SSY098 - Image Analysis

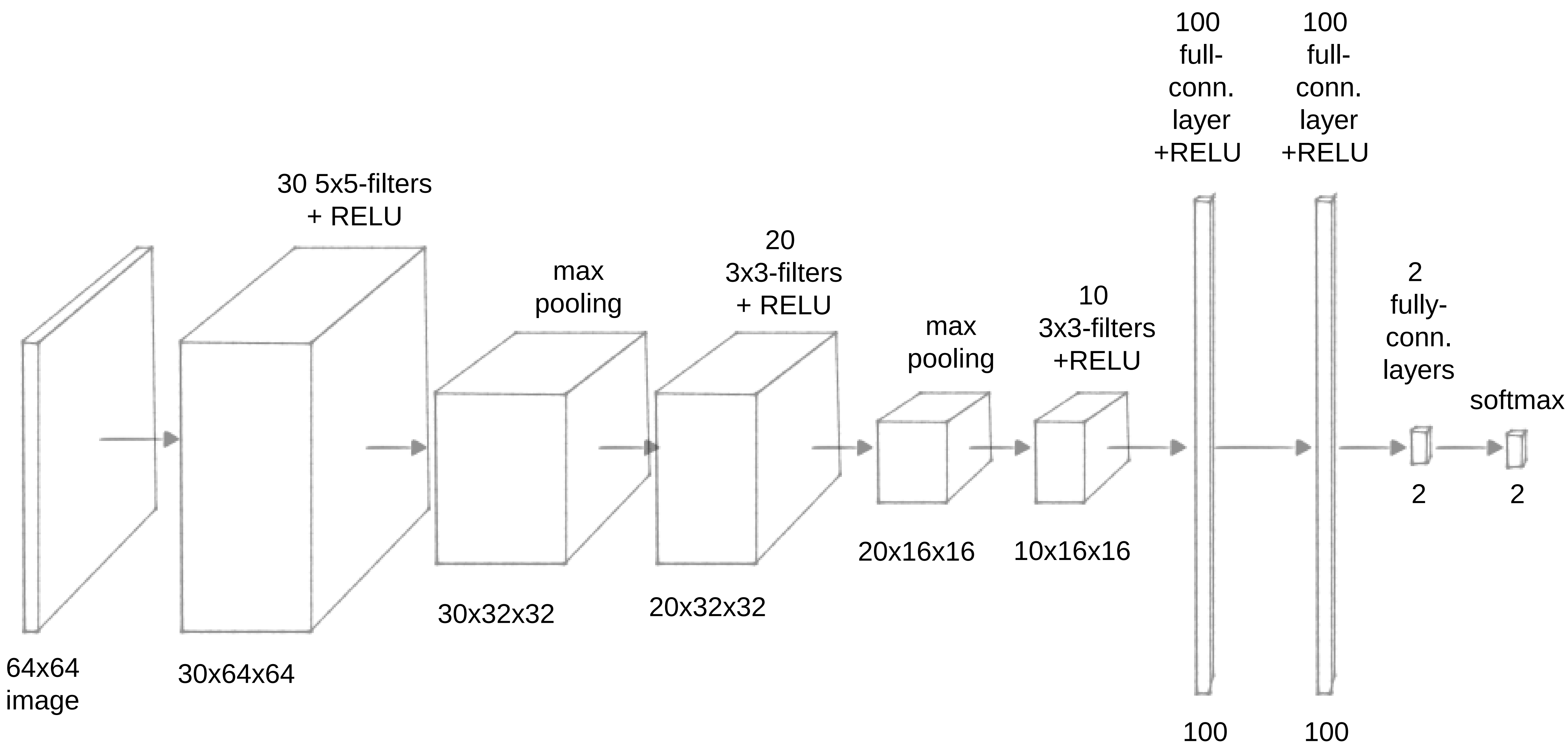
Lecture 7 - Robust Model Fitting

Torsten Sattler
(slides adapted from Olof Enqvist)

Last Lecture

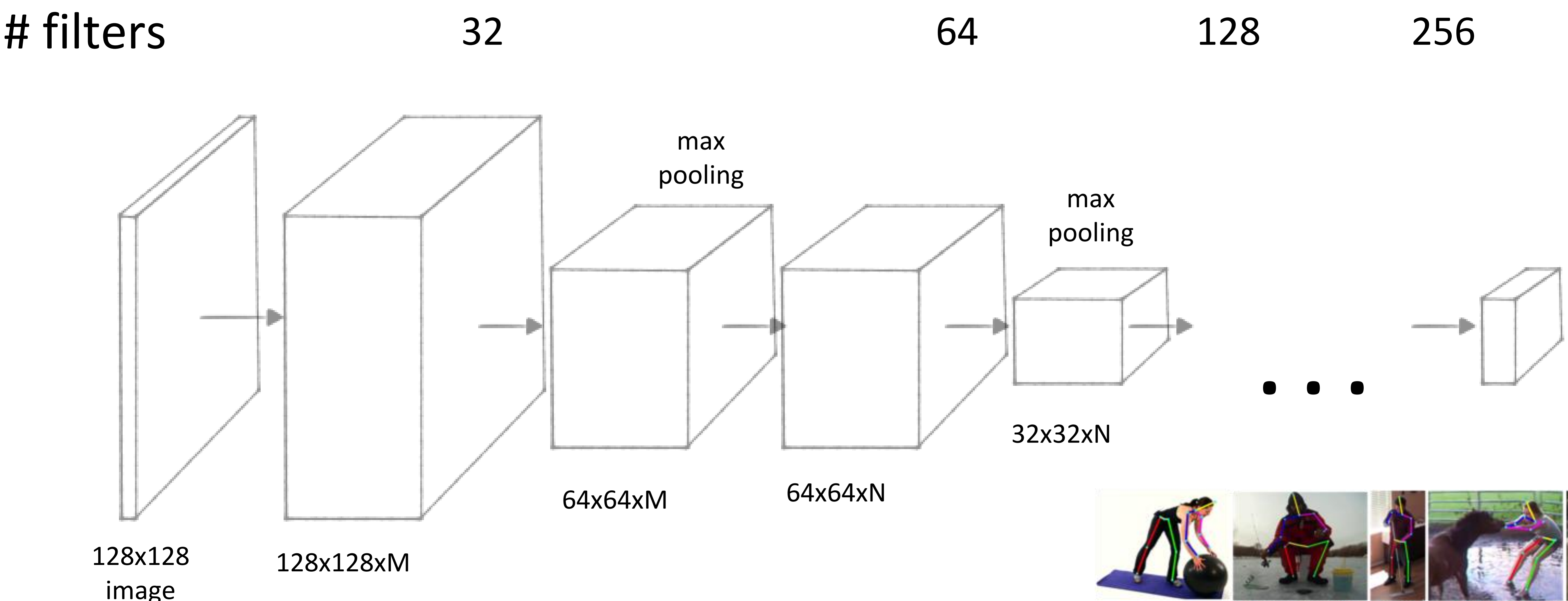
Jan. 20	Introduction, Linear classifiers and filtering	
Jan. 23	Filtering, gradients, scale	Lab 1
Jan. 27	Local features	
Jan. 30	Learning a classifier	Lab 2
Feb. 3	Convolutional neural networks	
Feb. 6	**More convolutional neural networks**	
Feb. 10	Robust model fitting and RANSAC	Lab 3
Feb. 13	Image registration	
Feb. 17	Camera Geometry	Lab 4
Feb. 20	More camera geometry	
Feb. 24	Generative neural networks	
Feb. 27	Generative neural networks	
Mar. 2	TBA	
Mar. 9	TBA	

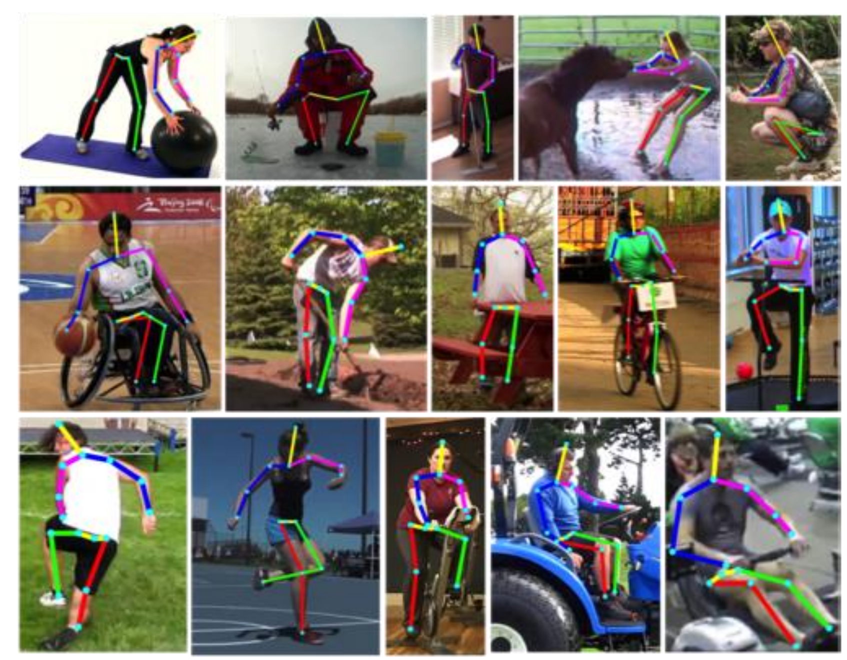
Last Lecture



Convolutional Neural Networks

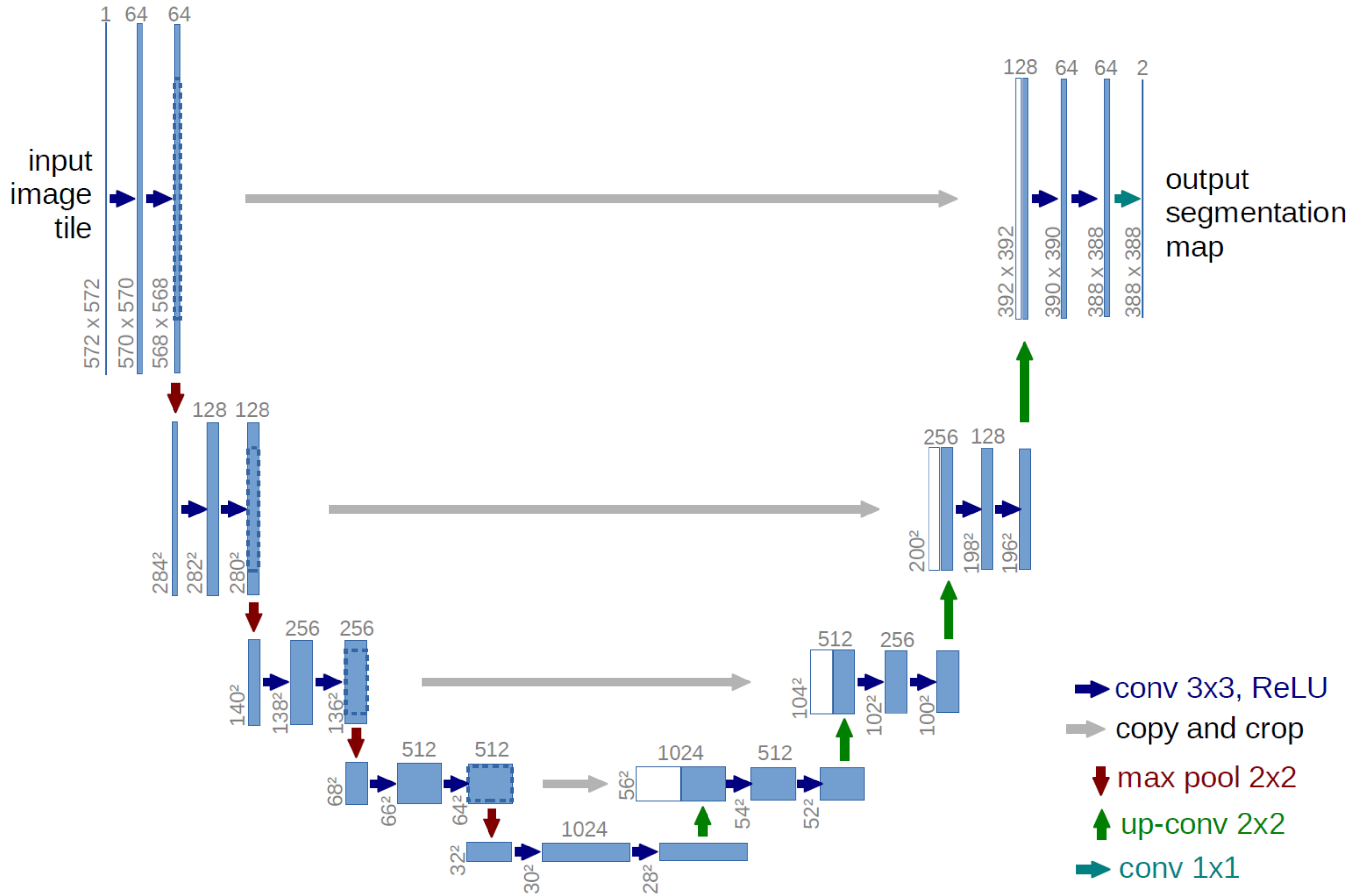
Last Lecture





Fully Convolutional Neural Networks

Last Lecture



Fully Convolutional Neural Networks

Last Lecture

$$w = \frac{1}{5}\begin{bmatrix} 0 & 1 & 0 \\ 1 & 1 & 1 \\ 0 & 1 & 0 \end{bmatrix}$$

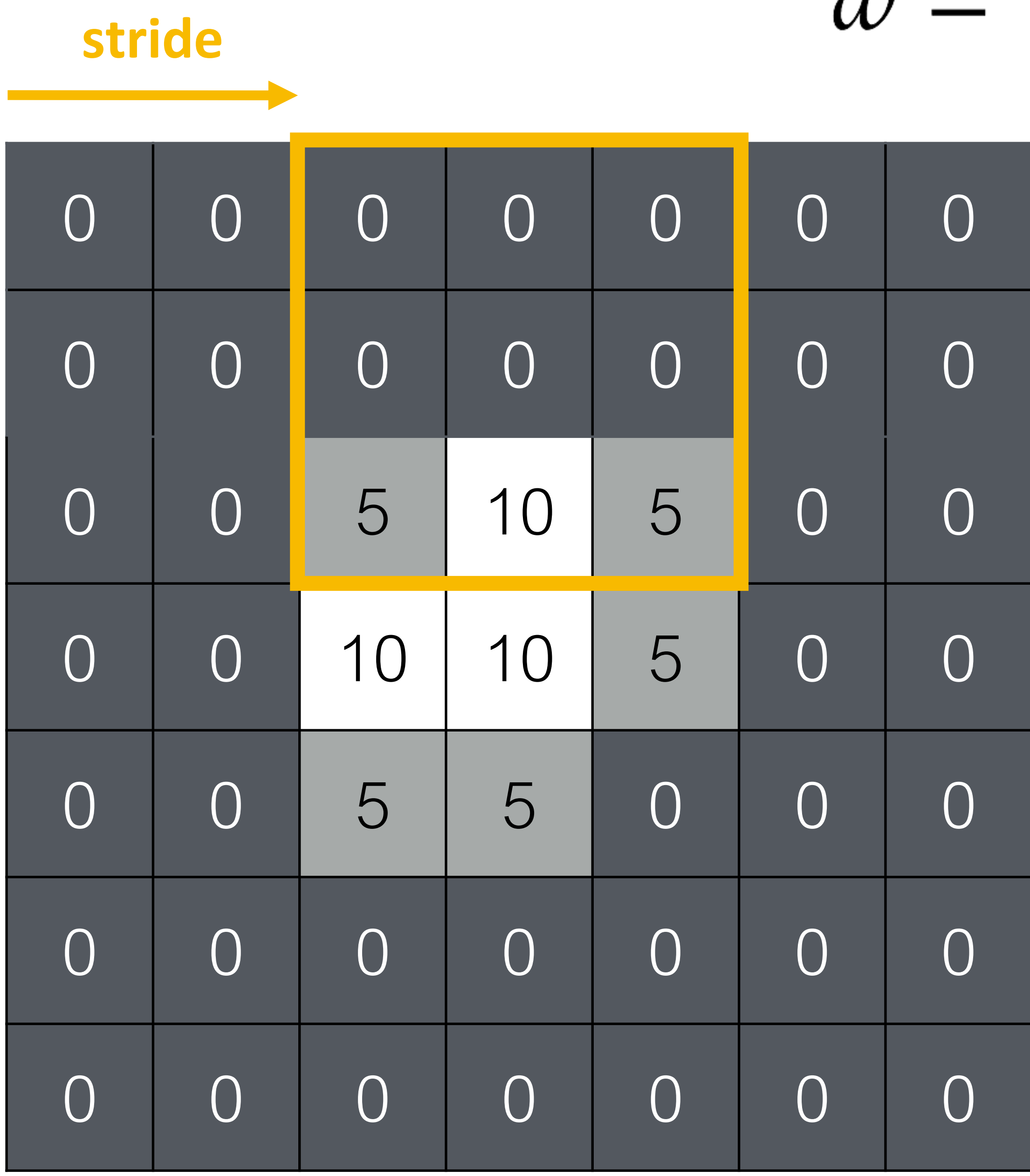
0	0	0	0	0	0	0
0	0	0	0	0	0	0
0	0	5	10	5	0	0
0	0	10	10	5	0	0
0	0	5	5	0	0	0
0	0	0	0	0	0	0
0	0	0	0	0	0	0

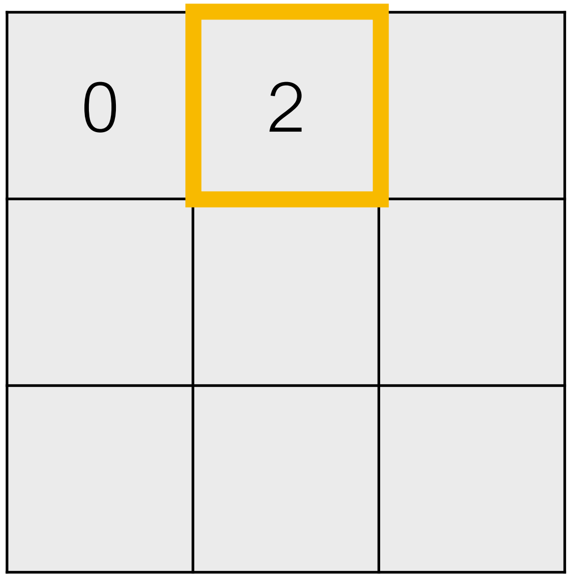
0		

Strided convolutions

Last Lecture

$$w = \frac{1}{5}\begin{bmatrix} 0 & 1 & 0 \\ 1 & 1 & 1 \\ 0 & 1 & 0 \end{bmatrix}$$





Strided convolutions

Last Lecture

$$w = \frac{1}{5}\begin{bmatrix} 0 & 1 & 0 \\ 1 & 1 & 1 \\ 0 & 1 & 0 \end{bmatrix}$$

0	0	0	0	0	0	0
0	0	0	0	0	0	0
0	0	5	10	5	0	0
0	0	10	10	5	0	0
0	0	5	5	0	0	0
0	0	0	0	0	0	0
0	0	0	0	0	0	0

0	2	0

Strided convolutions

Last Lecture

$$w = \frac{1}{5}\begin{bmatrix} 0 & 1 & 0 \\ 1 & 1 & 1 \\ 0 & 1 & 0 \end{bmatrix}$$

0	0	0	0	0	0	0
0	0	0	0	0	0	0
0	0	5	10	5	0	0
0	0	10	10	5	0	0
0	0	5	5	0	0	0
0	0	0	0	0	0	0
0	0	0	0	0	0	0

0	2	0
2		

Strided convolutions

Last Lecture

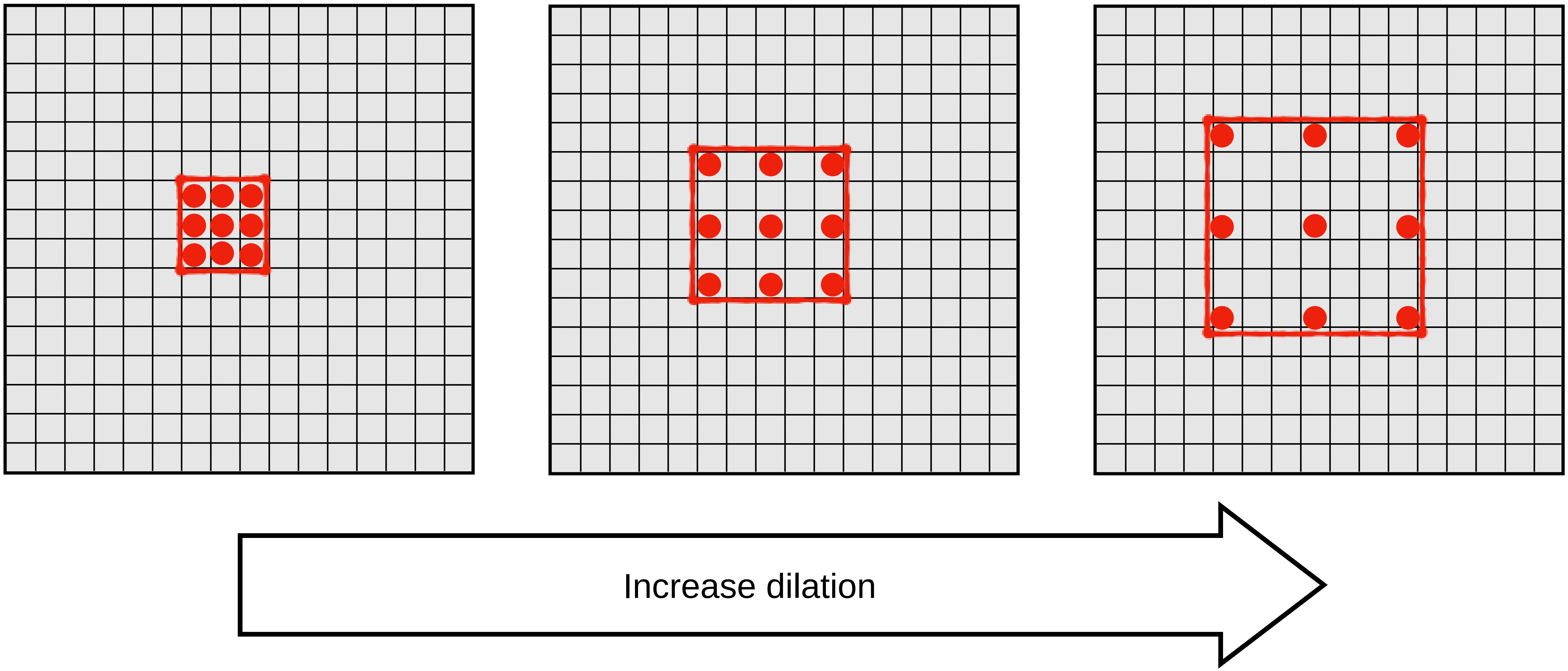
$$w = \frac{1}{5} \begin{bmatrix} 0 & 1 & 0 \\ 1 & 1 & 1 \\ 0 & 1 & 0 \end{bmatrix}$$

0	0	0	0	0	0	0
0	0	0	0	0	0	0
0	0	5	10	5	0	0
0	0	10	10	5	0	0
0	0	5	5	0	0	0
0	0	0	0	0	0	0
0	0	0	0	0	0	0

0	2	0
2	8	

Strided convolutions

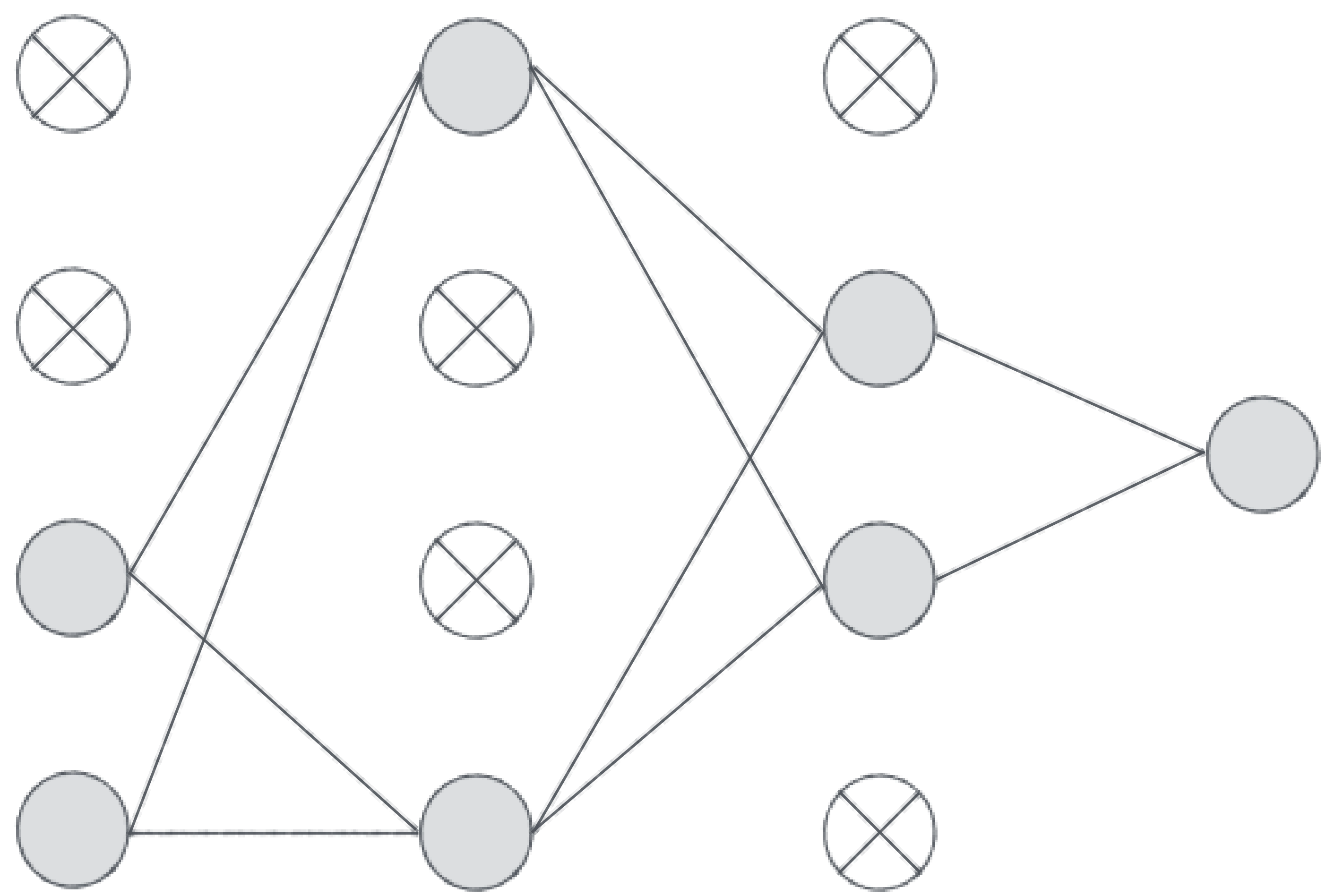
Last Lecture



Dilated Convolutions

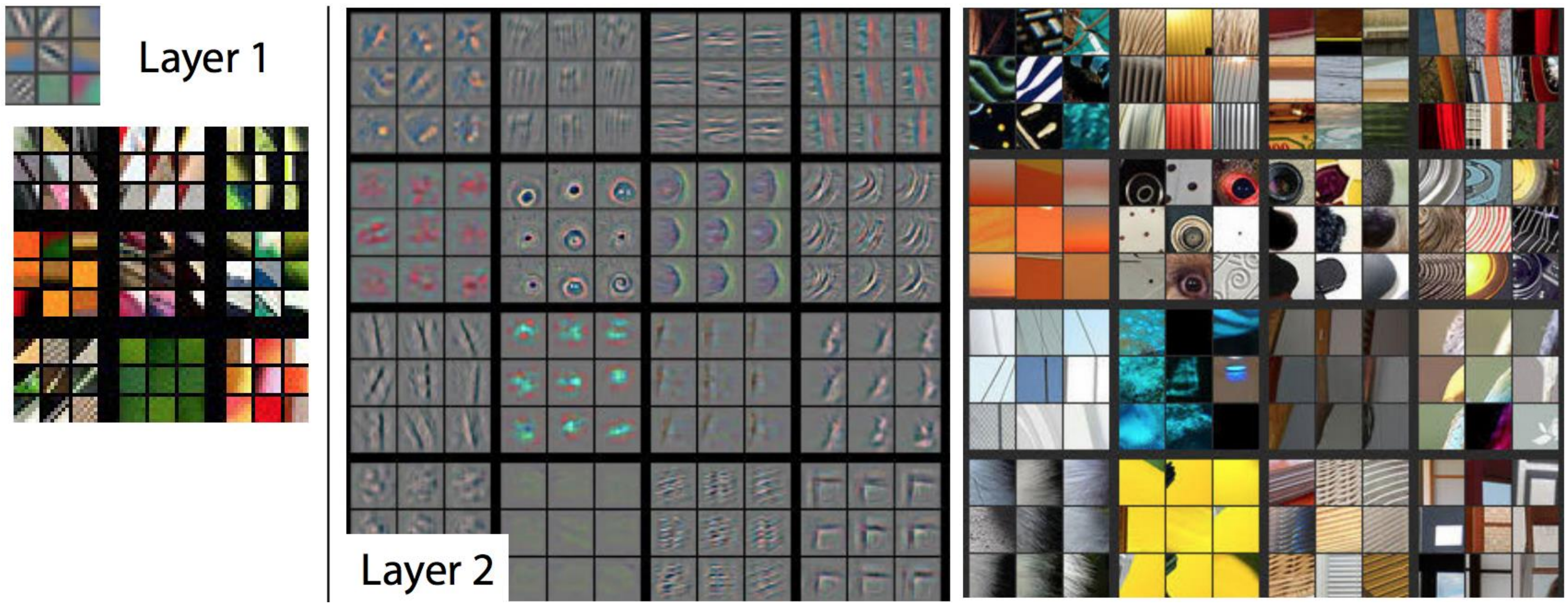
[Yu & Koltun, Multi-Scale Context Aggregation by Dilated Convolutions, ICLR 2016]

Last Lecture



Preventing overfitting: Dropout

Last Lecture



Low-level features: corners, edges, …

Transfer Learning

Today

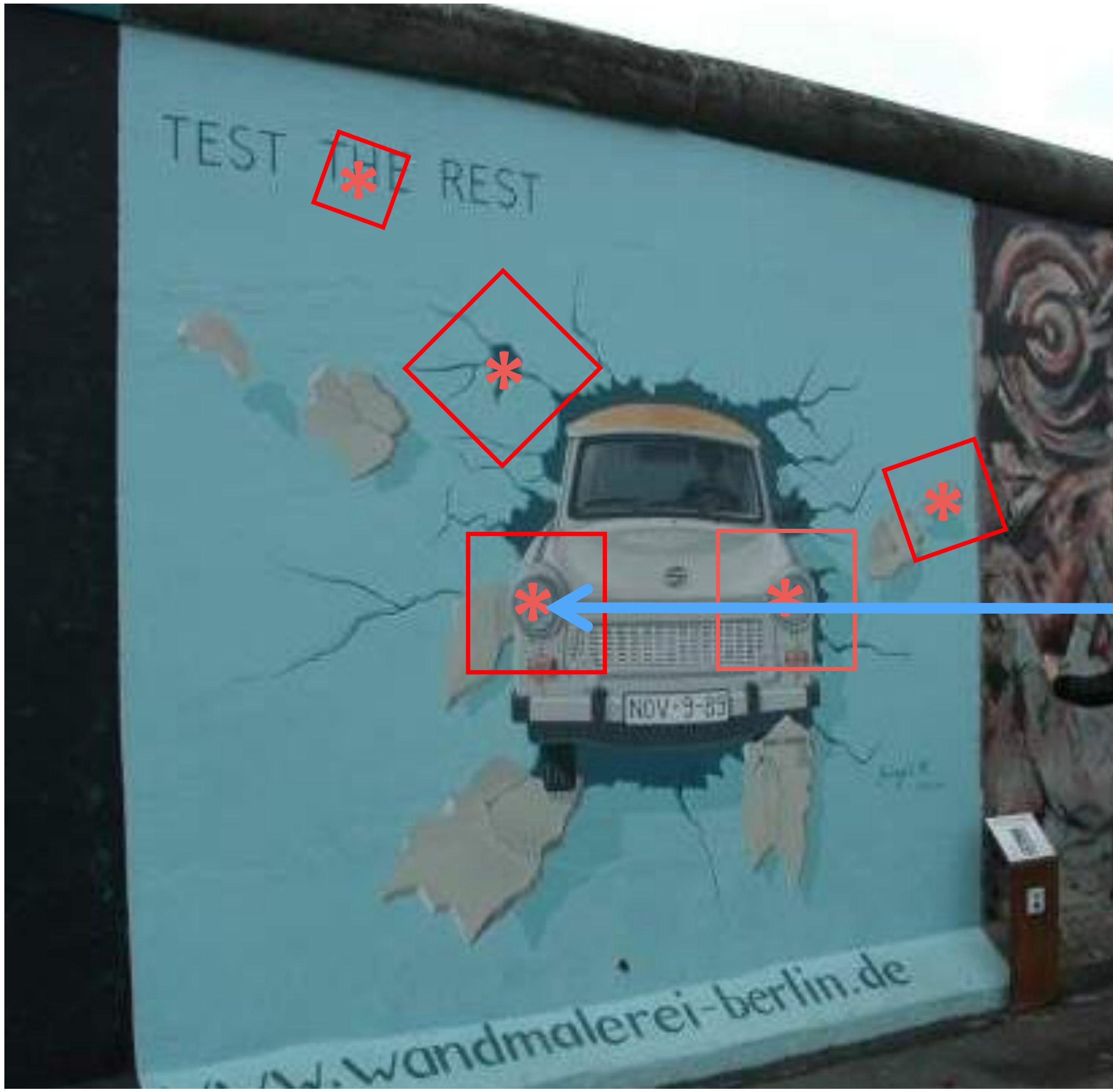
Jan. 20	Introduction, Linear classifiers and filtering	
Jan. 23	Filtering, gradients, scale	Lab 1
Jan. 27	Local features	
Jan. 30	Learning a classifier	Lab 2
Feb. 3	Convolutional neural networks	
Feb. 6	More convolutional neural networks	
Feb. 10	**Robust model fitting and RANSAC**	Lab 3
Feb. 13	Image registration	
Feb. 17	Camera Geometry	Lab 4
Feb. 20	More camera geometry	
Feb. 24	Generative neural networks	
Feb. 27	Generative neural networks	
Mar. 2	TBA	
Mar. 9	TBA	

Today

- Model Fitting Basics
- Robust Model Fitting
- RANdom SAmple Consensus (RANSAC)

Model Fitting Basics

SIFT Features



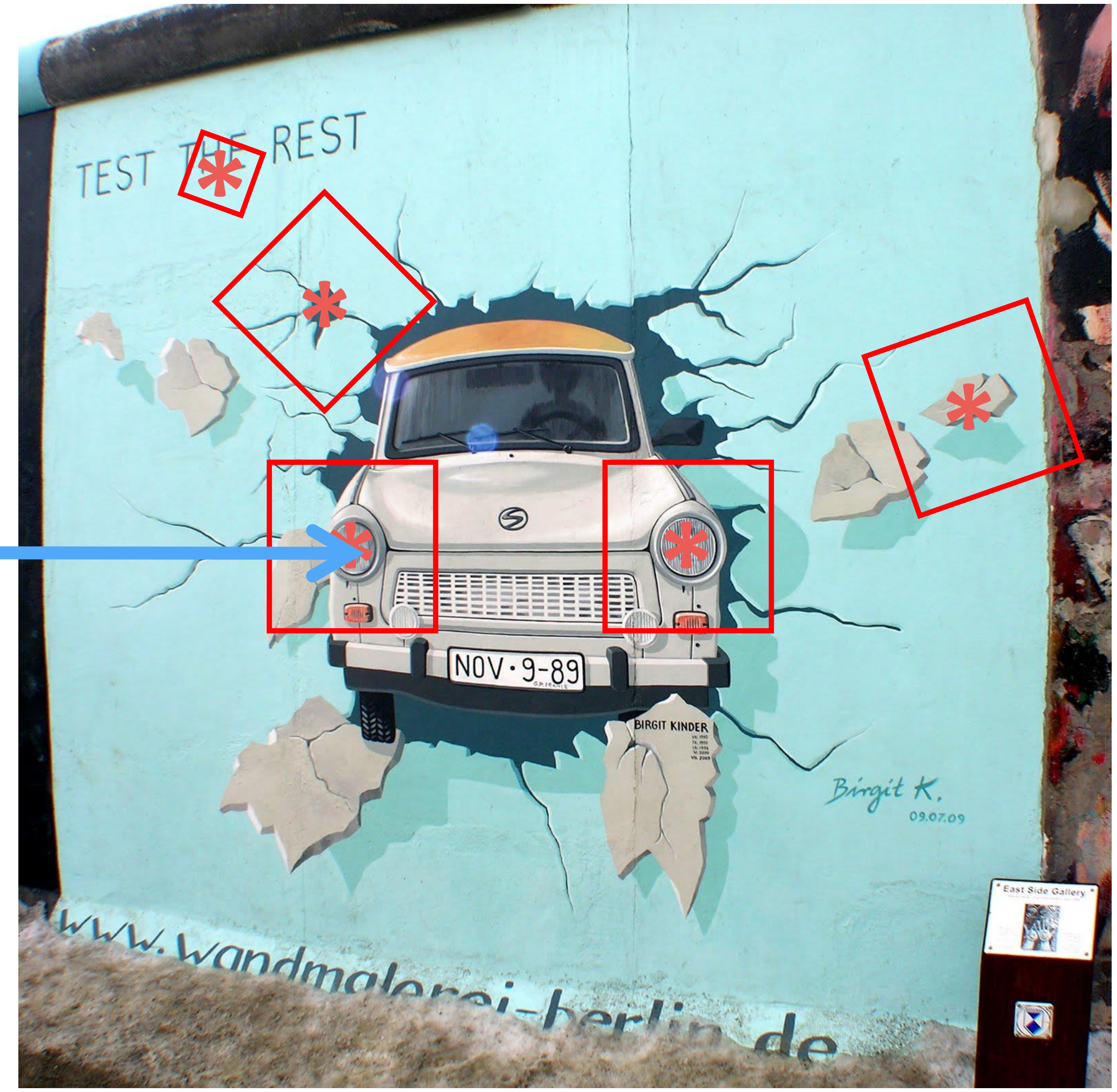


Image Stitching

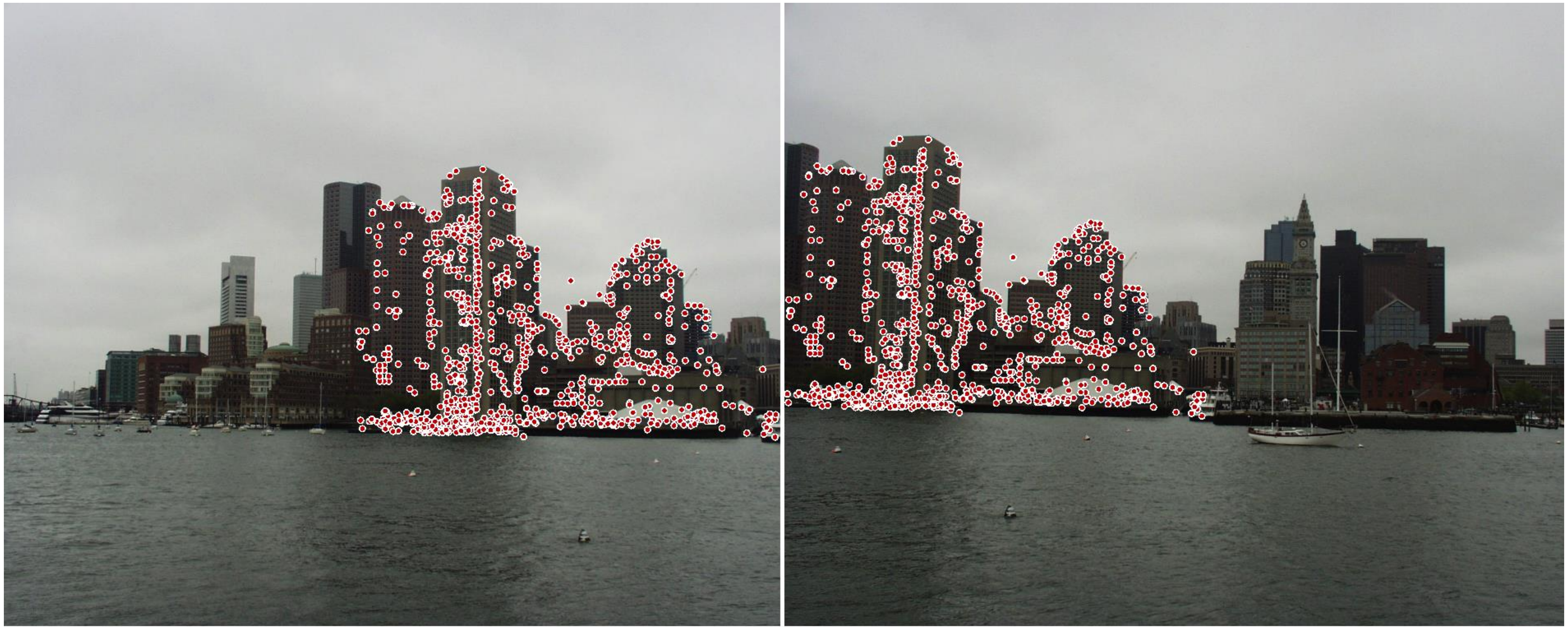
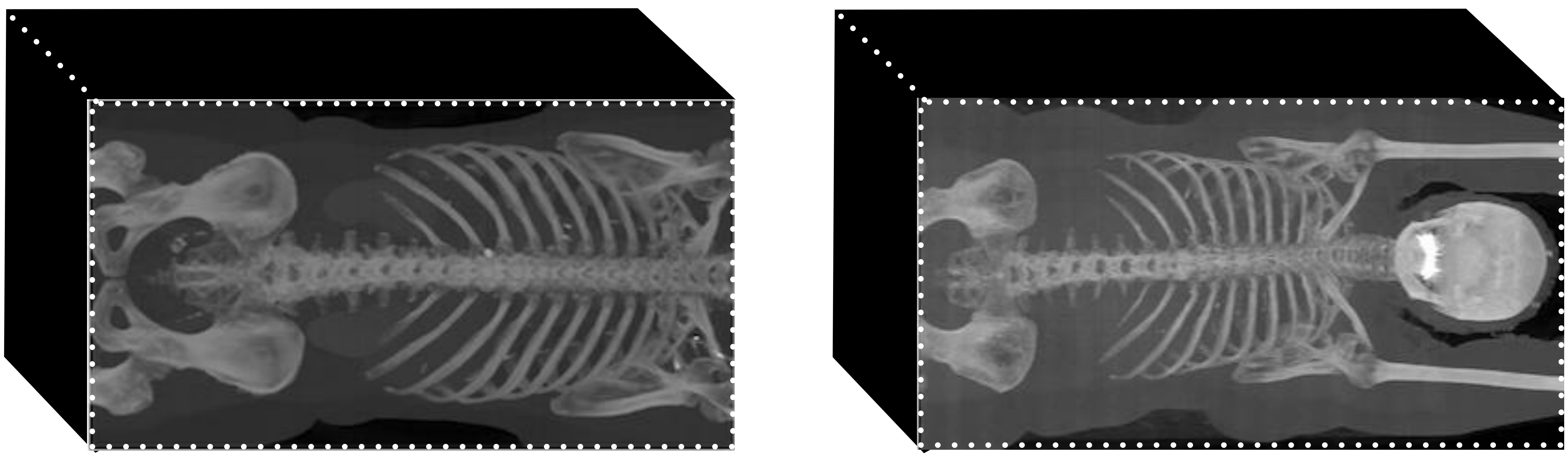


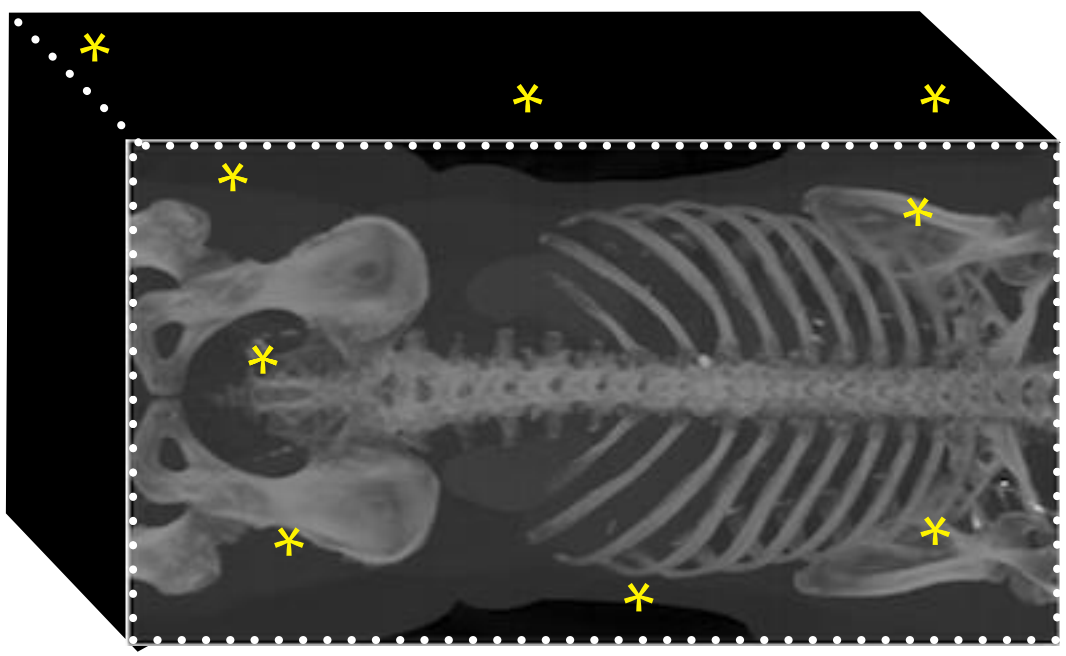
Image Stitching

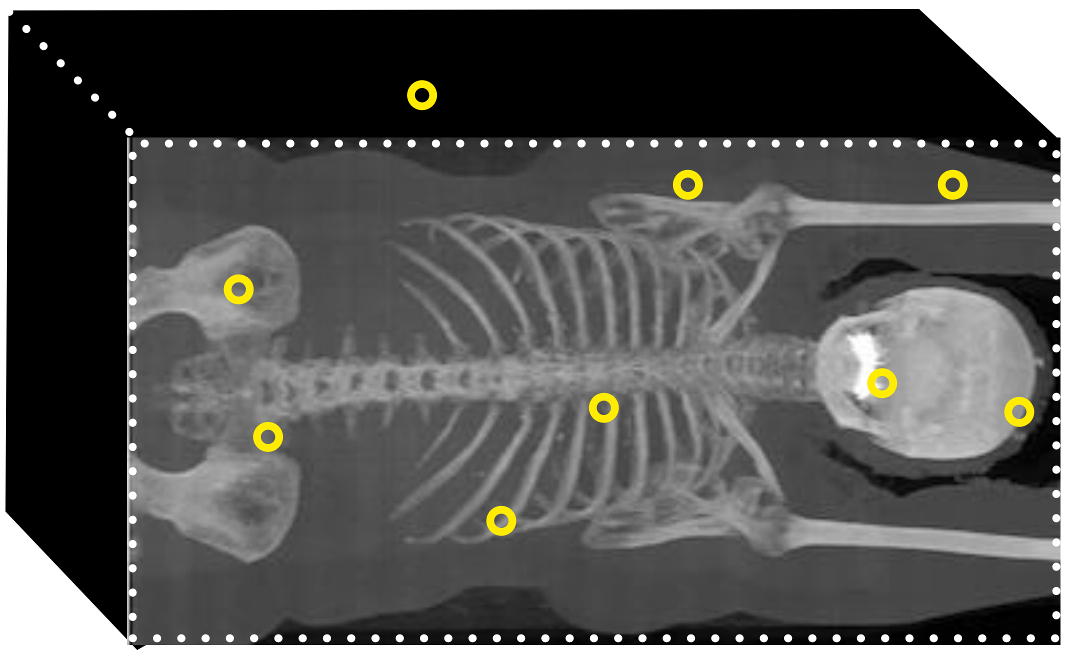
Registration



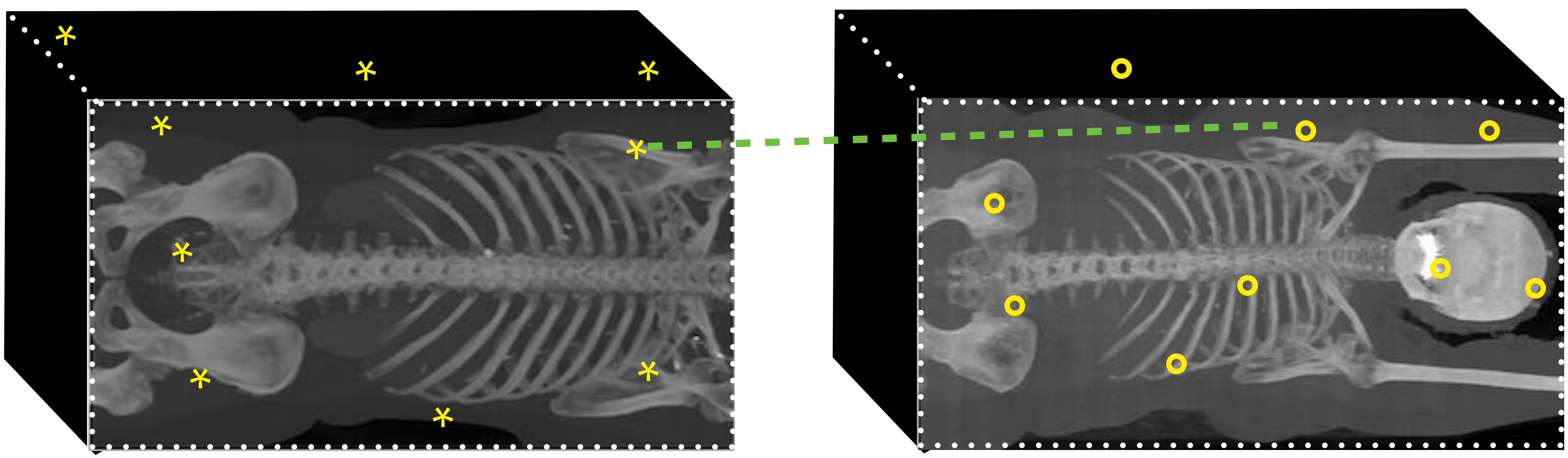
Estimate a transformation

Registration

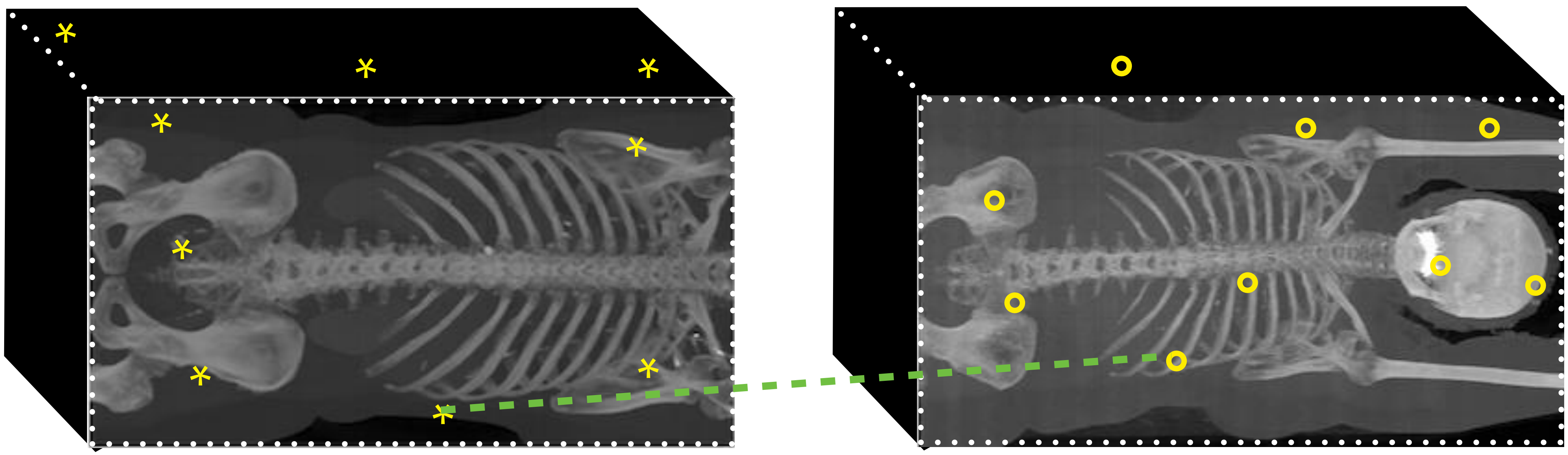




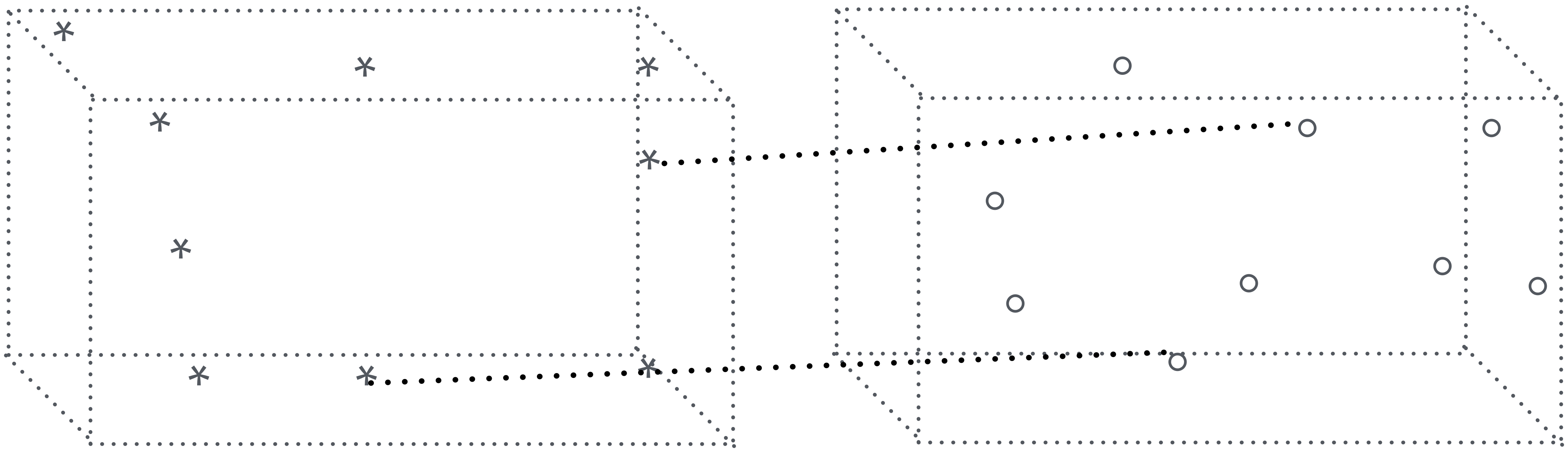
Registration



Registration

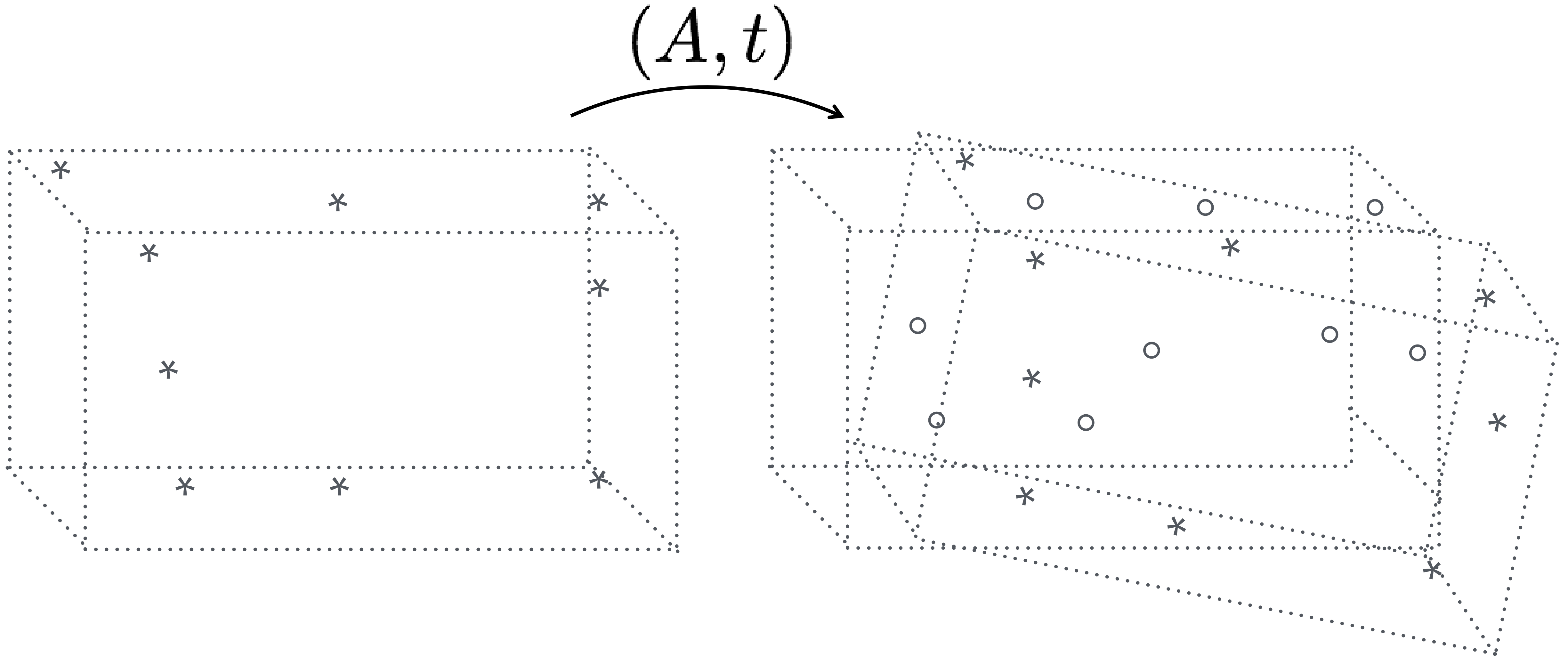


3D Point Set Registration



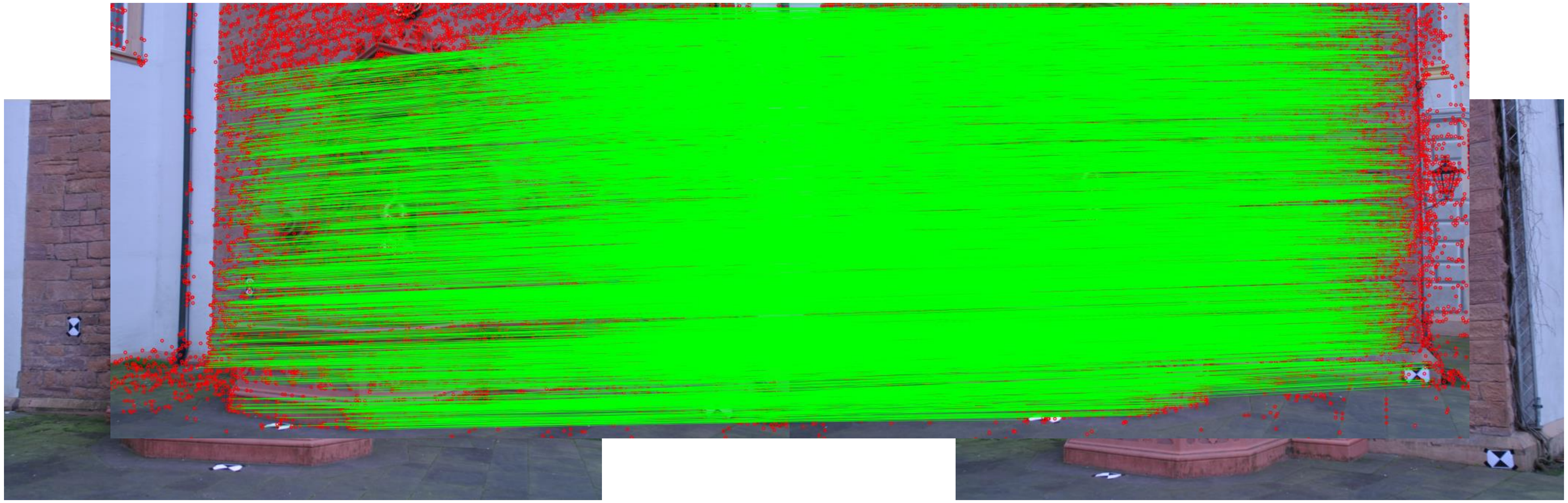
3D Point Set Registration

Find a consistent transformation

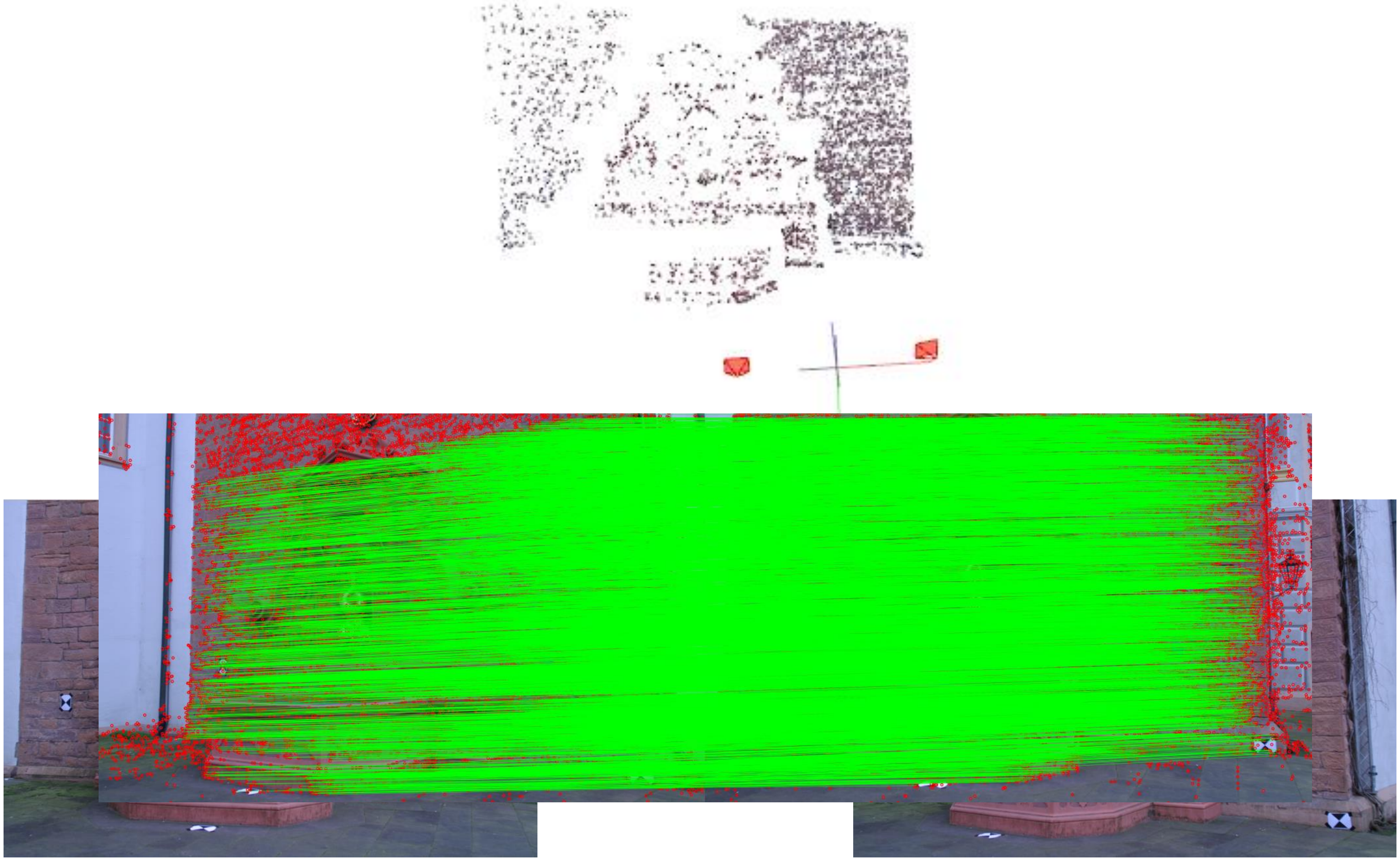


Estimating Camera Motion

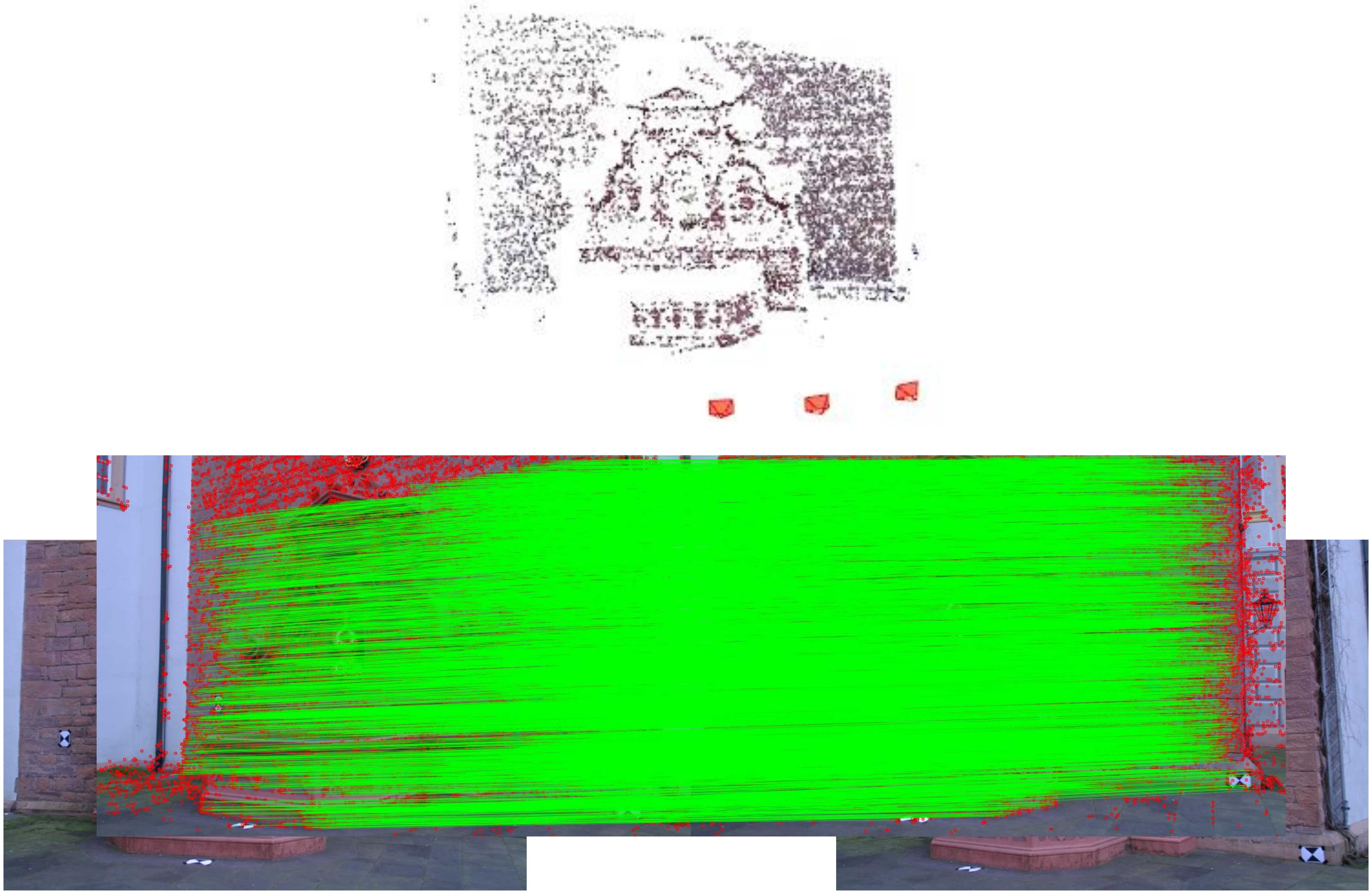
Estimating Camera Motion



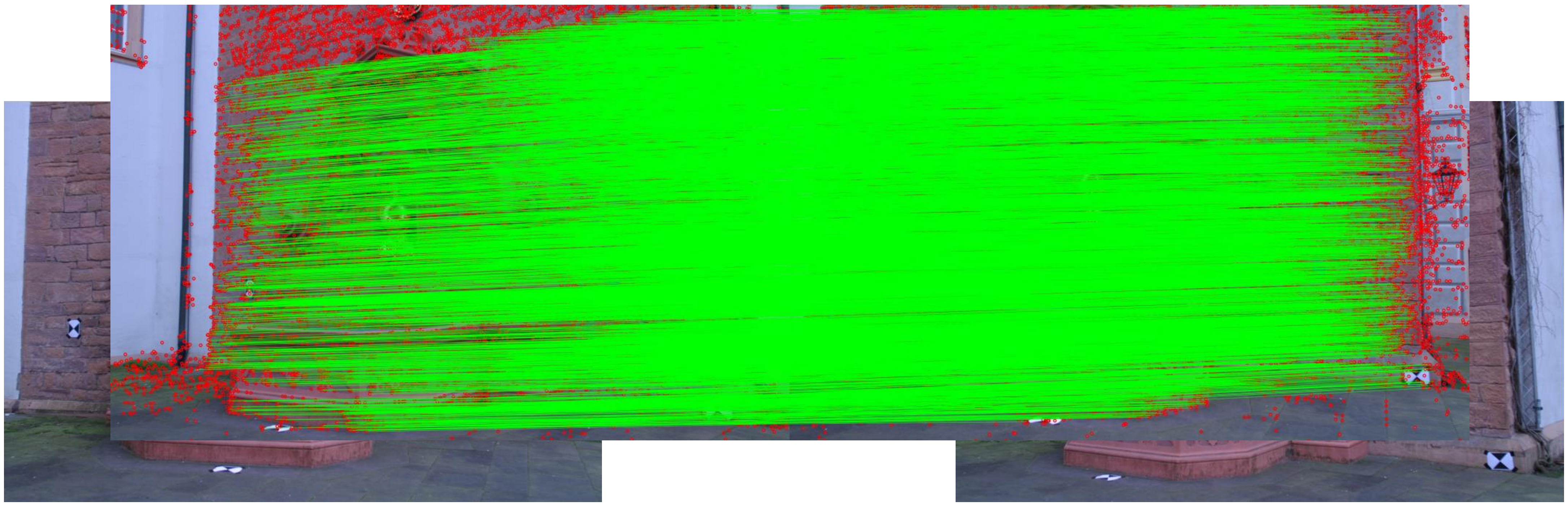
Estimating Camera Motion



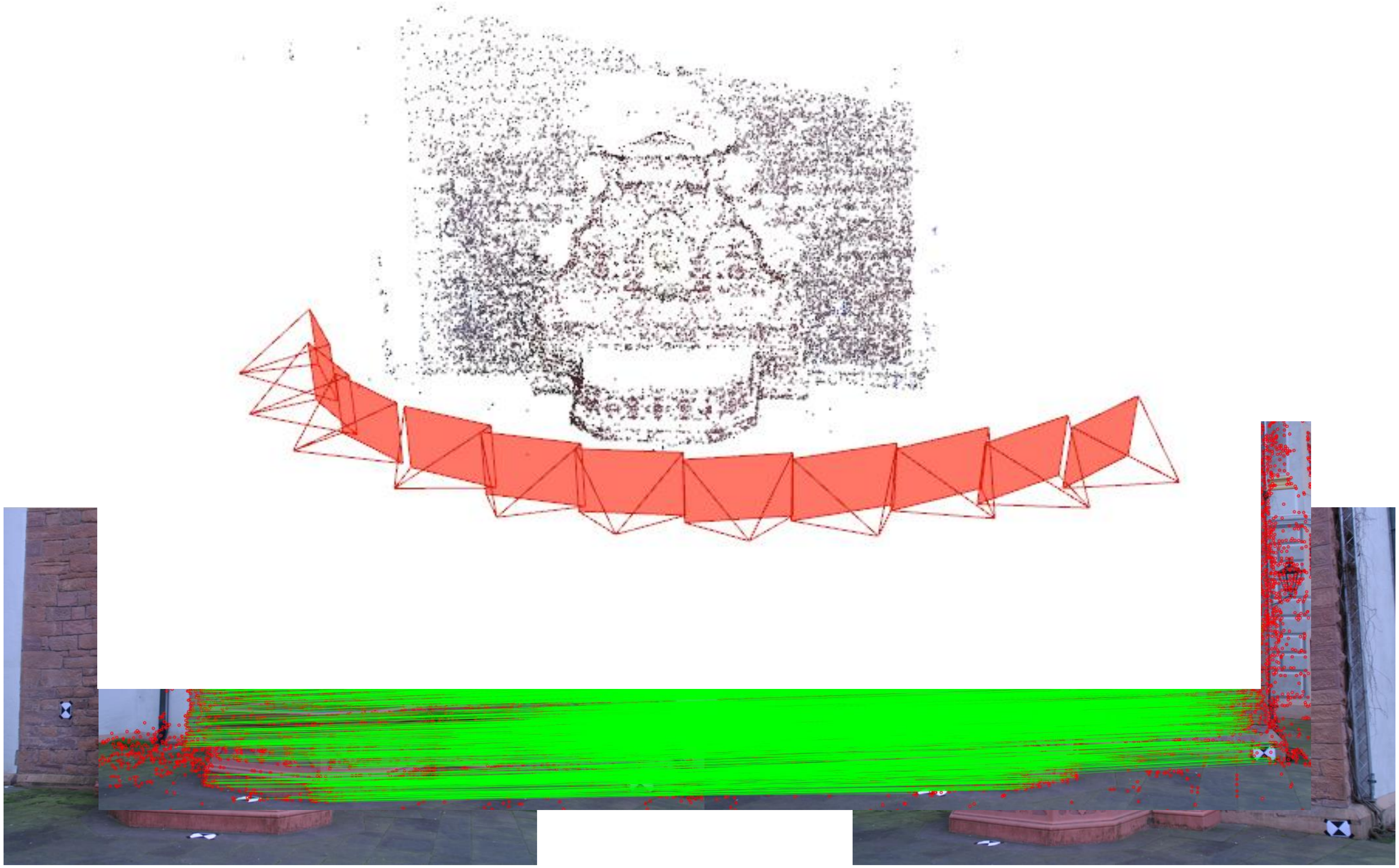
Estimating Camera Motion



Estimating Camera Motion



Estimating Camera Motion

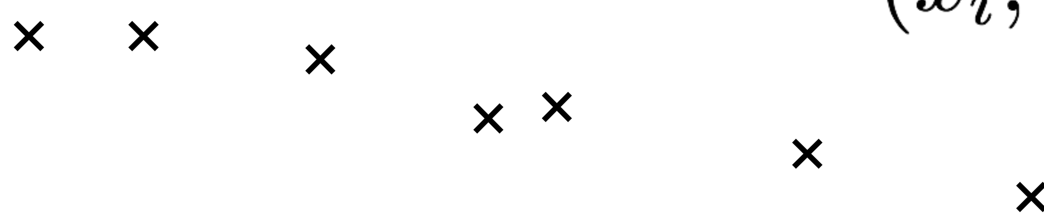


More in lecture 10

Measurements - Models - Parameters

Measurements

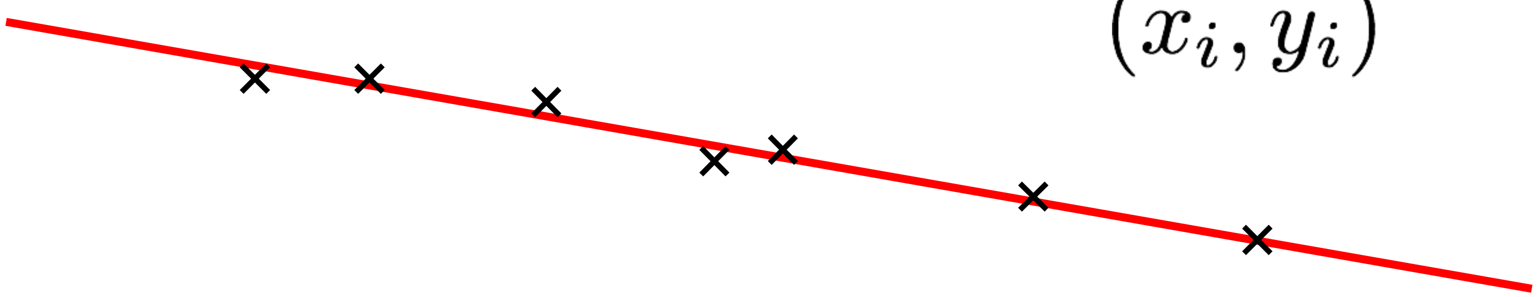
(x_i, y_i)



Measurements - Models - Parameters

Measurements

(x_i, y_i)

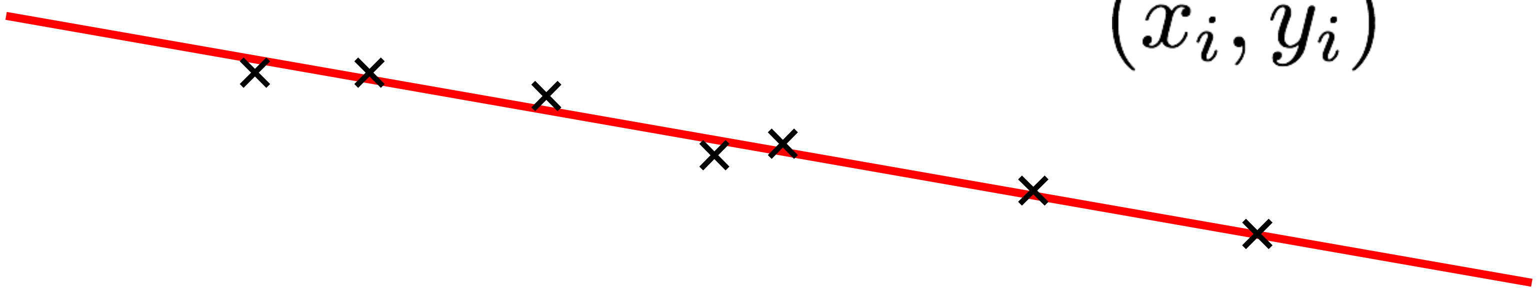


Model: Points on a line: $y_i = kx_i + m$

Measurements - Models - Parameters

Measurements

(x_i, y_i)

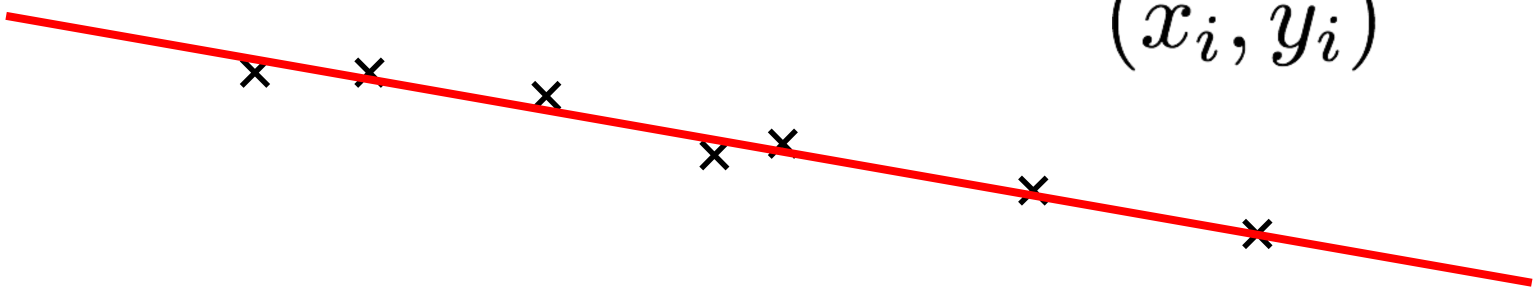


Model: Points on a line: $y_i = kx_i + m$ (explicit line representation)

Measurements - Models - Parameters

Measurements

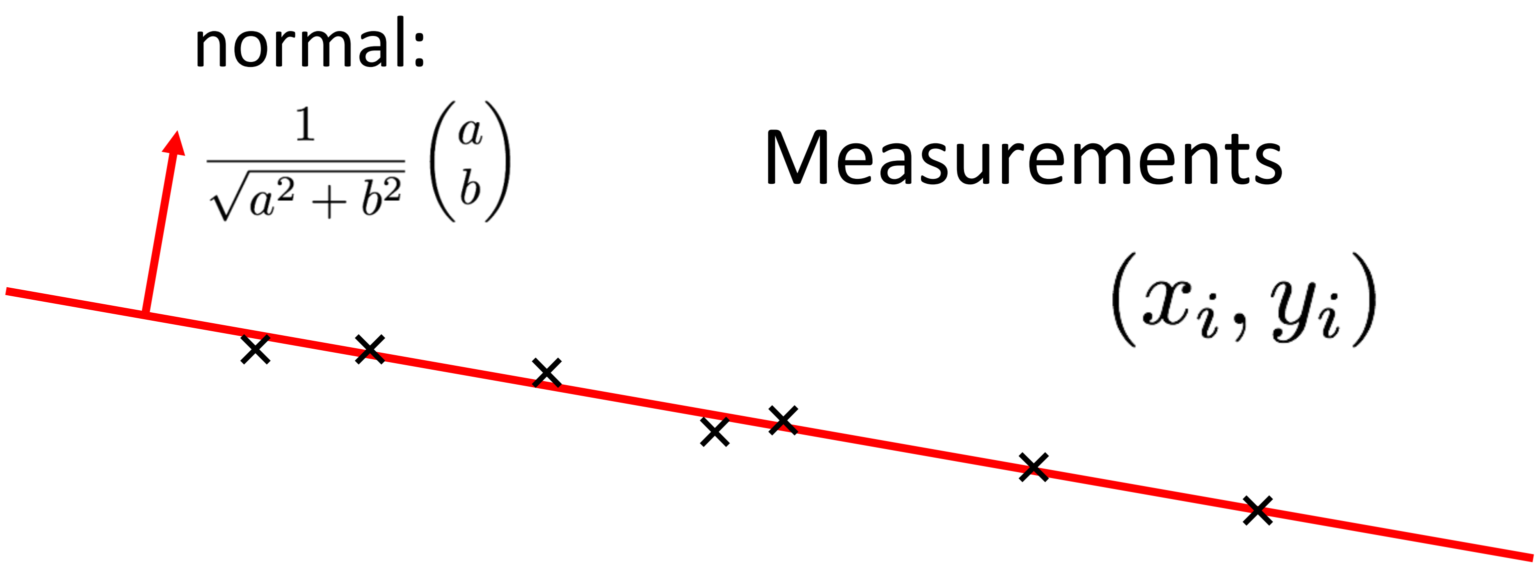
(x_i, y_i)



Model: Points on a line: $y_i = kx_i + m$ (explicit line representation)

Good formulation: $ax_i + by_i + c = 0$ (implicit line representation)

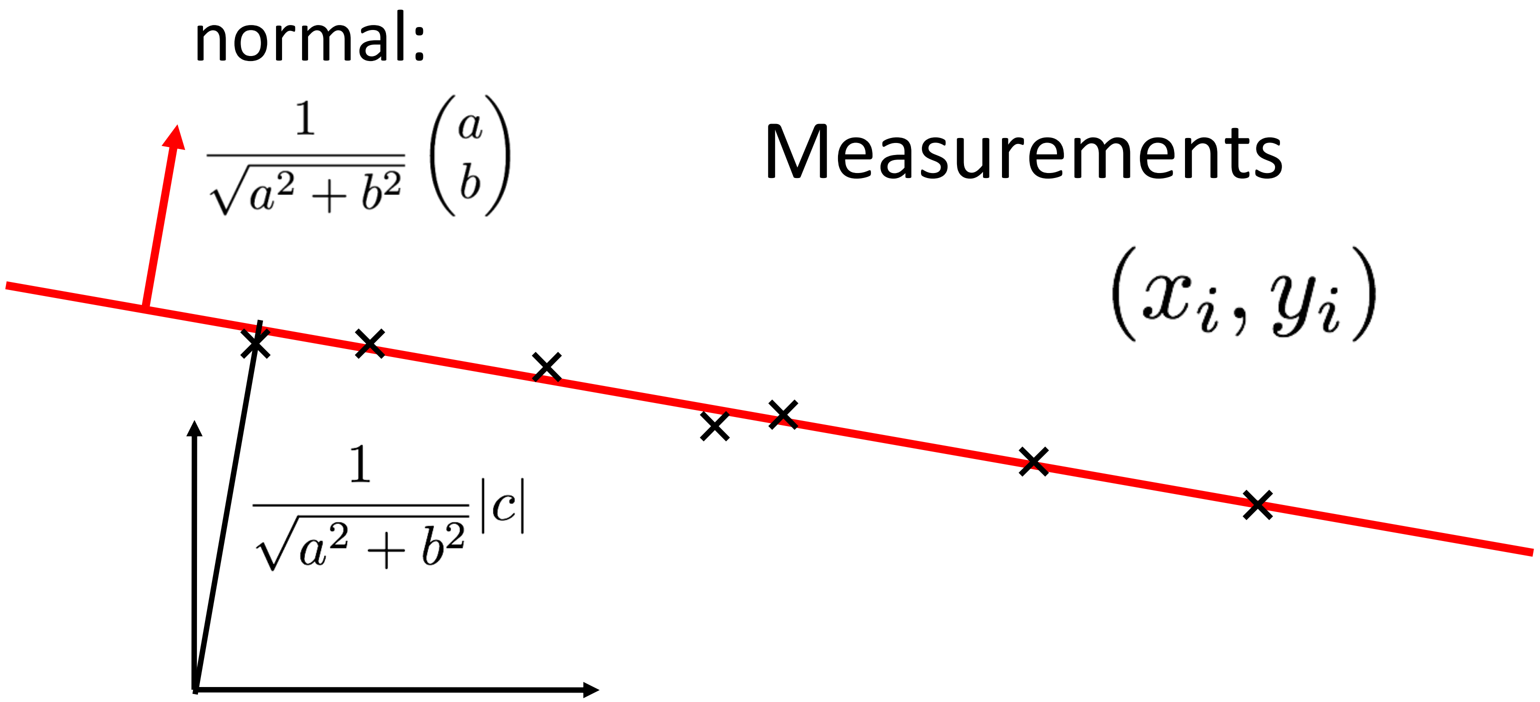
Measurements - Models - Parameters



Model: Points on a line: $y_i = kx_i + m$ (explicit line representation)

Good formulation: $ax_i + by_i + c = 0$ (implicit line representation)

Measurements - Models - Parameters



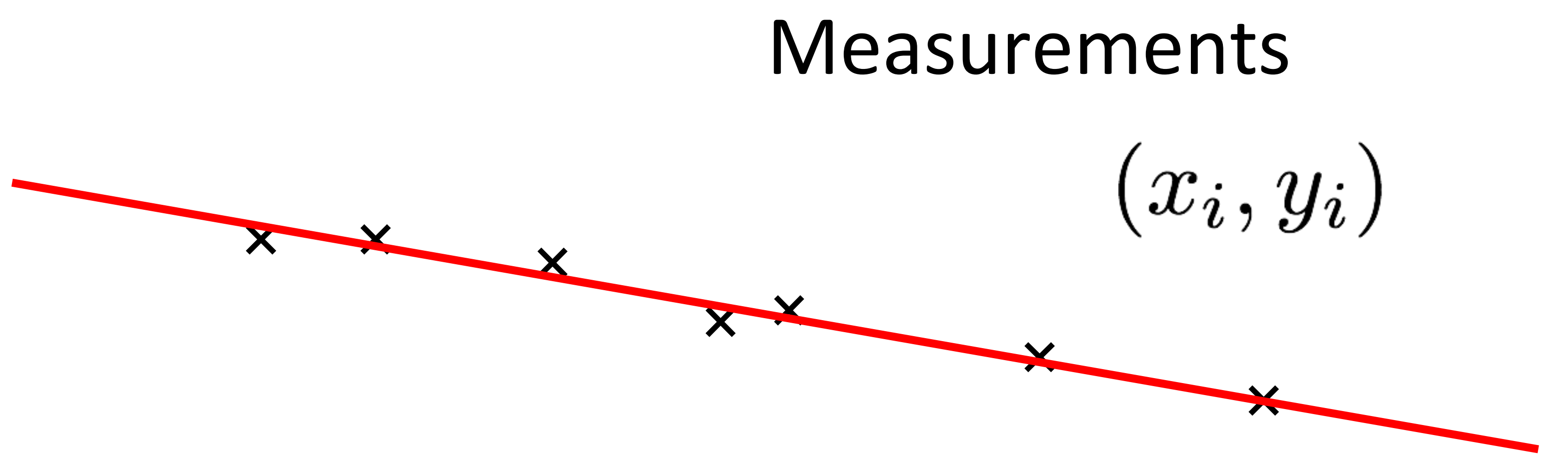
Measurements

$$(x_i, y_i)$$

Model: Points on a line: $y_i = kx_i + m$ (explicit line representation)

Good formulation: $ax_i + by_i + c = 0$ (implicit line representation)

Measurements - Models - Parameters



Model: Points on a line: $y_i = kx_i + m$ (explicit line representation)

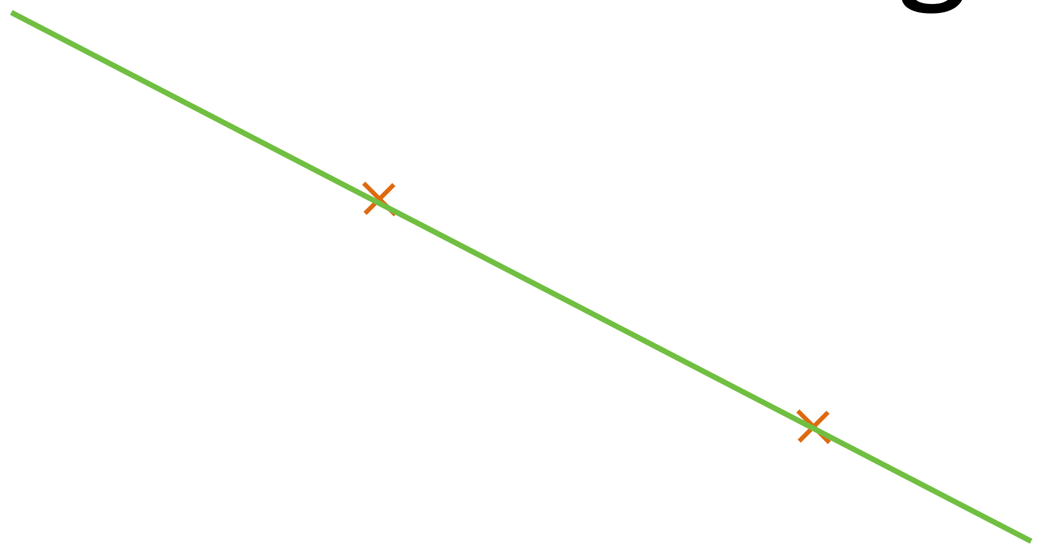
Good formulation: $ax_i + by_i + c = 0$ (implicit line representation)

Model fitting : Estimate a, b and c from measurements

$$\theta = (a, b, c)$$

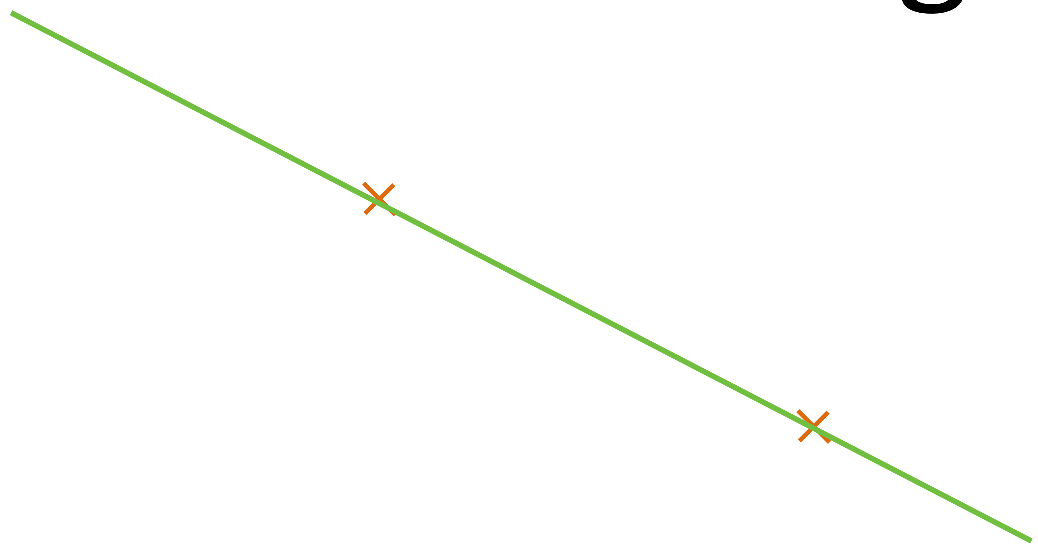
Line Fitting

Two measurements - exact solution



Line Fitting

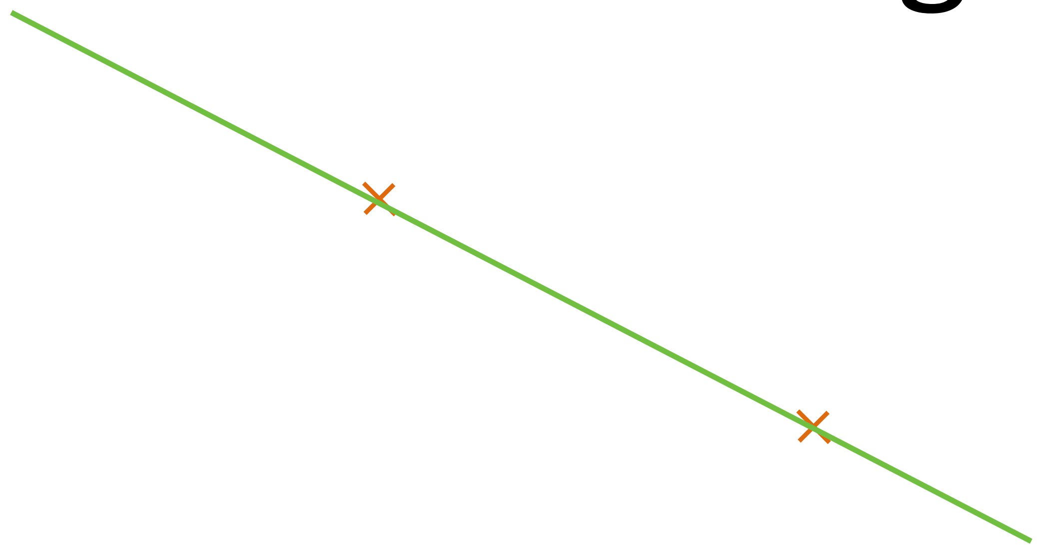
Two measurements - exact solution



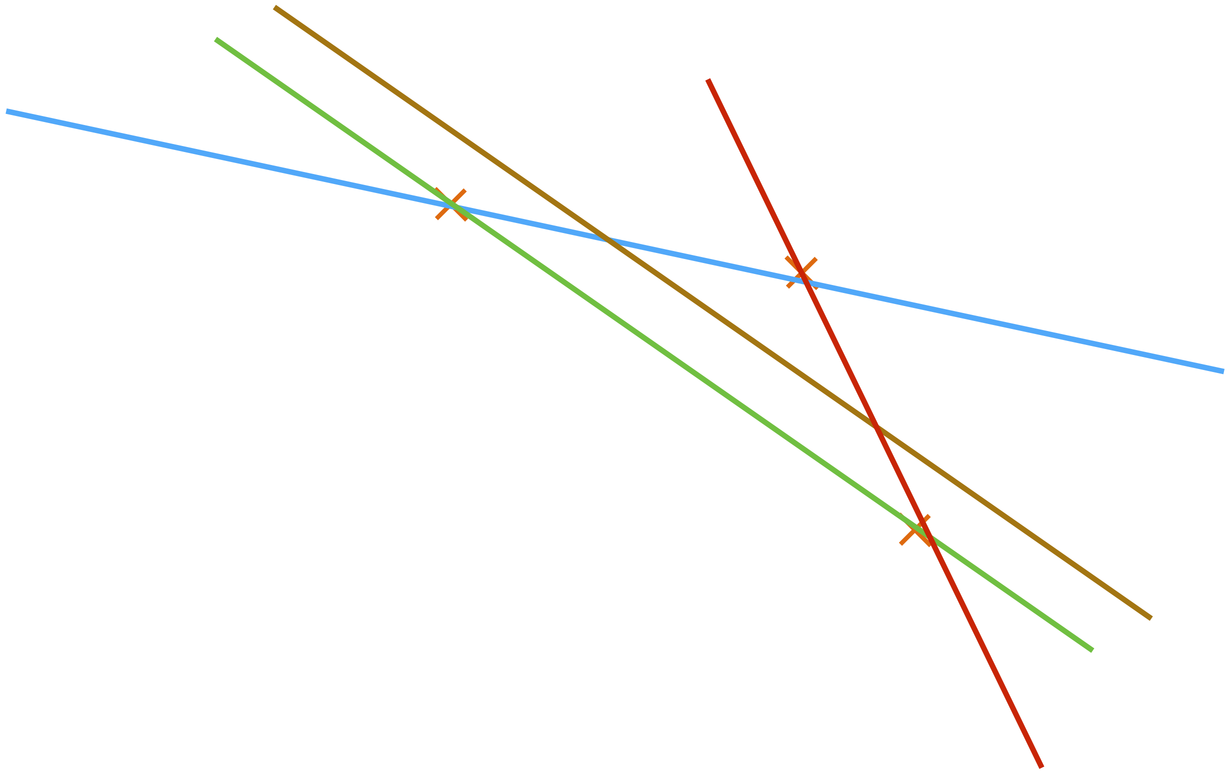
More than two?

Line Fitting

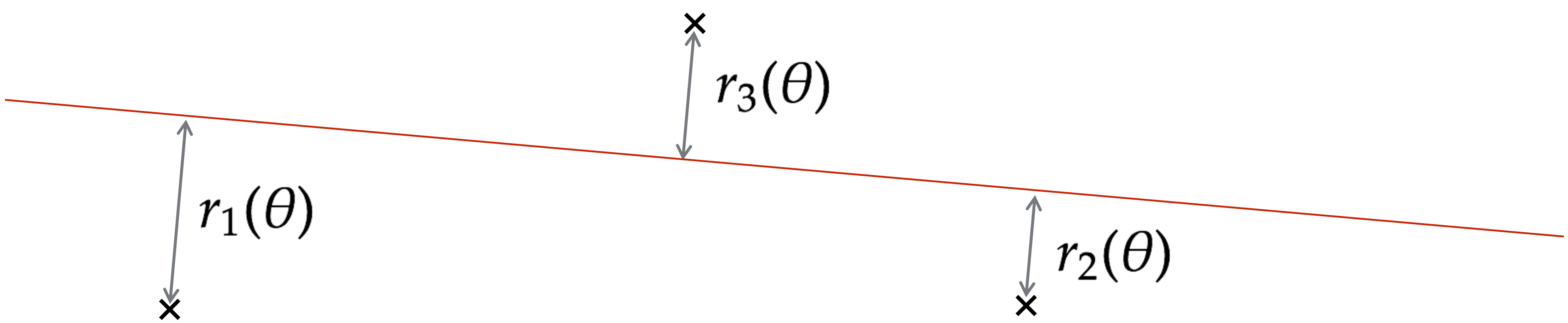
Two measurements - exact solution



More than two?



Residuals



$$r_i(\theta) = |ax + by + c| \text{ such that } a^2 + b^2 = 1$$

Model Fitting Problem

$$\theta = (a, b, c) \longrightarrow r_1(\theta), r_2(\theta), r_3(\theta), r_4(\theta), \ldots, r_n(\theta)$$

Which parameters best explain the residuals?

Why Do We Have Residuals?

Measurements are not exact.
They are affected by Gaussian noise!

Carl Friedrich Gauss

Why Do We Have Residuals?

Measurements are not exact.
They are affected by Gaussian noise!

Best thing to do: Least squares!

$$\min_{\theta} \sum r_i(\theta)^2$$

Carl Friedrich Gauss

Minimization under Gaussian Noise

- Assumption: Zero-mean, isotropic Gaussian noise on residuals

$$p(r_i(\theta)) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{r_i(\theta)^2}{2\sigma^2}}$$

Minimization under Gaussian Noise

- Assumption: Zero-mean, isotropic Gaussian noise on residuals

$$p(r_i(\theta)) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{r_i(\theta)^2}{2\sigma^2}}$$

- Maximum Likelihood Estimate: $\max\limits_{\theta} \prod\limits_{i} p(r_i(\theta))$

Minimization under Gaussian Noise

- Assumption: Zero-mean, isotropic Gaussian noise on residuals

$$p(r_i(\theta)) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{r_i(\theta)^2}{2\sigma^2}}$$

- Maximum Likelihood Estimate: $\max_\theta \prod_i p(r_i(\theta))$

- Minimizing negative log-likelihood: $\min_\theta - \sum_i \log(p(r_i(\theta)))$

Minimization under Gaussian Noise

- Assumption: Zero-mean, isotropic Gaussian noise on residuals

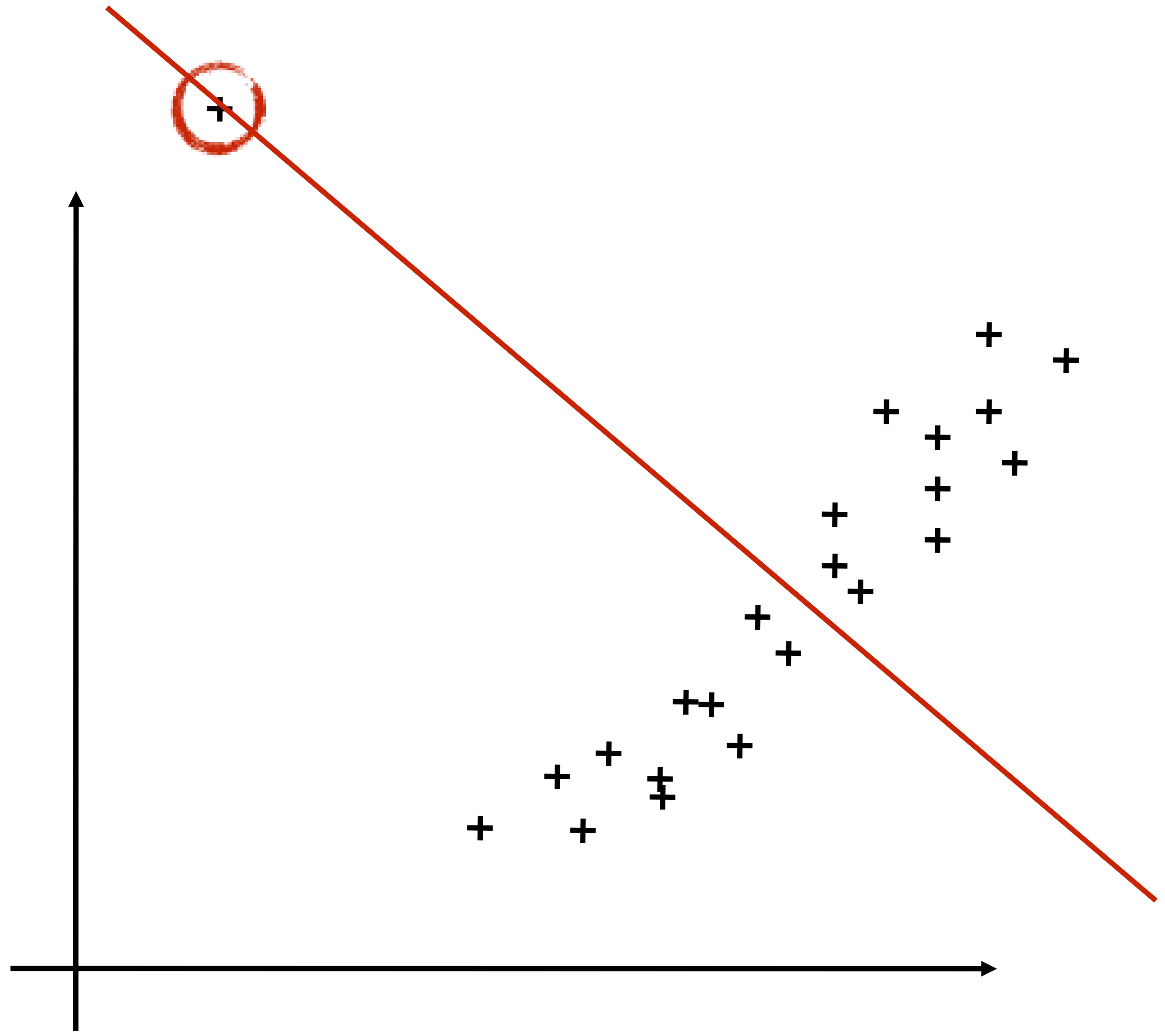
$$p(r_i(\theta)) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{r_i(\theta)^2}{2\sigma^2}}$$

- Maximum Likelihood Estimate: $\max_\theta \prod_i p(r_i(\theta))$

- Minimizing negative log-likelihood: $\min_\theta - \sum_i \log(p(r_i(\theta)))$

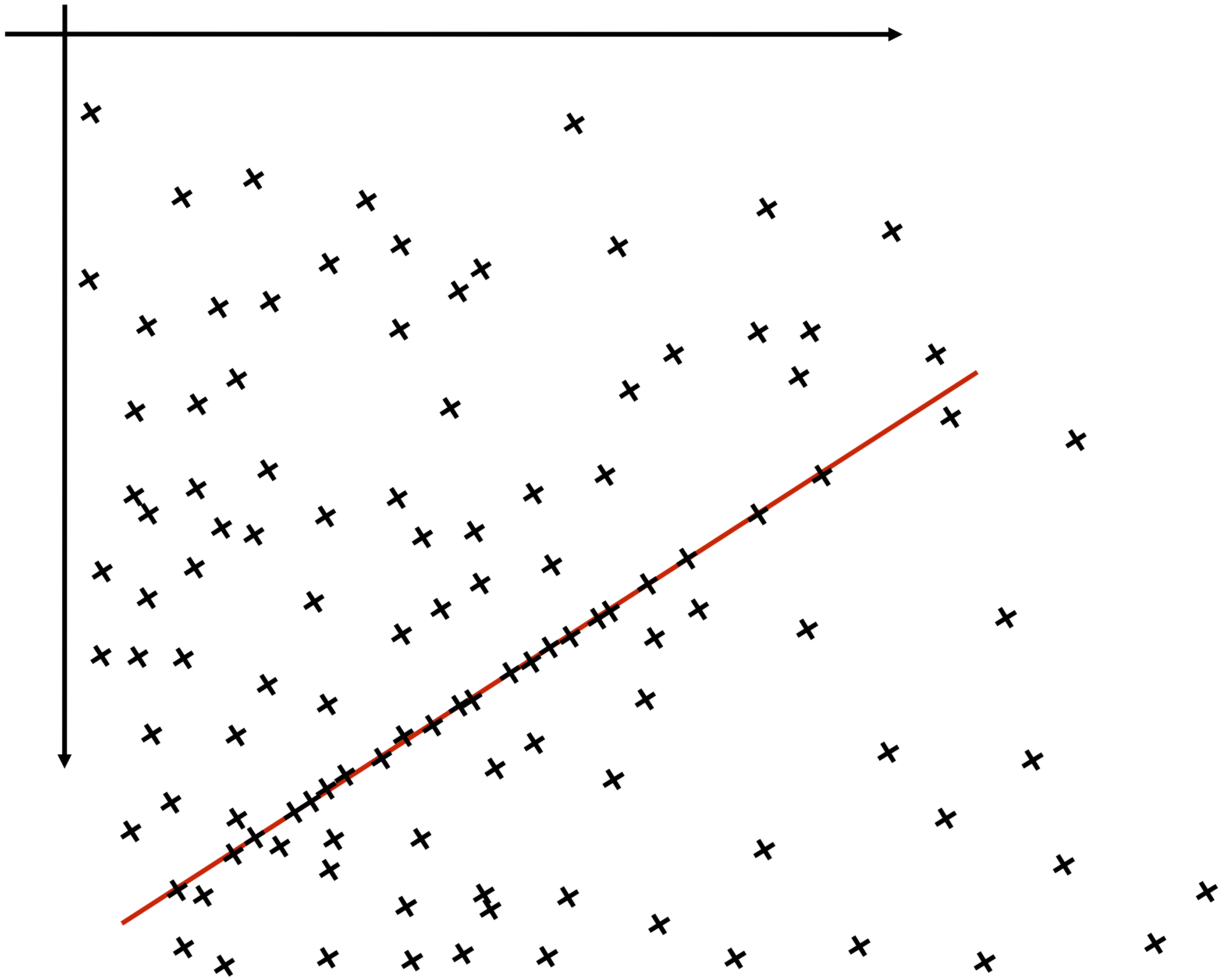
- Equivalent to least-squares fitting: $\min_\theta \sum_i r_i(\theta)^2$

Outliers



Rare unexpected measurements that don’t fit this model.

Outliers in Image Analysis



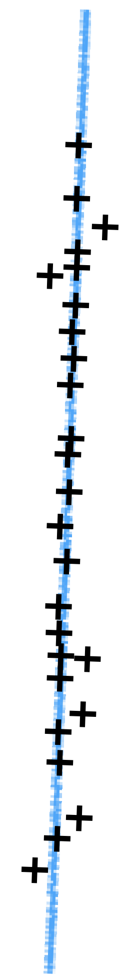
Frequent expected measurements that don't give useful information.

Outliers in Image Analysis

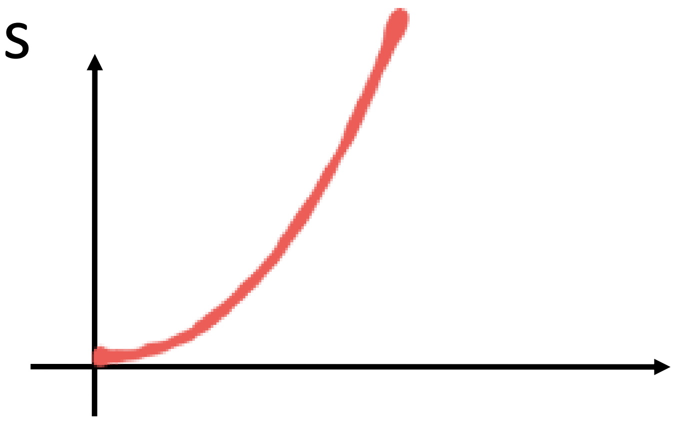
Are We Doing Something Wrong?

Robust Model Fitting

Least-Squares Fitting

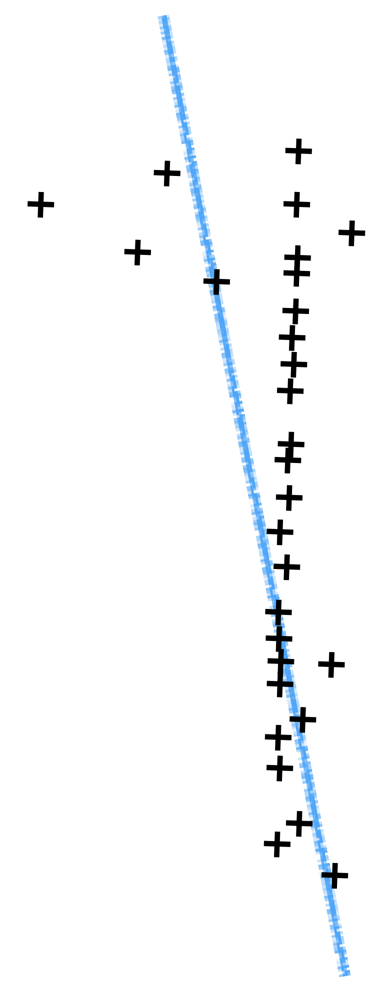


ℓ^2-solution

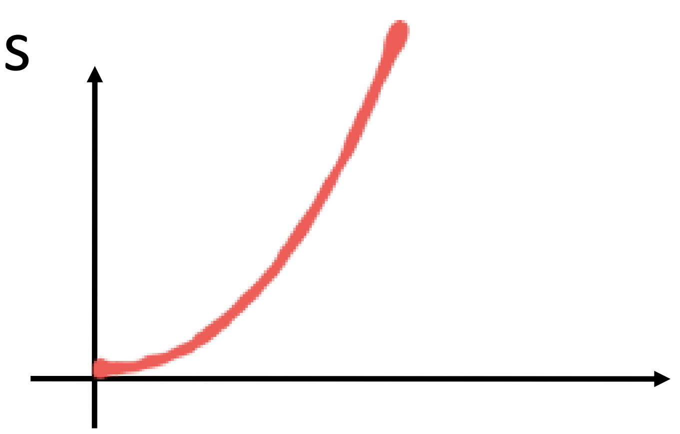


$$\min_{\theta} \sum r_i(\theta)^2$$

Least-Squares Fitting

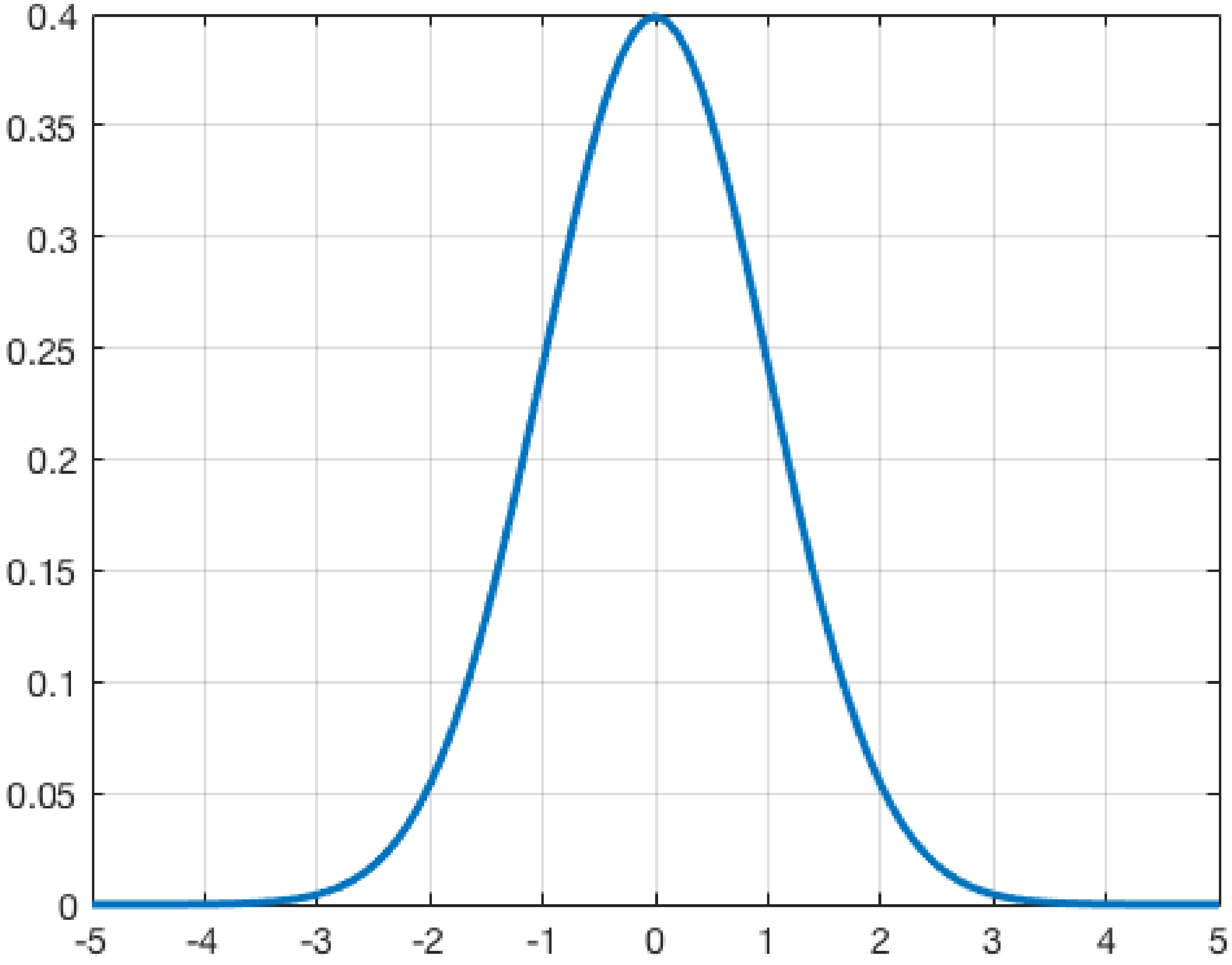


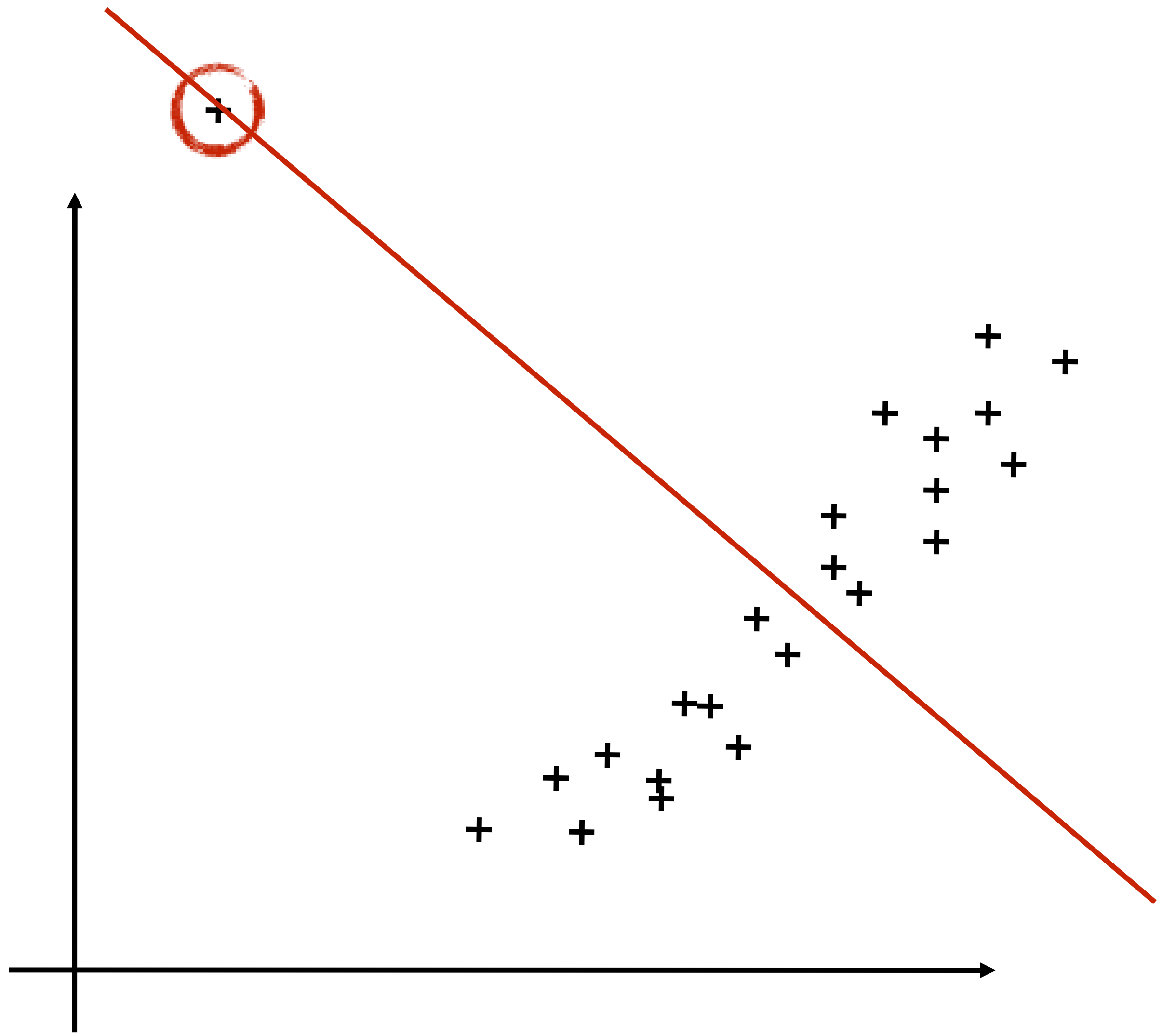
ℓ^2-solution



$$\min_\theta \sum r_i(\theta)^2$$

Minimization under Gaussian Noise

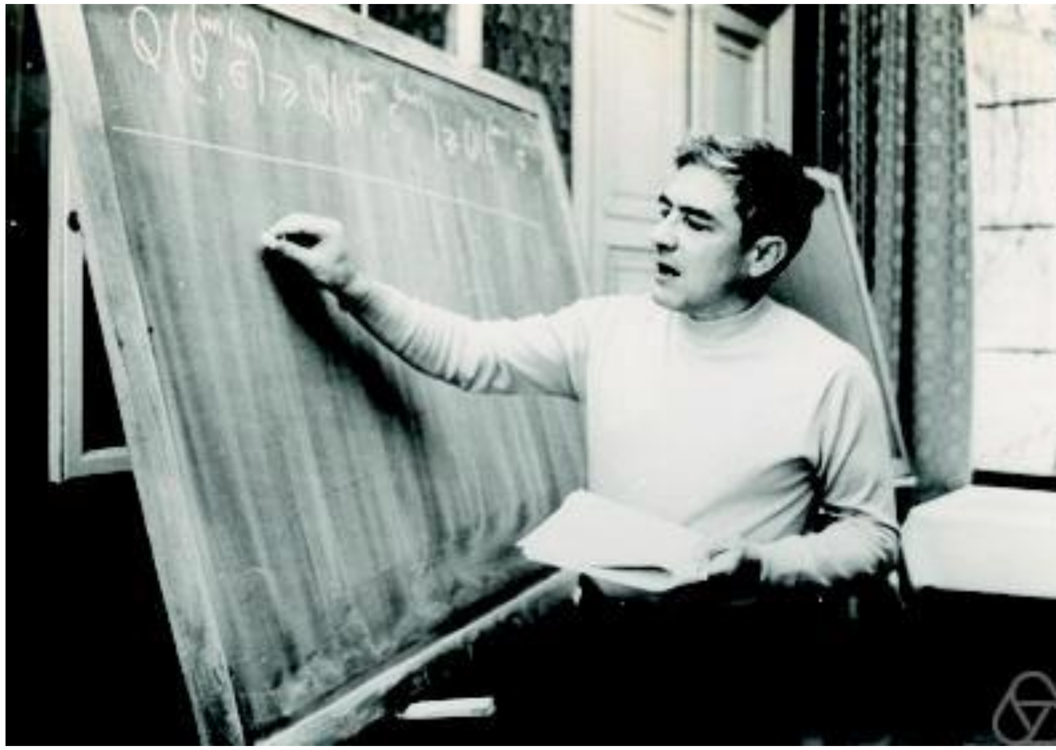




Rare unexpected measurements that don’t fit Gaussian noise model.

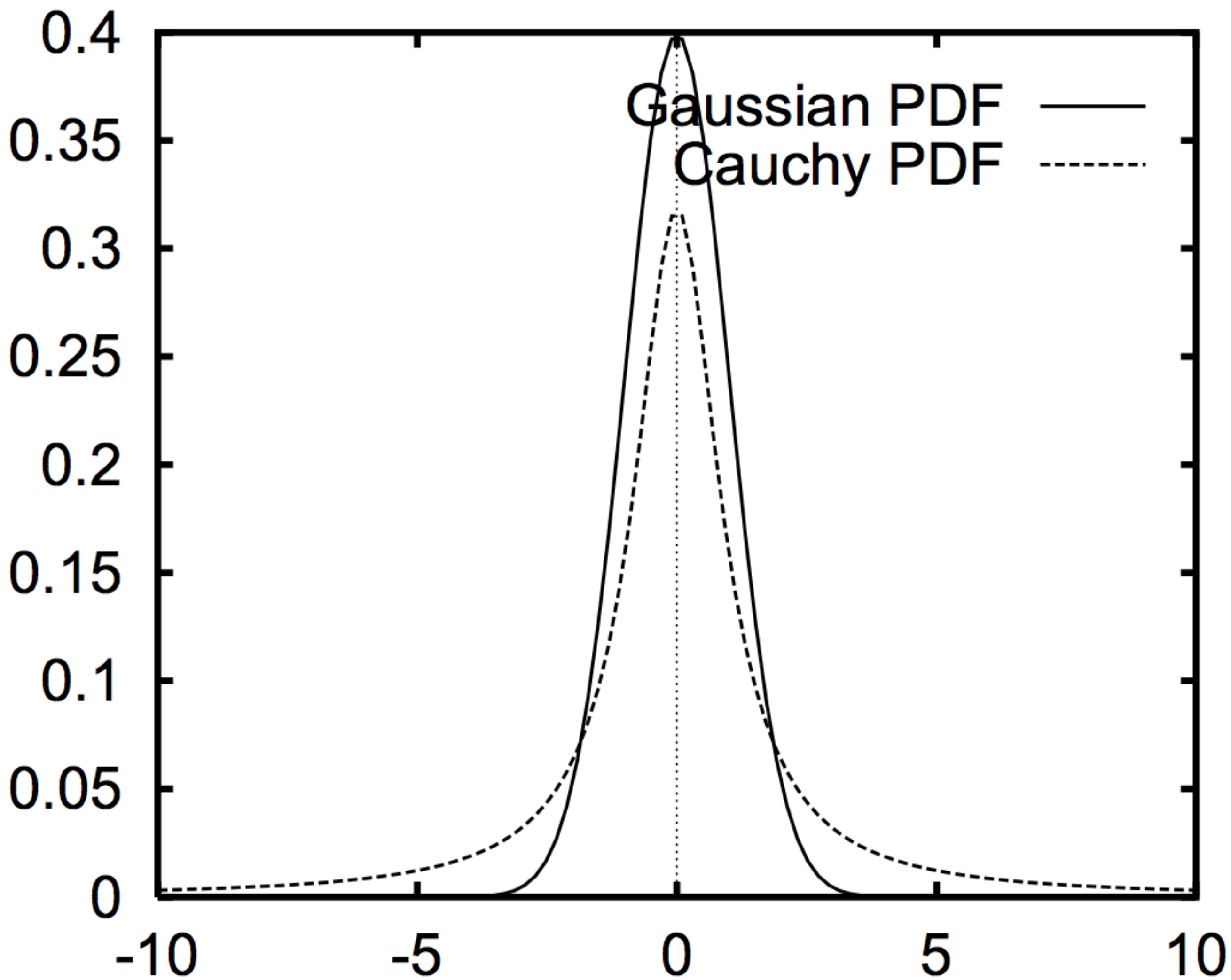
Why Do We Have Residuals?

There are all kinds of noise!
Use robust loss functions!



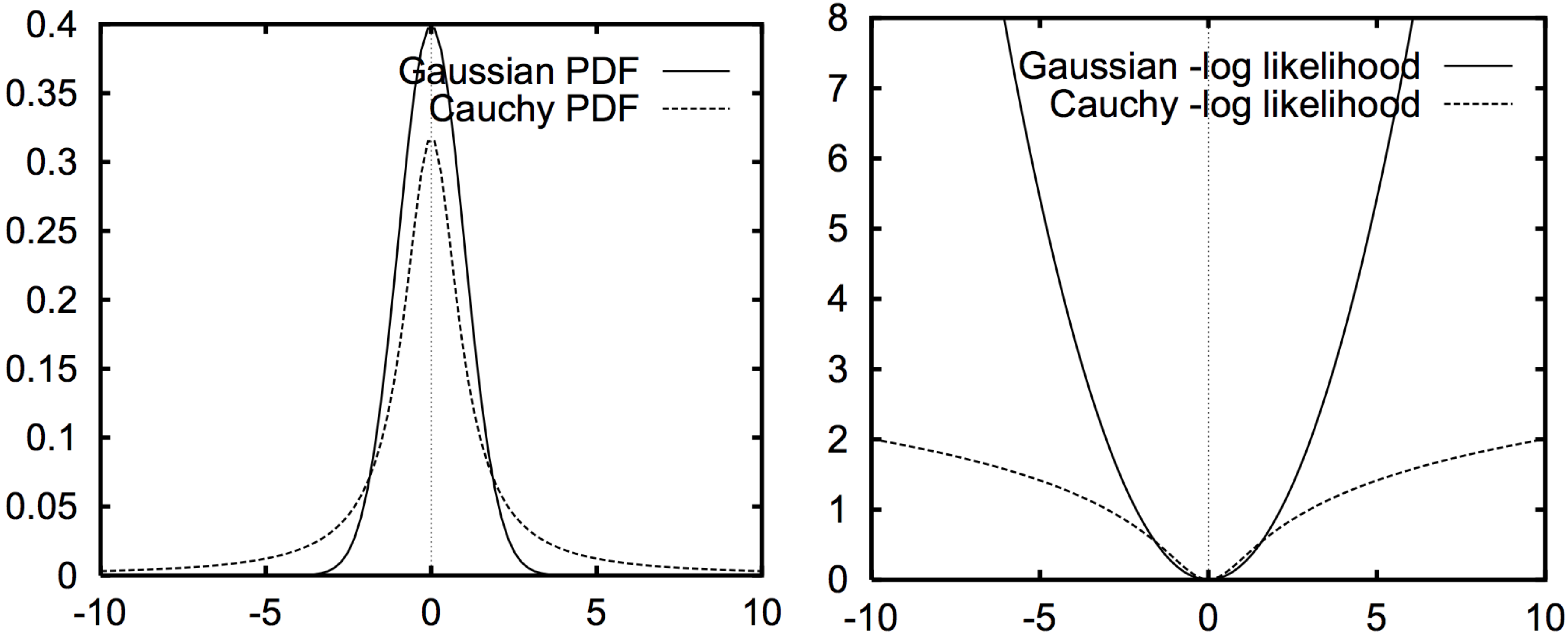
Peter Huber

Robust Loss Functions



Cauchy distribution: $$p(x) = \frac{1}{\pi(1+x^2)}$$

Robust Loss Functions



Cauchy distribution: $$p(x) = \frac{1}{\pi(1+x^2)}$$

Minimizing Robust Loss Functions

- Replace

$$\min_\theta \sum_i r_i(\theta)^2$$

- with

$$\min_\theta \sum_i f(|r_i(\theta)|)$$

Minimizing Robust Loss Functions

- Replace

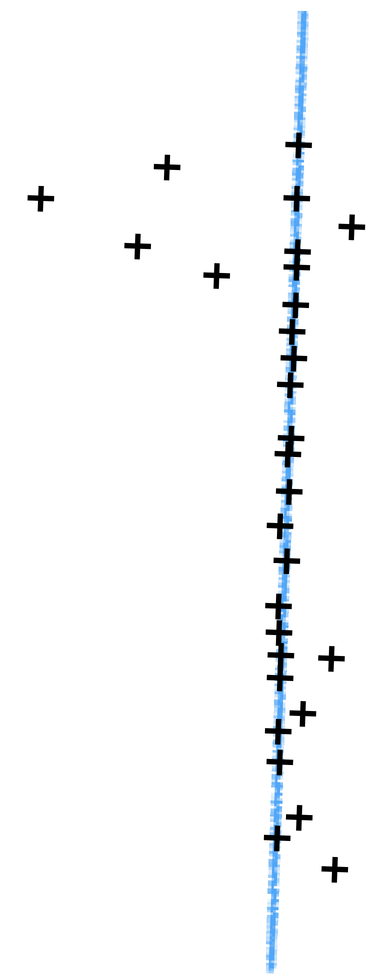
$$\min_\theta \sum_i r_i(\theta)^2$$

- with

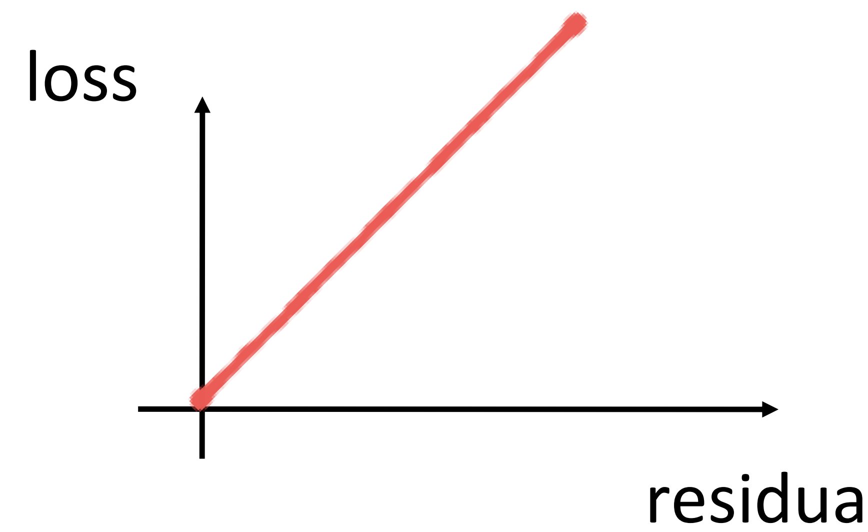
$$\min_\theta \sum_i f(|r_i(\theta)|)$$

robust loss function

Least Absolute Residual

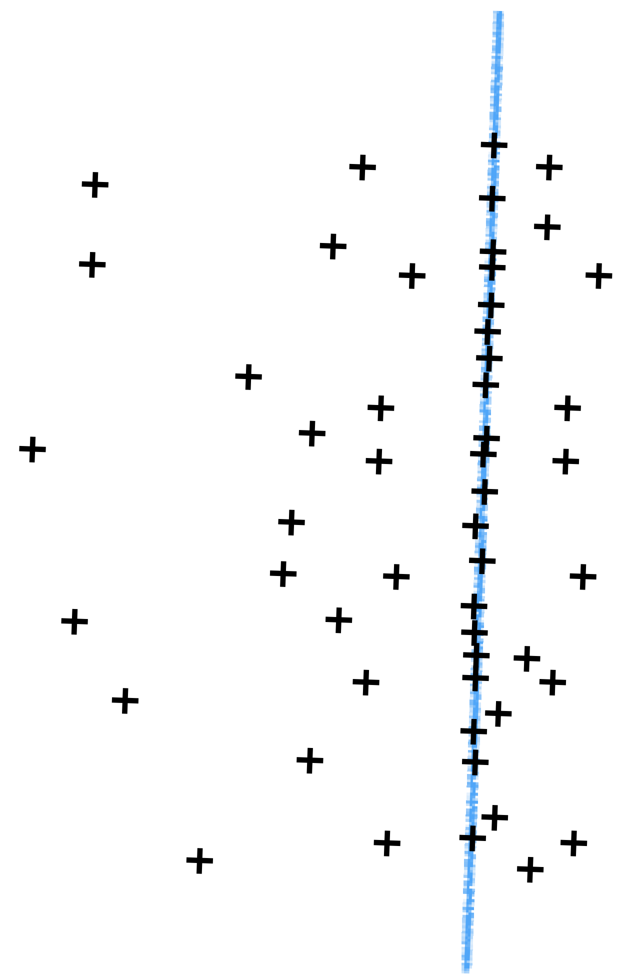


ℓ^1-solution

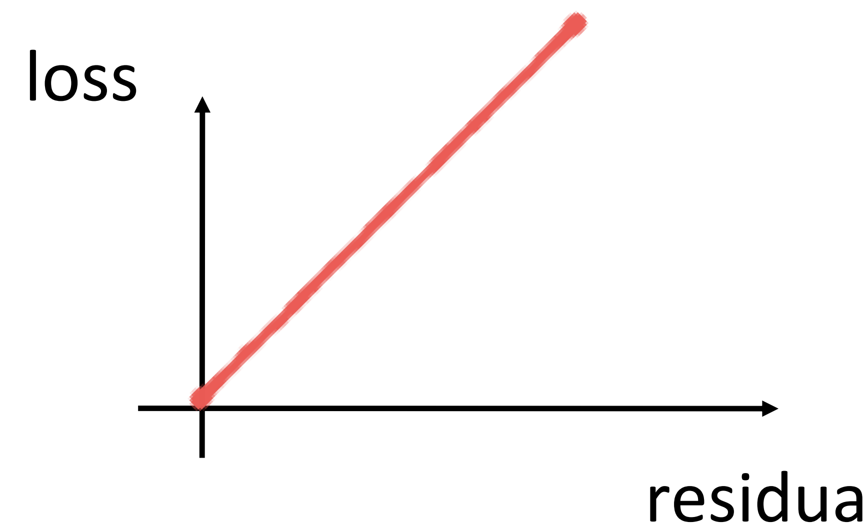


$$\min_{\theta} \sum |r_i(\theta)|$$

Least Absolute Residual

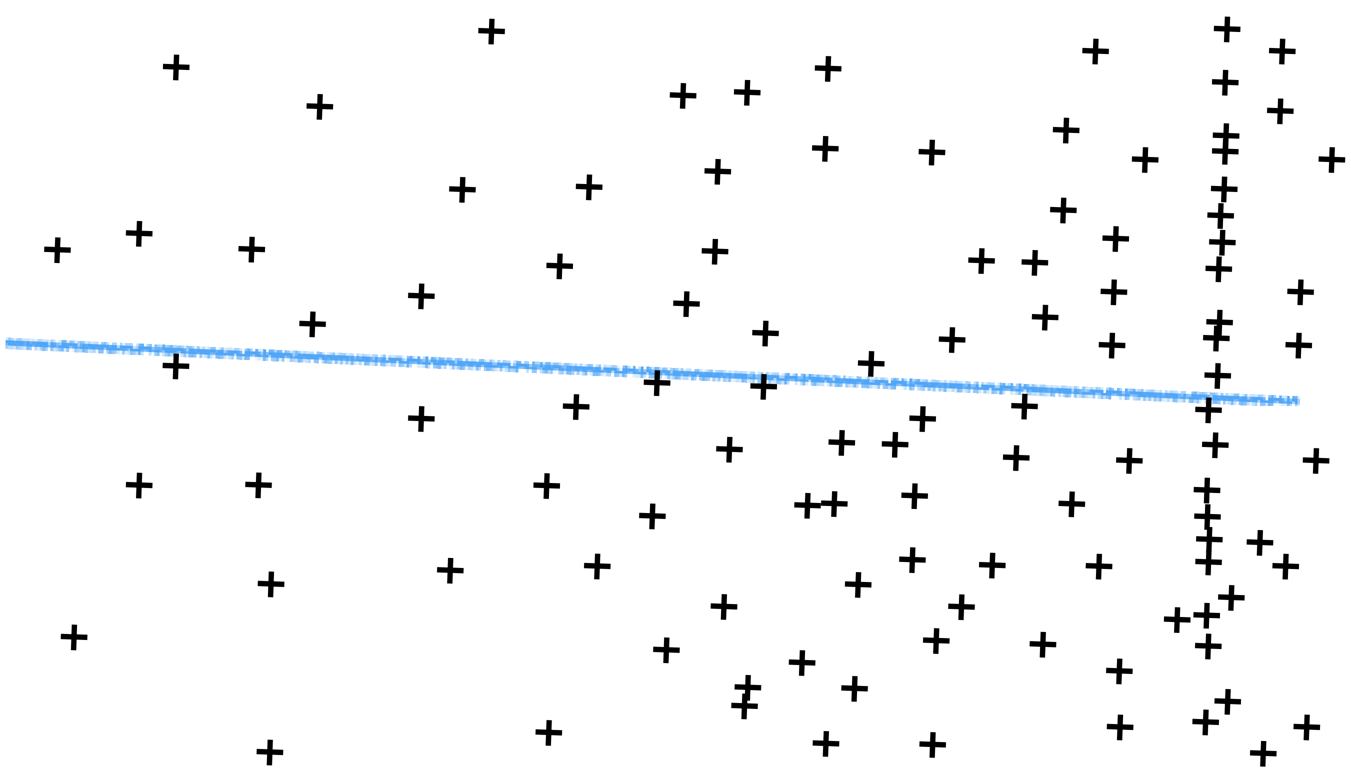


ℓ^1-solution

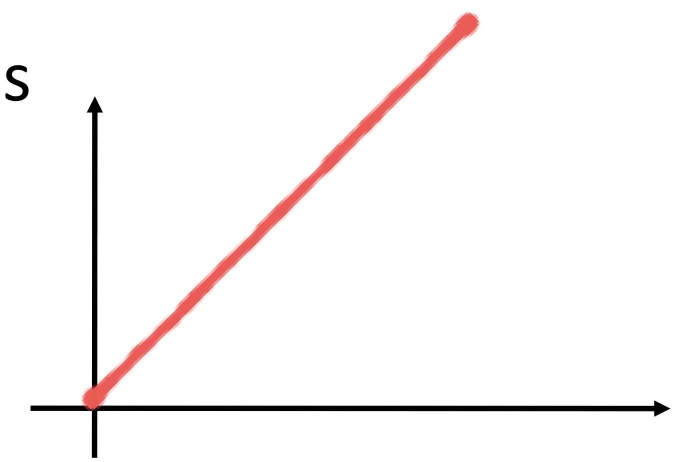


$$\min_{\theta} \sum |r_i(\theta)|$$

Least Absolute Residual

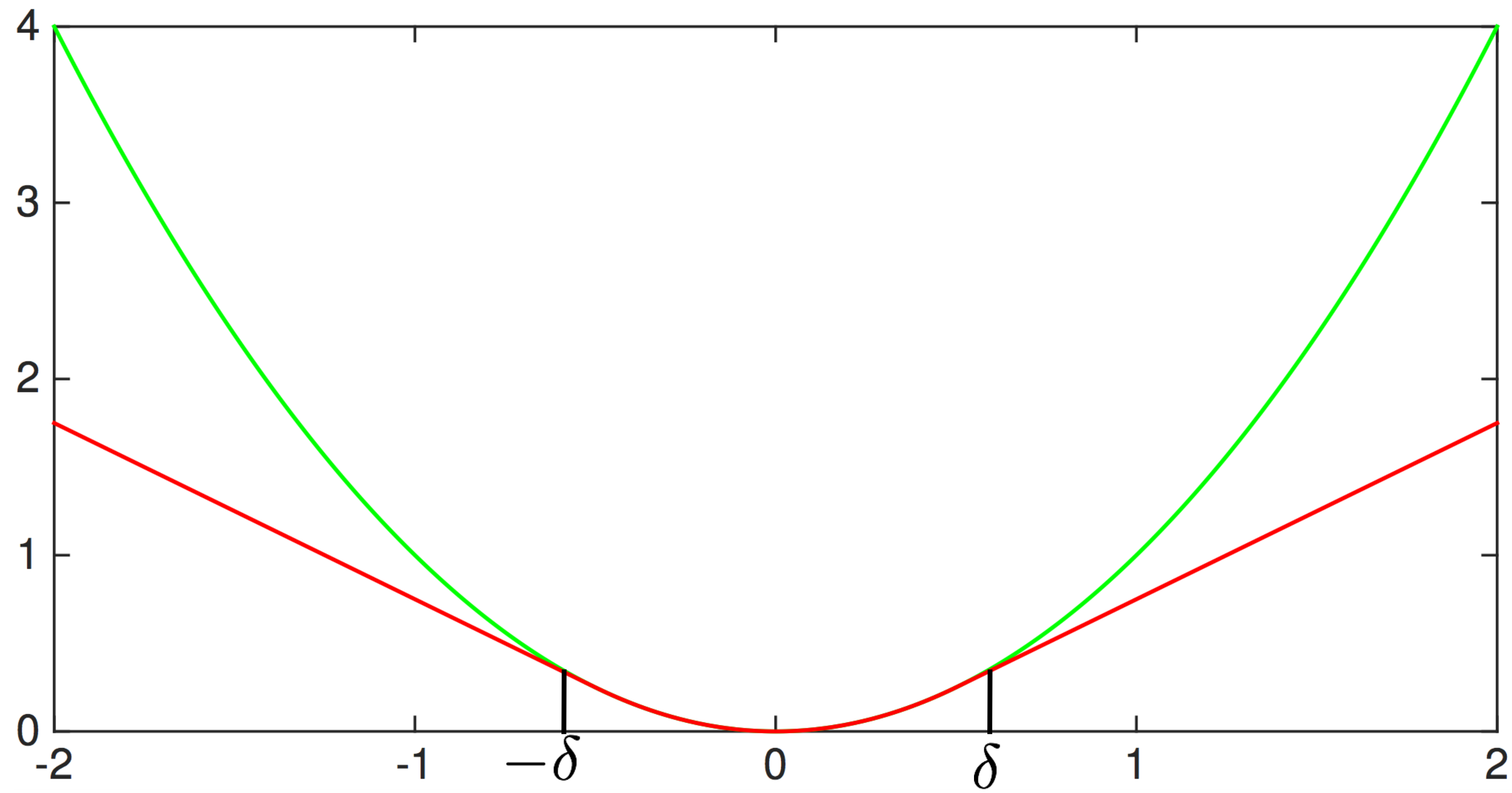


ℓ^1-solution



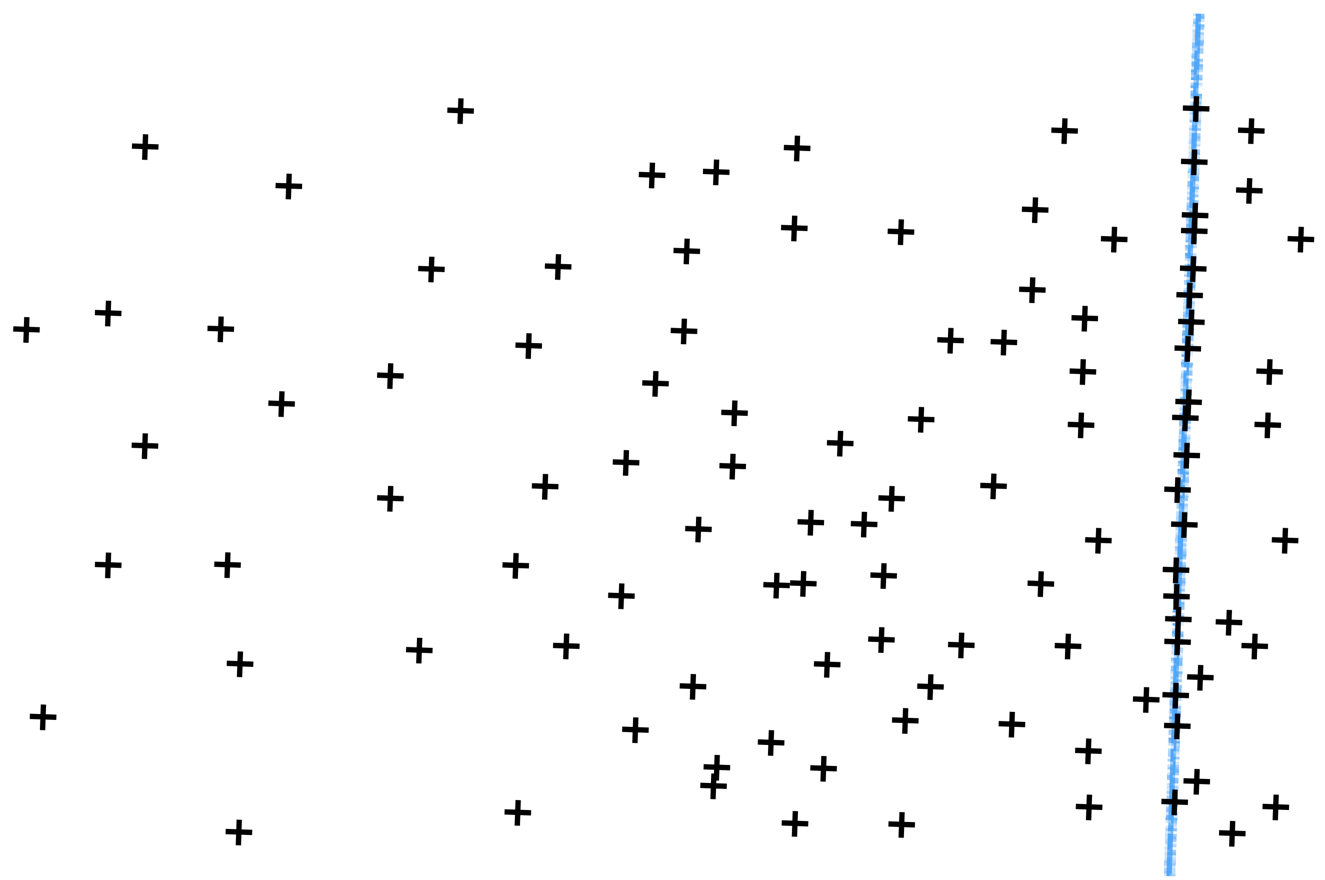
$$\min_{\theta} \sum |r_i(\theta)|$$

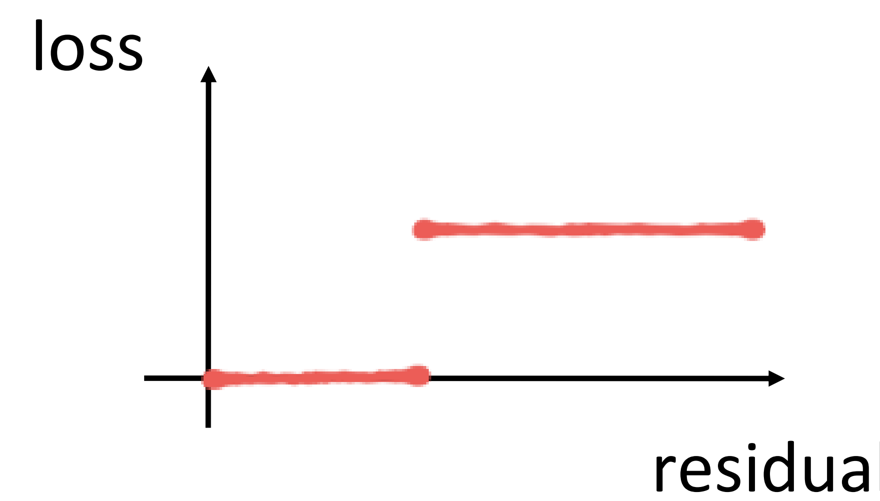
Huber Loss



$$\min_{\theta} \sum_i h(r_i(\theta)^2) \text{ with } h(x) = \begin{cases} x & \text{if } x < \delta \\ 2\delta\sqrt{x} - \delta^2 & \text{if } x \geq \delta \end{cases}$$

Minimizing Number of Outliers

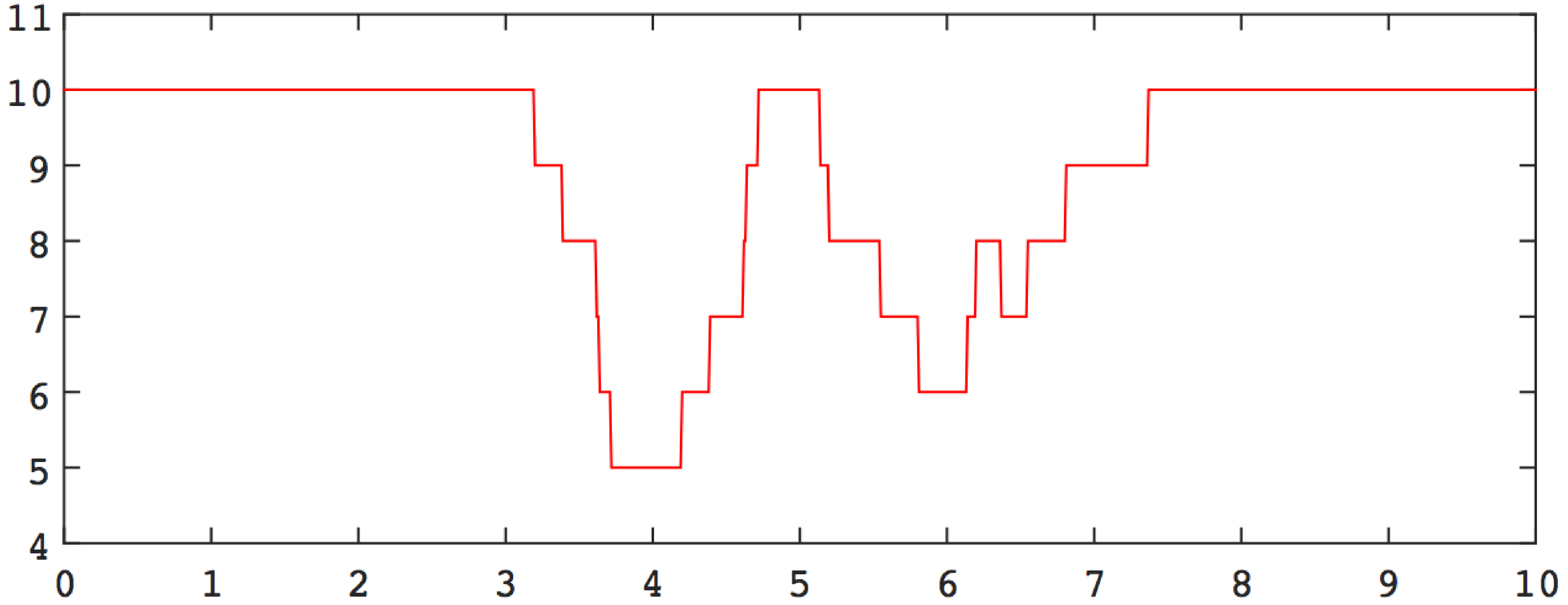




Outlier count

How to Minimize?

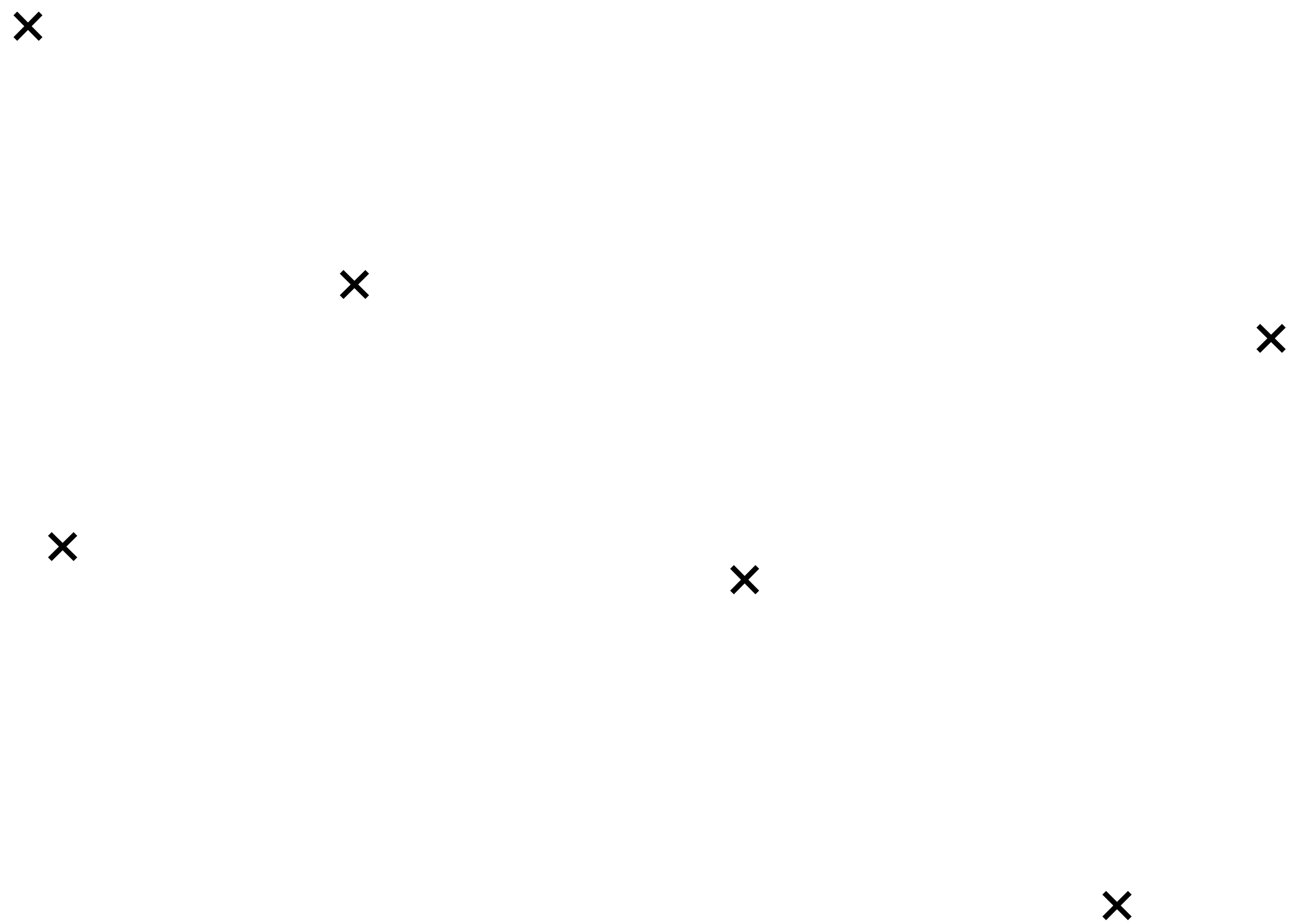
1D example:



RANdom SAmple Consensus (RANSAC)

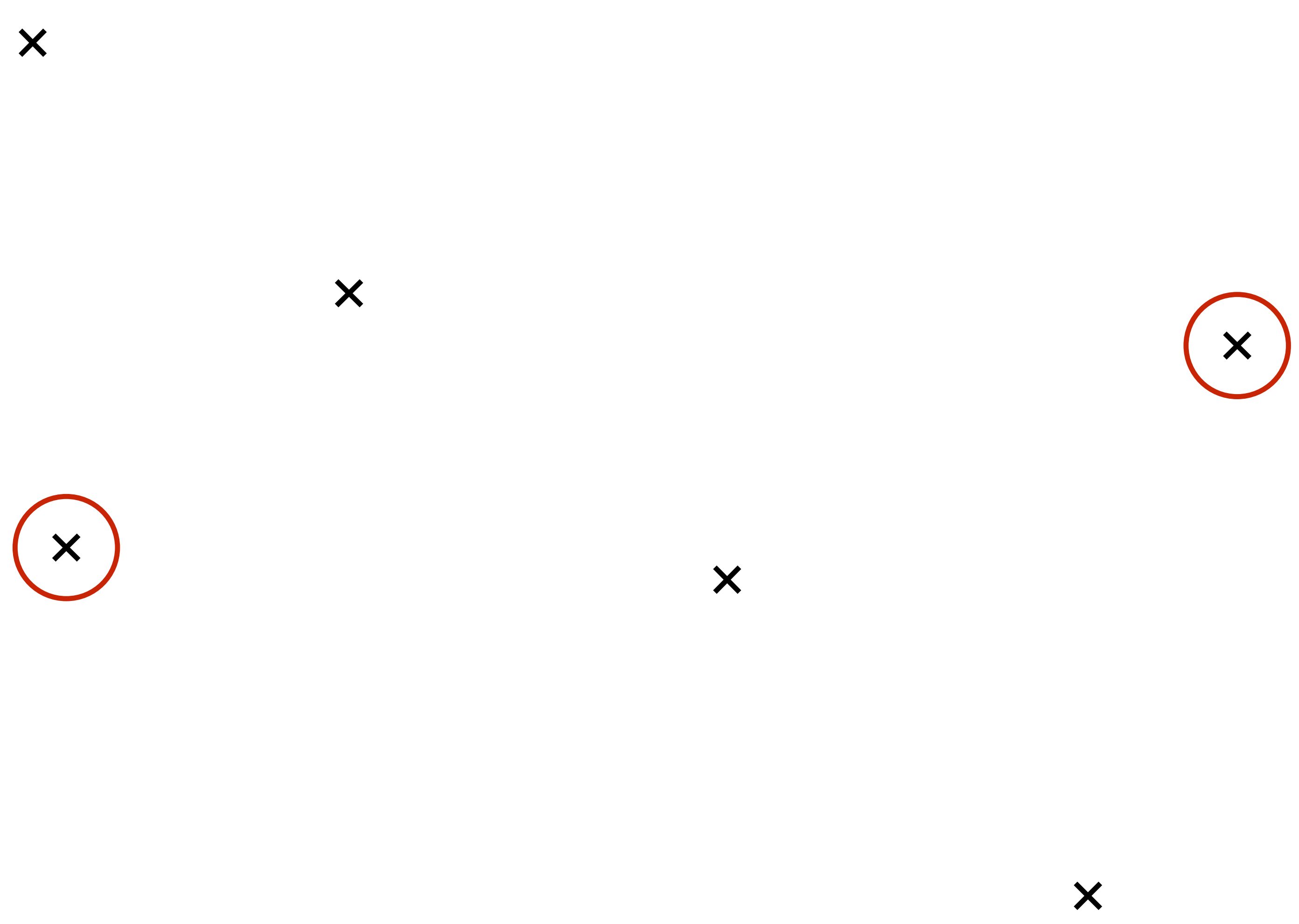
RANdom SAmple Consensus - RANSAC

Line fitting example



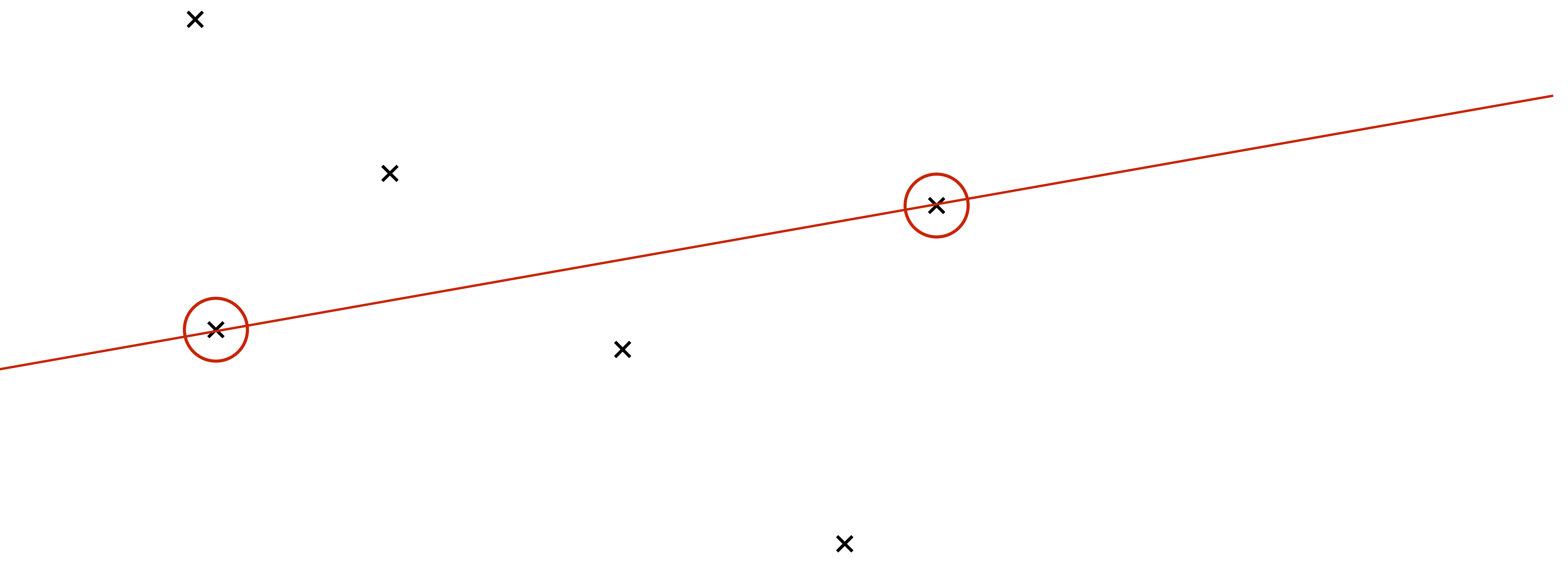
RANdom SAmple Consensus - RANSAC

Line fitting example



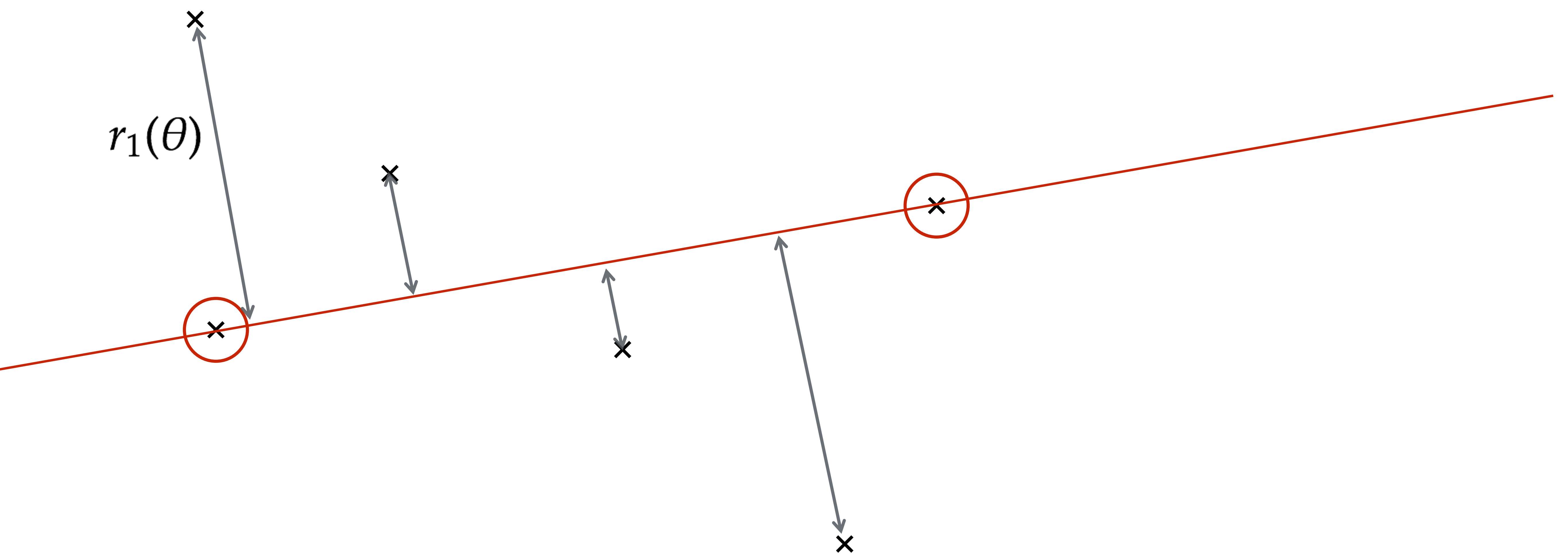
RANdom SAmple Consensus - RANSAC

Line fitting example



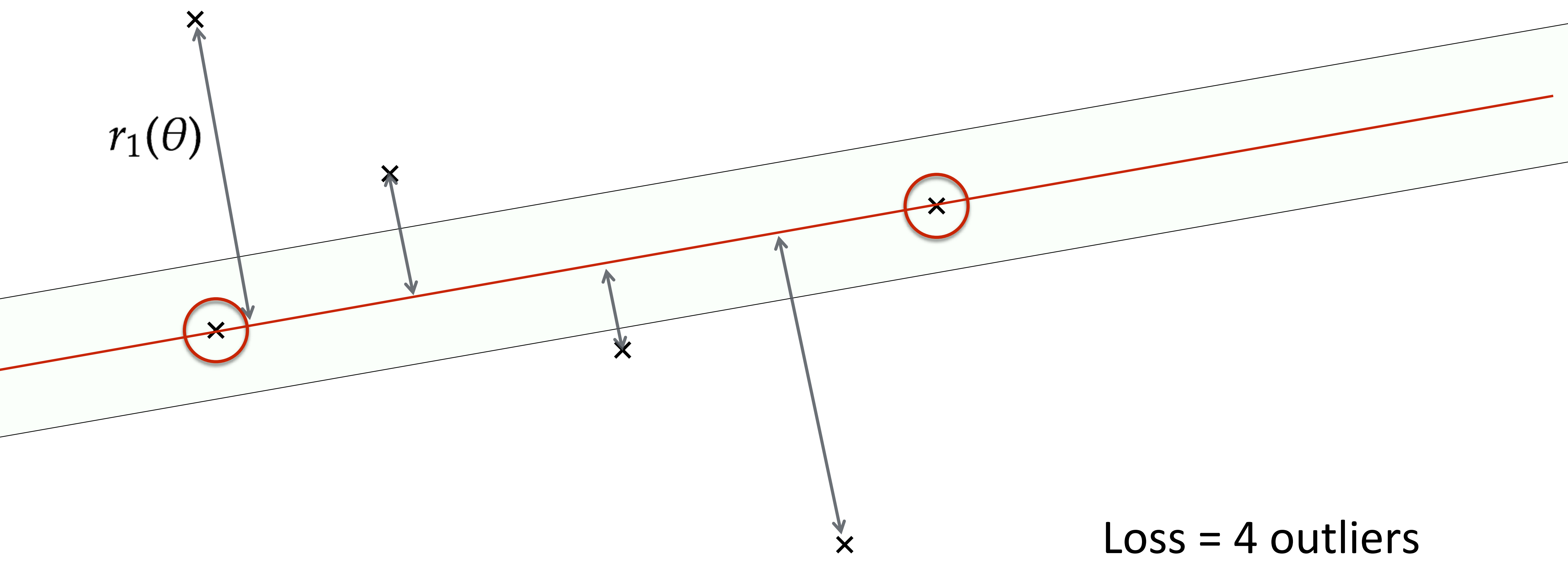
RANdom SAmple Consensus - RANSAC

Line fitting example



RANdom SAmple Consensus - RANSAC

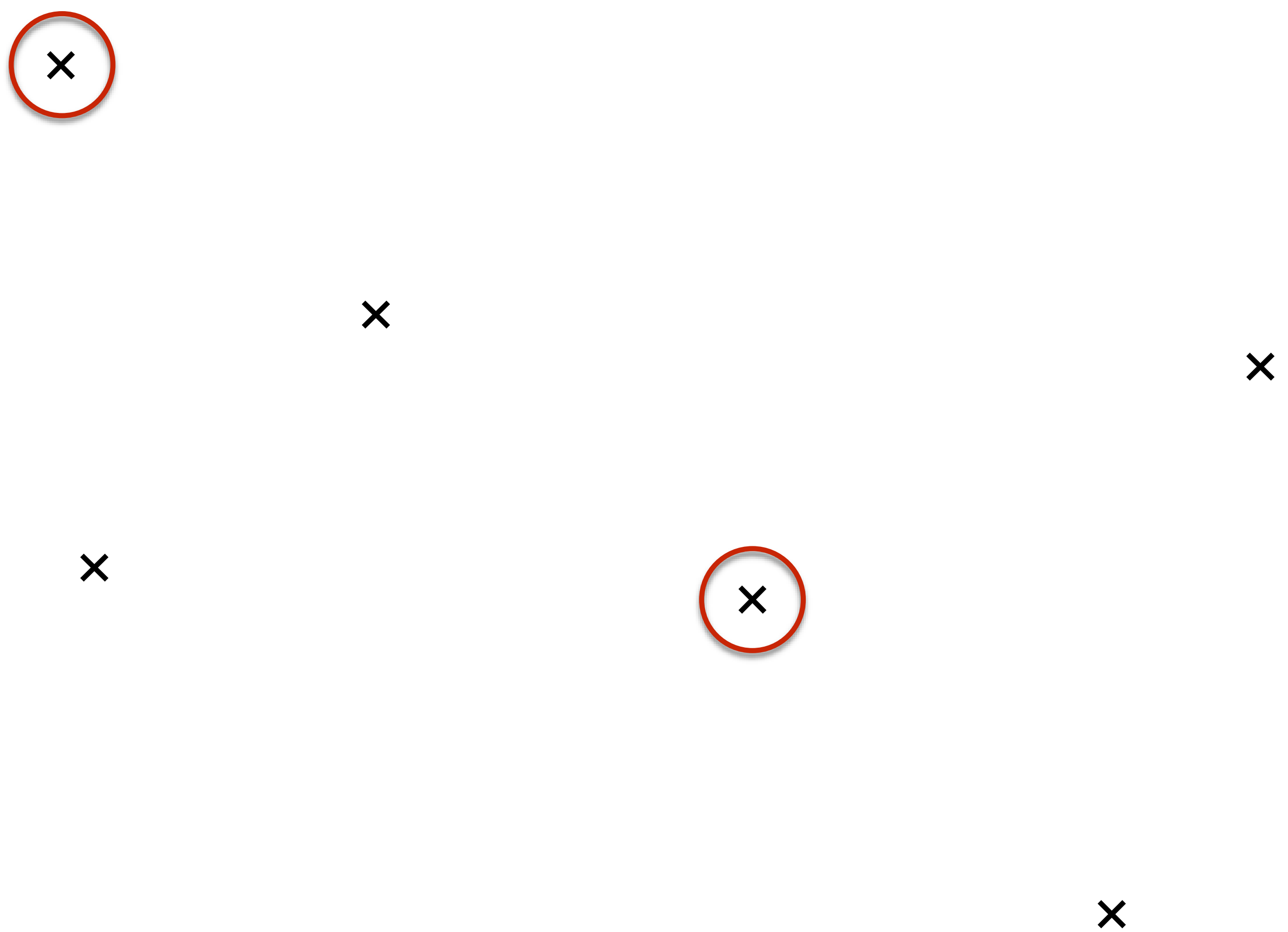
Line fitting example



RANdom SAmple Consensus - RANSAC

Line fitting example

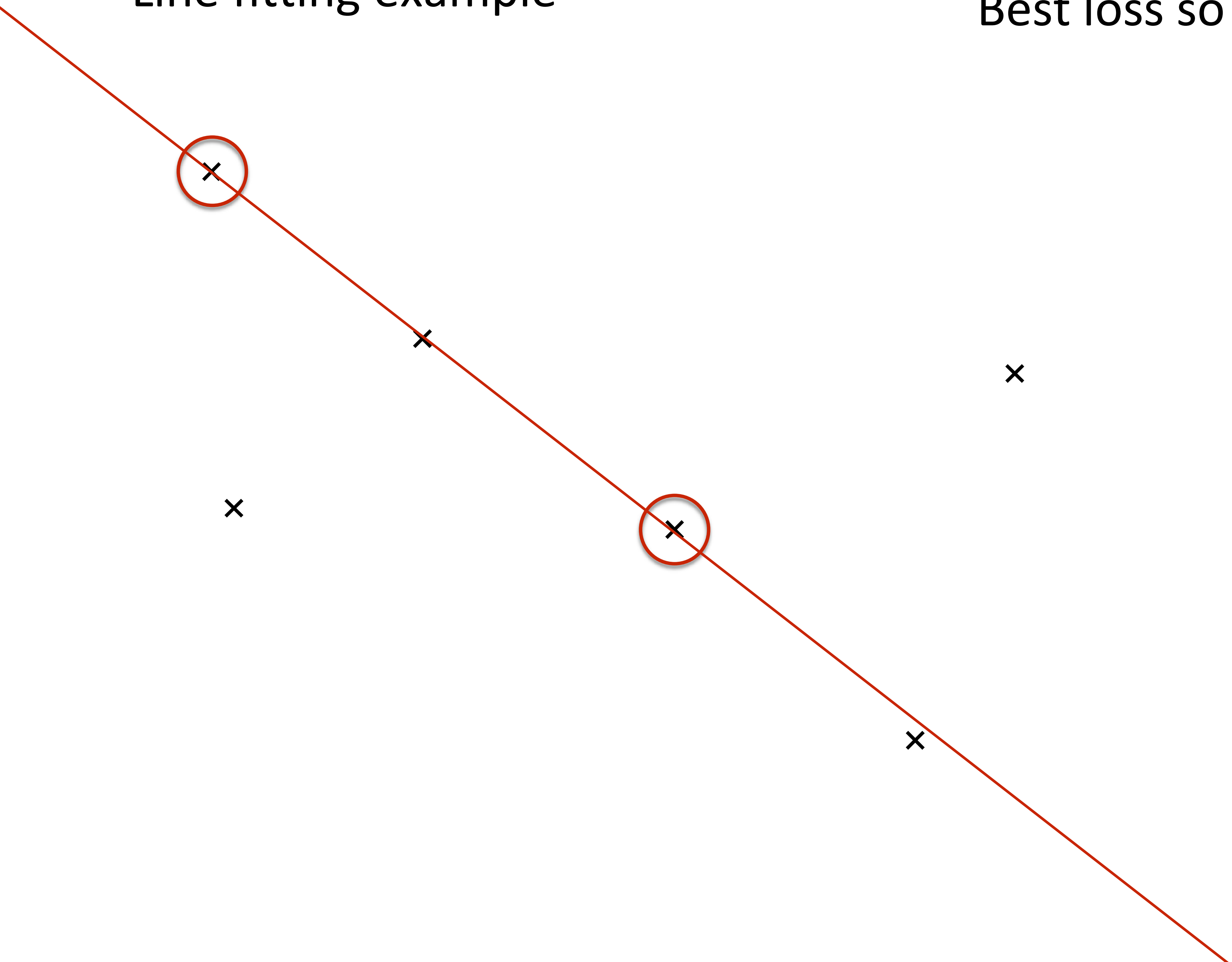
Best loss so far = 4 outliers



RANdom SAmple Consensus - RANSAC

Line fitting example

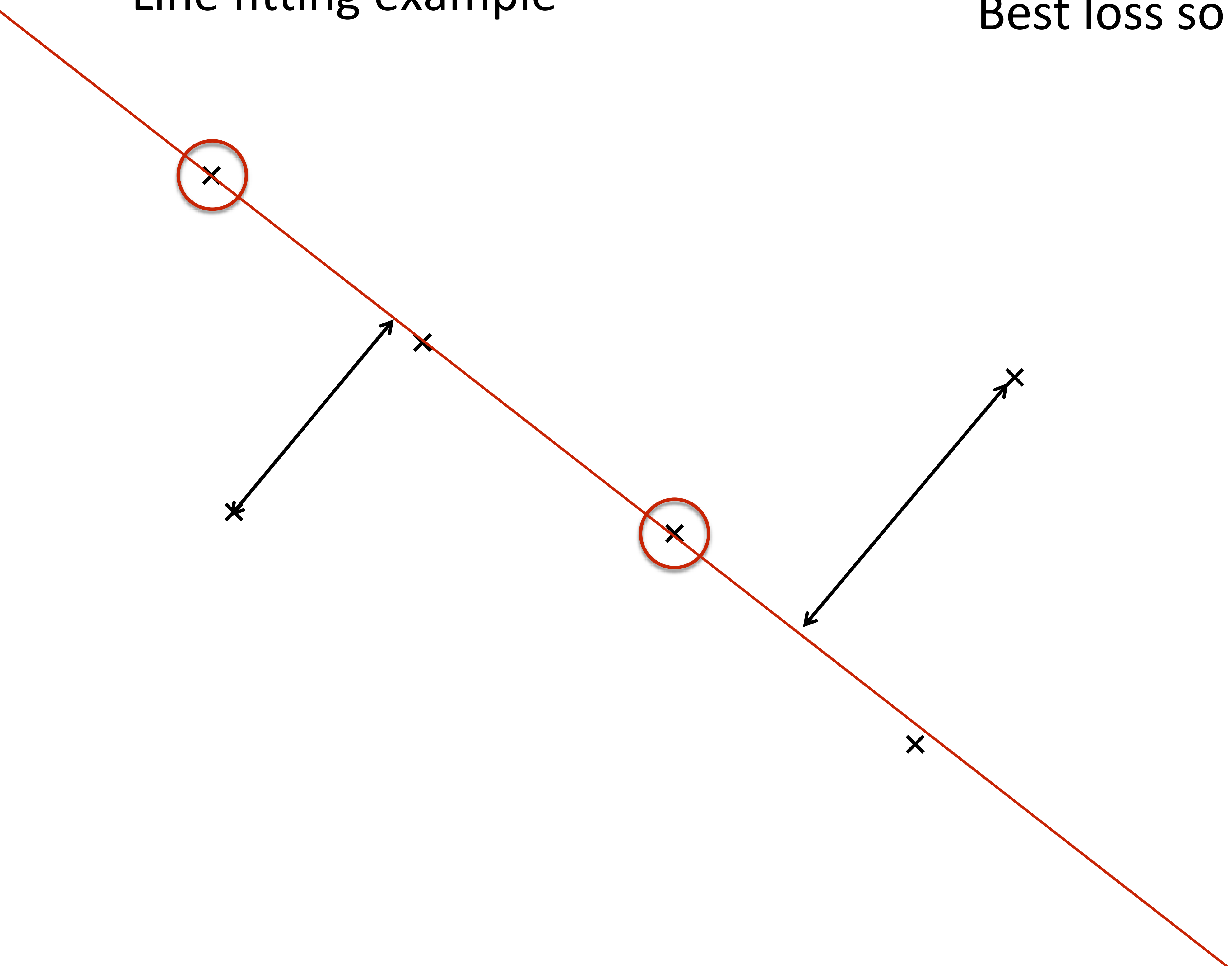
Best loss so far =   4 outliers



RANdom SAmple Consensus - RANSAC

Line fitting example

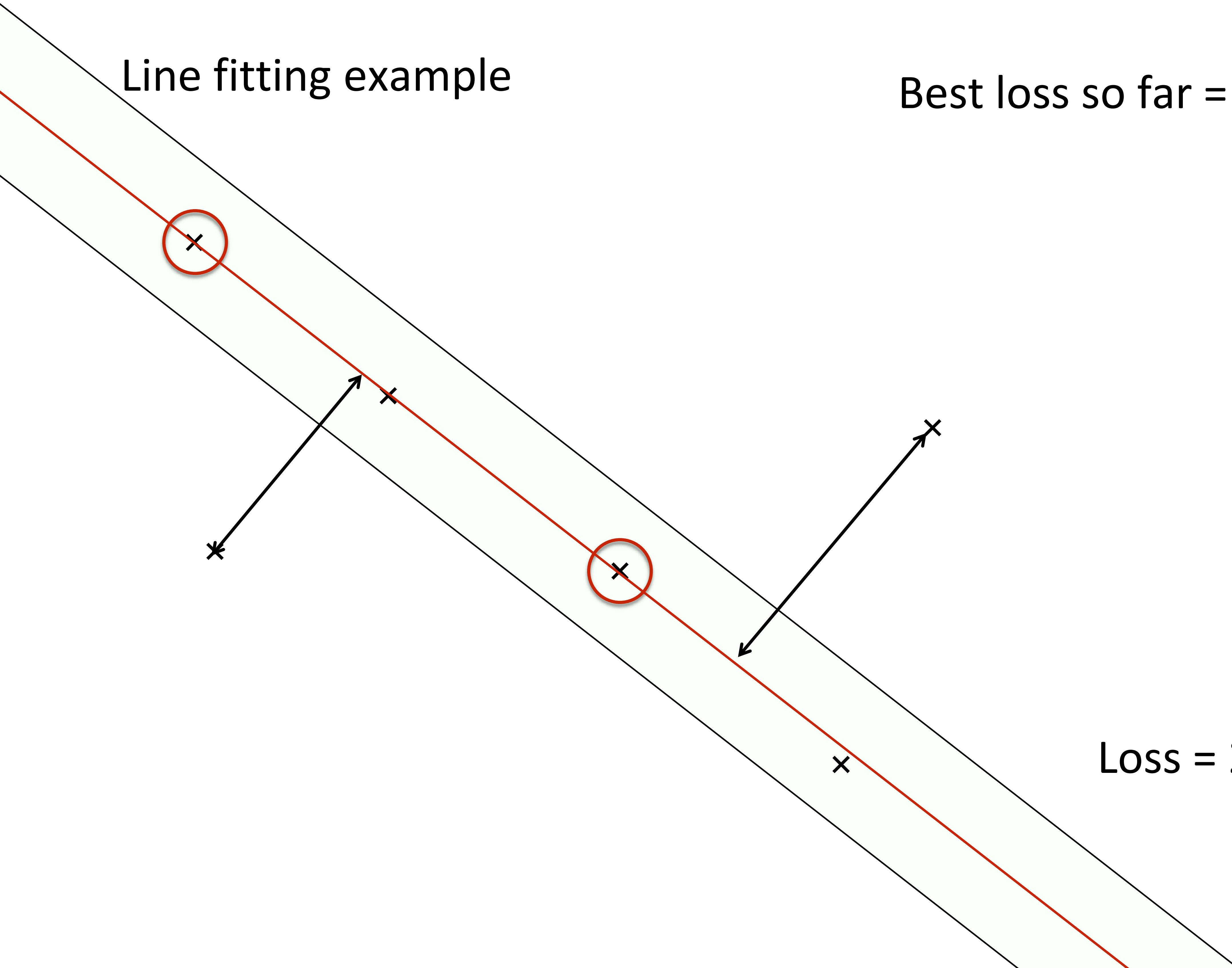
Best loss so far = 4 outliers



RANdom SAmple Consensus - RANSAC

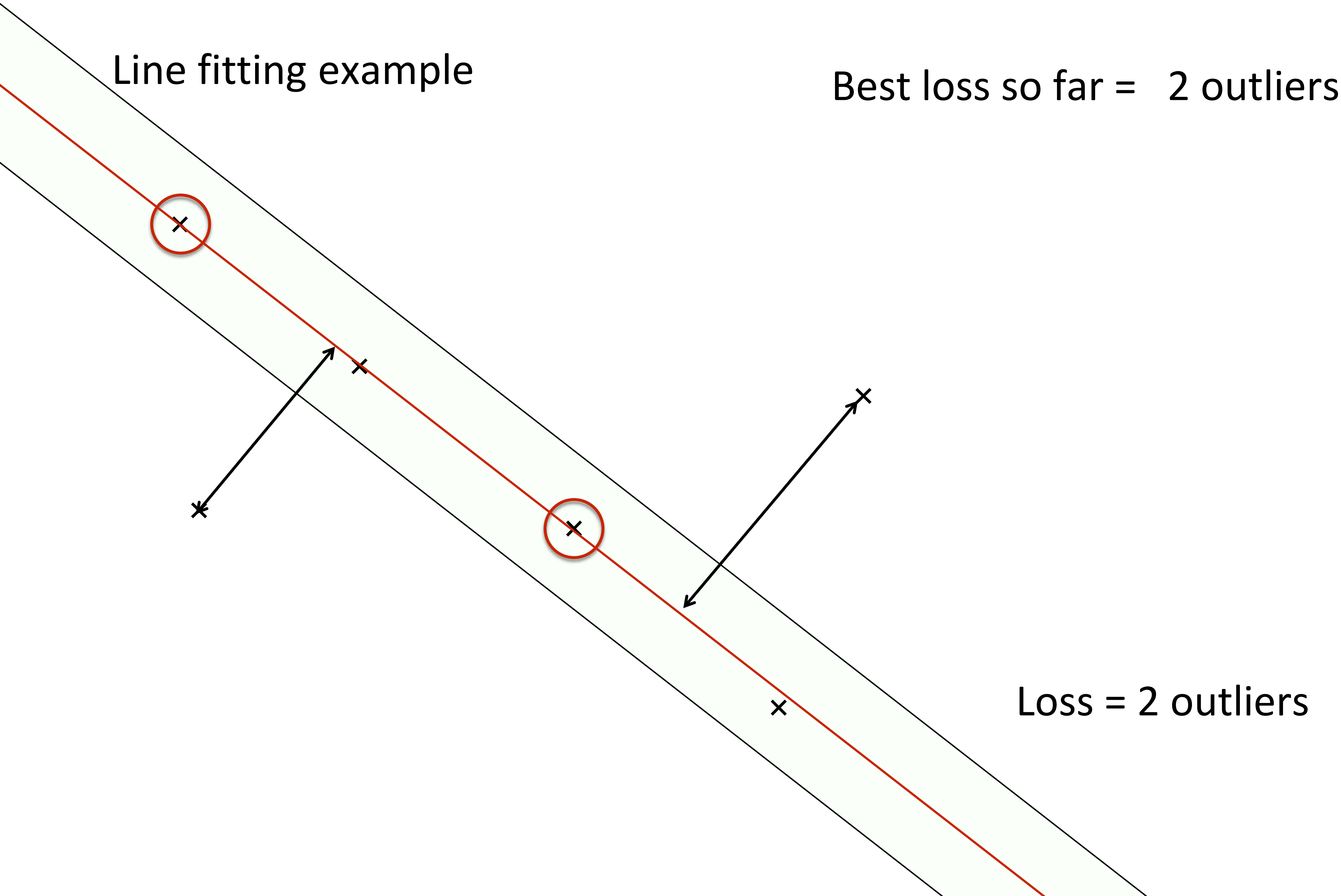
Line fitting example

Best loss so far = 4 outliers



Loss = 2 outliers

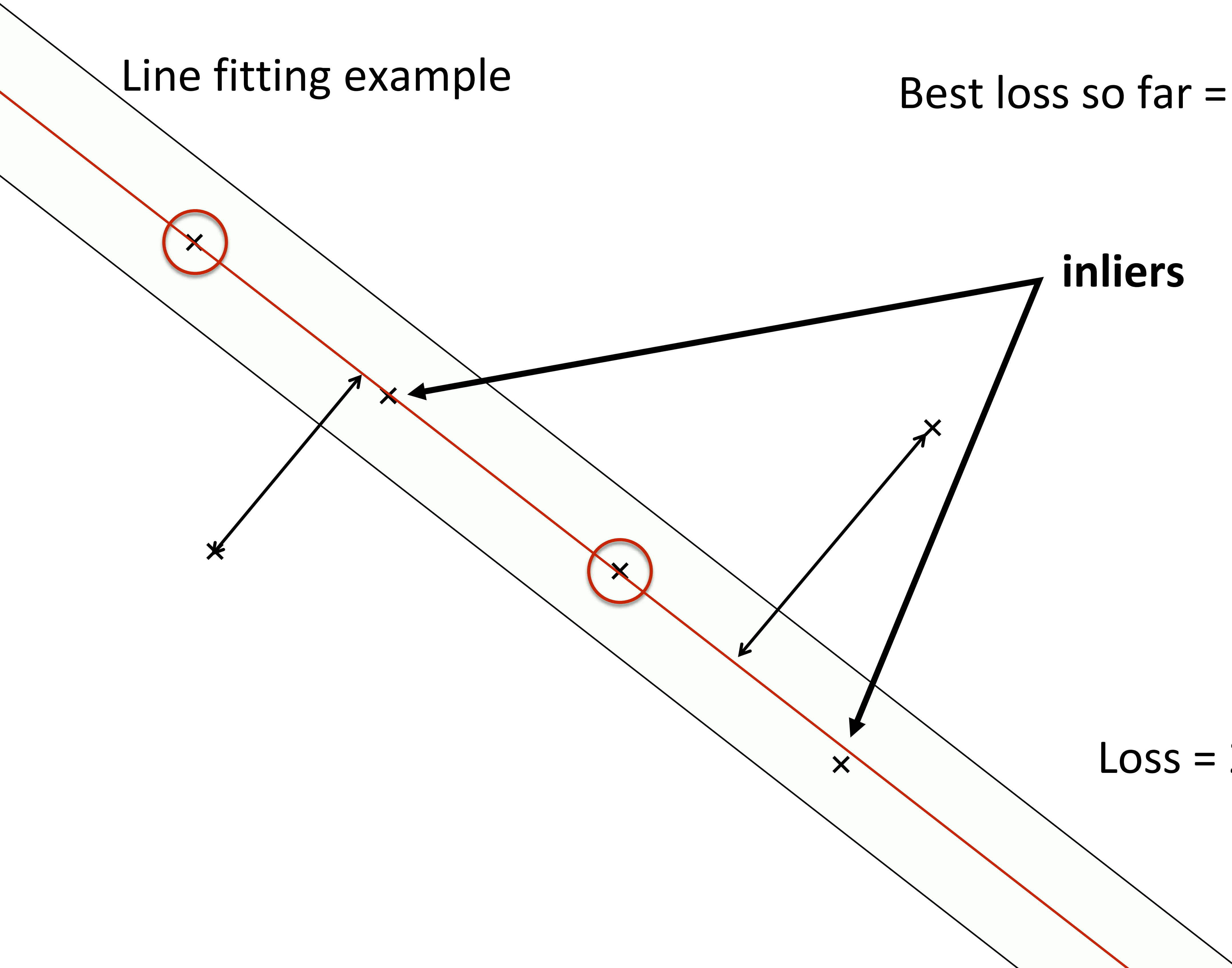
RANdom SAmple Consensus - RANSAC



RANdom SAmple Consensus - RANSAC

Line fitting example

Best loss so far = 2 outliers



Loss = 2 outliers

RANdom SAmple Consensus - RANSAC

While probability of missing correct model $>\eta$

Estimate model from *n* random data points

Estimate support (= #**inliers**) of model

If more inliers than previous best model

update best model

Return: Model with most inliers

[Fischler & Bolles, Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography. CACM 1981]

Termination Criterion

- Let's assume we know the **inlier ratio** ε (fraction of inliers)

[Fischler & Bolles, Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography. CACM 1981]

Termination Criterion

- Let's assume we know the **inlier ratio** ε (fraction of inliers)

- Probability of picking an inlier randomly: ε

[Fischler & Bolles, Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography. CACM 1981]

Termination Criterion

- Let's assume we know the **inlier ratio** ε (fraction of inliers)
- Probability of picking an inlier randomly: ε
- Probability of picking n inlier randomly: ε^n

[Fischler & Bolles, Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography. CACM 1981]

Termination Criterion

- Let's assume we know the **inlier ratio** ε (fraction of inliers)
- Probability of picking an inlier randomly: ε
- Probability of picking n inlier randomly: ε^n
- Probability of non-all inlier sample (≥ 1 outlier): $(1-\varepsilon^n)$

[Fischler & Bolles, Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography. CACM 1981]

Termination Criterion

- Let's assume we know the **inlier ratio** ε (fraction of inliers)
- Probability of picking an inlier randomly: ε
- Probability of picking n inlier randomly: ε^n
- Probability of non-all inlier sample (≥ 1 outlier): $(1-\varepsilon^n)$
- Probability of not picking all-inlier sample in k iterations: $(1-\varepsilon^n)^k$

[Fischler & Bolles, Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography. CACM 1981]

Termination Criterion

- Terminate if $(1-\varepsilon^n)^k < \eta$

[Fischler & Bolles, Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography. CACM 1981]

Termination Criterion

- Terminate if $(1-\varepsilon^n)^k < \eta$
- In practice: Compute maximum number iterations k_{max}

[Fischler & Bolles, Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography. CACM 1981]

Termination Criterion

- Terminate if $(1-\varepsilon^n)^k < \eta$
- In practice: Compute maximum number iterations k_{max}
  - Find k_{max} such that $(1-\varepsilon^n)^{k_{max}} = \eta$

[Fischler & Bolles, Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography. CACM 1981]

Termination Criterion

- Terminate if $(1-\varepsilon^n)^k < \eta$
- In practice: Compute maximum number iterations k_{max}
  - Find k_{max} such that $(1-\varepsilon^n)^{k_{max}} = \eta$

  $\Leftrightarrow \; k_{max} \ln(1-\varepsilon^n) = \ln(\eta)$

[Fischler & Bolles, Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography. CACM 1981]

Termination Criterion

- Terminate if $(1-\varepsilon^n)^k < \eta$

- In practice: Compute maximum number iterations k_{max}
  - Find k_{max} such that $(1-\varepsilon^n)^{k_{max}} = \eta$

$\Leftrightarrow \; k_{max} \ln(1-\varepsilon^n) = \ln(\eta)$

$\Leftrightarrow \; k_{max} = \ln(\eta) \;/\; \ln(1-\varepsilon^n)$

[Fischler & Bolles, Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography. CACM 1981]

Termination Criterion

- Terminate if $(1-\varepsilon^n)^k < \eta$
- In practice: Compute maximum number iterations k_{max}
  - Find k_{max} such that $(1-\varepsilon^n)^{k_{max}} = \eta$

  $\Leftrightarrow \; k_{max} \ln(1-\varepsilon^n) = \ln(\eta)$

  $\Leftrightarrow \; k_{max} = \ln(\eta) \;/\; \ln(1-\varepsilon^n)$

  - Note: $k_{max}(\varepsilon) > k_{max}(\varepsilon')$ if $\varepsilon < \varepsilon'$

[Fischler & Bolles, Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography. CACM 1981]

Termination Criterion

- Terminate if $(1-\varepsilon^n)^k < \eta$
- In practice: Compute maximum number iterations k_{max}
  - Find k_{max} such that $(1-\varepsilon^n)^{k_{max}} = \eta$

  $\Leftrightarrow \; k_{max} \ln(1-\varepsilon^n) = \ln(\eta)$

  $\Leftrightarrow \; k_{max} = \ln(\eta) \; / \; \ln(1-\varepsilon^n)$

  - Note: $k_{max}(\varepsilon) > k_{max}(\varepsilon')$ if $\varepsilon < \varepsilon'$
- How do we know inlier ratio ε?

[Fischler & Bolles, Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography. CACM 1981]

RANdom SAmple Consensus - RANSAC

Input: *m* data points

$k = 0$, $\boldsymbol{\varepsilon} = \boldsymbol{\varepsilon_0}$

$k_{max} = \log(\eta)/\log(1-\varepsilon^n)$

While $k < k_{max}$

Estimate model from *n* random data points

Estimate support (= #**inliers**) of model

If more inliers than previous best model

update best model

update $\boldsymbol{\varepsilon}$ = #inliers / m

update $k_{max} = \log(\eta)/\log(1-\varepsilon^n)$

Return: Model with most inliers

[Fischler & Bolles, Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography. CACM 1981]

Minimal Solvers

- Probability of picking $\mathtt{n}$ inlier randomly: $\varepsilon^{\mathtt{n}}$

[Fischler & Bolles, Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography. CACM 1981]

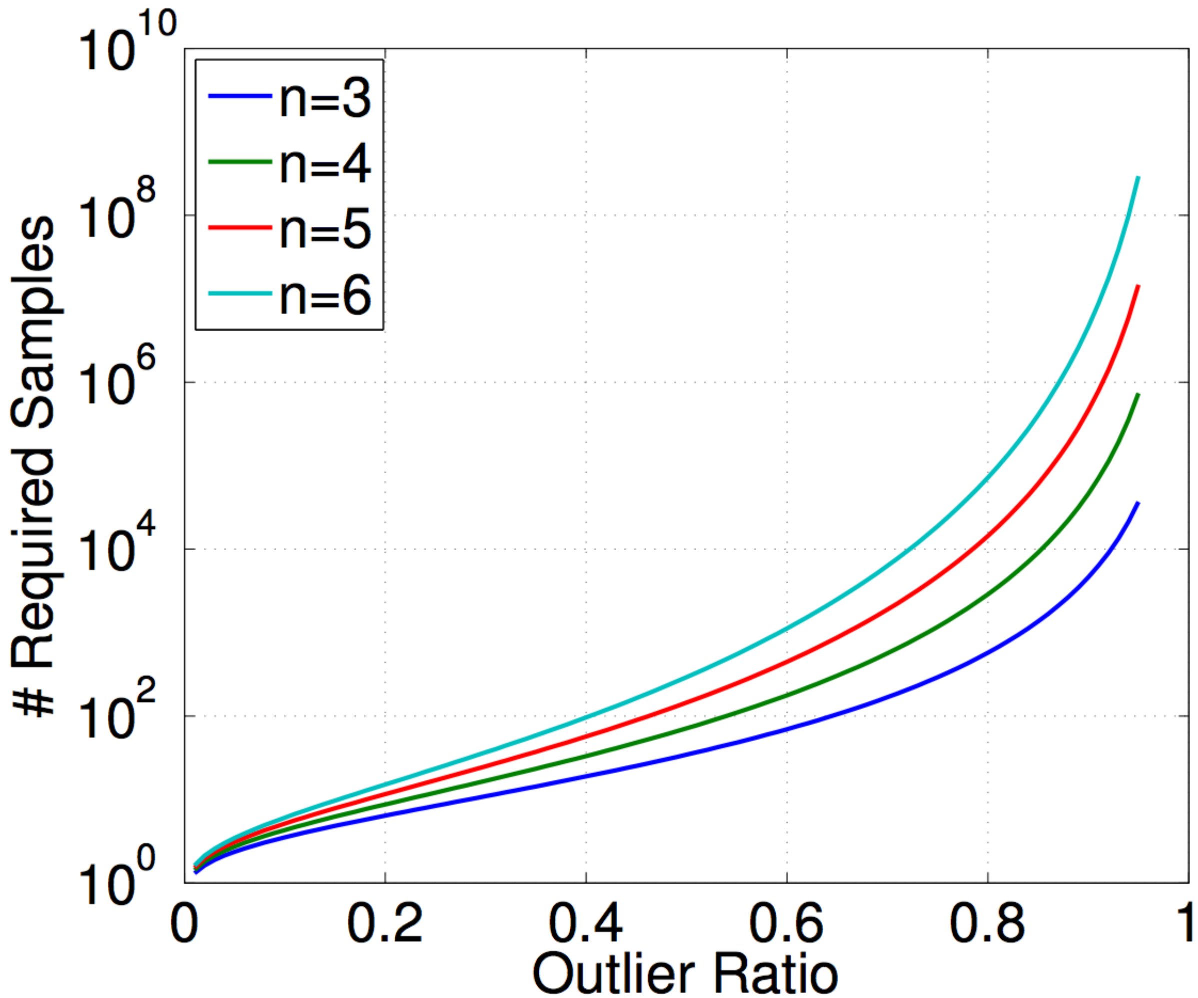
Minimal Solvers

- Probability of picking n inlier randomly: ε^n
- Maximize ε^n by minimizing n → **minimal solver**

[Fischler & Bolles, Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography. CACM 1981]

Minimal Solvers

- Probability of picking n inlier randomly: ε^{n}
- Maximize ε^{n} by minimizing n → **minimal solver**

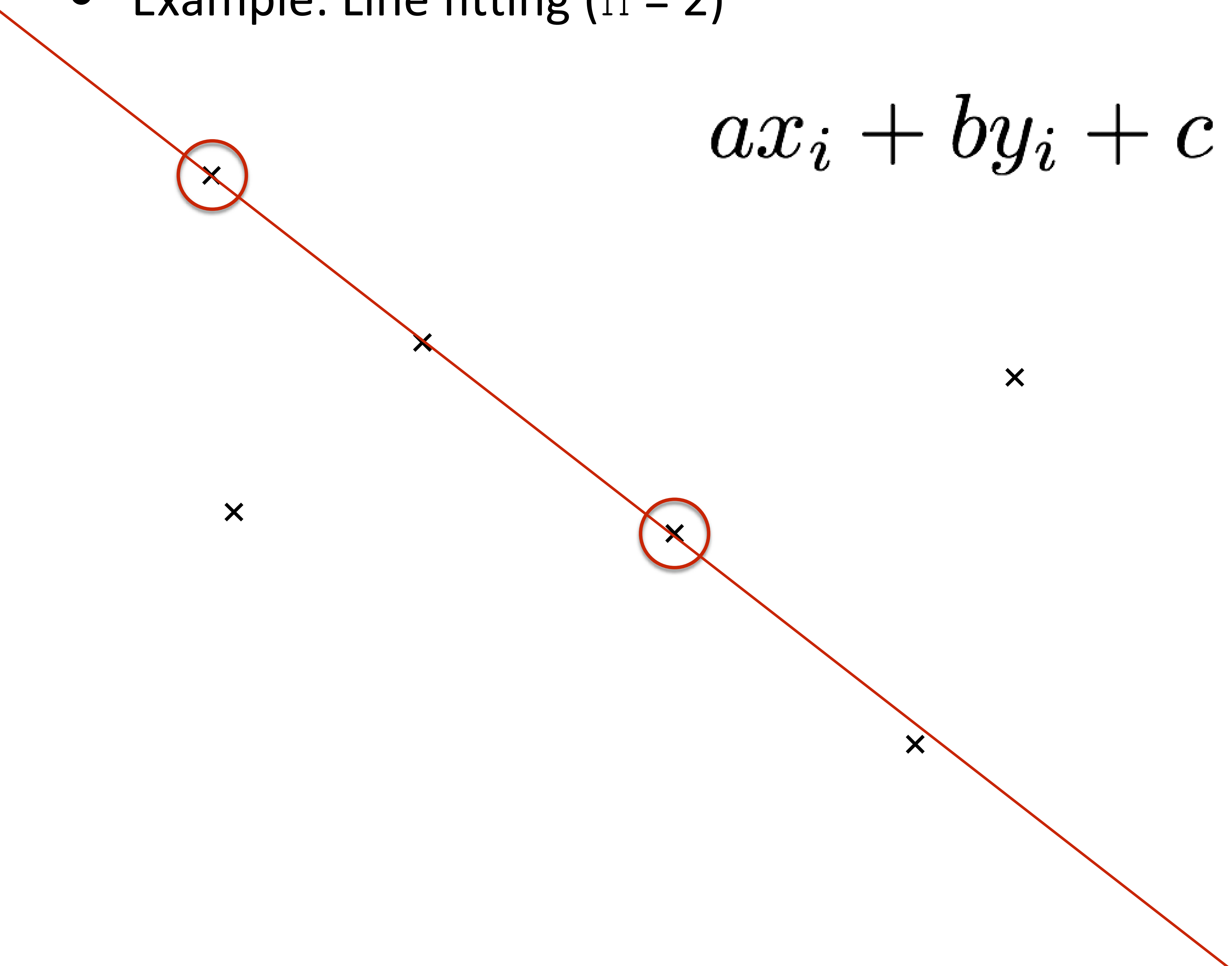


[Fischler & Bolles, Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography. CACM 1981]

Minimal Solvers

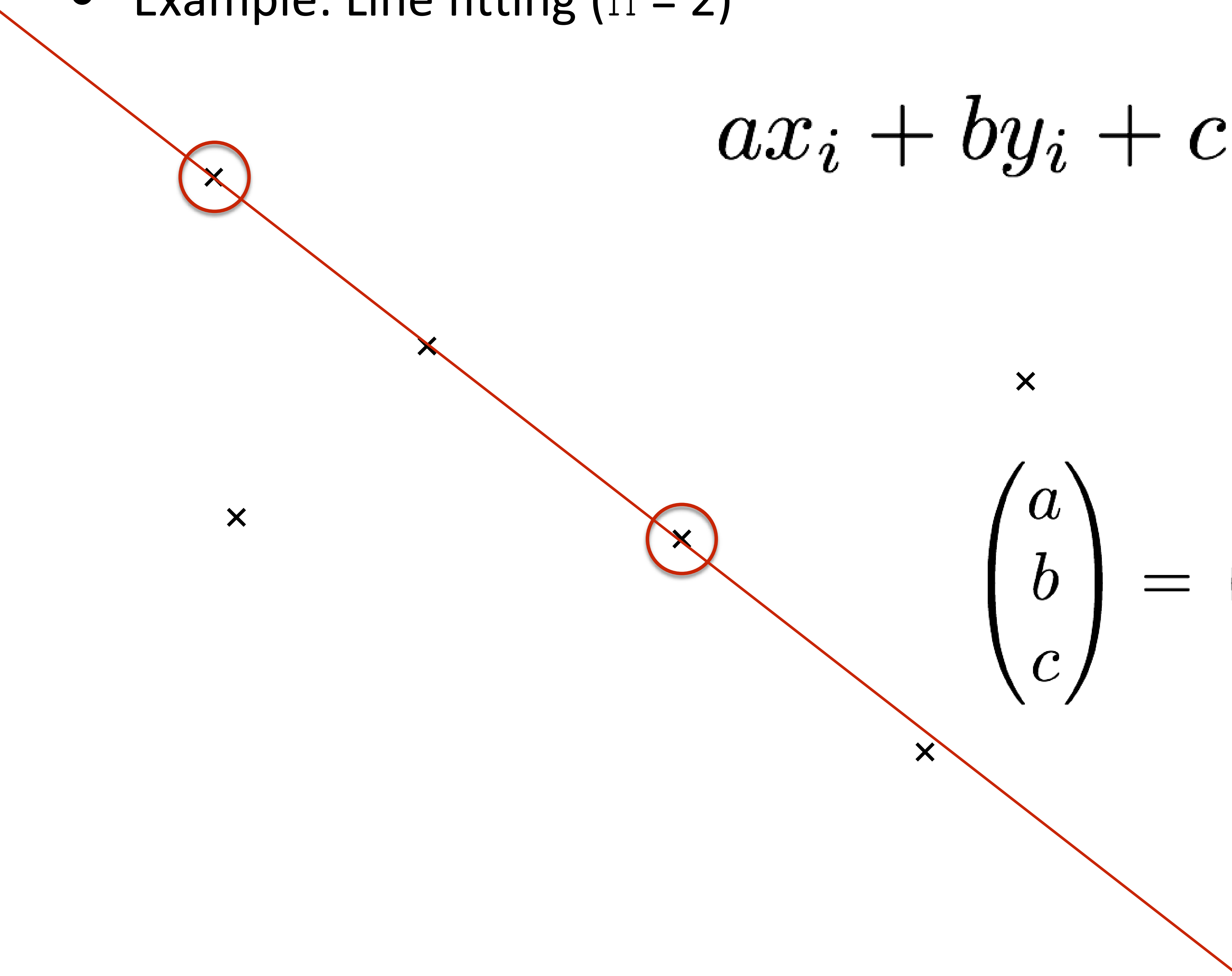
- Example: Line fitting (`n` = 2)

$$ax_i + by_i + c = 0$$



Minimal Solvers

- Example: Line fitting (n = 2)



$$a x_i + b y_i + c = 0$$

$$\begin{pmatrix} a \\ b \\ c \end{pmatrix} = \begin{pmatrix} x \\ y \\ 1 \end{pmatrix} \times \begin{pmatrix} \hat{x} \\ \hat{y} \\ 1 \end{pmatrix}$$

Minimal Solvers

- Lab 3: Create minimal solver for affine transformation

$$\begin{pmatrix} \hat{x} \\ \hat{y} \end{pmatrix} = \begin{pmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} + \begin{pmatrix} t_x \\ t_y \end{pmatrix}$$

Minimal Solvers

- Lab 3: Create minimal solver for affine transformation

$$\begin{pmatrix} \hat{x} \\ \hat{y} \end{pmatrix} = \begin{pmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} + \begin{pmatrix} t_x \\ t_y \end{pmatrix}$$

- **Hint**: Write affine transformation as linear system in the parameters of the affine transformation, solve the system

$$\mathbf{A} \begin{pmatrix} a_{11} \\ a_{12} \\ a_{21} \\ a_{22} \\ t_x \\ t_y \end{pmatrix} = \mathbf{b}$$

RANSAC in Practice

While probability of missing correct model $>\eta$

Estimate model from *n* random data points

Estimate support (= #**inliers**) of model

If more inliers than previous best model

local optimization

update best model, η

Refine best model via least squares on inliers

Return: refined best model

[Chum, Matas, Optimal Randomized RANSAC. PAMI 2008]

[Fischler & Bolles, Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography. CACM 1981]

RANSAC in Practice

While probability of missing correct model $> \eta$

Estimate model from *n* random data points

Estimate support (= #**inliers**) of model

[Chum, Matas, Optimal Randomized RANSAC. PAMI 2008]

If more inliers than previous best model

local optimization [Lebeda, Matas, Chum, Fixing the Locally Optimized RANSAC. BMVC 2012]

update best model, η

Refine best model via least squares on inliers

Return: refined best model

[Fischler & Bolles, Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography. CACM 1981]

Lessons Learned

- Main lessons from this lecture
  - What is model fitting?
  - Impact of outliers on least-squares estimates
  - Robust cost functions and their relation to probability distributions
  - RANSAC: Why? How?
  - What are minimal solvers?
- Next lecture: Image Registration

Next Lecture

Jan. 20	Introduction, Linear classifiers and filtering	
Jan. 23	Filtering, gradients, scale	Lab 1
Jan. 27	Local features	
Jan. 30	Learning a classifier	Lab 2
Feb. 3	Convolutional neural networks	
Feb. 6	More convolutional neural networks	
Feb. 10	Robust model fitting and RANSAC	Lab 3
Feb. 13	**Image registration**	
Feb. 17	Camera Geometry	Lab 4
Feb. 20	More camera geometry	
Feb. 24	Generative neural networks	
Feb. 27	Generative neural networks	
Mar. 2	TBA	
Mar. 9	TBA	