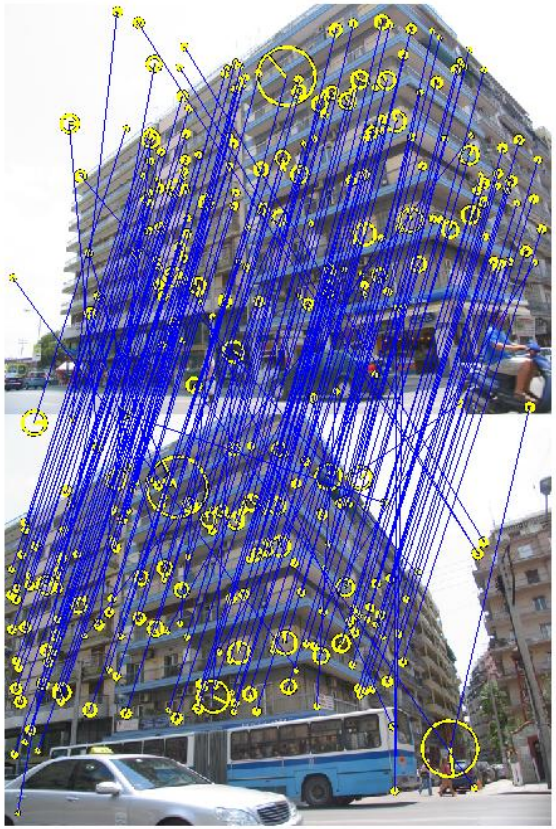
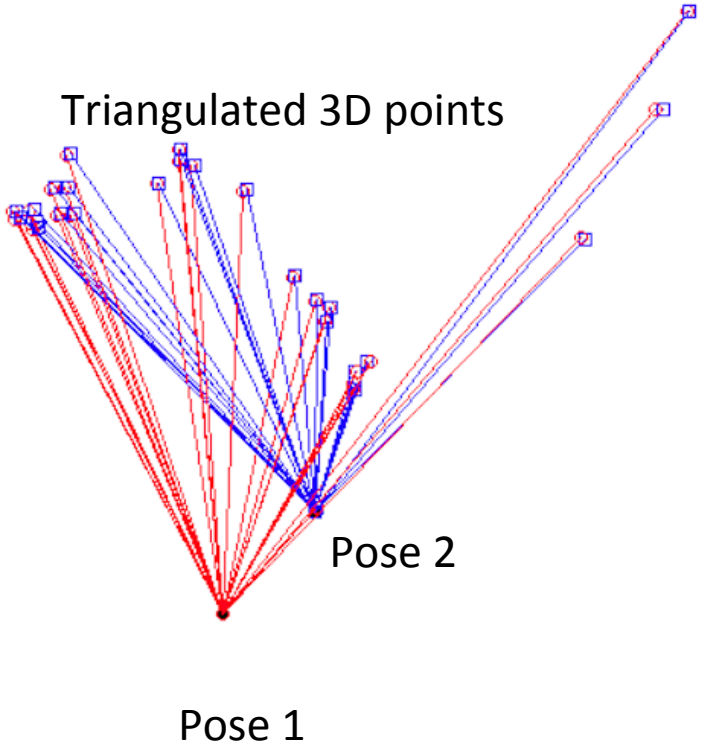
Perception: Visual Features

Kostas Daniilidis

What do we need features for?

- For finding points so that we solve localization and reconstruction





What do we need features for?

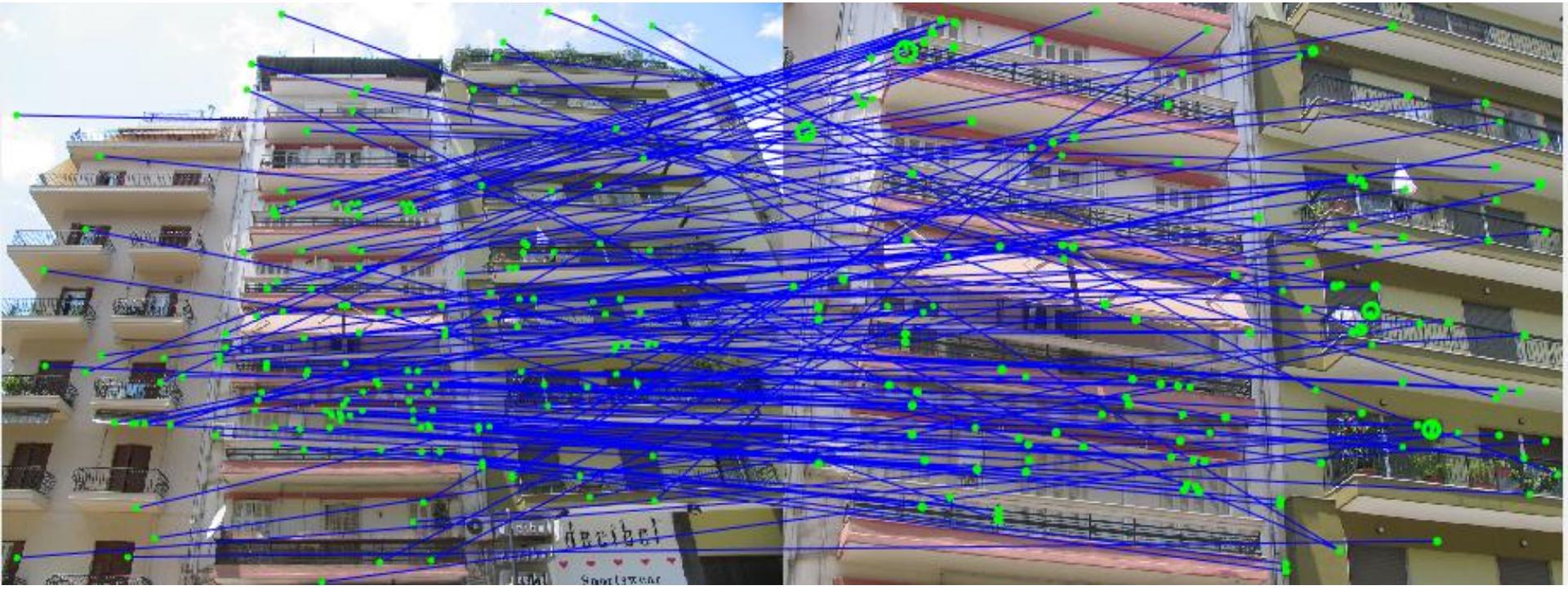
- For place recognition and other retrieval tasks

What we want from the features?

1. Detection repeatability:
   – When they are detected in one image to be detected in another one from the same scene even if image differs in scale and orientation

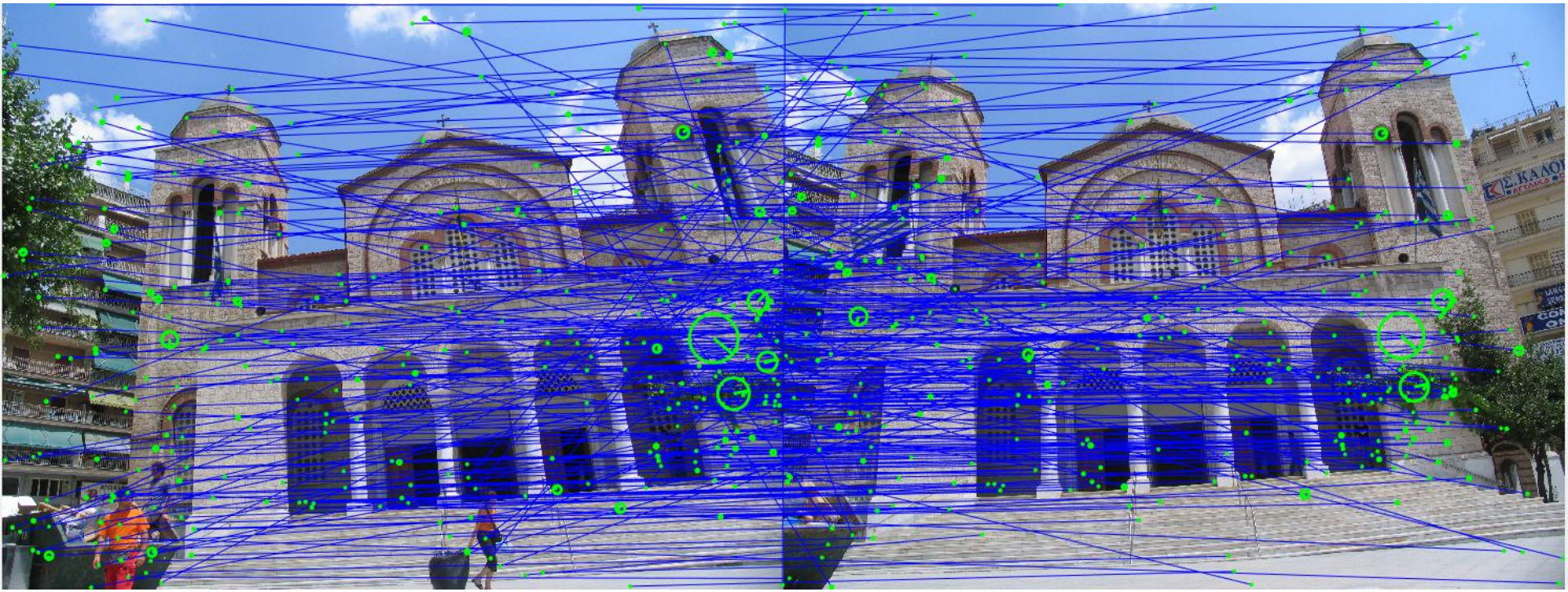
What we want from the features?

Features should be detected again even under severe scale changes. We call this property detection invariance.



What else do we want from features?

- We should be able to match features using a descriptor of the neighborhood.
- This descriptor should not change significantly under viewpoint changes like scale and rotation.
- We call this property descriptor invariance.



Invariant detection and description

Probably the most challenging of all properties is the scale invariance!

Tony Lindeberg

Feature detection with automatic scale selection

Authors	Tony Lindeberg
Publication date	1998/11/1
Journal	International Journal of Computer Vision
Volume	30
Issue	2

David Lowe

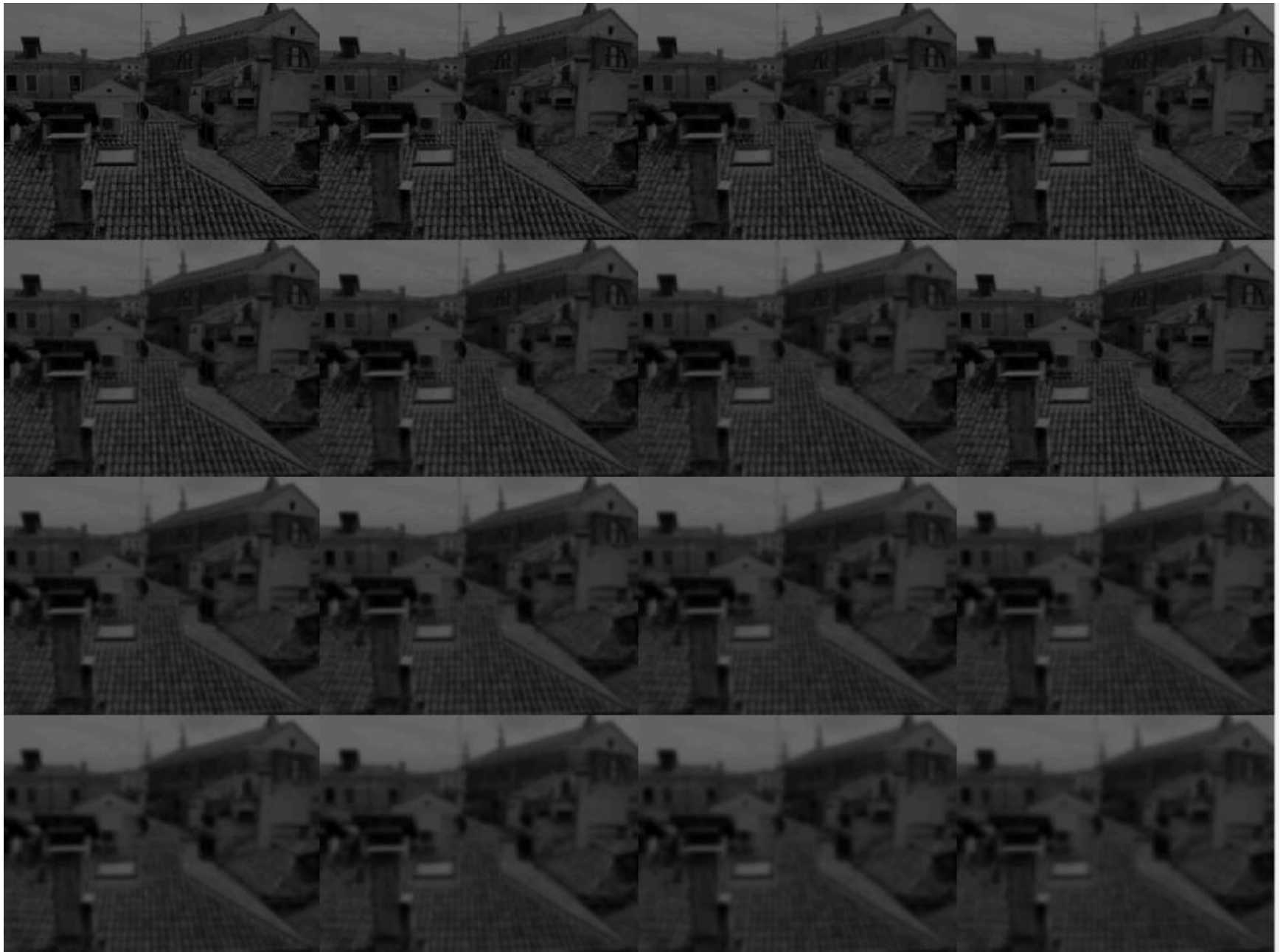
Distinctive image features from scale-invariant keypoints

Authors	David G Lowe
Publication date	2004/11/1
Journal	International journal of computer vision
Volume	60
Issue	2

SIFT: Scale Invariant Feature Transform

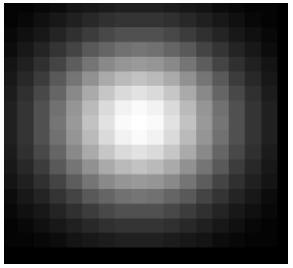
Title 1–20	Cited by	Year
Distinctive image features from scale-invariant keypoints DG Lowe International journal of computer vision 60 (2), 91-110	34581	2004
Object recognition from local scale-invariant features DG Lowe International Conference on Computer Vision, 1999, 1150-1157	10955	1999

The notion of scale space



How is scale space built?

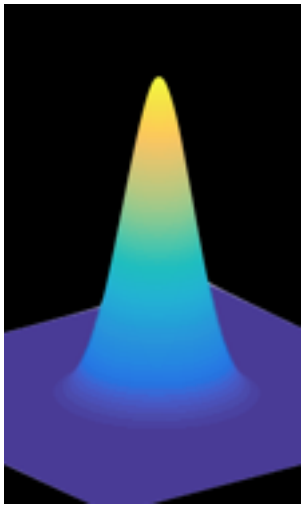
$*$



2D Gaussian

$=$

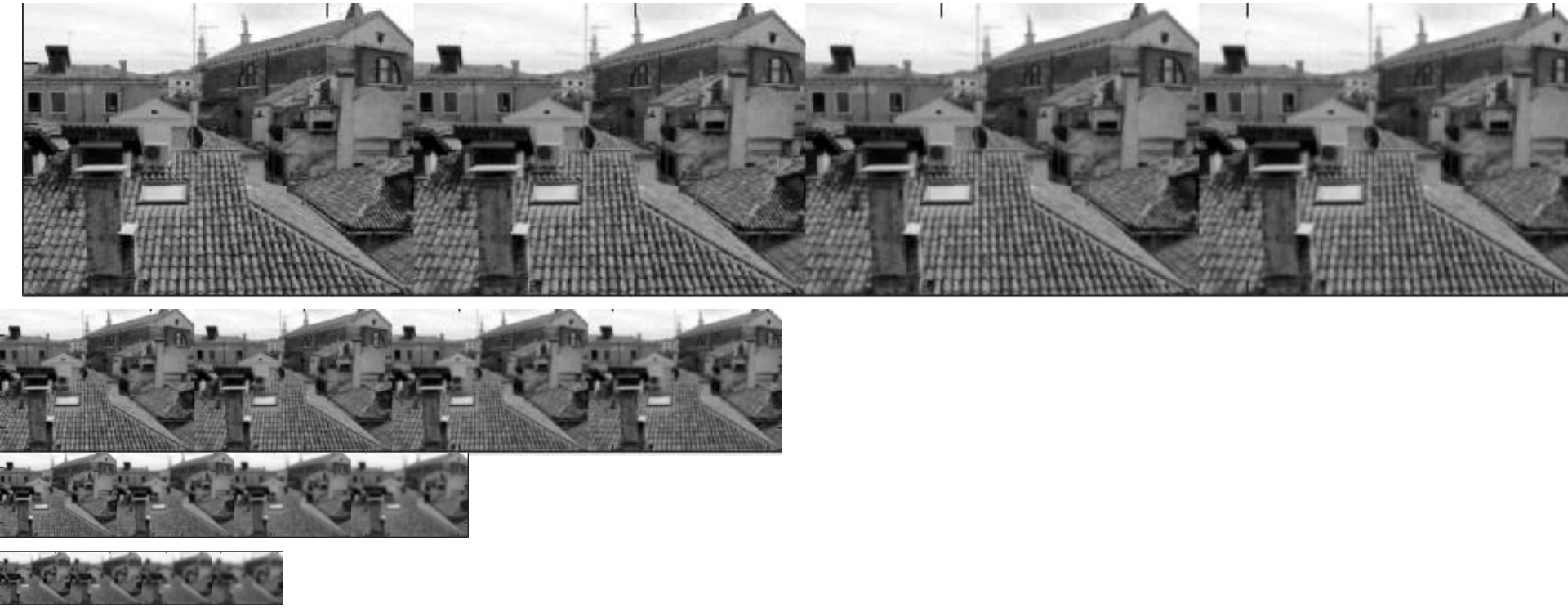
$*$ means convolution



$$L(x, y, \sigma) = G(x, y, \sigma) * I(x, y)$$

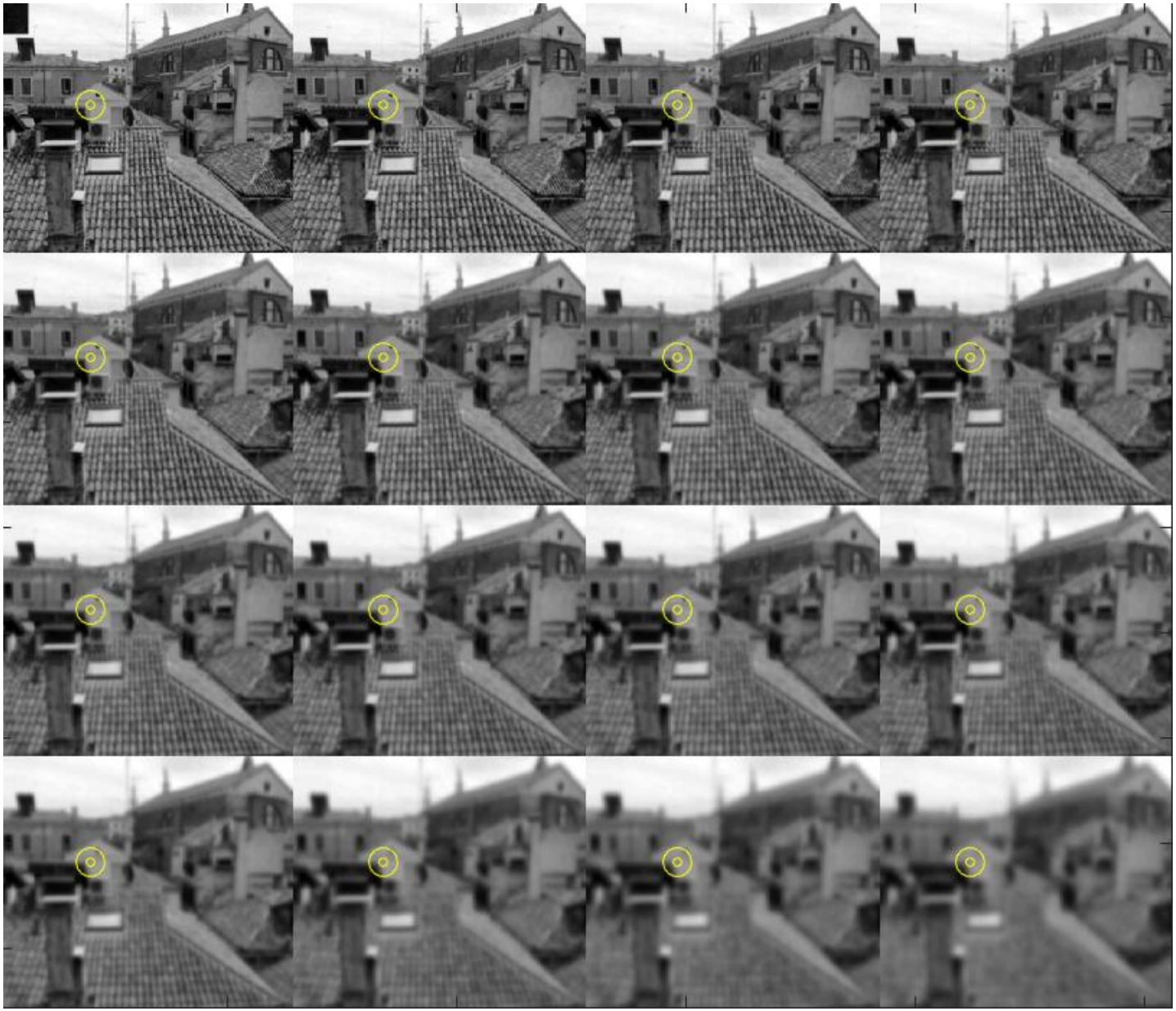
$$G(x, y, \sigma) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}}$$

The same scale but subsampled

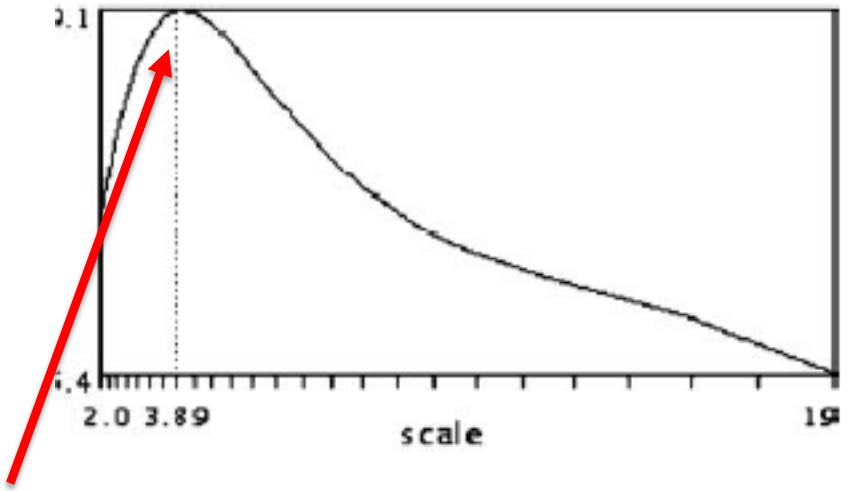


We subsampled it every time that the sigma of the Gaussian was doubled!

Now look at the same pixel across scale

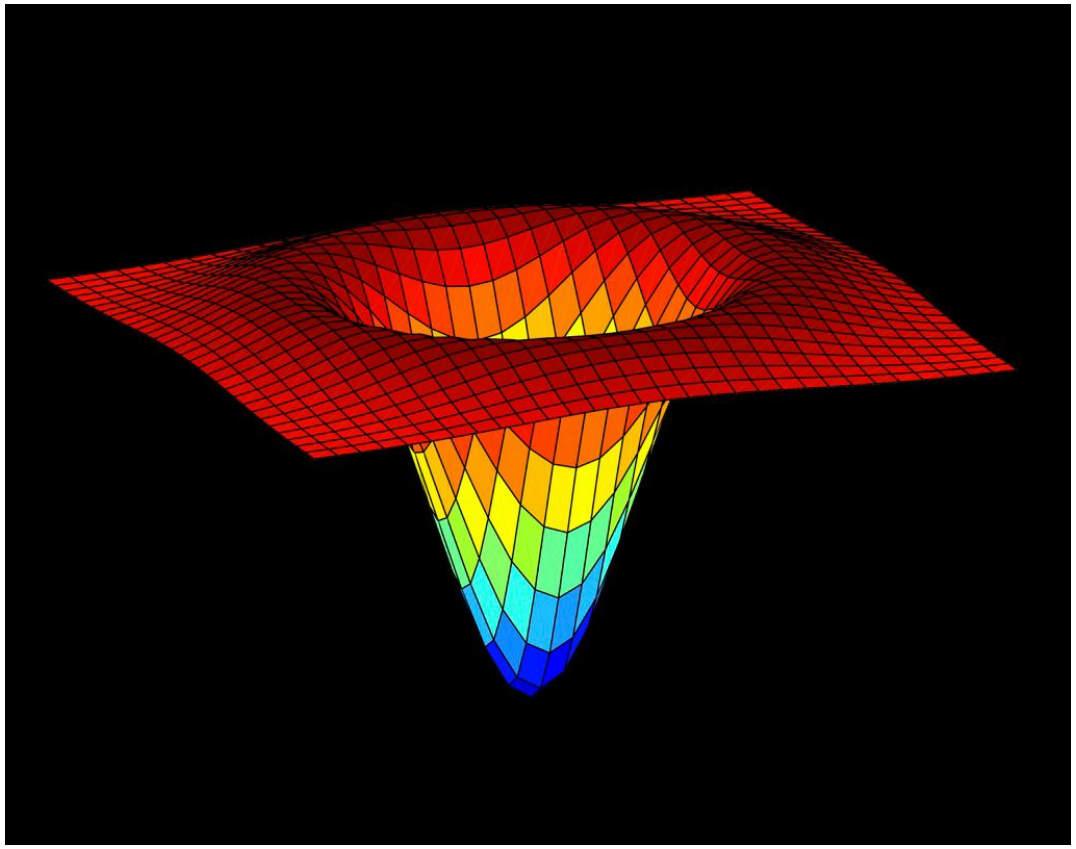


Scale selection



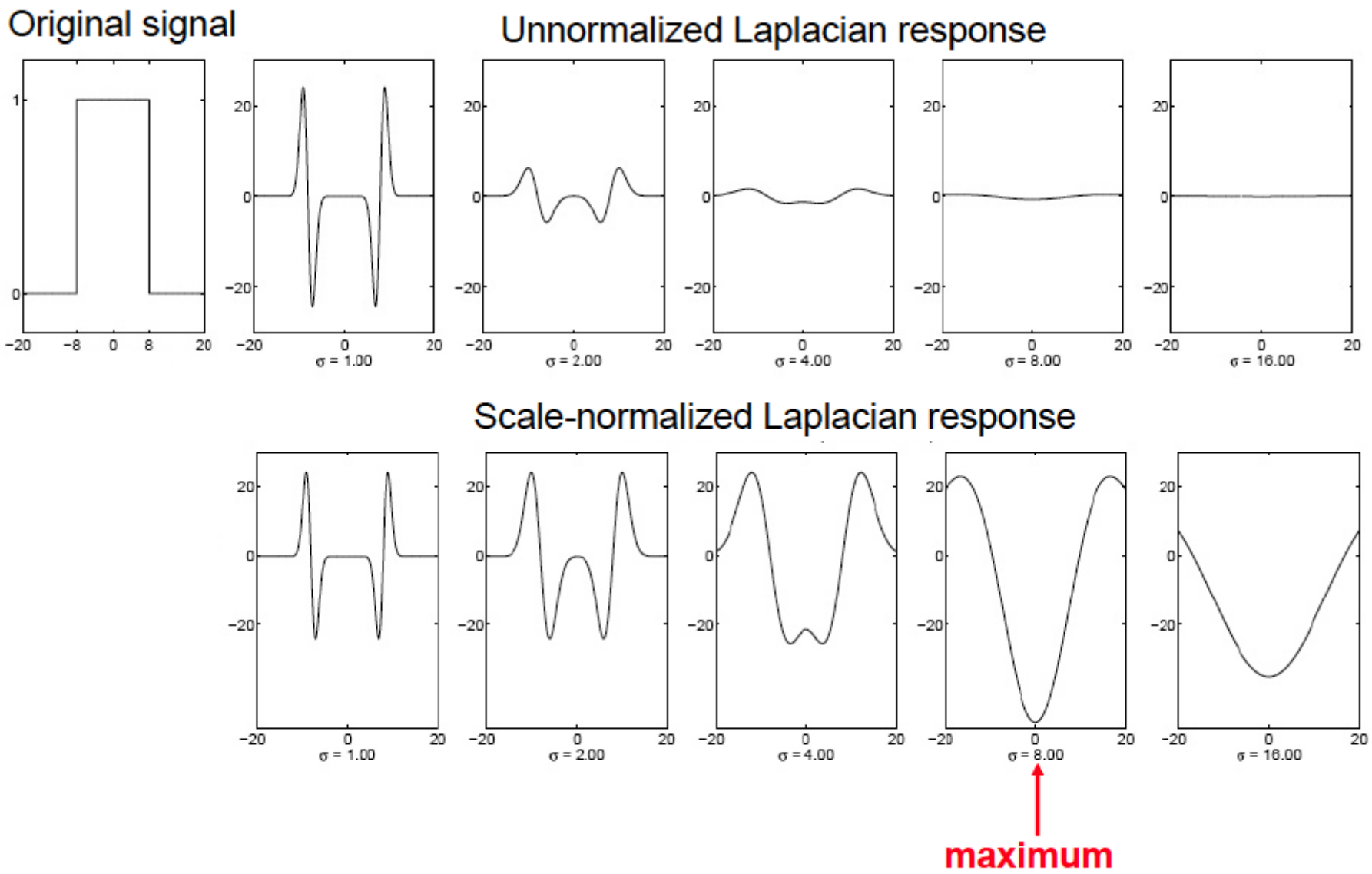
- The maximum across scale is the intrinsic scale of the image structure
- if the smoothed value is scale normalized.
- It turns out that only the derivatives of the Gaussian responses can be normalized.

We choose the 2nd derivative (trace of Hessian) , called Laplacian of Gaussian (LoG)



Which has the nice property that it can be approximated as the difference of two Gaussians and can detect blob like features!

Why do we need normalization?

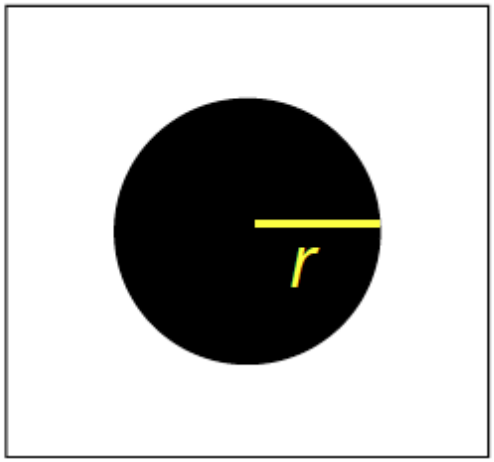


Scale selection

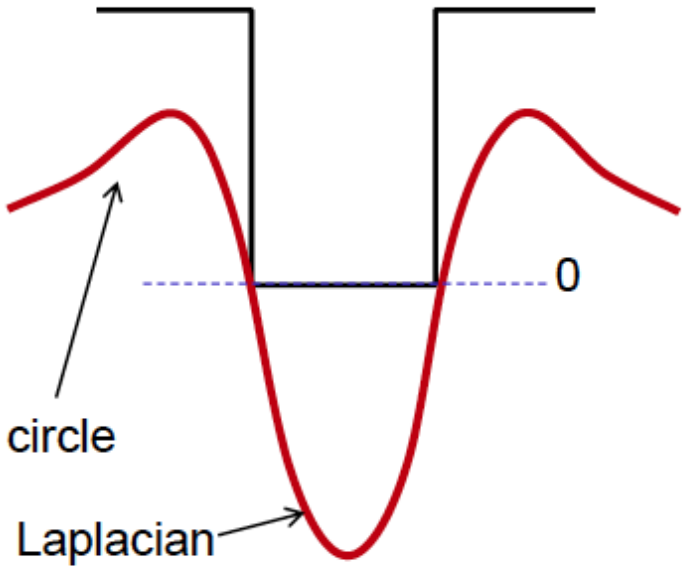
- At what scale does the Laplacian achieve a maximum response to a binary circle of radius r?
- To get maximum response, the zeros of the Laplacian have to be aligned with the circle
- The Laplacian is given by (up to scale):

$$(x^2 + y^2 - 2\sigma^2)\, e^{-(x^2+y^2)/2\sigma^2}$$

- Therefore, the maximum response occurs at $\sigma = r/\sqrt{2}$.

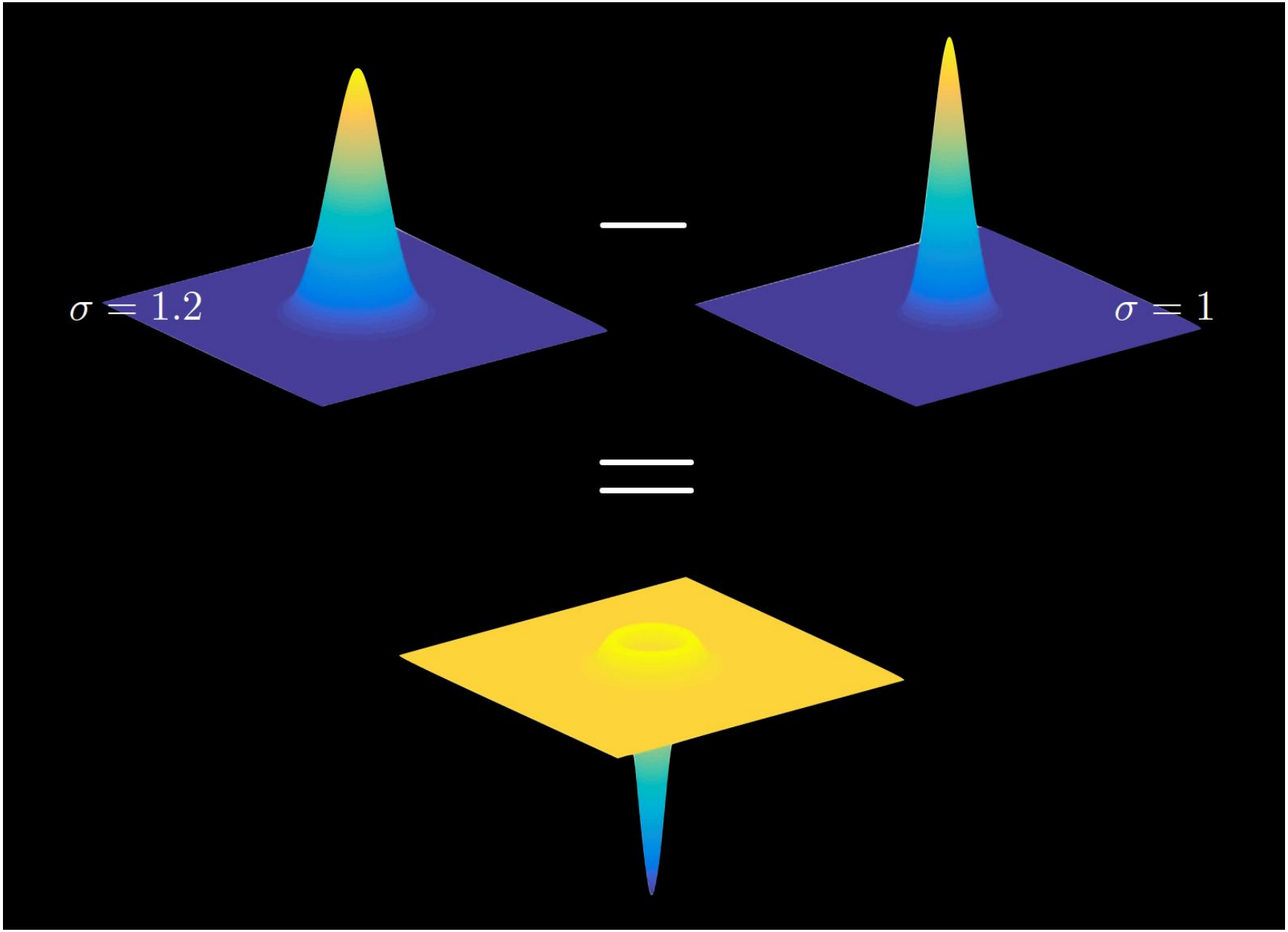


image

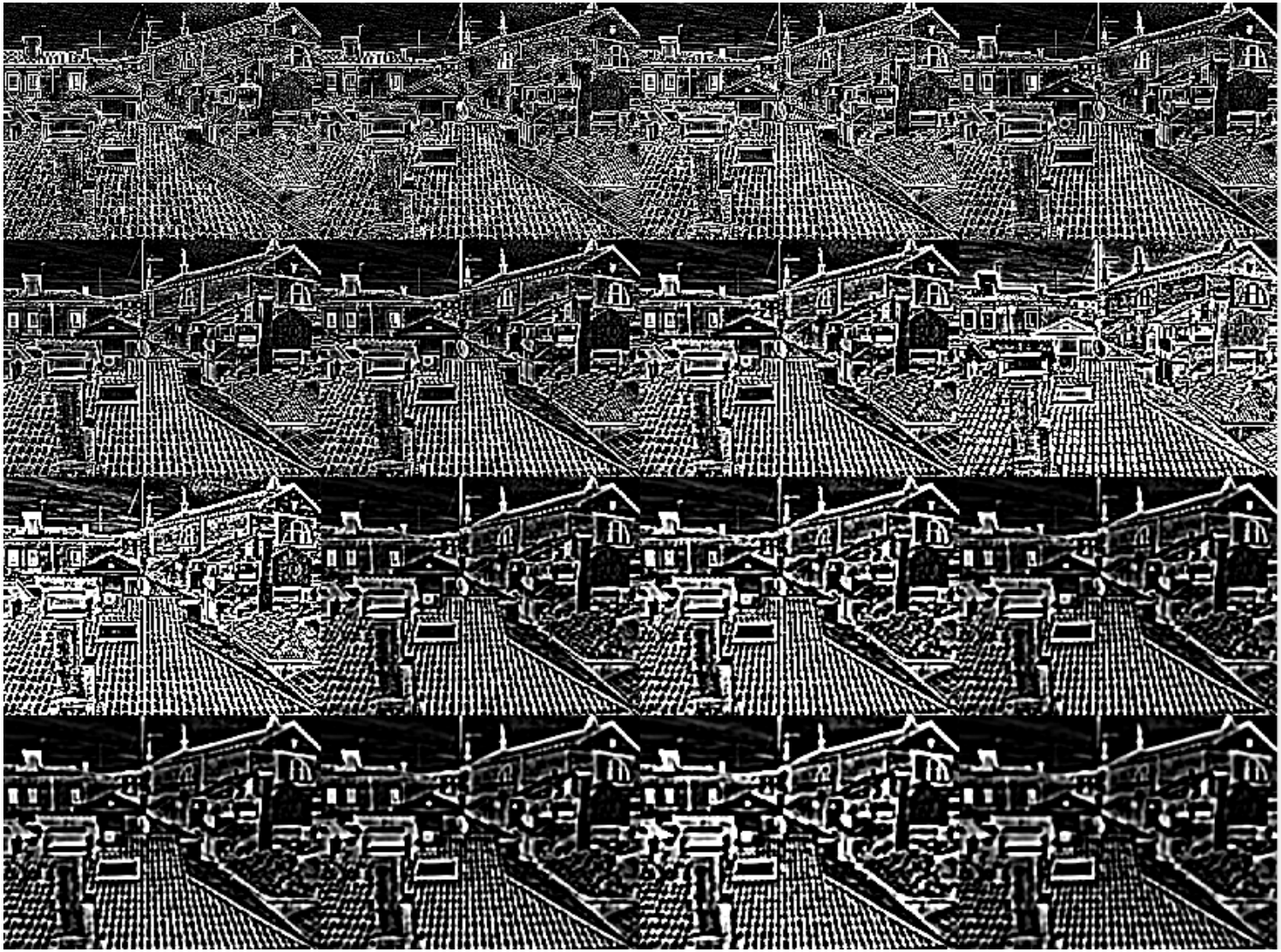


Slide from S. Lazebnik

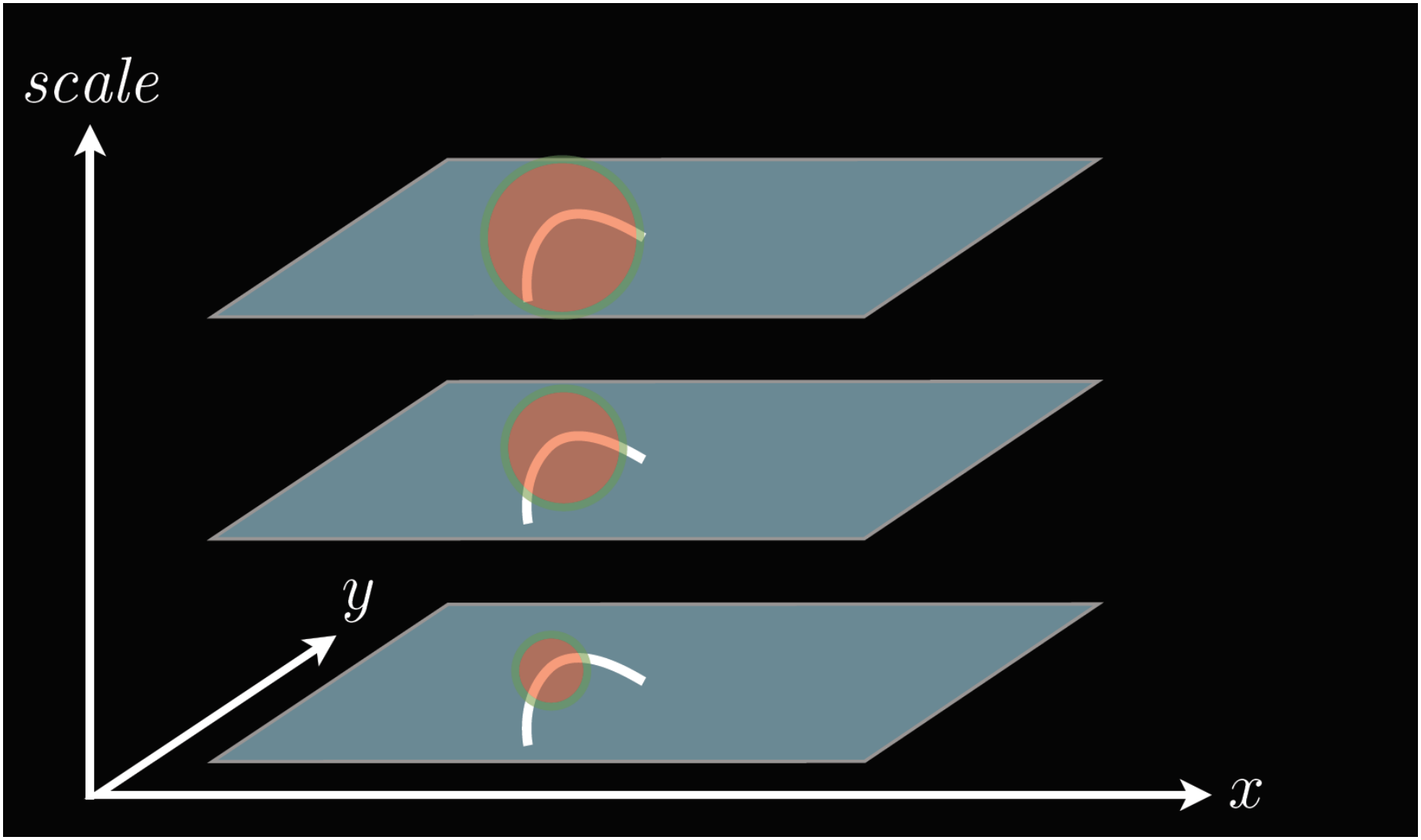
Difference of Gaussians (DoG)



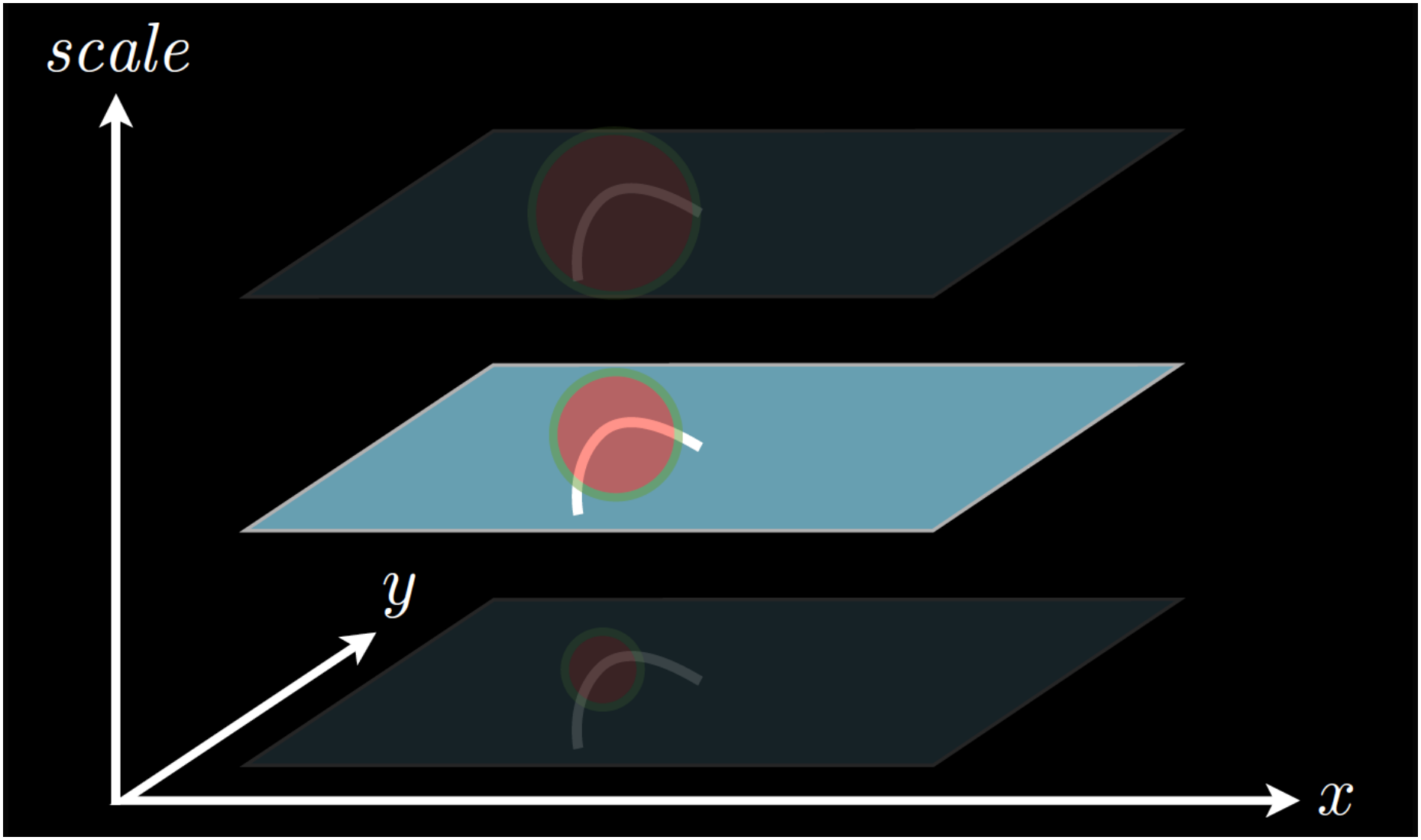
Laplacian Scale Space



We convolve DoG across space at different scales

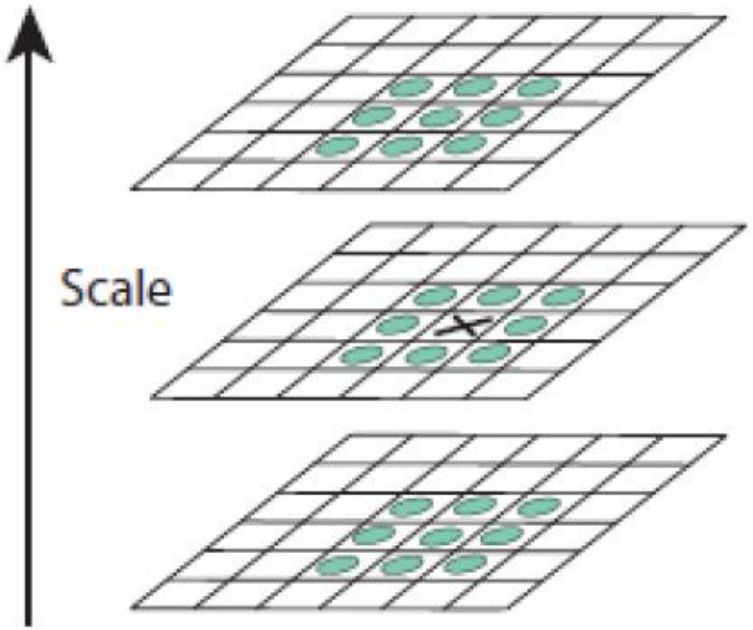


We convolve DoG across space at different scales and detect maximum



So, where is a SIFT keypoint?

- Definition of a keypoint: Maximum in the 3x3x3 (x,y,σ) region of the point.



Vedaldi's vlfeat

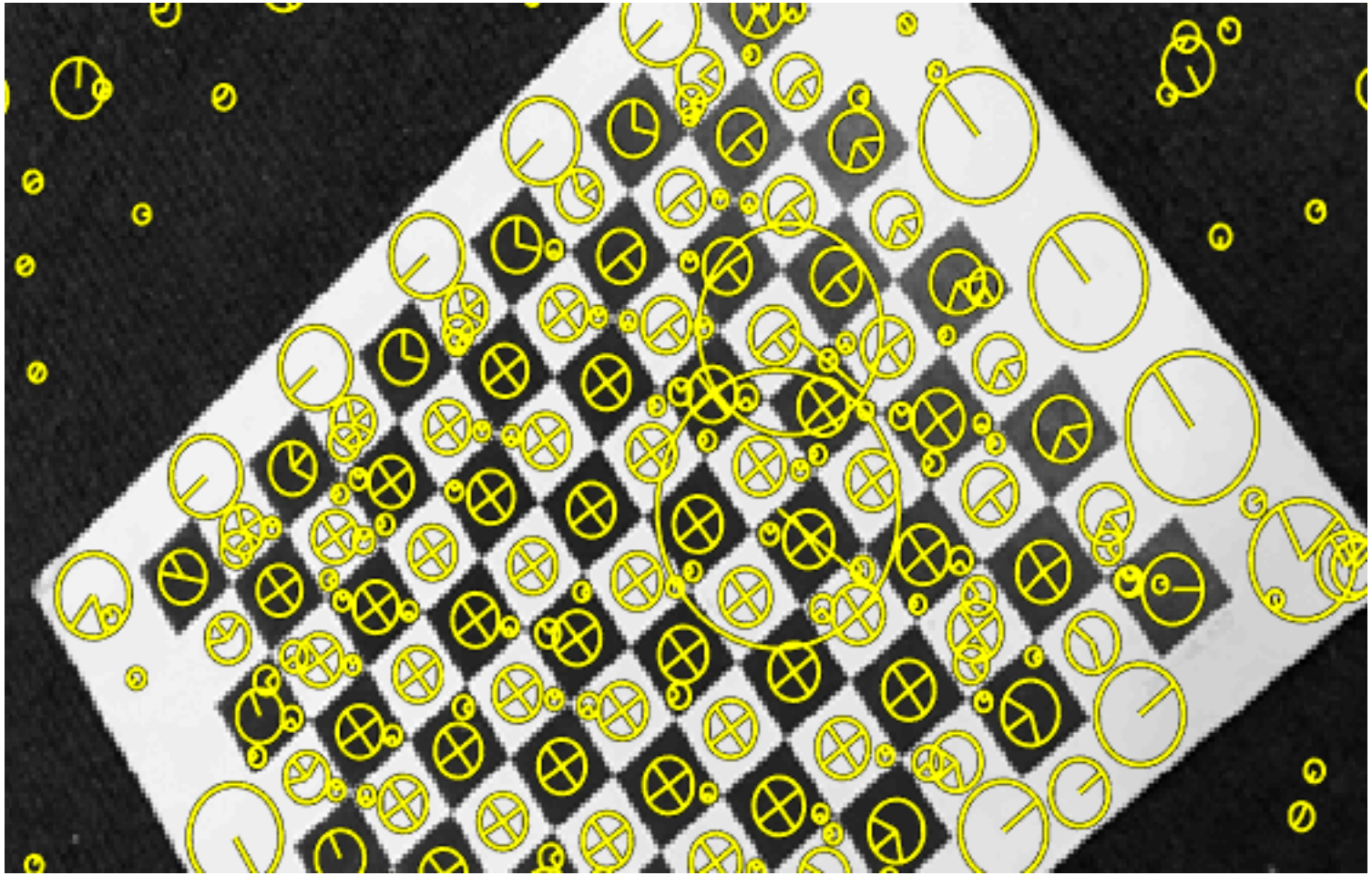
VLFeat.org

Home | Download | Tutorials | Applications | Documentation

Andrea Vedaldi, Ph.D. vedaldi@robots.ox.ac.uk
Associate Professor in Engineering Science
Information Engineering Building 30.05, Parks Road, Oxford, OX1 3PJ
Visual Geometry Group (directions)
Tel. +44 1865 273 127

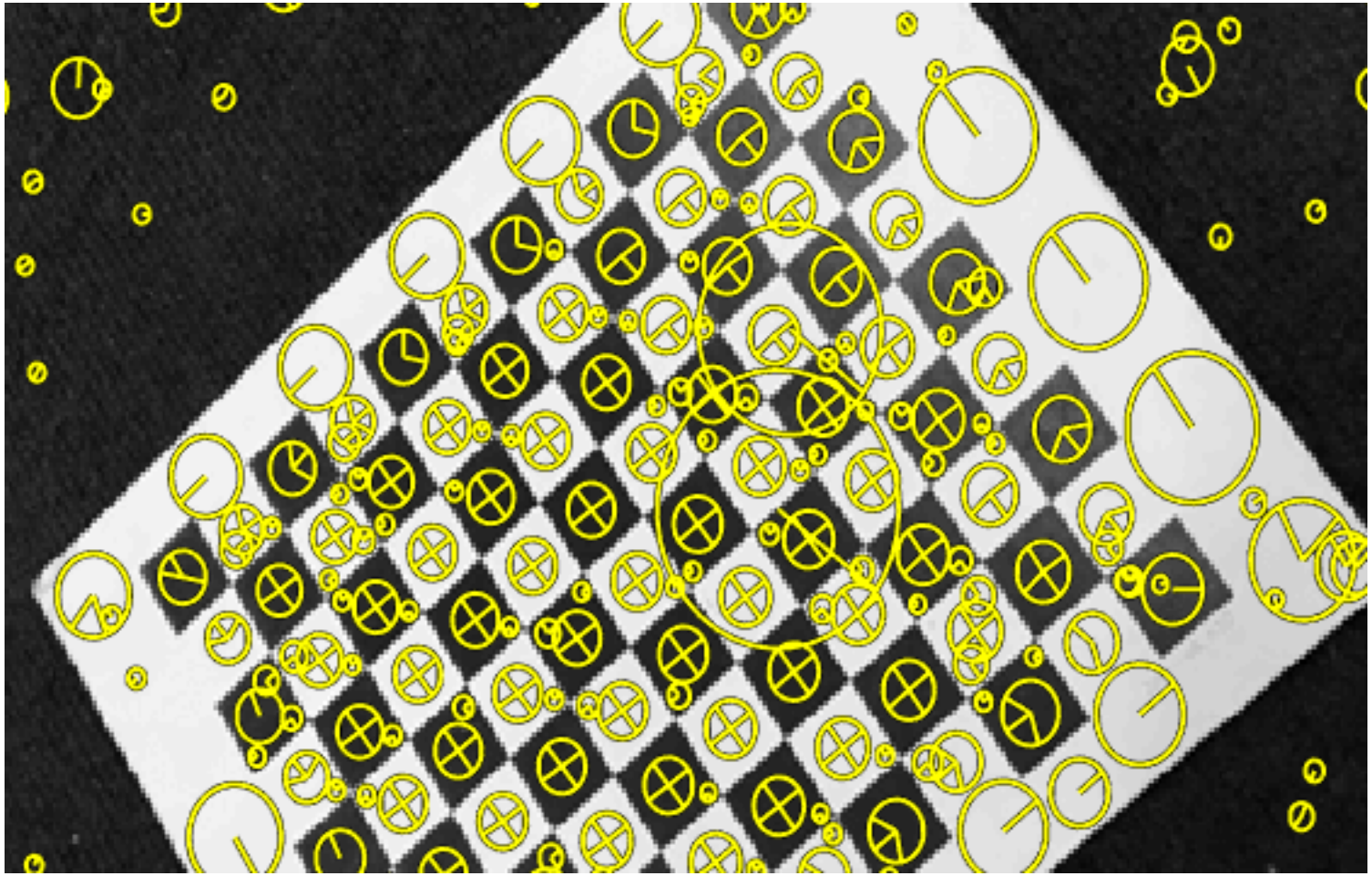
Résumé Google Scholar

Selected σ is visualized with a circle



Denoting the support region of the feature

Detector is rotation invariant



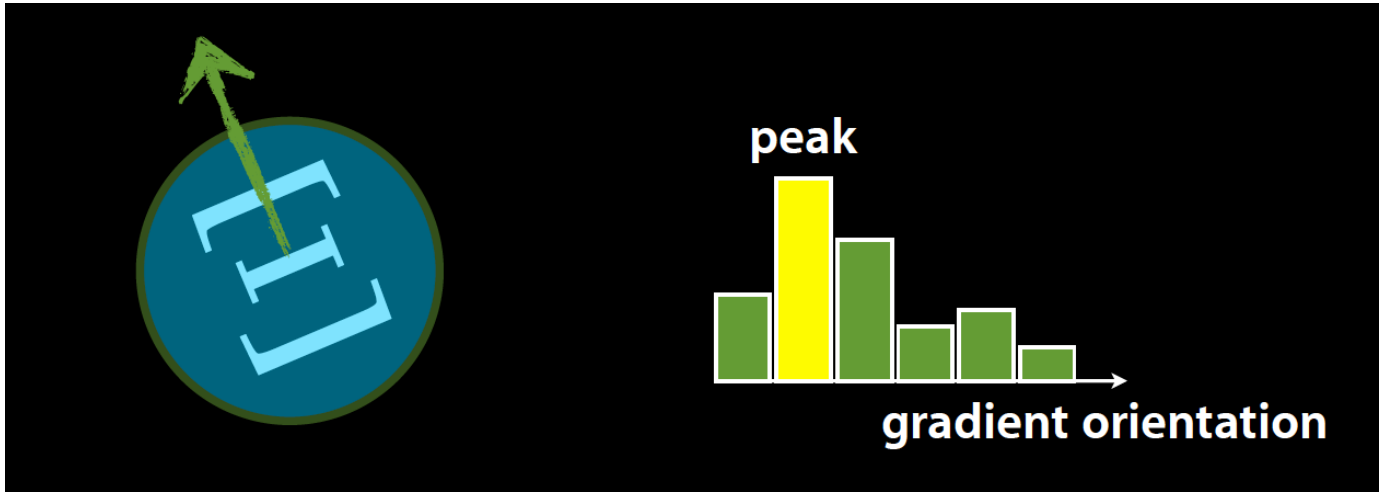
Because Laplacian is isotropic and a maximum in (x,y,σ) is invariant to rotations.

Descriptor invariance

- Since the intrinsic scale is detected (circle size) all circles will be normalized to a 16x16 region.

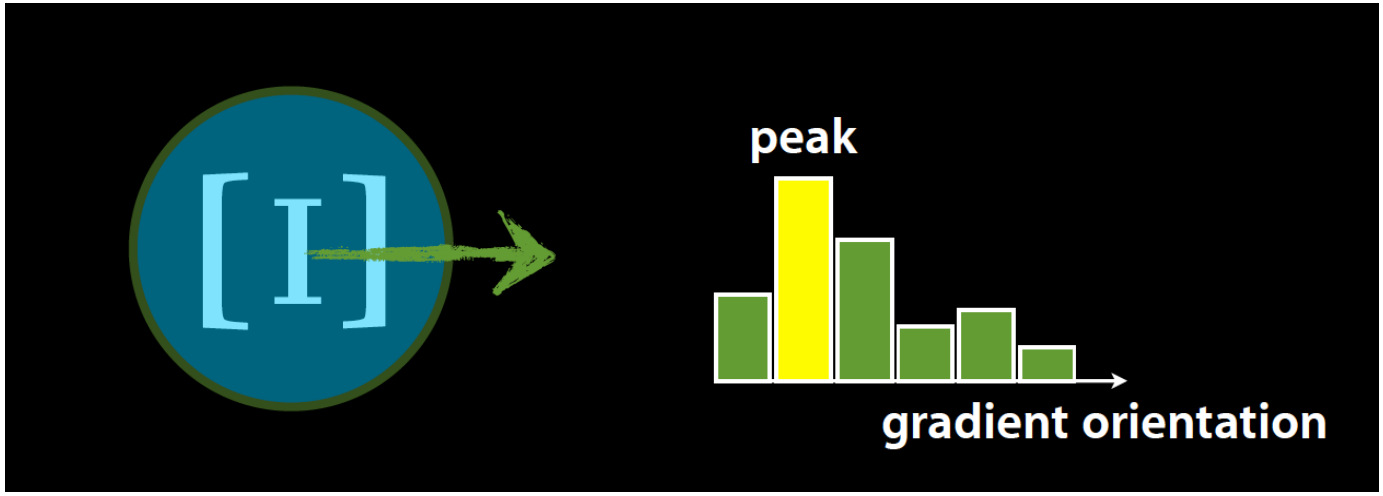
The descriptor should be also rotation invariant

- 1st Step: Find dominant orientation for the patch



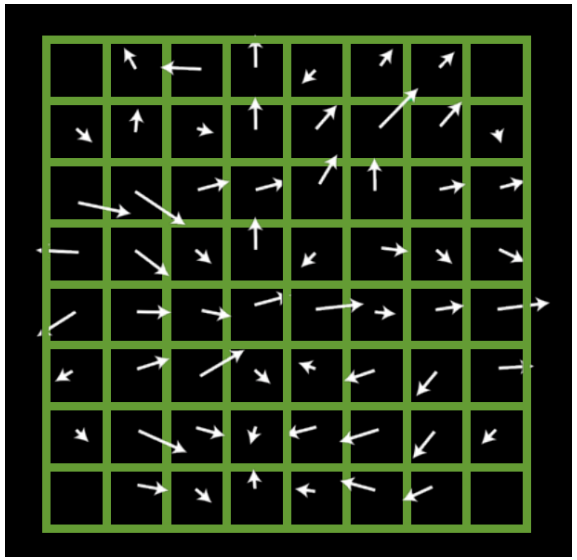
The descriptor should be rotation invariant

- 1st Step: Find dominant orientation for the patch
- 2nd Step: Rotate patch to point along x-axis



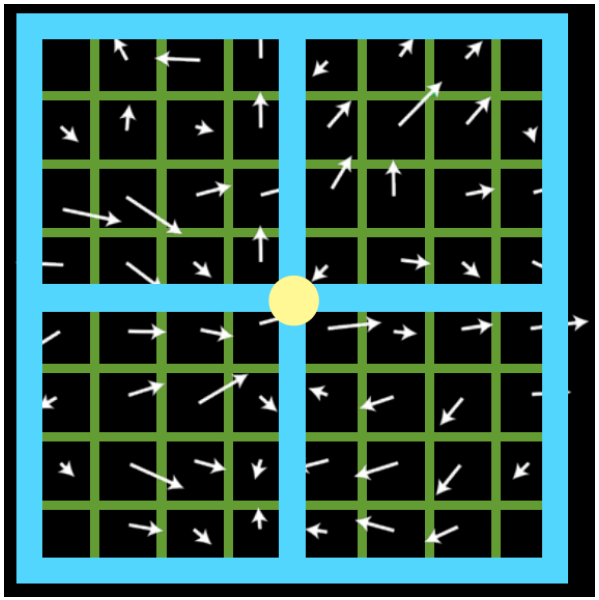
To extract a feature descriptor from a cell

- Compute Image Gradients



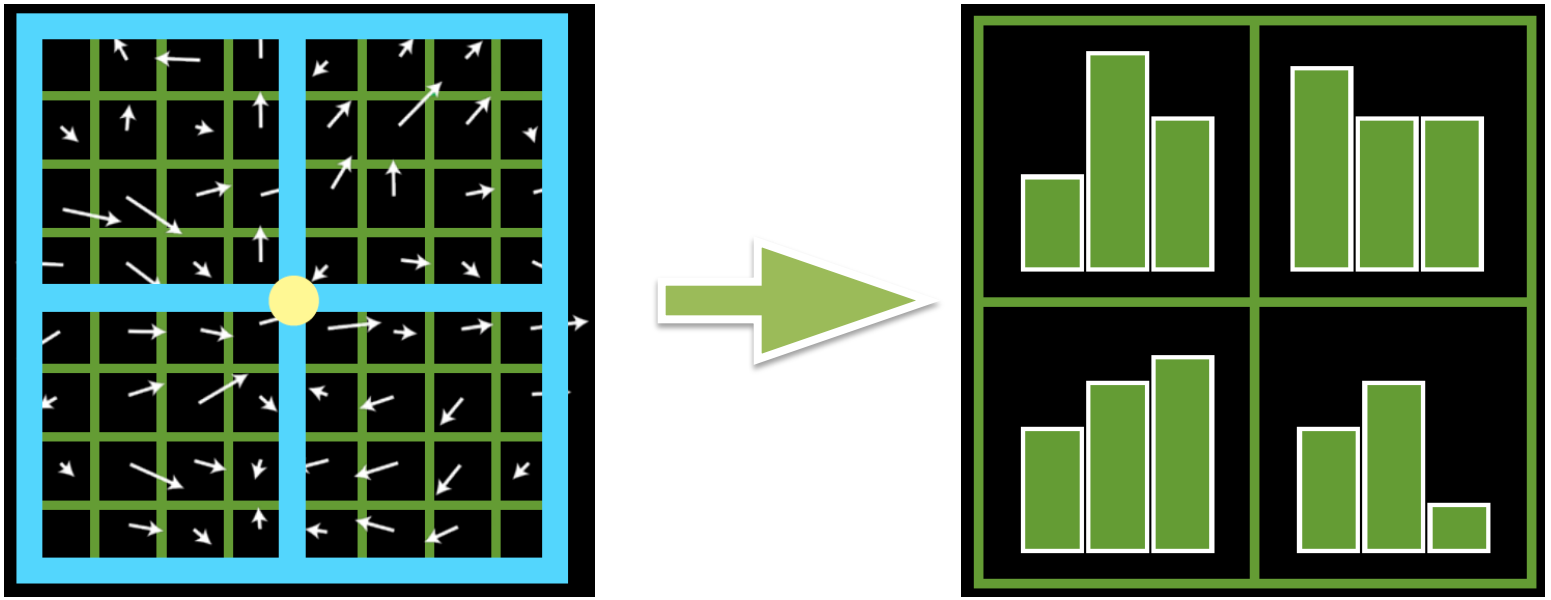
To extract a feature descriptor from a cell

- Compute Image Gradients
- Accumulate gradients along cells

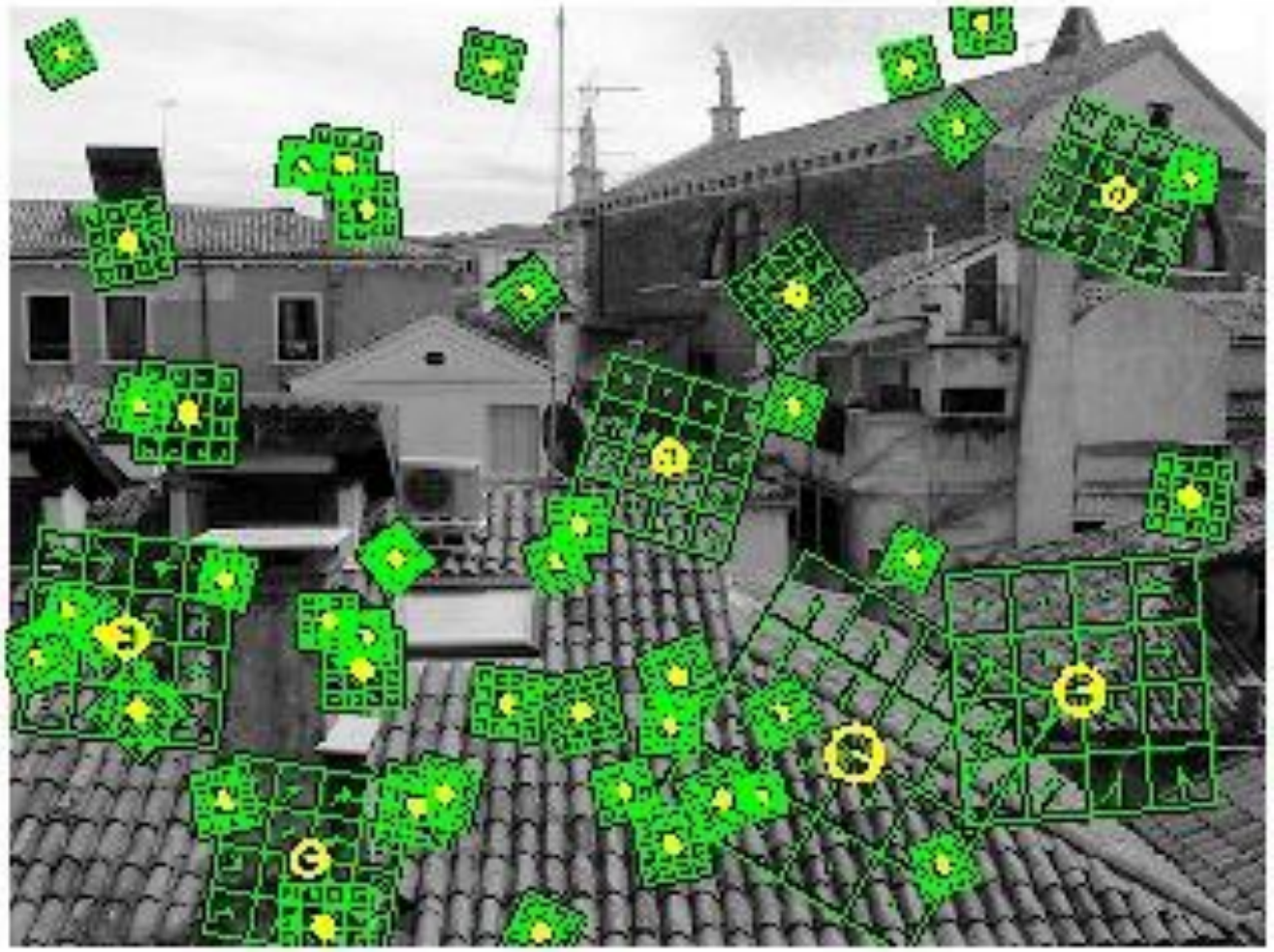


To extract a feature descriptor from a cell

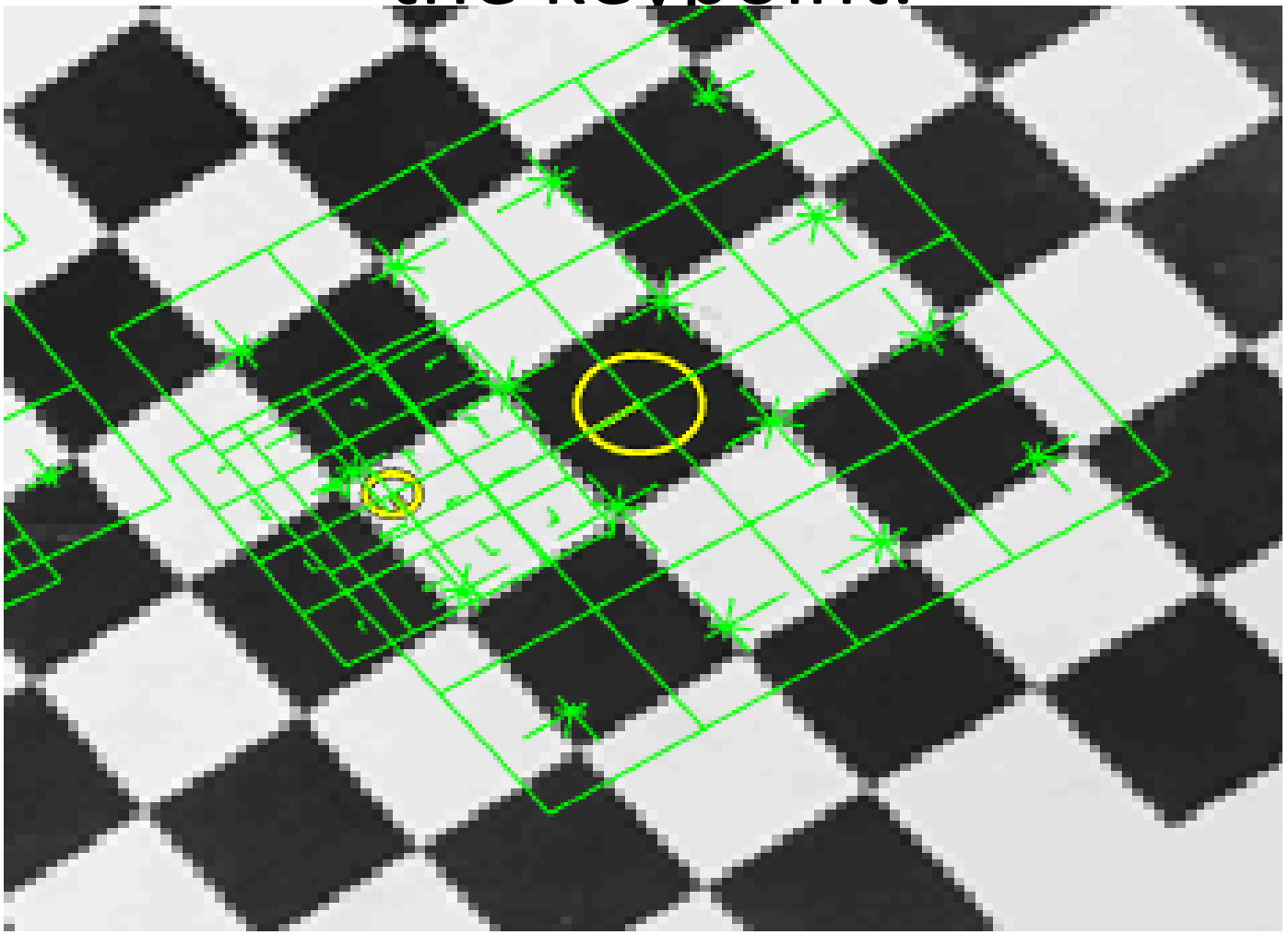
- Compute Image Gradients
- Accumulate gradients along cells
- Form image descriptor



As a matter of fact it is a 4x4 grid of histograms at each keypoint

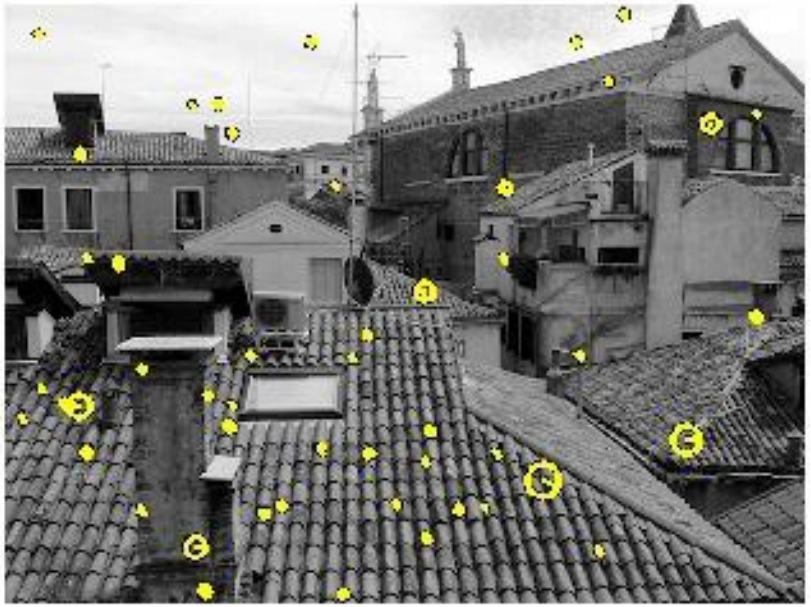


The descriptor is an 128x1 vector which together with σ,θ characterize the keypoint.



Example of SIFT detections and feature extraction

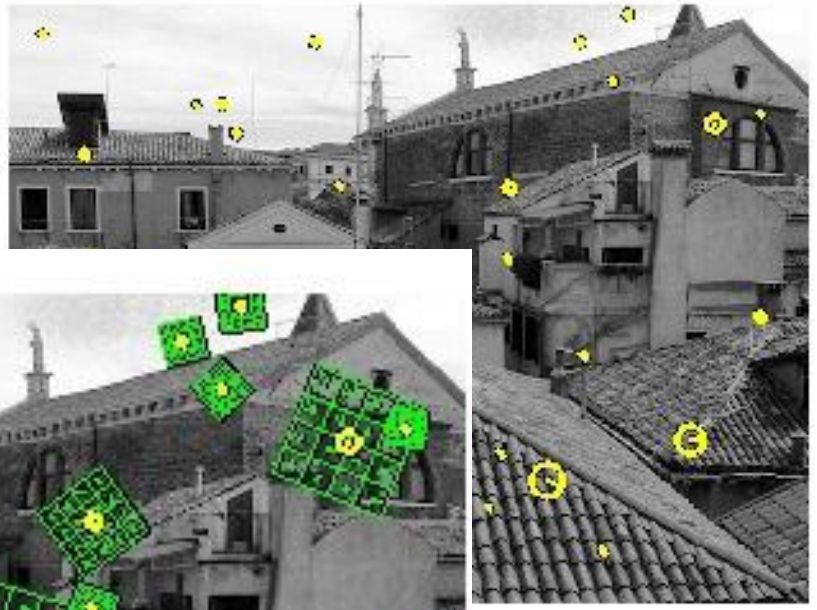
Input Image



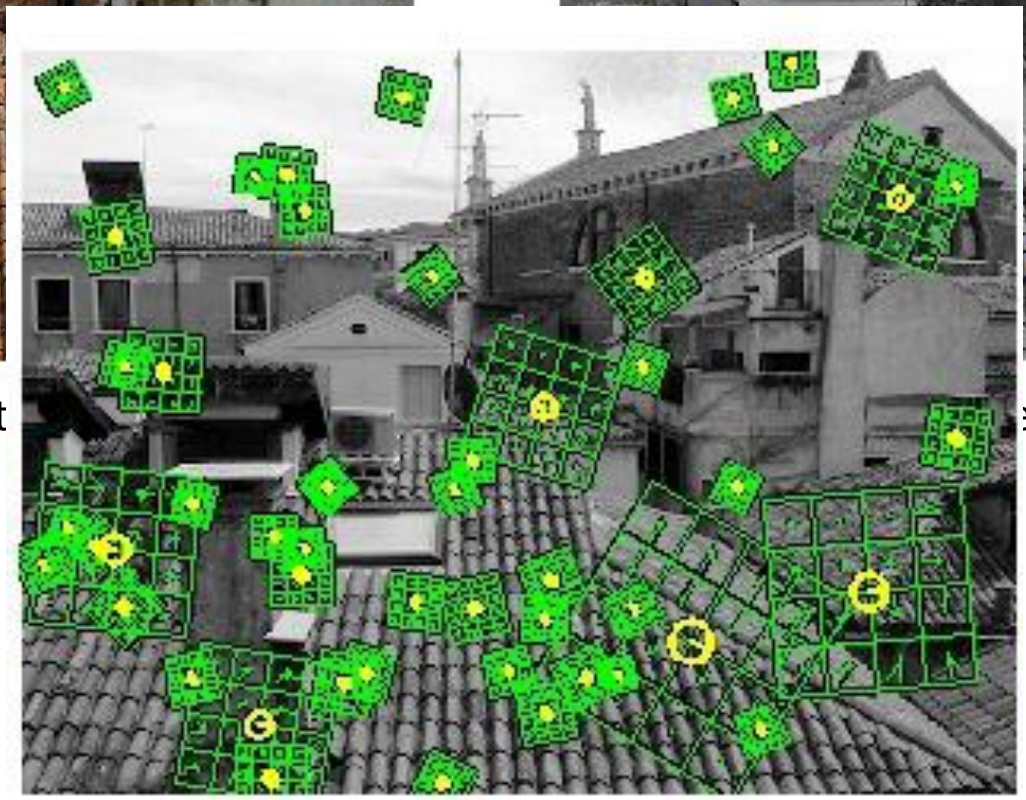
Example Detections

Example of SIFT detections and feature extraction

Input



ections



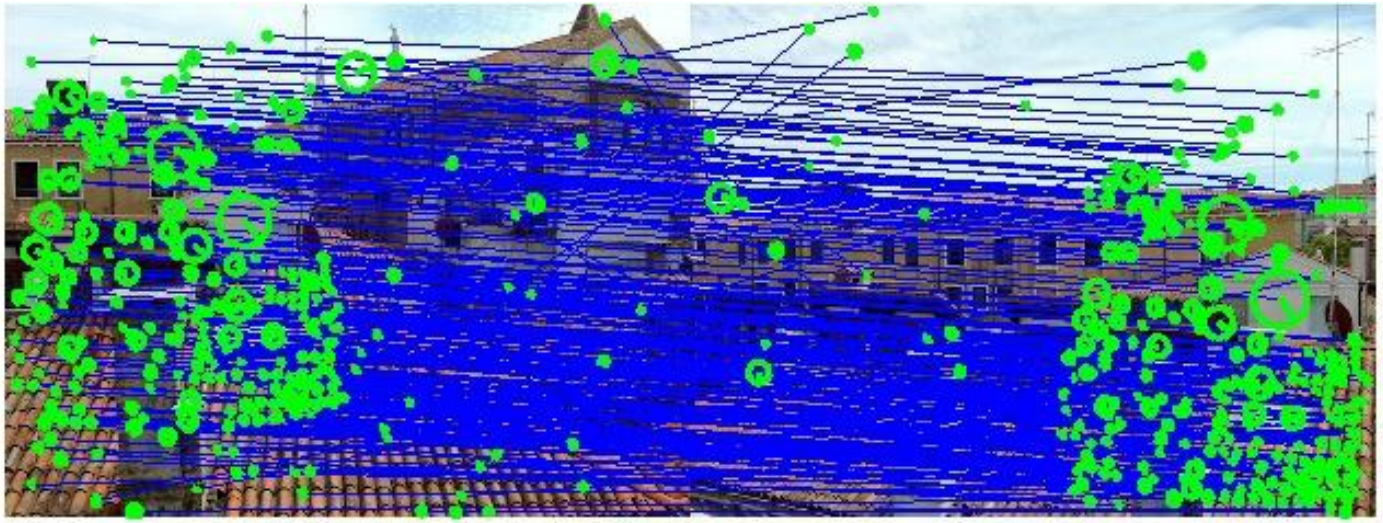
Extracted Feature Descriptors

Using SIFT for image matching

Original Image Pair

Using SIFT for image matching

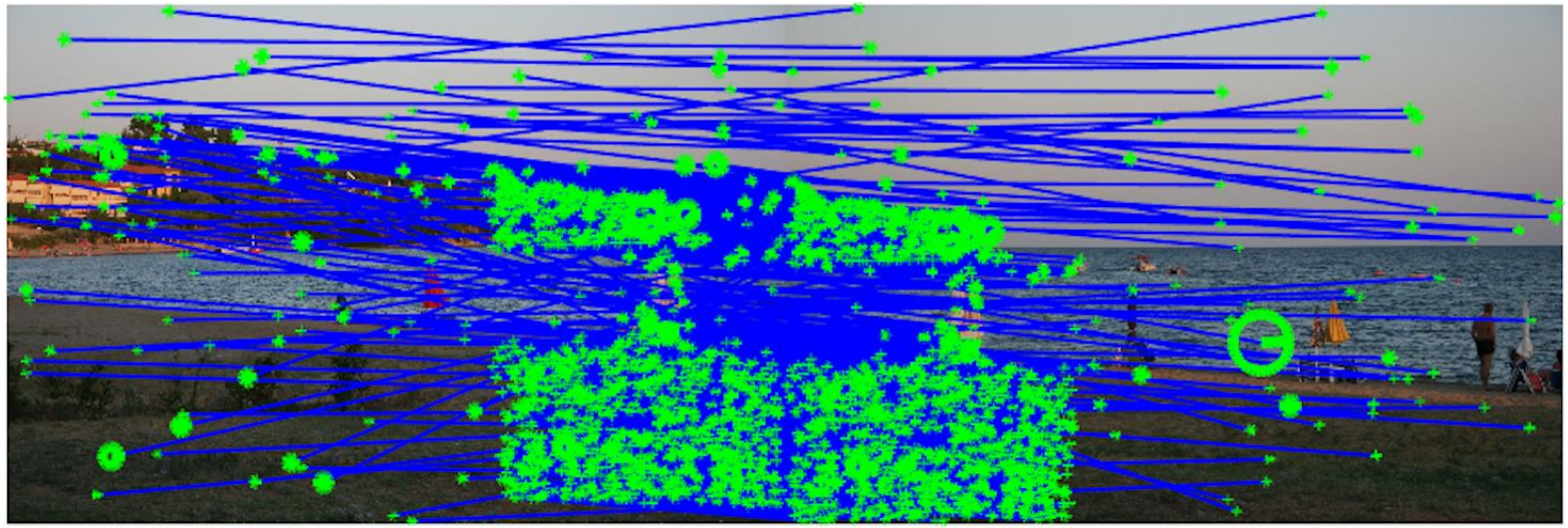
Original Image Pair



Matched features

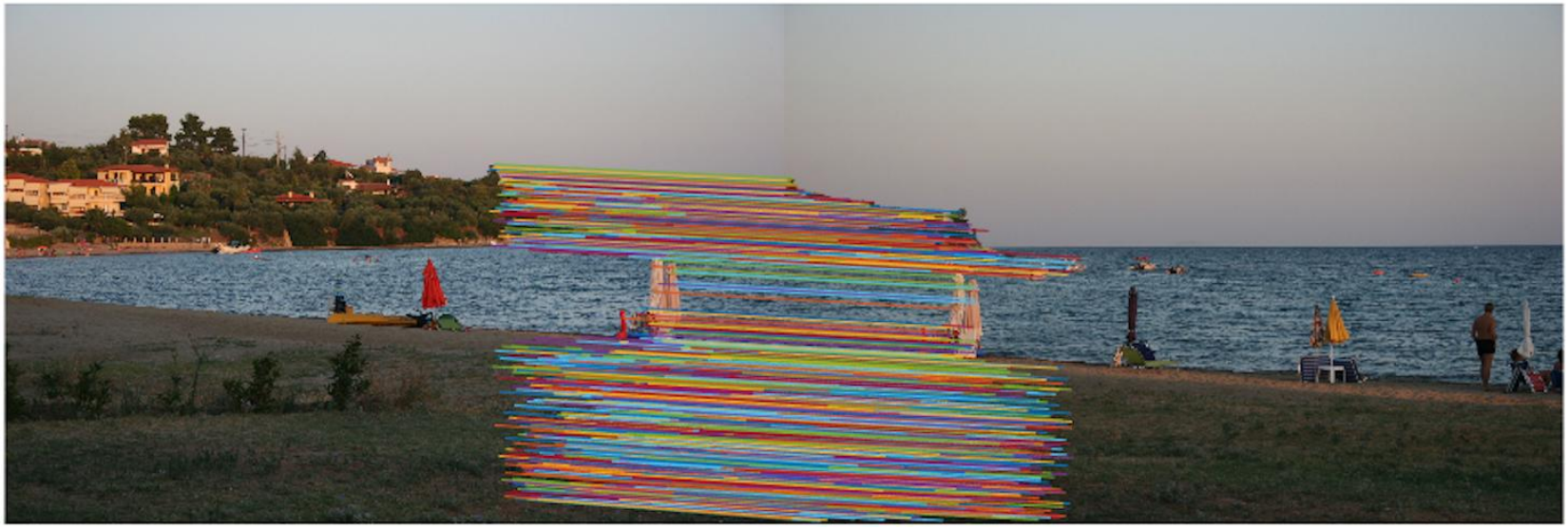
Create Image Mosaic

1. Get an image pair



2. Establish correspondences between matching features

Create Image Mosaic



3. Keep only consistent matches (inliers)

4. Compute homography and warp 2nd image

Create Image Mosaic

5. Repeat to extend the mosaic

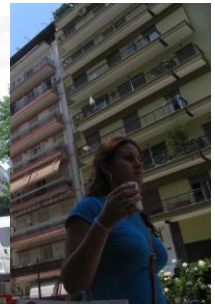
Find Location

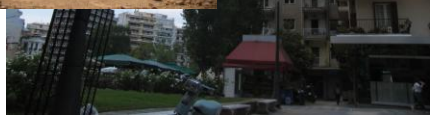
Query Image

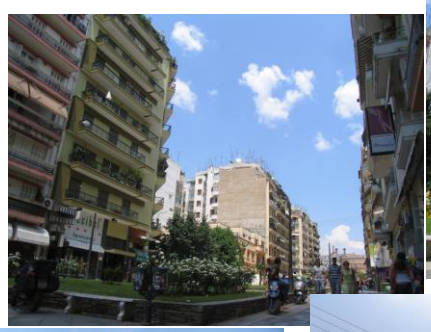
We want to find a match in a dataset of given images

Find Location

Query Image







Find Location

Query Image

Good Matches

Medium Matches

Bad Matches

Find Location

Good Match

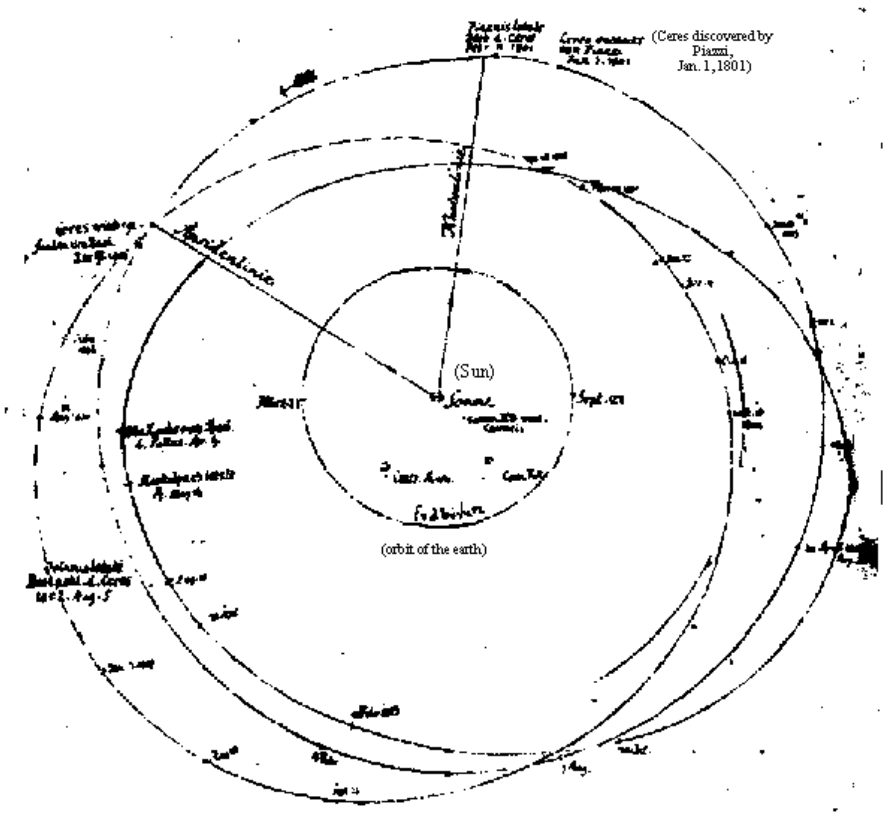
Medium Match

Bad Match

SIFT Features

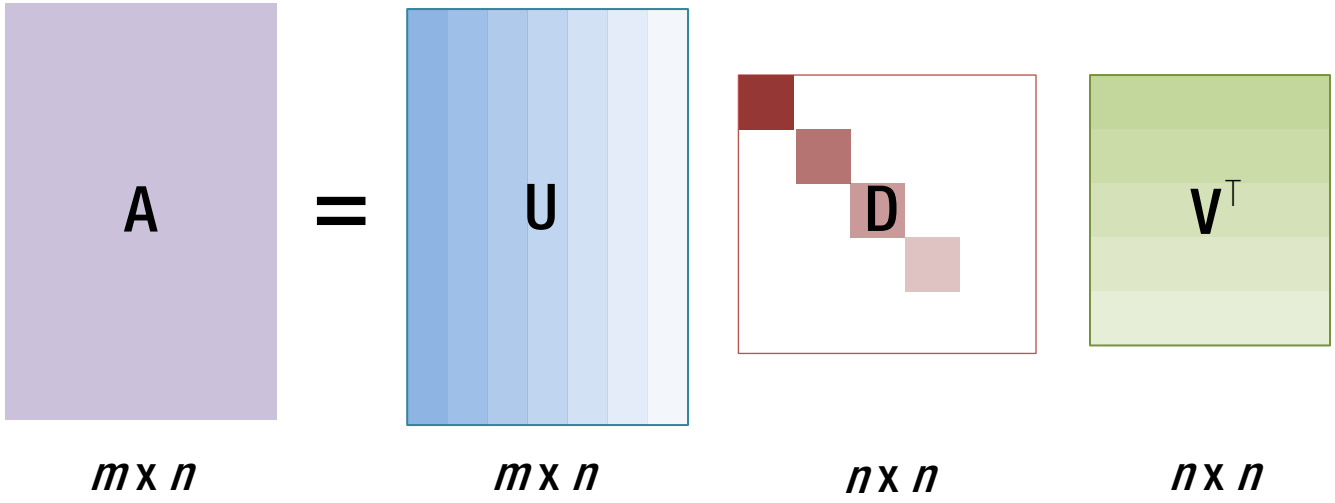
- SIFT detector can automatically
  - Select scale
  - Compute dominant rotation
- SIFT descriptor
  - Is a grid of histogram of gradient orientations
  - On a region normalized with respect to scale and rotation

1809, Carl Friedrich Gauss

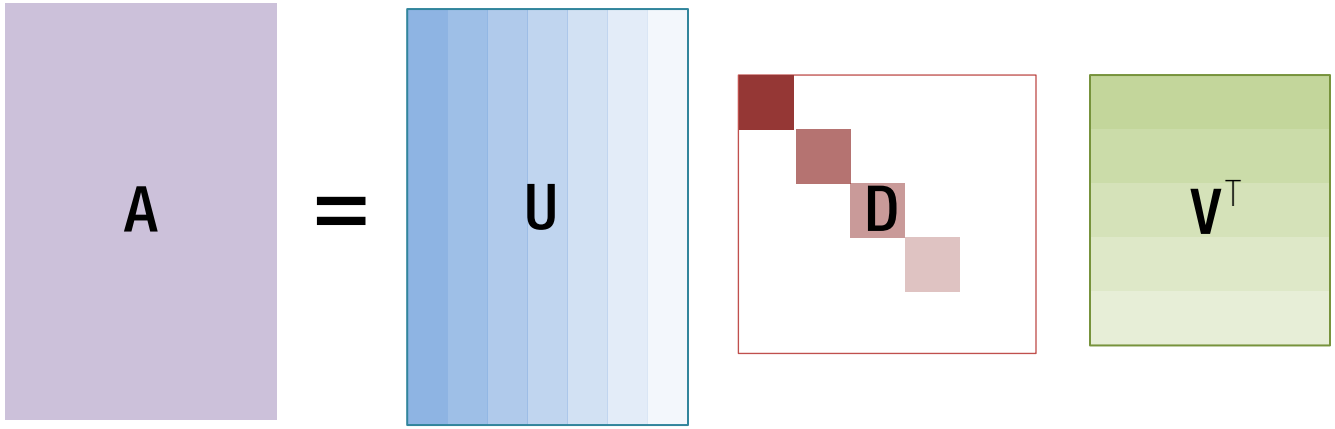


Sketch of the orbits of Ceres and Pallas (nachlaß Gauß, Handb. 4). Courtesy of Universitätsbibliothek Göttingen.

Singular Value Decomposition



Singular Value Decomposition



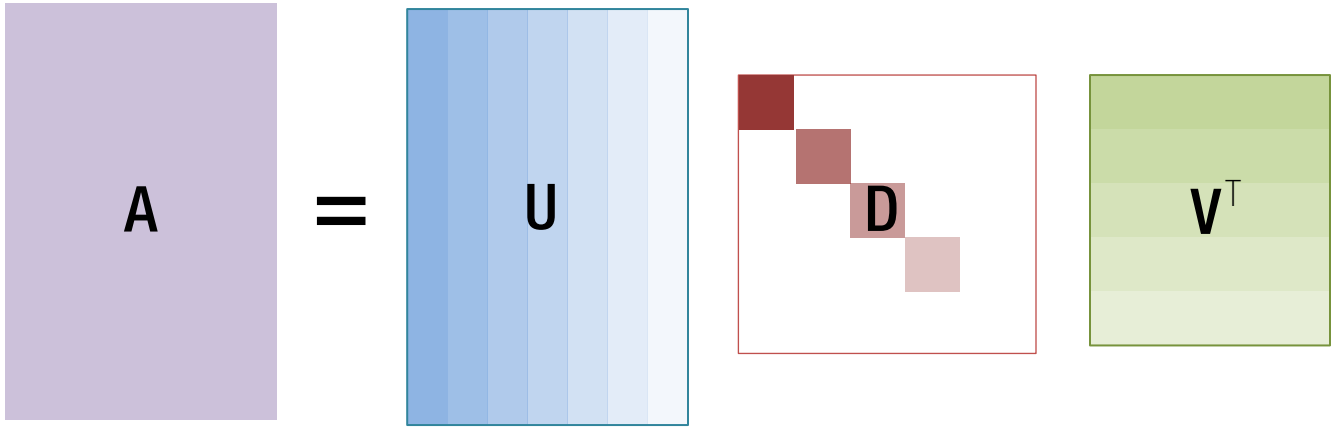
$m \times n$ $m \times n$ $n \times n$ $n \times n$

Column orthogonal matrix

$$\mathbf{U}_i^T \mathbf{U}_i = \|\mathbf{U}_i\| = 1$$

$$\mathbf{U}_j^T \mathbf{U}_i = \mathbf{U}_j^T \mathbf{U}_i = 0 \quad \text{for} \quad i \neq j$$

Singular Value Decomposition



$m \times n$ $\quad m \times n$ $\quad n \times n$ $\quad n \times n$

Column orthogonal matrix

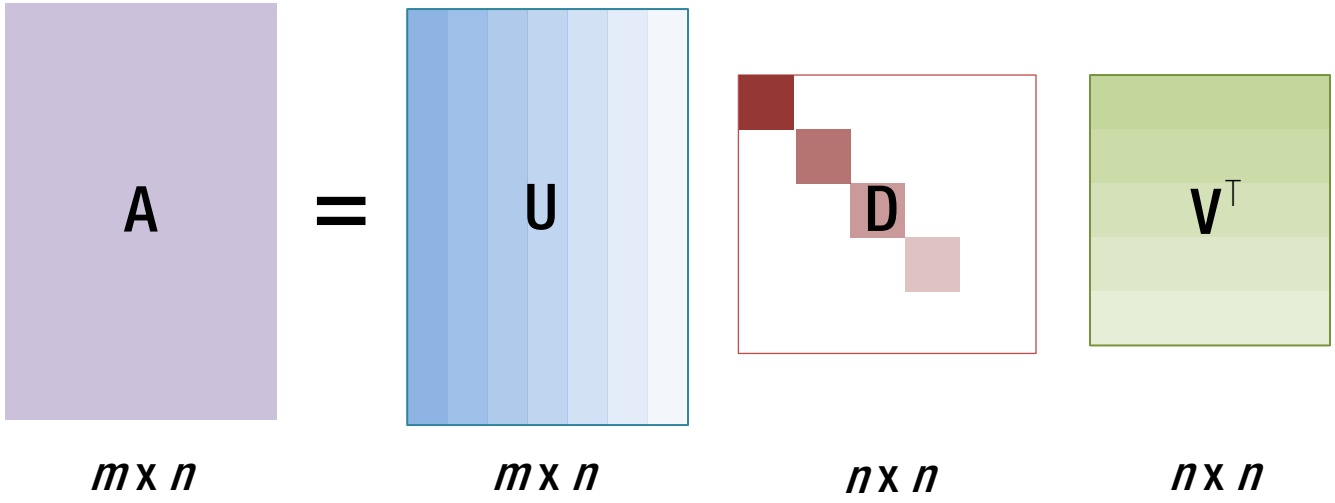
$$\mathbf{U}_i^T \mathbf{U}_i = \|\mathbf{U}_i\| = 1$$

$$\mathbf{U}_j^T \mathbf{U}_i = \mathbf{U}_j^T \mathbf{U}_i = 0 \quad \text{for} \quad i \neq j$$

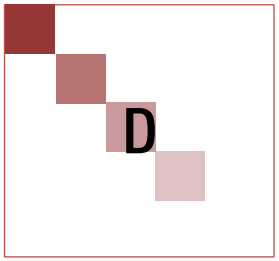
$$\mathbf{U}^T \mathbf{U} = \mathbf{I}_{m \times m}$$

$$\mathbf{V}^T \mathbf{V} = \mathbf{I}_{n \times n}$$

Singular Value Decomposition

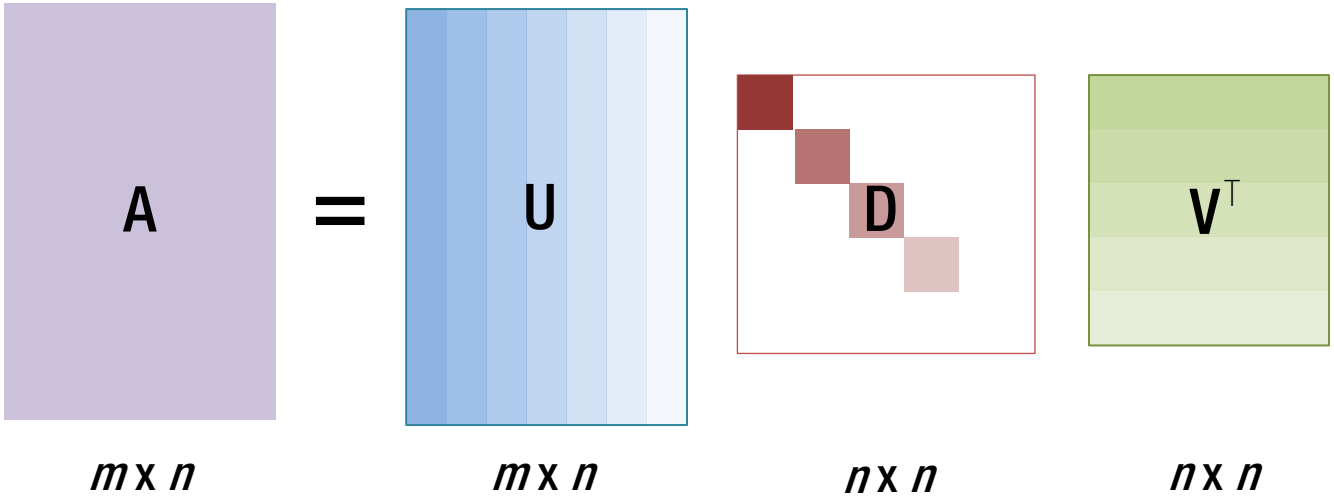


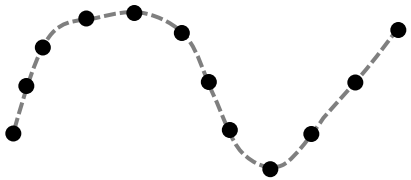
Singular value matrix

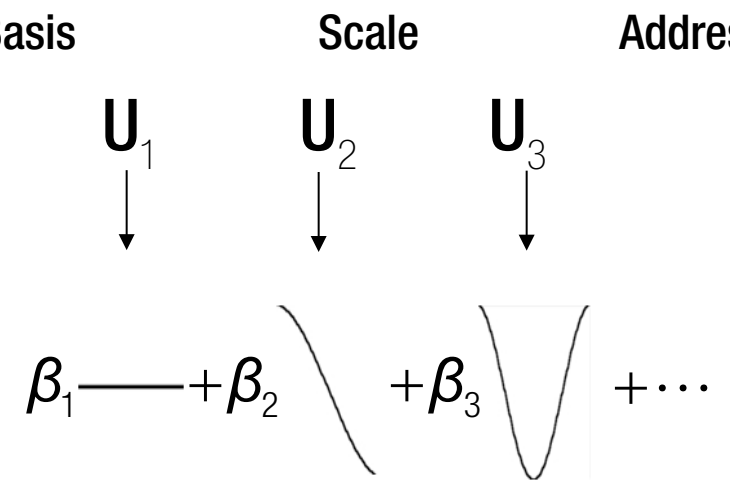


$= \mathrm{diag}\{\sigma_1, \sigma_2, \cdots, \sigma_n\}$ where $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_n \geq 0$

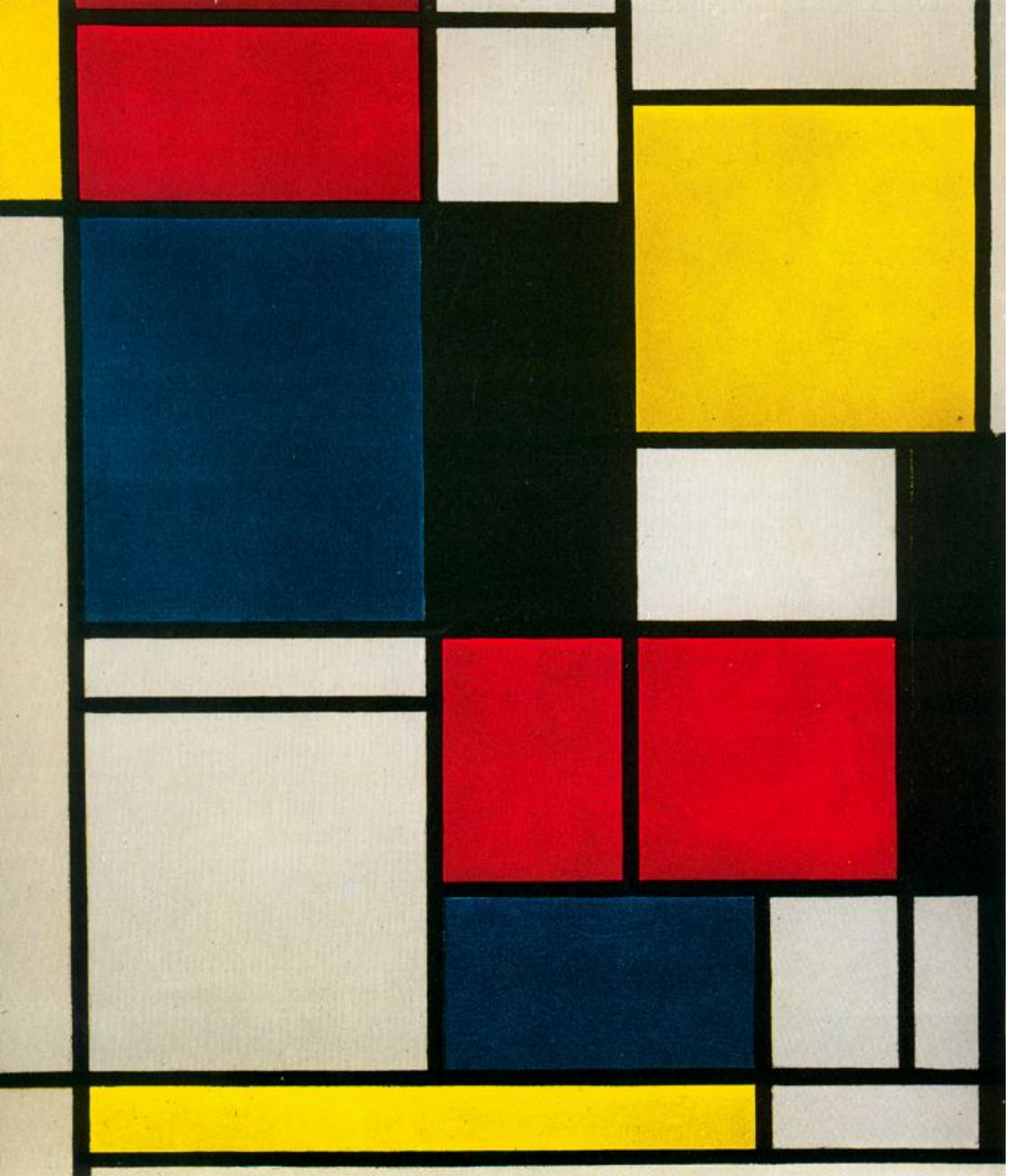
Singular Value Decomposition



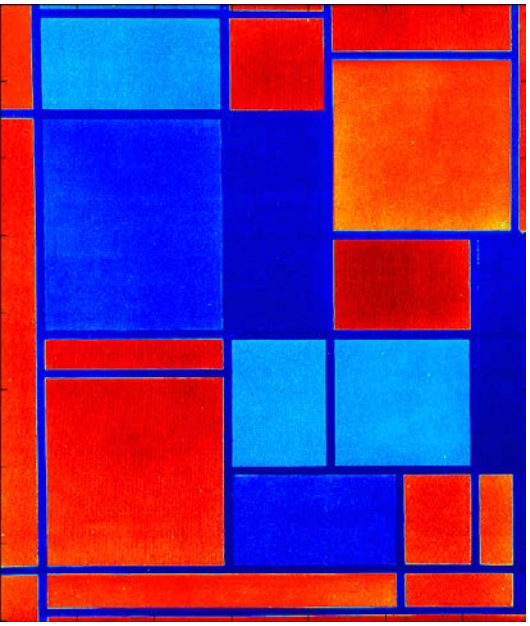




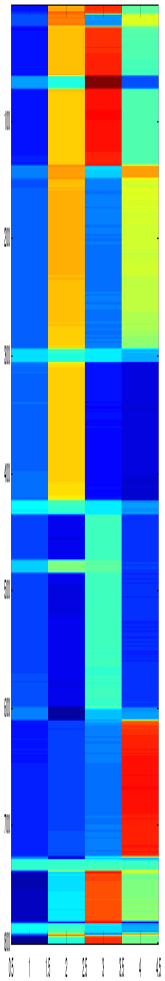
SVD as basis + transformed Address



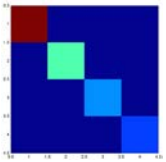
SVD of this?



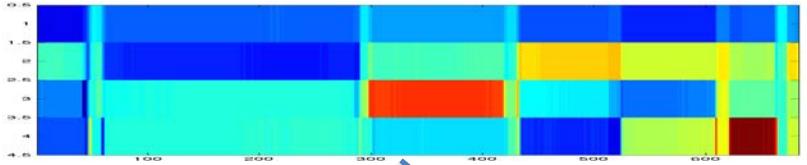
U(:,1:4)

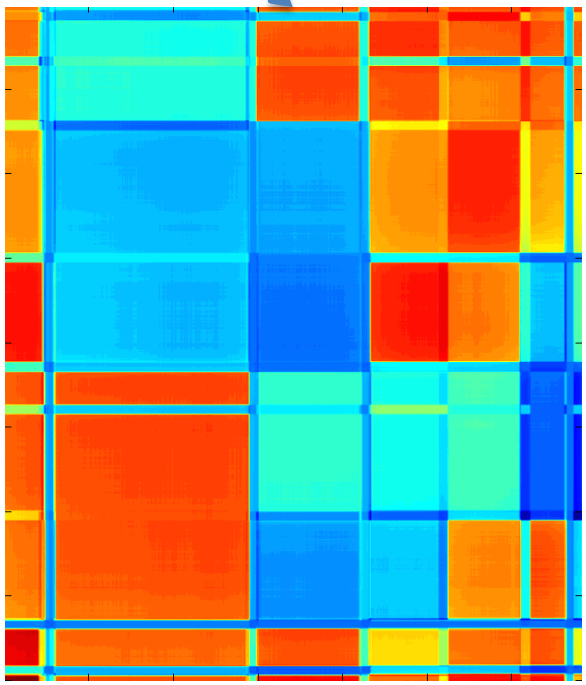


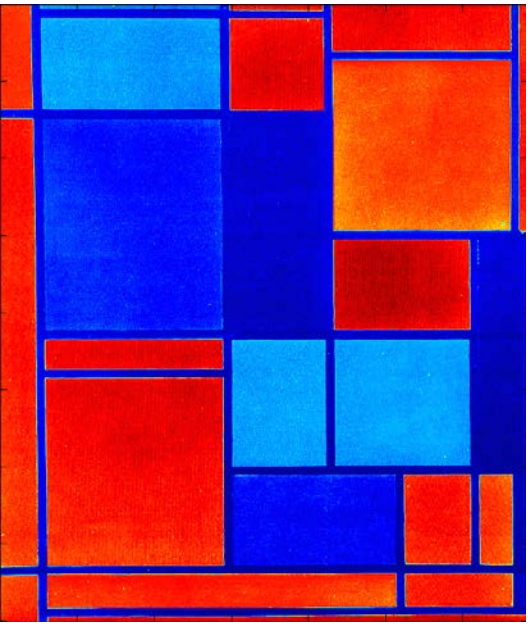
D(1:4,1:4)

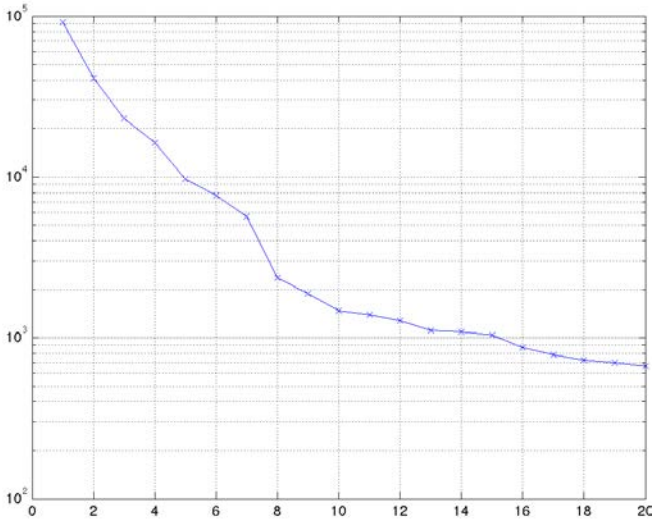


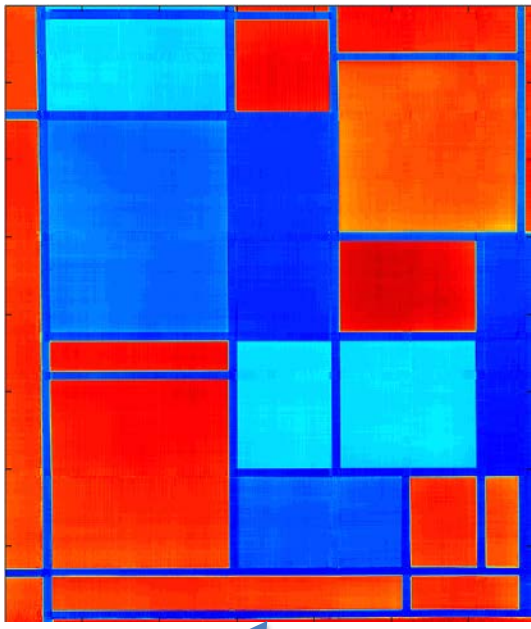
V(:,1:4)'







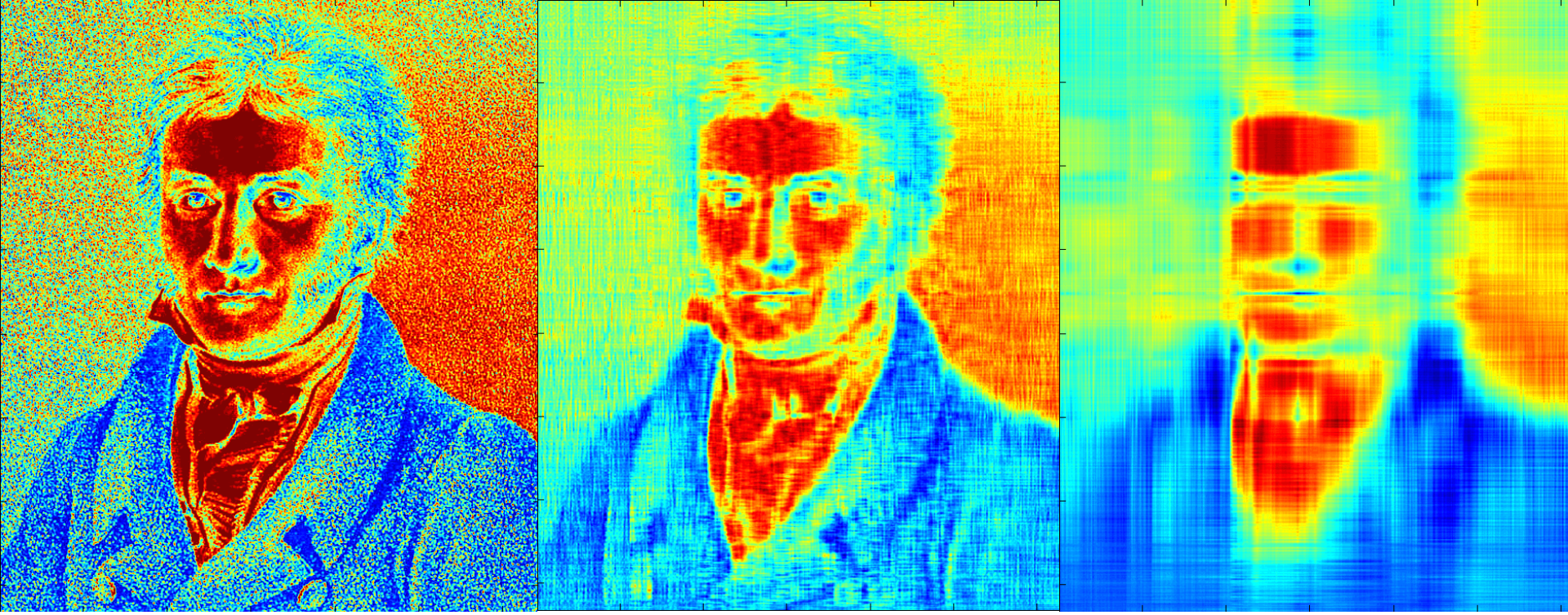


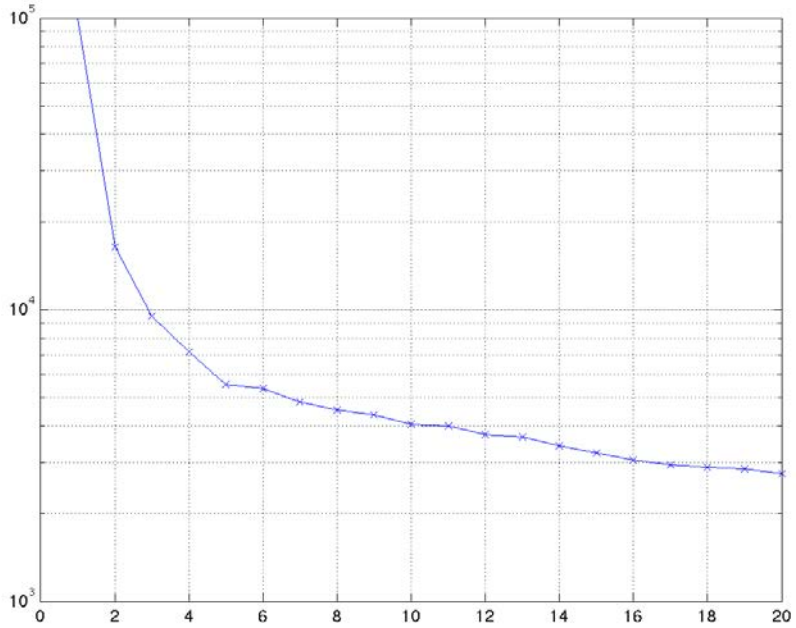


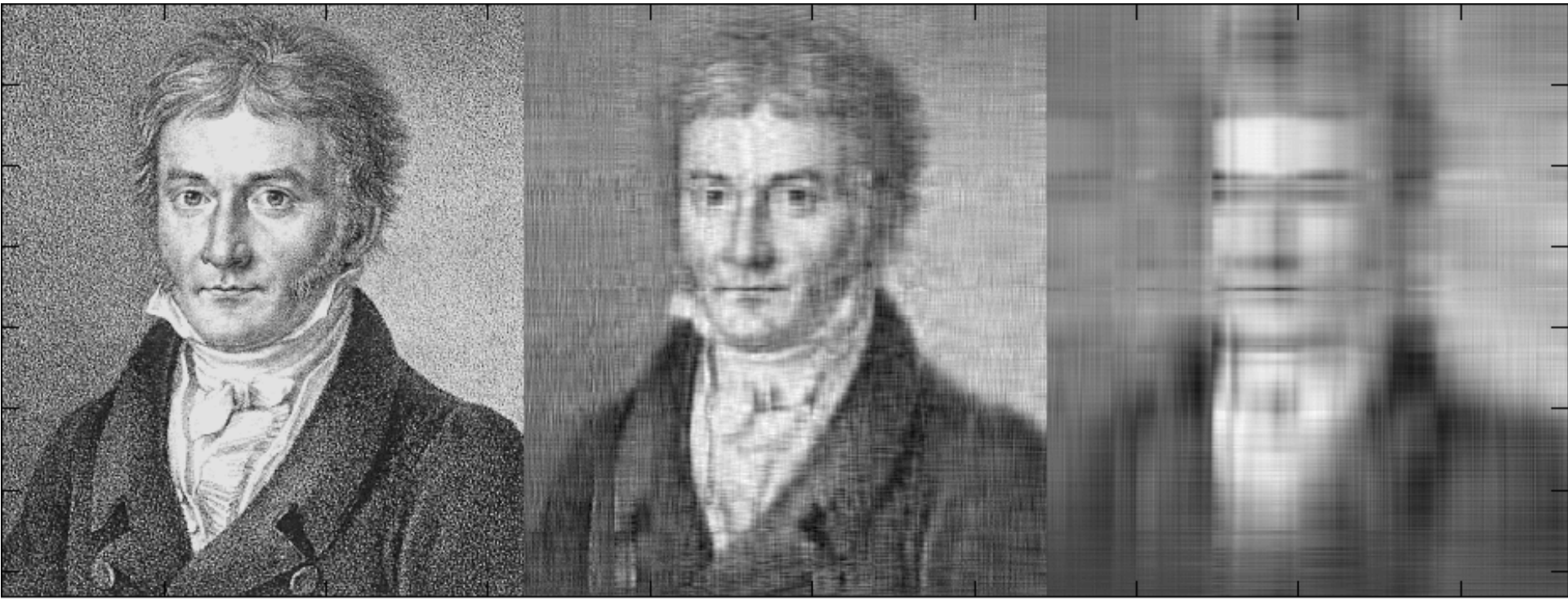
[u,d,v] = svd(I);

semilogy(diag(d(1:20,1:20)),'x-')

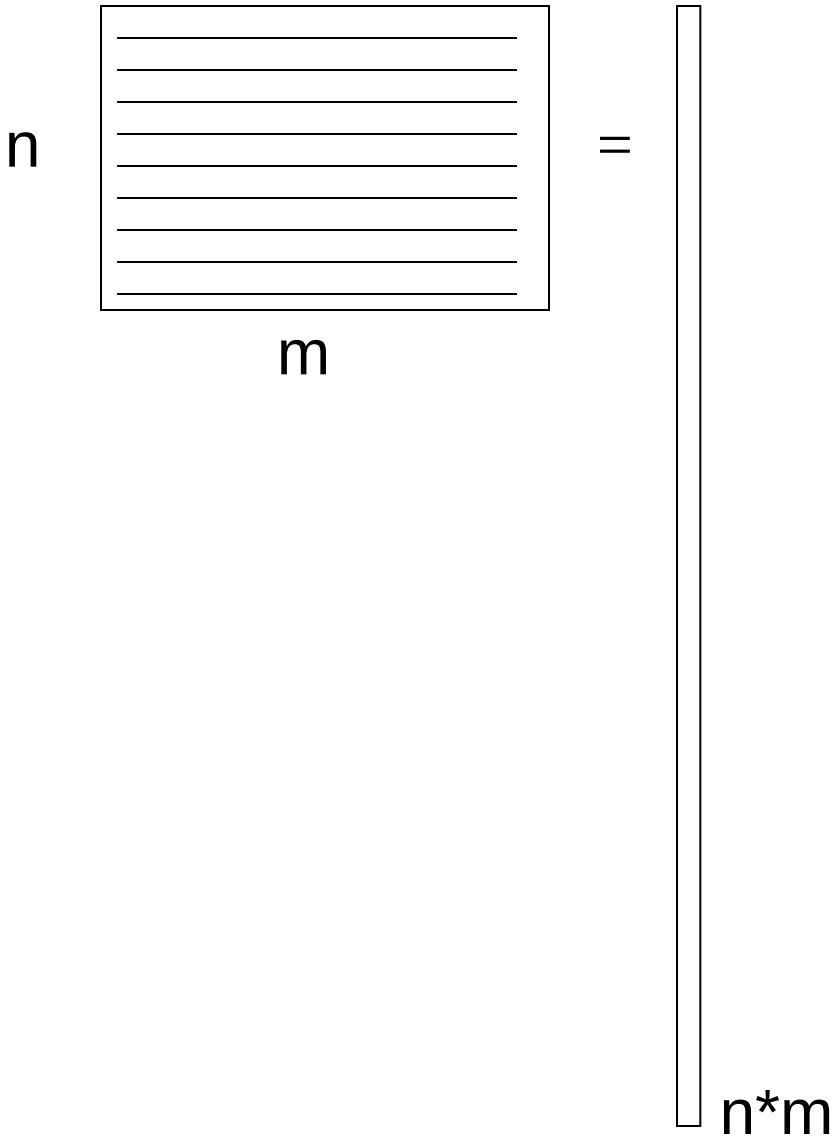
Im2 = u(:,1:20)*d(1:20,1:20)*v(:,1:20)';




10^5
10^4
10^3
0
2
4
6
8
10
12
14
16
18
20



Images as Vectors



Vector Mean

n

m

=

I1

+

I2

n*m

n*m

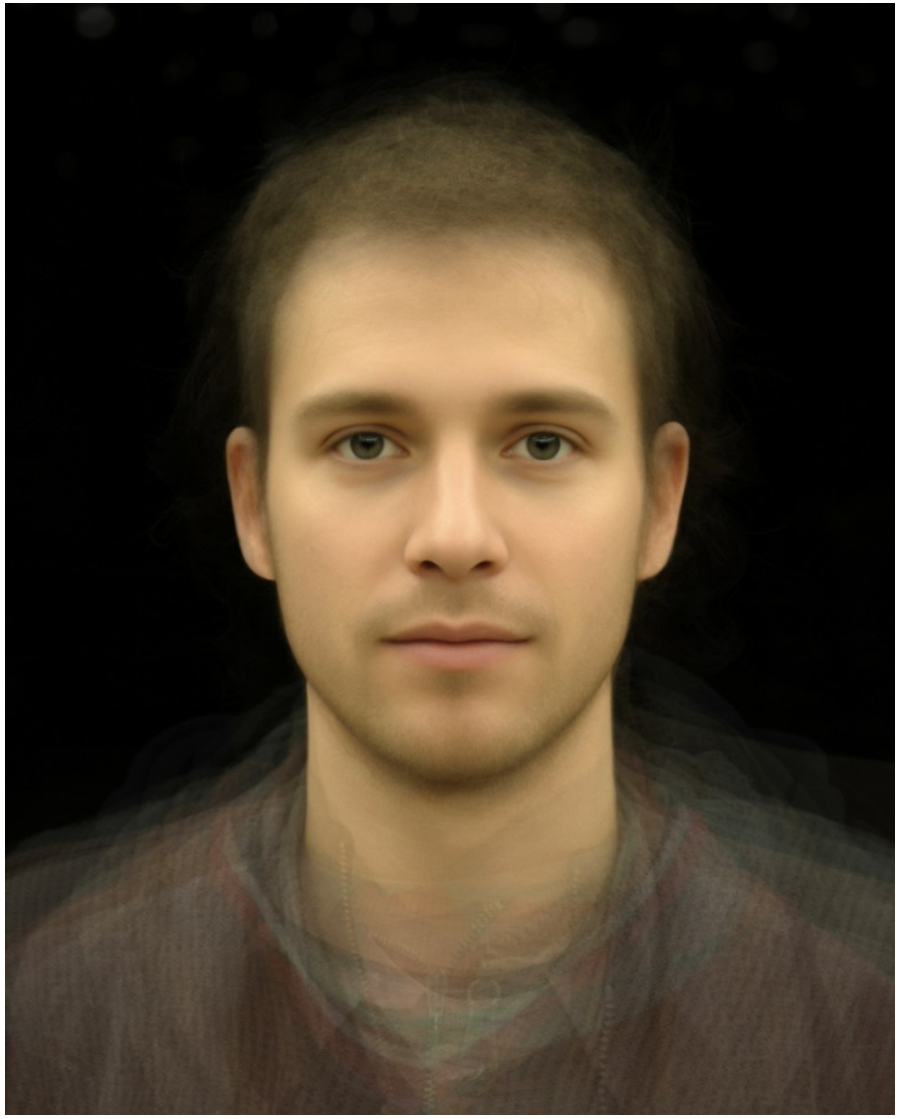
=

= mean image

Eigenfaces

Eigenfaces look somewhat like generic faces.

Eigen-images of Berlin



Eigen-images

Average of 16 individuals transformed via biometrical data of different ethnics

Average of 16 individuals transformed via biometrical data of different ages

Rank

Nullspace

(Fundamental matrix)

=

3 -0.9660 0.2586
-0.0029 -0.2586 -0.9660
-1.0000 -0.0005 0.0032

d(1,1)
ans = 1.0000e+004

d(2,2)
ans = 0.0021

d(3,3)
ans = 2.7838e-016

Rank(F) = 2

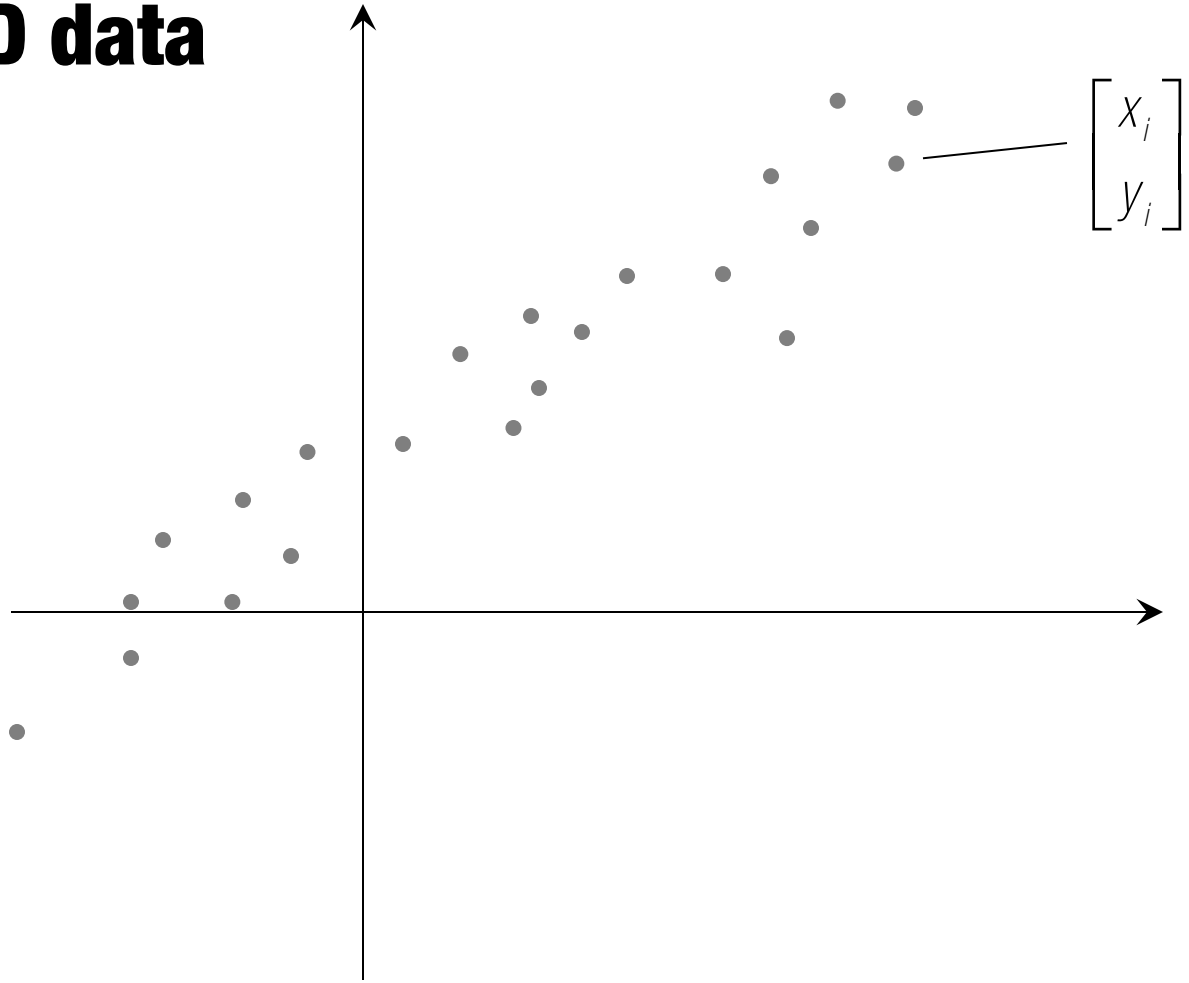
Matrix Inversion with SVD

Two types of Least Square Problem:

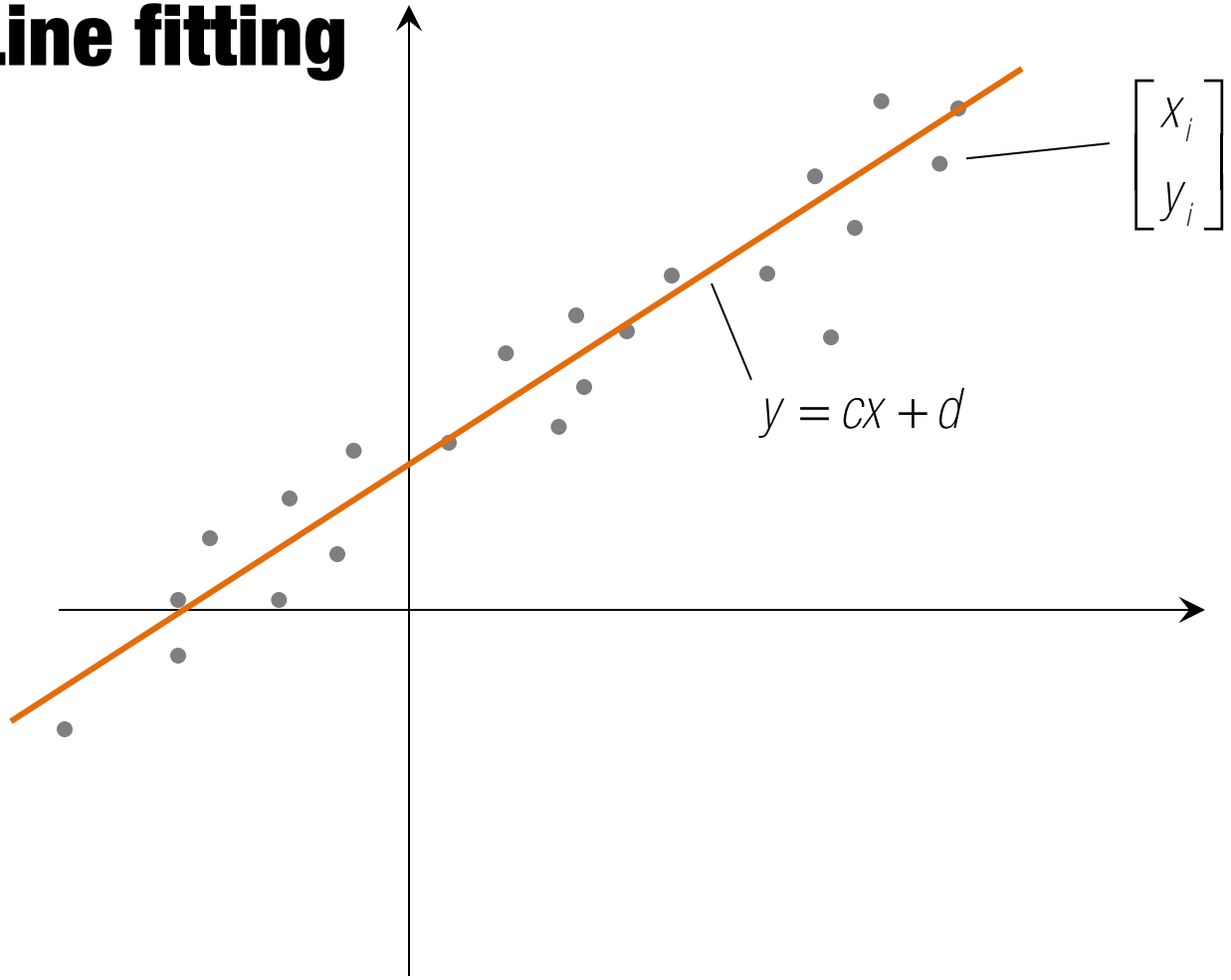
$$\min_{\mathbf{x}\in\mathbb{R}^n} \|\mathsf{A}\mathbf{x}-\mathbf{b}\|^2 \qquad \text{with} \qquad \|\mathbf{b}\| \neq 0$$

$$\min_{\mathbf{x}\in\mathbb{R}^n} \|\mathsf{A}\mathbf{x}\|^2 \qquad \text{s.t.} \qquad \|\mathbf{x}\| = 1$$

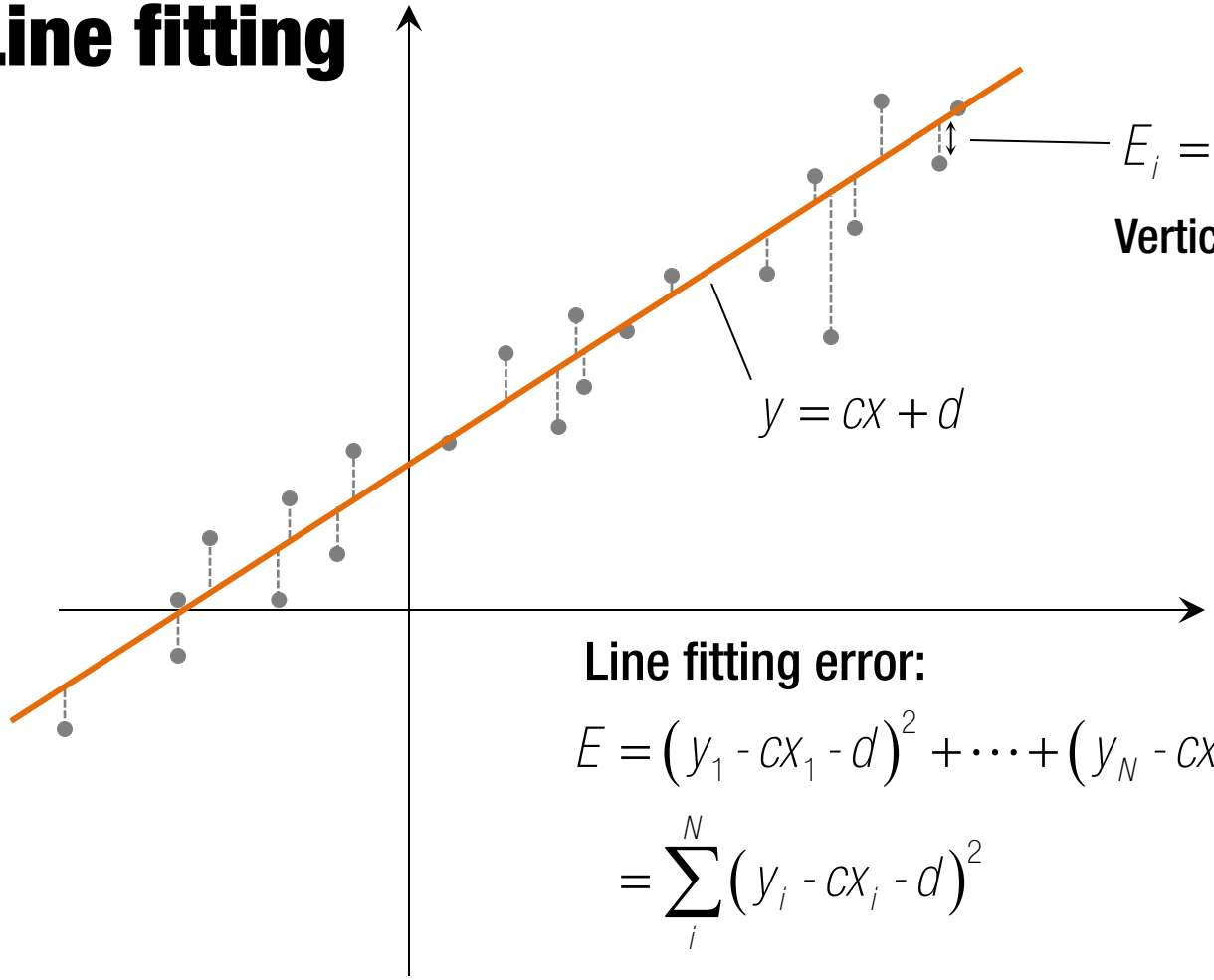
2D data



Line fitting



Line fitting



$$E_i = \left| y_i - cx_i - d \right|$$

Vertical distance

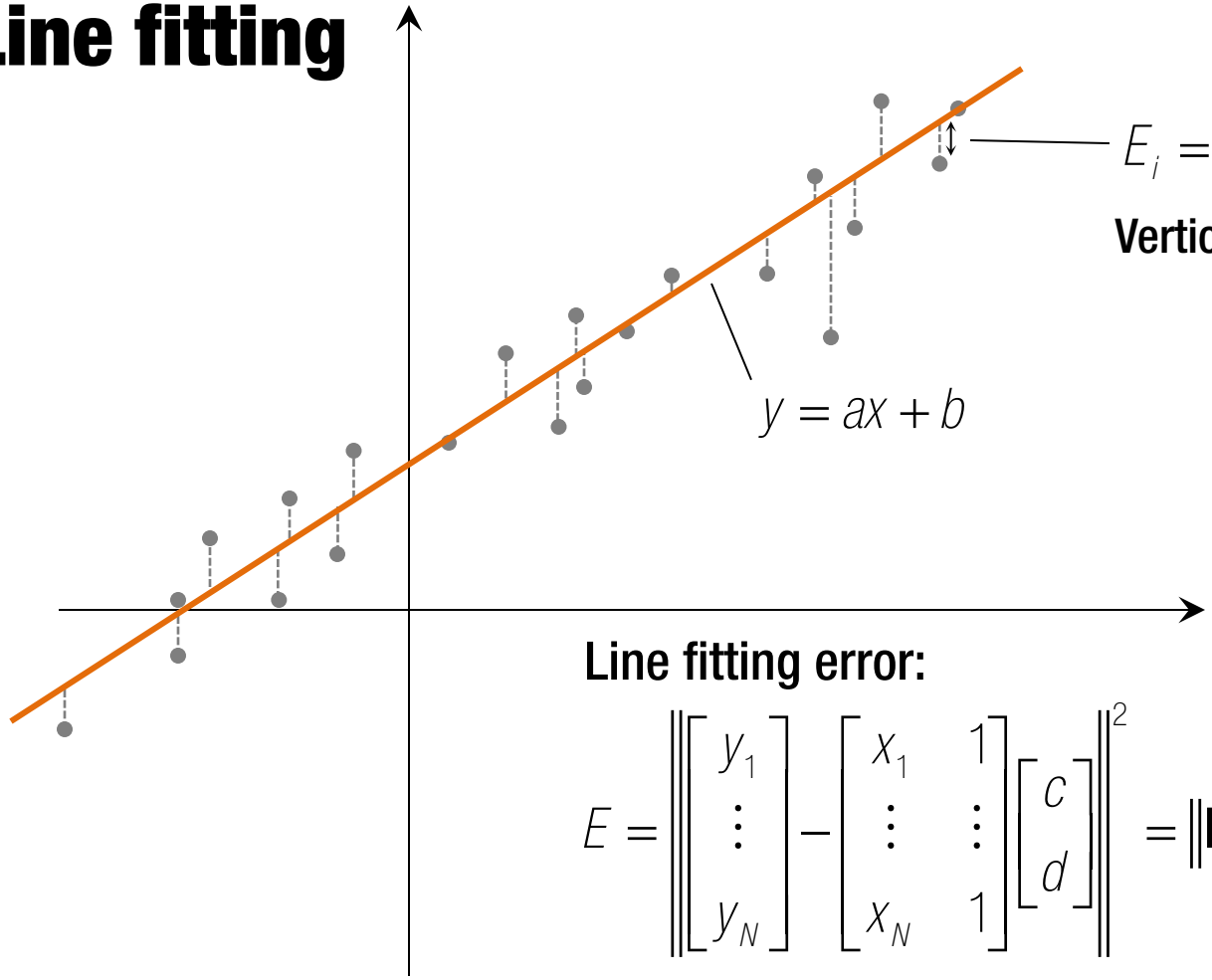
$$y = cx + d$$

Line fitting error:

$$E = \left(y_1 - cx_1 - d \right)^2 + \cdots + \left(y_N - cx_N - d \right)^2$$

$$= \sum_i^N \left(y_i - cx_i - d \right)^2$$

Line fitting



$$E_i = |y_i - ax_i - b|$$

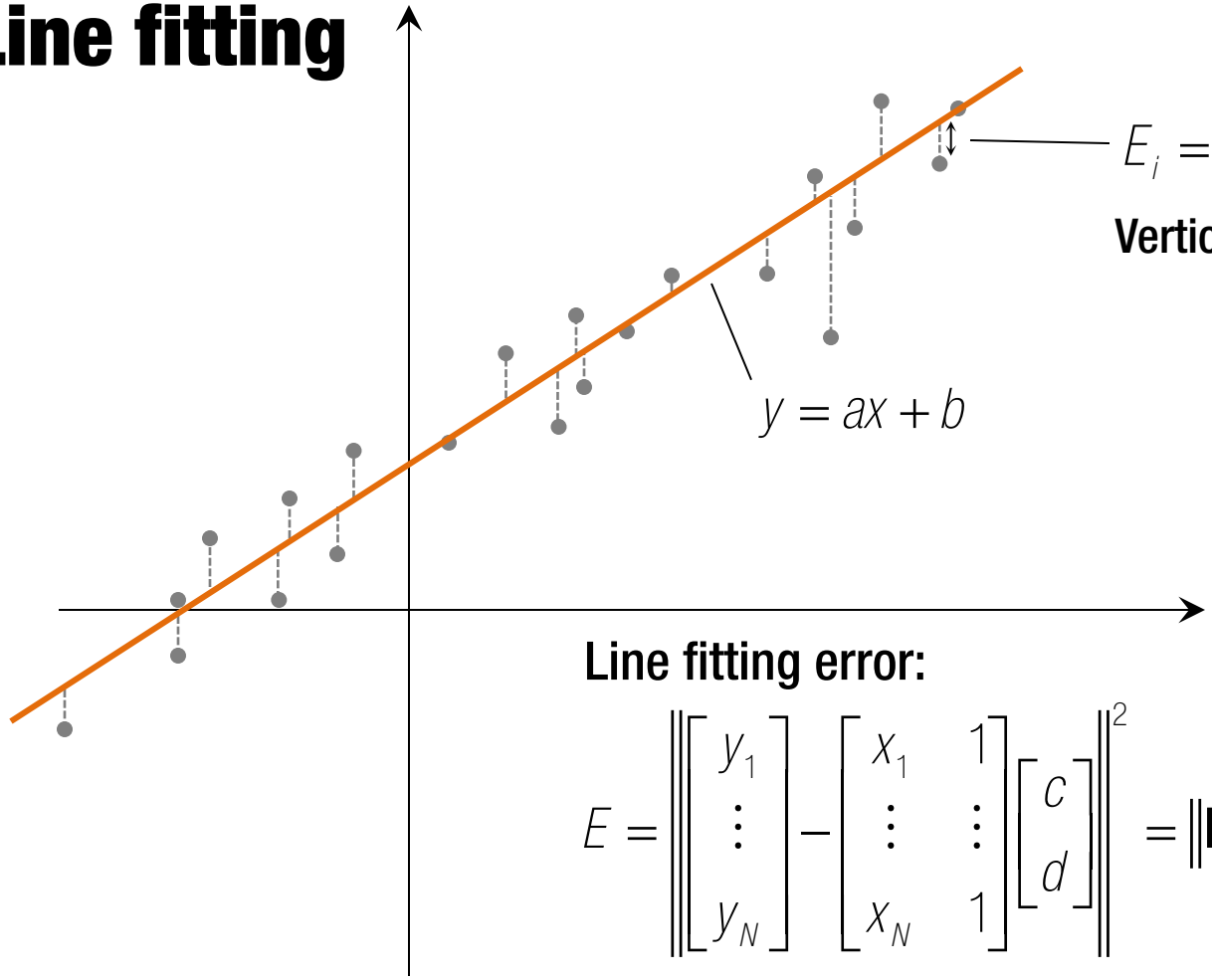
Vertical distance

$$y = ax + b$$

Line fitting error:

$$E = \left\| \begin{bmatrix} y_1 \\ \vdots \\ y_N \end{bmatrix} - \begin{bmatrix} x_1 & 1 \\ \vdots & \vdots \\ x_N & 1 \end{bmatrix} \begin{bmatrix} c \\ d \end{bmatrix} \right\|^2 = \|\mathbf{b} - \mathbf{Ax}\|^2$$

Line fitting



$E_i = |y_i - ax_i - b|$

Vertical distance

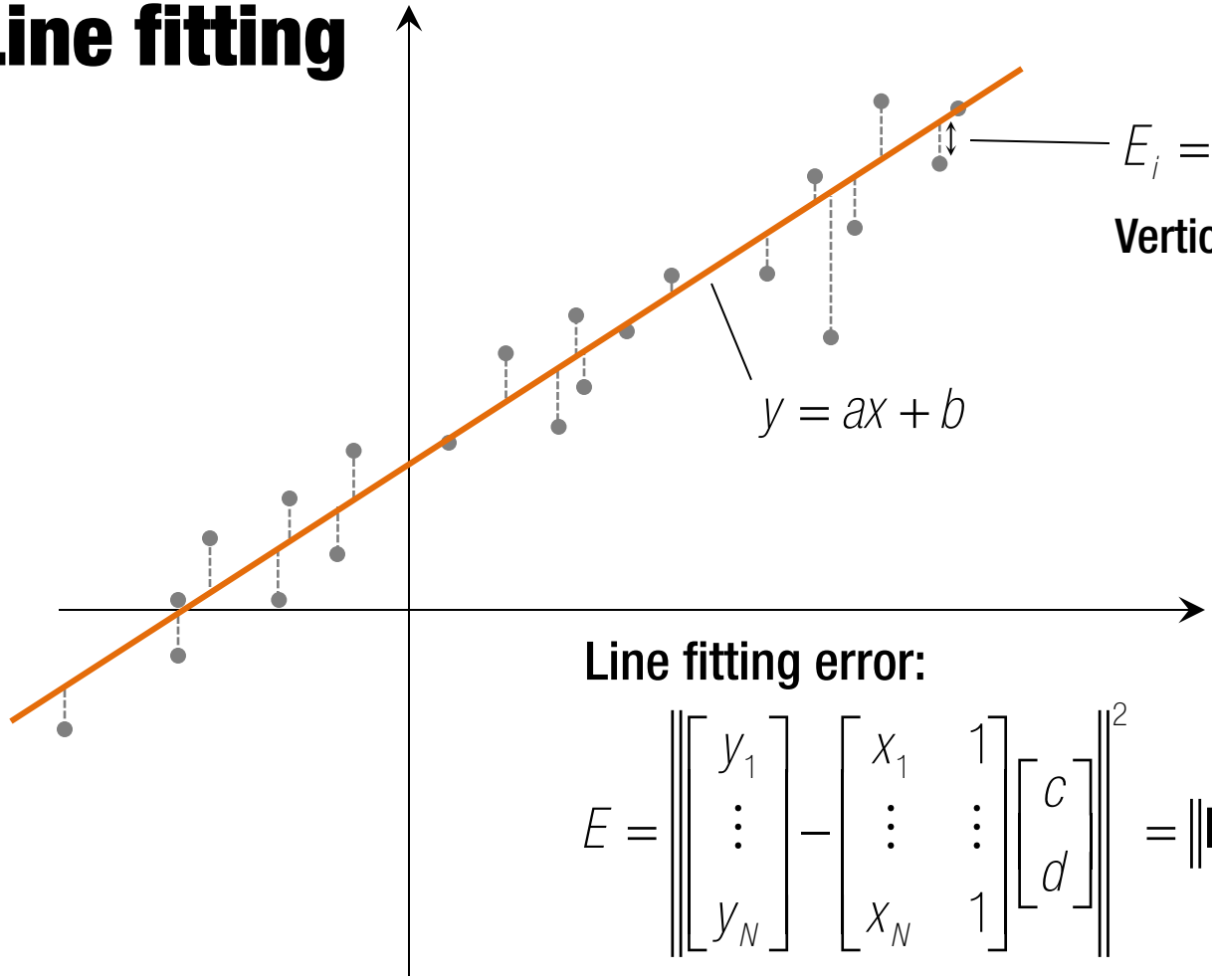
$y = ax + b$

Line fitting error:

$$E = \left\| \begin{bmatrix} y_1 \\ \vdots \\ y_N \end{bmatrix} - \begin{bmatrix} x_1 & 1 \\ \vdots & \vdots \\ x_N & 1 \end{bmatrix} \begin{bmatrix} c \\ d \end{bmatrix} \right\|^2 = \|\mathbf{b} - \mathbf{Ax}\|^2$$

$$= (\mathbf{b} - \mathbf{Ax})^{\mathsf{T}} (\mathbf{b} - \mathbf{Ax}) = \mathbf{b}^{\mathsf{T}}\mathbf{b} - 2\mathbf{b}^{\mathsf{T}}\mathbf{Ax} + \mathbf{x}^{\mathsf{T}}\mathbf{A}^{\mathsf{T}}\mathbf{Ax}$$

Line fitting



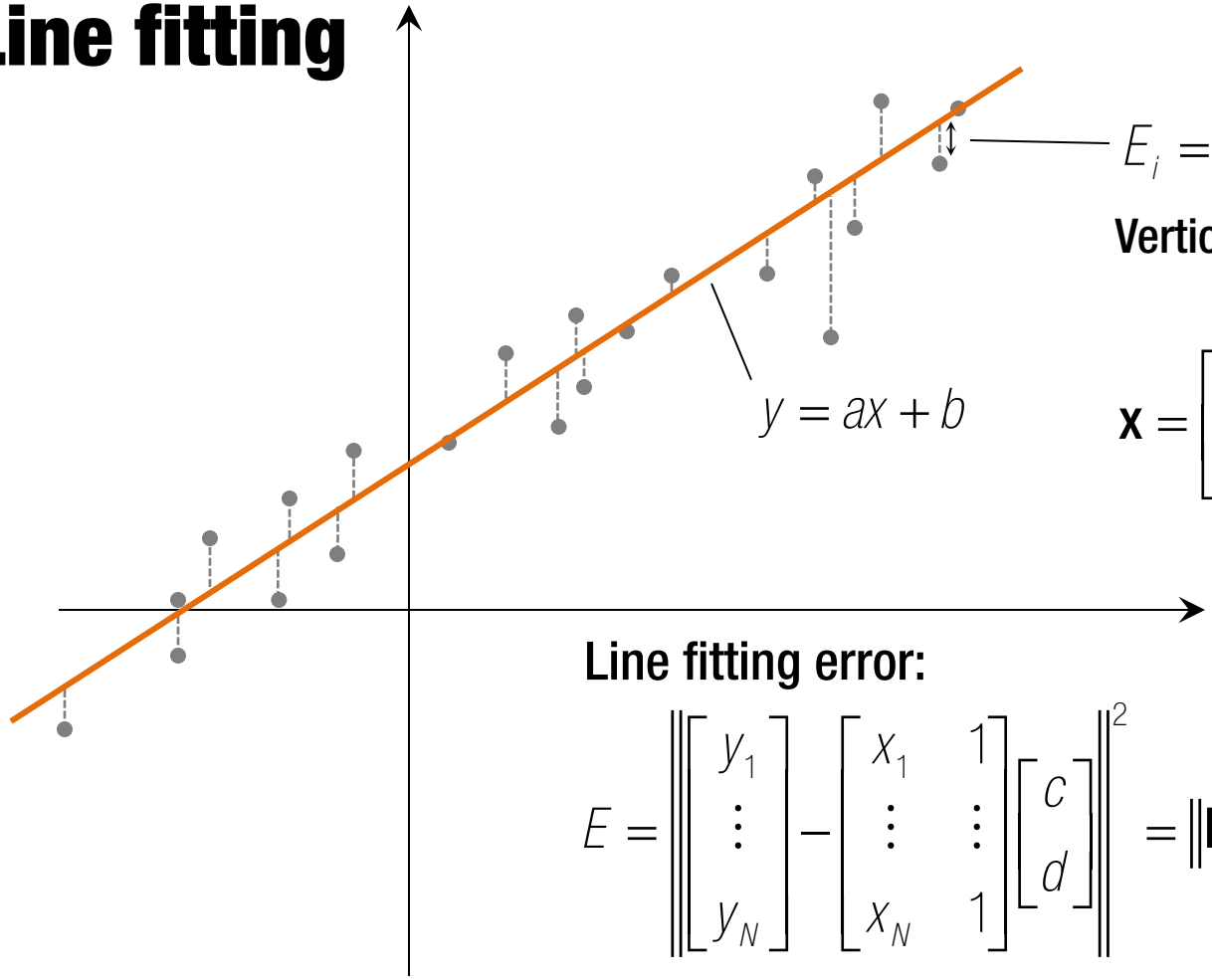
Line fitting error:

$$E = \left\| \begin{bmatrix} y_1 \\ \vdots \\ y_N \end{bmatrix} - \begin{bmatrix} x_1 & 1 \\ \vdots & \vdots \\ x_N & 1 \end{bmatrix} \begin{bmatrix} c \\ d \end{bmatrix} \right\|^2 = \|\mathbf{b} - \mathbf{Ax}\|^2$$

$$= (\mathbf{b} - \mathbf{Ax})^{\mathsf{T}} (\mathbf{b} - \mathbf{Ax}) = \mathbf{b}^{\mathsf{T}}\mathbf{b} - 2\mathbf{b}^{\mathsf{T}}\mathbf{Ax} + \mathbf{x}^{\mathsf{T}}\mathbf{A}^{\mathsf{T}}\mathbf{Ax}$$

$$\frac{\partial E}{\partial \mathbf{x}} = -2\mathbf{b}^{\mathsf{T}}\mathbf{A} + 2\mathbf{A}^{\mathsf{T}}\mathbf{Ax} = \mathbf{0} \rightarrow \mathbf{A}^{\mathsf{T}}\mathbf{Ax} = \mathbf{b}^{\mathsf{T}}\mathbf{A}$$

Line fitting



$$E_i = \left| y_i - ax_i - b \right|$$

Vertical distance

$$y = ax + b$$

$$\mathbf{x} = \begin{bmatrix} c \\ d \end{bmatrix} = \left(\mathbf{A}^{\mathsf{T}} \mathbf{A} \right)^{-1} \mathbf{b}^{\mathsf{T}} \mathbf{A}$$

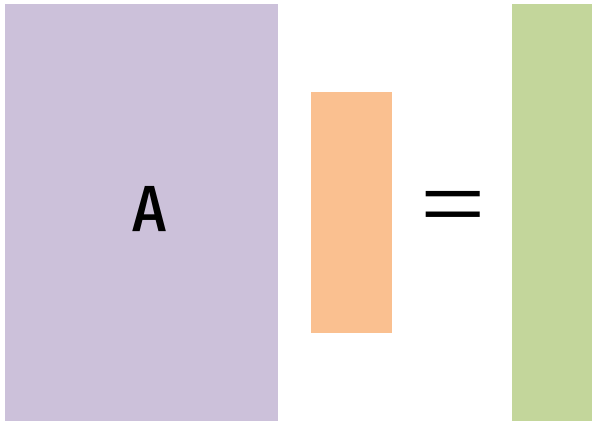
Line fitting error:

$$E = \left\| \begin{bmatrix} y_1 \\ \vdots \\ y_N \end{bmatrix} - \begin{bmatrix} x_1 & 1 \\ \vdots & \vdots \\ x_N & 1 \end{bmatrix} \begin{bmatrix} c \\ d \end{bmatrix} \right\|^2 = \left\| \mathbf{b} - \mathbf{A}\mathbf{x} \right\|^2$$

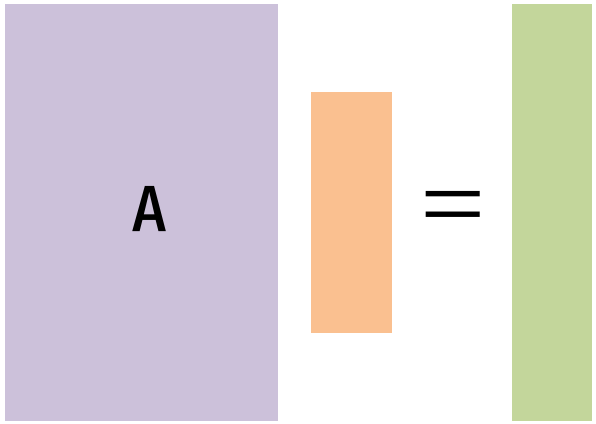
$$= \left(\mathbf{b} - \mathbf{A}\mathbf{x} \right)^{\mathsf{T}} \left(\mathbf{b} - \mathbf{A}\mathbf{x} \right) = \mathbf{b}^{\mathsf{T}}\mathbf{b} - 2\mathbf{b}^{\mathsf{T}}\mathbf{A}\mathbf{x} + \mathbf{x}^{\mathsf{T}}\mathbf{A}^{\mathsf{T}}\mathbf{A}\mathbf{x}$$

$$\frac{\partial E}{\partial \mathbf{x}} = -2\mathbf{b}^{\mathsf{T}}\mathbf{A} + 2\mathbf{A}^{\mathsf{T}}\mathbf{A}\mathbf{x} = \mathbf{0} \rightarrow \mathbf{A}^{\mathsf{T}}\mathbf{A}\mathbf{x} = \mathbf{b}^{\mathsf{T}}\mathbf{A}$$

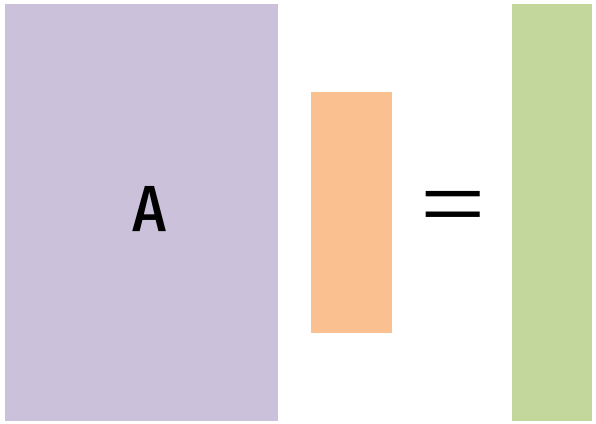
Linear Inhomogeneous Equations



Linear Inhomogeneous Equations



Linear Inhomogeneous Equations



$$x = A^{-1}b$$

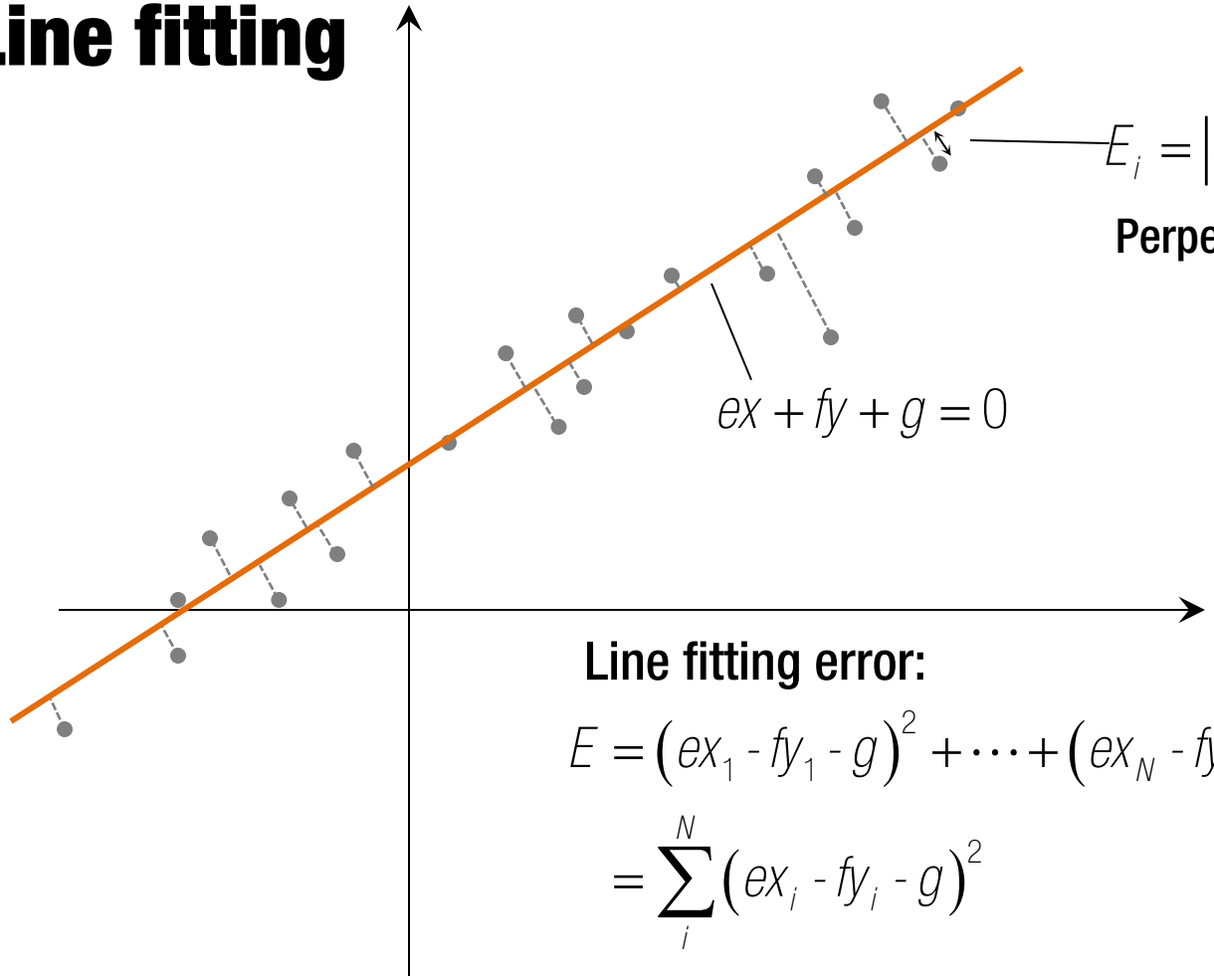
1 ...

Two types of Least Square Problem:

$$\min_{\mathbf{x}\in\mathbb{R}^n} \|\mathbf{A}\mathbf{x}-\mathbf{b}\|^2 \qquad \text{with} \qquad \|\mathbf{b}\| \neq 0$$

$$\min_{\mathbf{x}\in\mathbb{R}^n} \|\mathbf{A}\mathbf{x}\|^2 \qquad \text{s.t.} \qquad \|\mathbf{x}\| = 1$$

Line fitting



$E_i = |ex_i + fy_i + g|$

Perpendicular distance

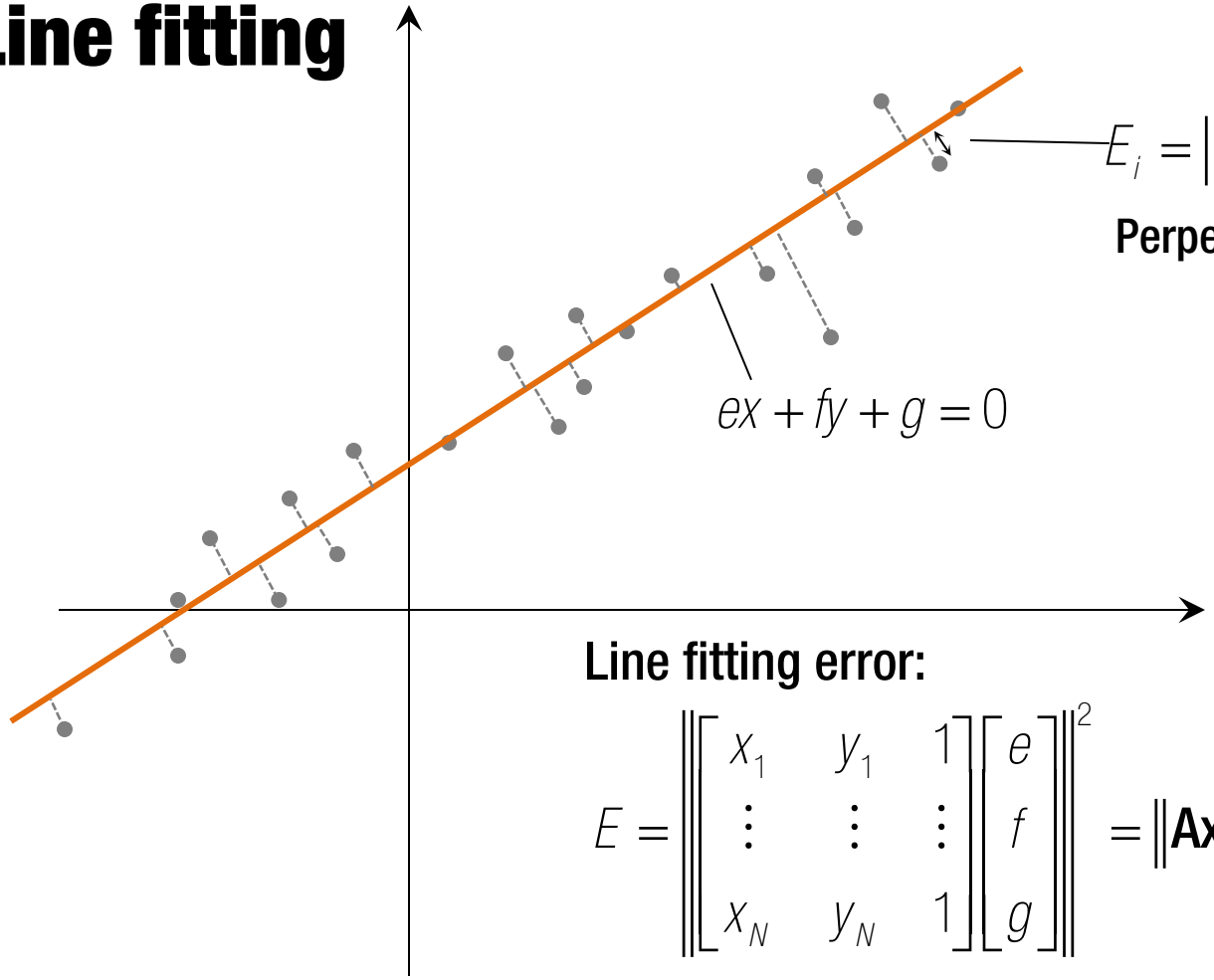
$ex + fy + g = 0$

Line fitting error:

$$E = (ex_1 - fy_1 - g)^2 + \cdots + (ex_N - fy_N - g)^2$$

$$= \sum_i^N (ex_i - fy_i - g)^2$$

Line fitting



$$E_i = |ex_i + fy_i + g|$$

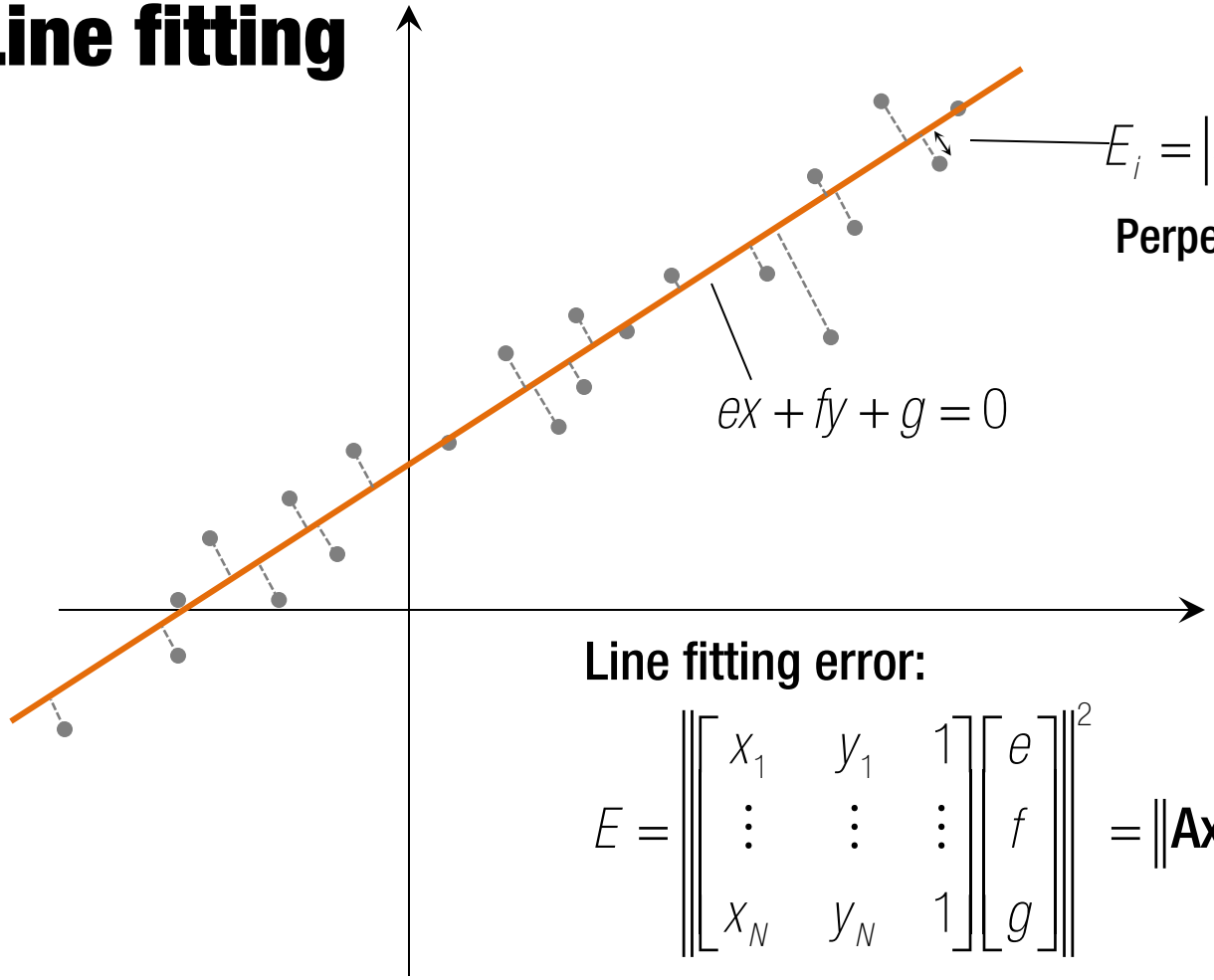
Perpendicular distance

$$ex + fy + g = 0$$

Line fitting error:

$$E = \left\| \begin{bmatrix} x_1 & y_1 & 1 \\ \vdots & \vdots & \vdots \\ x_N & y_N & 1 \end{bmatrix} \begin{bmatrix} e \\ f \\ g \end{bmatrix} \right\|^2 = \|\mathbf{Ax}\|^2$$

Line fitting



$$E_i = |ex_i + fy_i + g|$$

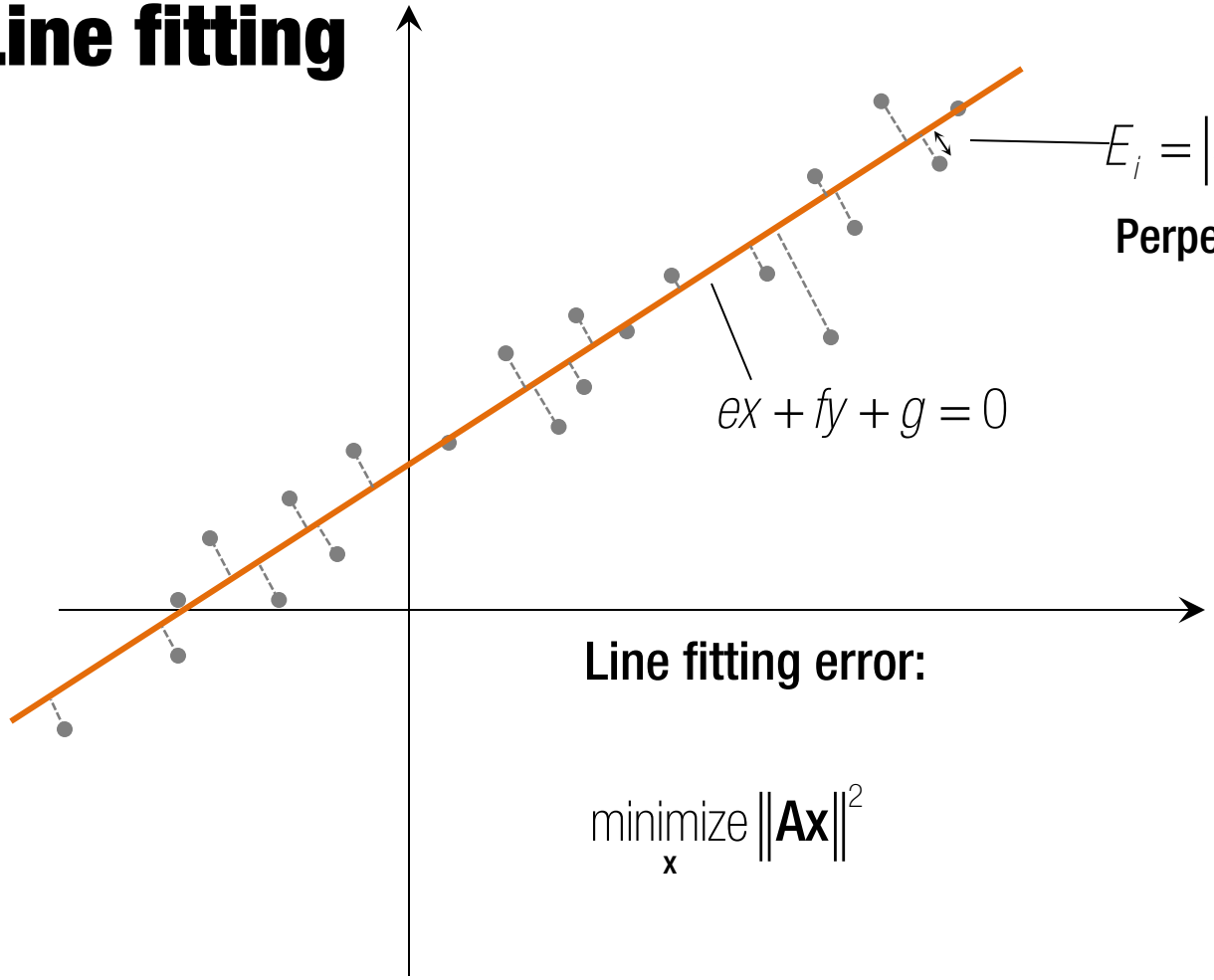
Perpendicular distance

$$ex + fy + g = 0$$

Line fitting error:

$$E = \left\| \begin{bmatrix} x_1 & y_1 & 1 \\ \vdots & \vdots & \vdots \\ x_N & y_N & 1 \end{bmatrix} \begin{bmatrix} e \\ f \\ g \end{bmatrix} \right\|^2 = \|\mathbf{Ax}\|^2$$

Line fitting



$E_i = |ex_i + fy_i + g|$

Perpendicular distance

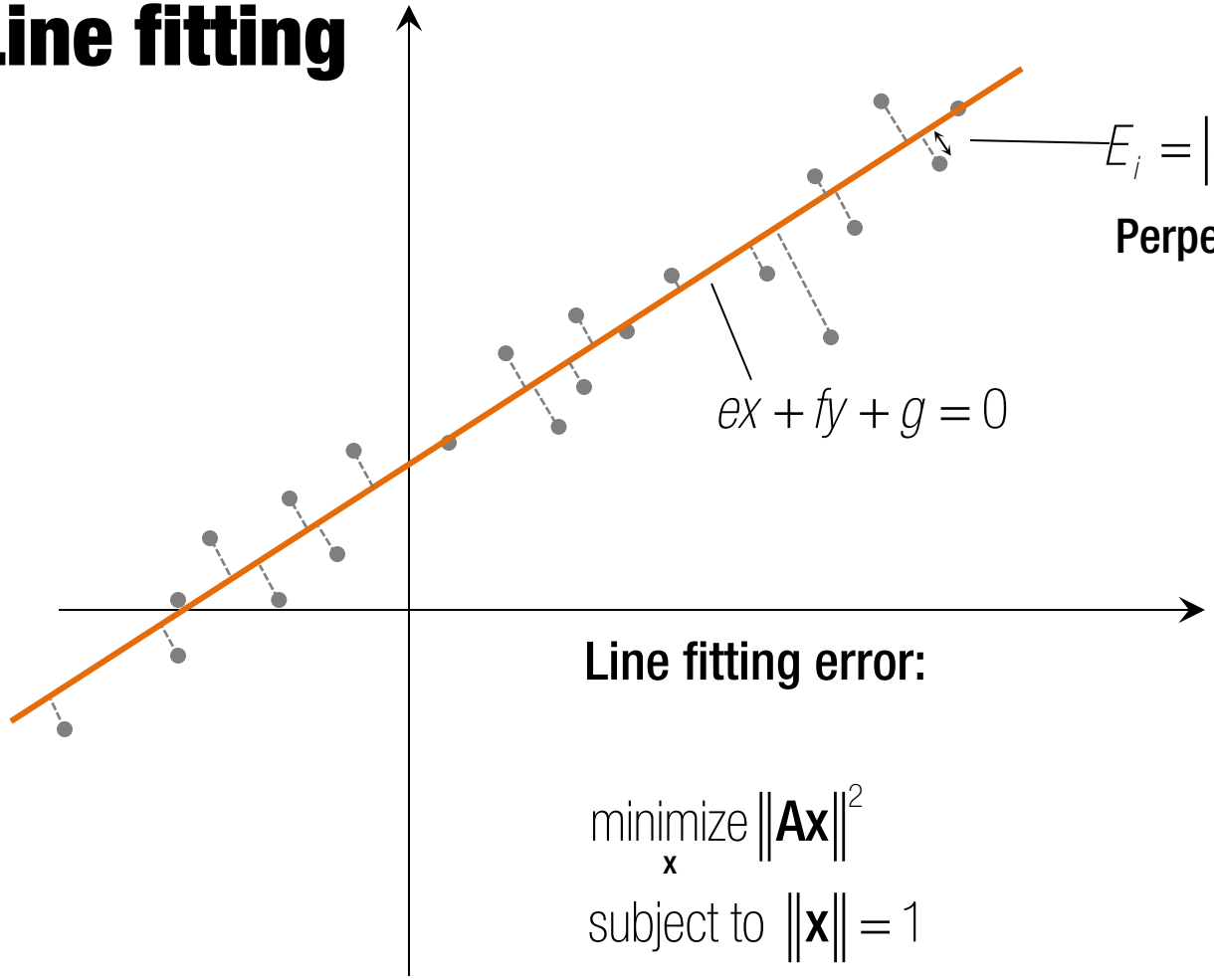
$ex + fy + g = 0$

Line fitting error:

$$\underset{\mathbf{x}}{\text{minimize}} \|\mathbf{Ax}\|^2$$

Trivial solution: $\mathbf{x} = \mathbf{0}$

Line fitting



$E_i = |ex_i + fy_i + g|$

Perpendicular distance

$ex + fy + g = 0$

Line fitting error:

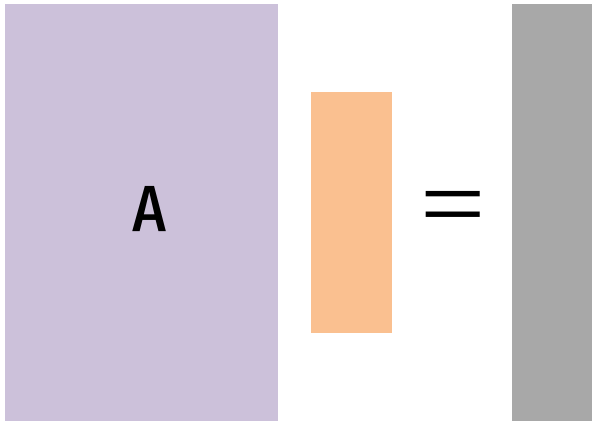
$$\underset{\mathbf{x}}{\text{minimize}} \|\mathbf{Ax}\|^2$$

$$\text{subject to } \|\mathbf{x}\| = 1$$

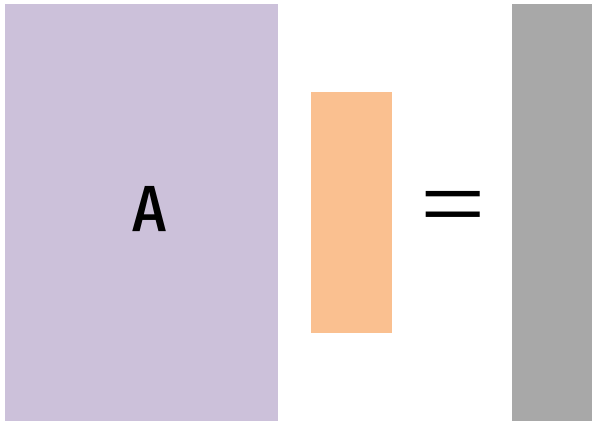
$$\mathbf{x} = \mathbf{V}_3 \quad \text{where } \mathbf{A} = \mathbf{UDV}^T$$

$$\mathbf{V} = [\mathbf{V}_1 \quad \mathbf{V}_2 \quad \mathbf{V}_3]$$

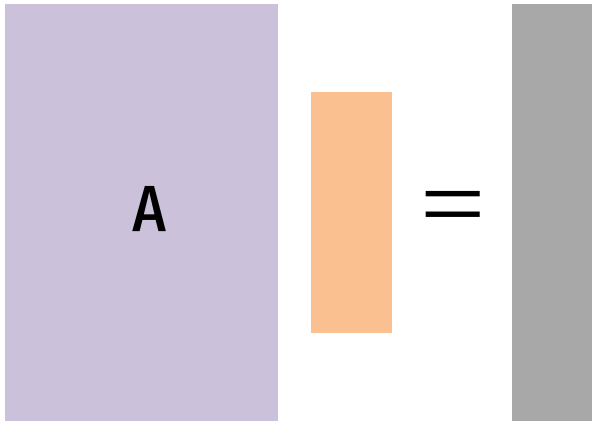
Linear Homogeneous Equations



Linear Homogeneous Equations



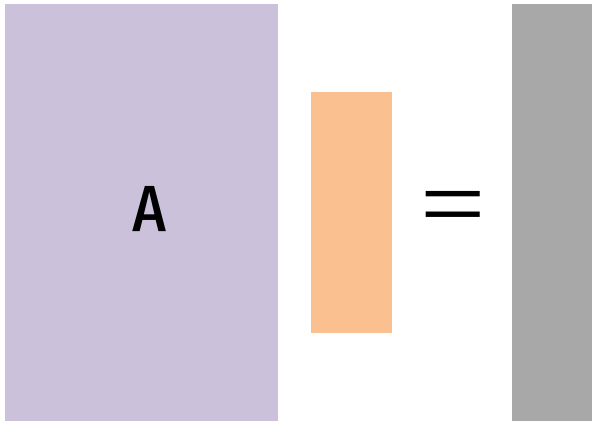
Linear Homogeneous Equations



lution

$$x = V_n$$

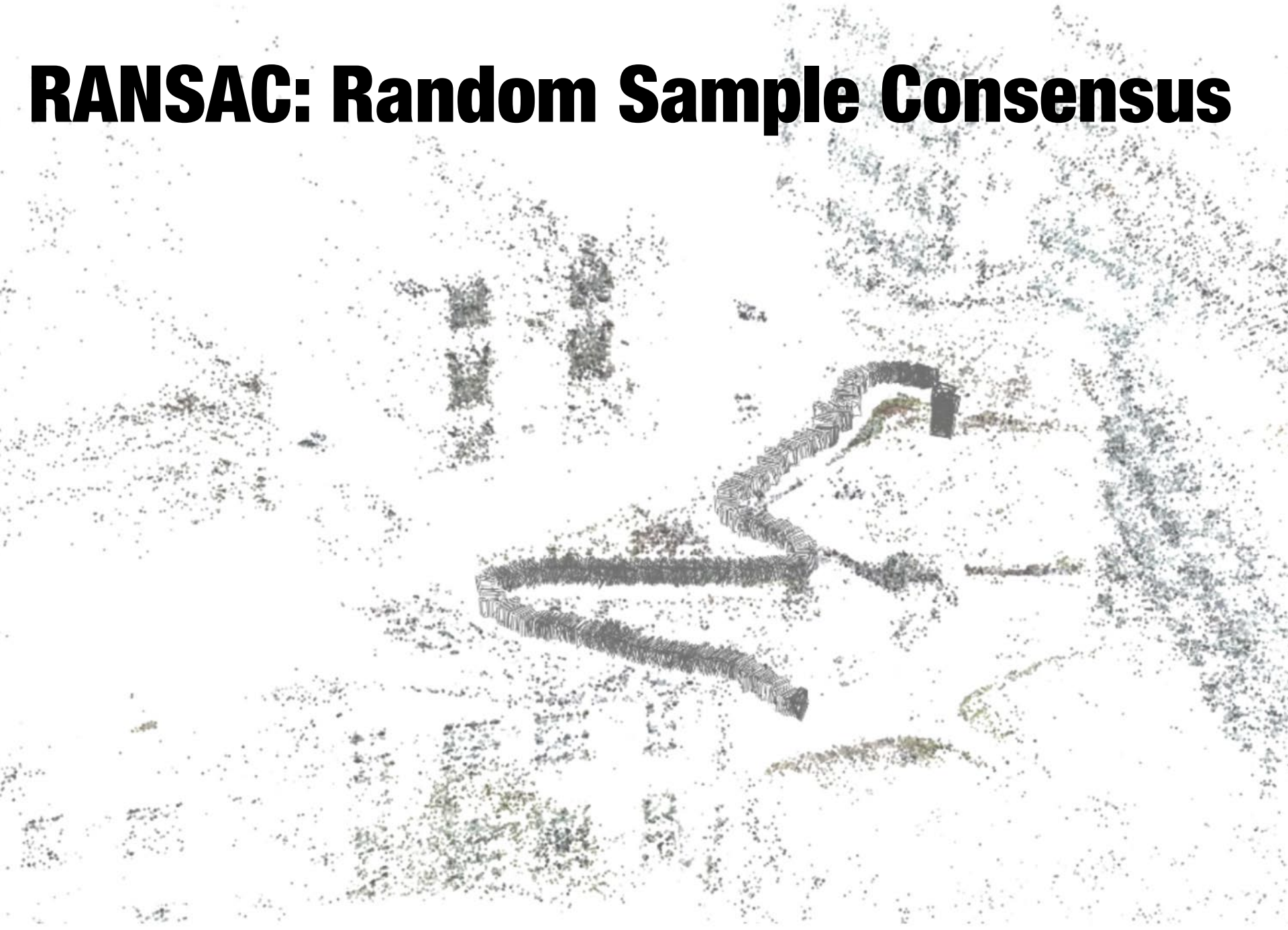
Linear Homogeneous Equations



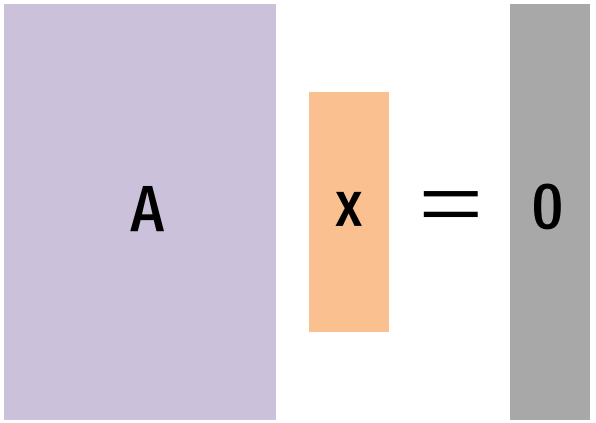
lution

$$x = V_n$$

RANSAC: Random Sample Consensus



Linear Homogeneous Equations



Linear least square solve produces a trivial solution:

$$\mathbf{x} = \left(\mathbf{A}^{\top}\mathbf{A}\right)^{-1}\mathbf{A}^{\top}\mathbf{b} \;\rightarrow\; \mathbf{x} = \mathbf{0}$$

An additional constraint on $\mathbf{x}$ to avoid the trivial solution: $\|\mathbf{x}\| = 1$

1) $\mathbf{rank(A)} = r < n - 1$: infinite number of solutions

$$\mathbf{x} = \lambda_{r+1}\mathbf{V}_{r+1} + \cdots + \lambda_n\mathbf{V}_n \quad \text{where} \quad \sum_{i=r+1}^{n} \lambda_i^2 = 1$$

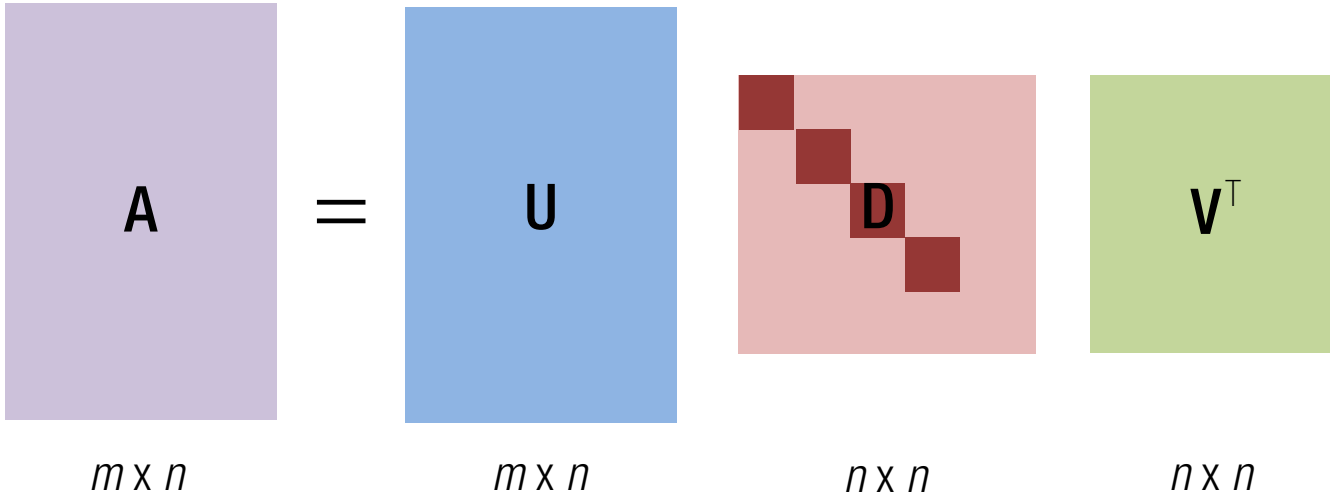
2) $\mathbf{rank(A)} = n - 1$: one exact solution

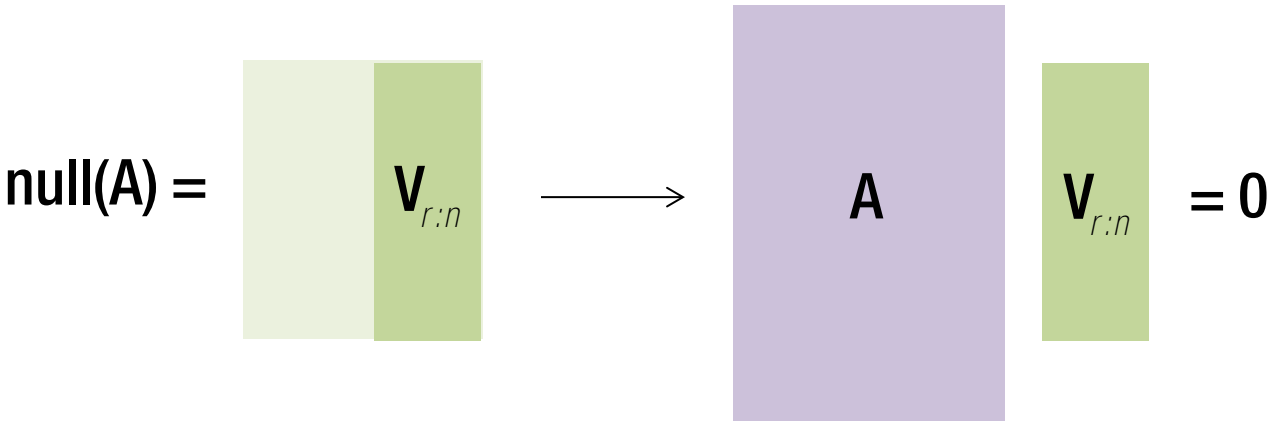
$$\mathbf{x} = \mathbf{V}_n$$

3) $n < m$: no exact solution in general (needs least squares)

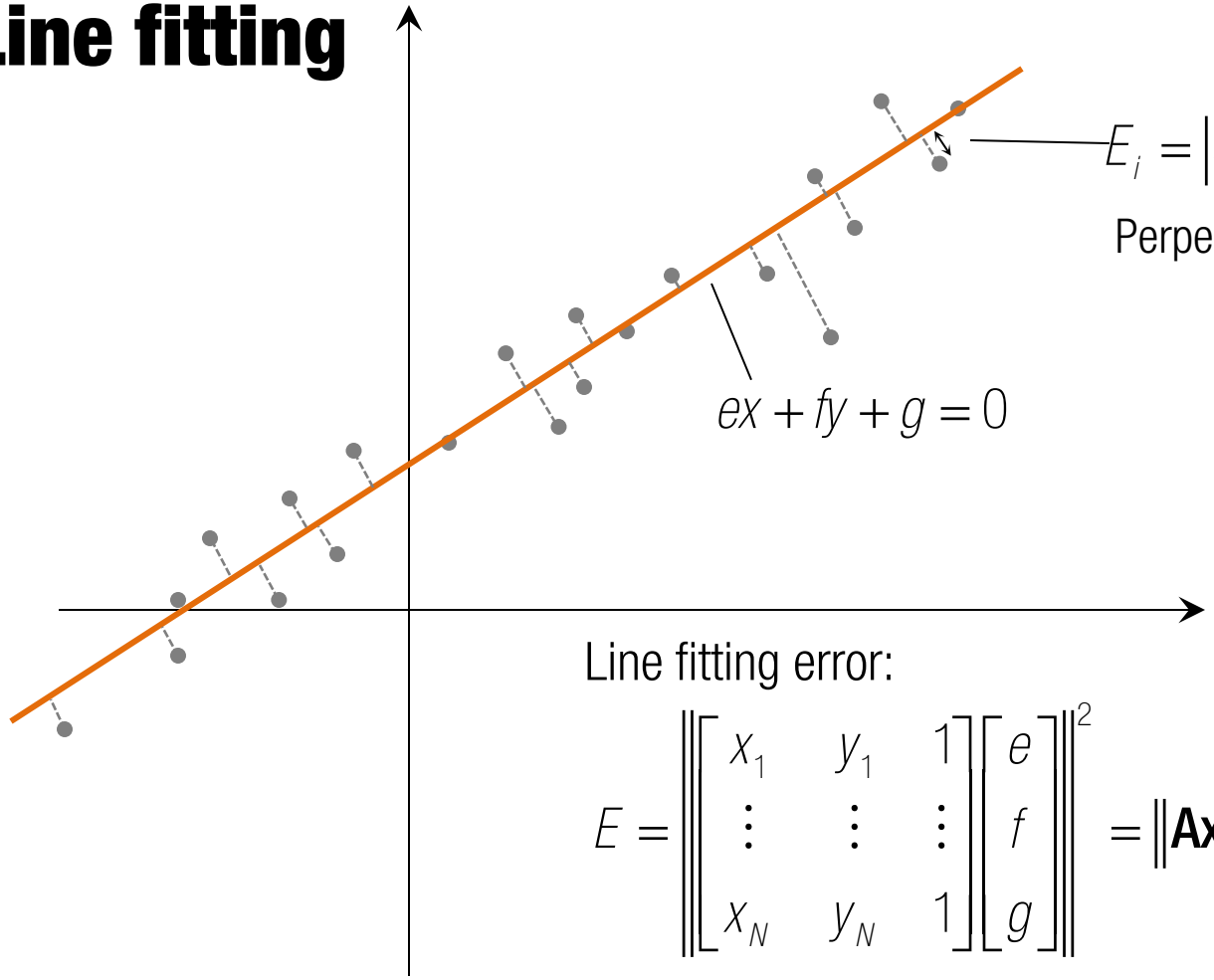
$$\min_{\mathbf{x}} \|\mathbf{Ax}\|^2 \text{ subject to } \|\mathbf{x}\| = 1 \;\rightarrow\; \mathbf{x} = \mathbf{V}_n$$

Nullspace





Line fitting



$E_i = |ex_i + fy_i + g|$

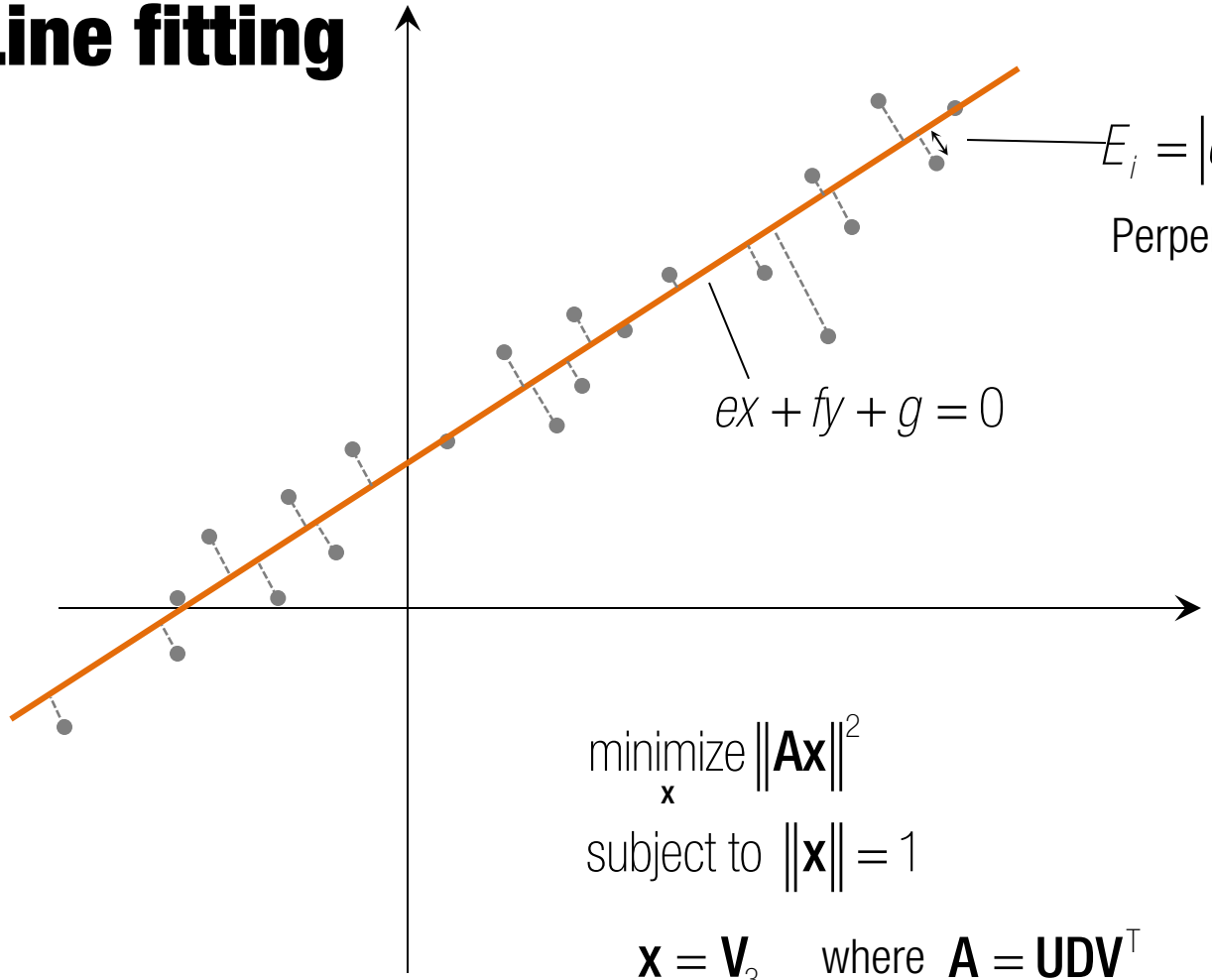
Perpendicular distance

$ex + fy + g = 0$

Line fitting error:

$$E = \left\| \begin{bmatrix} x_1 & y_1 & 1 \\ \vdots & \vdots & \vdots \\ x_N & y_N & 1 \end{bmatrix} \begin{bmatrix} e \\ f \\ g \end{bmatrix} \right\|^2 = \|\mathbf{Ax}\|^2$$

Line fitting



$E_i = |ex_i + fy_i + g|$

Perpendicular distance

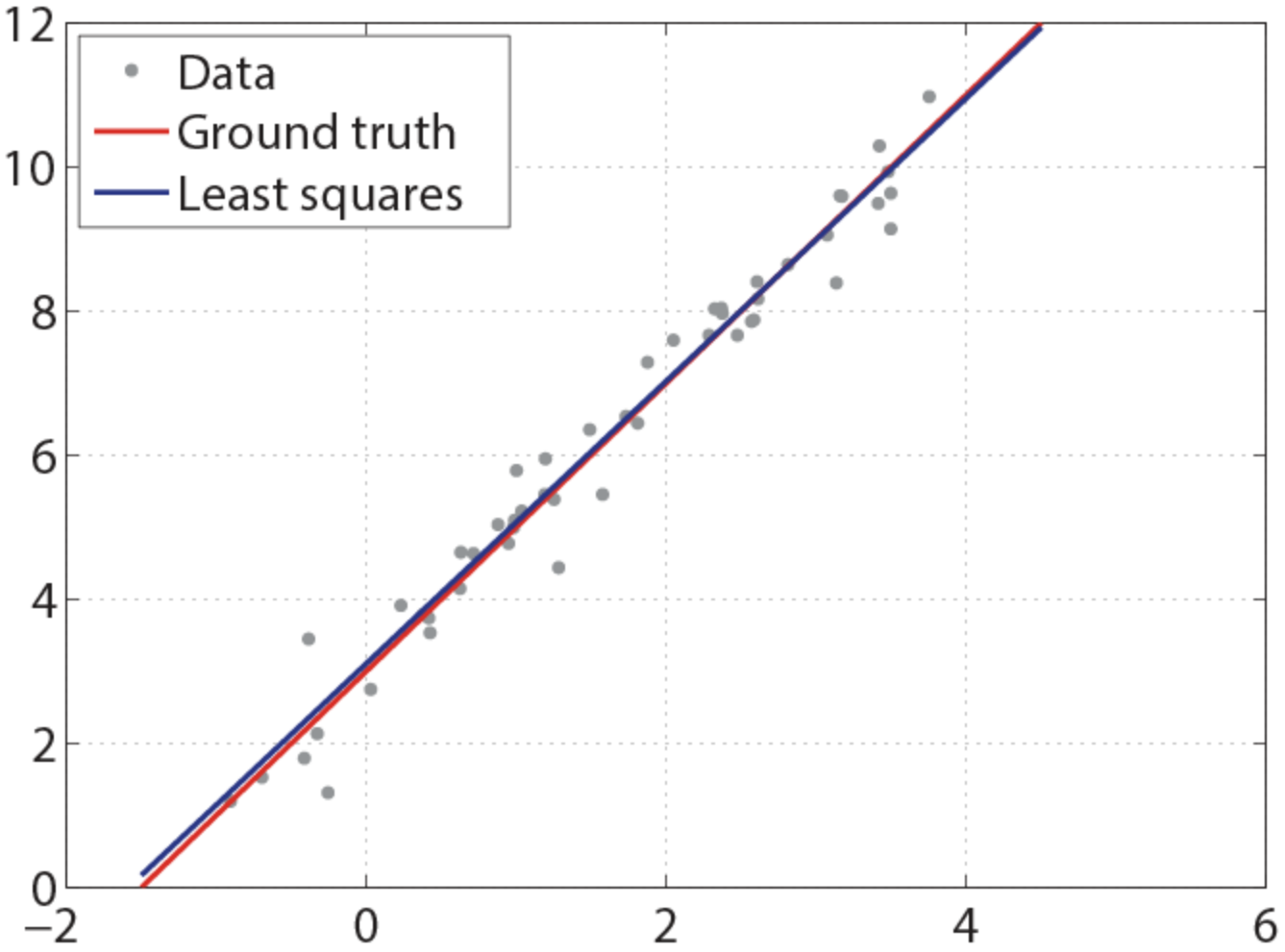
$ex + fy + g = 0$

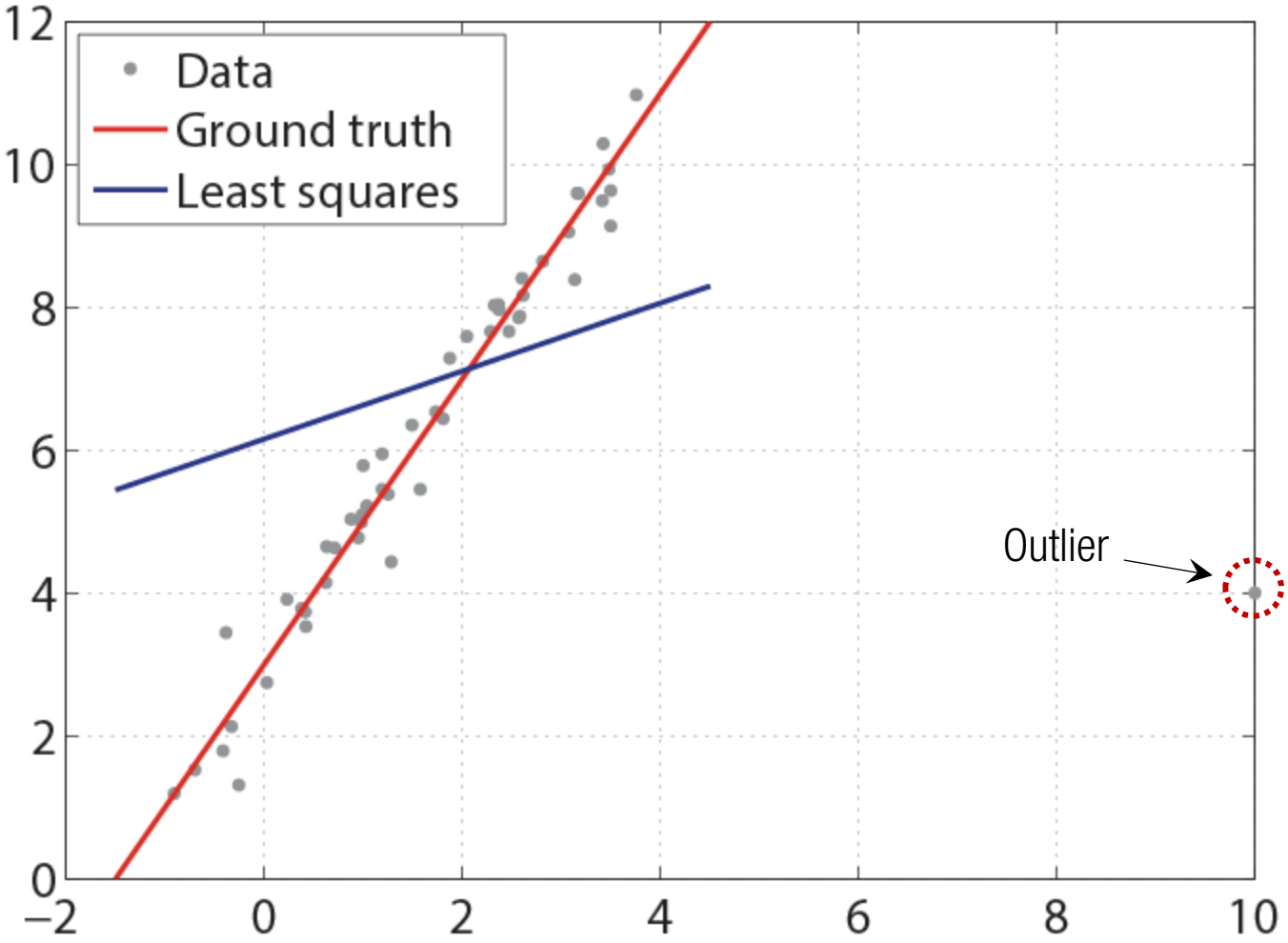
$$\underset{\mathbf{x}}{\text{minimize}} \|\mathbf{Ax}\|^2$$

$$\text{subject to } \|\mathbf{x}\| = 1$$

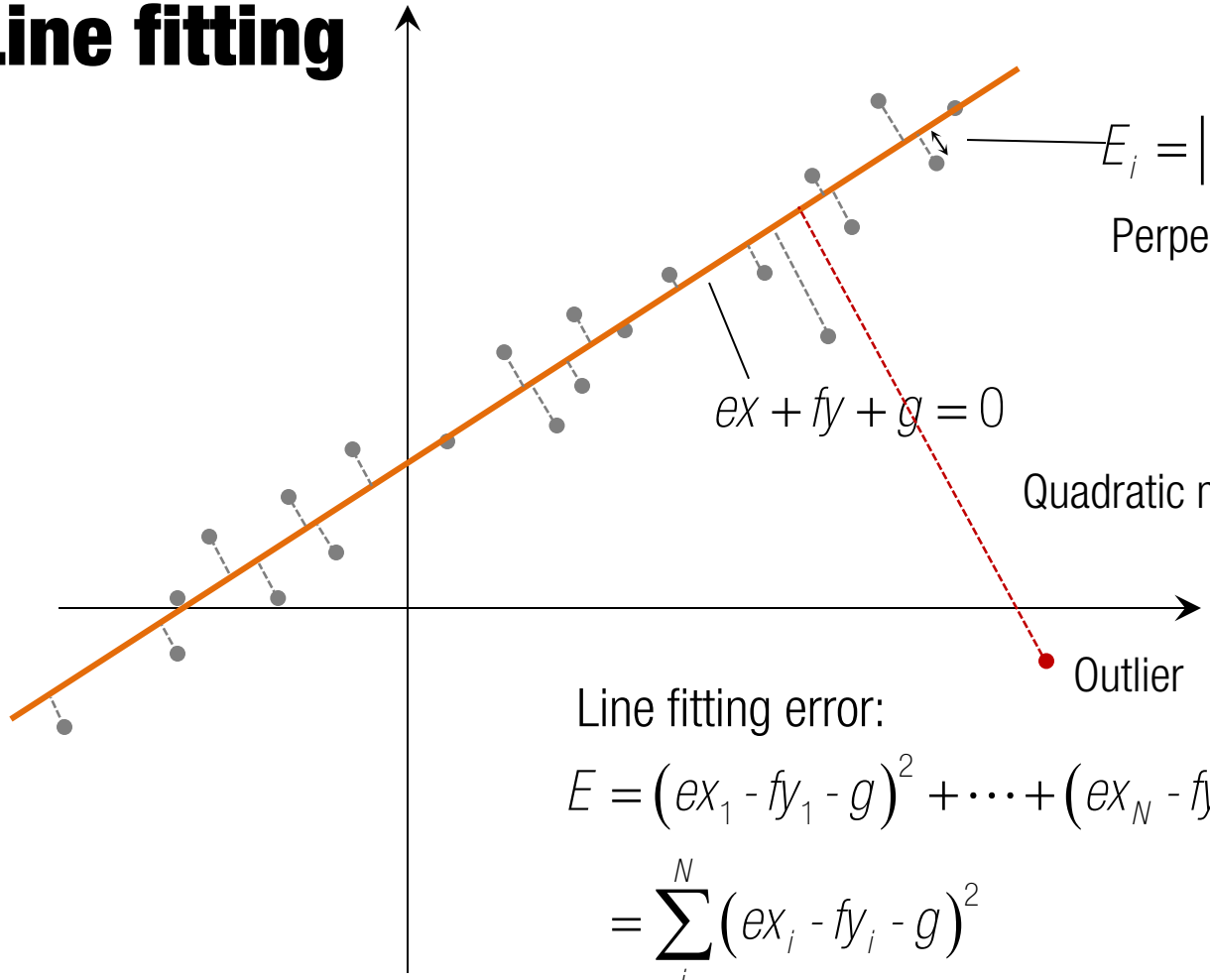
$$\mathbf{x} = \mathbf{V}_3 \quad \text{where } \mathbf{A} = \mathbf{UDV}^T$$

$$\mathbf{V} = [\mathbf{V}_1 \quad \mathbf{V}_2 \quad \mathbf{V}_3]$$


Data
Ground truth
Least squares
12
10
8
6
4
2
0
−2
0
2
4
6


Data
Ground truth
Least squares
Outlier
12
10
8
6
4
2
0
−2
0
2
4
6
8
10

Line fitting



$E_i = |ex_i + fy_i + g|$

Perpendicular distance

$ex + fy + g = 0$

Quadratic magnification of error of outliers

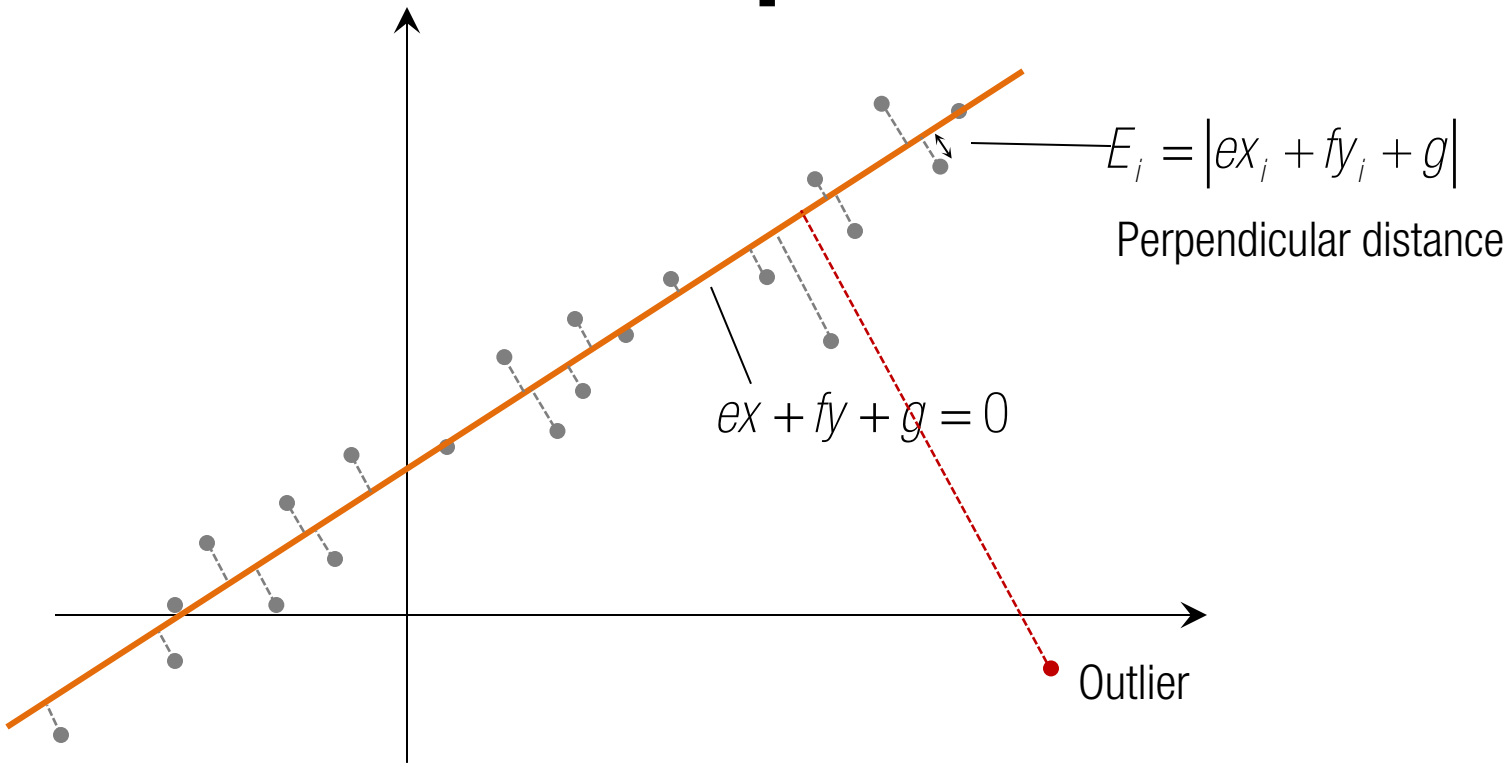
Outlier

Line fitting error:

$$E = (ex_1 - fy_1 - g)^2 + \cdots + (ex_N - fy_N - g)^2$$

$$= \sum_i^N (ex_i - fy_i - g)^2$$

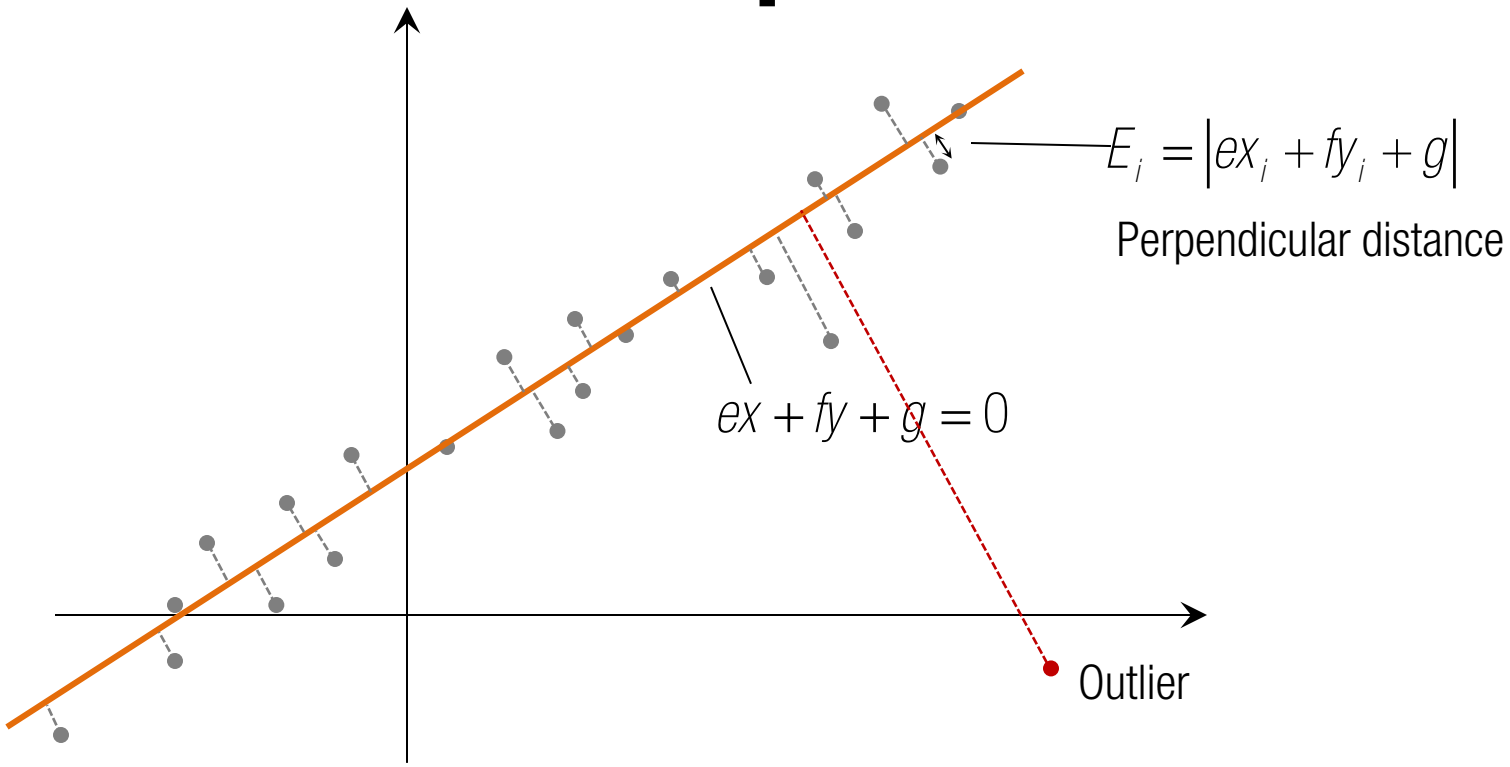
RANSAC: Random Sample Consensus



Strategy:

To find a model that accords with the maximum number of samples

RANSAC: Random Sample Consensus



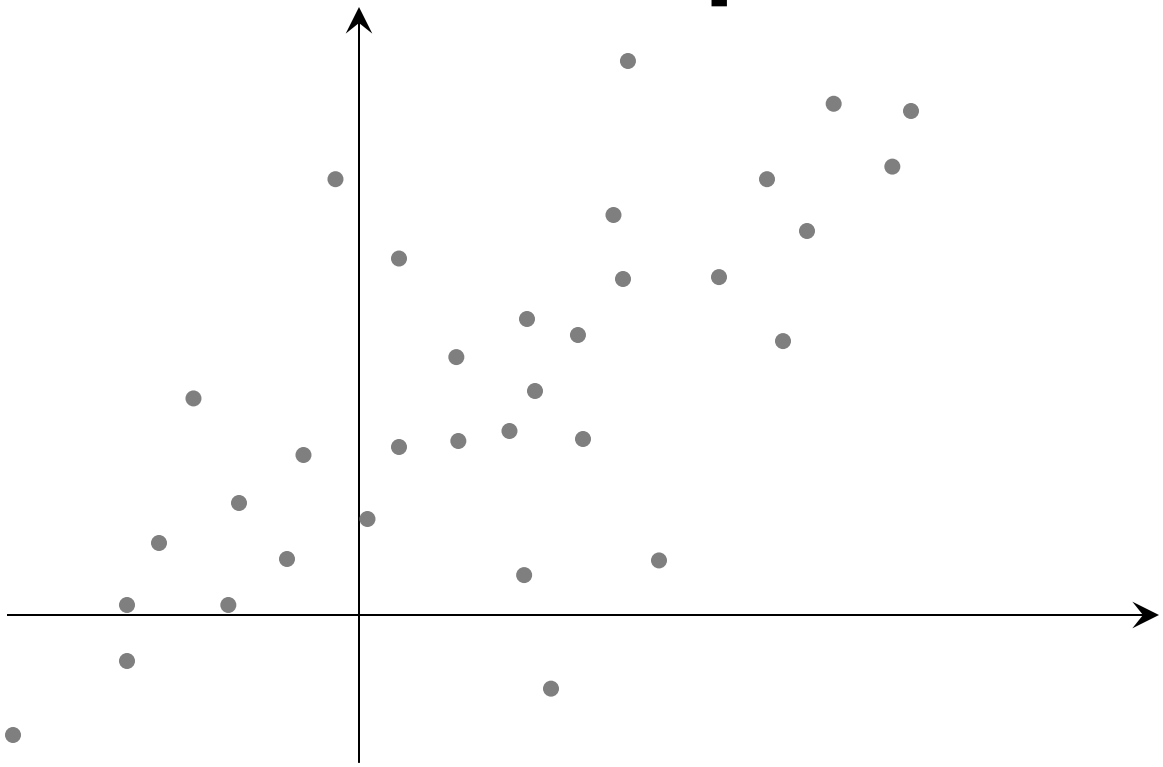
Strategy:

To find a model that accords with the maximum number of samples

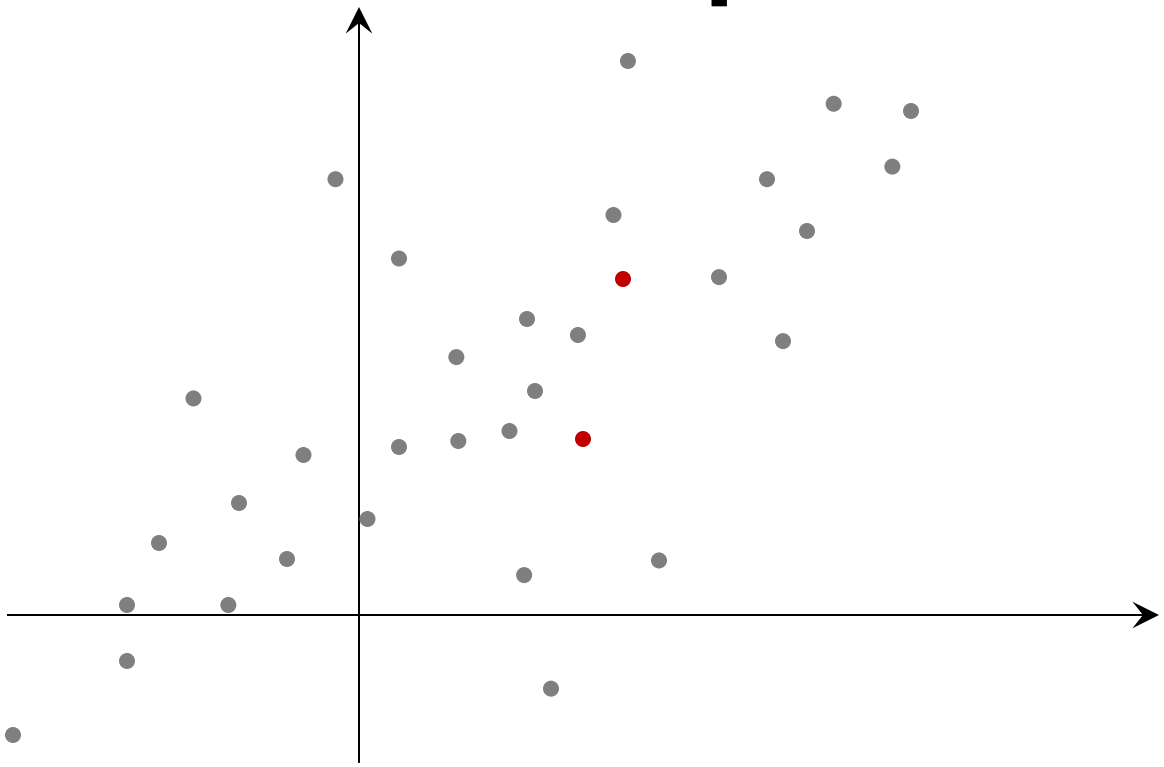
Assumptions:

1. Majority of good samples agree with the underlying model (good apples are same and simple.).
2. Bad samples does not consistently agree with a single model (all bad apples are different and complicated.).

RANSAC: Random Sample Consensus

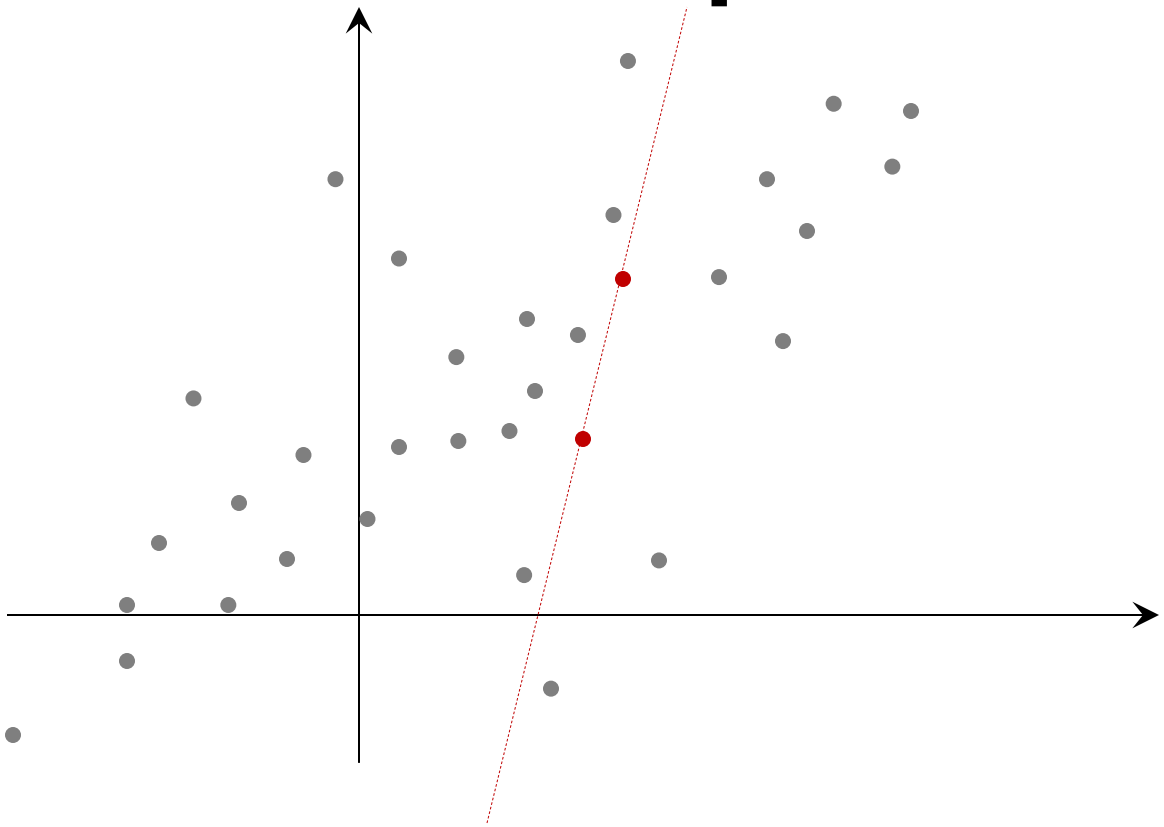


RANSAC: Random Sample Consensus



Random sampling

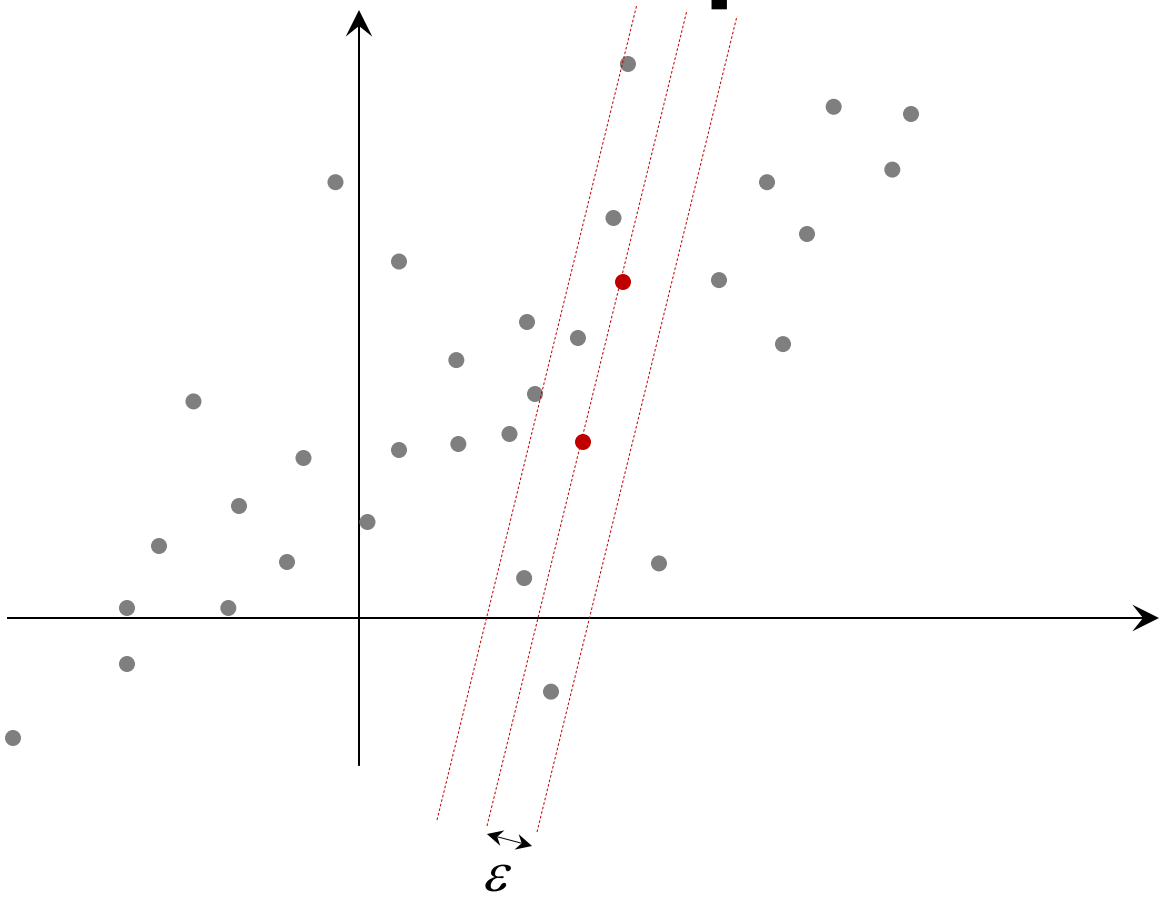
RANSAC: Random Sample Consensus



Random sampling

Model building

RANSAC: Random Sample Consensus

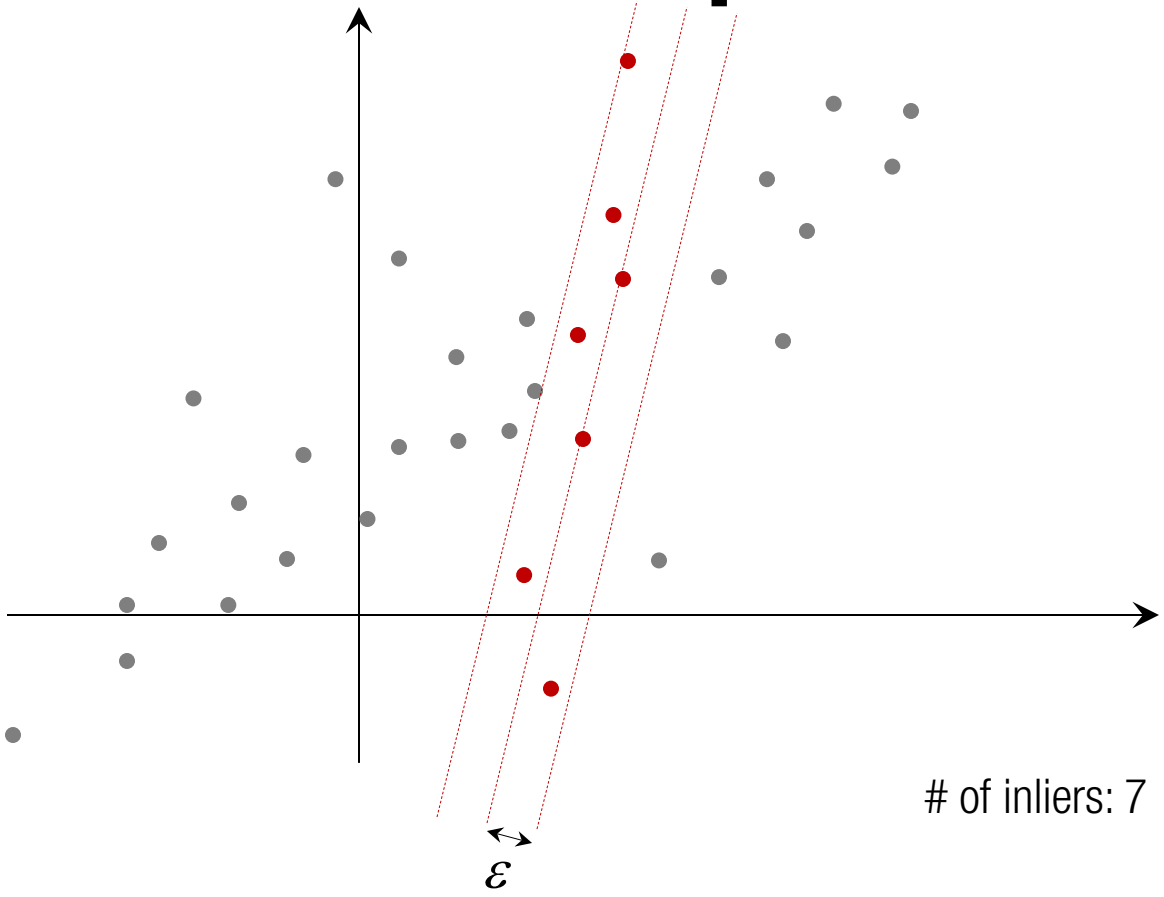


Random sampling

Model building

Thresholding

RANSAC: Random Sample Consensus



Random sampling

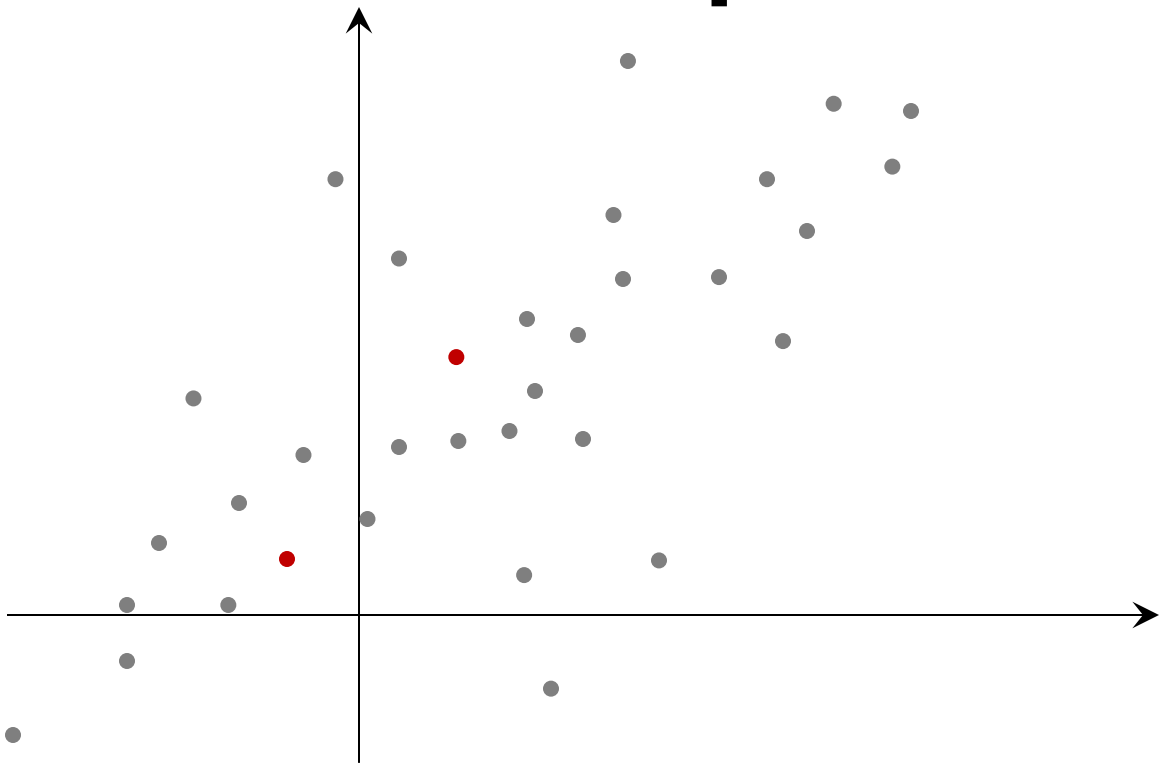
Model building

Thresholding

Inlier counting

# of inliers: 7

RANSAC: Random Sample Consensus



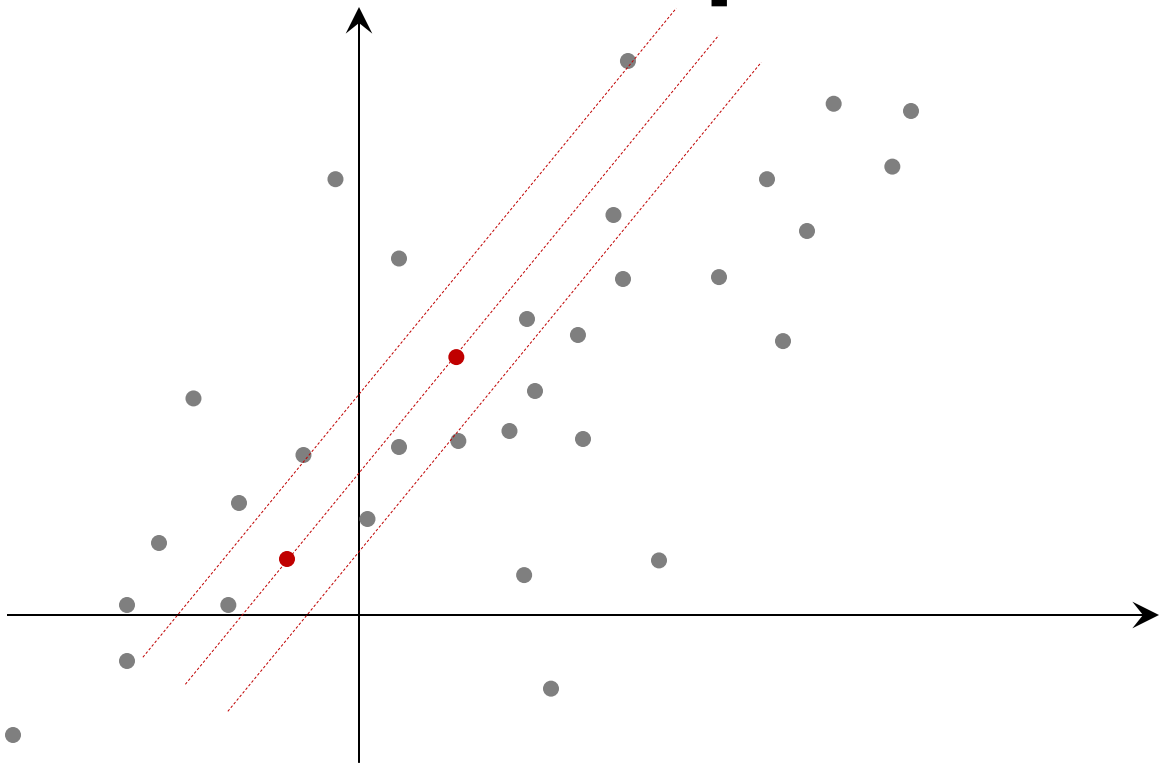
Random sampling

Model building

Thresholding

Inlier counting

RANSAC: Random Sample Consensus



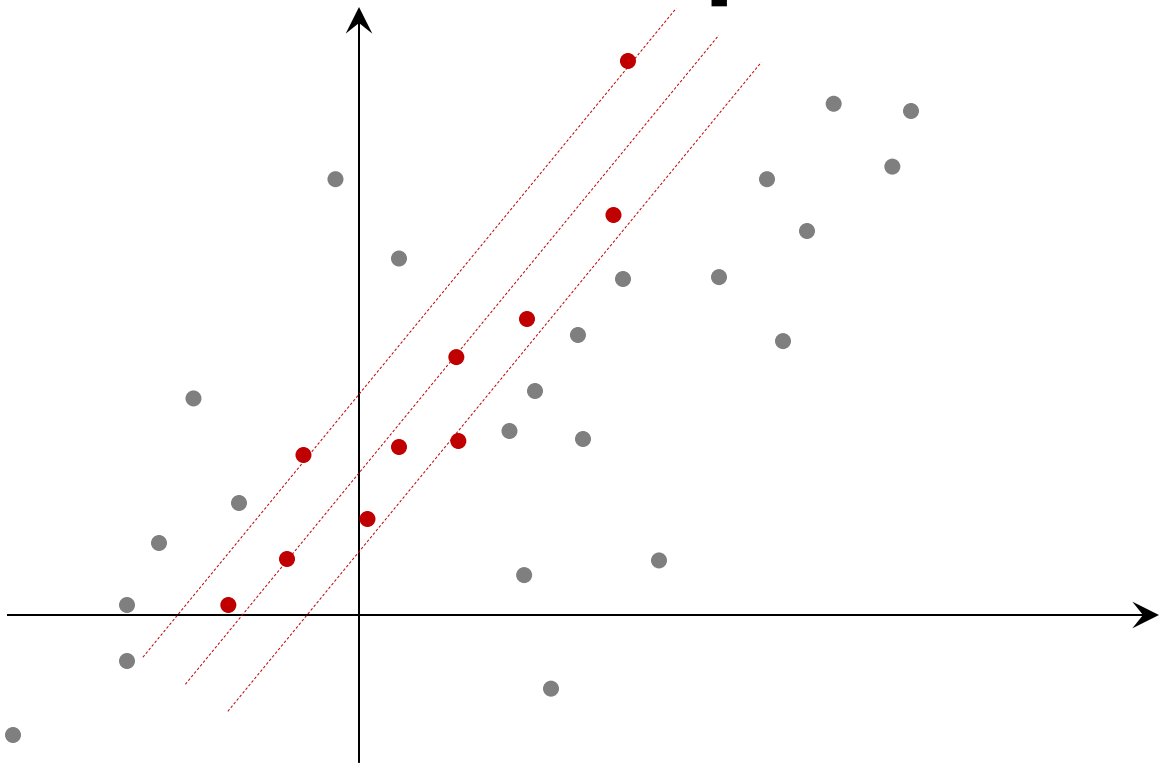
Random sampling

Model building

Thresholding

Inlier counting

RANSAC: Random Sample Consensus



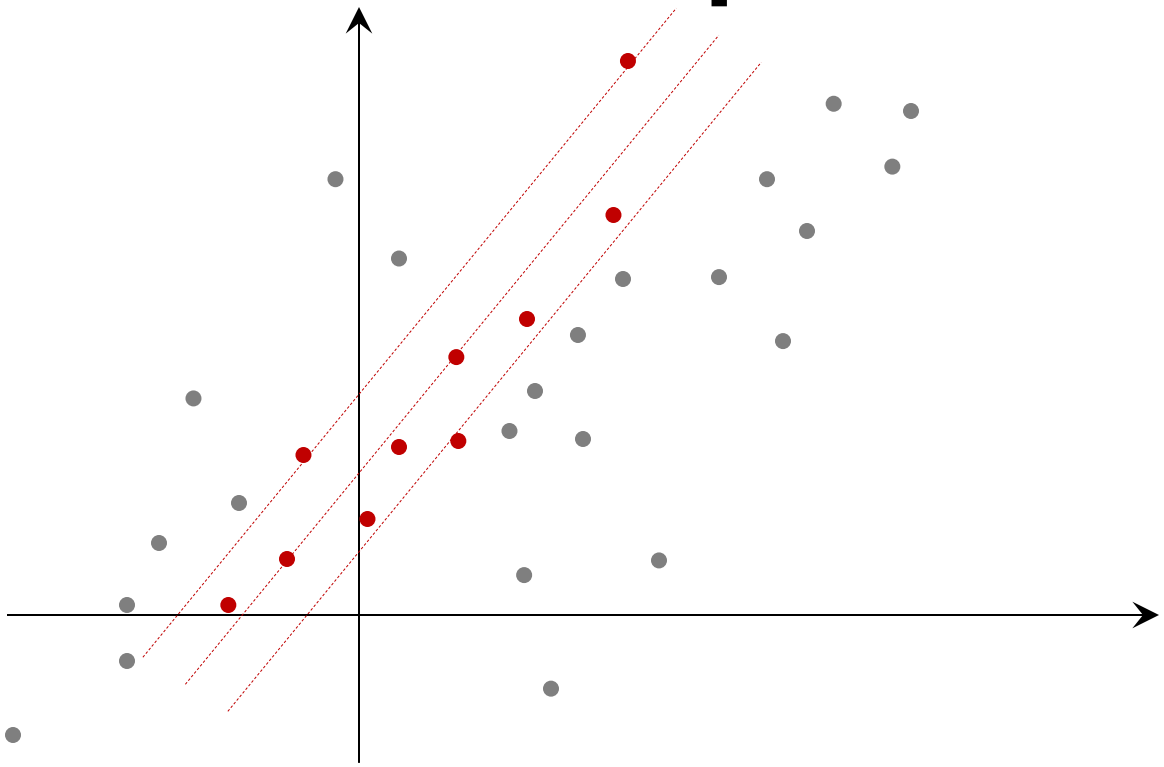
Random sampling

Model building

Thresholding

Inlier counting

RANSAC: Random Sample Consensus



Random sampling

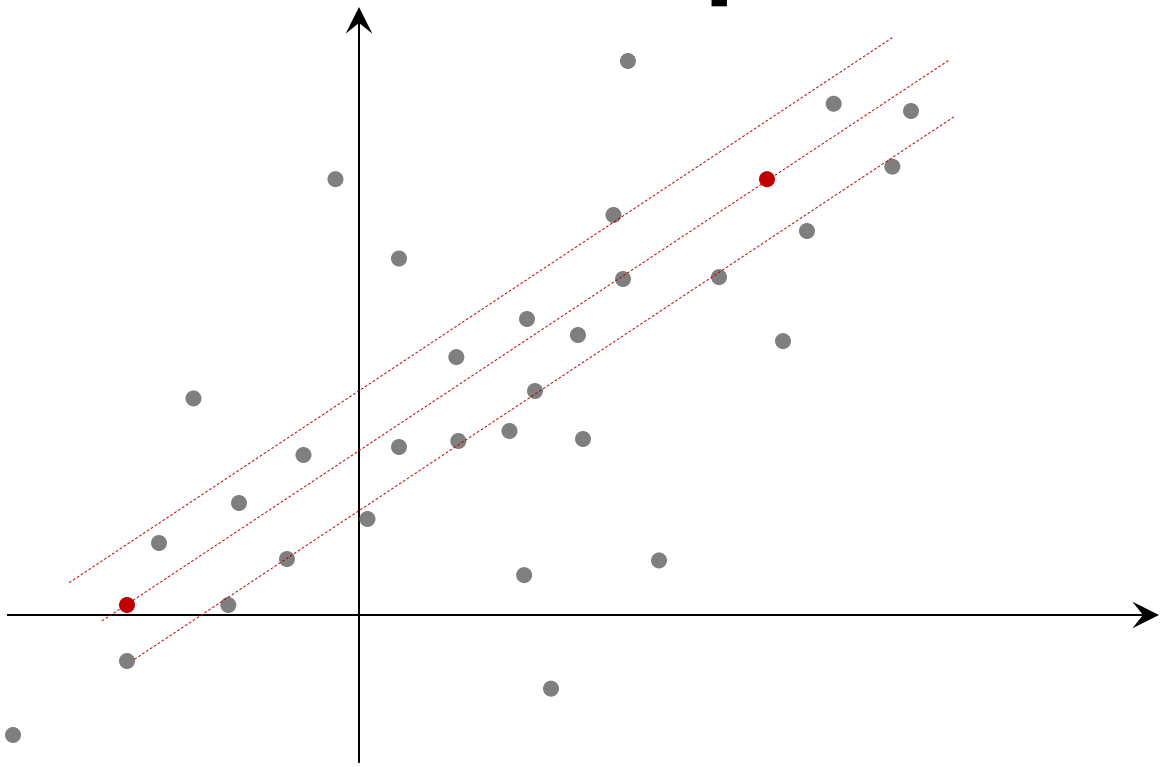
Model building

Thresholding

Inlier counting

# of inliers: 10

RANSAC: Random Sample Consensus



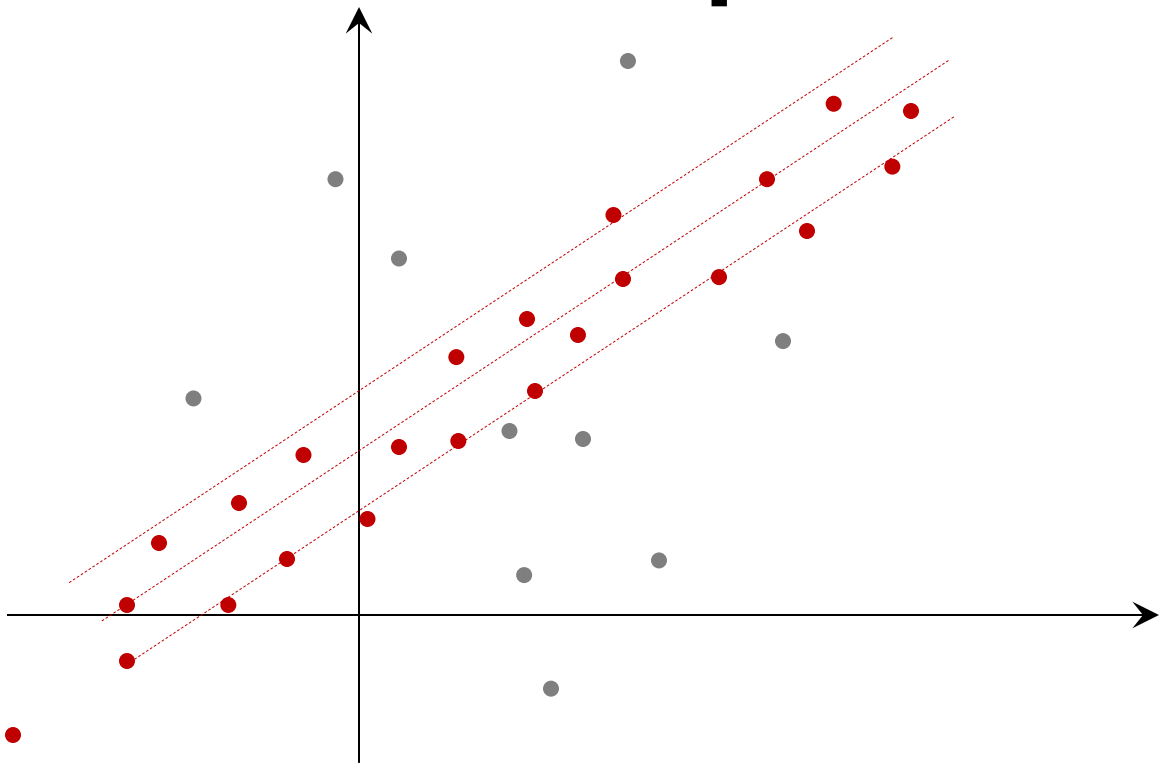
Random sampling

Model building

Thresholding

Inlier counting

RANSAC: Random Sample Consensus



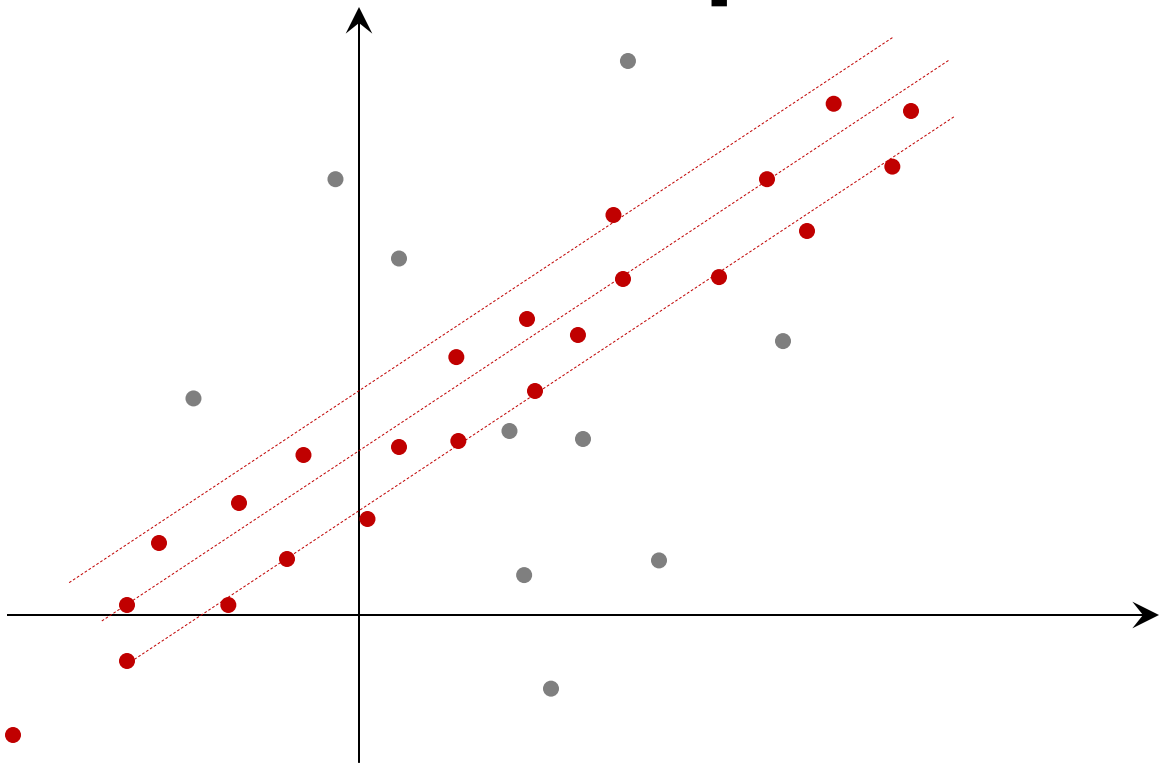
Random sampling

Model building

Thresholding

Inlier counting

RANSAC: Random Sample Consensus



Random sampling

Model building

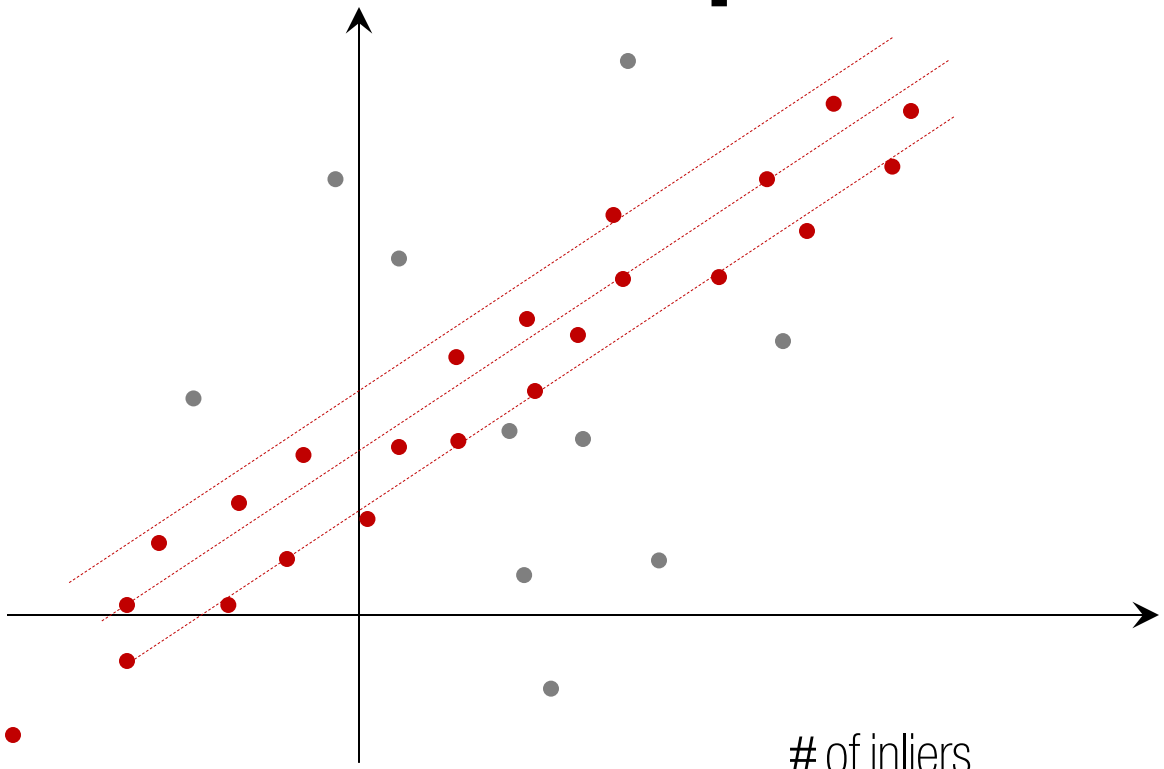
Thresholding

Inlier counting

# of inliers: 23

Maximum number of inliers

RANSAC: Random Sample Consensus



Random sampling

Model building

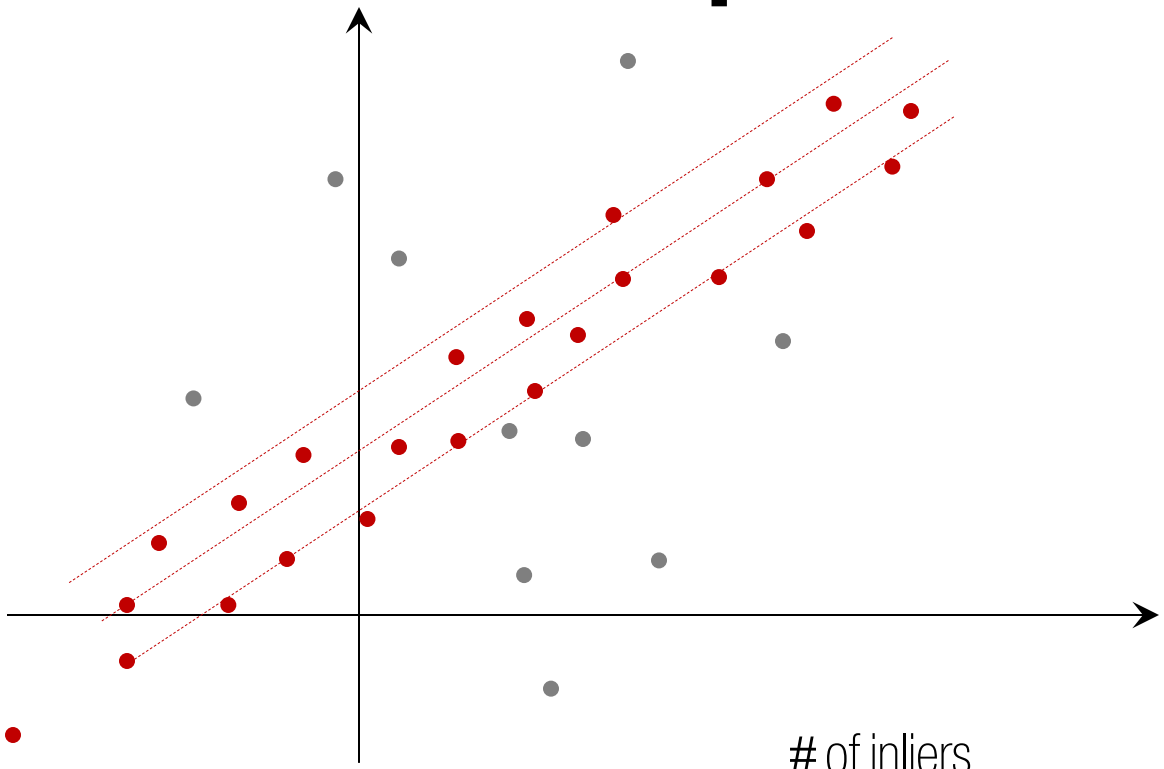
Thresholding

Inlier counting

Probability of choosing an inlier:

$$w = \frac{\#\text{ of inliers}}{\#\text{ of samples}}$$

RANSAC: Random Sample Consensus



Random sampling

Model building

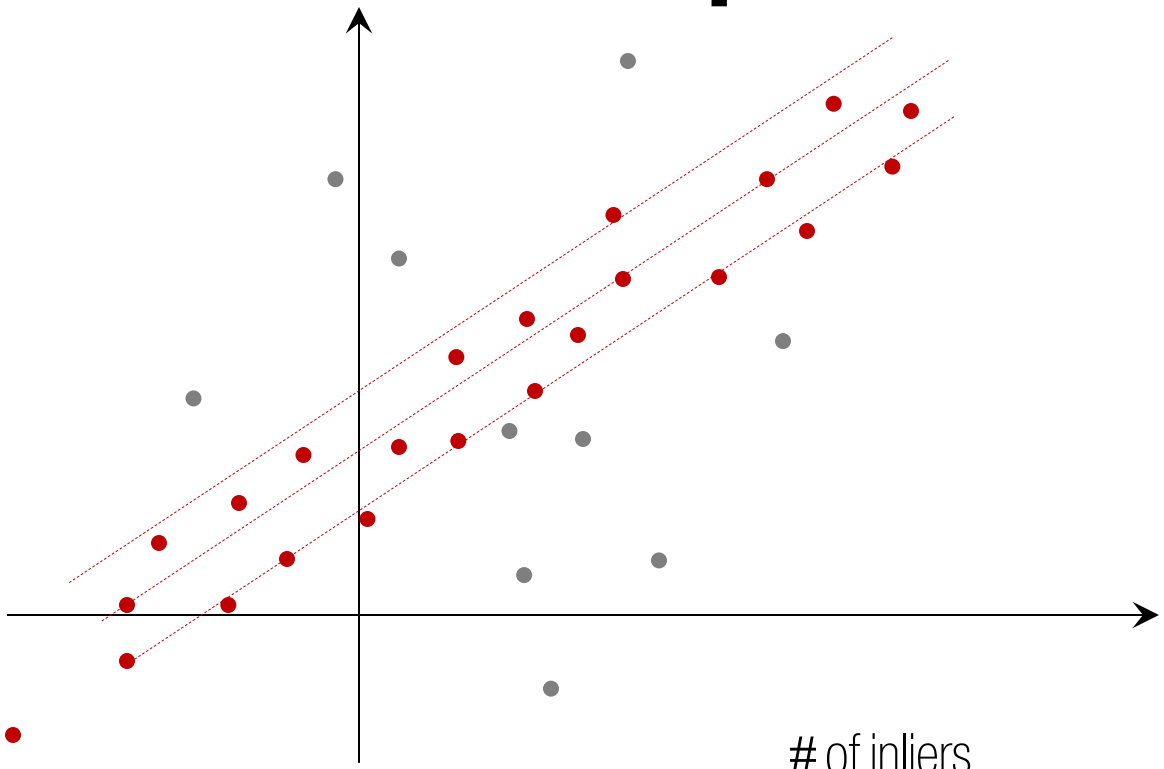
Thresholding

Inlier counting

Probability of choosing an inlier: $w = \frac{\#\text{ of inliers}}{\#\text{ of samples}}$

Probability of building a correct model: w^n where n is the number of samples to build a model.

RANSAC: Random Sample Consensus



Random sampling

Model building

Thresholding

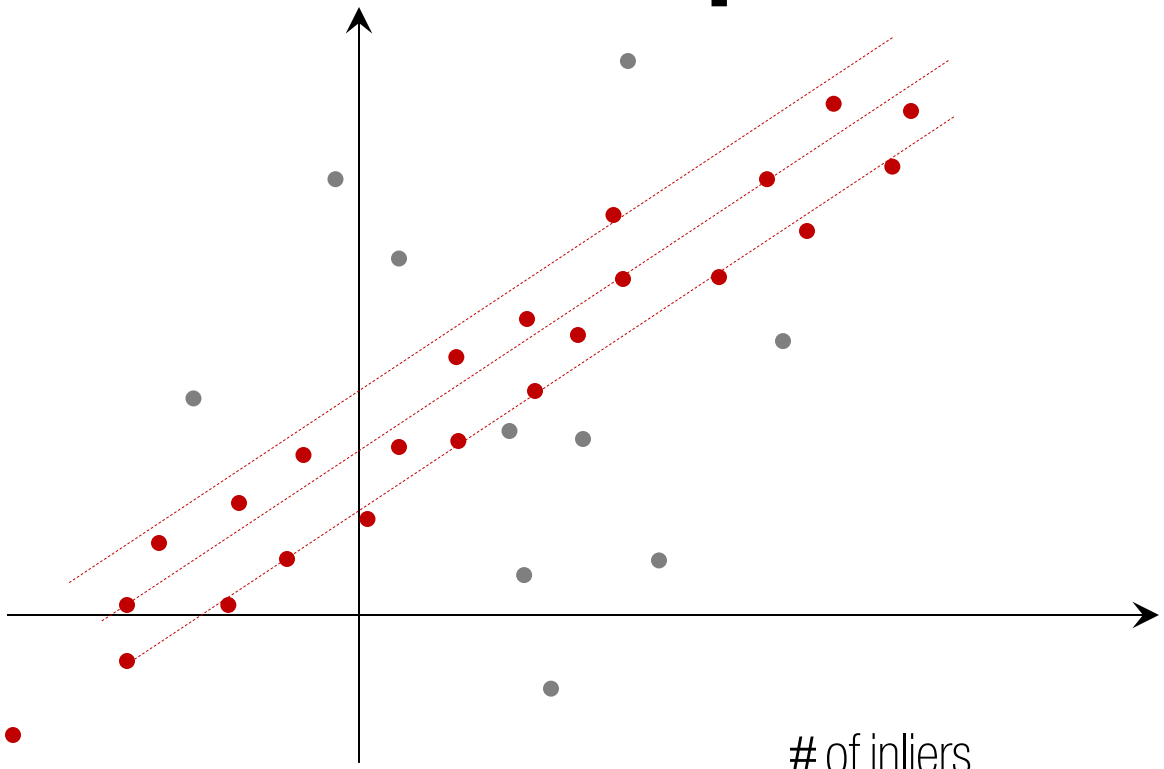
Inlier counting

Probability of choosing an inlier: $w = \frac{\#\text{ of inliers}}{\#\text{ of samples}}$

Probability of building a correct model: w^n where n is the number of samples to build a model.

Probability of not building a correct model during k iterations: $\left(1 - w^n\right)^k$

RANSAC: Random Sample Consensus



Random sampling

Model building

Thresholding

Inlier counting

Probability of choosing an inlier: $w = \frac{\#\text{ of inliers}}{\#\text{ of samples}}$

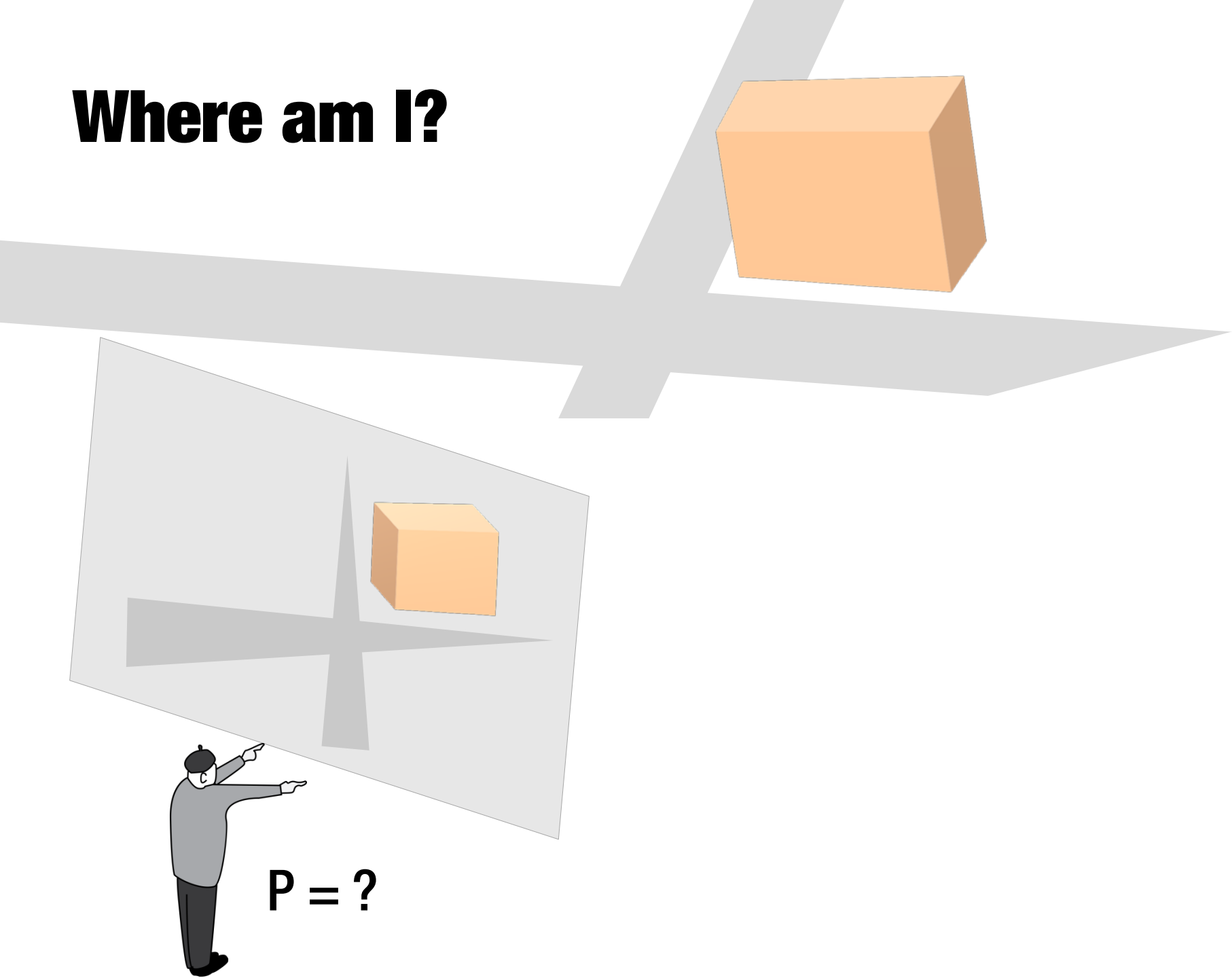
Probability of building a correct model: w^n where n is the number of samples to build a model.

Probability of not building a correct model during k iterations: $\left(1-w^n\right)^k$

$\left(1-w^n\right)^k = 1-p$ where p is desired RANSAC success rate.

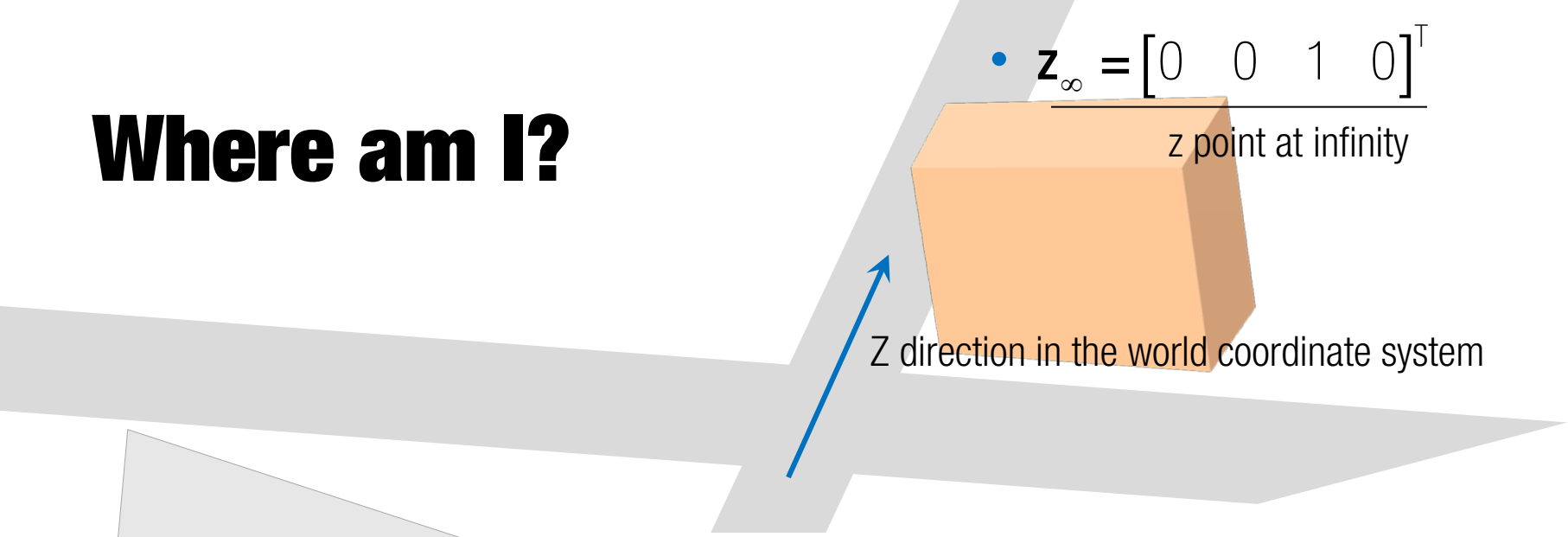
$$k = \frac{\log\left(1-p\right)}{\log\left(1-w^n\right)}$$

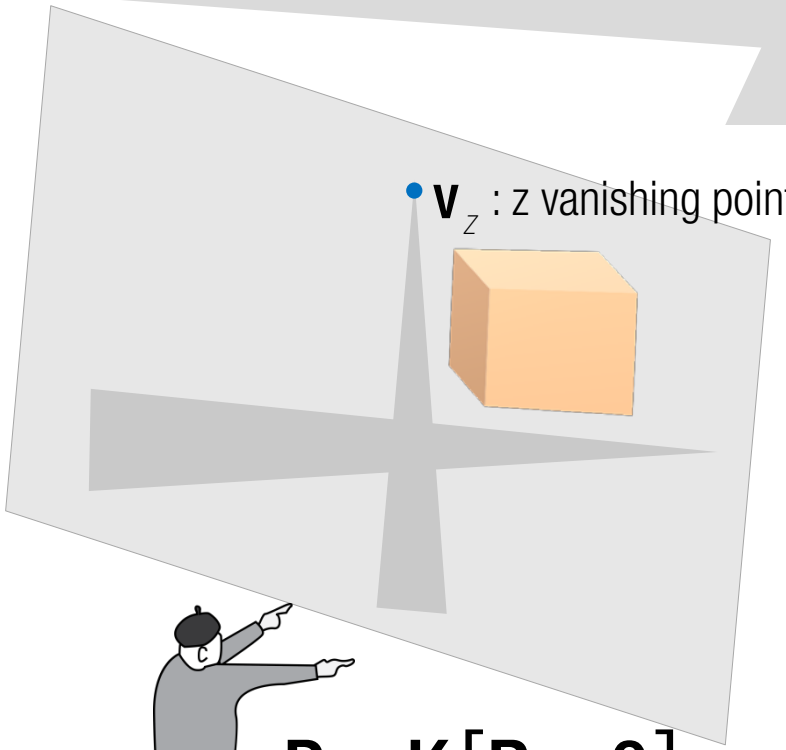
Where am I?



P = ?

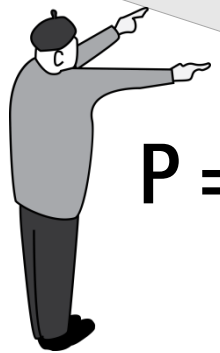
Where am I?





$$Z\mathbf{v}_z = \mathbf{K}\mathbf{r}_3$$

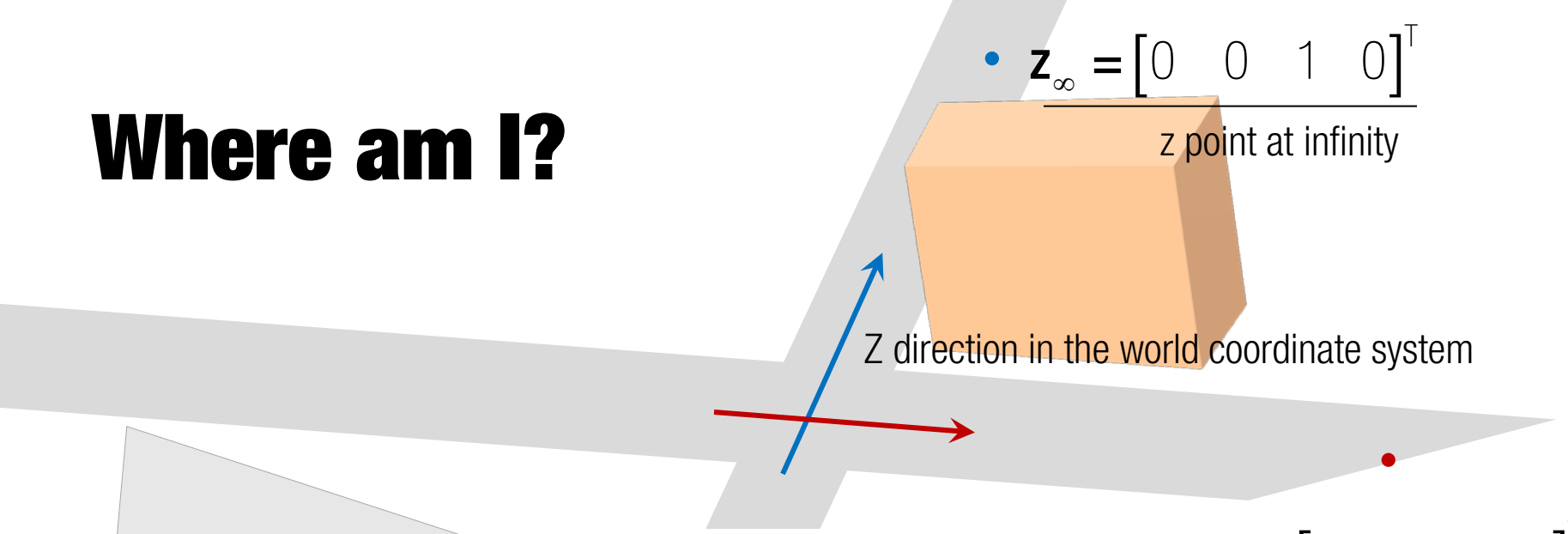
$$\mathbf{r}_3 = \mathbf{K}^{-1}\mathbf{v}_z \,/\, \left\|\mathbf{K}^{-1}\mathbf{v}_z\right\|$$

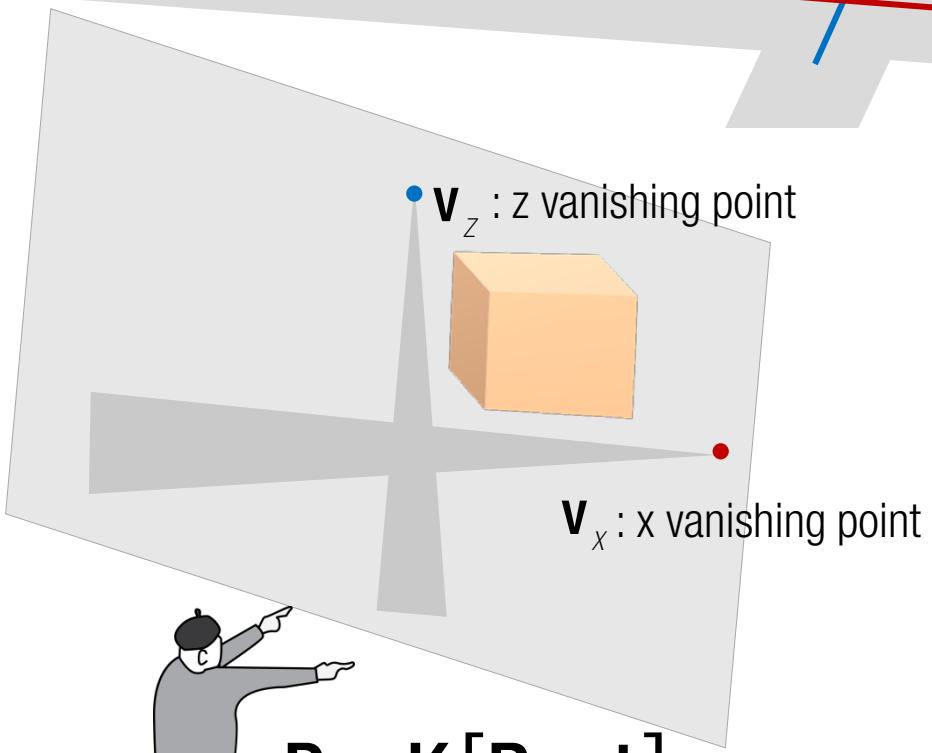


$$P = K[R \quad 0]$$

?

Where am I?

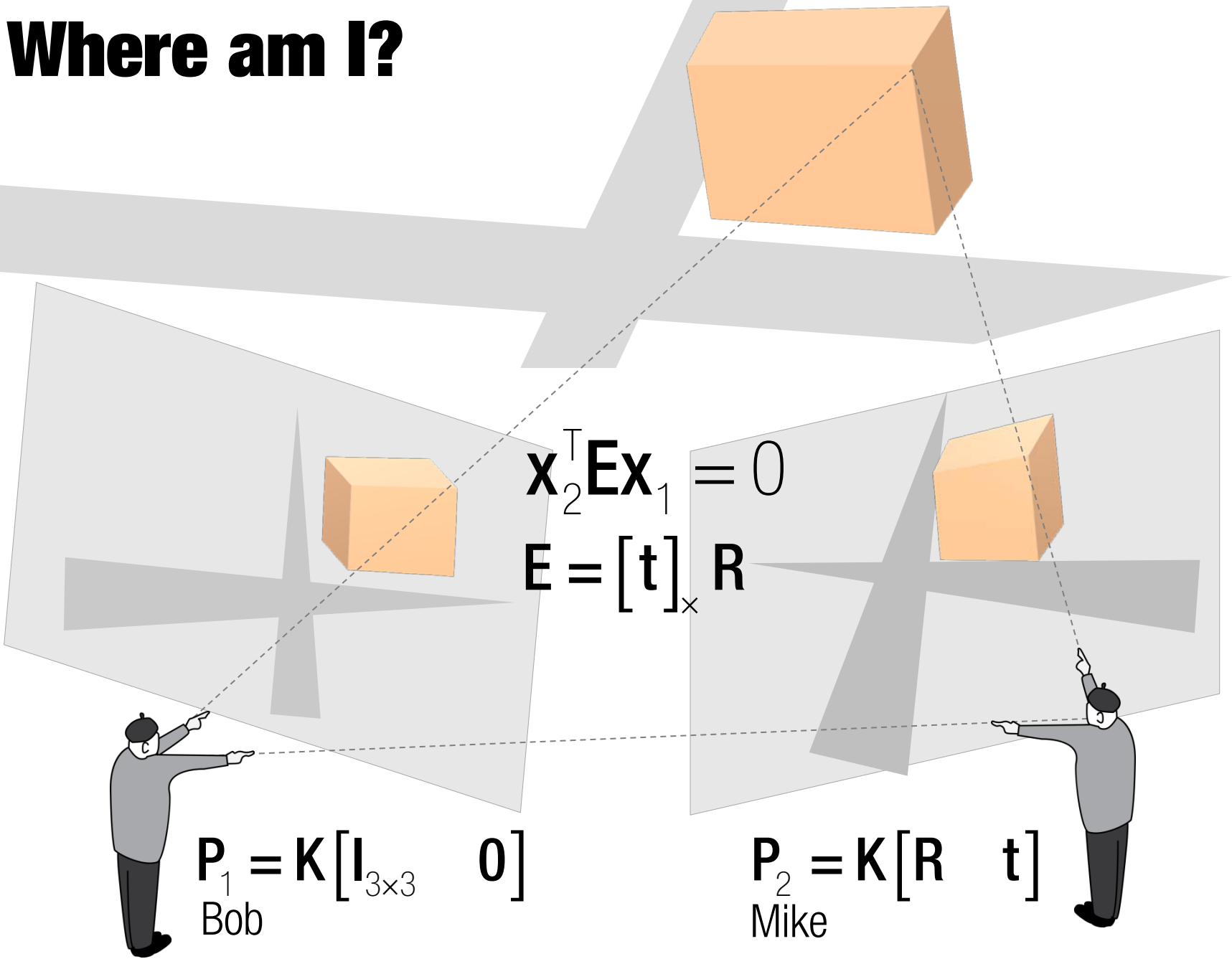




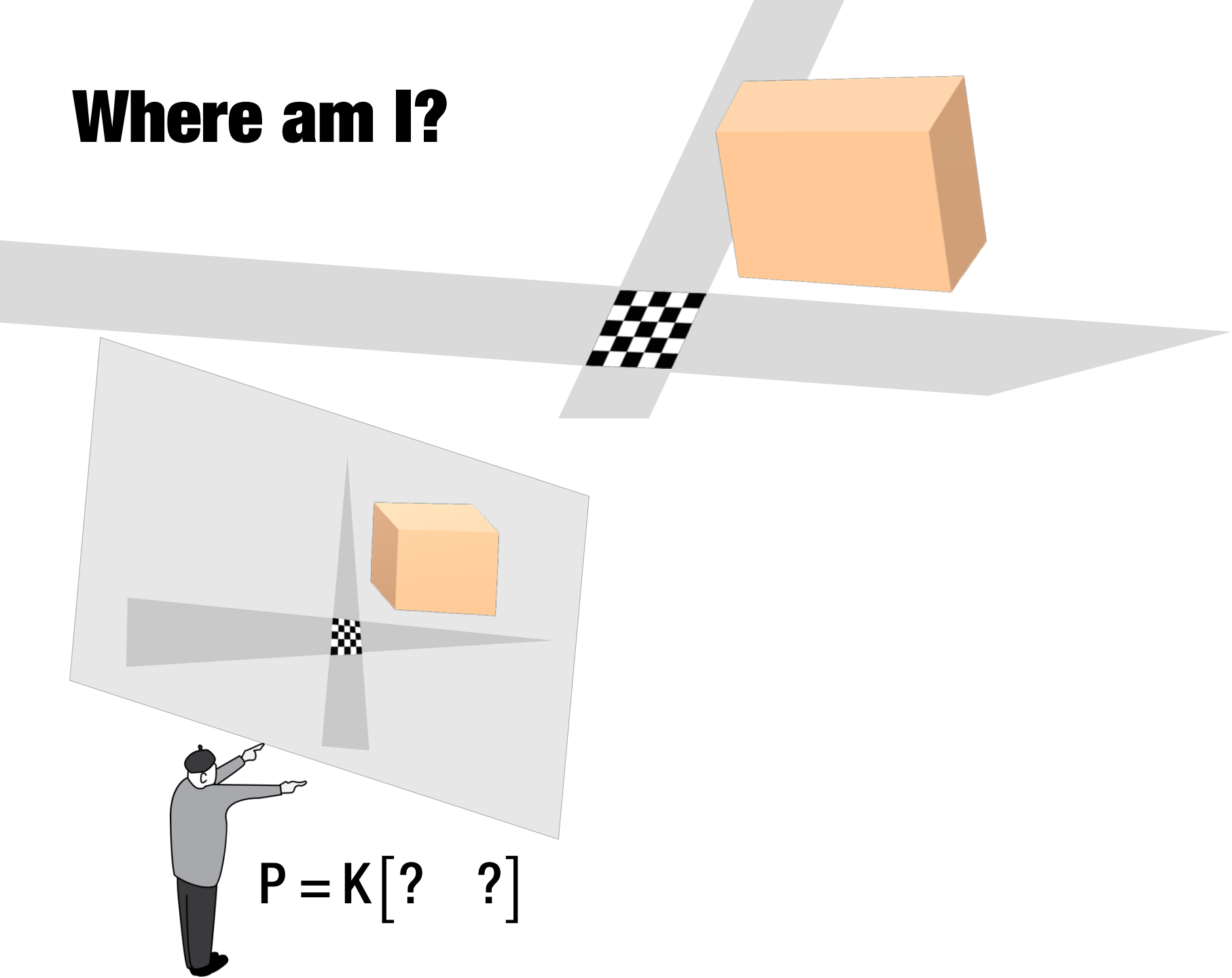
$$\mathbf{r}_3 = \mathbf{K}^{-1}\mathbf{v}_z \,/\, \left\|\mathbf{K}^{-1}\mathbf{v}_z\right\|$$
$$\mathbf{r}_1 = \mathbf{K}^{-1}\mathbf{v}_x \,/\, \left\|\mathbf{K}^{-1}\mathbf{v}_x\right\|$$
$$\mathbf{r}_2 = \mathbf{r}_3 \times \mathbf{r}_1$$

$$\mathbf{P} = \mathbf{K}[\mathbf{R} \quad \mathbf{t}]$$
$$?$$

Where am I?

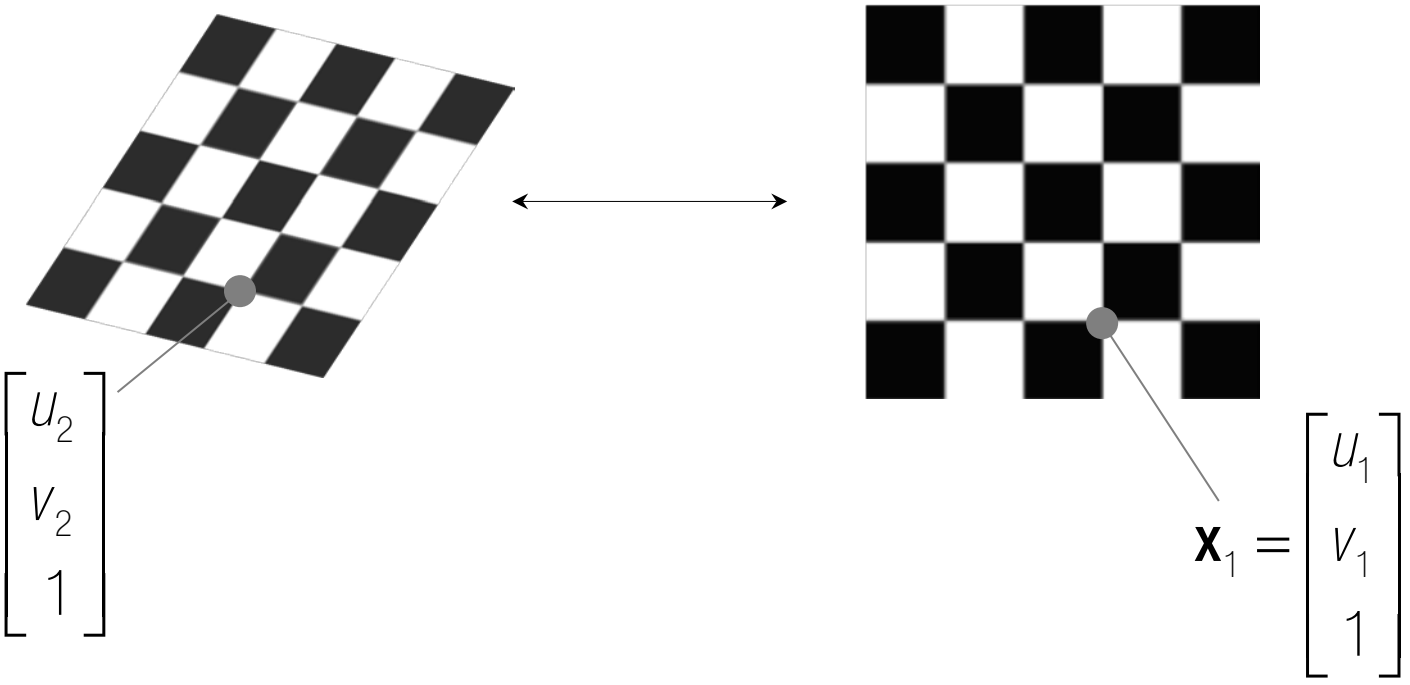


Where am I?

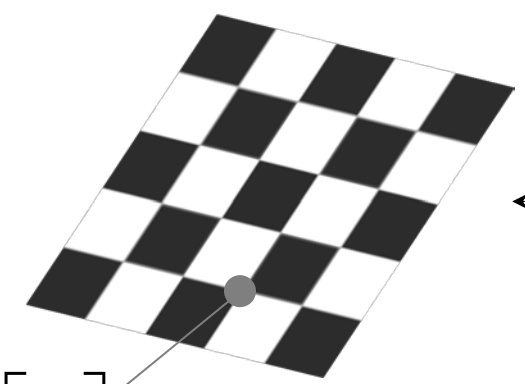


$$P = K\begin{bmatrix} ? & ? \end{bmatrix}$$

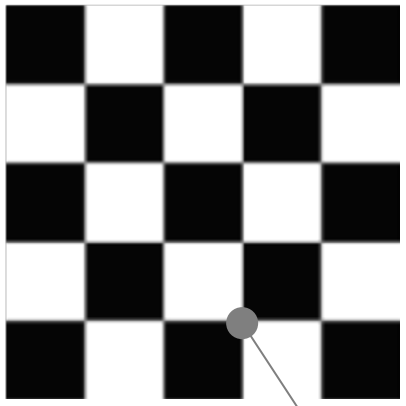
Homography Linear Estimation



Homography Linear Estimation



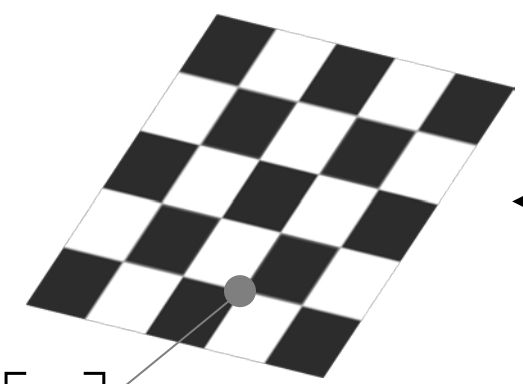
$$\mathbf{x}_2 = \mathbf{H}\mathbf{x}_1$$



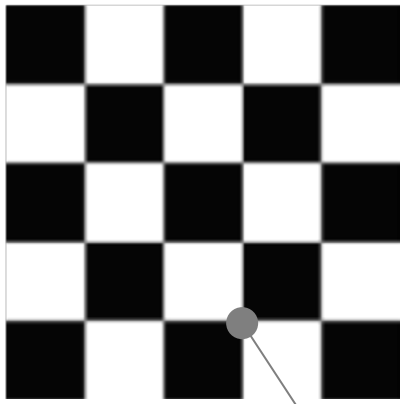
$$\mathbf{x}_2 = \begin{bmatrix} u_2 \\ v_2 \\ 1 \end{bmatrix}$$

$$\mathbf{x}_1 = \begin{bmatrix} u_1 \\ v_1 \\ 1 \end{bmatrix}$$

Homography Linear Estimation



$$\mathbf{x}_2 = \mathbf{H}\mathbf{x}_1$$

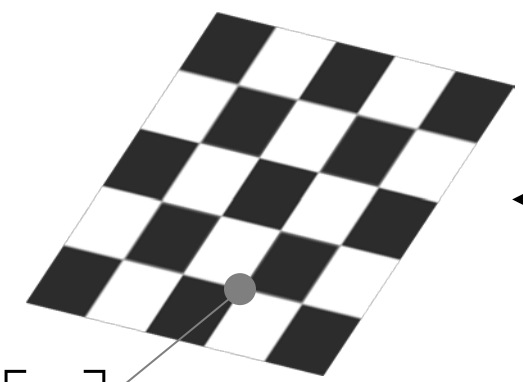


$$\mathbf{x}_2 = \begin{bmatrix} u_2 \\ v_2 \\ 1 \end{bmatrix}$$

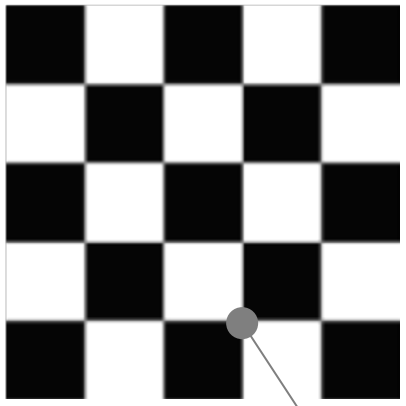
$$\mathbf{H} = \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{bmatrix}$$

$$\mathbf{x}_1 = \begin{bmatrix} u_1 \\ v_1 \\ 1 \end{bmatrix}$$

Homography Linear Estimation



$$\mathbf{x}_2 = \mathbf{H}\mathbf{x}_1$$



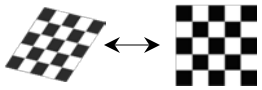
$$\mathbf{x}_2 = \begin{bmatrix} u_2 \\ v_2 \\ 1 \end{bmatrix}$$

$$\mathbf{H} = \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{bmatrix}$$

$$\mathbf{x}_1 = \begin{bmatrix} u_1 \\ v_1 \\ 1 \end{bmatrix}$$

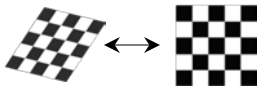
How to estimate $\mathbf{H}$?

Homography Linear Estimation



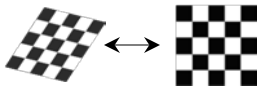
$$\mathbf{x}_2 = \mathbf{H}\mathbf{x}_1$$

Homography Linear Estimation



$$\mathbf{x}_2 = \mathbf{H}\mathbf{x}_1 \quad \longrightarrow \quad [\mathbf{x}_2]_\times \mathbf{H}\mathbf{x}_1 = \mathbf{0}$$

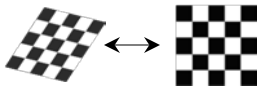
Homography Linear Estimation



$$\mathbf{x}_2 = \mathbf{H}\mathbf{x}_1 \longrightarrow [\mathbf{x}_2]_\times \mathbf{H}\mathbf{x}_1 = \mathbf{0}$$

$$\begin{bmatrix} u_2 \\ v_2 \\ 1 \end{bmatrix}_\times \begin{bmatrix} \mathbf{h}_1 \\ \mathbf{h}_2 \\ \mathbf{h}_3 \end{bmatrix} \mathbf{x}_1$$

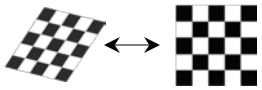
Homography Linear Estimation



$$\mathbf{x}_2 = \mathbf{H}\mathbf{x}_1 \quad \longrightarrow \quad [\mathbf{x}_2]_\times \mathbf{H}\mathbf{x}_1 = \mathbf{0}$$

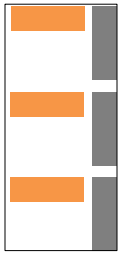
$$\begin{bmatrix} u_2 \\ v_2 \\ 1 \end{bmatrix}_\times \begin{bmatrix} \mathbf{h}_1 \\ \mathbf{h}_2 \\ \mathbf{h}_3 \end{bmatrix} \mathbf{x}_1 = \begin{bmatrix} u_2 \\ v_2 \\ 1 \end{bmatrix}_\times \begin{bmatrix} \mathbf{h}_1\mathbf{x}_1 \\ \mathbf{h}_2\mathbf{x}_1 \\ \mathbf{h}_3\mathbf{x}_1 \end{bmatrix}$$

Homography Linear Estimation

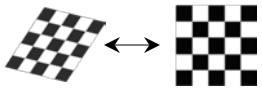


$$\mathbf{x}_2 = \mathbf{H}\mathbf{x}_1 \longrightarrow [\mathbf{x}_2]_\times \mathbf{H}\mathbf{x}_1 = \mathbf{0} \longrightarrow$$

$$\begin{bmatrix} u_2 \\ v_2 \\ 1 \end{bmatrix}_\times \begin{bmatrix} \mathbf{h}_1 \\ \mathbf{h}_2 \\ \mathbf{h}_3 \end{bmatrix} \mathbf{x}_1 = \begin{bmatrix} u_2 \\ v_2 \\ 1 \end{bmatrix}_\times \begin{bmatrix} \mathbf{h}_1\mathbf{x}_1 \\ \mathbf{h}_2\mathbf{x}_1 \\ \mathbf{h}_3\mathbf{x}_1 \end{bmatrix} = \begin{bmatrix} u_2 \\ v_2 \\ 1 \end{bmatrix}_\times$$

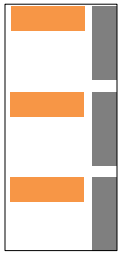


Homography Linear Estimation



$$\mathbf{x}_2 = \mathbf{H}\mathbf{x}_1 \longrightarrow [\mathbf{x}_2]_\times \mathbf{H}\mathbf{x}_1 = \mathbf{0} \longrightarrow$$

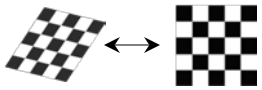
$$\begin{bmatrix} u_2 \\ v_2 \\ 1 \end{bmatrix}_\times \begin{bmatrix} \mathbf{h}_1 \\ \mathbf{h}_2 \\ \mathbf{h}_3 \end{bmatrix} \mathbf{x}_1 = \begin{bmatrix} u_2 \\ v_2 \\ 1 \end{bmatrix}_\times \begin{bmatrix} \mathbf{h}_1\mathbf{x}_1 \\ \mathbf{h}_2\mathbf{x}_1 \\ \mathbf{h}_3\mathbf{x}_1 \end{bmatrix} = \begin{bmatrix} u_2 \\ v_2 \\ 1 \end{bmatrix}_\times$$



$$=$$

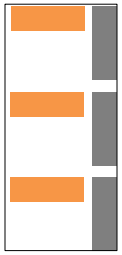
$$\begin{bmatrix} u_2 \\ v_2 \\ 1 \end{bmatrix}_\times$$

Homography Linear Estimation



$$\mathbf{x}_2 = \mathbf{H}\mathbf{x}_1 \longrightarrow [\mathbf{x}_2]_\times \mathbf{H}\mathbf{x}_1 = \mathbf{0} \longrightarrow$$

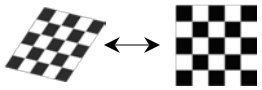
$$\begin{bmatrix} u_2 \\ v_2 \\ 1 \end{bmatrix}_\times \begin{bmatrix} \mathbf{h}_1 \\ \mathbf{h}_2 \\ \mathbf{h}_3 \end{bmatrix} \mathbf{x}_1 = \begin{bmatrix} u_2 \\ v_2 \\ 1 \end{bmatrix}_\times \begin{bmatrix} \mathbf{h}_1\mathbf{x}_1 \\ \mathbf{h}_2\mathbf{x}_1 \\ \mathbf{h}_3\mathbf{x}_1 \end{bmatrix} = \begin{bmatrix} u_2 \\ v_2 \\ 1 \end{bmatrix}_\times$$



$$= \begin{bmatrix} u_2 \\ v_2 \\ 1 \end{bmatrix}_\times$$

$$= \begin{bmatrix} u_2 \\ v_2 \\ 1 \end{bmatrix}_\times \begin{bmatrix} \mathbf{x}_1^\mathsf{T}\mathbf{h}_3^\mathsf{T} \\ \mathbf{x}_2^\mathsf{T}\mathbf{h}_3^\mathsf{T} \\ \mathbf{x}_3^\mathsf{T}\mathbf{h}_3^\mathsf{T} \end{bmatrix}$$

Homography Linear Estimation

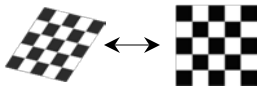


$$\mathbf{x}_2 = \mathbf{H}\mathbf{x}_1 \longrightarrow [\mathbf{x}_2]_\times \mathbf{H}\mathbf{x}_1 = \mathbf{0} \longrightarrow$$

$$\begin{bmatrix} u_2 \\ v_2 \\ 1 \end{bmatrix}_\times \begin{bmatrix} \mathbf{h}_1 \\ \mathbf{h}_2 \\ \mathbf{h}_3 \end{bmatrix} \mathbf{x}_1 = \begin{bmatrix} u_2 \\ v_2 \\ 1 \end{bmatrix}_\times \begin{bmatrix} \mathbf{h}_1\mathbf{x}_1 \\ \mathbf{h}_2\mathbf{x}_1 \\ \mathbf{h}_3\mathbf{x}_1 \end{bmatrix} = \begin{bmatrix} u_2 \\ v_2 \\ 1 \end{bmatrix}_\times \begin{bmatrix} \\ \\ \end{bmatrix} = \begin{bmatrix} u_2 \\ v_2 \\ 1 \end{bmatrix}_\times \begin{bmatrix} \\ \\ \end{bmatrix} = \begin{bmatrix} u_2 \\ v_2 \\ 1 \end{bmatrix}_\times \begin{bmatrix} \mathbf{x}_1^\top \mathbf{h}_1^\top \\ \mathbf{x}_1^\top \mathbf{h}_2^\top \\ \mathbf{x}_1^\top \mathbf{h}_3^\top \end{bmatrix}$$

$$= \begin{bmatrix} 0 & -1 & v_2 \\ 1 & 0 & -u_2 \\ -v_2 & u_2 & 0 \end{bmatrix} \begin{bmatrix} \mathbf{x}_1^\top & \mathbf{0}_{1\times 3} & \mathbf{0}_{1\times 3} \\ \mathbf{0}_{1\times 3} & \mathbf{x}_1^\top & \mathbf{0}_{1\times 3} \\ \mathbf{0}_{1\times 3} & \mathbf{0}_{1\times 3} & \mathbf{x}_1^\top \end{bmatrix} \begin{bmatrix} \mathbf{h}_1^\top \\ \mathbf{h}_2^\top \\ \mathbf{h}_3^\top \end{bmatrix}$$

Homography Linear Estimation



$$\mathbf{x}_2 = \mathbf{H}\mathbf{x}_1 \longrightarrow [\mathbf{x}_2]_\times \mathbf{H}\mathbf{x}_1 = \mathbf{0} \longrightarrow$$

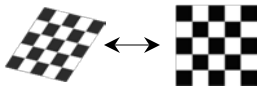
$$\begin{bmatrix} u_2 \\ v_2 \\ 1 \end{bmatrix}_\times \begin{bmatrix} \mathbf{h}_1 \\ \mathbf{h}_2 \\ \mathbf{h}_3 \end{bmatrix} \mathbf{x}_1 = \begin{bmatrix} u_2 \\ v_2 \\ 1 \end{bmatrix}_\times \begin{bmatrix} \mathbf{h}_1\mathbf{x}_1 \\ \mathbf{h}_2\mathbf{x}_1 \\ \mathbf{h}_3\mathbf{x}_1 \end{bmatrix} = \begin{bmatrix} u_2 \\ v_2 \\ 1 \end{bmatrix}_\times \begin{bmatrix} \\ \\ \end{bmatrix} = \begin{bmatrix} u_2 \\ v_2 \\ 1 \end{bmatrix}_\times \begin{bmatrix} \\ \\ \end{bmatrix} = \begin{bmatrix} u_2 \\ v_2 \\ 1 \end{bmatrix}_\times \begin{bmatrix} \mathbf{x}_1^\top\mathbf{h}_3^\top \\ \mathbf{x}_2^\top\mathbf{h}_3^\top \\ \mathbf{x}_3^\top\mathbf{h}_3^\top \end{bmatrix}$$

$$= \begin{bmatrix} 0 & -1 & v_2 \\ 1 & 0 & -u_2 \\ -v_2 & u_2 & 0 \end{bmatrix} \begin{bmatrix} \mathbf{x}_1^\top & \mathbf{0}_{1\times3} & \mathbf{0}_{1\times3} \\ \mathbf{0}_{1\times3} & \mathbf{x}_1^\top & \mathbf{0}_{1\times3} \\ \mathbf{0}_{1\times3} & \mathbf{0}_{1\times3} & \mathbf{x}_1^\top \end{bmatrix} \begin{bmatrix} \mathbf{h}_1^\top \\ \mathbf{h}_2^\top \\ \mathbf{h}_3^\top \end{bmatrix}$$

$$= \begin{bmatrix} \mathbf{0}_{1\times3} & -\mathbf{x}_1^\top & v_2\mathbf{x}_1^\top \\ \mathbf{x}_1^\top & \mathbf{0}_{1\times3} & -u_2\mathbf{x}_1^\top \\ -v_2\mathbf{x}_1^\top & u_2\mathbf{x}_1^\top & \mathbf{0}_{1\times3} \end{bmatrix} \begin{bmatrix} \mathbf{h}_1^\top \\ \mathbf{h}_2^\top \\ \mathbf{h}_3^\top \end{bmatrix} = \mathbf{0} \longrightarrow \mathbf{A}\mathbf{x} = \mathbf{0}$$

3 x 9

Homography Linear Estimation



$$\mathbf{x}_2 = \mathbf{H}\mathbf{x}_1 \longrightarrow [\mathbf{x}_2]_\times \mathbf{H}\mathbf{x}_1 = \mathbf{0}$$

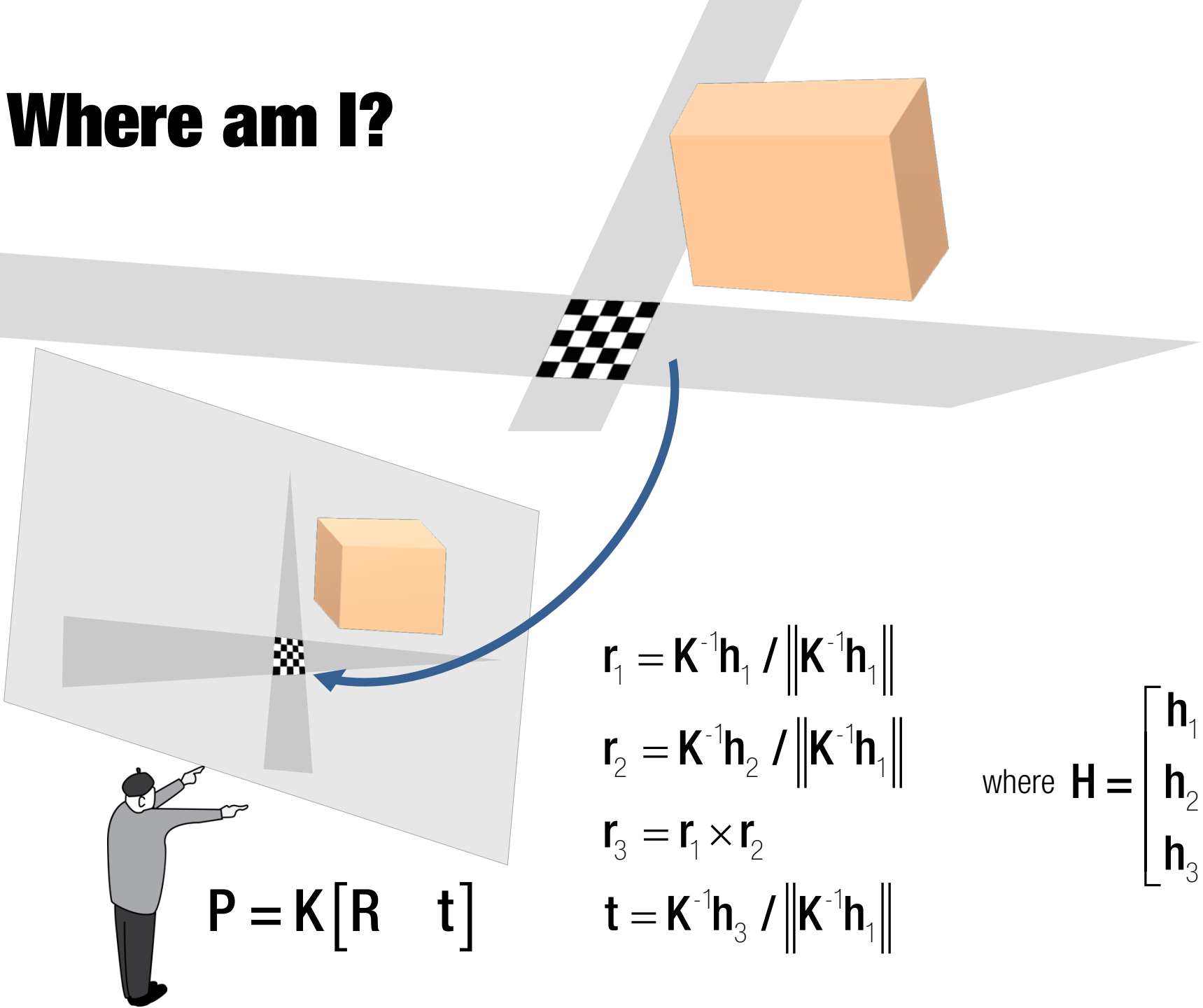
$$\begin{bmatrix} \mathbf{0}_{1\times3} & -\mathbf{x}_1^\top & v_2\mathbf{x}_1^\top \\ \mathbf{x}_1^\top & \mathbf{0}_{1\times3} & -u_2\mathbf{x}_1^\top \\ -v_2\mathbf{x}_1^\top & u_2\mathbf{x}_1^\top & \mathbf{0}_{1\times3} \end{bmatrix} \begin{bmatrix} \mathbf{h}_1^\top \\ \mathbf{h}_2^\top \\ \mathbf{h}_3^\top \end{bmatrix} = \mathbf{0} \longrightarrow \mathbf{A}\mathbf{x} = \mathbf{0}$$

3 x 9

rank( ) = 2 because $[\mathbf{x}_2]_\times$ is a rank 2 matrix.

Therefore, 4 point correspondences are required to estimate a homography.

Where am I?



$$P = K\begin{bmatrix} R & t \end{bmatrix}$$

$$r_1 = K^{-1}h_1 \,/\, \left\| K^{-1}h_1 \right\|$$

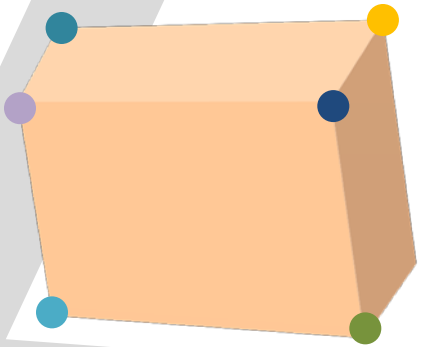
$$r_2 = K^{-1}h_2 \,/\, \left\| K^{-1}h_1 \right\|$$

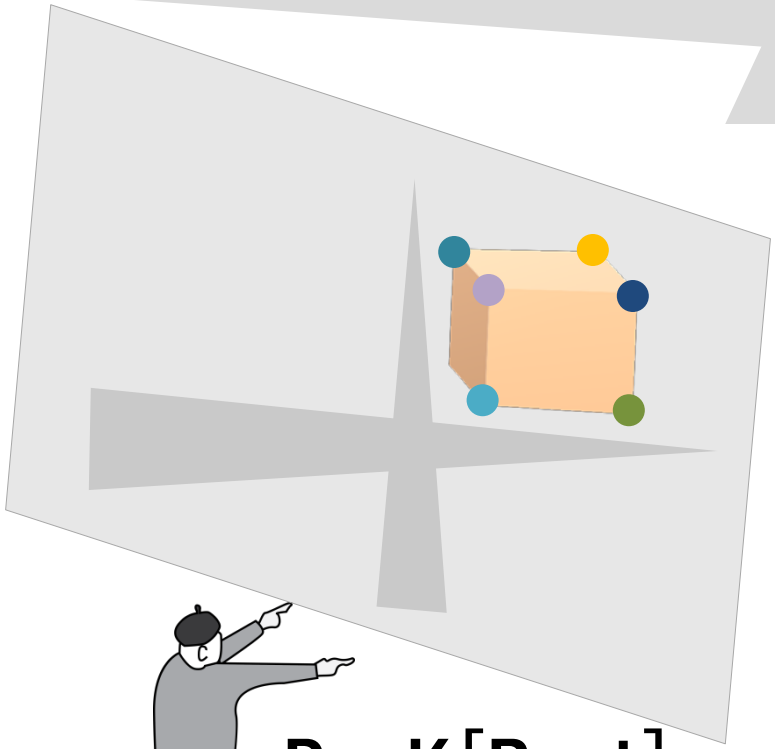
$$r_3 = r_1 \times r_2$$

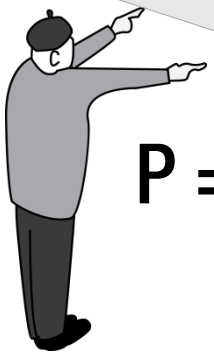
$$t = K^{-1}h_3 \,/\, \left\| K^{-1}h_1 \right\|$$

where $H = \begin{bmatrix} h_1 \\ h_2 \\ h_3 \end{bmatrix}$

Where am I?







$$P = K\left[\frac{R \quad t}{}\right]$$

?

Camera 3D Registration

Perspective-n-Point Algorithm

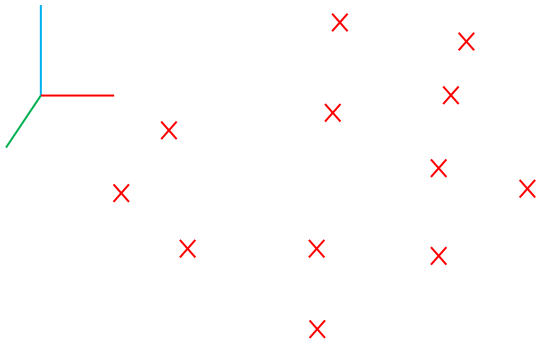
3D point cloud via triangulation

Where?

Perspective-n-Point

3D point cloud

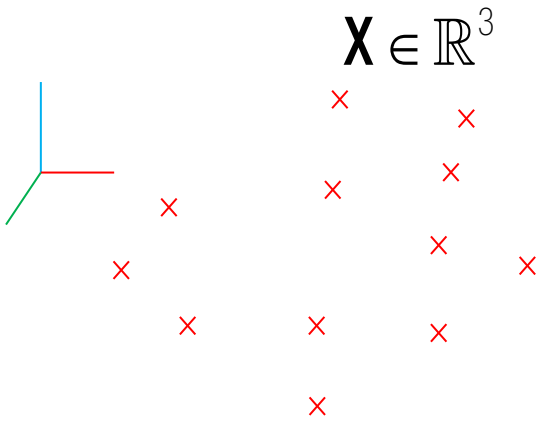
$$\mathbf{X} \in \mathbb{R}^3$$



Perspective-n-Point

2D image

3D point cloud



Perspective-n-Point

$\mathbf{x} \in \mathbb{R}^2$

3D point cloud

$\mathbf{X} \in \mathbb{R}^3$

2D image

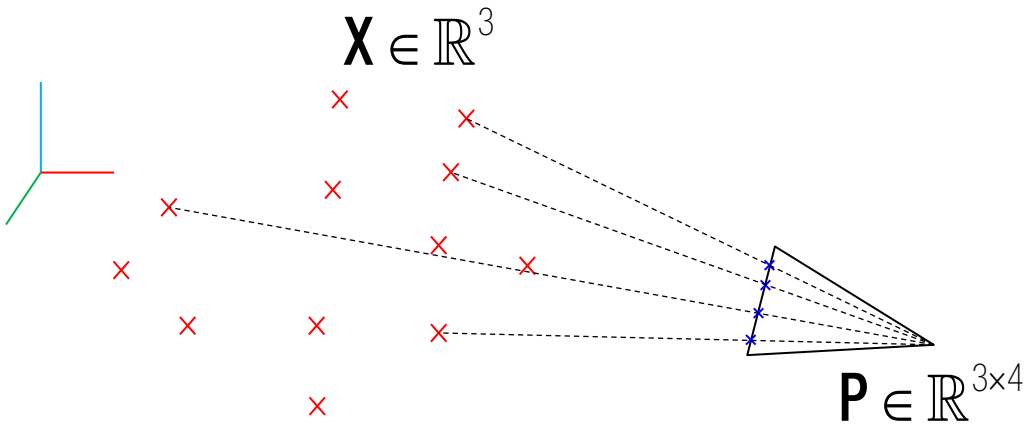
2D-3D correspondences

$\mathbf{x} \leftrightarrow \mathbf{X}$

Perspective-n-Point

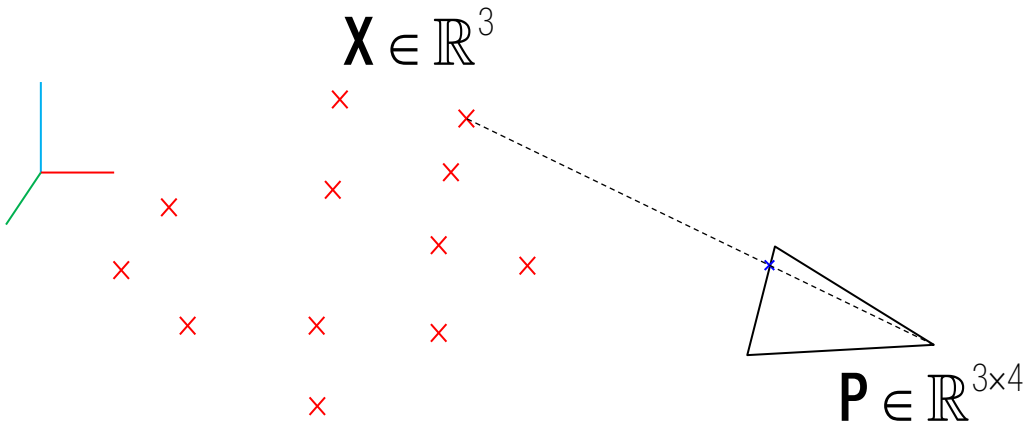
2D image

3D point cloud



Perspective-n-Point

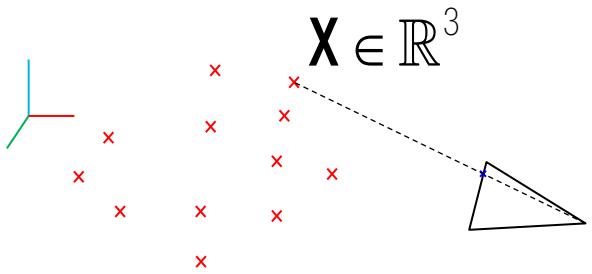
3D point cloud



2D image

$$\lambda \begin{bmatrix} \mathbf{x}_1 \\ 1 \end{bmatrix} = \mathbf{P} \begin{bmatrix} \mathbf{X}_1 \\ 1 \end{bmatrix} \longrightarrow \begin{bmatrix} \mathbf{x}_1 \\ 1 \end{bmatrix}_\times \mathbf{P} \begin{bmatrix} \mathbf{X}_1 \\ 1 \end{bmatrix} = \mathbf{0}$$

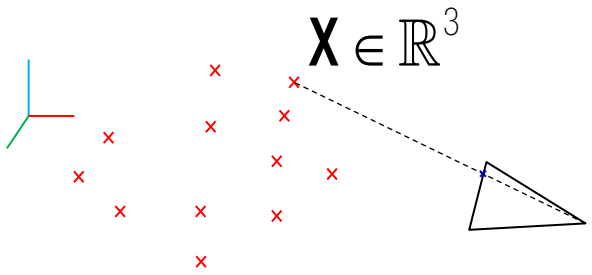
Perspective-n-Point



2D image

$$\begin{bmatrix} u \\ v \\ 1 \end{bmatrix}_{\times} \begin{bmatrix} \mathbf{P}_1 \\ \mathbf{P}_2 \\ \mathbf{P}_3 \end{bmatrix} \tilde{\mathbf{X}}$$

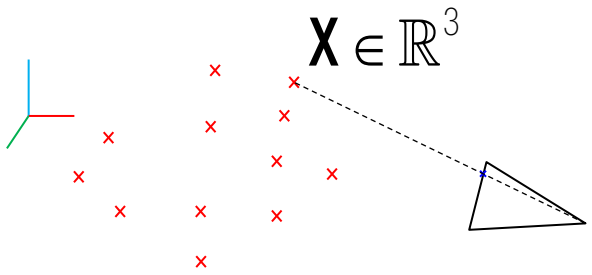
Perspective-n-Point



2D image

$$\begin{bmatrix} u \\ v \\ 1 \end{bmatrix}_\times \begin{bmatrix} \mathbf{P}_1 \\ \mathbf{P}_2 \\ \mathbf{P}_3 \end{bmatrix} \tilde{\mathbf{X}} = \begin{bmatrix} u \\ v \\ 1 \end{bmatrix}_\times \begin{bmatrix} \mathbf{P}_1 \tilde{\mathbf{X}} \\ \mathbf{P}_2 \tilde{\mathbf{X}} \\ \mathbf{P}_3 \tilde{\mathbf{X}} \end{bmatrix}$$

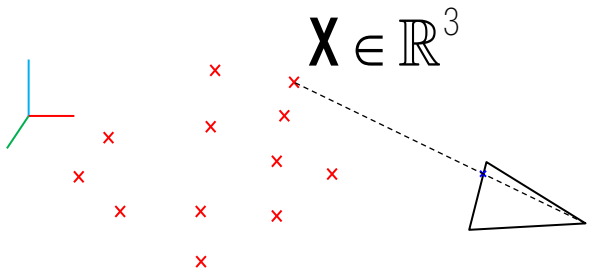
Perspective-n-Point



2D image

$$\begin{bmatrix} u \\ v \\ 1 \end{bmatrix}_\times \begin{bmatrix} \mathbf{P}_1 \\ \mathbf{P}_2 \\ \mathbf{P}_3 \end{bmatrix} \tilde{\mathbf{X}} = \begin{bmatrix} u \\ v \\ 1 \end{bmatrix}_\times \begin{bmatrix} \mathbf{P}_1 \tilde{\mathbf{X}} \\ \mathbf{P}_2 \tilde{\mathbf{X}} \\ \mathbf{P}_3 \tilde{\mathbf{X}} \end{bmatrix} = \begin{bmatrix} 0 & -1 & v \\ 1 & 0 & -u \\ -v & u & 0 \end{bmatrix} \begin{bmatrix} \tilde{\mathbf{X}}^\mathsf{T} & \mathbf{0}_{1\times4} & \mathbf{0}_{1\times4} \\ \mathbf{0}_{1\times4} & \tilde{\mathbf{X}}^\mathsf{T} & \mathbf{0}_{1\times4} \\ \mathbf{0}_{1\times4} & \mathbf{0}_{1\times4} & \tilde{\mathbf{X}}^\mathsf{T} \end{bmatrix} \begin{bmatrix} \mathbf{P}_1^\mathsf{T} \\ \mathbf{P}_2^\mathsf{T} \\ \mathbf{P}_3^\mathsf{T} \end{bmatrix} =$$

Perspective-n-Point



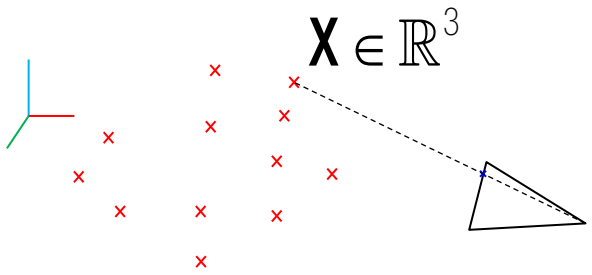
2D image

$$\begin{bmatrix} u \\ v \\ 1 \end{bmatrix}_\times \begin{bmatrix} \mathbf{P}_1 \\ \mathbf{P}_2 \\ \mathbf{P}_3 \end{bmatrix} \tilde{\mathbf{X}} = \begin{bmatrix} u \\ v \\ 1 \end{bmatrix}_\times \begin{bmatrix} \mathbf{P}_1 \tilde{\mathbf{X}} \\ \mathbf{P}_2 \tilde{\mathbf{X}} \\ \mathbf{P}_3 \tilde{\mathbf{X}} \end{bmatrix} = \begin{bmatrix} 0 & -1 & v \\ 1 & 0 & -u \\ -v & u & 0 \end{bmatrix} \begin{bmatrix} \tilde{\mathbf{X}}^\mathsf{T} & \mathbf{0}_{1\times4} & \mathbf{0}_{1\times4} \\ \mathbf{0}_{1\times4} & \tilde{\mathbf{X}}^\mathsf{T} & \mathbf{0}_{1\times4} \\ \mathbf{0}_{1\times4} & \mathbf{0}_{1\times4} & \tilde{\mathbf{X}}^\mathsf{T} \end{bmatrix} \begin{bmatrix} \mathbf{P}_1^\mathsf{T} \\ \mathbf{P}_2^\mathsf{T} \\ \mathbf{P}_3^\mathsf{T} \end{bmatrix} =$$

$$\begin{bmatrix} \mathbf{0}_{1\times4} & -\tilde{\mathbf{X}}^\mathsf{T} & v\tilde{\mathbf{X}}^\mathsf{T} \\ \tilde{\mathbf{X}}^\mathsf{T} & \mathbf{0}_{1\times4} & -u\tilde{\mathbf{X}}^\mathsf{T} \\ -v\tilde{\mathbf{X}}^\mathsf{T} & u\tilde{\mathbf{X}}^\mathsf{T} & \mathbf{0}_{1\times4} \end{bmatrix} \begin{bmatrix} \mathbf{P}_1^\mathsf{T} \\ \mathbf{P}_2^\mathsf{T} \\ \mathbf{P}_3^\mathsf{T} \end{bmatrix} = \mathbf{0}$$

3x12 matrix

Perspective-n-Point



2D image

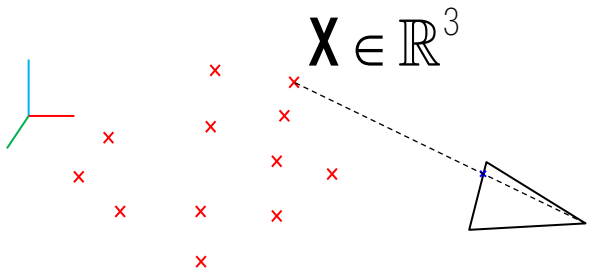
$$\begin{bmatrix} u \\ v \\ 1 \end{bmatrix}_\times \begin{bmatrix} \mathbf{P}_1 \\ \mathbf{P}_2 \\ \mathbf{P}_3 \end{bmatrix} \tilde{\mathbf{X}} = \begin{bmatrix} u \\ v \\ 1 \end{bmatrix}_\times \begin{bmatrix} \mathbf{P}_1 \tilde{\mathbf{X}} \\ \mathbf{P}_2 \tilde{\mathbf{X}} \\ \mathbf{P}_3 \tilde{\mathbf{X}} \end{bmatrix} = \begin{bmatrix} 0 & -1 & v \\ 1 & 0 & -u \\ -v & u & 0 \end{bmatrix} \begin{bmatrix} \tilde{\mathbf{X}}^\mathsf{T} & \mathbf{0}_{1\times4} & \mathbf{0}_{1\times4} \\ \mathbf{0}_{1\times4} & \tilde{\mathbf{X}}^\mathsf{T} & \mathbf{0}_{1\times4} \\ \mathbf{0}_{1\times4} & \mathbf{0}_{1\times4} & \tilde{\mathbf{X}}^\mathsf{T} \end{bmatrix} \begin{bmatrix} \mathbf{P}_1^\mathsf{T} \\ \mathbf{P}_2^\mathsf{T} \\ \mathbf{P}_3^\mathsf{T} \end{bmatrix} =$$

$$\begin{bmatrix} \mathbf{0}_{1\times4} & -\tilde{\mathbf{X}}^\mathsf{T} & v\tilde{\mathbf{X}}^\mathsf{T} \\ \tilde{\mathbf{X}}^\mathsf{T} & \mathbf{0}_{1\times4} & -u\tilde{\mathbf{X}}^\mathsf{T} \\ -v\tilde{\mathbf{X}}^\mathsf{T} & u\tilde{\mathbf{X}}^\mathsf{T} & \mathbf{0}_{1\times4} \end{bmatrix} \begin{bmatrix} \mathbf{P}_1^\mathsf{T} \\ \mathbf{P}_2^\mathsf{T} \\ \mathbf{P}_3^\mathsf{T} \end{bmatrix} = \mathbf{0}$$

3x12 matrix

$$\mathbf{A}\ \mathbf{x} = \mathbf{0}$$

Perspective-n-Point



2D image

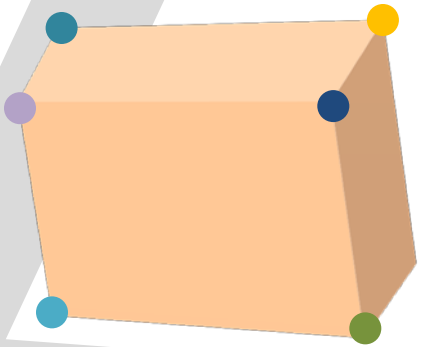
$$\begin{bmatrix} u \\ v \\ 1 \end{bmatrix}_\times \begin{bmatrix} \mathbf{P}_1 \\ \mathbf{P}_2 \\ \mathbf{P}_3 \end{bmatrix} \tilde{\mathbf{X}} = \begin{bmatrix} u \\ v \\ 1 \end{bmatrix}_\times \begin{bmatrix} \mathbf{P}_1 \tilde{\mathbf{X}} \\ \mathbf{P}_2 \tilde{\mathbf{X}} \\ \mathbf{P}_3 \tilde{\mathbf{X}} \end{bmatrix} = \begin{bmatrix} 0 & -1 & v \\ 1 & 0 & -u \\ -v & u & 0 \end{bmatrix} \begin{bmatrix} \tilde{\mathbf{X}}^\mathsf{T} & \mathbf{0}_{1\times4} & \mathbf{0}_{1\times4} \\ \mathbf{0}_{1\times4} & \tilde{\mathbf{X}}^\mathsf{T} & \mathbf{0}_{1\times4} \\ \mathbf{0}_{1\times4} & \mathbf{0}_{1\times4} & \tilde{\mathbf{X}}^\mathsf{T} \end{bmatrix} \begin{bmatrix} \mathbf{P}_1^\mathsf{T} \\ \mathbf{P}_2^\mathsf{T} \\ \mathbf{P}_3^\mathsf{T} \end{bmatrix} =$$

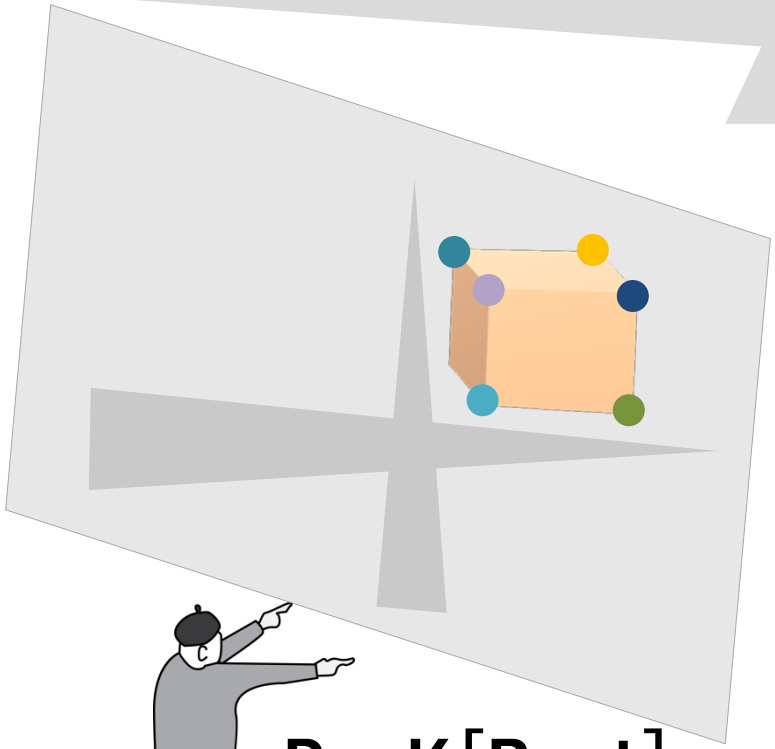
$$\begin{bmatrix} \mathbf{0}_{1\times4} & -\tilde{\mathbf{X}}^\mathsf{T} & v\tilde{\mathbf{X}}^\mathsf{T} \\ \tilde{\mathbf{X}}^\mathsf{T} & \mathbf{0}_{1\times4} & -u\tilde{\mathbf{X}}^\mathsf{T} \\ -v\tilde{\mathbf{X}}^\mathsf{T} & u\tilde{\mathbf{X}}^\mathsf{T} & \mathbf{0}_{1\times4} \end{bmatrix} \begin{bmatrix} \mathbf{P}_1^\mathsf{T} \\ \mathbf{P}_2^\mathsf{T} \\ \mathbf{P}_3^\mathsf{T} \end{bmatrix} = \mathbf{0}$$

3x12 matrix (rank 2)

$$\begin{bmatrix} \mathbf{A}_1 \\ \mathbf{A}_2 \\ \vdots \\ \mathbf{A}_6 \end{bmatrix} \mathbf{x} = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix}$$

Where am I?





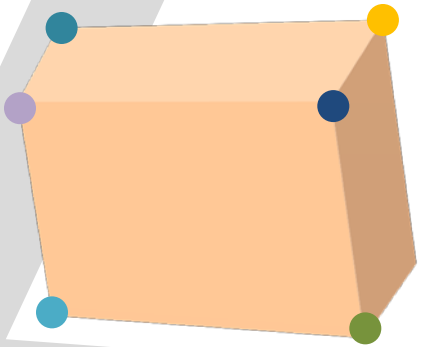
$$R = K^{-1}P_{1:3}$$

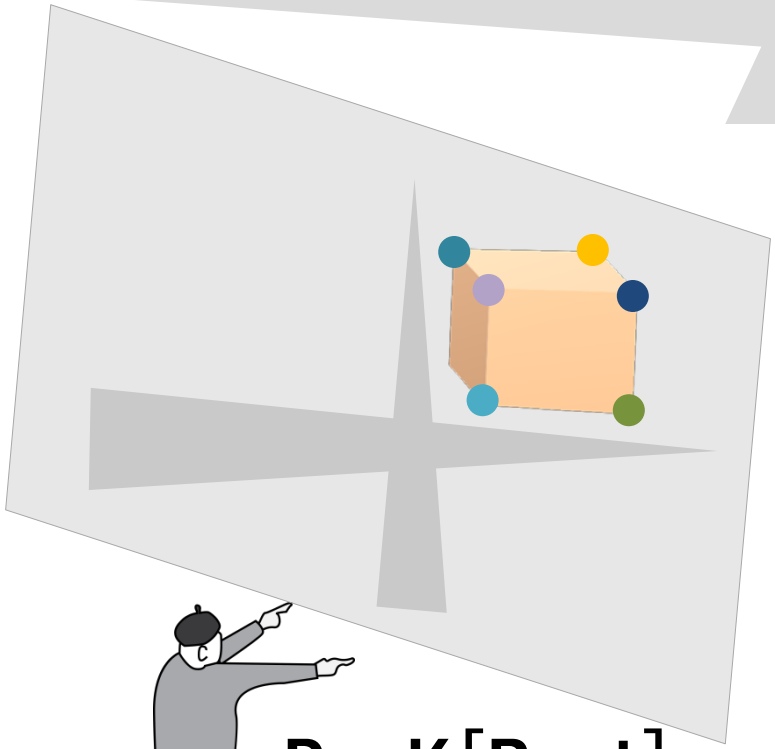
$P_{1:3}$: First three columns of P.

$$P = K\left[\underline{R \quad t}\right]$$

?

Where am I?





$$P = K\left[\underline{R \quad t}\right]$$

$$?$$

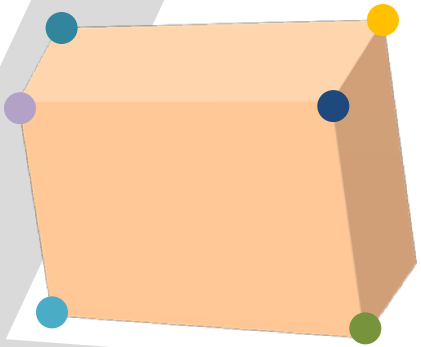
$$R = K^{-1}P_{1:3}$$

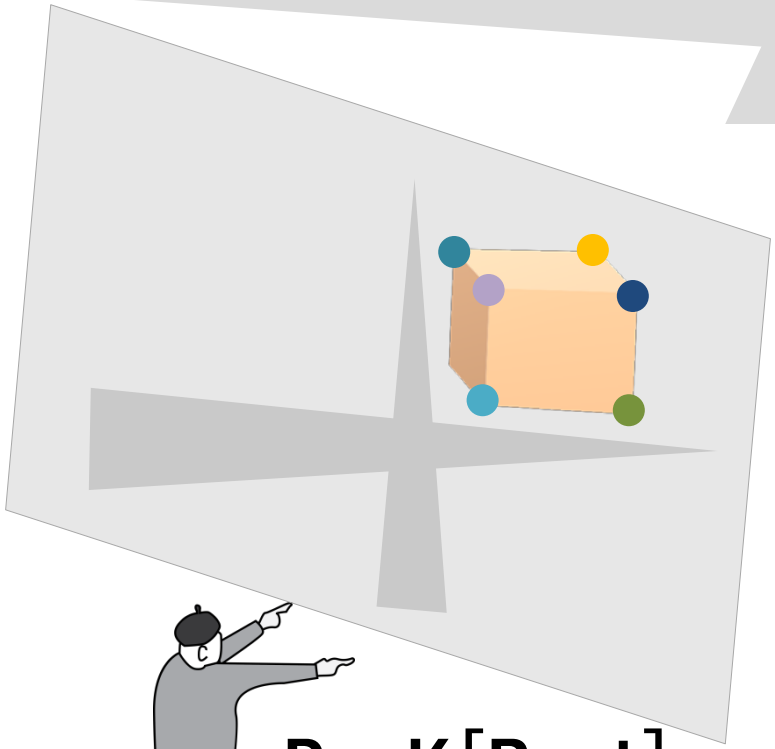
$P_{1:3}$: First three columns of P.

$$R_{+} = UV^{T} \text{ where } UDV^{T} = R$$

Enforcing orthogonal constraint of rotation matrix

Where am I?





$$P = K\left[\underline{R \quad t}\right]$$

?

$$R = K^{-1}P_{1:3}$$

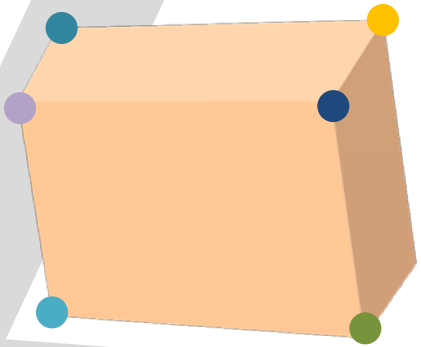
$P_{1:3}$: First three columns of P.

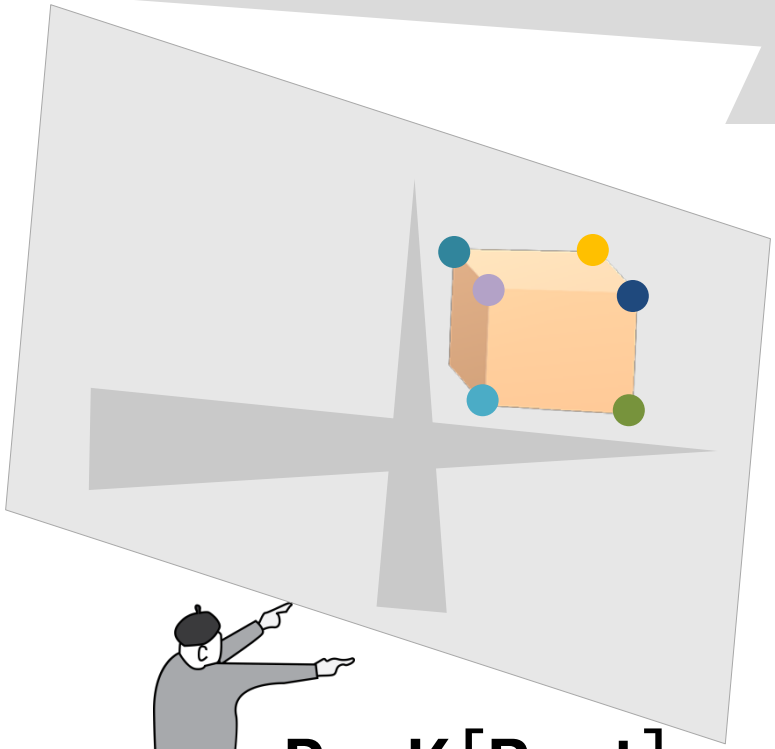
$$R_{+} = UV^{T} \text{ where } UDV^{T} = R$$

Enforcing orthogonal constraint of rotation matrix

$$t = K^{-1}P_{4} / \sigma_{1} \text{ where } D = \text{diag}(\sigma_{1}, \sigma_{2}, \sigma_{3})$$

Where am I?





$$P = K\begin{bmatrix} R & t \end{bmatrix}$$

?

$$R = K^{-1}P_{1:3}$$

$P_{1:3}$: First three columns of P.

$$R_+ = UV^T \quad \text{where} \quad UDV^T = R$$

Enforcing orthogonal constraint of rotation matrix

$$t = K^{-1}P_4 / \sigma_1 \quad \text{where} \quad D = \text{diag}(\sigma_1, \sigma_2, \sigma_3)$$

$$P = K\begin{bmatrix} R_+ & t \end{bmatrix}$$

Camera 3D Registration

Perspective-n-Point Algorithm

Robot Perception:
Pose from 3D Point Correspondences
or
the Procrustes Problem

Kostas Daniilidis

Procrustes Problem

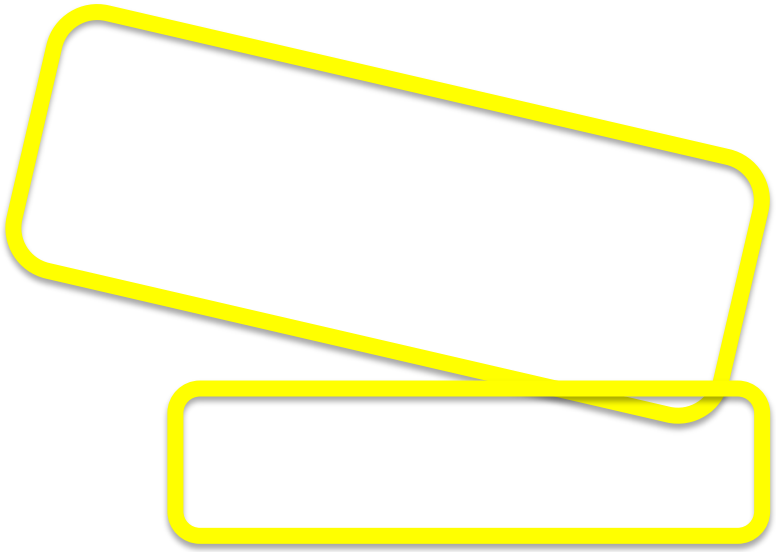
Given two shapes find the scaling, rotation, and translation that fits one into the other.

3D-3D Pose or Procrustes Problem

Given correspondences of points $A_i \in \mathbb{R}^3$ and $B_i \in \mathbb{R}^3$ find the scaling, rotation, and translation transformation, called *similitude* transformation, that satisfies

$$A_i = sRB_i + T$$

for $R \in SO(3)$, $T \in \mathbb{R}$, and $s \in \mathbb{R}^+$.



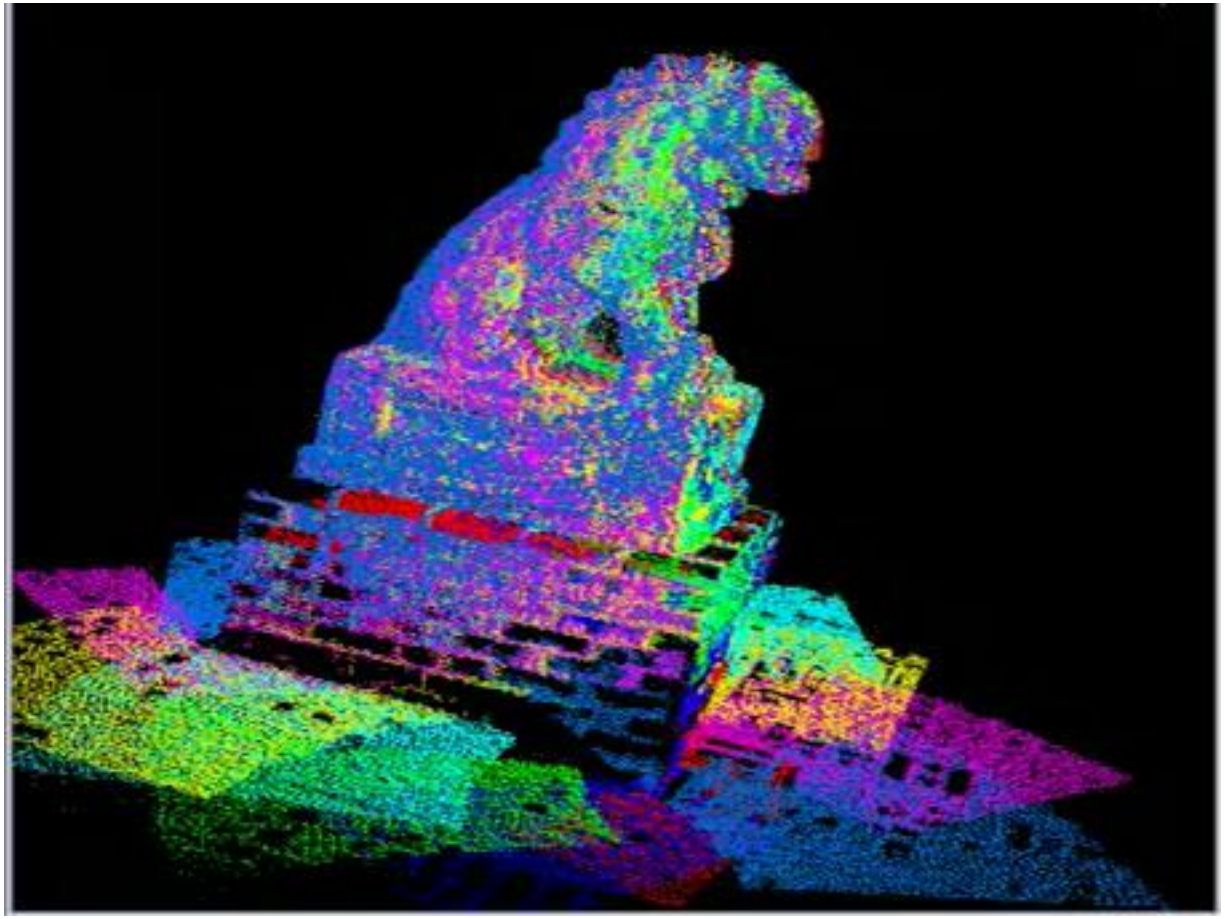
3D-3D Pose or Procrustes Problem

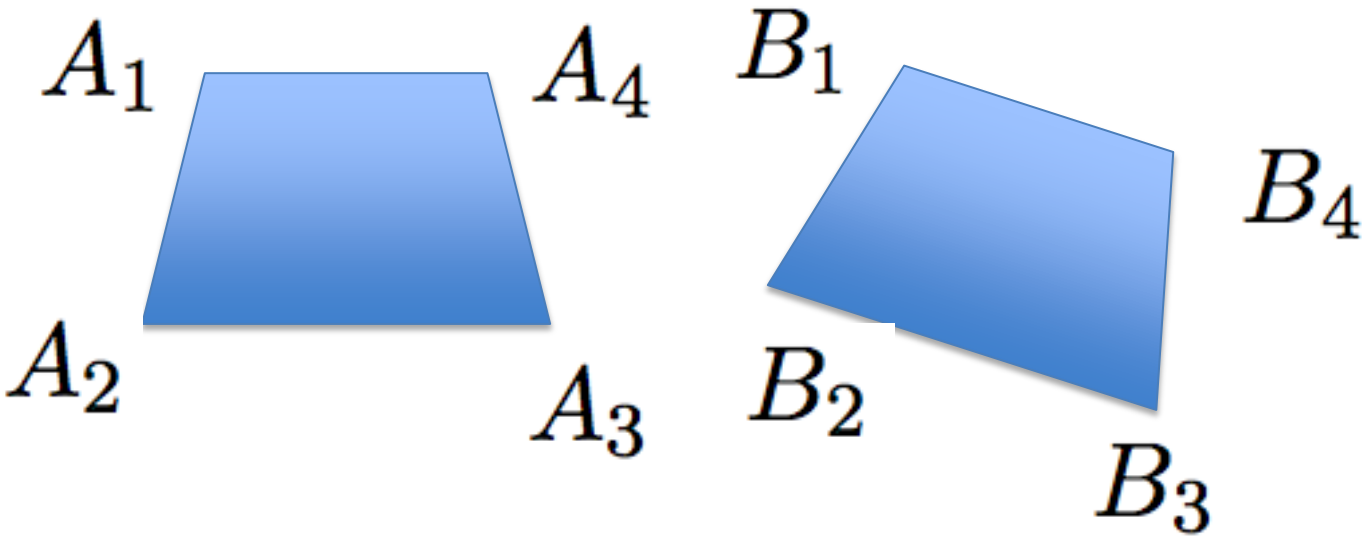
In the camera rigid pose problem scale $s = 1$ is known:

$$Z_i p_i^{cam} = R P_i^{obj} + T$$

This is the last step of the P3P problem or the entire problem of finding rigid pose when we know the depth at every point (e.g., in am RGB-D sensor).

3D-3D Registration enables the creation of 3D models from multiple point clouds:

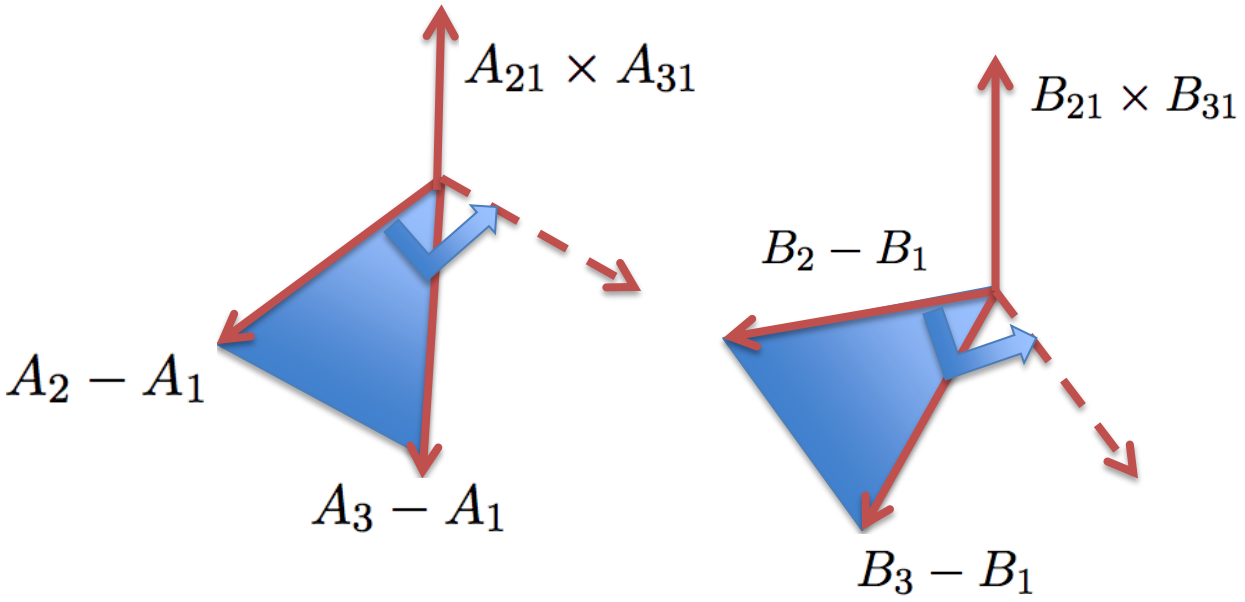


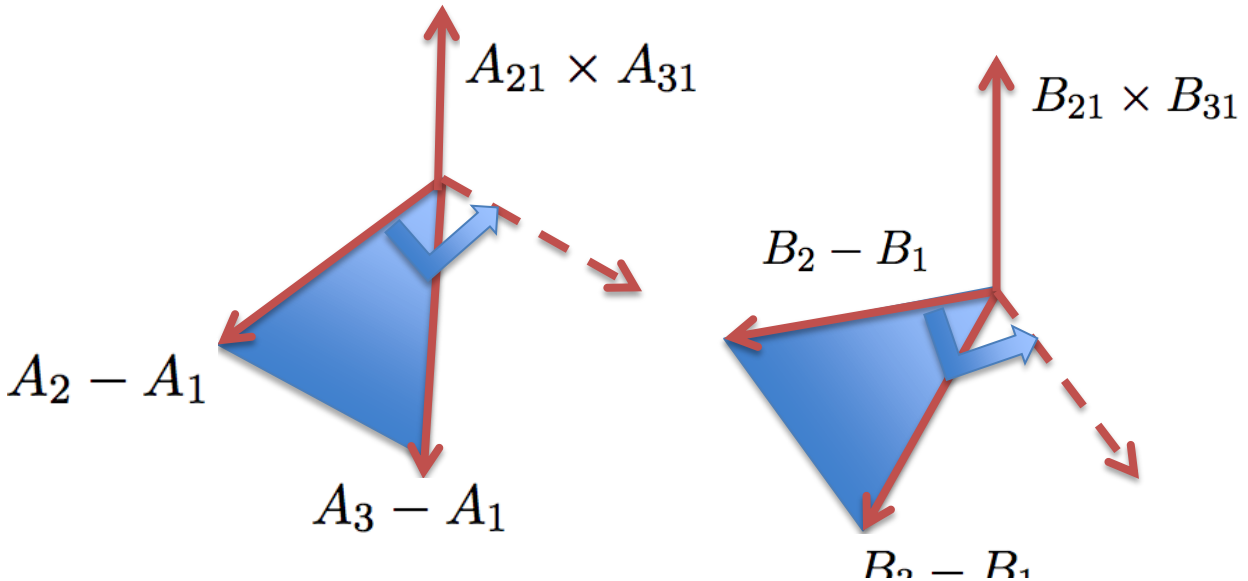


How do we solve for R, T from n point correspondences?

$$A_i = RB_i + T$$

What is the minimal number of points needed?





Three non-collinear points suffice: each triangle $A_{i=1...3}$ and $B_{i=1...3}$ make an orthogonal basis

$$\begin{pmatrix} A_{21} & (A_{21} \times A_{31}) \times A_{21} & A_{21} \times A_{31} \end{pmatrix}$$

and

$$\begin{pmatrix} B_{21} & (B_{21} \times B_{31}) \times B_{21} & B_{21} \times B_{31} \end{pmatrix}$$

Rotation between two orthogonal bases is unique.

We solve a minimization problem for $N > 3$ point correspondences:

$$\min_{R,T} \sum_{i}^{N} \|A_i - RB_i + T\|^2$$

After differentiating with respect to T we observe that the translation is the difference between the centroids:

$$T = \frac{1}{N} \sum_{i}^{N} A_i - R \frac{1}{N} \sum_{i}^{N} B_i = \bar{A} - R\bar{B}$$

We solve a minimization problem for $N > 3$ point correspondences:

$$\min_{R,T} \sum_{i}^{N} \|A_i - RB_i + T\|^2$$

After differentiating with respect to T we observe that the translation is the difference between the centroids:

$$T = \frac{1}{N} \sum_{i}^{N} A_i - R \frac{1}{N} \sum_{i}^{N} B_i = \bar{A} - R\bar{B}$$

We subtract the centroids $\bar{A}$ and $\bar{B}$ and rewrite the objective function as

$$\min_R \|A - RB\|_F^2$$

where

$$A = \begin{pmatrix} A_1 - \bar{A} & \dots & A_N - \bar{A} \end{pmatrix}$$

and

$$B = \begin{pmatrix} B_1 - \bar{B} & \dots & B_N - \bar{B} \end{pmatrix}$$

We rewrite the Frobenius norm using the trace of the matrix

$$\|A - RB\|_F^2 = tr(AA^T) + tr(BB^T) - tr(RBA^T) - tr(AB^T R^T)$$

and observe that only the two last terms depend on the unknown R yielding a maximization problem.

We rewrite the Frobenius norm using the trace of the matrix

$$\|A - RB\|_F^2 = tr(A^T A) + tr(B^T B) - tr(A^T RB) - tr(B^T R^T A)$$

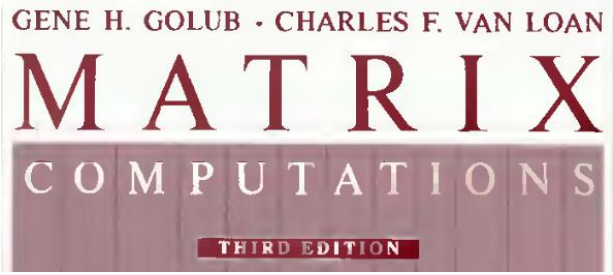
and observe that only the two last terms depend on the unknown R yielding a maximization problem.

Even without using the properties of the trace we can see that both last terms are equal to

$$\sum_i^N R(B_i - \bar{B})(A_i - \bar{A})^T = tr(RBA^T)$$

The 3D-3D pose problem reduced to

$$\max_R tr(RBA^T)$$



If the SVD of BA^T is USV^T and $Z = V^T RU$

$$tr(RBA^T) = tr(RUSV^T) = tr(ZS) = \sum_1^3 z_{ii}\sigma_i \leq \sum_1^3 \sigma_i$$

and, hence, the upper bound is obtained by setting $Z = V^T RU = I$

$$R = VU^T$$

If the SVD of BA^T is USV^T and $Z = V^T RU$

$$tr(RBA^T) = tr(RUSV^T) = tr(ZS) = \sum_1^3 z_{ii}\sigma_i \leq \sum_1^3 \sigma_i$$

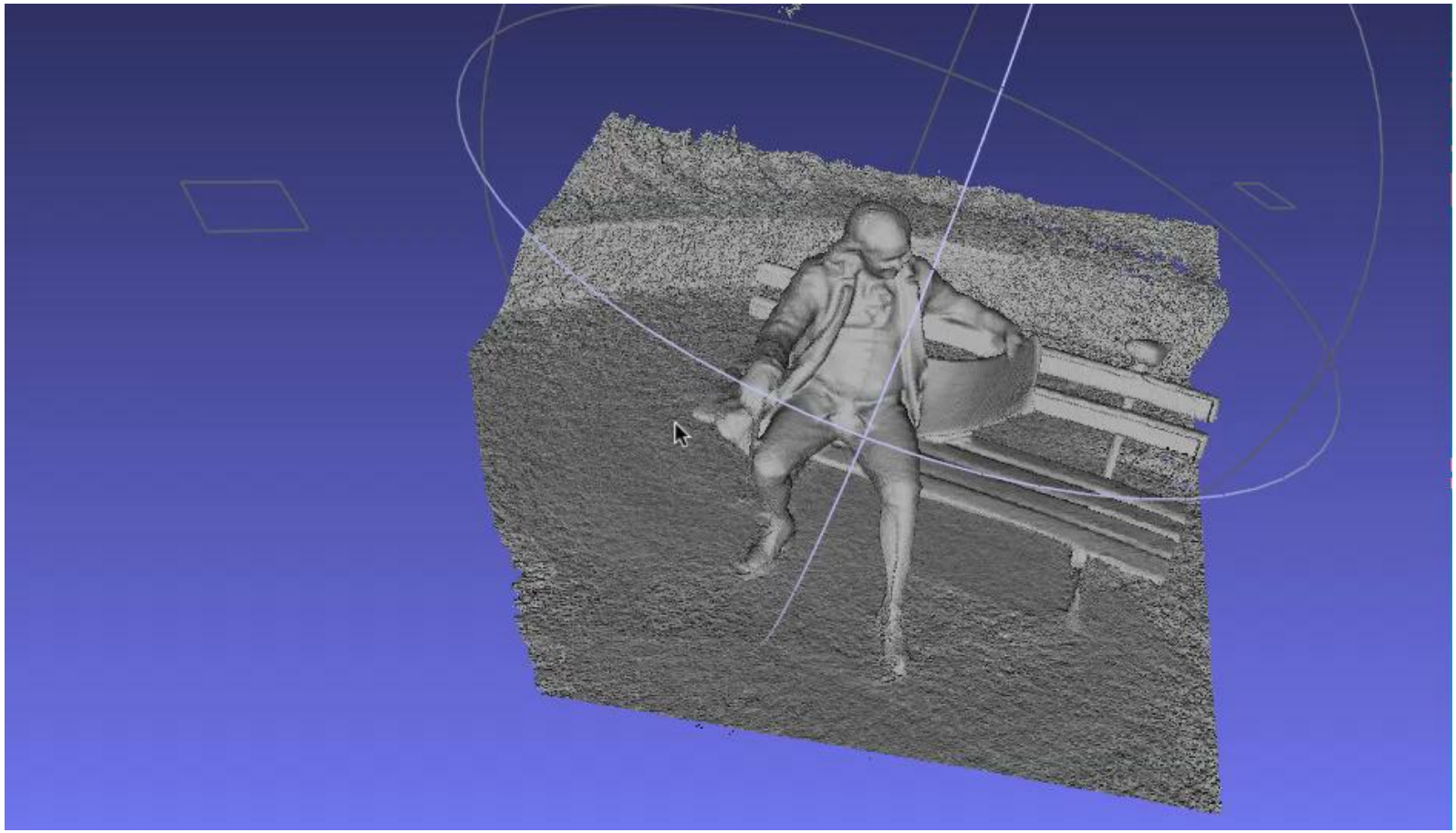
and, hence, the upper bound is obtained by setting $Z = V^T RU = I$

$$R = VU^T$$

To guarantee that it has determinant 1

$$R = V \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & \det(UV^T) \end{pmatrix} U^T$$

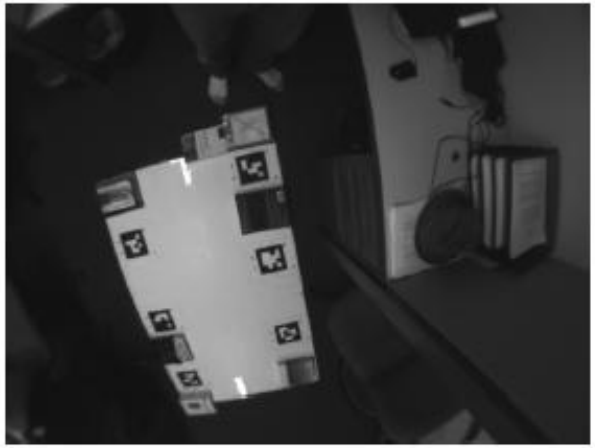
3D-3D Registration enables the creation of 3D models from multiple point clouds:

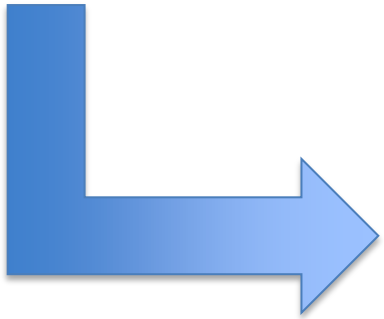


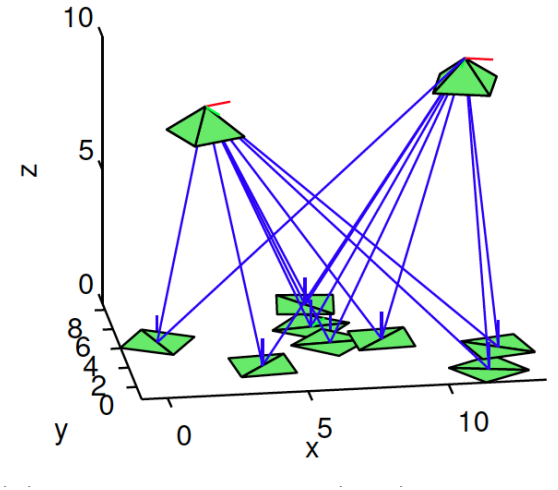
Robot Perception:
Pose from Projective Transformations

Kostas Daniilidis

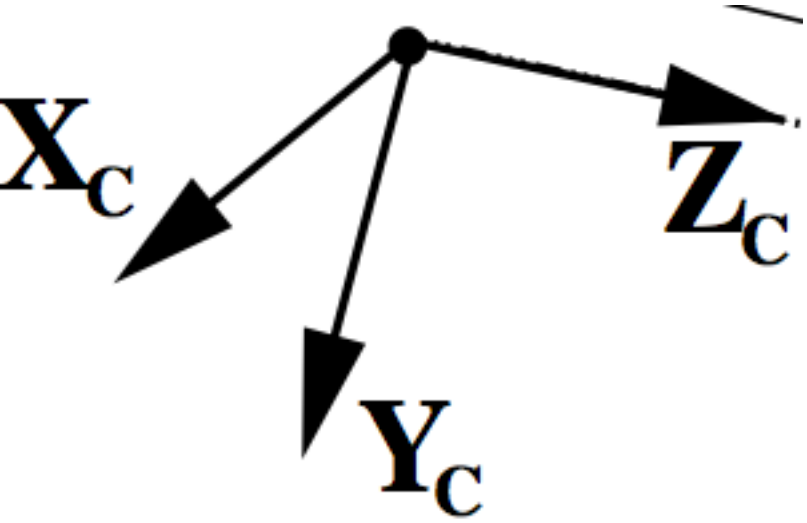
Using the projective transformation the pose of a robot with respect to a planar pattern:

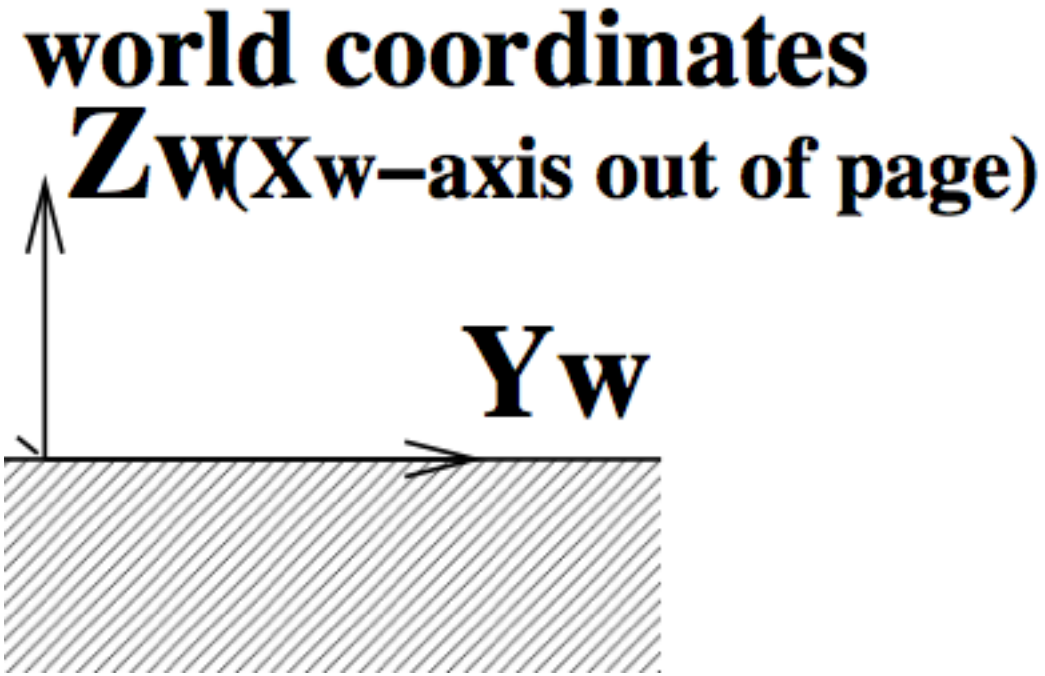






Pose from reference points on plane Z_w=0





Pose from Projective Transformation

Recall the projection from world to camera

$$\begin{pmatrix} u \\ v \\ w \end{pmatrix} \sim K \begin{pmatrix} r_1 & r_2 & r_3 & T \end{pmatrix} \begin{pmatrix} X \\ Y \\ Z \\ W \end{pmatrix}$$

and assume that all points in the world lie in the ground plane $Z = 0$.

Then the transformation reads

$$\begin{pmatrix} u \\ v \\ w \end{pmatrix} \sim K \begin{pmatrix} r_1 & r_2 & T \end{pmatrix} \begin{pmatrix} X \\ Y \\ W \end{pmatrix}$$

Pose from Projective Transformation

Recall the projection from world to camera

$$\begin{pmatrix} u \\ v \\ w \end{pmatrix} \sim K \begin{pmatrix} r_1 & r_2 & r_3 & T \end{pmatrix} \begin{pmatrix} X \\ Y \\ Z \\ W \end{pmatrix}$$

and assume that all points in the world lie in the ground plane $Z = 0$.

Then the transformation reads

$$\begin{pmatrix} u \\ v \\ w \end{pmatrix} \sim K \begin{pmatrix} r_1 & r_2 & T \end{pmatrix} \begin{pmatrix} X \\ Y \\ W \end{pmatrix}$$

H is a transformation from $\mathbb{P}^2$ to $\mathbb{P}^2$:

$$H \sim K \begin{pmatrix} r_1 & r_2 & T \end{pmatrix}$$

Is it a projective transformation? Let us inspect its determinant:

$$\det \begin{pmatrix} r_1 & r_2 & T \end{pmatrix} = T^T (r_1 \times r_2)$$

which vanishes only if the camera lies in the ground plane $Z = 0$. In this case all points would project on a line.

Since $\det(K) = f^2$, H is invertible iff

$$T^T (r_1 \times r_2) \neq 0$$

H is a transformation from $\mathbb{P}^2$ to $\mathbb{P}^2$:

$$H \sim K \begin{pmatrix} r_1 & r_2 & T \end{pmatrix}$$

Is it a projective transformation? Let us inspect its determinant:

$$\det \begin{pmatrix} r_1 & r_2 & T \end{pmatrix} = T^T (r_1 \times r_2)$$

which vanishes only if the camera lies in the ground plane $Z = 0$. In this case all points would project on a line.

Since $\det(K) = f^2$, H is invertible iff

$$T^T (r_1 \times r_2) \neq 0$$

H is a transformation from $\mathbb{P}^2$ to $\mathbb{P}^2$:

$$H \sim K \begin{pmatrix} r_1 & r_2 & T \end{pmatrix}$$

Is it a projective transformation? Let us inspect its determinant:

$$\det \begin{pmatrix} r_1 & r_2 & T \end{pmatrix} = T^T (r_1 \times r_2)$$

which vanishes only if the camera lies in the ground plane $Z = 0$. In this case all points would project on a line.

Since $\det(K) = f^2$, H is invertible iff

$$T^T (r_1 \times r_2) \neq 0$$

Suppose we estimate an H from $N \geq 4$ correspondences.

Let us assume that we know the intrinsic parameters K.

Pose estimation means finding R, T given H and intrinsics K.

We observe that

$$K^{-1}H = \begin{pmatrix} r_1 & r_2 & T \end{pmatrix}$$

has specific properties: its first two columns are orthogonal unit vectors.

Nothing guarantees that that the H we computed will satisfy this condition.

Suppose we estimate an H from $N \geq 4$ correspondences.

Let us assume that we know the intrinsic parameters K.

Pose estimation means finding R, T given H and intrinsics K.

We observe that

$$K^{-1}H = \begin{pmatrix} r_1 & r_2 & T \end{pmatrix}$$

has specific properties: its first two columns are orthogonal unit vectors.

Nothing guarantees that that the H we computed will satisfy this condition.

Suppose we estimate an H from $N \geq 4$ correspondences.

Let us assume that we know the intrinsic parameters K.

Pose estimation means finding R, T given H and intrinsics K.

We observe that

$$K^{-1}H = \begin{pmatrix} r_1 & r_2 & T \end{pmatrix}$$

has specific properties: its first two columns are orthogonal unit vectors.

Nothing guarantees that that the H we computed will satisfy this condition.

Let us name the columns of $K^{-1}H$:

$$K^{-1}H = \begin{pmatrix} h'_1 & h'_2 & h'_3 \end{pmatrix}$$

We seek orthogonal r_1 and r_2 that are the closest to h'_1 and h'_2. The solution to this problem is given by the Singular Value Decomposition.

We find the orthogonal matrix R that is the closest to $\begin{pmatrix} h'_1 & h'_2 & h'_1 \times h'_2 \end{pmatrix}$:

$$\underset{R \in SO(3)}{\arg\min} \left\| R - \begin{pmatrix} h'_1 & h'_2 & h'_1 \times h'_2 \end{pmatrix} \right\|_F^2$$

Let us name the columns of $K^{-1}H$:

$$K^{-1}H = \begin{pmatrix} h'_1 & h'_2 & h'_3 \end{pmatrix}$$

We seek orthogonal r_1 and r_2 that are the closest to h'_1 and h'_2. The solution to this problem is given by the Singular Value Decomposition.

We find the orthogonal matrix R that is the closest to $\begin{pmatrix} h'_1 & h'_2 & h'_1 \times h'_2 \end{pmatrix}$:

$$\underset{R \in SO(3)}{\arg\min} \| R - \begin{pmatrix} h'_1 & h'_2 & h'_1 \times h'_2 \end{pmatrix} \|_F^2$$

$$\underset{R \in SO(3)}{\arg\min} \left\| R - \begin{pmatrix} h'_1 & h'_2 & h'_1 \times h'_2 \end{pmatrix} \right\|_F^2$$

If the SVD of

$$\begin{pmatrix} h'_1 & h'_2 & h'_1 \times h'_2 \end{pmatrix} = USV^T$$

then the solution is

$$R = U \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & \det(UV^T) \end{pmatrix} V^T$$

The diagonal matrix is inserted to guarantee that $\det(R) = 1$.

To find the translation : $T = h'_3 / \|h'_1\|$

$$\underset{R \in SO(3)}{\arg\min} \left\| R - \begin{pmatrix} h'_1 & h'_2 & h'_1 \times h'_2 \end{pmatrix} \right\|_F^2$$

If the SVD of

$$\begin{pmatrix} h'_1 & h'_2 & h'_1 \times h'_2 \end{pmatrix} = USV^T$$

then the solution is

$$R = U \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & \det(UV^T) \end{pmatrix} V^T$$

The diagonal matrix is inserted to guarantee that $\det(R) = 1$.

To find the translation : $T = h'_3 / \|h'_1\|$

$$\underset{R \in SO(3)}{\arg\min} \left\| R - \begin{pmatrix} h'_1 & h'_2 & h'_1 \times h'_2 \end{pmatrix} \right\|_F^2$$

If the SVD of

$$\begin{pmatrix} h'_1 & h'_2 & h'_1 \times h'_2 \end{pmatrix} = USV^T$$

then the solution is

$$R = U \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & \det(UV^T) \end{pmatrix} V^T$$

The diagonal matrix is inserted to guarantee that $\det(R) = 1$.

To find the translation : $T = h'_3 / \|h'_1\|$

$$\underset{R \in SO(3)}{\arg\min} \left\| R - \begin{pmatrix} h'_1 & h'_2 & h'_1 \times h'_2 \end{pmatrix} \right\|_F^2$$

If the SVD of

$$\begin{pmatrix} h'_1 & h'_2 & h'_1 \times h'_2 \end{pmatrix} = USV^T$$

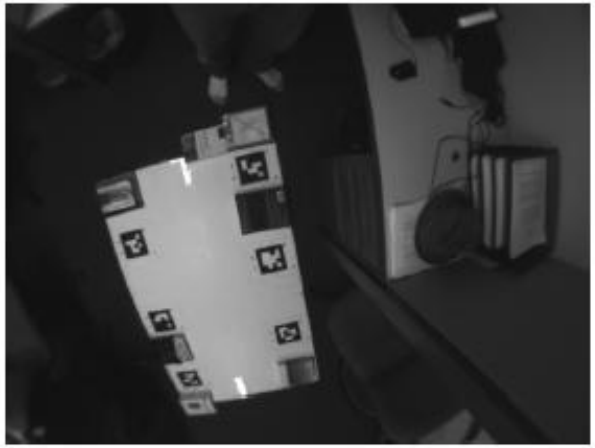
then the solution is

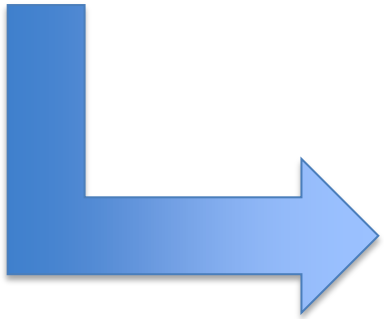
$$R = U \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & \det(UV^T) \end{pmatrix} V^T$$

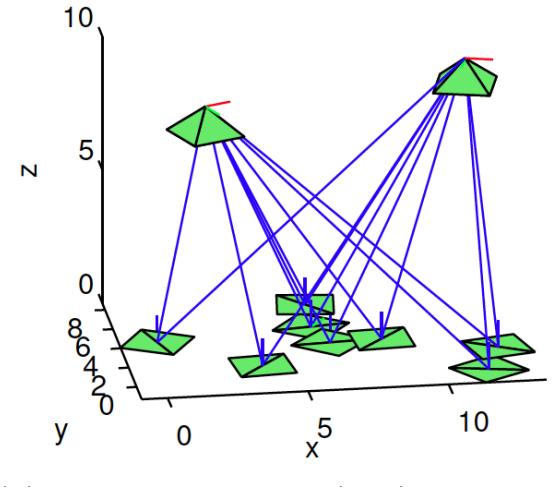
The diagonal matrix is inserted to guarantee that $\det(R) = 1$.

To find the translation : $T = h'_3 / \|h'_1\|$

Using the projective transformation the pose of a robot with respect to a planar pattern:







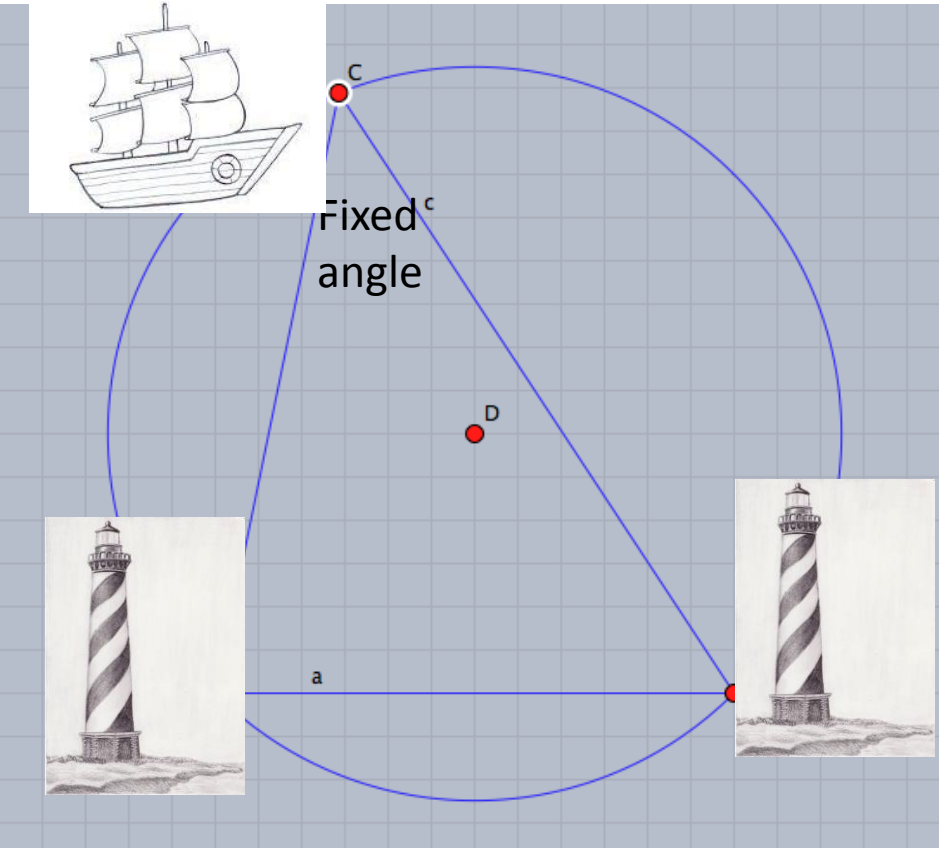
Robot Perception:
Pose from Point Correspondences
or
the Perspective N Point Problem (PnP)

Kostas Daniilidis

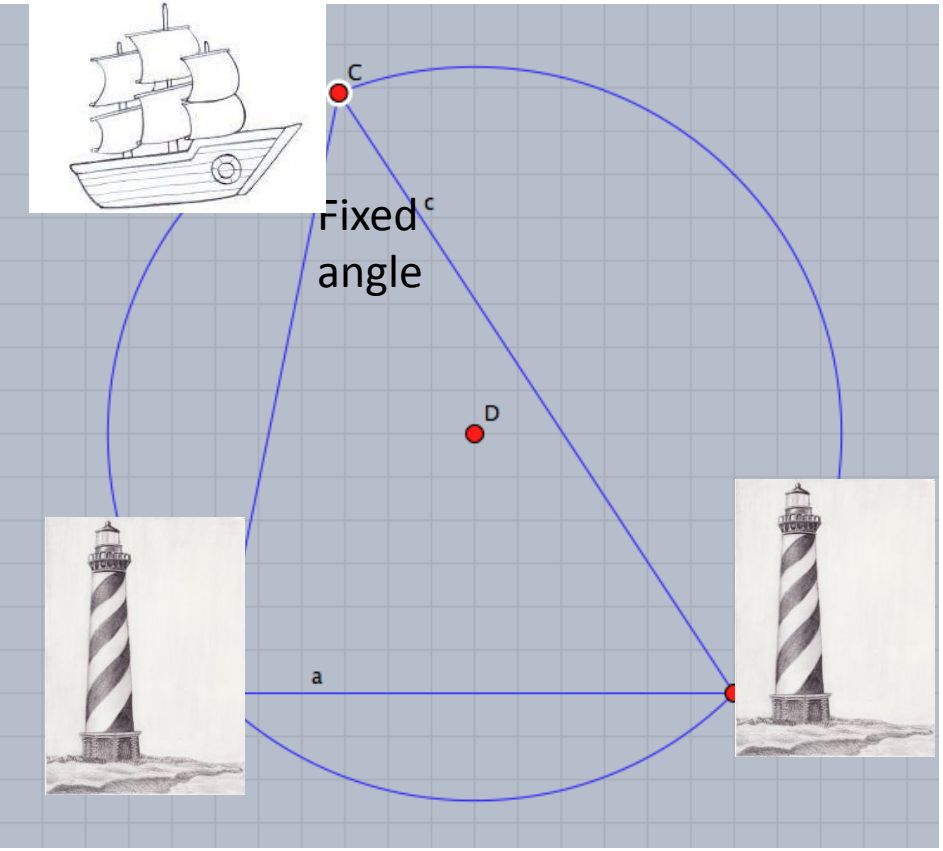
Where is the car if we know the projections of 3 points with known 3D coordinates?

Get back to an old idea of bearing!

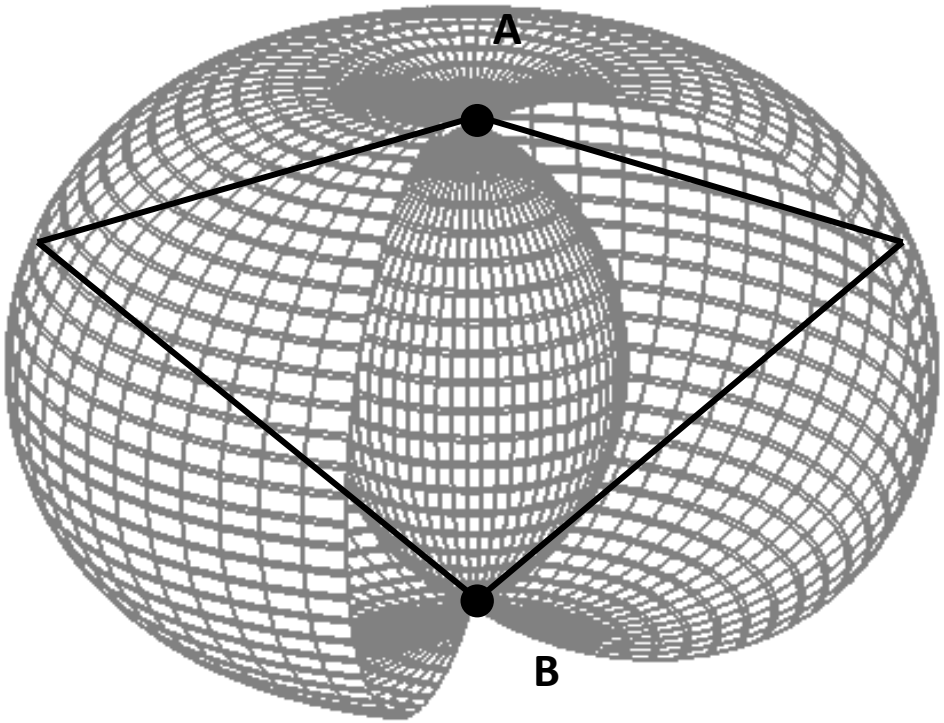
If we see two points like two lighthouses under a fixed angle, where am I?



Two points are not sufficient, camera can lie anywhere on a circle in 2D



… and on a toroid in 3D:



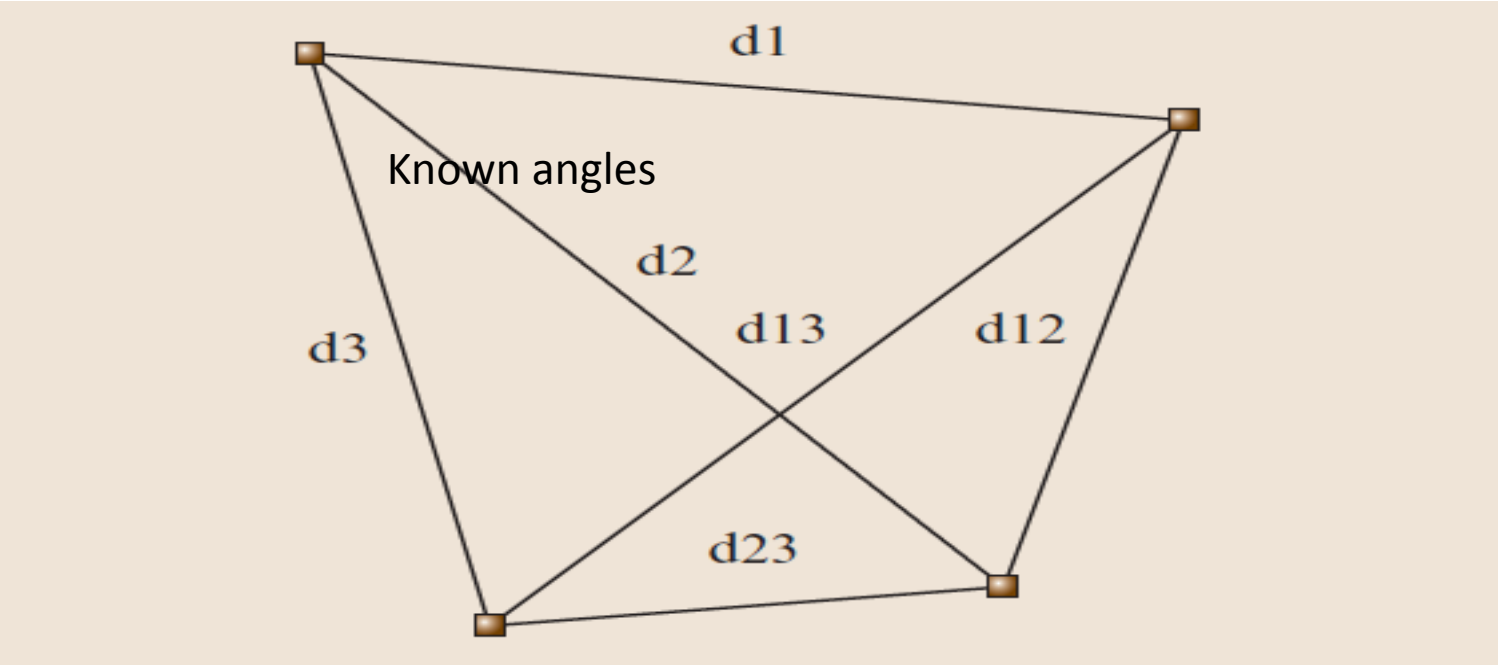
We need a 3rd point: The Perspective 3-Point problem of P3P

Or in Photogrammetry the Resection Problem:

The Snellius-Pothenot problem!

J. A. Grunert, "Das Pothenotische Problem in erweiterter Gestalt nebst Über seine Anwendungen in der Geodäsie, *Grunerts Archiv für Mathematik und Physik*, Band 1, 1841, pp. 238–248.

We need a 3rd point: The Perspective 3-Point problem of P3P



Let us assume two scene points and denote the known angle between their projections p_i and p_j as δ_{ij}

Let us denote the squared distance $\|P_i - P_j\|^2$ with d_{ij}^2 and the lengths of P_j with d_j.

Then cosine law reads

$$d_i^2 + d_j^2 - 2d_i d_j \cos\delta_{ij} = d_{ij}^2$$

Pose from 3 points: The perspective 3-point problem (P3P)

$$d_i^2 + d_j^2 - 2d_i d_j \cos\delta_{ij} = d_{ij}^2$$

If we can recover d_i and d_j the rest will be an absolute orientation problem

$$\lambda_j p_j = \mathbf{R} P_j + T$$

to recover translation and rotation between camera and world coordinate system.

Pose from 3 points: The perspective 3-point problem (P3P)

$$d_i^2 + d_j^2 - 2 d_i d_j \cos \delta_{ij} = d_{ij}^2$$

If we can recover d_i and d_j the rest will be an absolute orientation problem

$$\lambda_j p_j = \mathbf{R} P_j + T$$

to recover translation and rotation between camera and world coordinate system.

The cosine law

$$d_i^2 + d_j^2 - 2 d_i d_j \cos \delta_{ij} = d_{ij}^2$$

applies for each point pair. With 3 points we could solve 3 quadratic equations for $d_{i=1\ldots3}$.

Set $d_2 = u d_1$ and $d_3 = v d_1$ and solve all three equations for d_1:

$$\begin{aligned} d_1^2 &= \frac{d_{23}^2}{u^2 + v^2 - 2uv \cos \delta_{23}} \\ d_1^2 &= \frac{d_{13}^2}{1 + v^2 - 2v \cos \delta_{13}} \\ d_1^2 &= \frac{d_{12}^2}{u^2 + 1 - 2u \cos \delta_{12}} \end{aligned}$$

which is equivalent to two quadratic equations in u and v.

The cosine law

$$d_i^2 + d_j^2 - 2 d_i d_j \cos \delta_{ij} = d_{ij}^2$$

applies for each point pair. With 3 points we could solve 3 quadratic equations for $d_{i=1\ldots3}$.

Set $d_2 = u d_1$ and $d_3 = v d_1$ and solve all three equations for d_1:

$$\begin{aligned} d_1^2 &= \frac{d_{23}^2}{u^2 + v^2 - 2uv \cos \delta_{23}} \\ d_1^2 &= \frac{d_{13}^2}{1 + v^2 - 2v \cos \delta_{13}} \\ d_1^2 &= \frac{d_{12}^2}{u^2 + 1 - 2u \cos \delta_{12}} \end{aligned}$$

which is equivalent to two quadratic equations in u and v.

$$d_{13}^2(u^2+v^2-2uv\cos\delta_{23}) = d_{23}^2(1+v^2-2v\cos\delta_{13}) \quad (1)$$
$$d_{12}^2(1+v^2-2v\cos\delta_{13}) = d_{13}^2(u^2+1-2u\cos\delta_{12}) \quad (2)$$

We solve (1) for u^2,
we insert it in (2)
and solve for u which appears only linearly.
Then we insert u back in (1) and obtain a quartic (4th degree) in v which can have as many as 4 real solutions for v.

In the next lecture we will see how we can solve

$$P_i^c = RP_i + T$$

for R, T.

DIRECT SOLUTIONS OF THE PnP PROBLEM

Pose from N points in space given intrinsic parameters K and correspondences $(X_i, Y_i, Z_i, x_i, y_i)_{i=1...N}$
where

$$\begin{pmatrix} x_i \\ y_i \\ 1 \end{pmatrix} \sim K^{-1} \begin{pmatrix} u_i \\ v_i \\ 1 \end{pmatrix}$$

where u_i, v_i are pixel coordinates.

DIRECT SOLUSTIONS

Pose from N points in space given intrinsic parameters K and correspondences $(X_i, Y_i, Z_i, x_i, y_i)_{i=1...N}$
where

$$\begin{pmatrix} x_i \\ y_i \\ 1 \end{pmatrix} \sim K^{-1} \begin{pmatrix} u_i \\ v_i \\ 1 \end{pmatrix}$$

where u_i, v_i are pixel coordinates.

Given $(X_i, Y_i, Z_i, x_i, y_i)_{i=1...N}$
find R, T such that

$$\lambda_i \begin{pmatrix} x_i \\ y_i \\ 1 \end{pmatrix} = R \begin{pmatrix} X_i \\ Y_i \\ Z_i \end{pmatrix} + T$$

How many points do we need?

$$
\begin{aligned}
x_i &= \frac{r_{11}X_i + r_{12}Y_i + r_{13}Z_i + T_1}{r_{31}X_i + r_{32}Y_i + r_{33}Z_i + T_3} \\
y_i &= \frac{r_{21}X_i + r_{22}Y_i + r_{23}Z_i + T_2}{r_{31}X_i + r_{32}Y_i + r_{33}Z_i + T_3}
\end{aligned}
$$

Assuming that point correspondences produce independent equations we would obtain $2N$ equations for N points.

The unknowns of a rigid transformation are 6, so it seems that the minimal number of points is 3.

A brute-force approach would be to eliminate depths λ_i and try to find R, T such that

$$
\begin{aligned}
x_i &= \frac{r_{11}X_i + r_{12}Y_i + r_{13}Z_i + T_1}{r_{31}X_i + r_{32}Y_i + r_{33}Z_i + T_3} \\
y_i &= \frac{r_{21}X_i + r_{22}Y_i + r_{23}Z_i + T_2}{r_{31}X_i + r_{32}Y_i + r_{33}Z_i + T_3}
\end{aligned}
$$

This is a non-linear problem which needs an iterative approach.

A "dirty" trick would be to solve for a vector of 12 unknowns (R', T') and then find the closest orthogonal R to R'.