

# Robot Mapping

## Robust Least Squares for SLAM

**Gian Diego Tipaldi, Wolfram Burgard**

---

Courtesy for most images: Pratik Agarwal

# Least Squares in General

- Minimizes the **sum of the squared errors**
- ML estimation in the Gaussian case

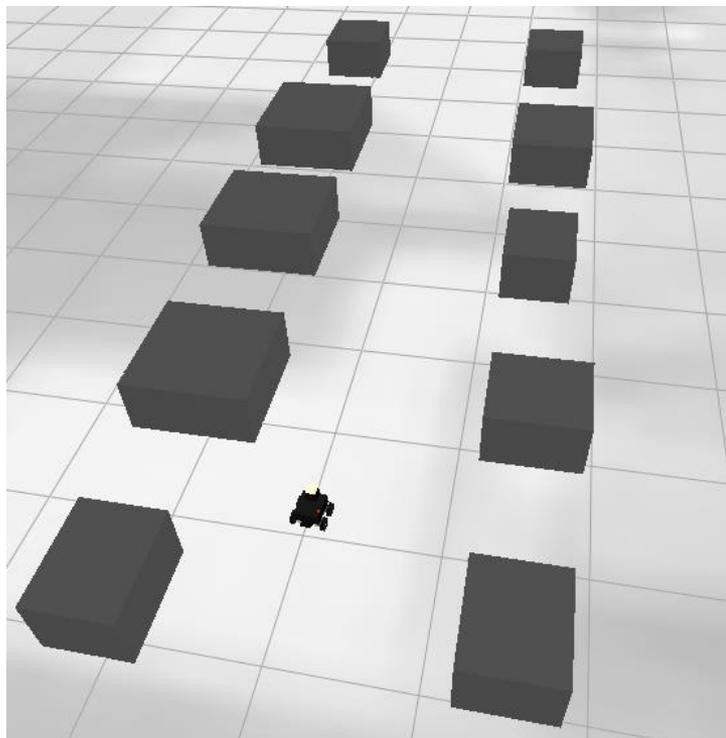
## Problems:

- **Sensitive to outliers**
- **Only Gaussians** (single modes)

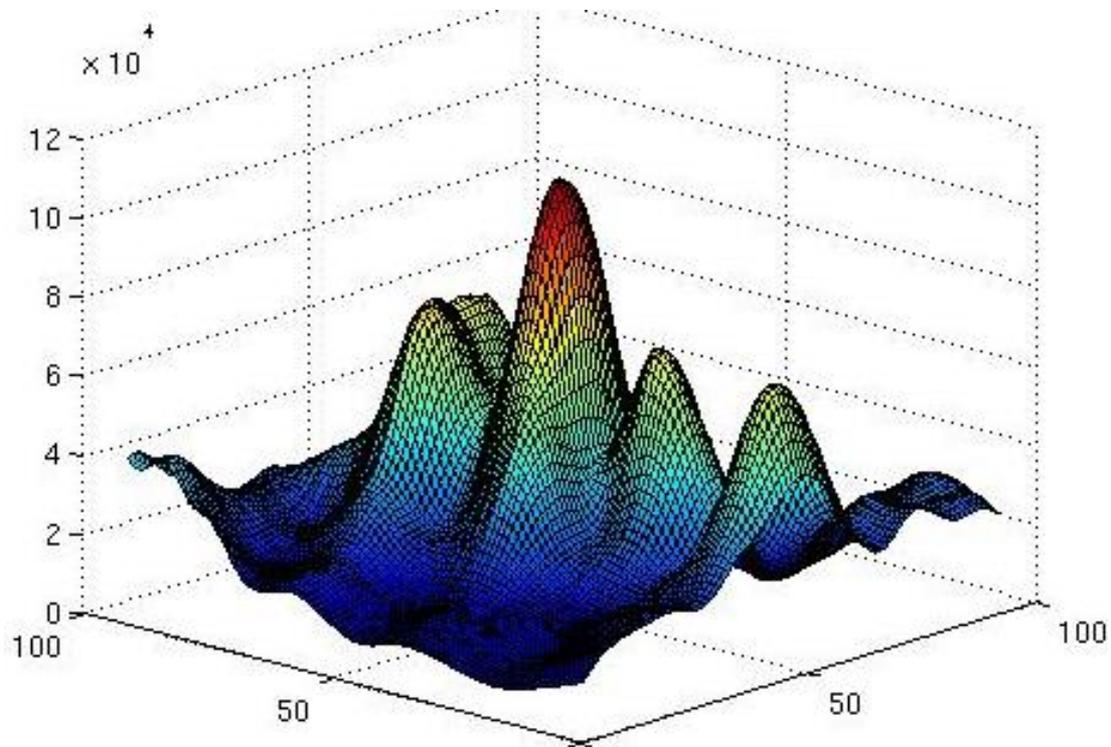
# Data Association Is Ambiguous And Not Always Perfect

- Places that look identical
- Similar rooms in the same building
- Cluttered scenes
- GPS multi pass (signal reflections)
- ...

# Example

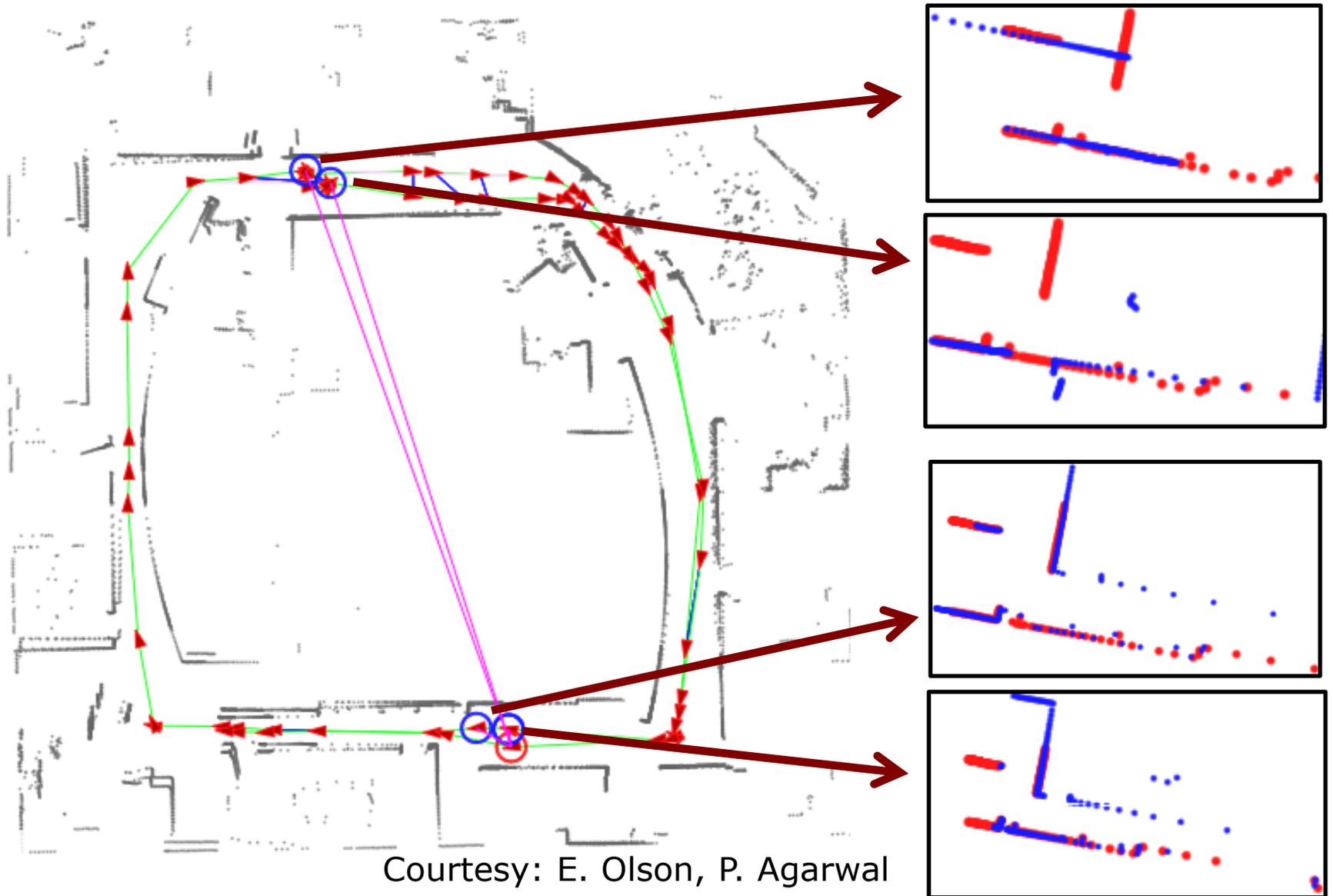


3D world

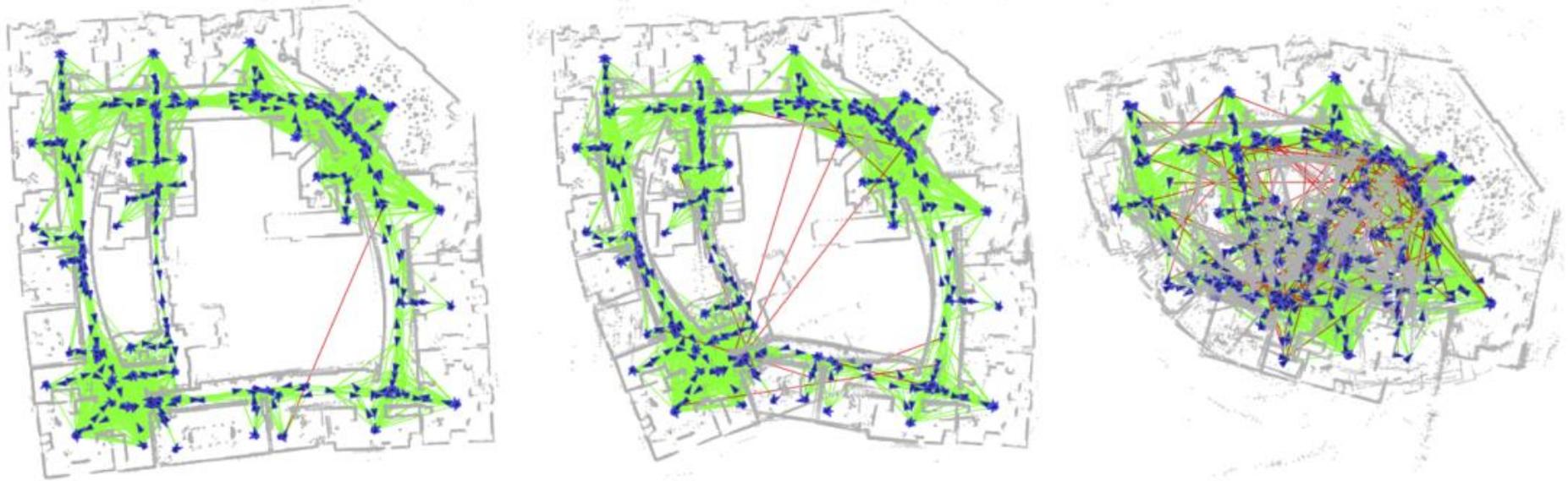


belief about the  
robot's pose

# Such Situations Occur In Reality



# Committing To The Wrong Mode Can Lead to Mapping Failures



# Data Association Is Ambiguous And Not Always Perfect

- Places that look identical
- Similar rooms in the same building
- Cluttered scenes
- GPS multi pass (signal reflections)
- ...

**How to incorporate that  
into graph-based SLAM?**

# Data Association Is Ambiguous And Not Always Perfect

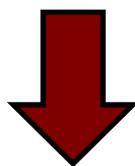
- Places that look identical
- Similar rooms in the same building
- Cluttered scenes
- GPS multi pass (signal reflections)
- ...

**How to incorporate that  
into graph-based SLAM?**

# Mathematical Model

- We can express a multi-modal belief by a sum of Gaussians

$$p(\mathbf{z} \mid \mathbf{x}) = \eta \exp\left(-\frac{1}{2} \mathbf{e}_{ij}^T \boldsymbol{\Omega}_{ij} \mathbf{e}_{ij}\right)$$



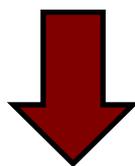
$$p(\mathbf{z} \mid \mathbf{x}) = \sum_k w_k \eta_k \exp\left(-\frac{1}{2} \mathbf{e}_{ijk}^T \boldsymbol{\Omega}_{ijk} \mathbf{e}_{ijk}\right)$$

**Sum of Gaussians with k modes**

# Problem

- During error minimization, we consider the negative log likelihood

$$-\log p(\mathbf{z} | \mathbf{x}) = \frac{1}{2} \mathbf{e}_{ij}^T \Omega_{ij} \mathbf{e}_{ij} - \log \eta$$



$$-\log p(\mathbf{z} | \mathbf{x}) = -\log \sum_k w_k \eta_k \exp\left(-\frac{1}{2} \mathbf{e}_{ijk}^T \Omega_{ijk} \mathbf{e}_{ijk}\right)$$

**The log cannot be moved inside the sum!**

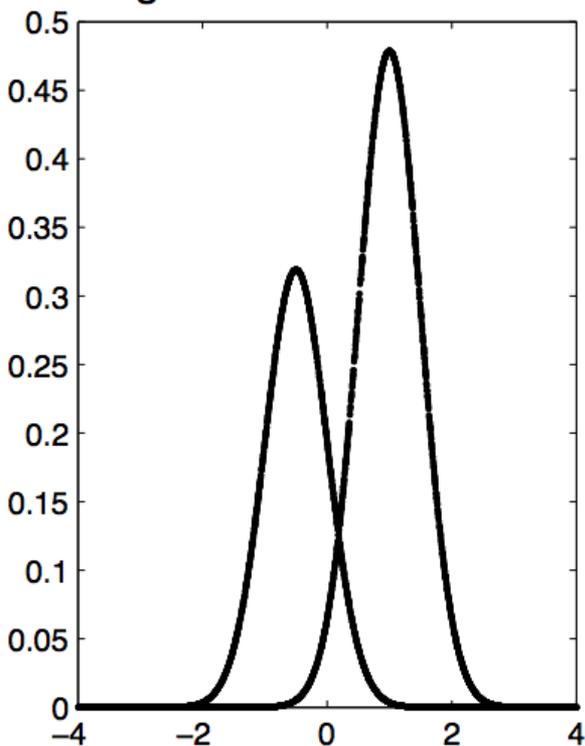
# Max-Mixture Approximation

- Instead of computing the sum of Gaussians at  $\mathbf{X}$ , compute the maximum of the Gaussians

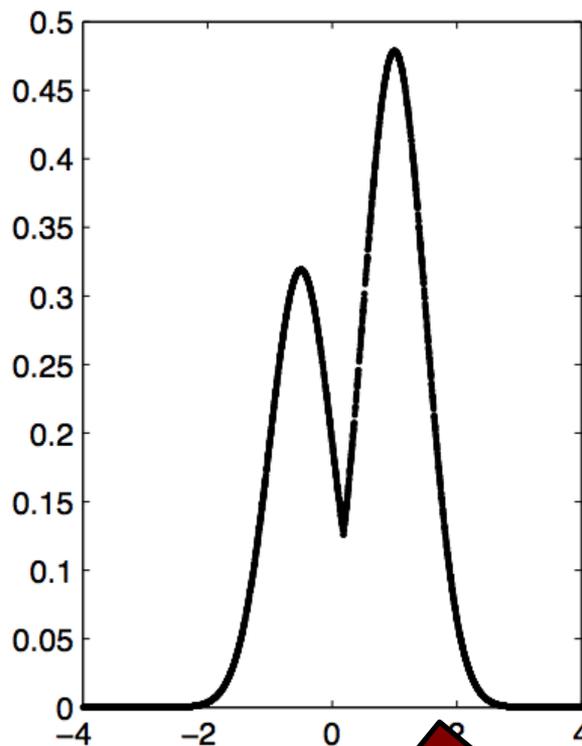
$$\begin{aligned} p(\mathbf{z} \mid \mathbf{x}) &= \sum_k w_k \eta_k \exp\left(-\frac{1}{2} \mathbf{e}_{ijk}^T \boldsymbol{\Omega}_{ijk} \mathbf{e}_{ijk}\right) \\ &\simeq \max_k w_k \eta_k \exp\left(-\frac{1}{2} \mathbf{e}_{ijk}^T \boldsymbol{\Omega}_{ijk} \mathbf{e}_{ijk}\right) \end{aligned}$$

# Max-Mixture Approximation

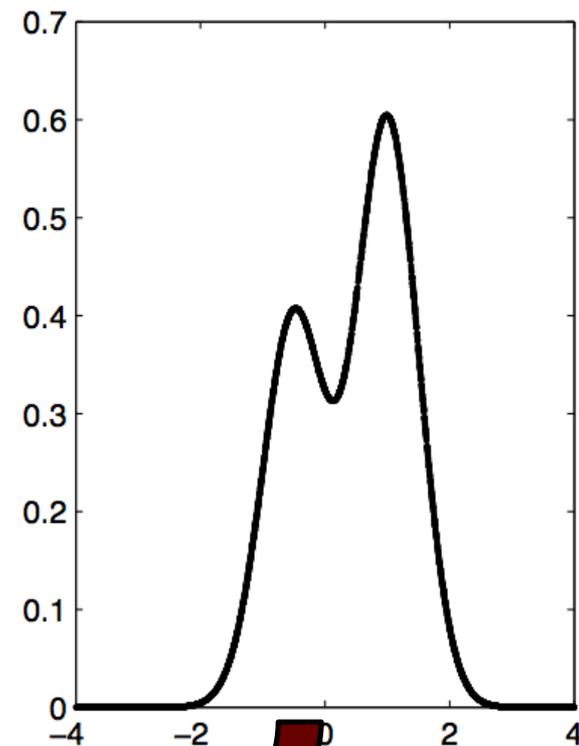
Original bi-modal mixture



Max-mixture



Sum-mixture

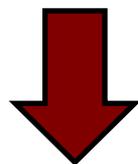


**approximation error**

# Log Likelihood Of The Max-Mixture Formulation

- The log can be moved inside the max operator

$$p(\mathbf{z} | \mathbf{x}) \simeq \max_k w_k \eta_k \exp\left(-\frac{1}{2} \mathbf{e}_{ijk}^T \boldsymbol{\Omega}_{ijk} \mathbf{e}_{ijk}\right)$$



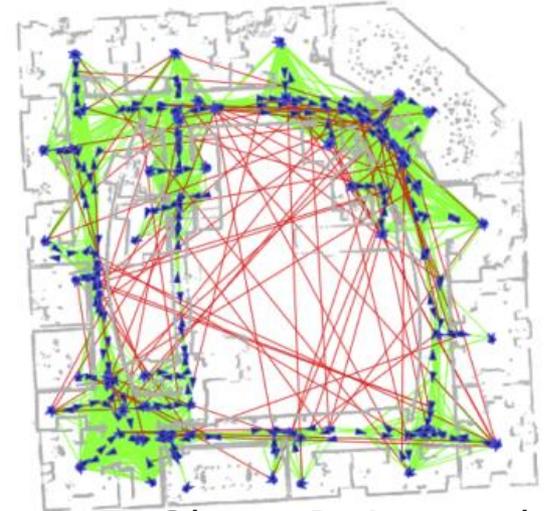
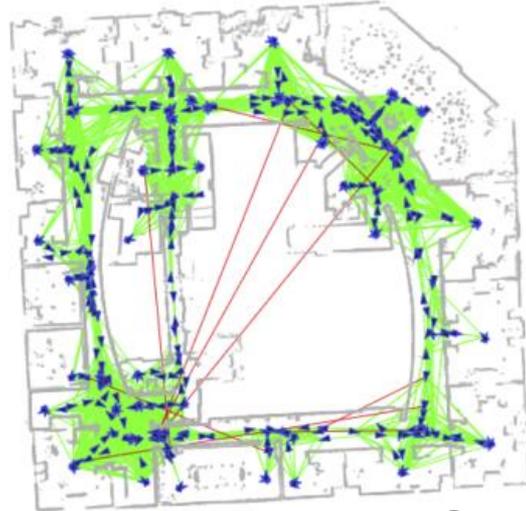
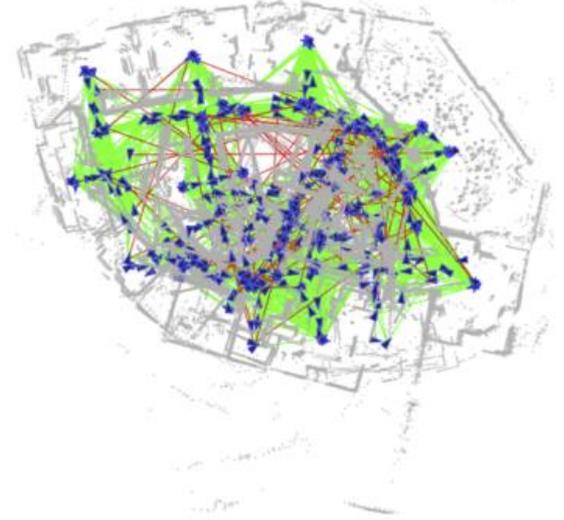
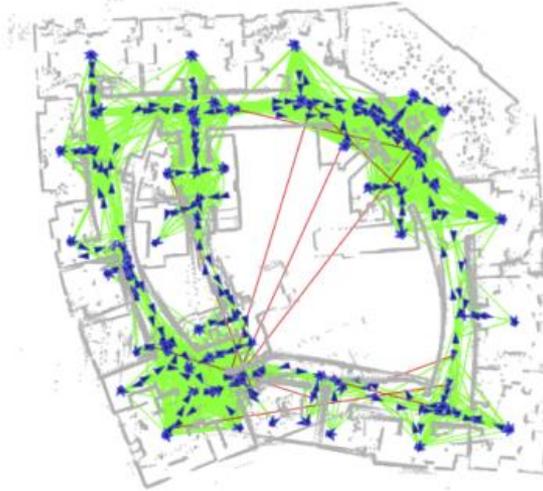
$$\log p(\mathbf{z} | \mathbf{x}) \simeq \max_k -\frac{1}{2} \mathbf{e}_{ijk}^T \boldsymbol{\Omega}_{ijk} \mathbf{e}_{ijk} + \log(w_k \eta_k)$$

$$\text{or: } -\log p(\mathbf{z} | \mathbf{x}) \simeq \min_k \frac{1}{2} \mathbf{e}_{ijk}^T \boldsymbol{\Omega}_{ijk} \mathbf{e}_{ijk} - \log(w_k \eta_k)$$

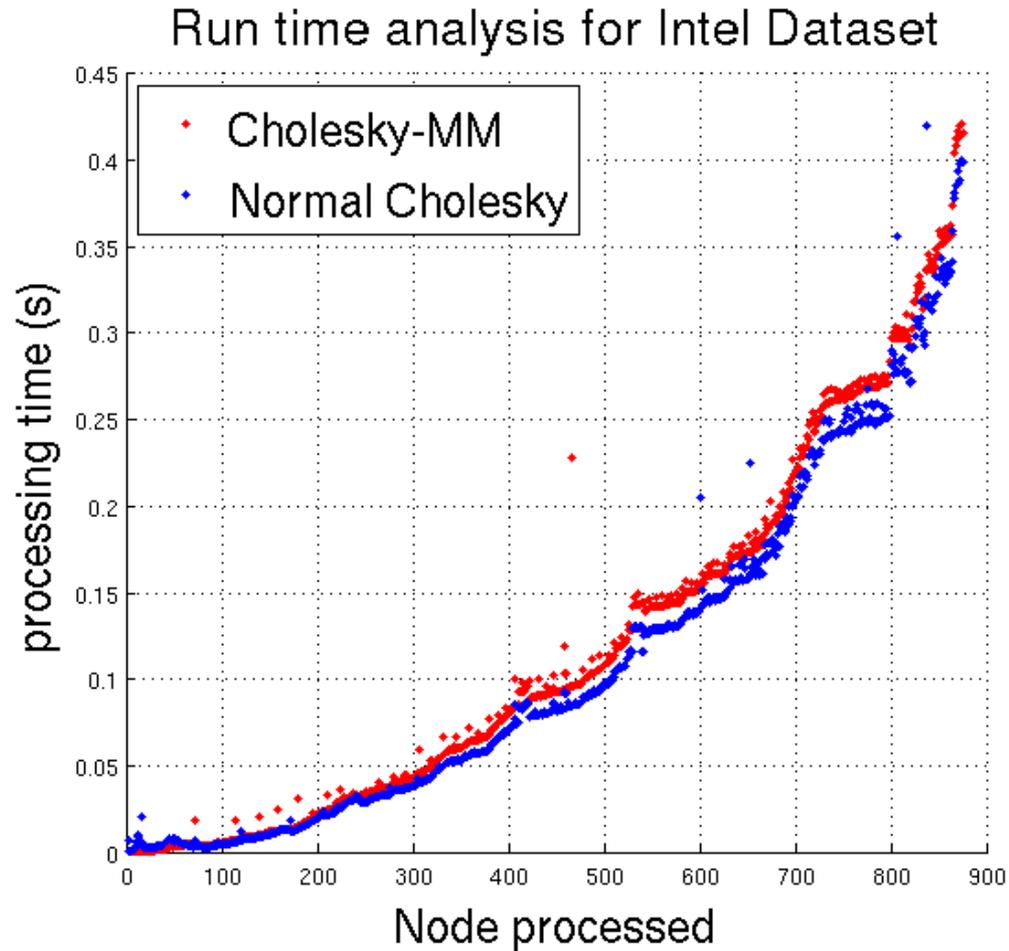
# Integration

- With the max-mixture formulation, the log likelihood again results in local quadratic forms
- Easy to integrate in the optimizer:
  1. Evaluate all  $k$  components
  2. Select the component with the maximum log likelihood
  3. Perform the optimization as before using only the max components (as a single Gaussian)

# Performance (Gauss vs. MM)



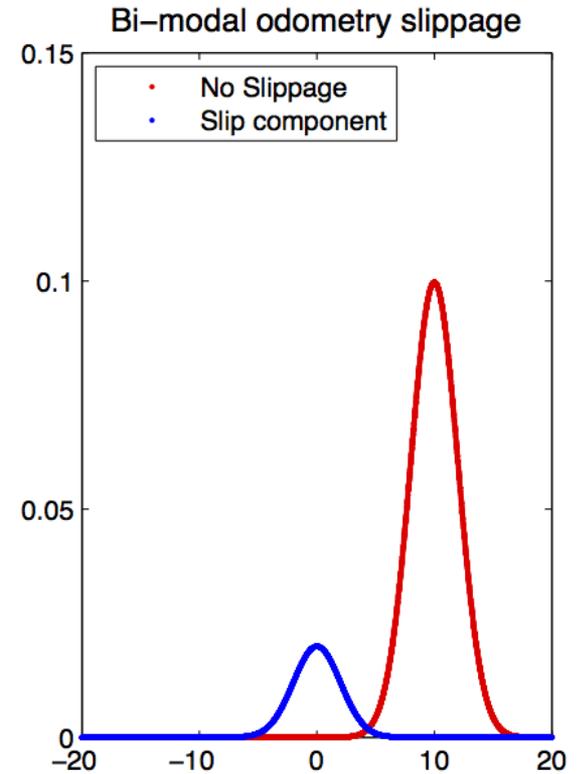
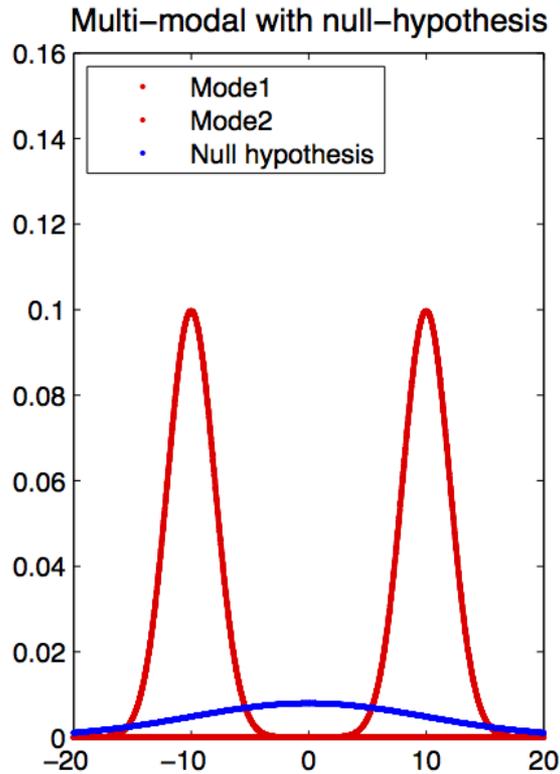
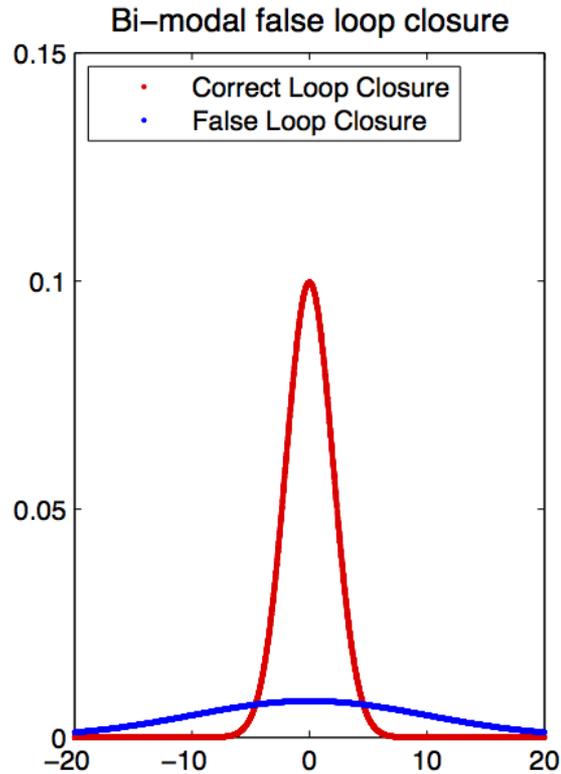
# Runtime



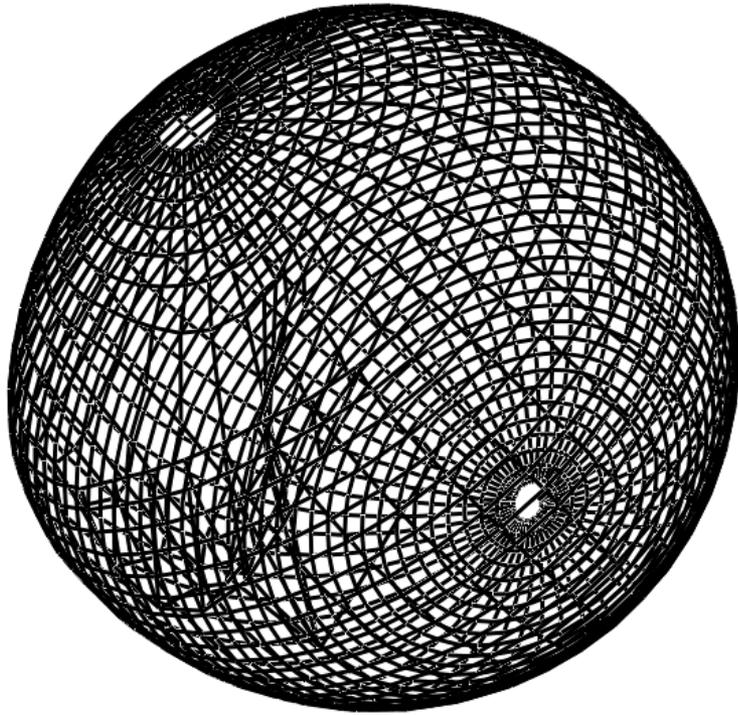
# Max-Mixture and Outliers

- MM formulation is useful for multi-modal measurements (D.A. ambiguities)
- MM is also a handy tool for outliers (D.A. failures)
- Here, one mode represents the edge and a second model uses a flat Gaussian for the outlier hypothesis

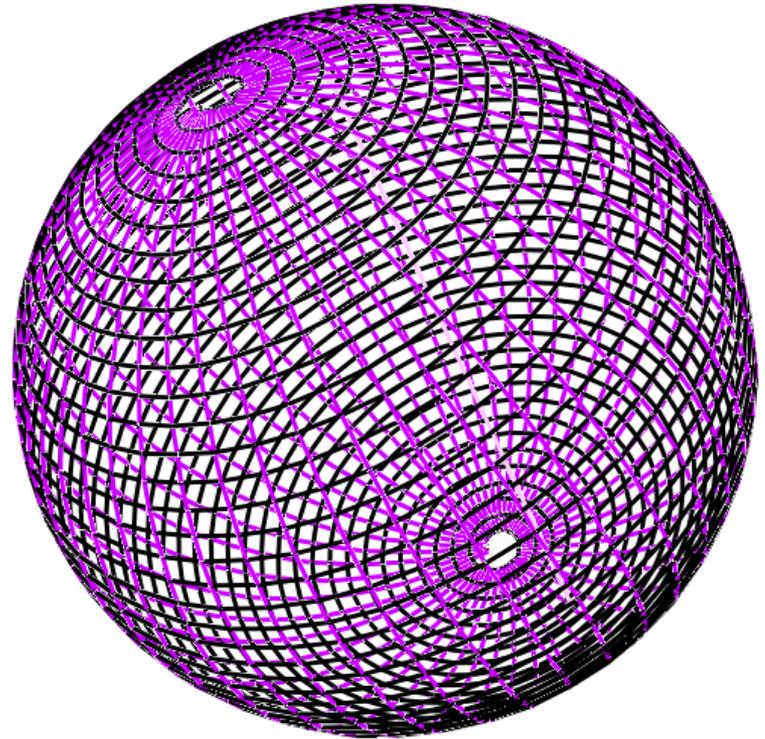
# Max-Mixture and Outliers



# Performance (1 outlier)

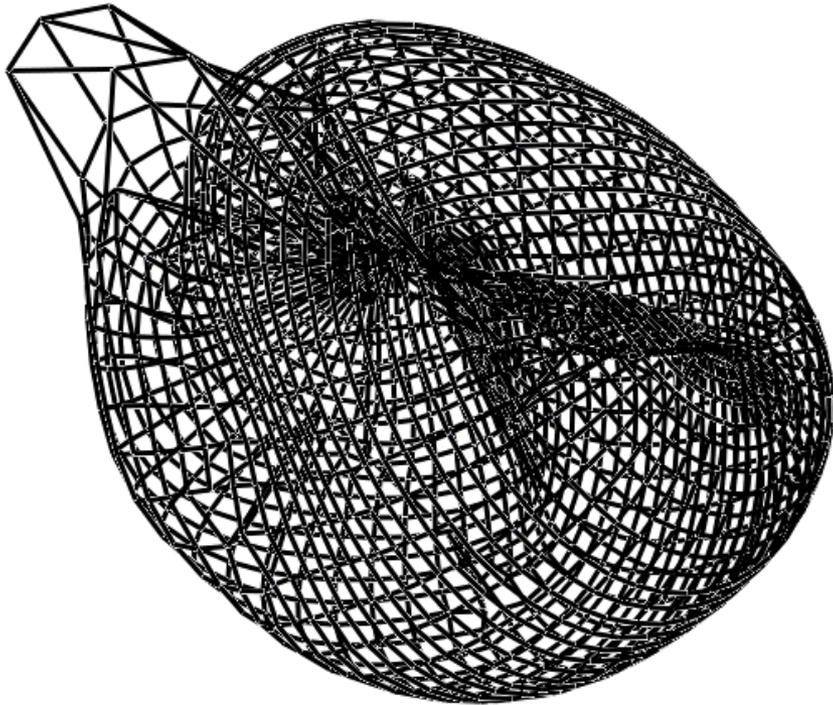


Gauss-Newton

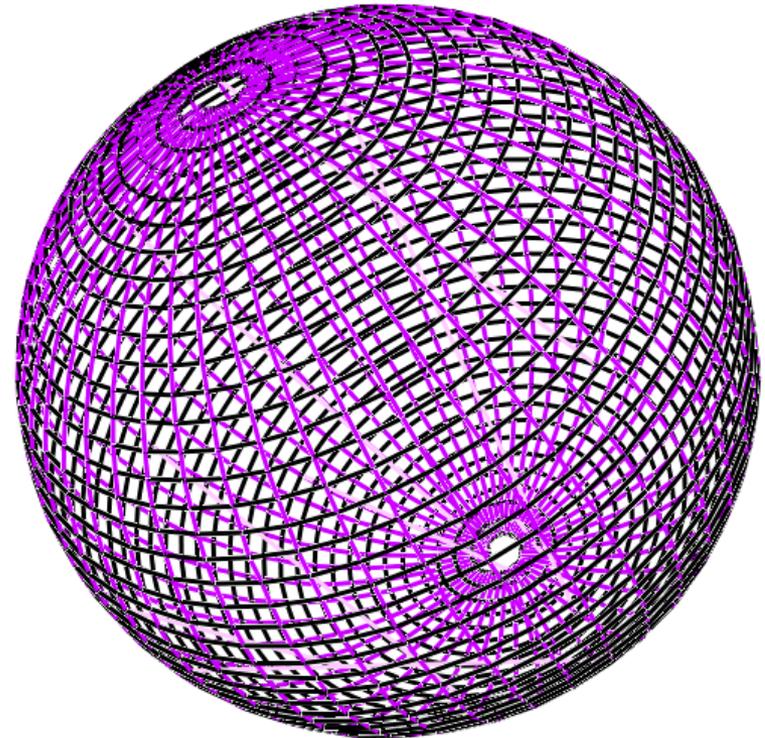


MM Gauss-Newton

# Performance (10 outliers)

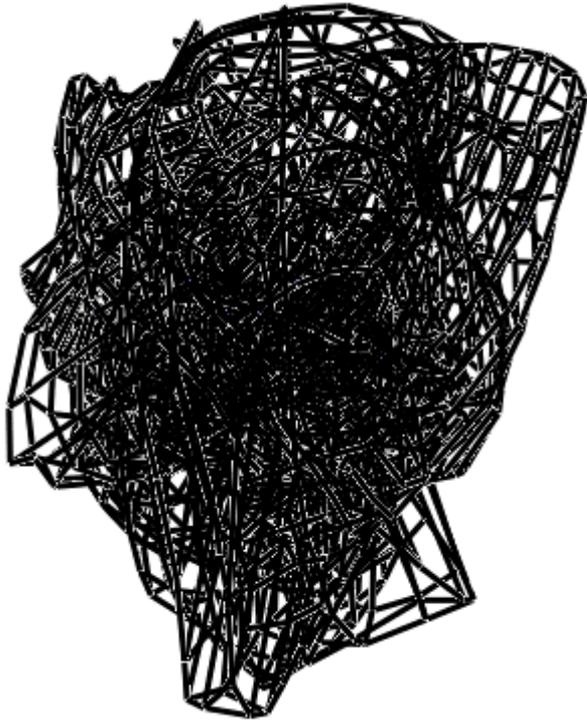


Gauss-Newton

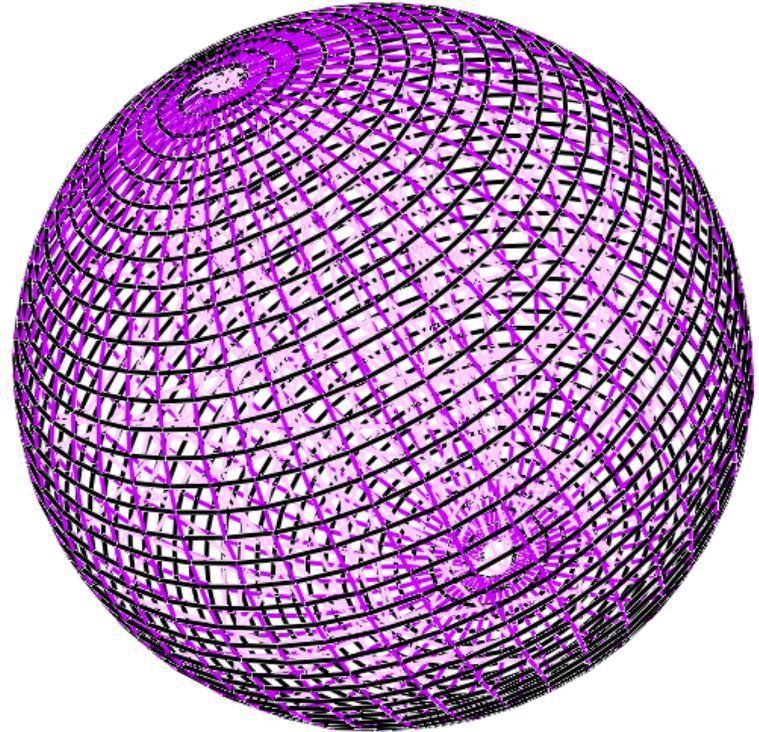


MM Gauss-Newton

# Performance (100 outliers)



Gauss-Newton



MM Gauss-Newton

# Standard Gaussian Least Squares

$$X^* = \operatorname{argmin}_X \sum_{ij} \underbrace{\mathbf{e}_{ij}(X)^T \Omega_{ij} \mathbf{e}_{ij}(X)}_{\chi_{ij}^2}$$

# Dynamic Covariance Scaling

$$X^* = \operatorname{argmin}_X \sum_{ij} \underbrace{\mathbf{e}_{ij}(X)^T \Omega_{ij} \mathbf{e}_{ij}(X)}_{\chi_{ij}^2}$$

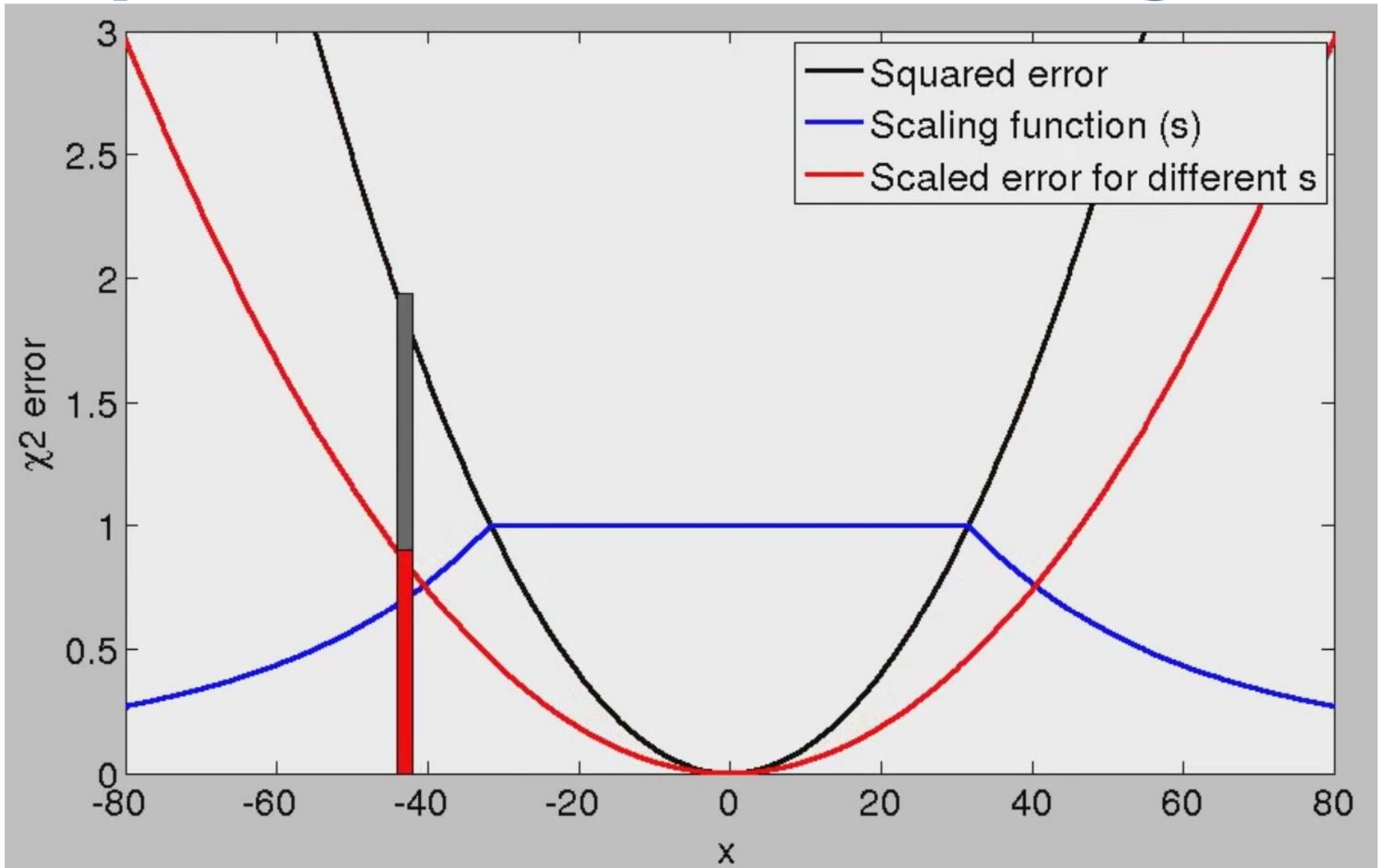
$$X^* = \operatorname{argmin}_X \sum_{ij} \mathbf{e}_{ij}(X)^T (s_{ij}^2 \Omega_{ij}) \mathbf{e}_{ij}(X)$$


# Scaling Parameter

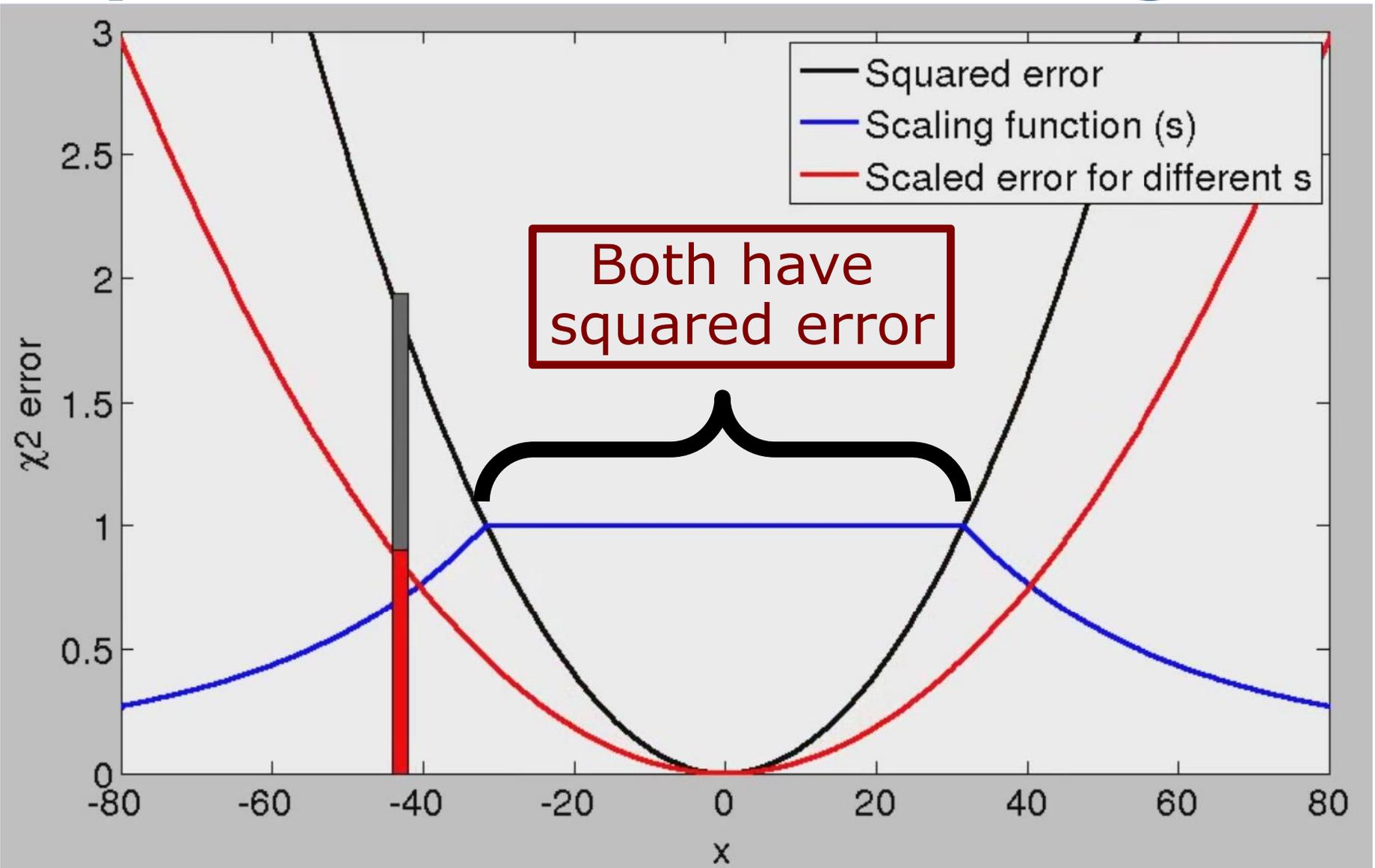
$$X^* = \operatorname{argmin}_X \sum_{ij} \mathbf{e}_{ij}(X)^T \left( s_{ij}^2 \Omega_{ij} \right) \mathbf{e}_{ij}(X)$$


$$s_{ij} = \min \left( 1, \frac{2\Phi}{\Phi + \chi_{ij}^2} \right)$$

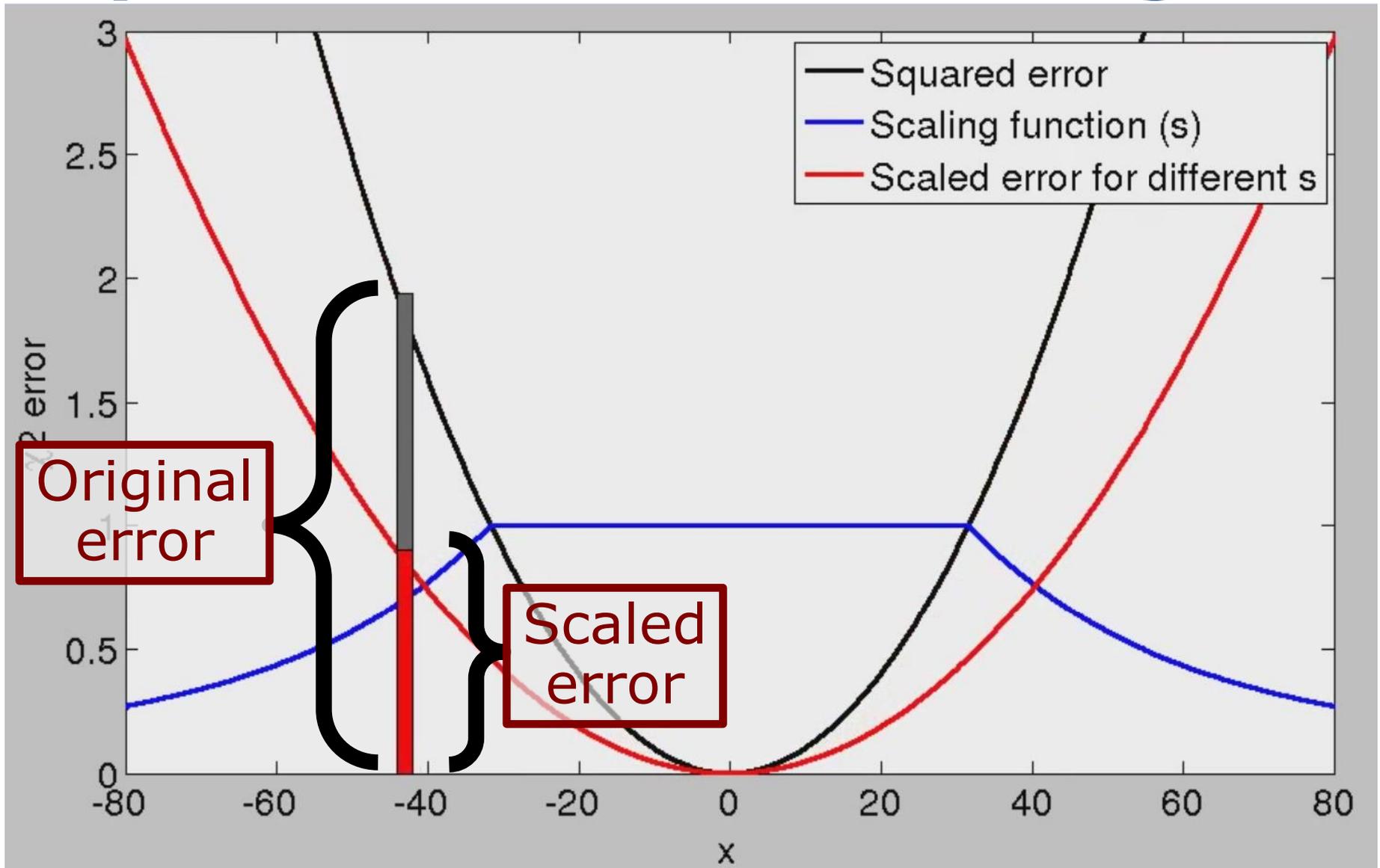
# Dynamic Covariance Scaling



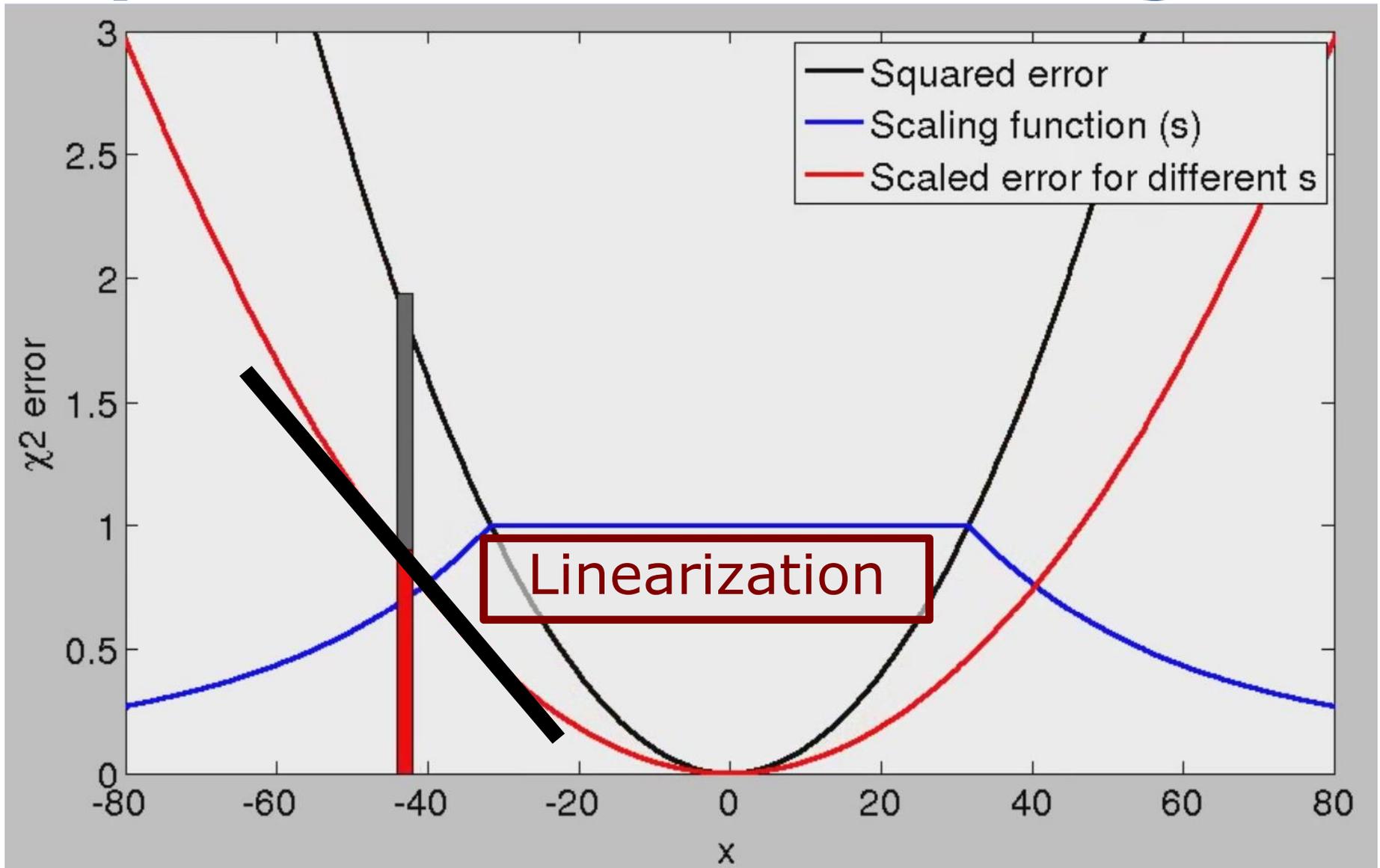
# Dynamic Covariance Scaling



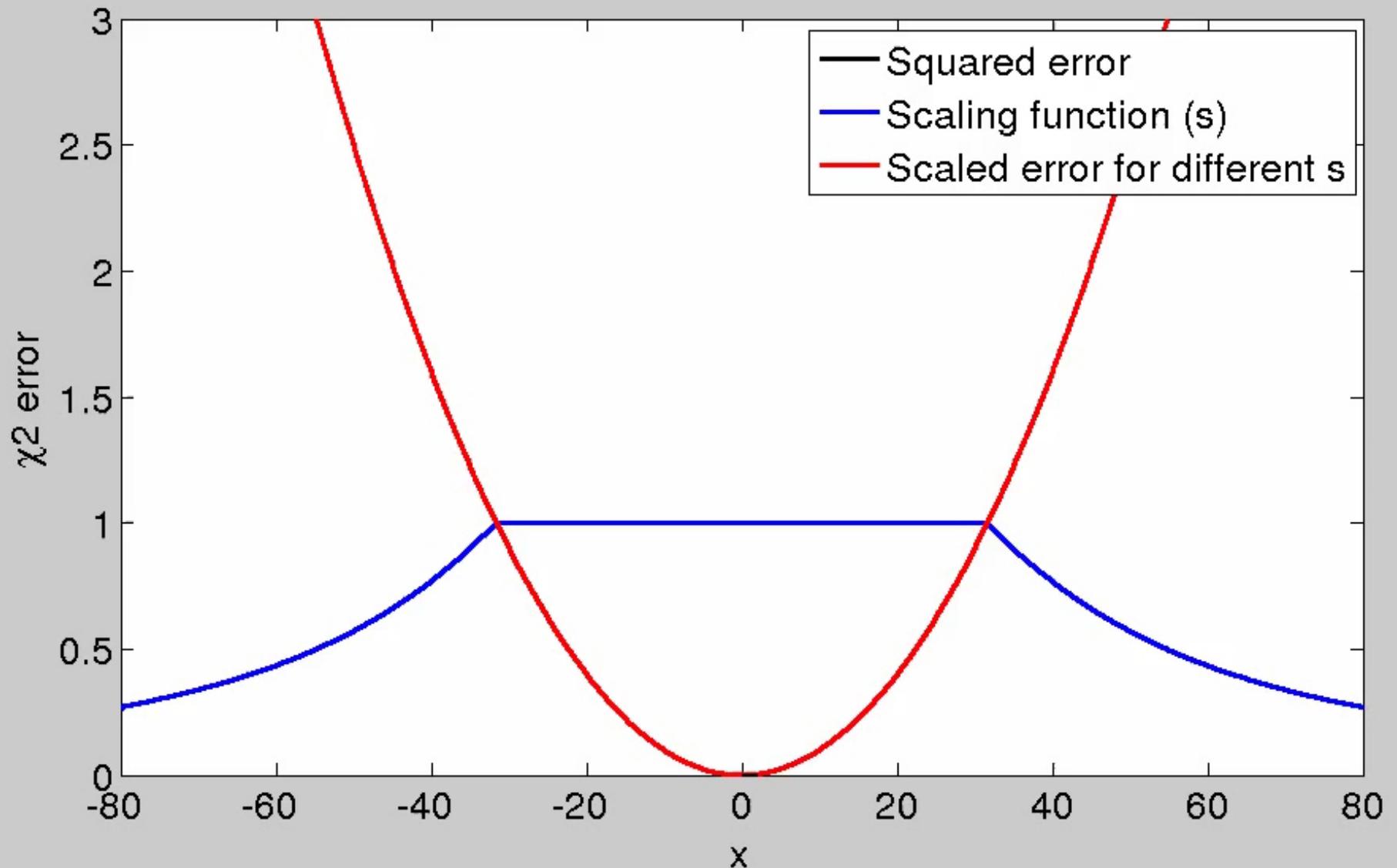
# Dynamic Covariance Scaling



# Dynamic Covariance Scaling

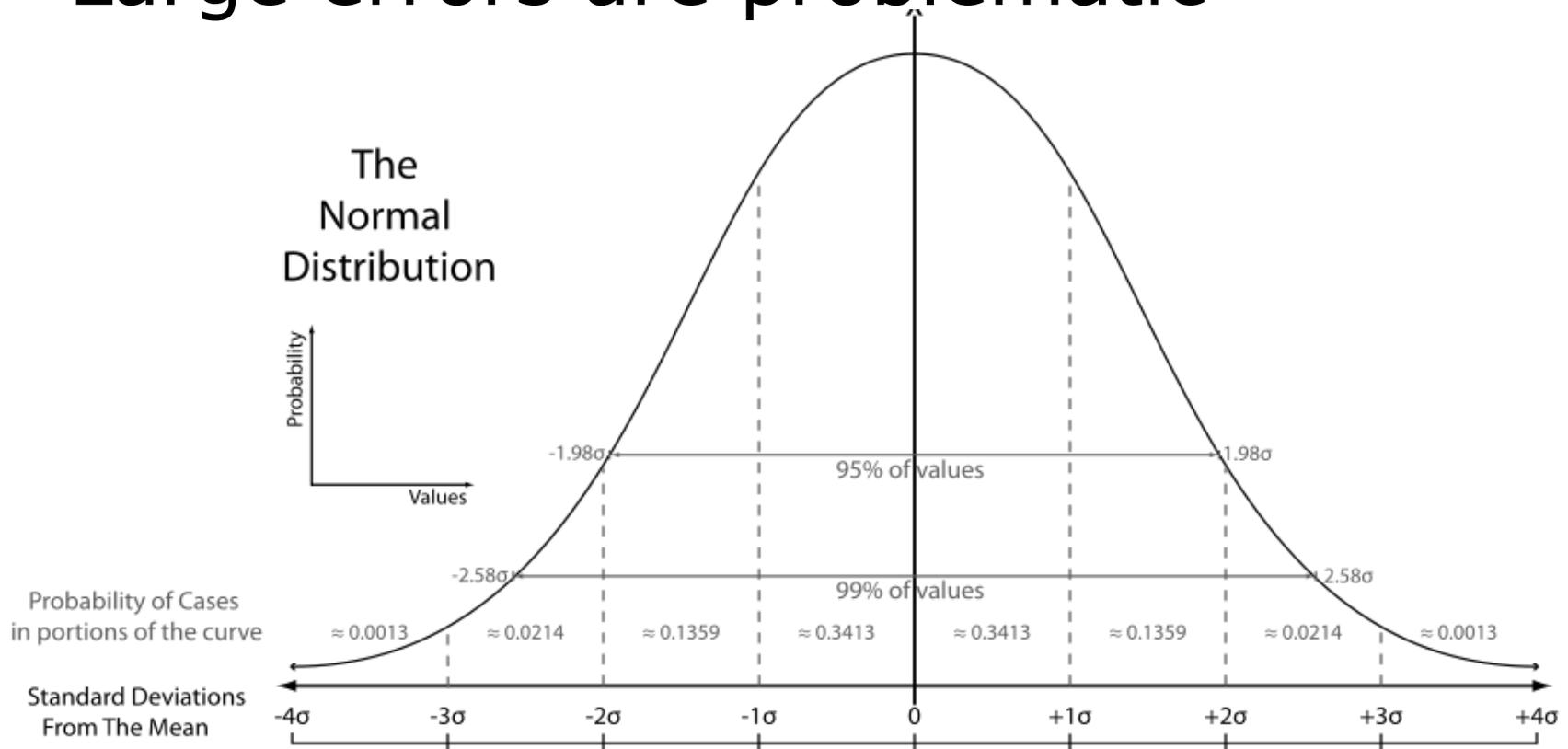


# Dynamic Covariance Scaling



# Optimizing With Outliers

- Assuming a Gaussian error in the measurement is not always realistic
- Large errors are problematic



# Robust M-Estimators

- Assume non-normally-distributed noise
- Intuitively: PDF with “heavy tails”
- $\rho(e)$  function used to define the PDF

$$p(e) = \exp(-\rho(e))$$

- Minimizing the neg. log likelihood

$$\mathbf{x}^* = \underset{\mathbf{x}}{\operatorname{argmin}} \sum_i \rho(e_i(\mathbf{x}))$$

# Different Rho Functions

- Gaussian:  $\rho(e) = e^2$
- Absolute values (L1 norm):  $\rho(e) = |e|$
- Huber M-estimator

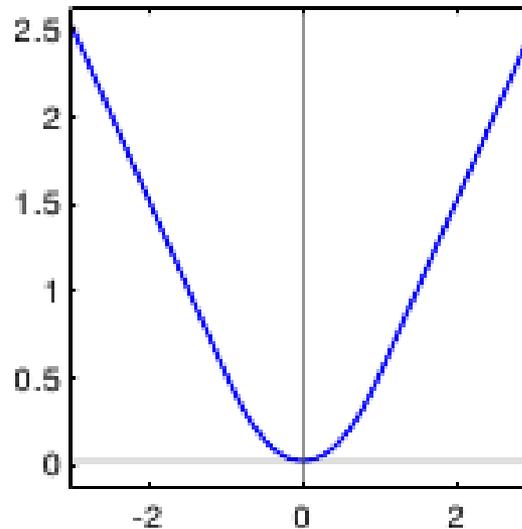
$$\rho(e) = \begin{cases} \frac{e^2}{2} & \text{if } |e| < c \\ c(|e| - \frac{c}{2}) & \text{otherwise} \end{cases}$$

- Several others (Tukey, Cauchy, Blake-Zisserman, Corrupted Gaussian, ...)

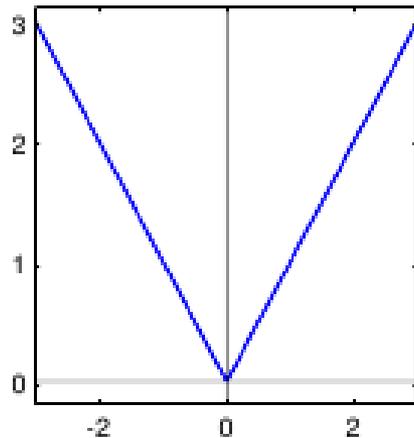
# Huber

- Mixture of a quadratic and a linear function

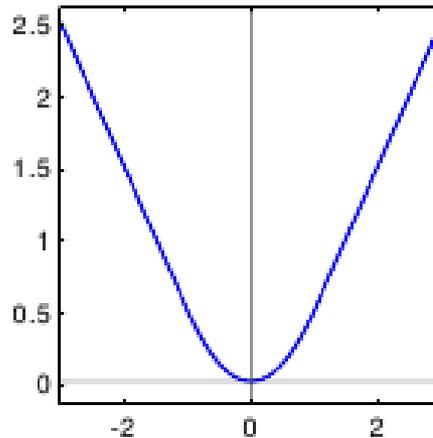
$$\rho(e) = \begin{cases} \frac{e^2}{2} & \text{if } |e| < c \\ c(|e| - \frac{c}{2}) & \text{otherwise} \end{cases}$$



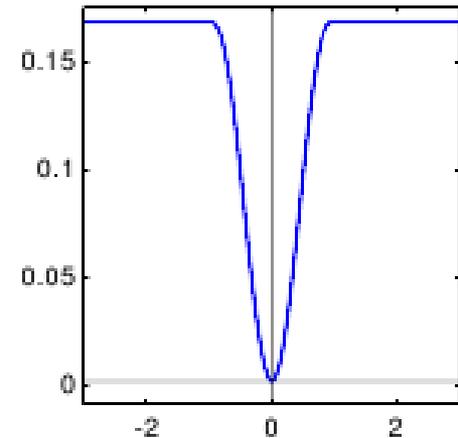
# Different Rho Functions



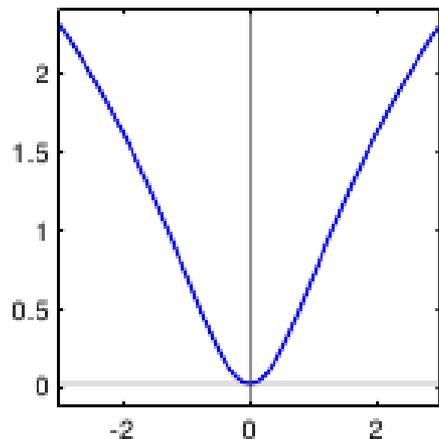
L1 norm



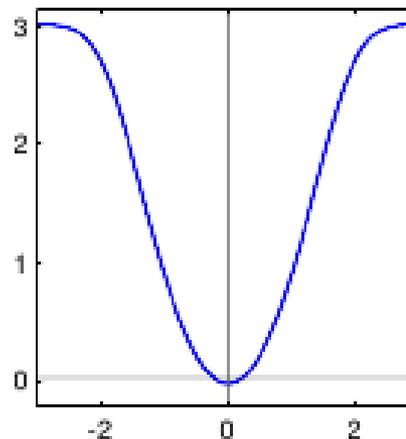
Huber



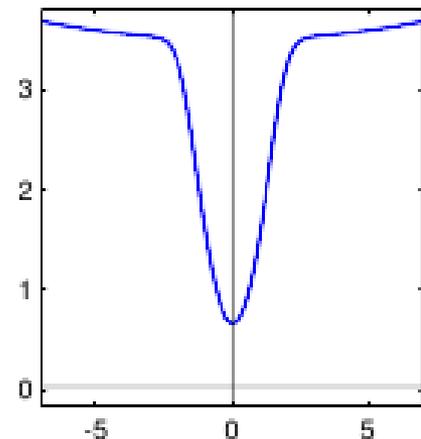
Tukey



Cauchy

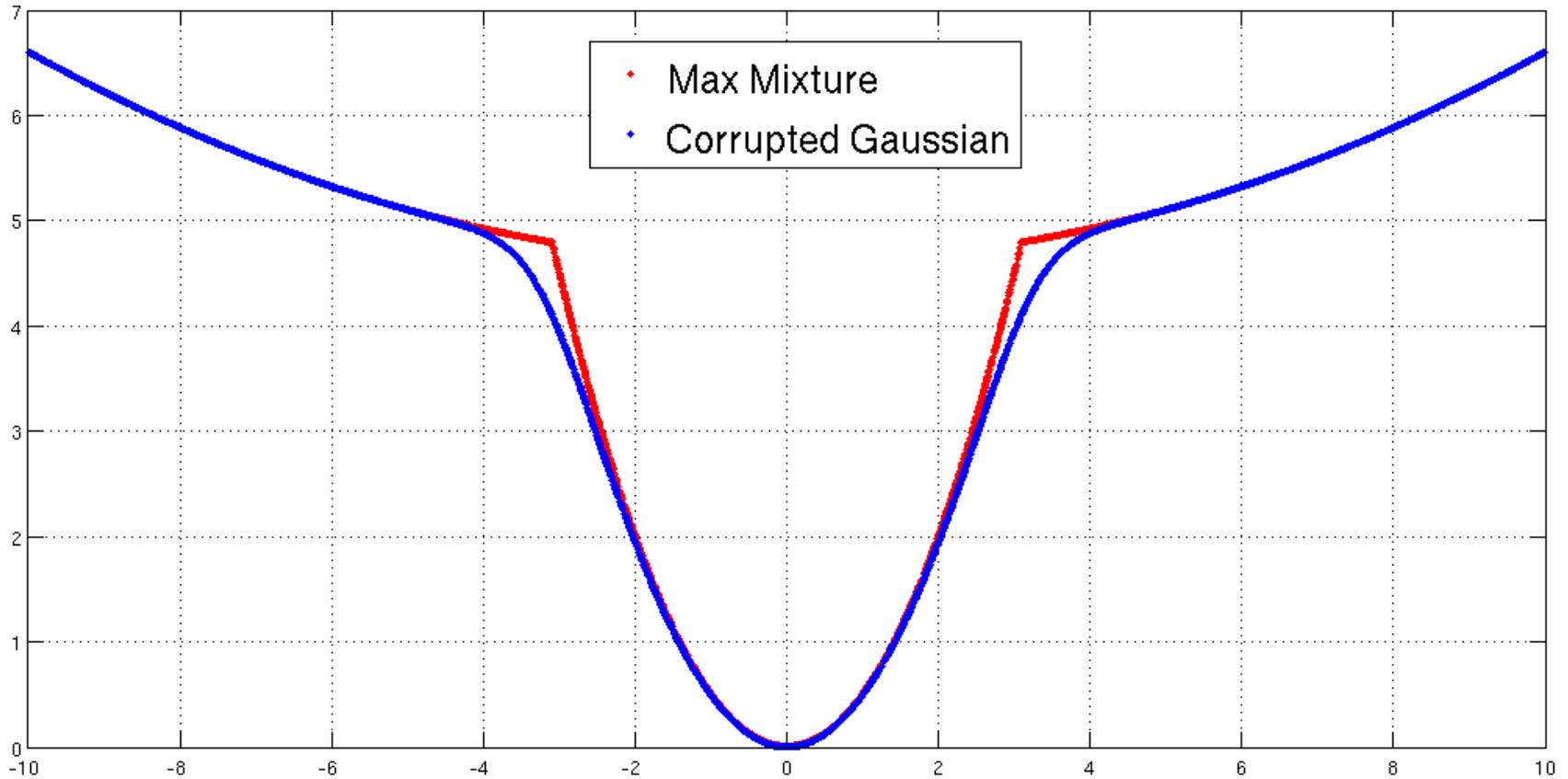


Blake-Zisserman



Corrupted G.

# MM Cost Function For Outliers



# Robust Estimation

- Choice of the rho function depends on the problem at hand
- Huber function is often used
- MM for outlier handling is similar to a corrupted Gaussian
- MM additionally supports multi-model measurements
- Dynamic Covariance Scaling is a robust M-estimator

# Conclusions

- Sum of Gaussians cannot be used easily in the optimization framework
- Max-Mixture formulation approximates the sum by the max operator
- This allows for handling data association ambiguities and failures
- Minimal performance overhead
- Minimal code changes for integration

# Literature

## **Max-Mixture Approach:**

- Olson, Agarwal: “Inference on Networks of Mixtures for Robust Robot Mapping”

## **Dynamic Covariance Scaling:**

- Agarwal, Tipaldi, Spinello, Stachniss, Burgard: “Robust Map Optimization Using Dynamic Covariance Scaling”

# Slide Information

- These slides have been created by Cyrill Stachniss as part of the robot mapping course taught in 2012/13 and 2013/14. I created this set of slides partially extending existing material of Edwin Olson, Pratik Agarwal, and myself.
- I tried to acknowledge all people that contributed image or video material. In case I missed something, please let me know. If you adapt this course material, please make sure you keep the acknowledgements.
- Feel free to use and change the slides. If you use them, I would appreciate an acknowledgement as well. To satisfy my own curiosity, I appreciate a short email notice in case you use the material in your course.
- My video recordings are available through YouTube:  
[http://www.youtube.com/playlist?list=PLgnQpQtFTOGQrZ4O5QzbIHgI3b1JHimN\\_&feature=g-list](http://www.youtube.com/playlist?list=PLgnQpQtFTOGQrZ4O5QzbIHgI3b1JHimN_&feature=g-list)