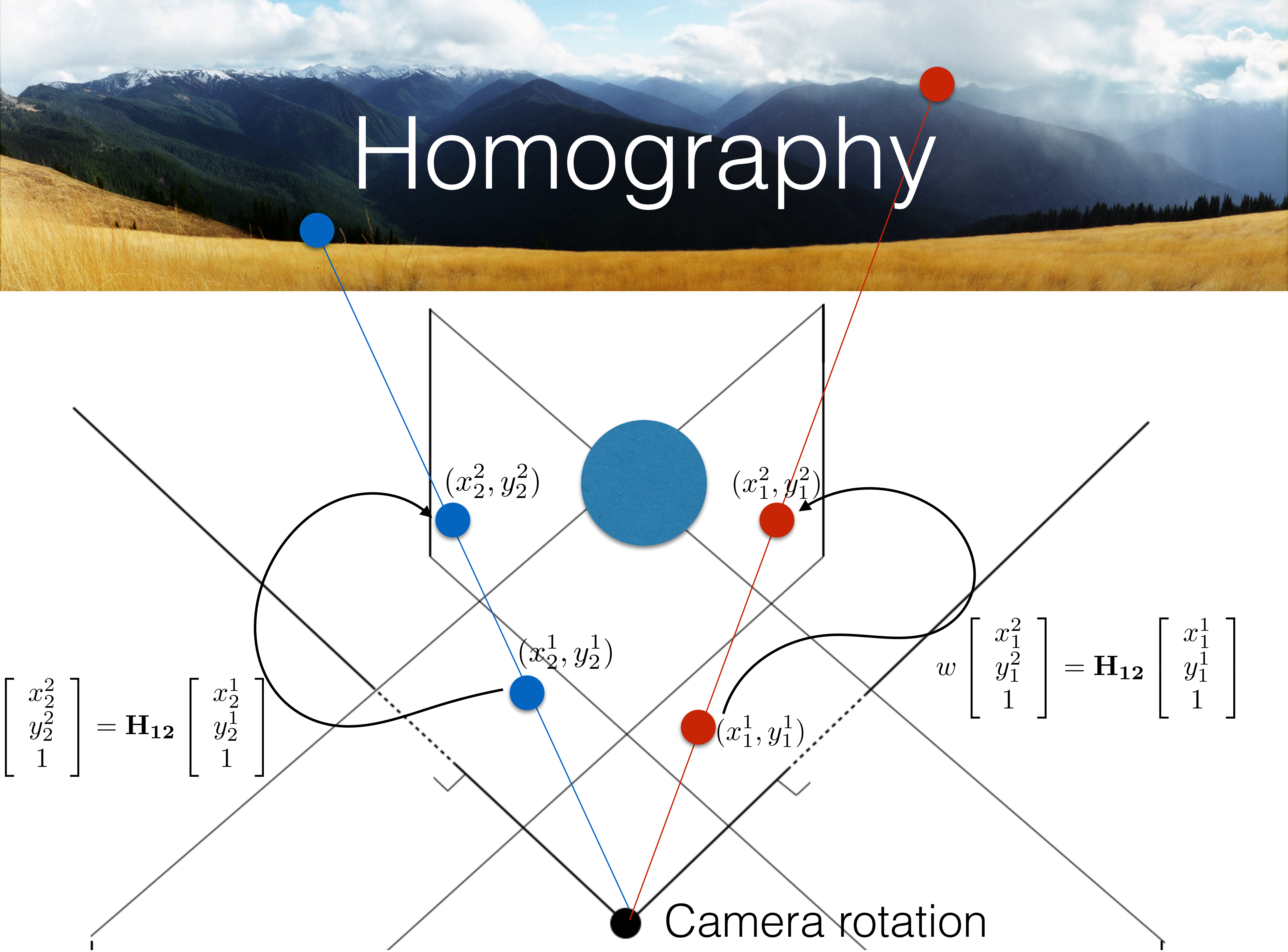
Homography

- Projections of points (**that lie on a common plane**) in two cameras are related via *homography*



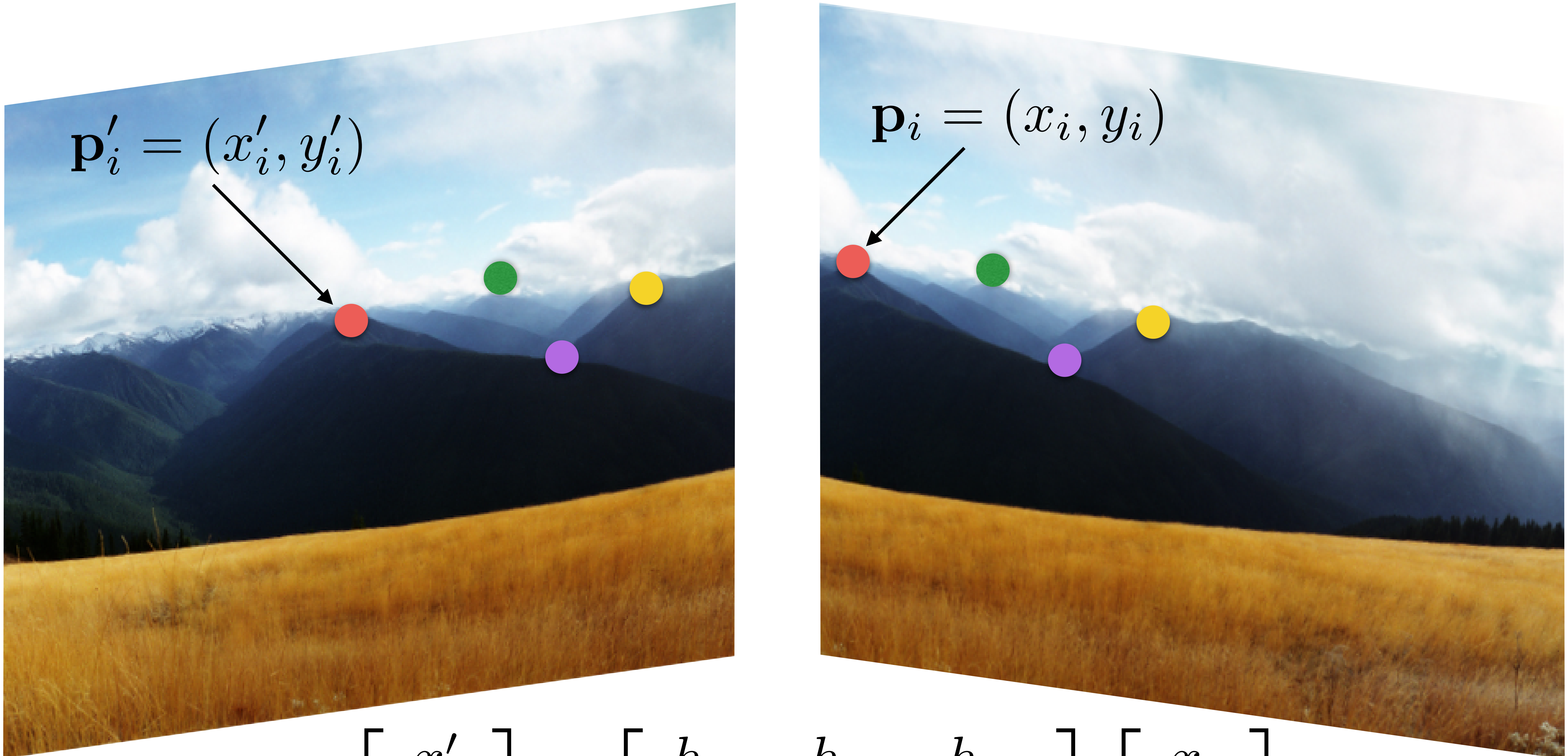
$$w \begin{bmatrix} x_2 \\ y_2 \\ 1 \end{bmatrix} = \mathbf{H_{12}} \begin{bmatrix} x_1 \\ y_1 \\ 1 \end{bmatrix}$$

Homography



$$\begin{bmatrix} x_2^2 \\ y_2^2 \\ 1 \end{bmatrix} = \mathbf{H_{12}} \begin{bmatrix} x_2^1 \\ y_2^1 \\ 1 \end{bmatrix}$$

$$w \begin{bmatrix} x_1^2 \\ y_1^2 \\ 1 \end{bmatrix} = \mathbf{H_{12}} \begin{bmatrix} x_1^1 \\ y_1^1 \\ 1 \end{bmatrix}$$



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

9 Unknowns

Need 4 correspondences

Can’t solve it using pseudo-inverse

Solving for *Homography* Using Point Correspondences

Re-write as homogeneous equations:

$$x'_i = \frac{h_{11}x_i + h_{12}y_i + h_{13}}{h_{31}x_i + h_{32}y_i + h_{33}} \qquad y'_i = \frac{h_{21}x_i + h_{22}y_i + h_{23}}{h_{31}x_i + h_{32}y_i + h_{33}}$$

Re-writing the above an matrix form:

$$\begin{bmatrix} x_i & y_i & 1 & 0 & 0 & 0 & -x_i x'_i & -y_i x'_i & -x'_i \\ 0 & 0 & 0 & x_i & y_i & 1 & -x_i y'_i & -y_i y'_i & -y'_i \end{bmatrix} \begin{bmatrix} h_{11} \\ h_{12} \\ h_{13} \\ h_{21} \\ h_{22} \\ h_{23} \\ h_{31} \\ h_{32} \\ h_{33} \end{bmatrix} = \mathbf{0}$$

Solving for *Homography* Using Point Correspondences

Setting up the $\mathbf{A}\mathbf{h} = \mathbf{0}$ problem using all available point correspondences

$$
\underbrace{\begin{bmatrix}
x_1 & y_1 & 1 & 0 & 0 & 0 & -x_1x'_1 & -y_1x'_1 & -x'_1 \\
0 & 0 & 0 & x_1 & y_1 & 1 & -x_1y'_1 & -y_1y'_1 & -y'_1 \\
\vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\
x_i & y_i & 1 & 0 & 0 & 0 & -x_ix'_i & -y_ix'_i & -x'_i \\
0 & 0 & 0 & x_i & y_i & 1 & -x_iy'_i & -y_iy'_i & -y'_i \\
\vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\
x_n & y_n & 1 & 0 & 0 & 0 & -x_nx'_n & -y_nx'_n & -x'_n \\
0 & 0 & 0 & x_n & y_n & 1 & -x_ny'_n & -y_ny'_n & -y'_n
\end{bmatrix}}_{\mathbf{A}}
\underbrace{\begin{bmatrix}
h_{11} \\ h_{12} \\ h_{13} \\ h_{21} \\ h_{22} \\ h_{23} \\ h_{31} \\ h_{32} \\ h_{33}
\end{bmatrix}}_{\mathbf{h}}
= \mathbf{0}
$$

$$= \mathbf{0}$$

Solution is the *Null Space* of $\mathbf{A}$

Solving for *Homography* Using Point Correspondences

Solution

Need a minimum of 4 point correspondences

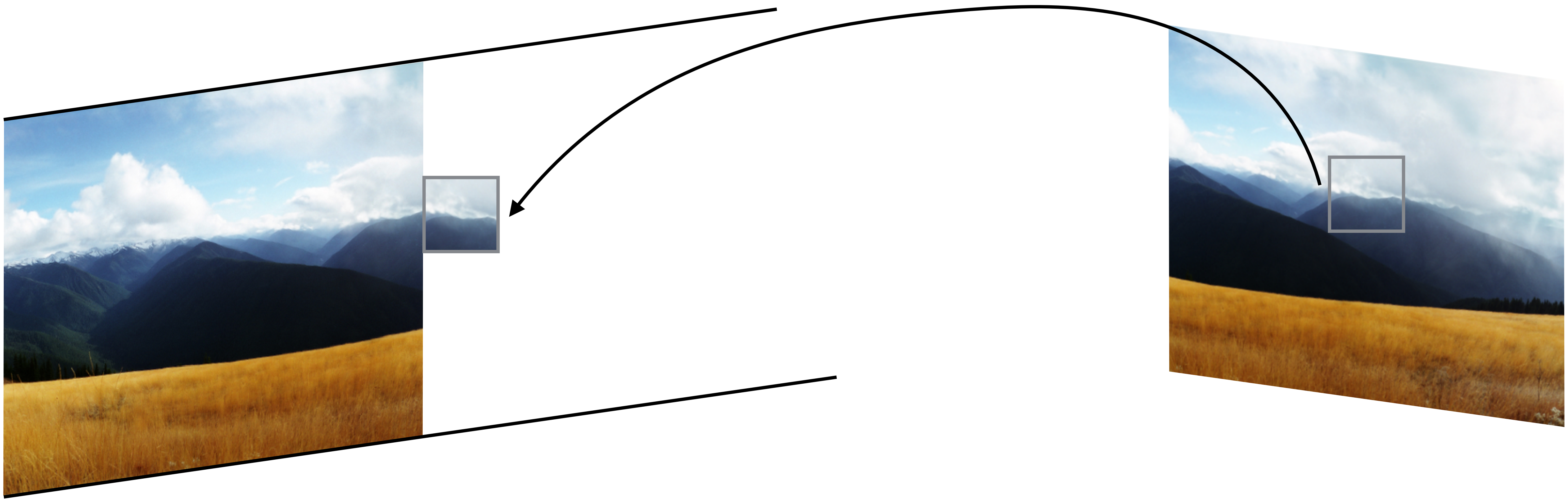
Estimate using least-square fitting

$$\mathbf{h}^* = \underset{\mathbf{h}}{\operatorname{argmin}} \|\mathbf{A}\mathbf{h}\|^2 \text{ s.t. } \|\mathbf{h}\| = 1$$

Solution is the eigenvector corresponding to the smallest eigenvalue of $\mathbf{A}^T\mathbf{A}$

Where have seen this before ?

Image Stitching



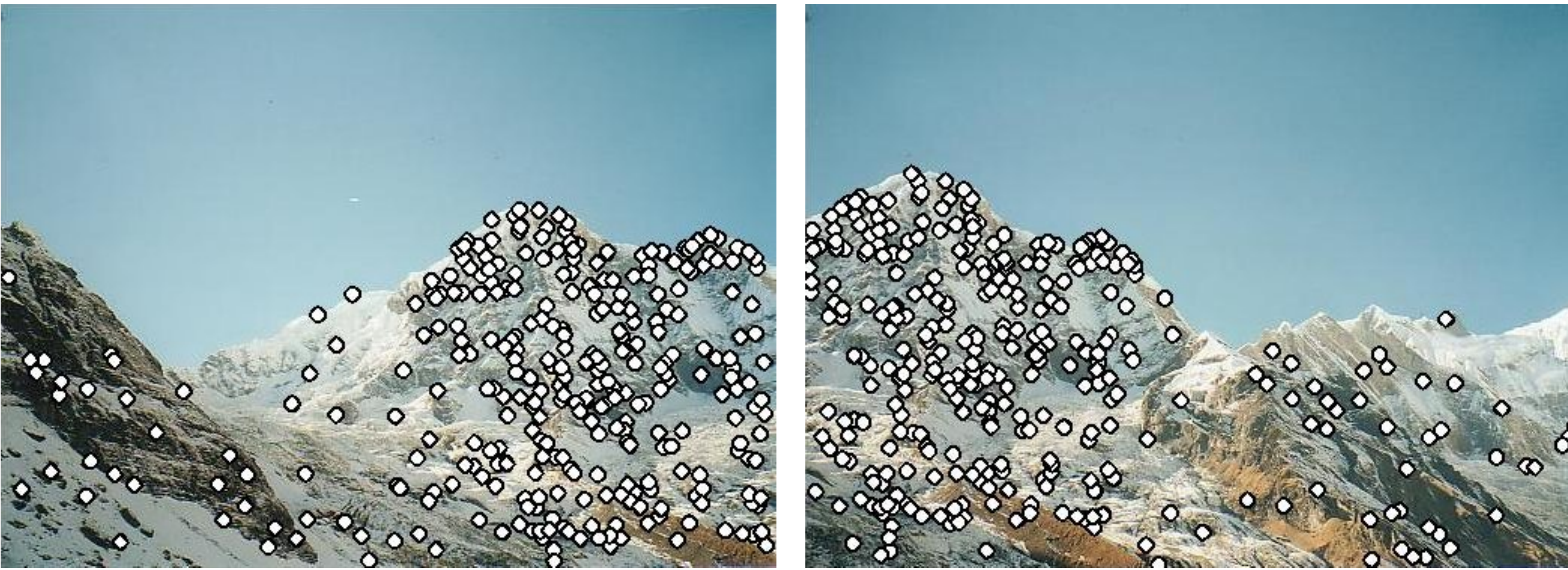
Estimate homography

Use it to fill in colors from the other image

Image Stitching

Source: L. Lazebnik

Image Stitching



Extract features

Source: L. Lazebnik

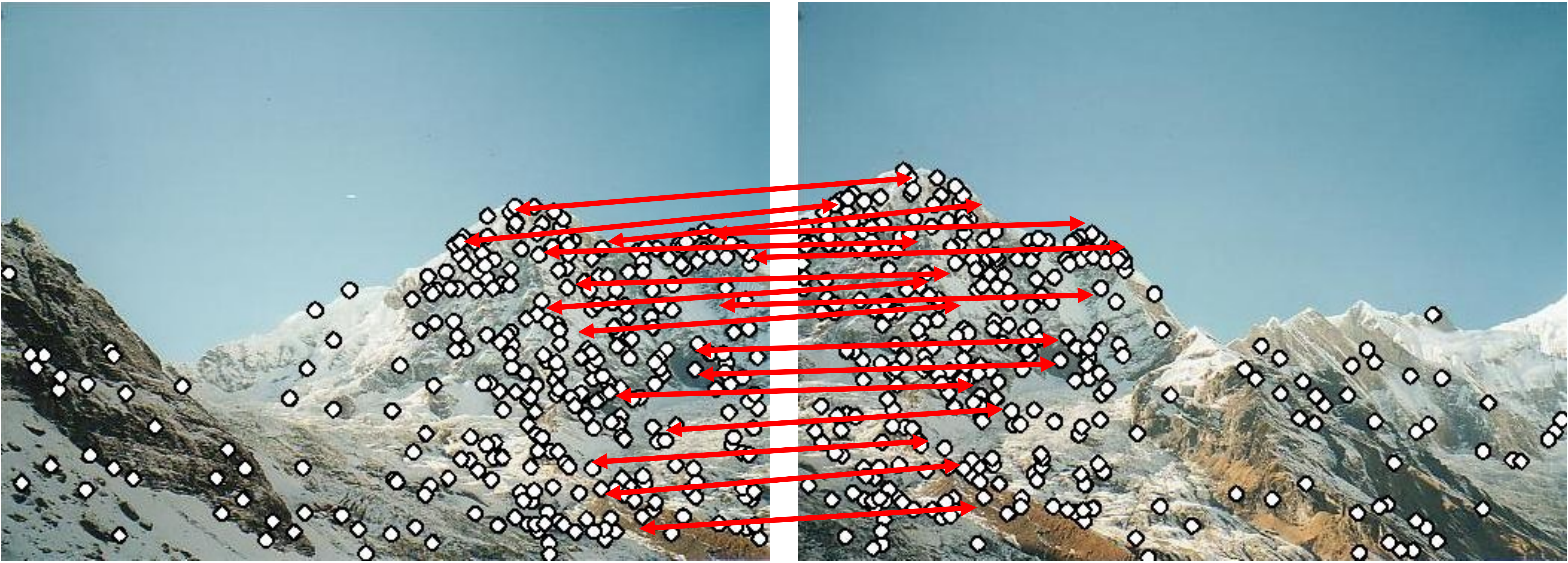
Image Stitching

Extract features

Compute *putative* matches

Source: L. Lazebnik

Image Stitching



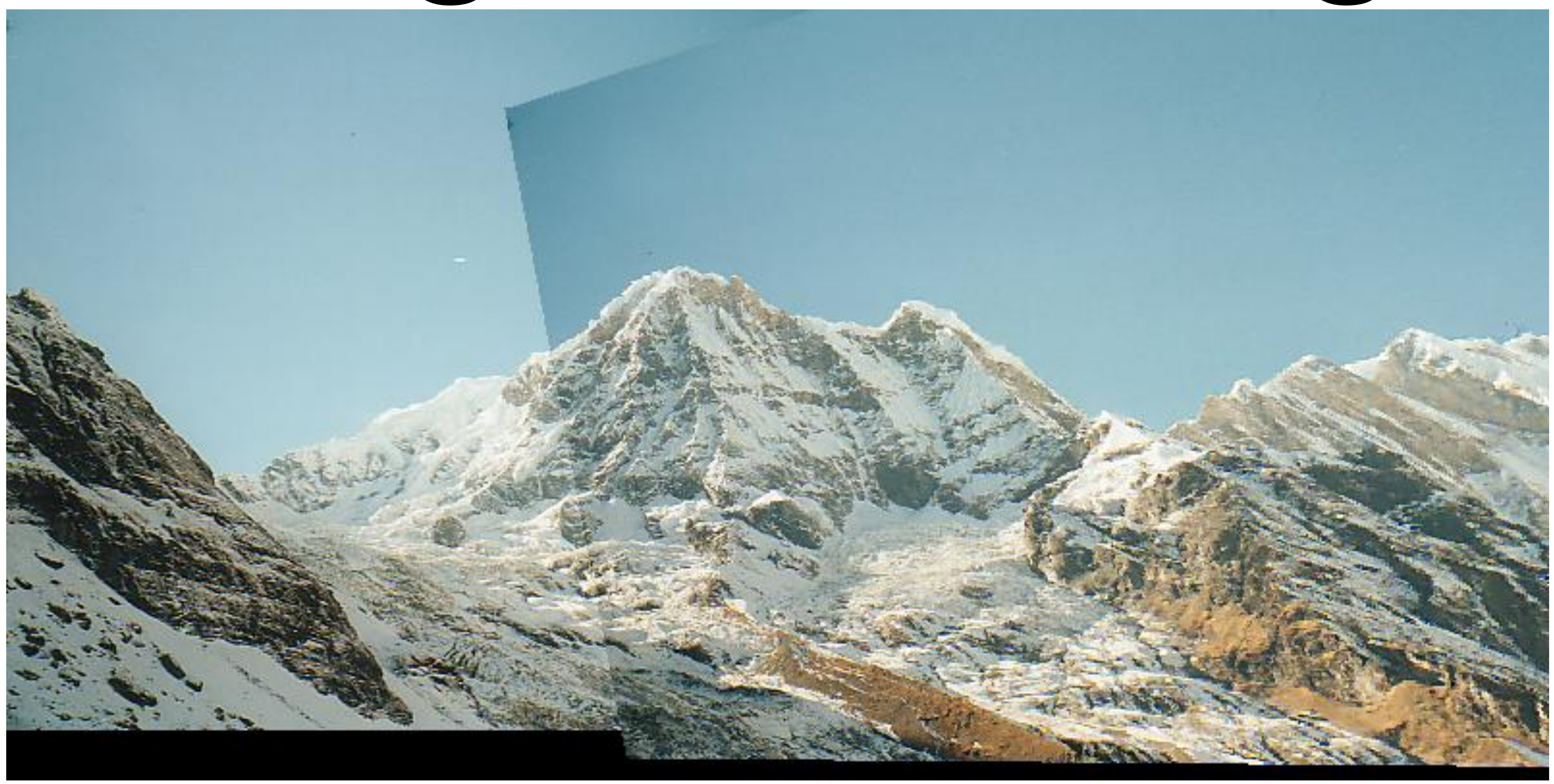
Extract features

Compute *putative* matches

Use RANSAC to estimate *homography*

Source: L. Lazebnik

Image Stitching



Extract features

Compute *putative* matches

Use RANSAC to estimate *homography*

Perform image stitching

Source: L. Lazebnik

Summary

- Model fitting
- Least squares
- Robust least squares
- RANSAC
- Hough transform
- Example: Image stitching