From Stereo to Graphical Models

23.04.2014

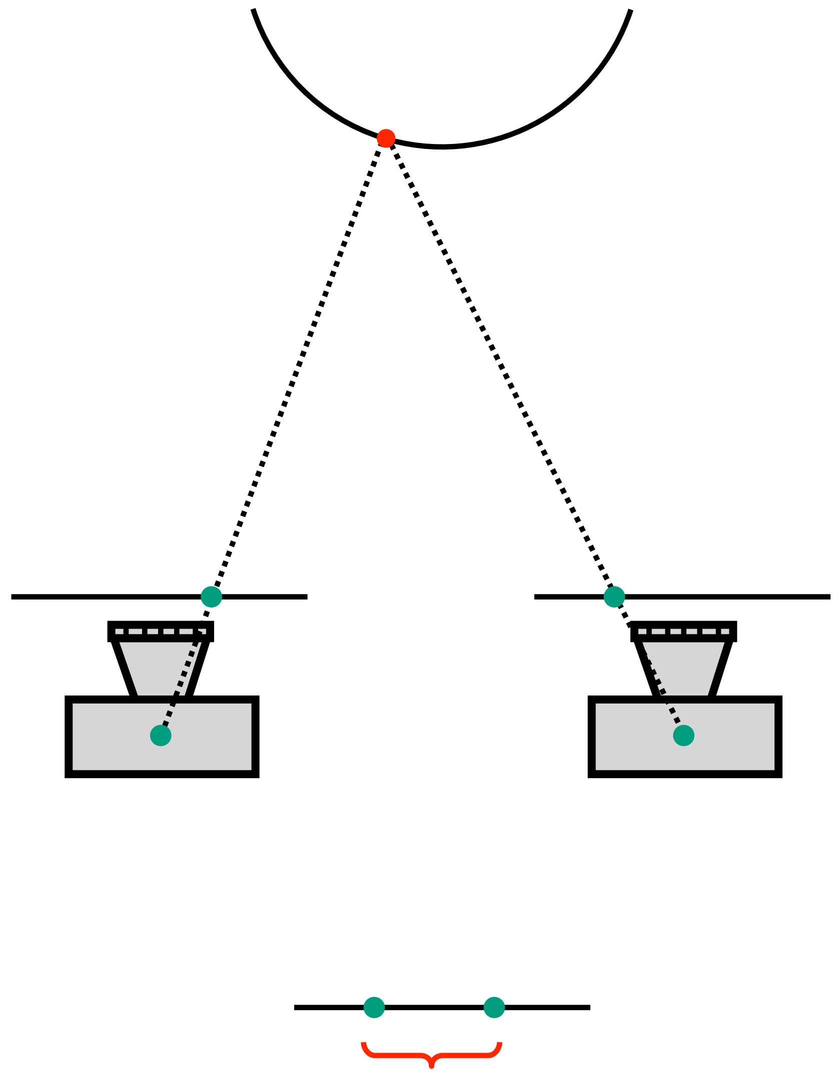
TECHNISCHE UNIVERSITÄT DARMSTADT

visual inference

Running Example: Binocular Stereo

From known geometry of the cameras and estimated disparity, recover depth in the scene

Left



binocular disparity

Right

Triangulation

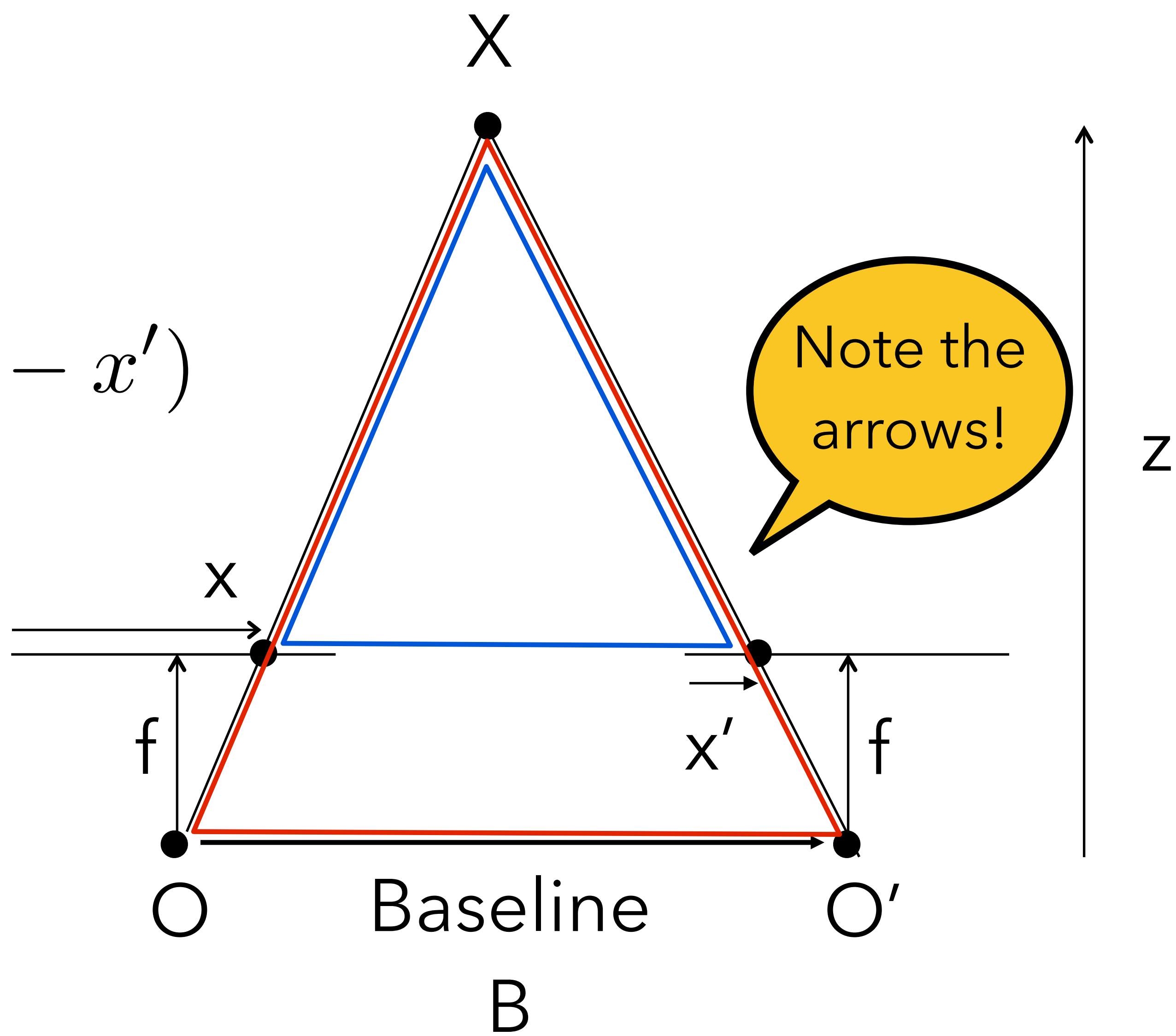
$$\frac{Z-f}{B-(x-x')} = \frac{Z}{B}$$

$$B \cdot Z - B \cdot f = B \cdot Z - Z \cdot (x - x')$$

$$Z = \frac{B \cdot f}{x - x'}$$

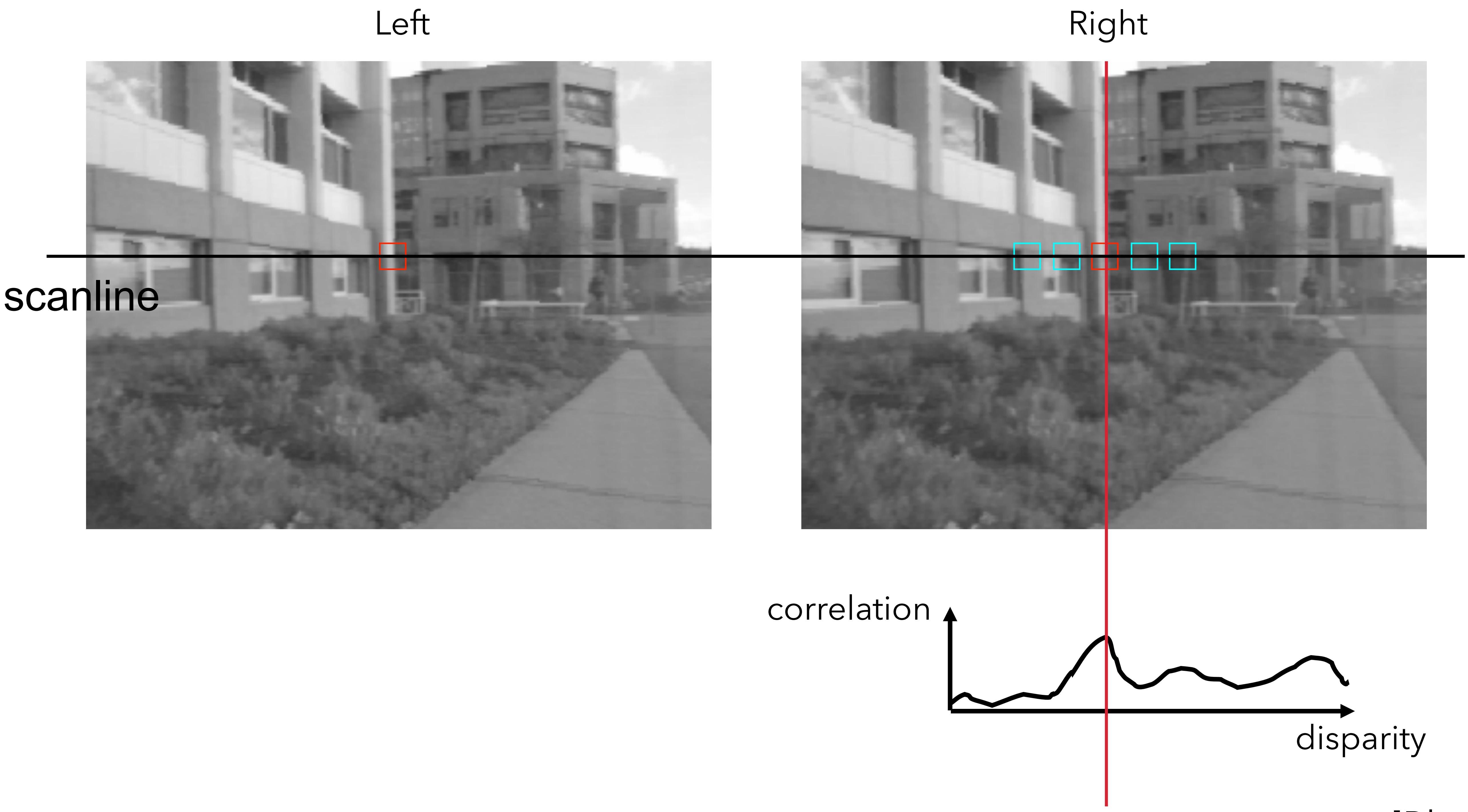
$$d = x - x' = \frac{B \cdot f}{Z}$$

disparity



- Parallel image planes: disparity from equal triangles

Correspondence Using Correlation



[Black]

Window-based Correlation

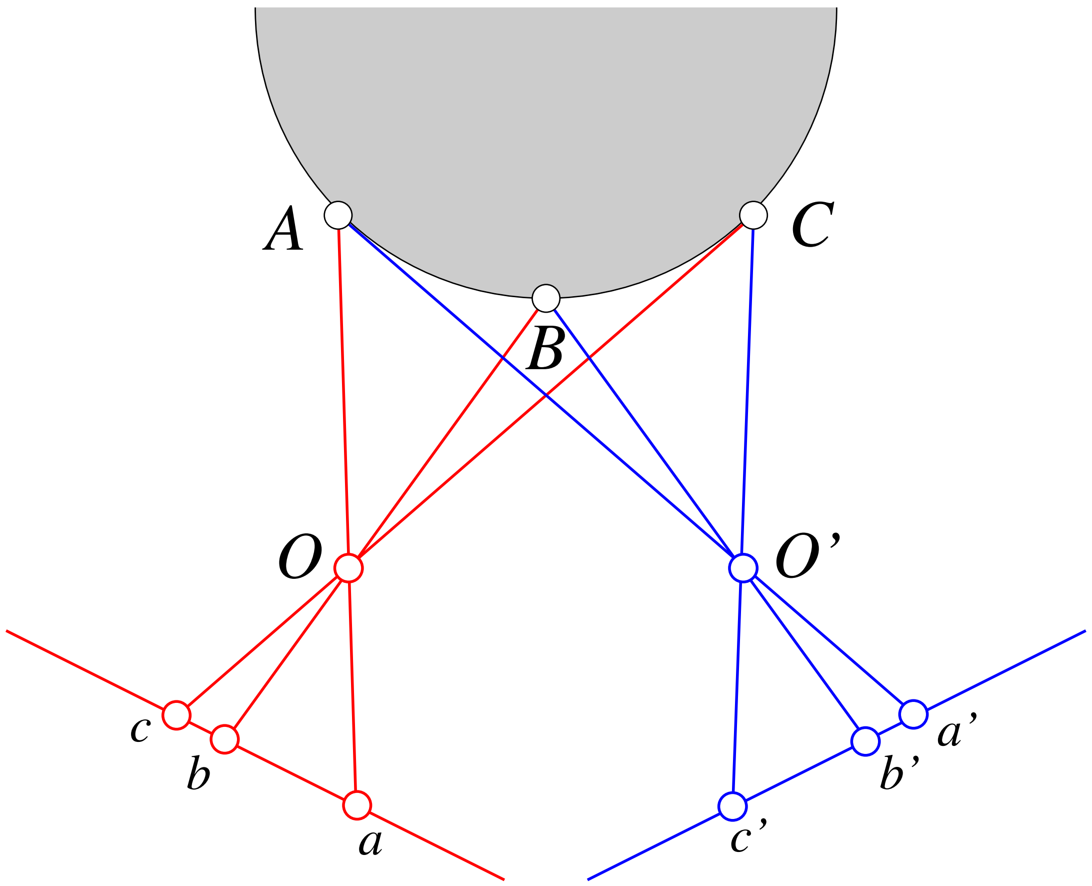
$m = 3$ $m = 20$

[Seitz]

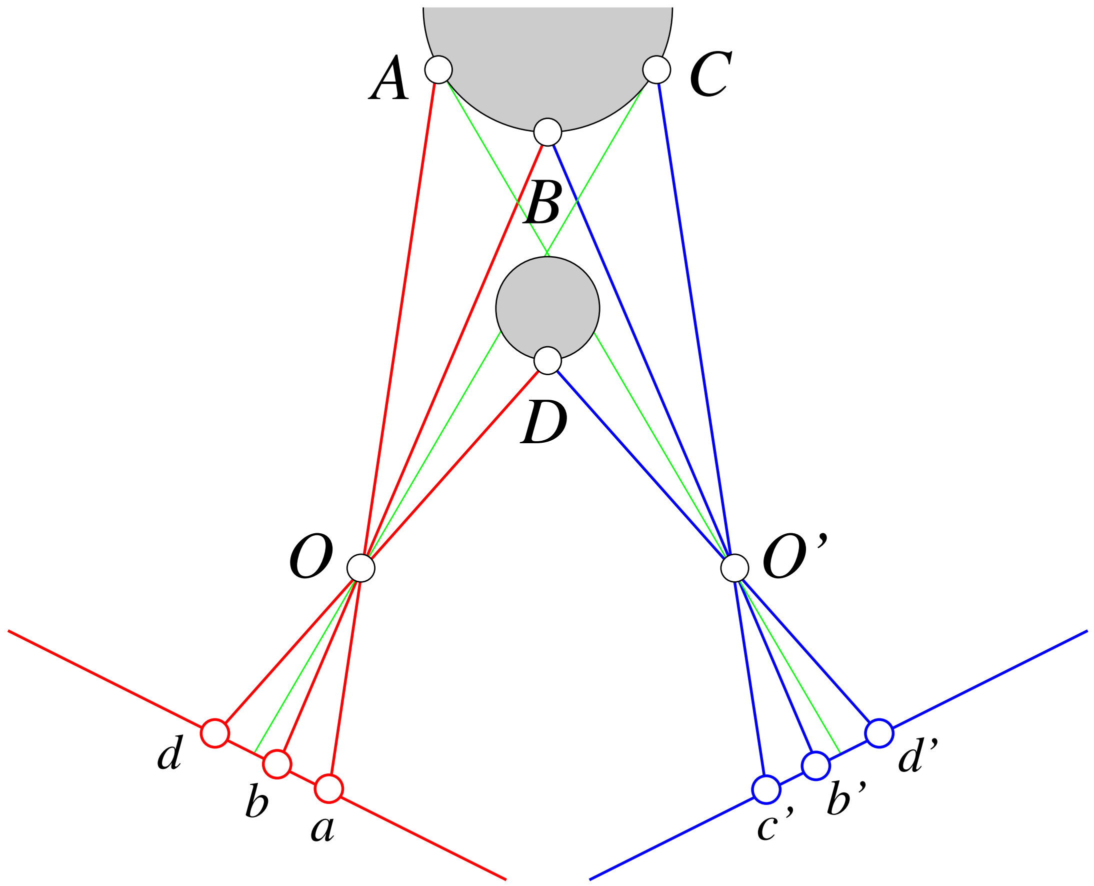
- We identified the following problems:
  - If the window is small, window-based matching often matches to an incorrect location, because the patch is not descriptive enough.
  - Large windows improve the matching, but lead to other artifacts, because the disparity is not constant within the window, etc.

Ordering Constraint

- Assume that the order of features on the epipolar line is the inverse of that in 3D:



Ordering constraint works fine



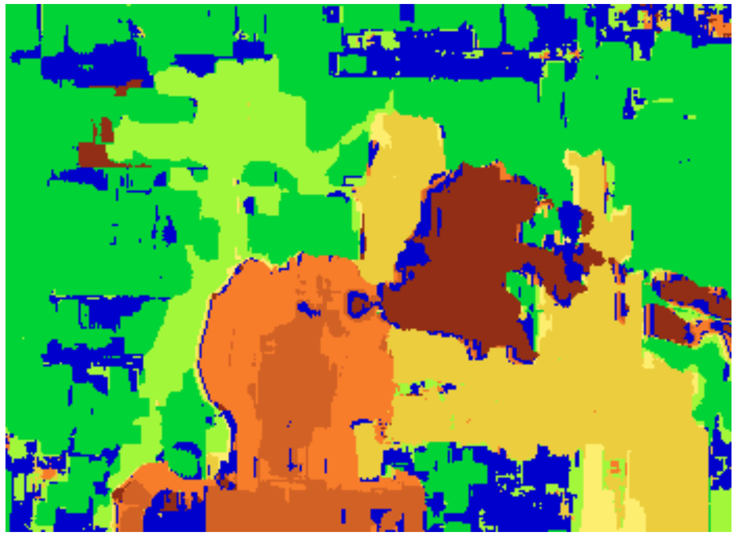
Failure!

From [Ponce & Forsyth]

Cooperative Stereo

TECHNISCHE UNIVERSITÄT DARMSTADT

- First proposed by Marr & Poggio in 1976:
  - Assume uniqueness of matches:
    - No pixel or feature can be matched twice.
  - Assume ordering constraint.
- Why do we want to still use these constraints?
  - They help resolve ambiguities in texture-less areas.
  - Encourage spatial continuity.
  - Improve the results by removing spurious depth discontinuities.



Previous problem:
Lots of spurious depth discontinuities

Stereo Matching with Dynamic Programming

- Results:


- Much better results than with window based matching!
- But no consistency between scanlines!

Today

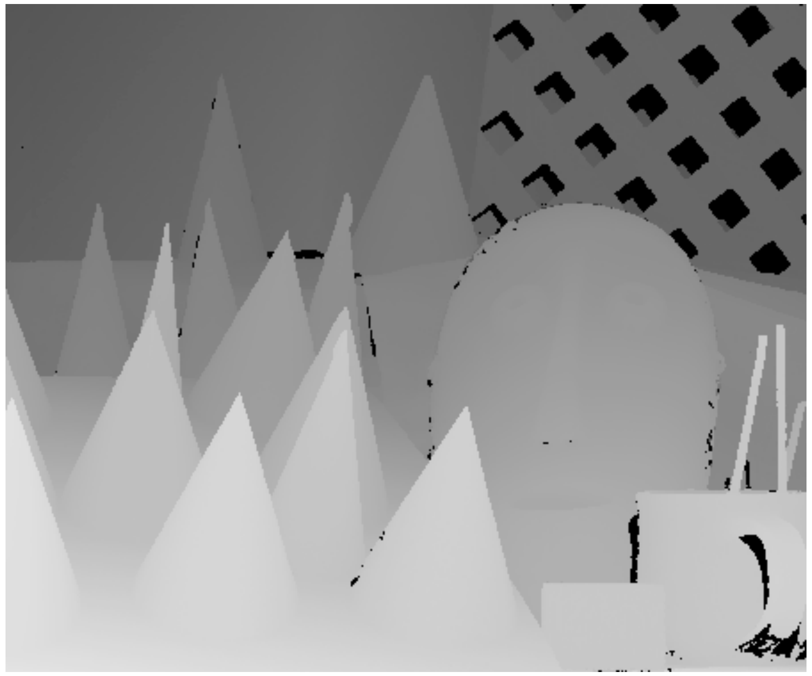
- How can we impose regularity constraints that impose global consistency?
  - This is also called **regularization**.
- Goals:
  - We want to go beyond matching windows.
  - We want consistency within and between scanlines.
  - We want a model of consistency that is well supported by the properties of the real world, e.g. by real scene depth or real motion.
  - We want a model that is computationally manageable.
  - We would like to find a model of consistency that does not only work for stereo or flow, but also for other applications.

Vision as Probabilistic Inference

- In the introduction we discussed that:
  - Many vision problems are underconstrained and require prior knowledge to be solved (as is the case here).
  - We almost always have to deal with uncertain ("noisy") data.
- Both of these are key motivations for using **probabilistic approaches** to computer vision:
  - We regard both the measurement and the interpretation as uncertain (Bayesian approach).
  - We model the problem using probabilities (or prob. densities).
  - We obtain the solution using methods of probabilistic inference.
- What does all this mean?

Stereo using Probabilistic Methods

Uncertain image measurements



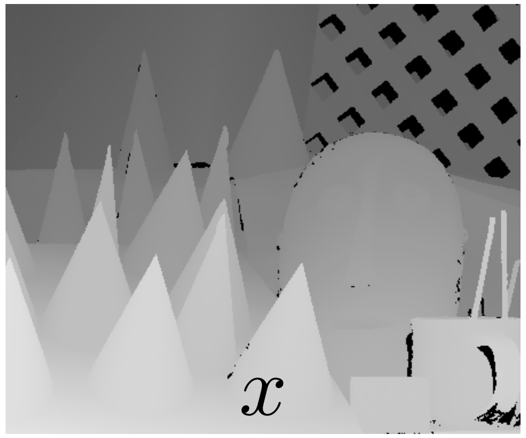
Uncertain state of the world

We're given

Want to know

Stereo using Probabilistic Methods

Uncertain image measurements



Uncertain state of the world

- Model using posterior distribution: $p(\text{state}|\text{images})$
  - Describe the probability of the state of the world given the image measurements.
- How do we find the "best" state of the world?
  - Using probabilistic inference, e.g. we maximize w.r.t. state x

Modeling the Posterior

- How do we model the posterior?
  - This can be done directly (discriminative approaches), but we will not do this now as it is more difficult.
- Instead, we simplify the modeling problem by applying Bayes' rule (generative approach):

likelihood (observation model)

prior

$$\underbrace{p(\text{state}|\text{images})}_{\text{posterior}} = \frac{\overbrace{p(\text{images}|\text{state})}^{\text{likelihood (observation model)}} \cdot \overbrace{p(\text{state})}^{\text{prior}}}{\underbrace{p(\text{images})}_{\text{normalization term (constant)}}}$$

posterior

normalization term (constant)

Modeling the Posterior

likelihood (observation model)

likelihood (observation model)

prior

prior

$$p(\text{state}|\text{images}) = \frac{p(\text{images}|\text{state}) \cdot p(\text{state})}{p(\text{images})}$$

$$p(x|y) = \frac{p(y|x)\,p(x)}{p(y)}$$

posterior

posterior

normalization term (constant)

- The likelihood $p(y|x)$ is an observation model that describes how we obtained the image measurements, given a particular state of the world.
- The prior $p(x)$ models our a-priori assumptions about the world, or the state of the world.
- The normalization term $p(y)$ can often be ignored, because it only depends on the image measurements, which are given to us.

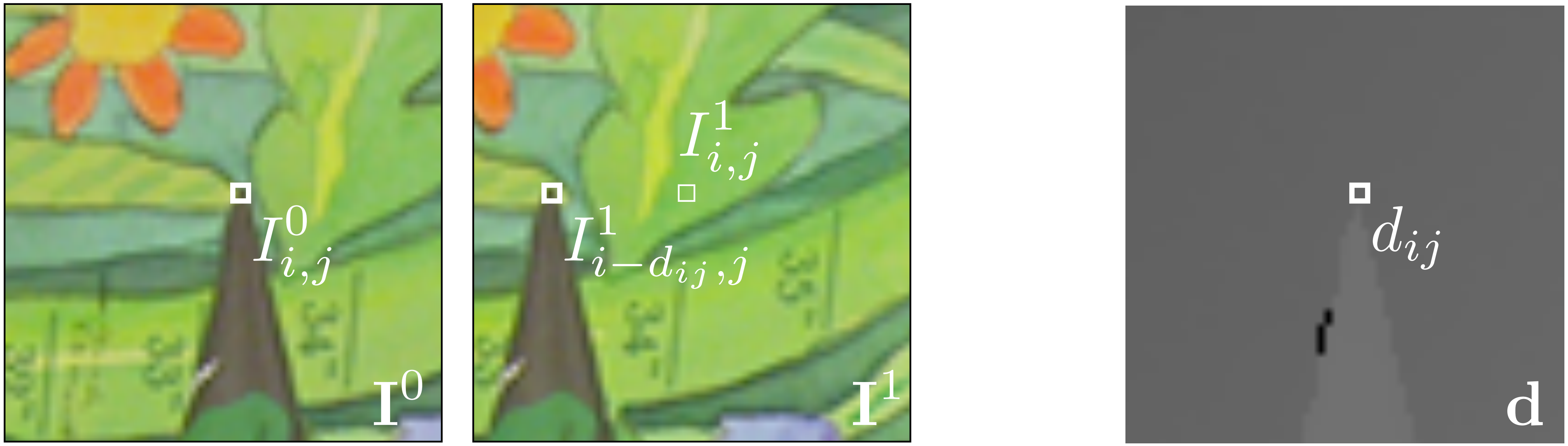
Modeling the Likelihood

- Again: The likelihood $p(y|x)$ is an observation model that describes how we obtained the image measurements, given a particular state of the world.
  - In stereo, the likelihood describes how consistent the measure image pair is, given the disparity (the "state of the world"):
    $$p(\mathbf{I}^0, \mathbf{I}^1 | \mathbf{d})$$
  - We typically assume conditional independence of the pixels, that is given the disparity, we assume that the intensity of the different pixels sites is independent.

$$\begin{aligned} p(\mathbf{I}^0, \mathbf{I}^1 | \mathbf{d}) &= \prod_{i,j} p(I^0_{i,j}, \mathbf{I}^1 | \mathbf{d}) \\ &= \prod_{i,j} p(I^0_{i,j}, \mathbf{I}^1 | d_{ij}) \end{aligned}$$

Only depends on disparity at (i, j)

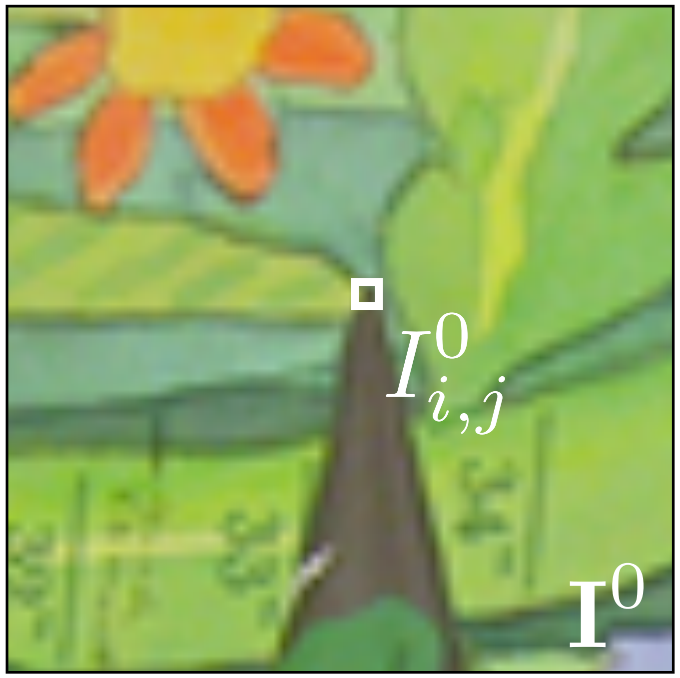
Modeling the Likelihood

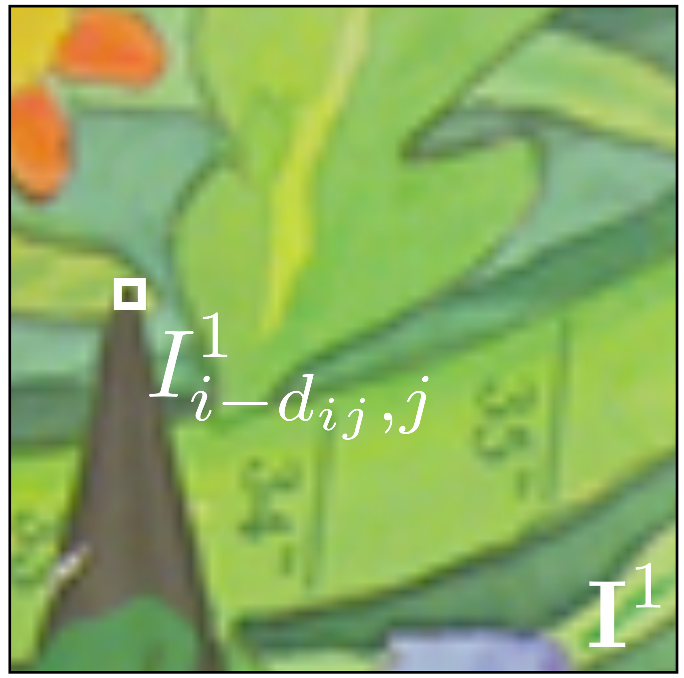


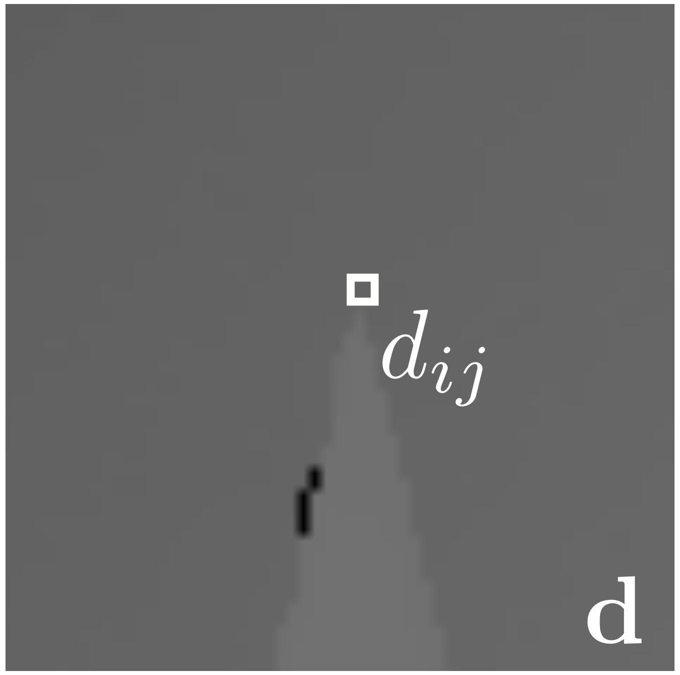
- A simple model:
  - We test how well the corresponding **pixels** match.

$$
\begin{aligned}
p(\mathbf{I}^0, \mathbf{I}^1 | \mathbf{d}) &= \prod_{i,j} p(I^0_{i,j}, \mathbf{I}^1 | d_{ij}) \\
&= \prod_{i,j} f\left(I^0_{i,j} - I^1_{(i-d_{ij}),j}\right)
\end{aligned}
$$

Modeling the Likelihood







- $f(\cdot)$ is a probabilistic model of how well two pixels match that are related by the local disparity.
  - How do we choose it?
  - We could just assume that it is Gaussian, no? Sure.

$$p(\mathbf{I}^0, \mathbf{I}^1 | \mathbf{d}) = \prod_{i,j} f(I^0_{i,j} - I^1_{(i-d_{ij}),j}) = \prod_{i,j} \mathcal{N}(I^0_{i,j} - I^1_{(i-d_{ij}),j};\ 0, \sigma^2)$$

Probability vs. Cost

- Even if it is not obvious, we have seen this model already.
  - Let's take the negative log:

$$
\begin{aligned}
-\log p(\mathbf{I}^0, \mathbf{I}^1 | \mathbf{d}) &= -\log\left(\prod_{i,j} \mathcal{N}(I^0_{i,j} - I^1_{(i-d_{ij}),j};\ 0, \sigma^2)\right) = \sum_{i,j} -\log \mathcal{N}(I^0_{i,j} - I^1_{(i-d_{ij}),j};\ 0, \sigma^2) \\
&= \sum_{i,j} \frac{1}{2\sigma^2}(I^0_{i,j} - I^1_{(i-d_{ij}),j})^2 + NM \log(\sqrt{2\pi}\sigma) \\
&= \frac{1}{2\sigma^2} \sum_{i,j} (I^0_{i,j} - I^1_{(i-d_{ij}),j})^2 + \text{const}
\end{aligned}
$$

  - This is just like the SSD matching criterion, but with a "window" size of 1.
  - So if we just had this likelihood, maximizing it w.r.t. the disparity is the same as minimizing the SSD cost.

Probability vs. Cost vs. Energy

TECHNISCHE UNIVERSITÄT DARMSTADT

- More generally:
  - We can relate cost functions and probabilities.
  - Costs are often interpreted as energies (in the physical sense).
    - A solution with low cost is like a low-energy state that a physical system likes to settle in.
  - Probability and energy can be related through the Boltzmann (or Gibbs) distribution:

temperature

energy of a state

$$p(x) = \frac{1}{Z(T)} e^{-\frac{1}{T} E(x)}$$

probability of a state

normalization term (partition function)

Probability vs. Cost vs. Energy

TECHNISCHE UNIVERSITÄT DARMSTADT

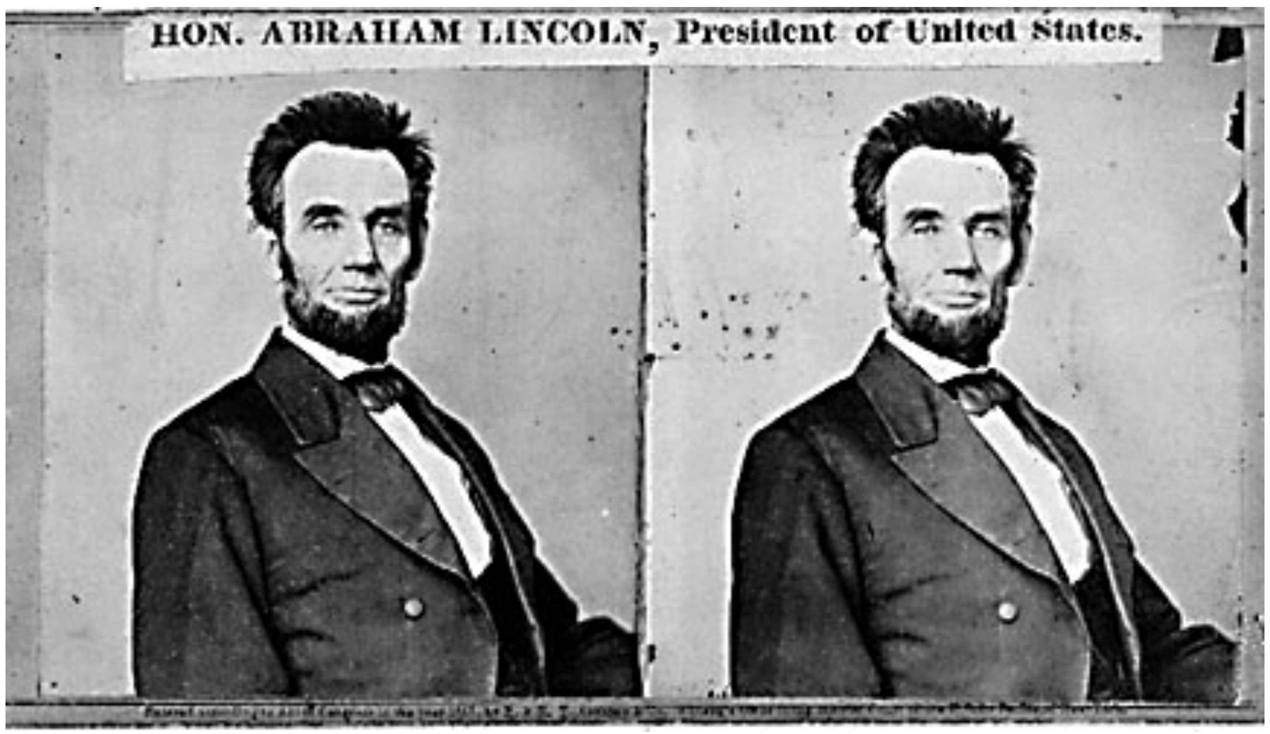
- Where do we get the temperature from? Does it matter?
  - That depends on how we infer the state of the world.
- If we maximize the posterior, then the temperature does not matter.
  - This is called **maximum a-posteriori** (MAP) estimation (more next class).
- If we, for example, compute the mean of the posterior, then the temperature generally does matter.

Two Questions

- Why did we need a window when we did matching along the scanlines (epipolar lines) using the correlation or the SSD, when now we use a single pixel?
  - We (will) have a prior that helps us overcome this! Hold on...
- Is it reasonable to assume a Gaussian distribution when modeling the likelihood?
  - As a first approximation, yes. But we may suffer from the same problems that we discussed in conjunction with SSD-based matching (cf. CV1).
  - Occlusions, shading, shadows, gain control, etc. can make the assumption inappropriate.

Limitations of brightness constancy

TECHNISCHE UNIVERSITÄT DARMSTADT



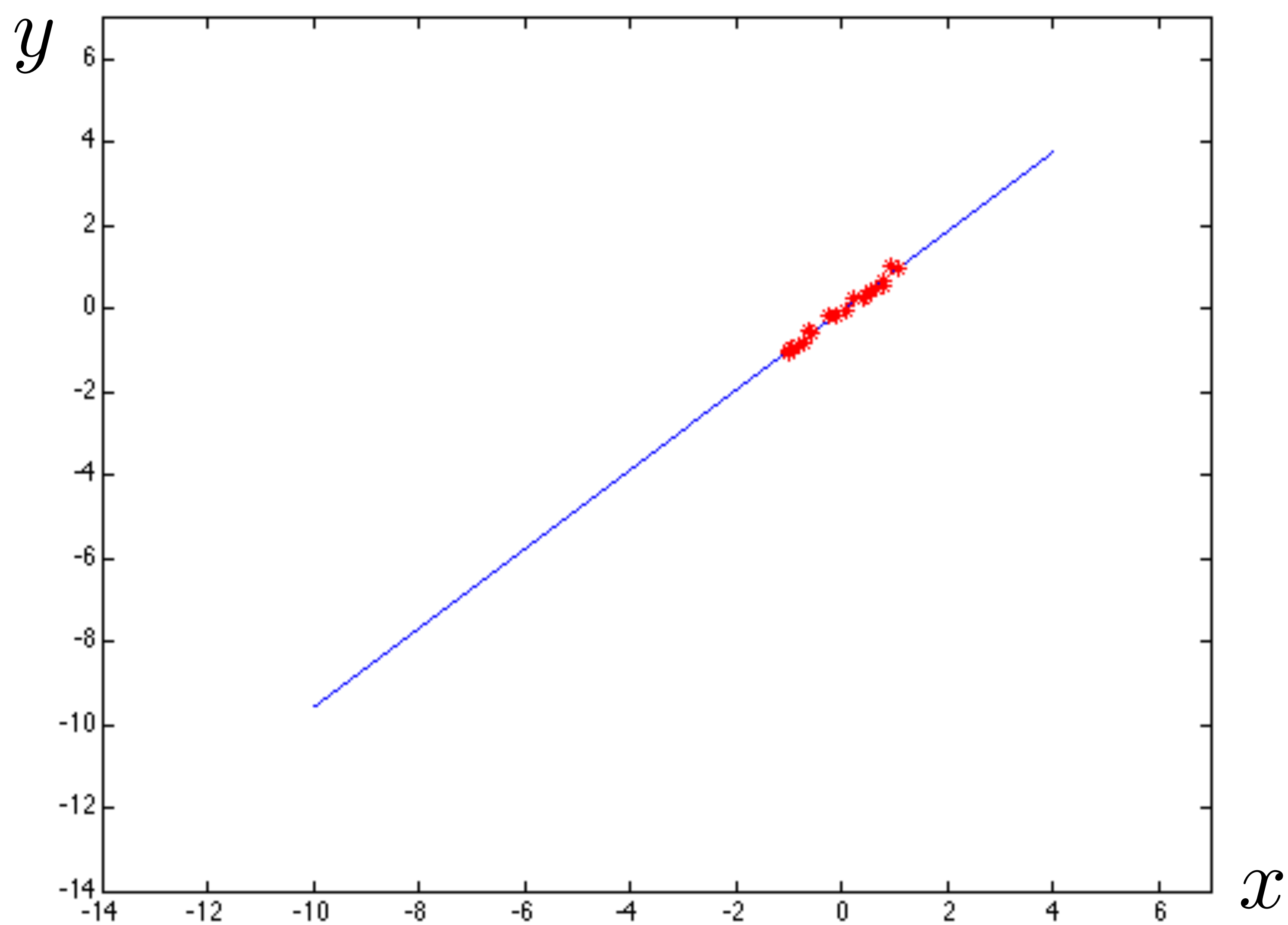
Noise, textureless surfaces

Non-Lambertian surfaces, specularities

How can we match or "fit" our model robustly?

Excursion: Fitting Lines

- Fitting models to data can get arbitrarily complex, but at the end of the day there is an important base case.
- Fitting lines to data:



[FP]

Least Squares

- First attempt: Least squares fit (Gauss)
  - Assume a simple parametric model for the line and minimize the least squares error.
  - Line model: $$y = ax + b$$
  - Sum of squared differences: $$\sum_i (y_i - ax_i - b)^2$$
  - Least squares: $$(a^*, b^*) = \arg\min_{a,b} \sum_i (y_i - ax_i - b)^2$$
  - Differentiate w.r.t. a:

$$\frac{\partial}{\partial a} \sum_i (y_i - ax_i - b)^2 = -2 \sum_i x_i (y_i - ax_i - b)$$

Least Squares

- Differentiate w.r.t. b:

$$\frac{\partial}{\partial b} \sum_i (y_i - a x_i - b)^2 = -2 \sum_i (y_i - a x_i - b)$$

- Set both to zero and put into an equation system:

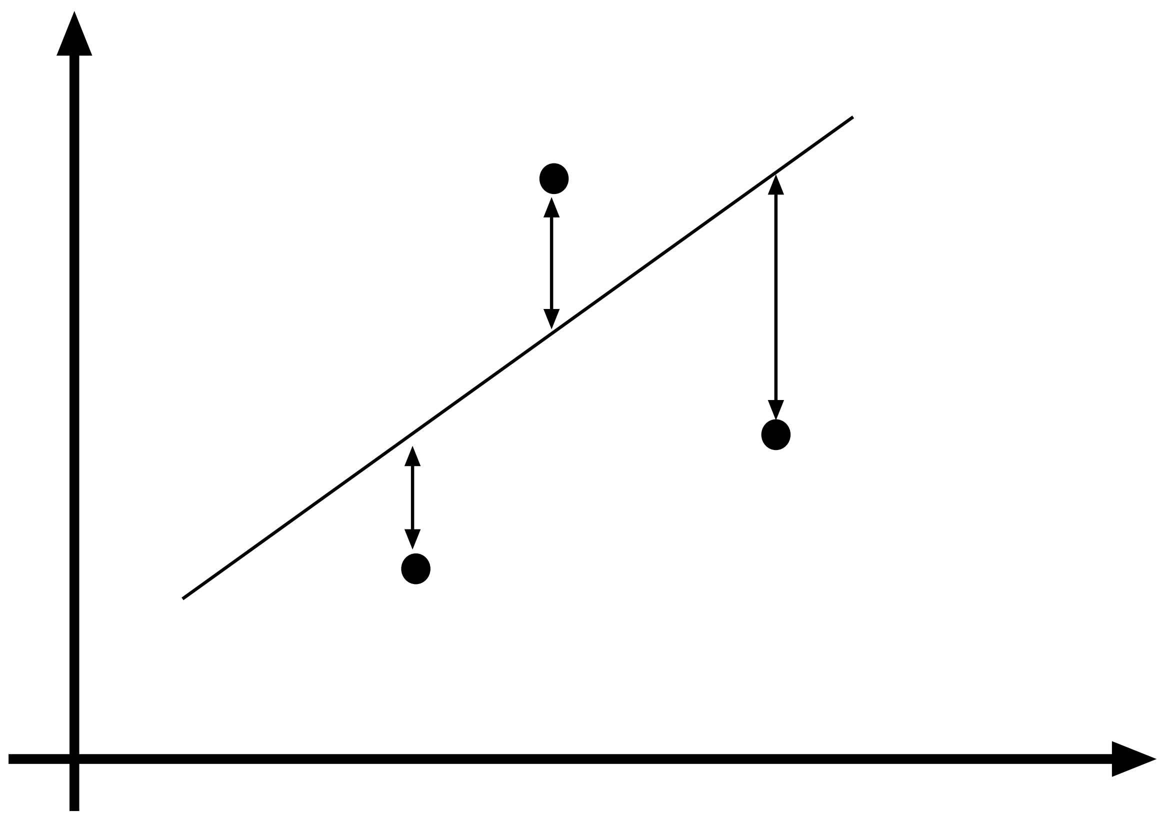
$$\begin{pmatrix} \sum_i x_i^2 & \sum_i x_i \\ \sum_i x_i & n \end{pmatrix} \begin{pmatrix} a \\ b \end{pmatrix} = \begin{pmatrix} \sum_i x_i y_i \\ \sum_i y_i \end{pmatrix}$$

- And solve...
  - Easy enough, but **two major problems**.

Problem 1

TECHNISCHE UNIVERSITÄT DARMSTADT

- Sensitivity to the coordinate system:

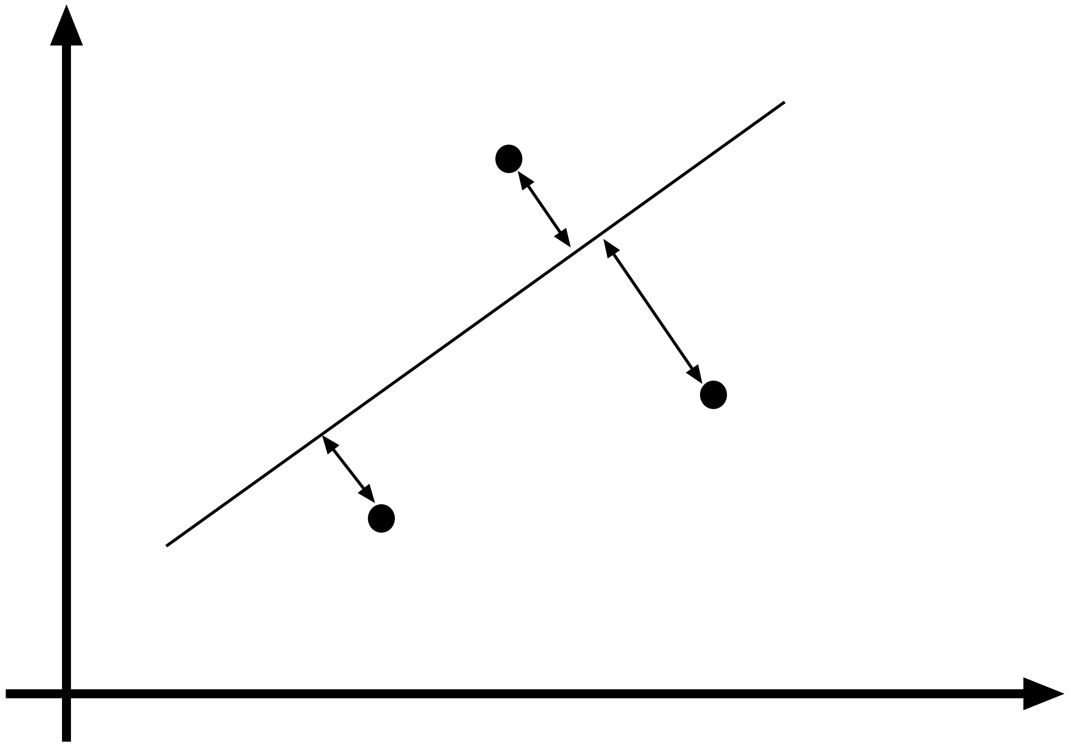


[FP]

  - We cannot represent vertical lines!
  - If we rotate our coordinate system, we get a different solution.
- Clearly not what we want.

Solution

TECHNISCHE UNIVERSITÄT DARMSTADT



- Don't measure the error in y-direction, but instead measure the **distance to the line** and use that to define the error.

Total Least Squares

TECHNISCHE UNIVERSITÄT DARMSTADT

- Next attempt: **Total Least Squares**
  - Different line representation that admits vertical lines:

$$ax + by + c = 0$$

  - Scaling problem: Any multiple also fulfills the line equation.
  - Introduce scaling constraint to remove ambiguity, e.g.:

$$a^2 + b^2 = 1$$

    - Note that we also have to fix the sign of one variable to remove any ambiguity.
  - Useful fact:
    - The distance to the line can be expressed as

$$|ax + by + c| \quad \text{s.t.} \;\; a^2 + b^2 = 1$$

    - See FP, exercise 15.1.

Total Least Squares

- Constrained minimization problem:

$$(a^*, b^*) = \arg\min_{a,b} \sum_i |ax_i + by_i + c|^2 \quad \text{s.t. } a^2 + b^2 = 1$$

- As usual, use a Lagrange multiplier λ
  - After similar steps as before, we obtain an equation system:

$$\begin{pmatrix} \sum_i x_i^2 & \sum_i x_i y_i & \sum_i x_i \\ \sum_i x_i y_i & \sum_i y_i^2 & \sum_i y_i \\ \sum_i x_i & \sum_i y_i & n \end{pmatrix} \begin{pmatrix} a \\ b \\ c \end{pmatrix} = \lambda \begin{pmatrix} 2a \\ 2b \\ 0 \end{pmatrix}$$

  - It follows that:

$$c = -\frac{a}{n} \sum_i x_i - \frac{b}{n} \sum_i y_i$$

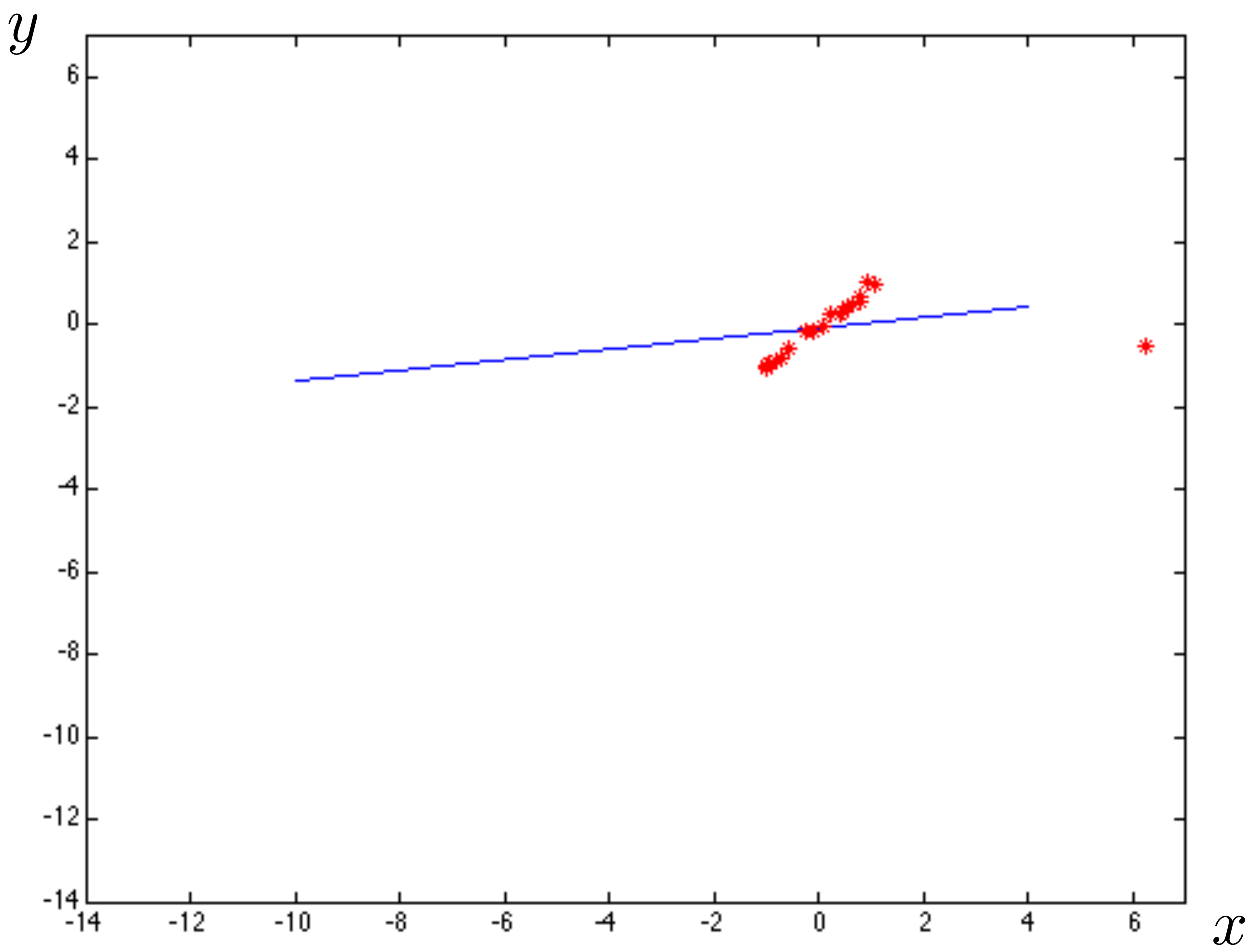
Total Least Squares

- Substituting this back in we obtain an eigenvalue problem:

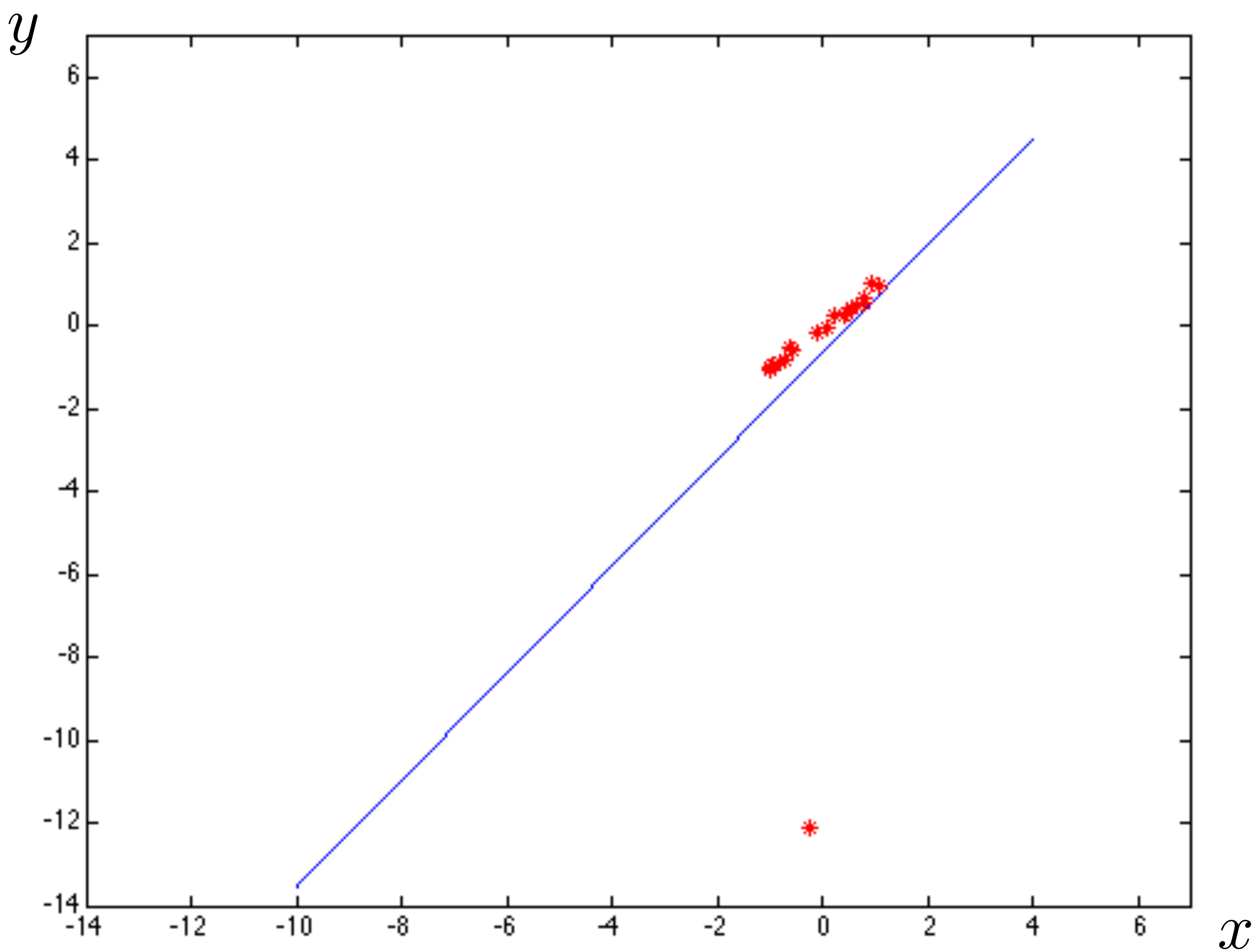
$$\begin{pmatrix} \sum_i x_i^2 - \frac{1}{n}(\sum_i x_i)^2 & \sum_i x_i y_i - \frac{1}{n}(\sum_i x_i)(\sum_i y_i) \\ \sum_i x_i y_i - \frac{1}{n}(\sum_i x_i)(\sum_i y_i) & \sum_i y_i^2 - \frac{1}{n}(\sum_i y_i)^2 \end{pmatrix} \begin{pmatrix} a \\ b \end{pmatrix} = \mu \begin{pmatrix} a \\ b \end{pmatrix}$$

  - Since eigenvectors can be scaled arbitrarily, we can easily fulfill the constraint.
  - We choose the eigenvector that minimizes the objective (as opposed to maximizes it).
- Much better, but we still have a problem...

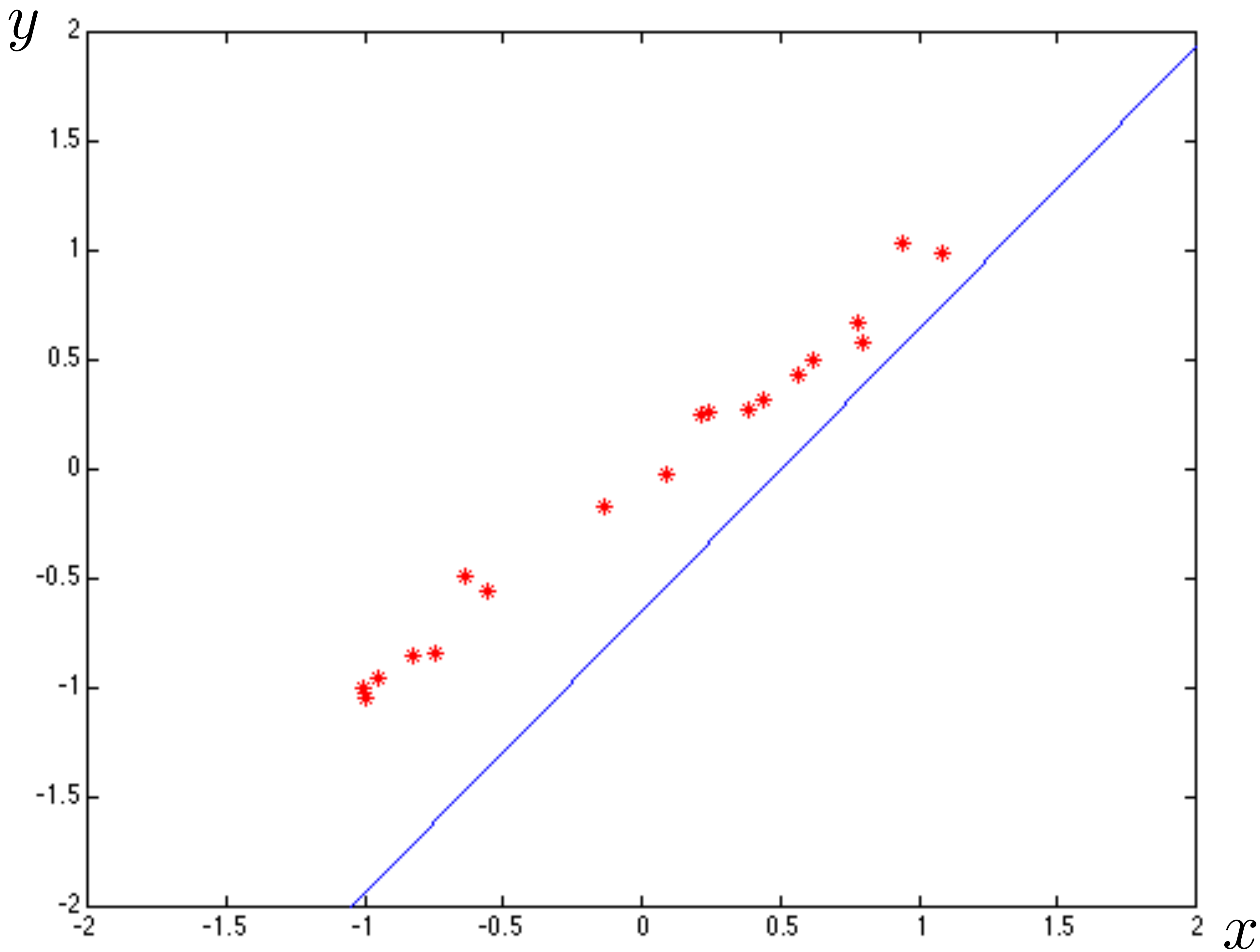
Problem 2: Outliers



Problem 2: Outliers



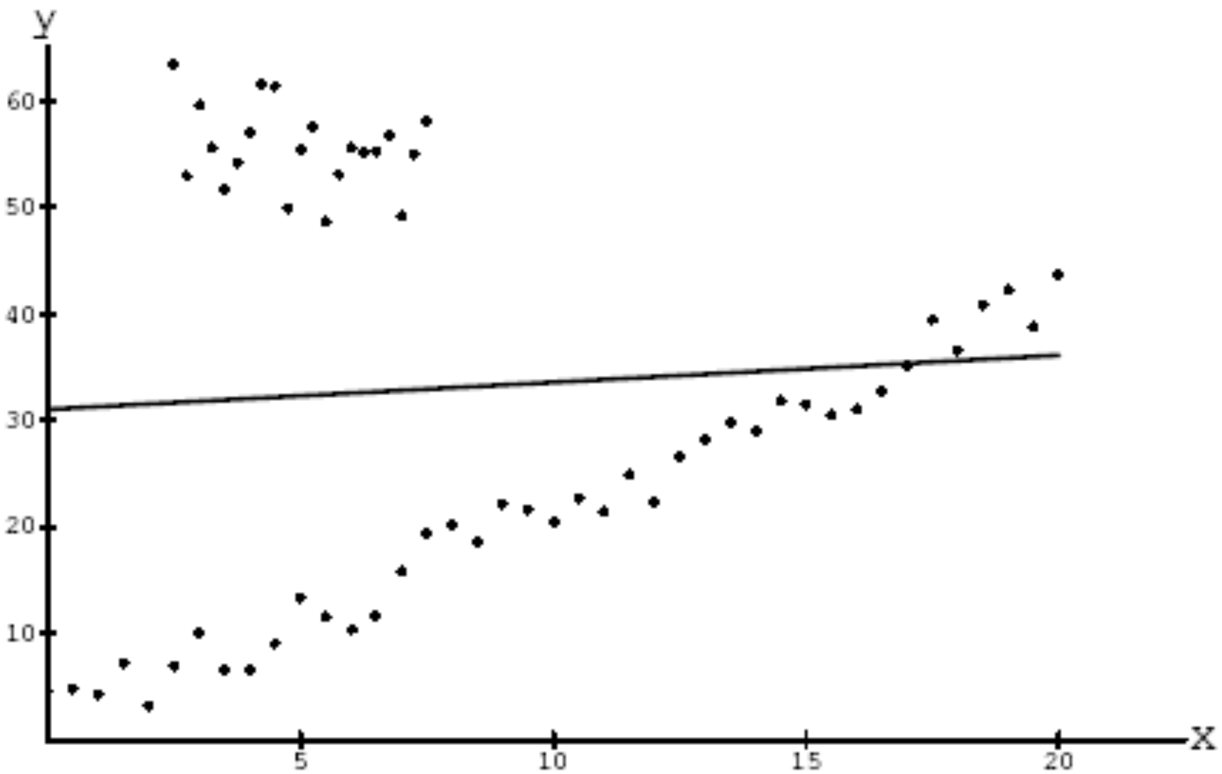
Problem 2: Outliers



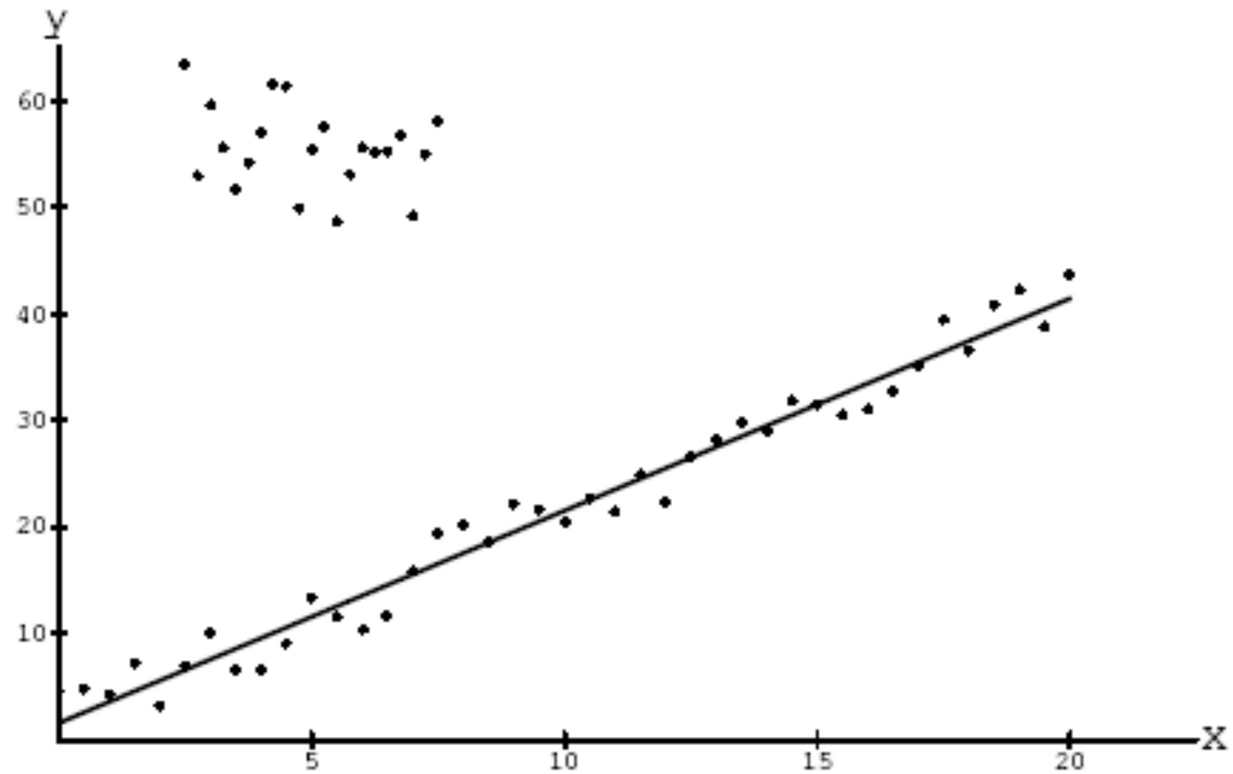
Zoomed in from previous figure

Another case of outliers

We have:



We want:



Problem: Already a single outlier can make the estimate arbitrarily bad.

Line Fitting as Inference

- To see how we can solve this problem, we will first reinterpret what we have done so far.
  - The data points we are trying to fit can be interpreted as being samples from a probability distribution.
- **Generative model** for the data points:
  - We first assume that for each data point (x_i, y_i) there is a true point (u_i, v_i) somewhere on the line to which we add noise that is orthogonal to the line.
  - If we for now assume that the noise is Gaussian, we can formalize this as:

$$\begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} u \\ v \end{pmatrix} + n \begin{pmatrix} a \\ b \end{pmatrix} \qquad \text{where} \qquad \begin{aligned} n &\sim \mathcal{N}(0, \sigma) \\ au + bv + c &= 0 \\ a^2 + b^2 &= 1 \end{aligned}$$

Line Fitting as Inference

TECHNISCHE UNIVERSITÄT DARMSTADT

- With this simple generative model, we can write the likelihood of our measurements $\mathcal{X} = \{(x_i, y_i), i = 1, \ldots, n\}$ as:

$$p(\mathcal{X}|a,b,c) = \prod_i p(x_i, y_i|a,b,c)$$

- To find the best line, we use the fact that the noise variable n is just the distance to the line:

$$\log p(\mathcal{X}|a,b,c) = -\frac{1}{2\sigma^2} \sum_i (ax_i + by_i + c)^2 + \text{const}$$

- This means that maximizing the log-likelihood of this probabilistic model is the same as total least squares.

Robust Fitting

- Now that we have formulated line fitting using probabilistic inference it is quite easy to fix our problem:
  - We have to change the Gaussian distribution assumption and use something **more robust**.
- **Robust fitting:**
  - Decrease the influence of outliers that are unlikely to correspond to the model that we want to fit.

Robust Fitting

- For now we will assume a general error distribution and define the log-likelihood as:

$$\log p(\mathcal{X}|a,b,c) = -\sum_i \rho(ax_i + by_i + c;\ \sigma) + \text{const}$$

  - Here, we use a robust error function: $\rho(x;\ \sigma)$
  - In the Gaussian case, we would have:

$$\rho(x;\ \sigma) = \frac{1}{2\sigma^2}x^2$$

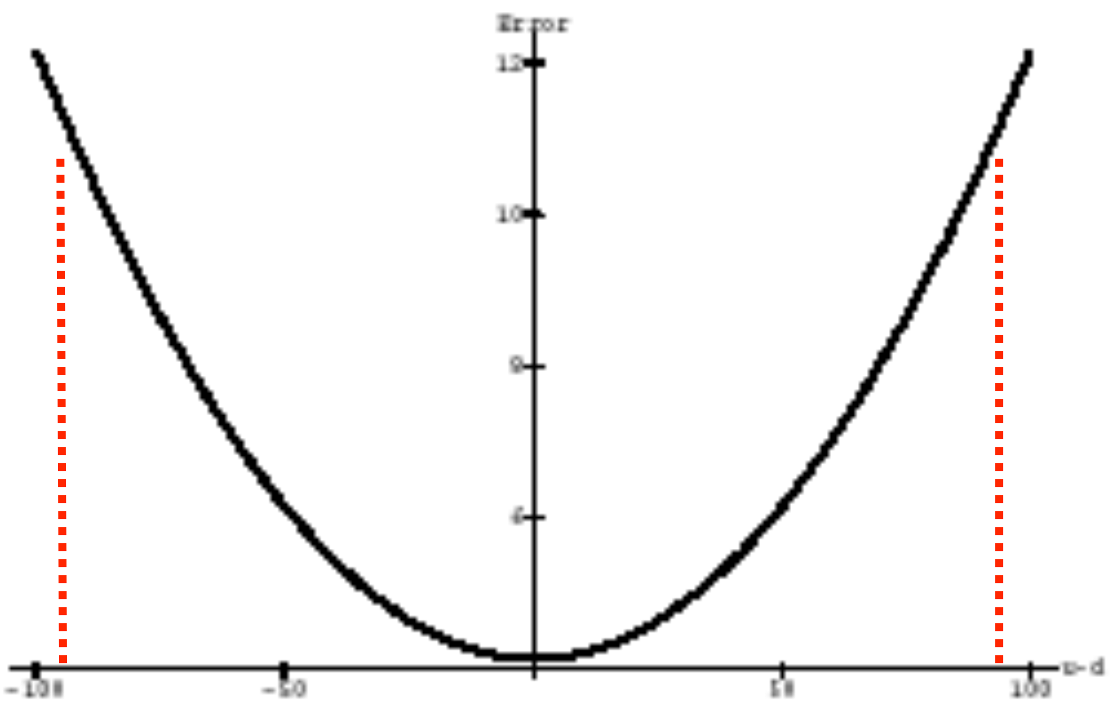
- To study appropriate robust error functions and the associated distributions, let us formalize a few things...

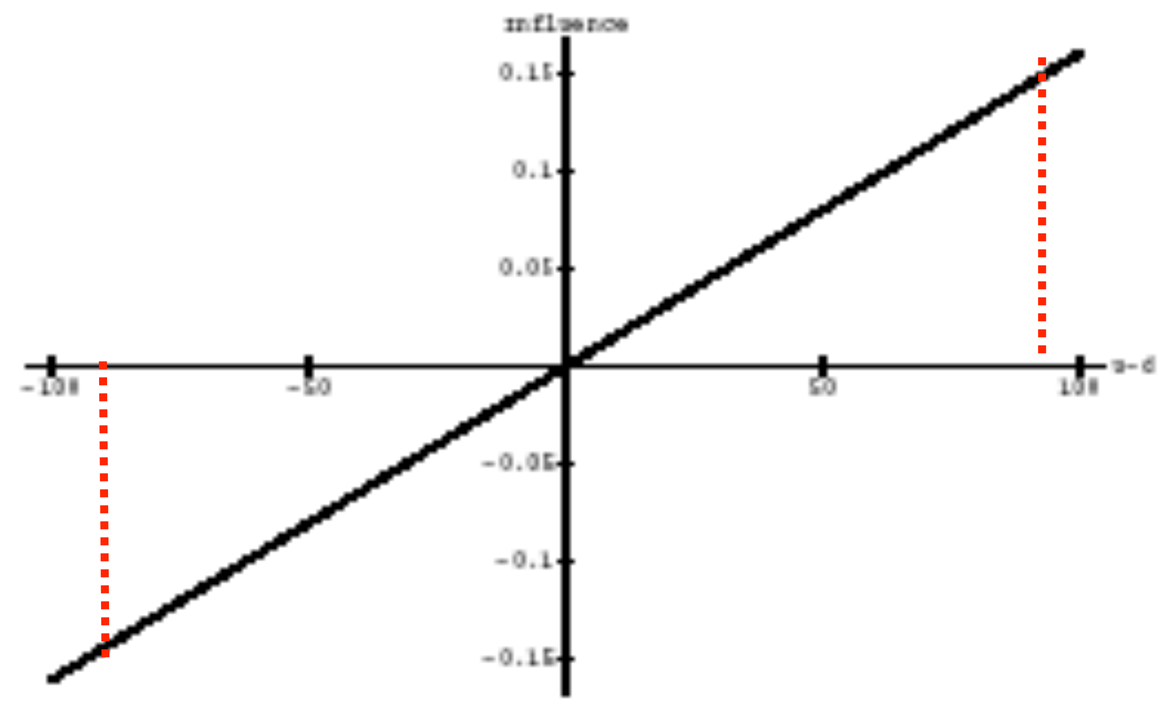
Influence & Breakdown point

- **Breakdown point:**
  - Percentage of outliers required to make the solution arbitrarily bad.
- **Influence:**
  - The dependency of the estimation on how “bad” the outlier is.
- For least squares or total least squares:
  - The influence of an outlier is linear, i.e. the estimate is linearly affected by the outlier.
  - The breakdown point is 0% - not robust!

Influence of the Squared Error

- The influence is proportional to the derivative of the error function.

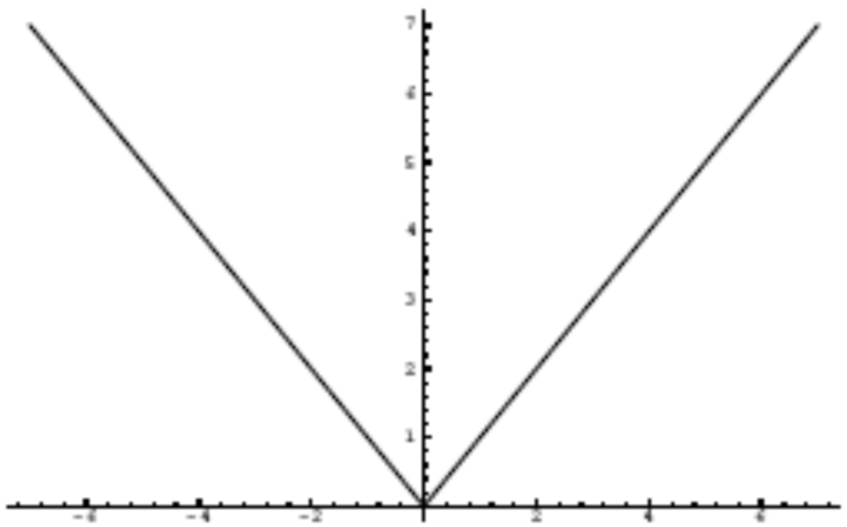




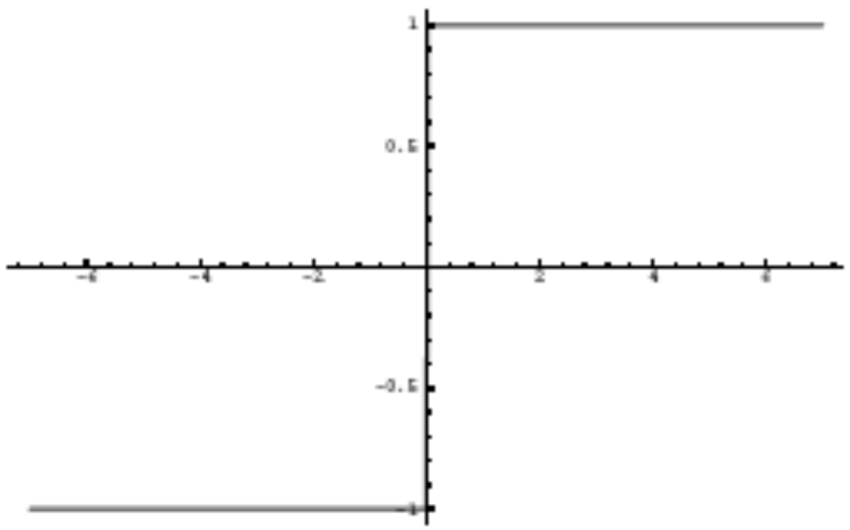
Want to give less influence to points beyond some value.

L1 Norm

- Another popular error function is the absolute value function ("L1"):
  - This corresponds to the exponential or Laplacian distribution.



$$\rho(x;\ \sigma) = \frac{|x|}{\sigma}$$



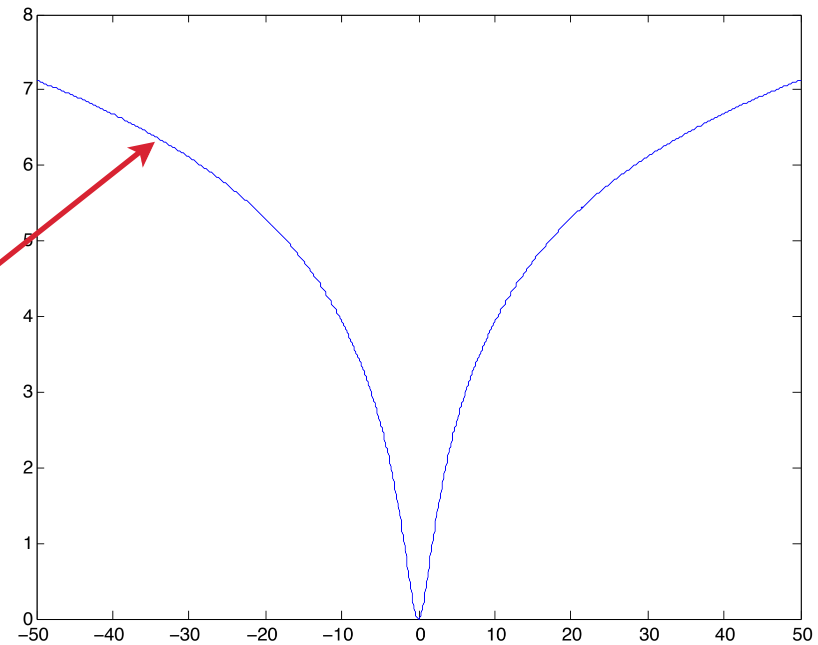
$$\rho'(x;\ \sigma) = \frac{1}{\sigma}\mathrm{sgn}(x)$$

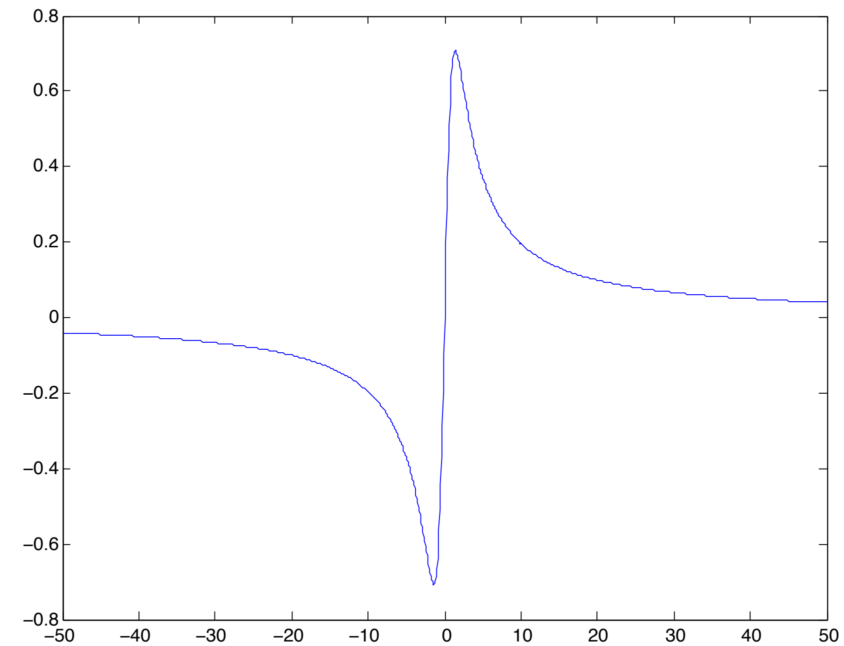
  - Problem: The influence is still too high - not very robust either.

Lorentzian - Student-t

- We can also look at the Student-t distribution and the associated error function (Lorentzian):

heavy-tailed distribution



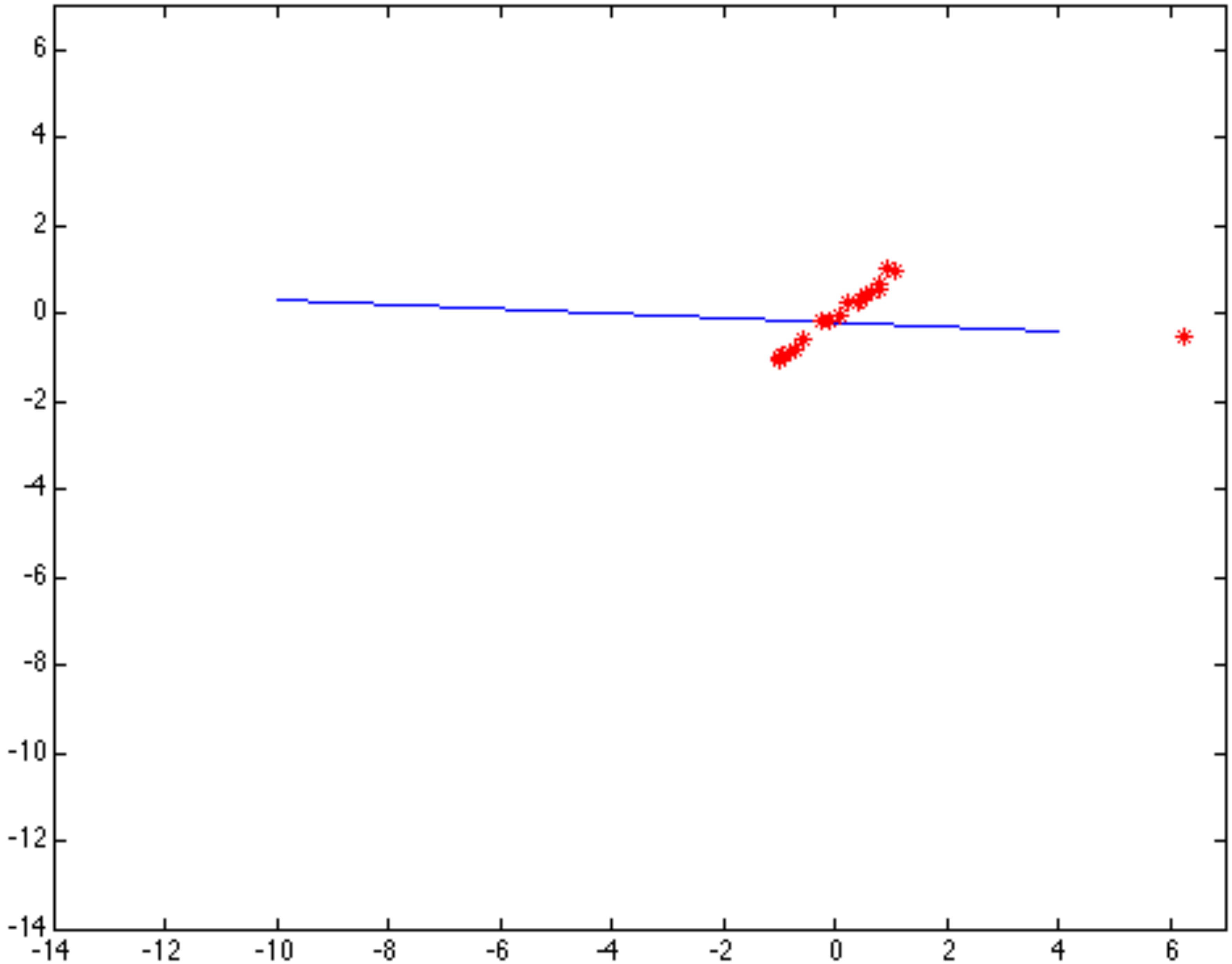


$$\rho(x;\ \sigma) = \log\left(1 + \frac{1}{2\sigma^2}x^2\right) \qquad \rho'(x;\ \sigma) = \frac{2x}{2\sigma^2 + x^2}$$

- The influence function is **redescending**, i.e. the influence goes down at some point - robust!

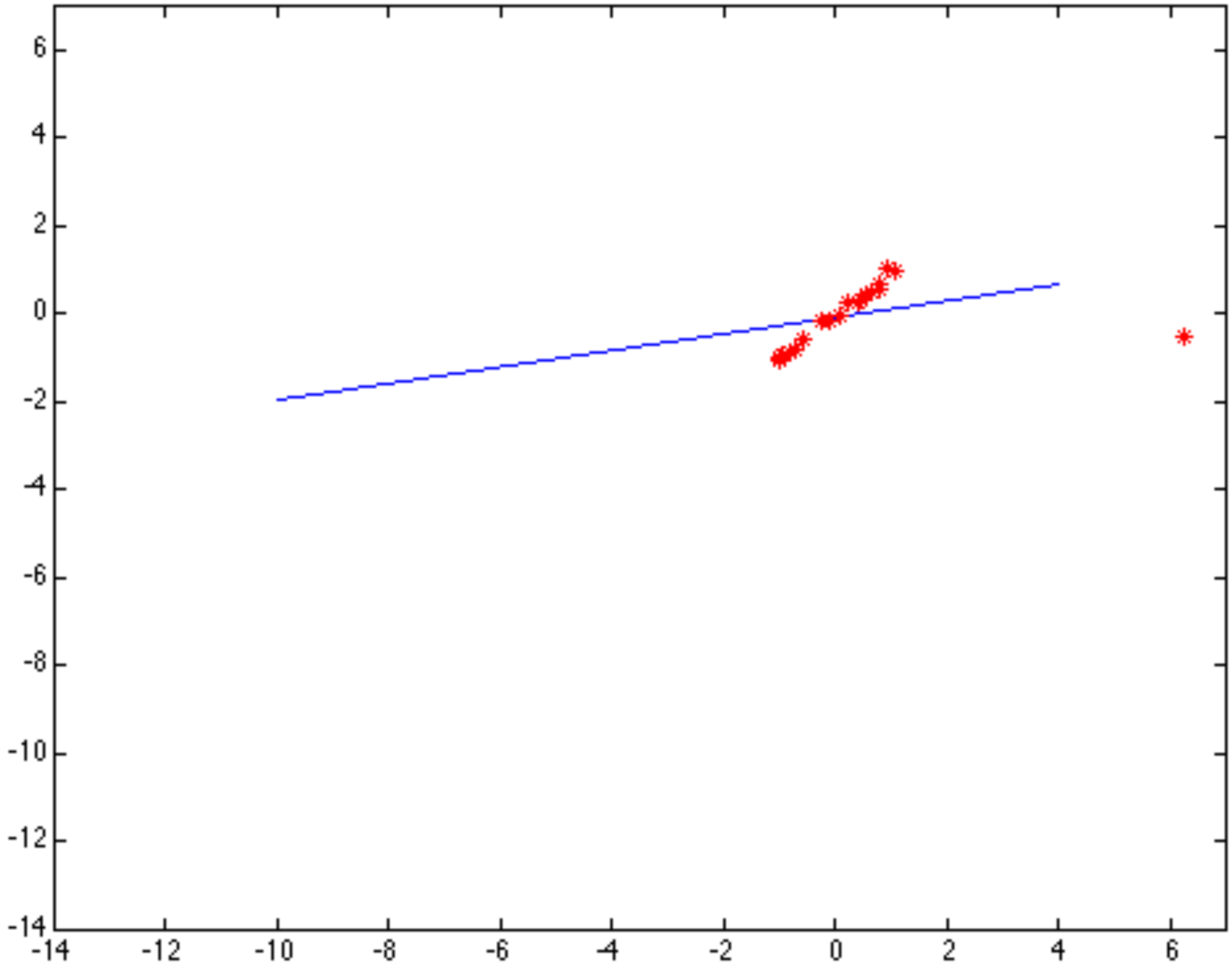
Choice of the robust scale

Too small



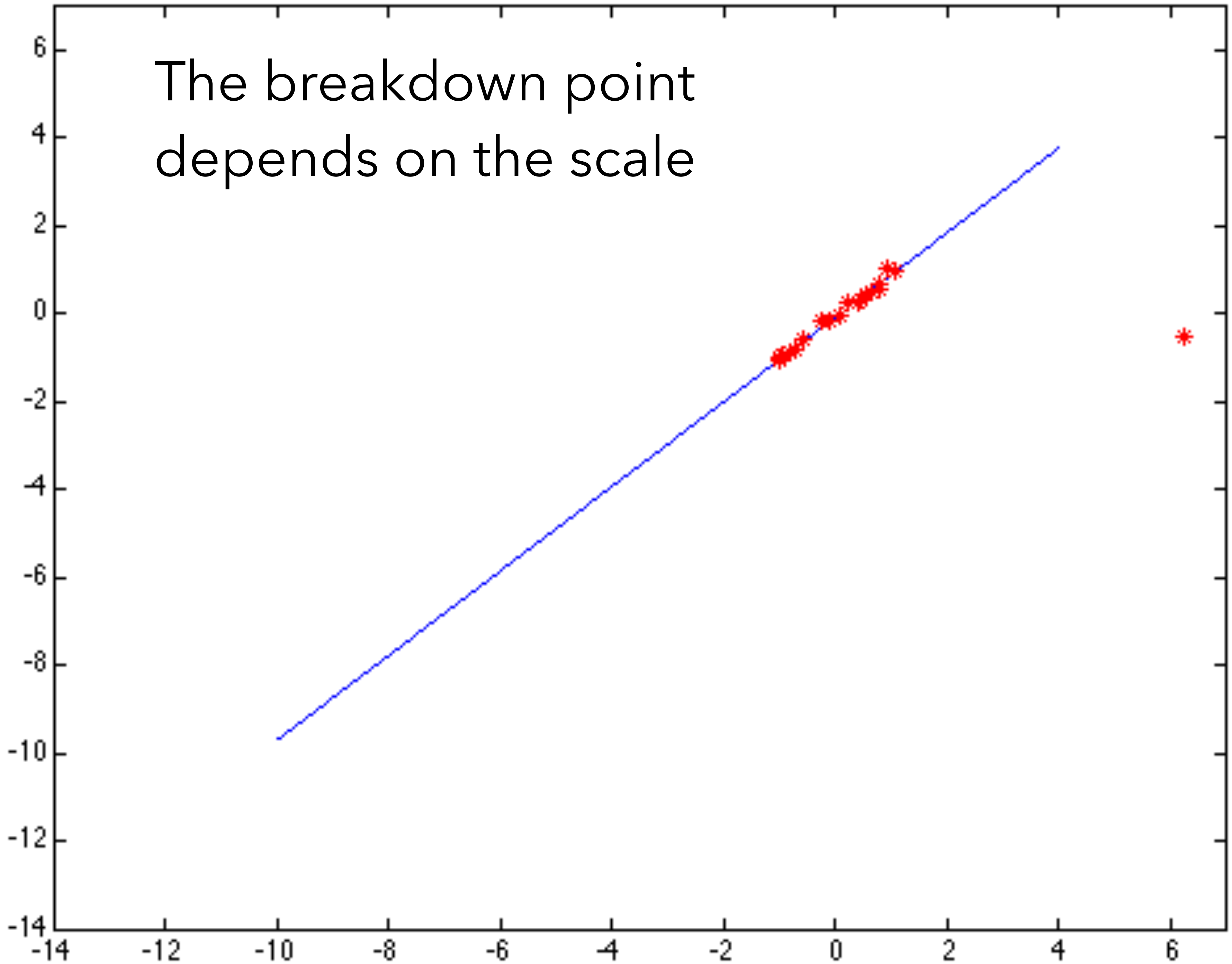
Choice of the robust scale

Too large

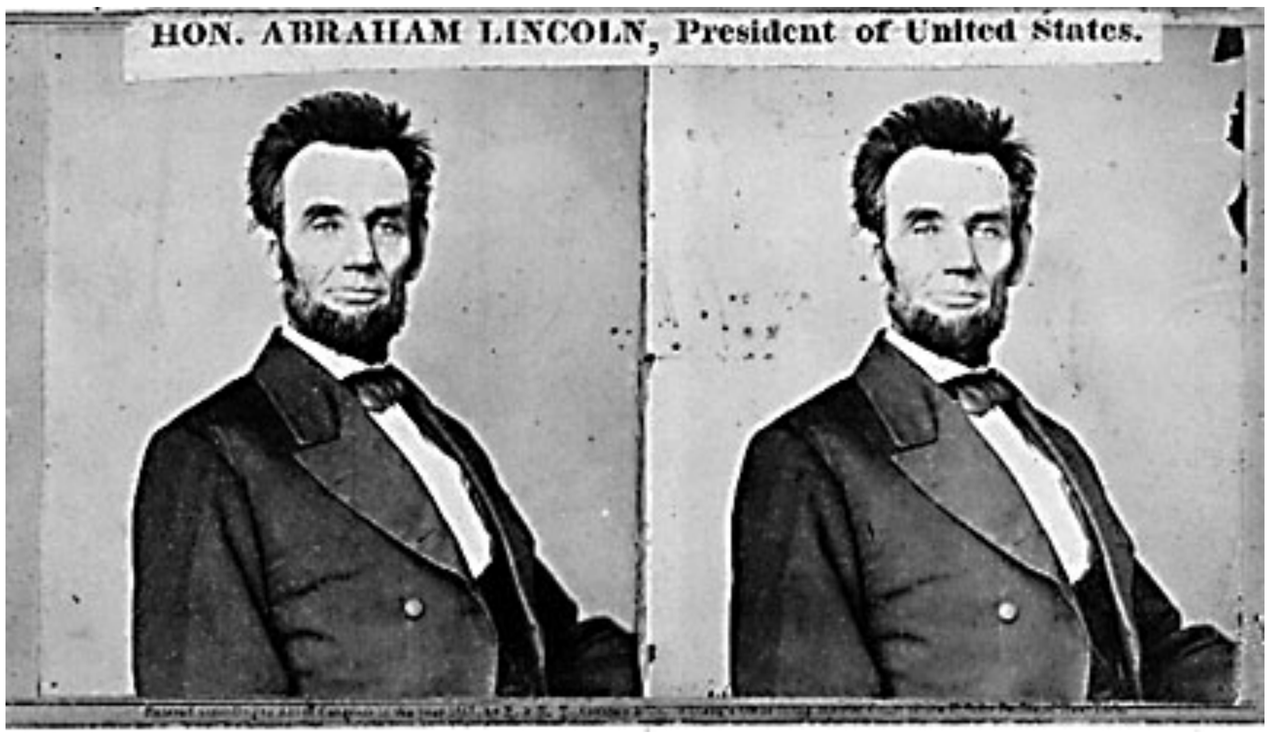


Choice of the robust scale

Just right



Limitations of brightness constancy



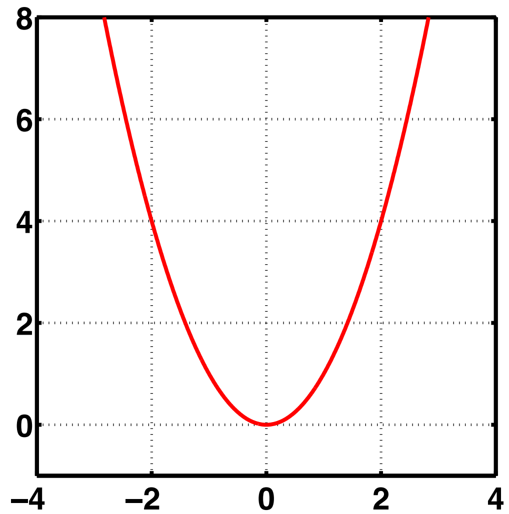
Noise, textureless surfaces

Non-Lambertian surfaces, specularities

How can we match or “fit” our model robustly?

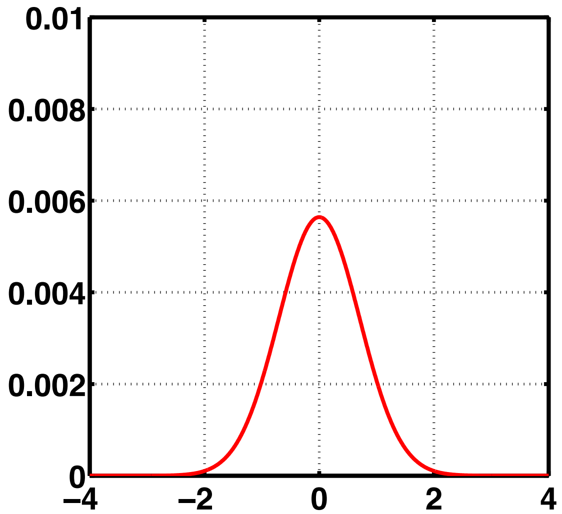
Gaussian Likelihood

- Because of such effects, the Gaussian assumption is likely to be inappropriate.
  - We can use a **robust** likelihood model that is more robust to such errors in the modeling assumption.
    - Remember: We assumed that the brightness of an image feature is the same in both cameras...
    - We discussed robust error functions to define costs.
  - Similarly, we can use (more) robust probability distributions.



Cost or energy

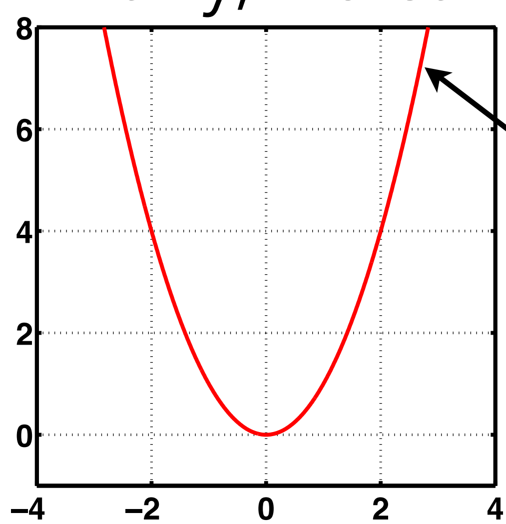
$$p(x) = \frac{1}{Z} e^{-E(x)} = \frac{1}{Z} e^{-x^2}$$



Probability density

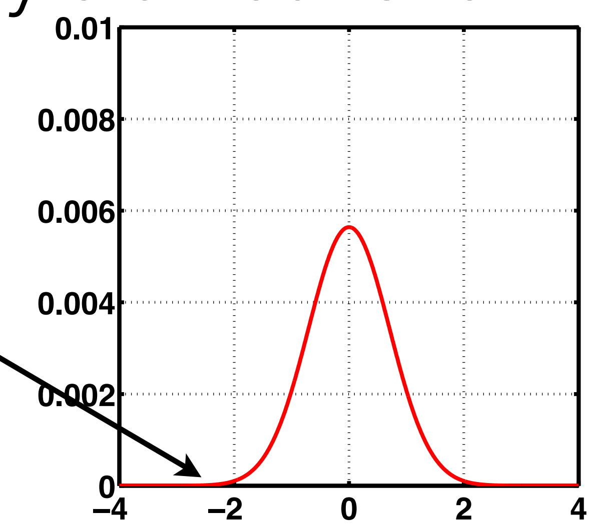
Gaussian Likelihood

- Because of such effects, the Gaussian assumption is likely to be inappropriate.
  - We can use a **robust** likelihood model that is more robust to such errors in the modeling assumption.
    - Remember: We assumed that the brightness of an image feature is the same in both cameras...
    - We discussed robust error functions to define costs.
  - Similarly, we can use (more) robust probability distributions.



Cost or energy

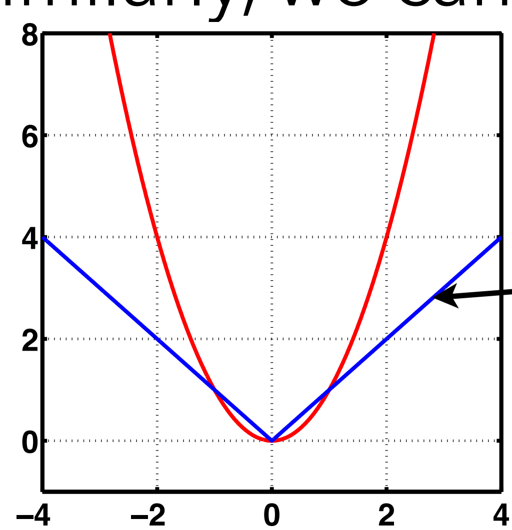
Large penalty / low probability for outliers



Probability density

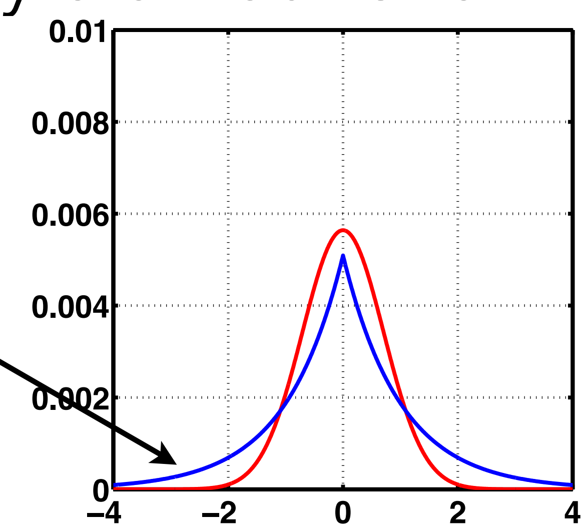
Robust Likelihood

- Because of such effects, the Gaussian assumption is likely to be inappropriate.
  - We can use a **robust** likelihood model that is more robust to such errors in the modeling assumption.
    - Remember: We assumed that the brightness of an image feature is the same in both cameras...
    - We discussed robust error functions to define costs.
  - Similarly, we can use (more) robust probability distributions.



Cost or energy

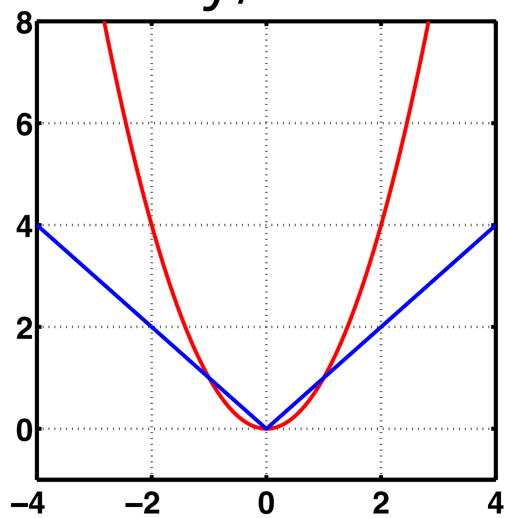
Smaller penalty / higher probability for outliers



Probability density

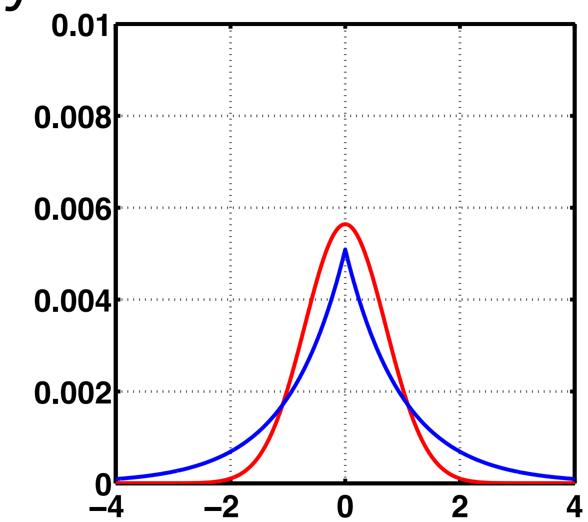
Robust Likelihood

- Because of such effects, the Gaussian assumption is likely to be inappropriate.
  - We can use a **robust** likelihood model that is more robust to such errors in the modeling assumption.
    - Remember: We assumed that the brightness of an image feature is the same in both cameras...
    - We discussed robust error functions to define costs.
  - Similarly, we can use (more) robust probability distributions.



Cost or energy

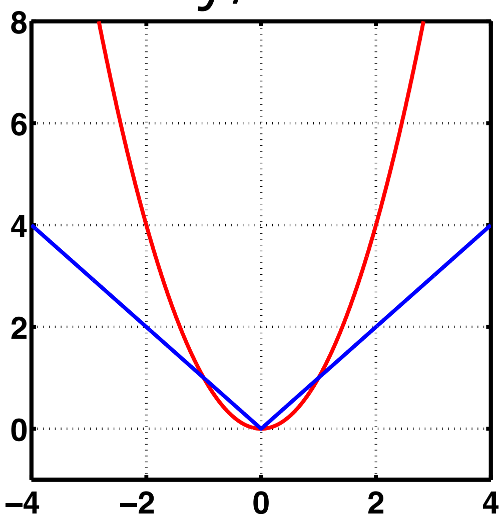
$$p(x) = \frac{1}{\hat{Z}} e^{-\hat{E}(x)} = \frac{1}{\hat{Z}} e^{-|x|}$$



Probability density

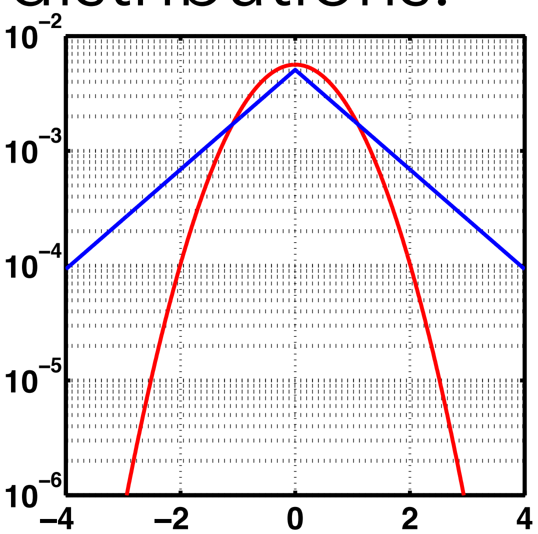
Robust Likelihood

- Because of such effects, the Gaussian assumption is likely to be inappropriate.
  - We can use a **robust** likelihood model that is more robust to such errors in the modeling assumption.
    - Remember: We assumed that the brightness of an image feature is the same in both cameras...
    - We discussed robust error functions to define costs.
  - Similarly, we can use (more) robust probability distributions.



Cost or energy

$$p(x) = \frac{1}{\hat{Z}} e^{-\hat{E}(x)} = \frac{1}{\hat{Z}} e^{-|x|}$$

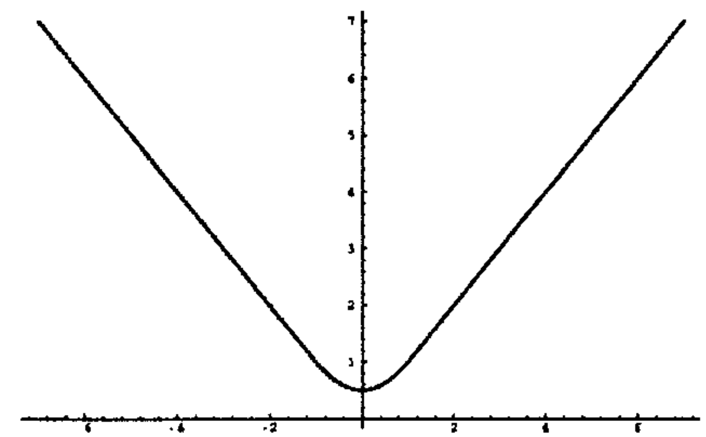


Probability density

Robust Penalizers

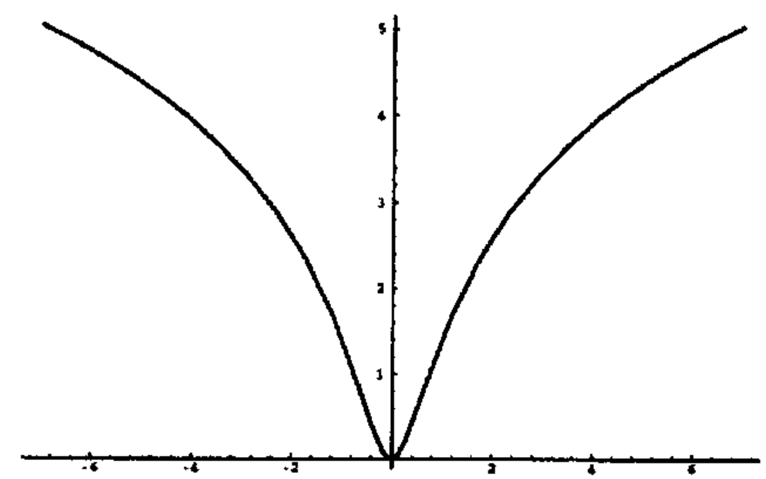
- There is a whole zoo of other robust penalty functions (interpreted as cost or energy), e.g.:
  - Huber's min-max estimator

$$\rho(x;\epsilon) = \begin{cases} x^2/2\epsilon + \epsilon/2, & |x| \leq \epsilon \\ |x|, & |x| > \epsilon \end{cases}$$



  - Lorentzian

$$\rho(x;\sigma) = \log\left(1 + \frac{x^2}{2\sigma^2}\right)$$



    - equivalent to Student's t-distribution

$$p(x|\sigma,\alpha) \propto \left(1 + \frac{x^2}{2\sigma^2}\right)^{-\alpha}$$

Modeling the Prior

TECHNISCHE UNIVERSITÄT DARMSTADT

- Again: The prior $p(x)$ models our a-priori assumptions about the world, or the state of the world.
  - In stereo the prior models how probable it is to have a certain disparity map
    - ... in the absence of any other information.
  - We wanted to model that nearby pixels have similar disparities.
  - But we also need to allow for depth / disparity discontinuities.
  - How do we formalize such a prior probability mathematically?

What is Consistency?

- Before we can do anything, we need to ask ourselves what it means to have spatial consistency or regularity.
- Let’s look at some data to get inspiration:

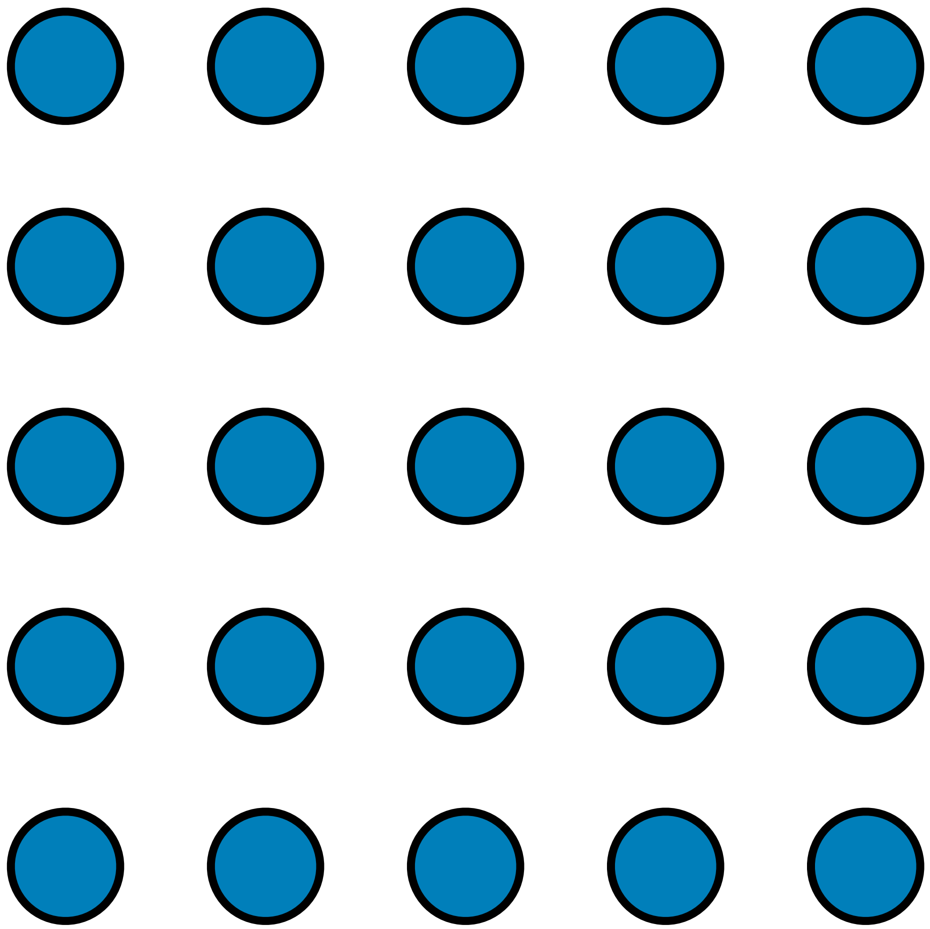
Range image - Scene depth from a range scanner

What can we conclude from range images?

- They help us see more clearly what we know from everyday life:
  - The depth of nearby points in the scene is (almost) the same.
  - But sometimes, there are depth discontinuities, for example at object boundaries.
- How do we model this?

Modeling Compatibilities

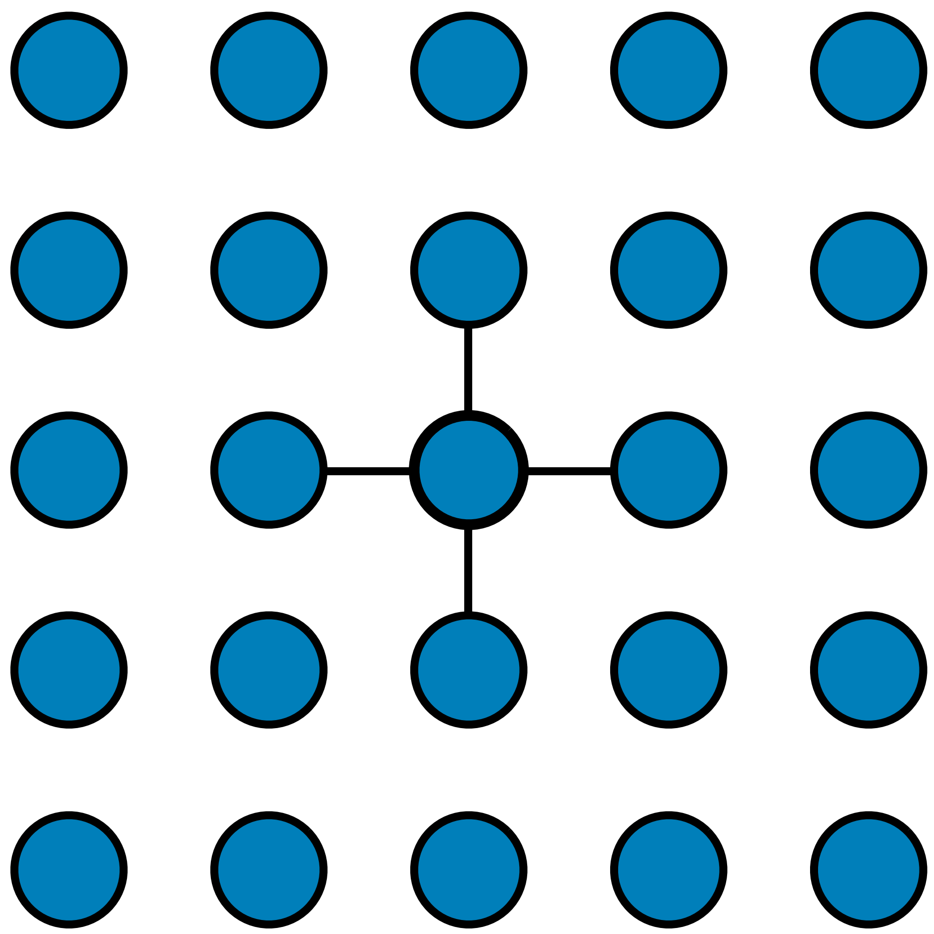
- Pixel grid:



Let's assume that we want to model how compatible or consistent a pixel is with its 4 nearest neighbors.

Modeling Compatibilities

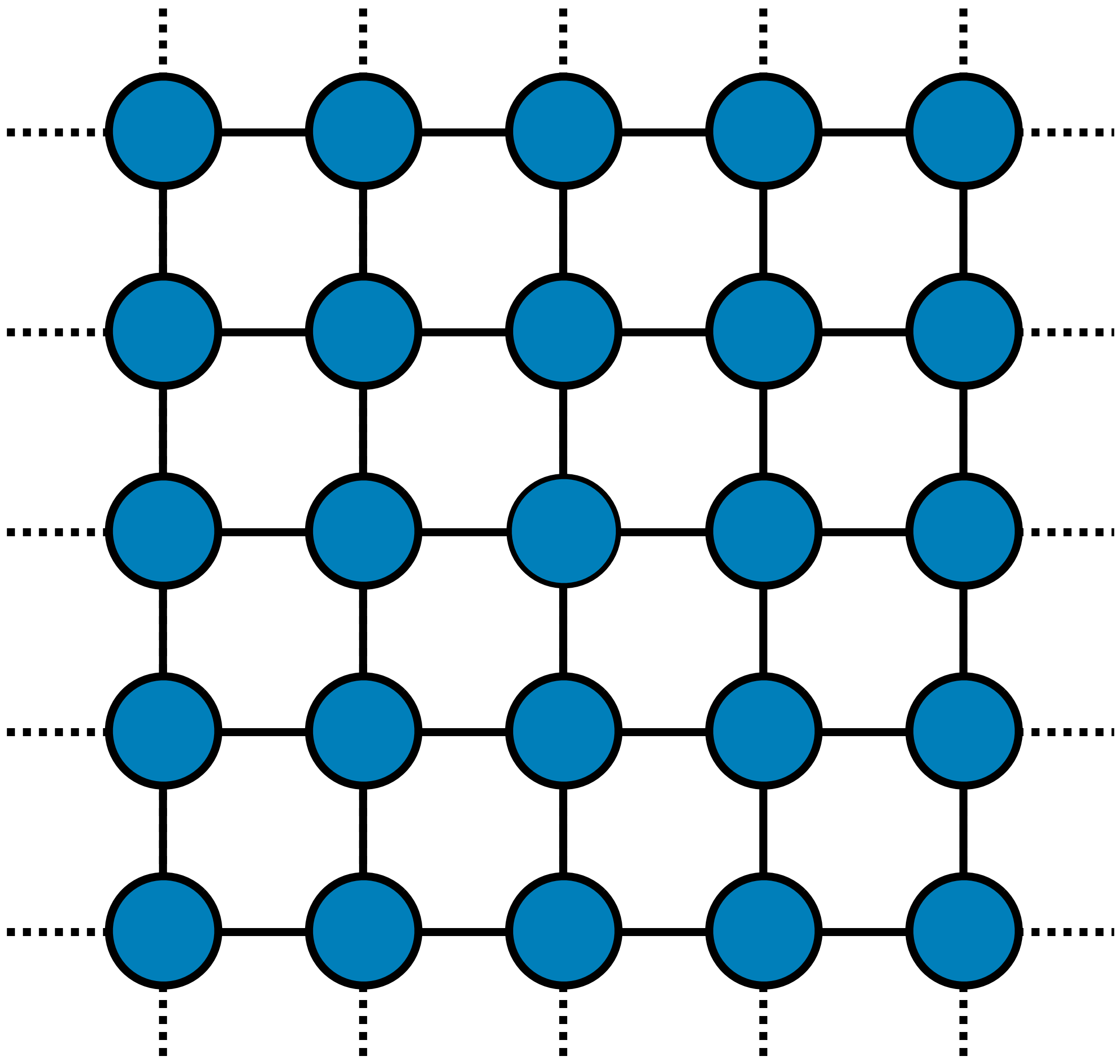
- Pixel grid (as nodes of a graph):



Denote this by drawing a line (edge) between two pixels (nodes).

Modeling Compatibilities

- Pixel grid (as nodes of a graph):

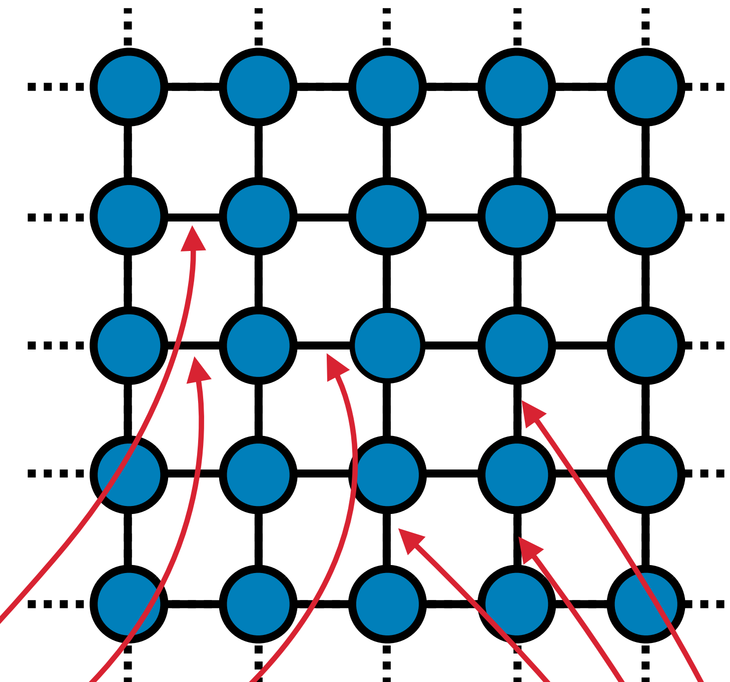


We do this for all pixels.

Markov Random Fields

TECHNISCHE
UNIVERSITÄT
DARMSTADT

- This is a particular instance of a so-called **graphical model**, or more specifically a so called **Markov random field (MRF)**.
  - We will get to the more general case soon, but let's preview what this means mathematically.
  - Each edge (in this particular graph) corresponds to a term in the prior that models how compatible two neighboring pixels are in terms of their disparity:



compatibility of horizontal neighbors

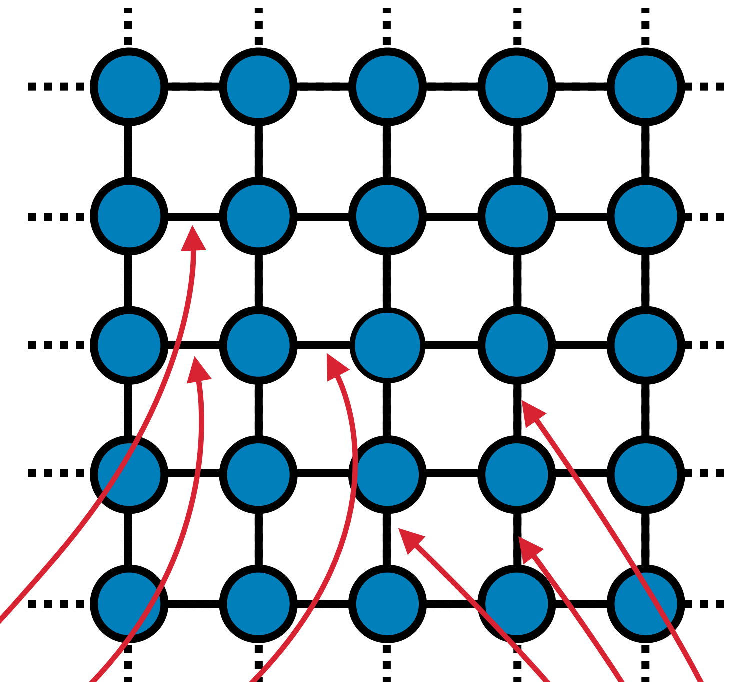
compatibility of vertical neighbors

$$p(\mathbf{d}) = \prod_{i,j} f_H(d_{i,j}, d_{i+1,j}) \cdot f_V(d_{i,j}, d_{i,j+1})$$

product over all the pixels

Markov Random Fields

- MRF prior:
  - Each edge (in this particular graph) corresponds to a term that models how compatible two neighboring pixels are in terms of their disparity.
  - Typically, formulate in terms of the disparity difference



compatibility of horizontal neighbors

compatibility of vertical neighbors

$$p(\mathbf{d}) = \underbrace{\prod_{i,j}} \underbrace{f_H(d_{i,j} - d_{i+1,j})} \cdot \underbrace{f_V(d_{i,j} - d_{i,j+1})}$$

product over all the pixels

Potts Model

TECHNISCHE UNIVERSITÄT DARMSTADT

- Define very simple compatibility functions:

$$f_H(d_{i,j} - d_{i+1,j}) = \frac{1}{Z(T)} \exp\left\{ \frac{1}{T} \delta(d_{i,j} - d_{i+1,j}) \right\}$$

- Kronecker delta:

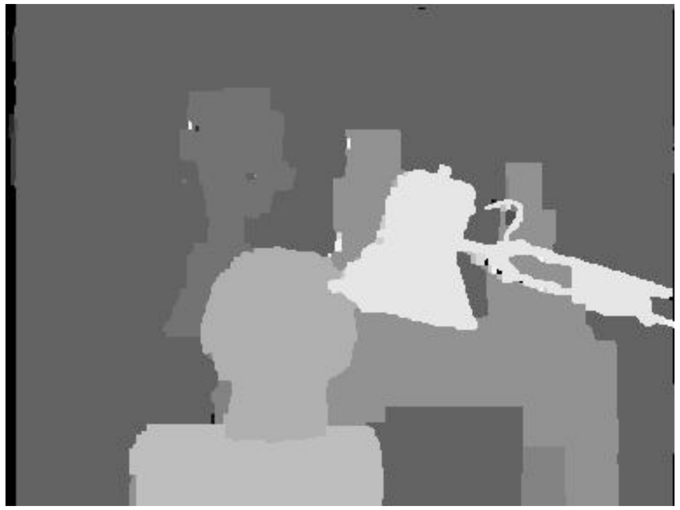
$$\delta(a - b) = \begin{cases} 1, & a = b \\ 0, & a \neq b \end{cases}$$

- This prior:
  - Prefers to have the same disparities at neighboring pixels.
  - But allows for disparity discontinuities with no penalty for large discontinuities.
  - Is called a **Potts model**.
    - Originally from statistical physics (magnetism)

Stereo with Markov Random Fields

- We are now ready to define a probabilistic model for stereo:
  - Define an observation model, for example using the Gaussian likelihood we discussed.
  - Define a simple prior that enforces our intuitive prior knowledge about disparities / scene depth. This can, for example, be done using a Potts model.
  - Then we can do stereo reconstruction by doing inference with this model (more next class).

Example result
(from Tappen & Freeman):



Stereo with Markov Random Fields

TECHNISCHE UNIVERSITÄT DARMSTADT

- Summary so far:
  - We have defined the stereo problem using probabilistic models.
  - We were able to integrate prior knowledge about the disparity maps using a Markov random field based prior.
  - We can solve for the disparity map using probabilistic inference.

- Advantages over window-based matching approach:
  - Clearly much better results.
  - The prior allows us to have a “window” size of 1 for matching.
  - But of course, there a problems... hold on.

Readings

- Intro to probability: Prince Ch. 2
- Graphical models: Prince Ch. 10.1-10.5
- For next time: Prince. Ch. 12.1, 12.2, 12.4