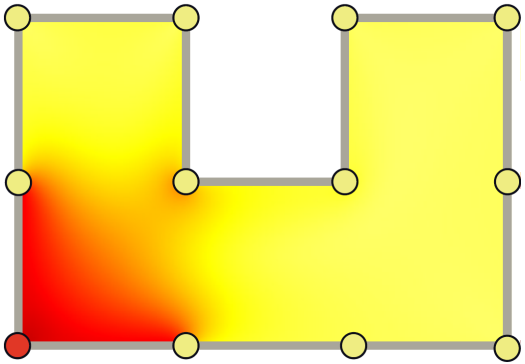
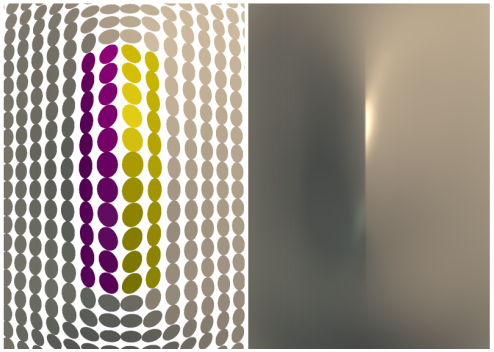
Representation and Approximation of Visual Data





Dr. Cengiz Öztireli

Computer Graphics Laboratory

ETH Zürich

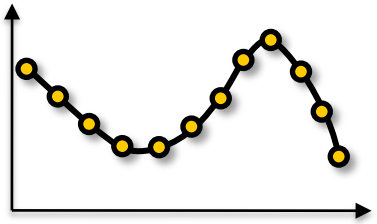
ETH zürich

Disney Research

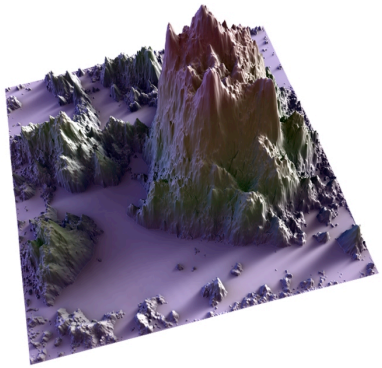
cgl

Representation

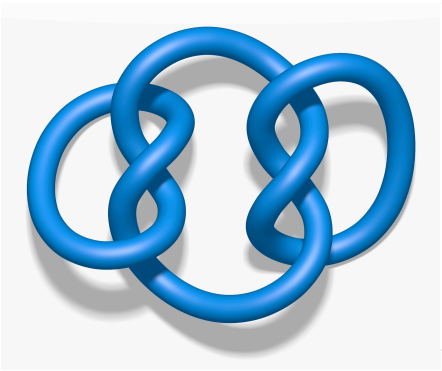
- Continuous vs. discrete



Image



Height Field

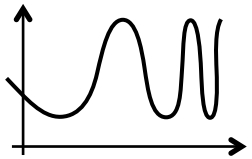


3D Model

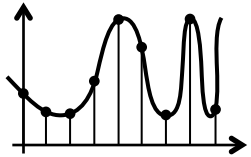
Texture

Representation

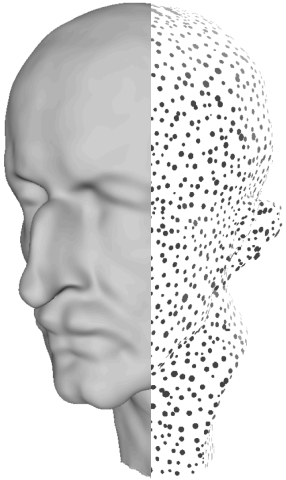
- Discretized forms of visual data

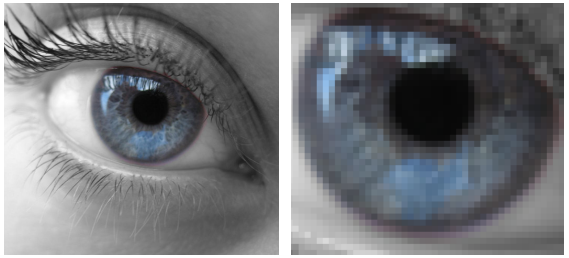


Continuous



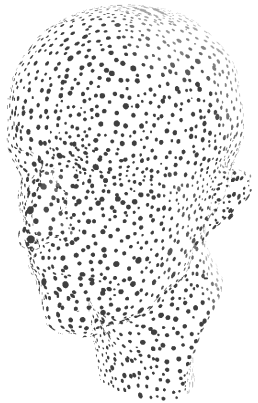
Sampled





Representation

- Convert between cont. and discrete forms



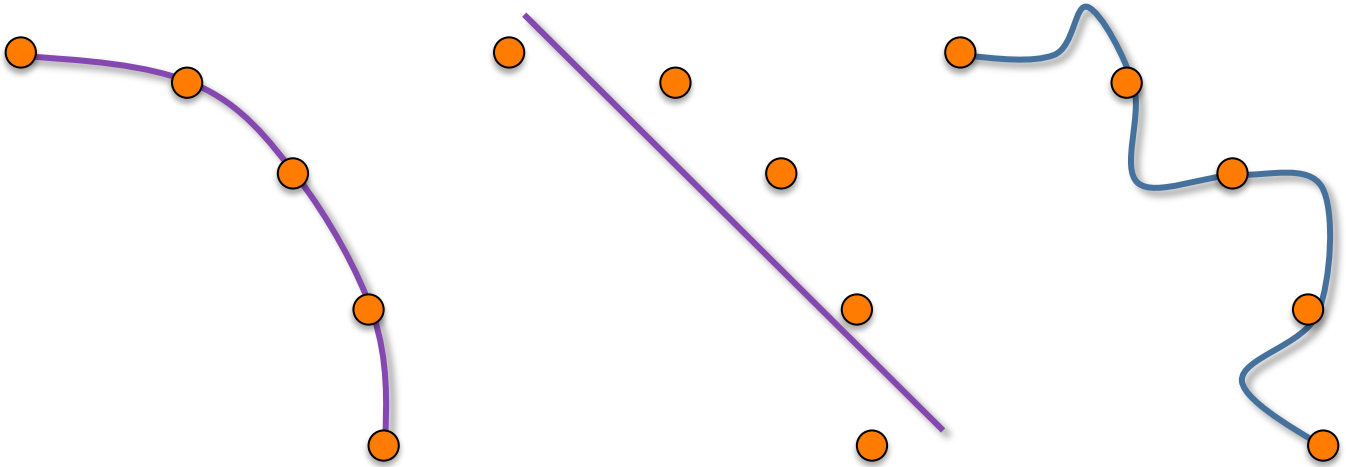
f_i

Reconstruction

Sampling

$f(\mathbf{x})$

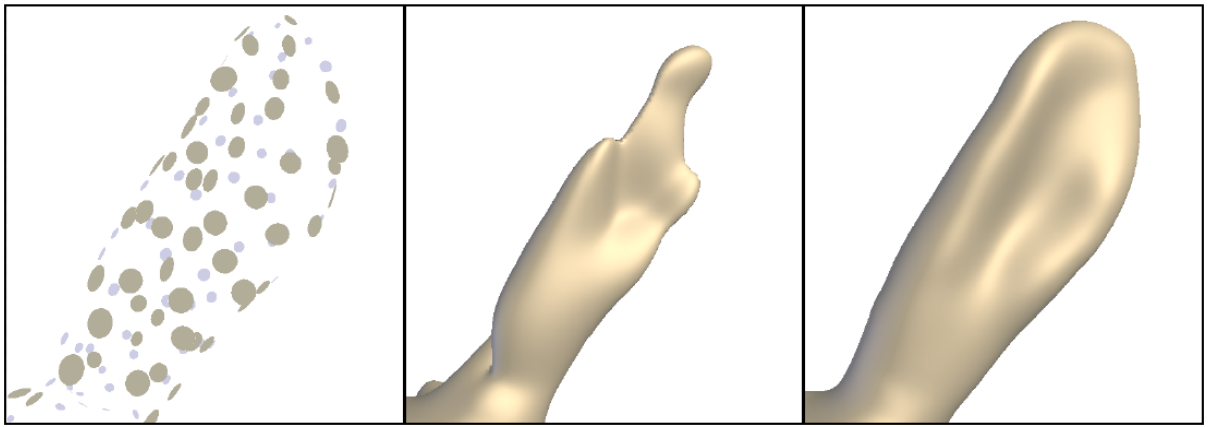
The approximation problem



Infinitely many solutions!
Ill-posed problem without prior assumptions.

The approximation problem

- Some solutions are preferred



The approximation problem

- Desired properties
  - Consistently intuitive results
  - Robust to noise in the data
  - Efficient to compute
  - Amenable to analytic analysis

The approximation problem

- Example problems

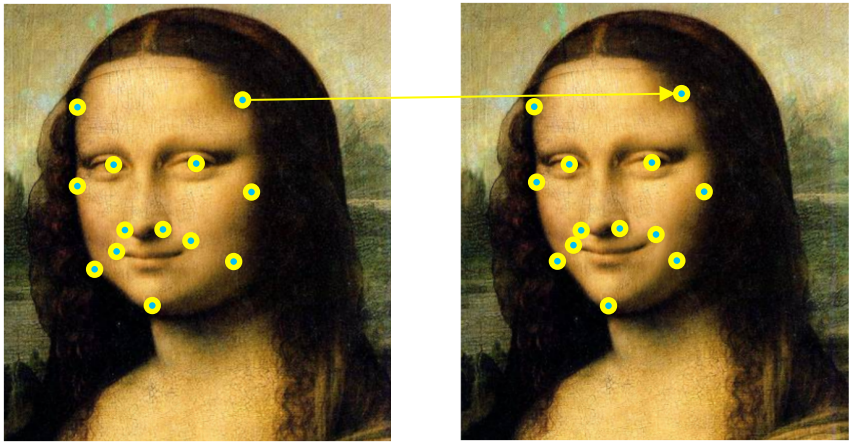
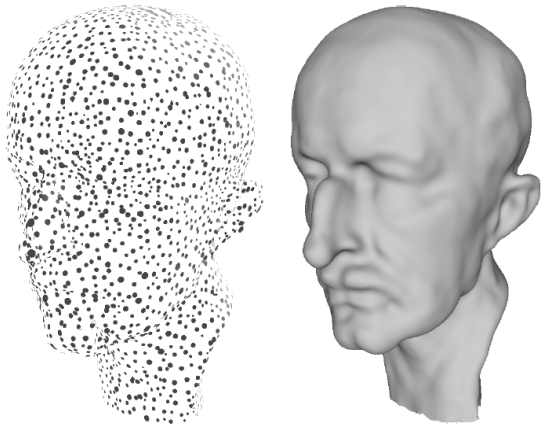


Image deformations



3D surface reconstruction

The approximation problem

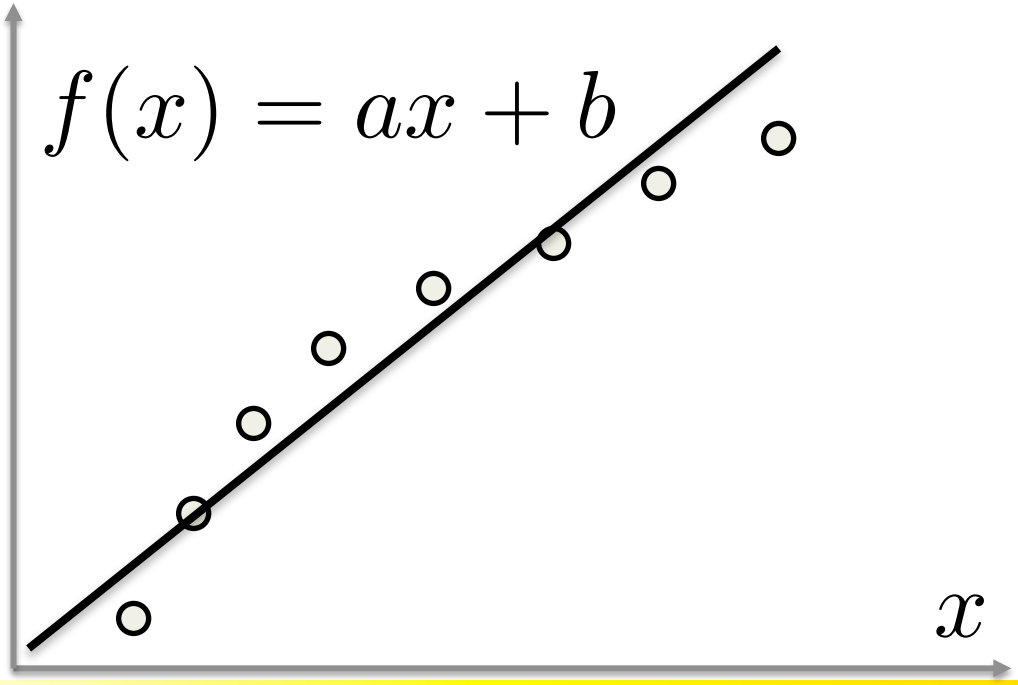
- Example problems

Edge-preserving image filtering

Edge-preserving mesh smoothing

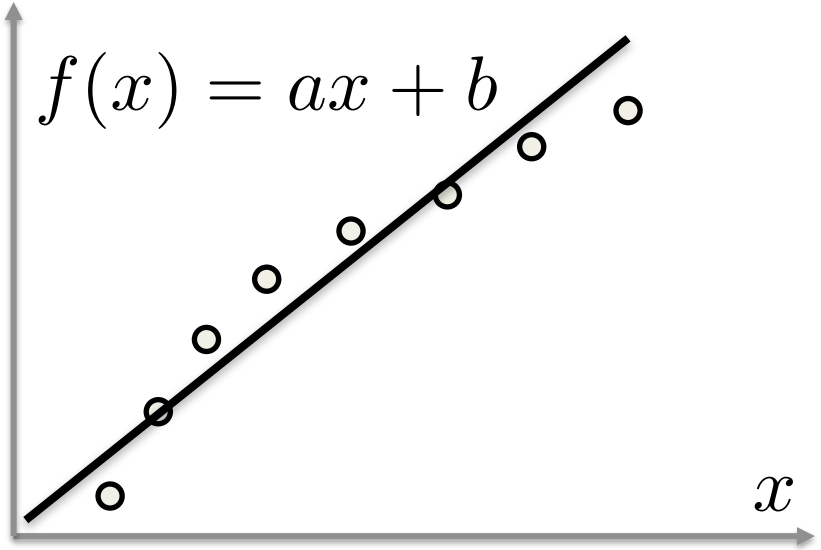
Least Squares

- Problem



Least Squares

- Problem



$$\min_{a,b} \sum_{i=1}^{n} (f(x_i) - y_i)^2$$

$$\min_{a,b} \sum_{i=1}^{n} (ax_i + b - y_i)^2$$

Least Squares

- Multi-dimensional problem

$$f(\mathbf{x}) : \mathbb{R}^d \rightarrow \mathbb{R} \qquad \min_{f \in \Pi_m^d} \sum_i \left(f(\mathbf{x}_i) - f_i\right)^2$$

Π_m^d : polynomials of degree m in d dimensions

$$f(\mathbf{x}) = \mathbf{b}(\mathbf{x})^T \mathbf{c}$$

$$m = 2, d = 2 \qquad \mathbf{b}(\mathbf{x}) = \begin{bmatrix} 1 & x & y & x^2 & y^2 & xy \end{bmatrix}^T$$

$$f(\mathbf{x}) = c_0 + c_1 x + c_2 y + c_3 x^2 + c_4 y^2 + c_5 xy$$

Least Squares

- Multi-dimensional problem

$$f(\mathbf{x}) : \mathbb{R}^d \rightarrow \mathbb{R} \qquad \min_{f \in \Pi_m^d} \sum_i \left(f(\mathbf{x}_i) - f_i\right)^2$$

$$f(\mathbf{x}) = \mathbf{b}(\mathbf{x})^T \mathbf{c}$$

$$\min_{\mathbf{c}} E(\mathbf{c}) \qquad E(\mathbf{c}) = \sum_i \left(\mathbf{b}(\mathbf{x}_i)^T \mathbf{c} - f_i\right)^2$$

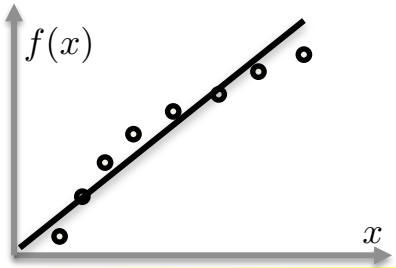
Least Squares

- Multi-dimensional problem

$$\min_{\mathbf{c}} E(\mathbf{c}) \qquad E(\mathbf{c}) = \sum_i \left(\mathbf{b}(\mathbf{x}_i)^T \mathbf{c} - f_i\right)^2$$

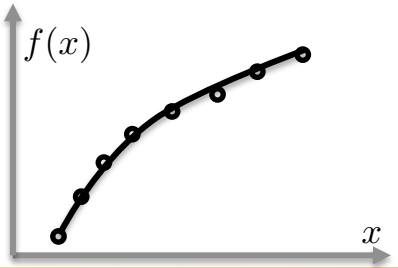
$$m = 1, d = 1$$

$$E(\mathbf{c}) = \sum_i \left(c_0 + c_1 x_i - f_i\right)^2$$



$$m = 2, d = 1$$

$$E(\mathbf{c}) = \sum_i \left(c_0 + c_1 x_i + c_2 x_i^2 - f_i\right)^2$$



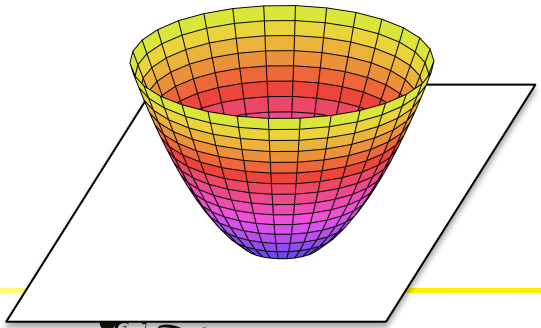
Least Squares

- Multi-dimensional problem

$$\min_{\mathbf{c}} E(\mathbf{c}) \qquad E(\mathbf{c}) = \sum_i \left(\mathbf{b}(\mathbf{x}_i)^T \mathbf{c} - f_i\right)^2$$

$$m = 2, d = 2$$

$$E(\mathbf{c}) = \sum_i \left(c_0 + c_1 x + c_2 y + c_3 x^2 + c_4 y^2 + c_5 xy - f_i\right)^2$$



Least Squares

- Solution of the multi-dimensional problem

$$\min_{\mathbf{c}} E(\mathbf{c}) \qquad E(\mathbf{c}) = \sum_i \left(\mathbf{b}(\mathbf{x}_i)^T\mathbf{c} - f_i\right)^2 \qquad \mathbf{b}(\mathbf{x}_i) = [b_1(\mathbf{x}_i) \cdots b_m(\mathbf{x}_i)]^T$$

$$\frac{\partial E(\mathbf{c})}{\partial c_k} = \sum_i 2b_k(\mathbf{x}_i)\left[\mathbf{b}(\mathbf{x}_i)^T\mathbf{c} - f_i\right] = 0$$

$$\frac{\partial E(\mathbf{c})}{\partial \mathbf{c}} = 2\sum_i \mathbf{b}(\mathbf{x}_i)\left[\mathbf{b}(\mathbf{x}_i)^T\mathbf{c} - f_i\right] = 0$$

$$\sum_i \mathbf{b}(\mathbf{x}_i)\mathbf{b}(\mathbf{x}_i)^T\mathbf{c} = \sum_i \mathbf{b}(\mathbf{x}_i) f_i$$

$$\mathbf{c} = \left[\sum_i \mathbf{b}(\mathbf{x}_i)\mathbf{b}(\mathbf{x}_i)^T\right]^{-1} \sum_i \mathbf{b}(\mathbf{x}_i) f_i$$

Least Squares

- Solution of the multi-dimensional problem

Example

$$m = 2, d = 1 \quad E(\mathbf{c}) = \sum_i \left(c_0 + c_1 x + c_2 x^2 - f_i\right)^2$$

$$\sum_i \begin{bmatrix} 1 & x_i & x_i^2 \\ x_i & x_i^2 & x_i^3 \\ x_i^2 & x_i^3 & x_i^4 \end{bmatrix} \begin{bmatrix} c_0 \\ c_1 \\ c_2 \end{bmatrix} = \sum_i \begin{bmatrix} 1 \\ x_i \\ x_i^2 \end{bmatrix} f_i$$

Weighted Least Squares

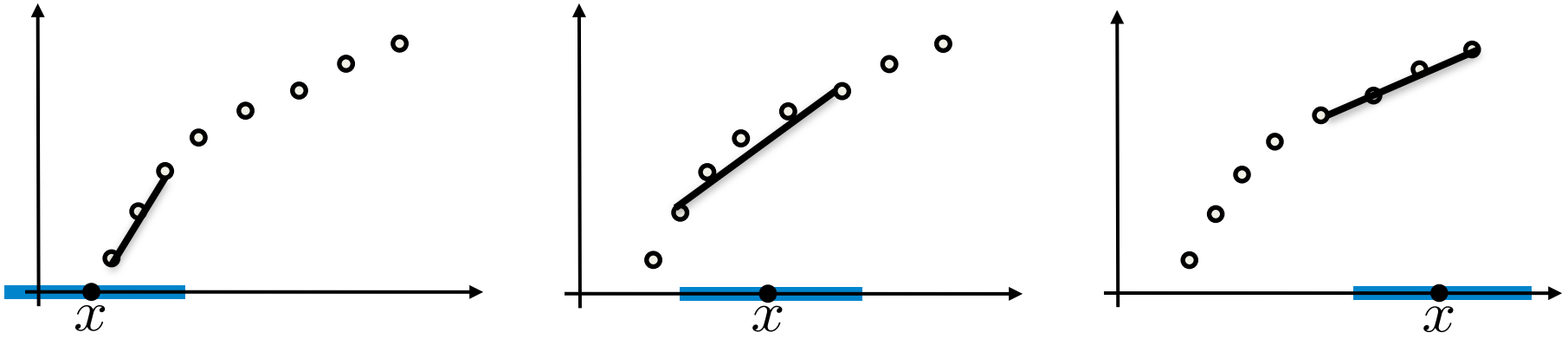
- Multiply the terms with given weights

LS $\quad \min_{\mathbf{c}} E(\mathbf{c}) \qquad E(\mathbf{c}) = \sum_i \left(\mathbf{b}(\mathbf{x}_i)^T \mathbf{c} - f_i\right)^2$

WLS $\quad \min_{\mathbf{c}} E(\mathbf{c}) \qquad E(\mathbf{c}) = \sum_i \left(\mathbf{b}(\mathbf{x}_i)^T \mathbf{c} - f_i\right)^2 w_i$

Moving Least Squares

- Idea: make the weights local



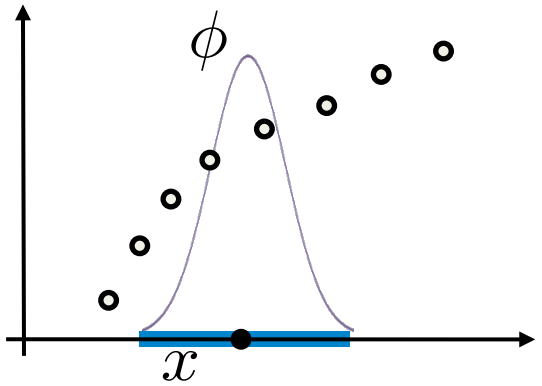
Moving Least Squares

- Idea: make the weights local

$$f(\mathbf{x}) = \min_{f_{\mathbf{x}} \in \Pi_m^d} \sum_i \phi(||\mathbf{x} - \mathbf{x}_i||)\,(f_{\mathbf{x}}(\mathbf{x}_i) - f_i)^2$$

Local approximation

Weights depend on x



Moving Least Squares

- Idea: make the weights local

$$\mathbf{c}(\mathbf{x}) = \operatorname{argmin}_{\mathbf{c}} E_{\mathbf{x}}(\mathbf{c}) = \sum_i \phi(||\mathbf{x} - \mathbf{x}_i||) \left(\mathbf{b}(\mathbf{x}_i)^T \mathbf{c} - f_i\right)^2$$

$$f(\mathbf{x}) = \mathbf{b}(\mathbf{x})^T \mathbf{c}(\mathbf{x})$$

In comparison, LS:

$$\mathbf{c} = \operatorname{argmin}_{\mathbf{c}} E(\mathbf{c}) = \sum_i \left(\mathbf{b}(\mathbf{x}_i)^T \mathbf{c} - f_i\right)^2$$

$$f(\mathbf{x}) = \mathbf{b}(\mathbf{x})^T \mathbf{c}$$

Moving Least Squares

- Local solution

$$\mathbf{c}(\mathbf{x}) = \left[\sum_i \phi_i(\mathbf{x})\mathbf{b}(\mathbf{x}_i)\mathbf{b}(\mathbf{x}_i)^T\right]^{-1} \sum_i \phi_i(\mathbf{x})\mathbf{b}(\mathbf{x}_i) f_i$$

$$\phi_i(\mathbf{x}) = \phi(||\mathbf{x} - \mathbf{x}_i||)$$

$$f(\mathbf{x}) = \mathbf{b}(\mathbf{x})^T \mathbf{c}(\mathbf{x})$$

Moving Least Squares

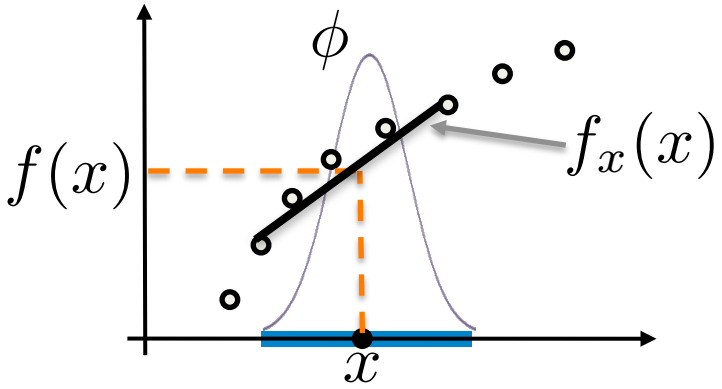
- Local solution

Example $m = 1, d = 1$

$$\min_{c_0, c_1} \sum_i \phi_i(x) \left(c_0 + c_1 x_i - f_i\right)^2$$

$$f_x(x) = c_0 + c_1 x$$

$$f(x) = f_x(x)$$



Moving Least Squares

- Interpretation in terms of basis functions

$$\mathbf{c}(\mathbf{x}) = \underbrace{\left[\sum_i \phi_i(\mathbf{x})\mathbf{b}(\mathbf{x}_i)\mathbf{b}(\mathbf{x}_i)^T\right]^{-1}}_{\mathbf{B}(\mathbf{x})} \sum_i \phi_i(\mathbf{x})\mathbf{b}(\mathbf{x}_i) f_i$$

$$f(\mathbf{x}) = \mathbf{b}(\mathbf{x})^T \mathbf{c}(\mathbf{x})$$

$$f(\mathbf{x}) = \sum_i u_i(\mathbf{x}) f_i \qquad u_i(\mathbf{x}) = \phi_i(\mathbf{x})\mathbf{b}(\mathbf{x})^T \mathbf{B}(\mathbf{x})^{-1}\mathbf{b}(\mathbf{x}_i)$$

Moving Least Squares

- Extension: reproducing local functions

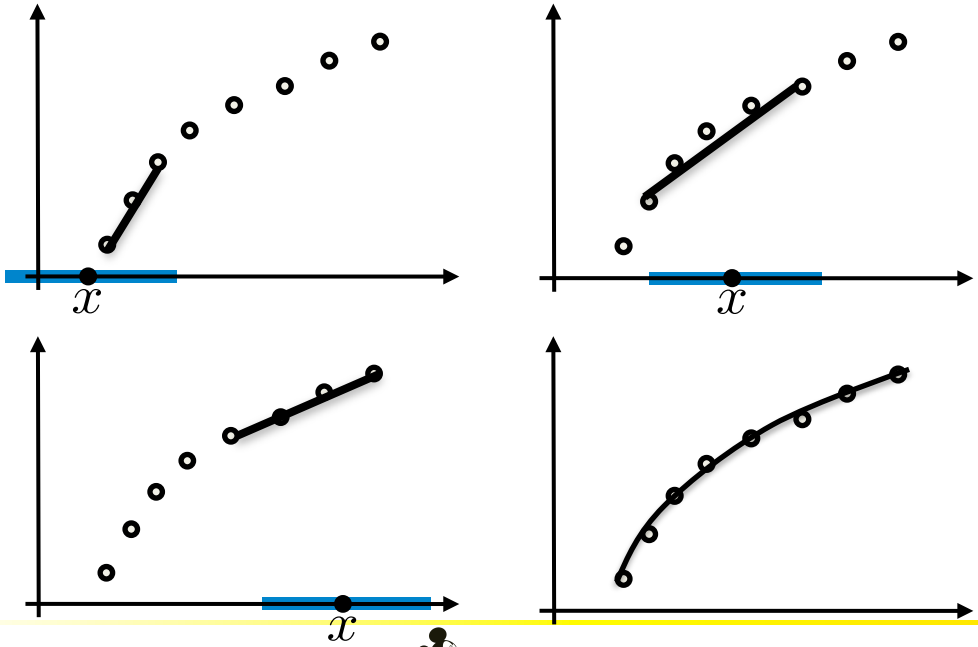
$$\mathbf{c}(\mathbf{x}) = \text{argmin}_{\mathbf{c}}\ E_{\mathbf{x}}(\mathbf{c}) = \sum_i \phi(||\mathbf{x} - \mathbf{x}_i||)\left(\mathbf{b}(\mathbf{x}_i)^T\mathbf{c} - f_i(\mathbf{x})\right)^2$$

$$f(\mathbf{x}) = \mathbf{b}(\mathbf{x})^T\mathbf{c}(\mathbf{x})$$

Example $f_i(\mathbf{x}) = \mathbf{n}^T(\mathbf{x} - \mathbf{x}_i)$

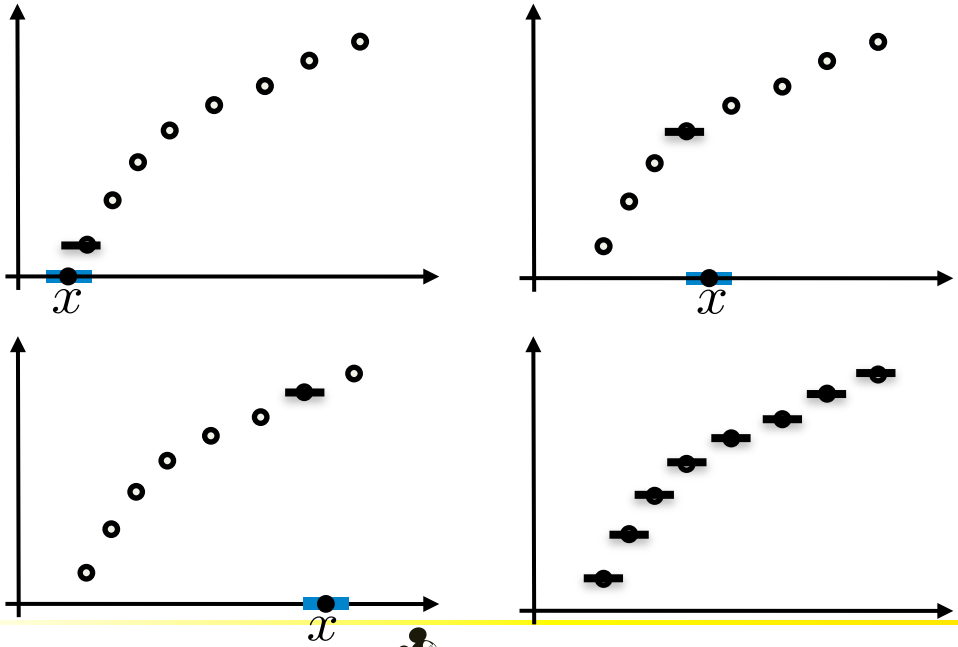
Moving Least Squares

- Sampling conditions



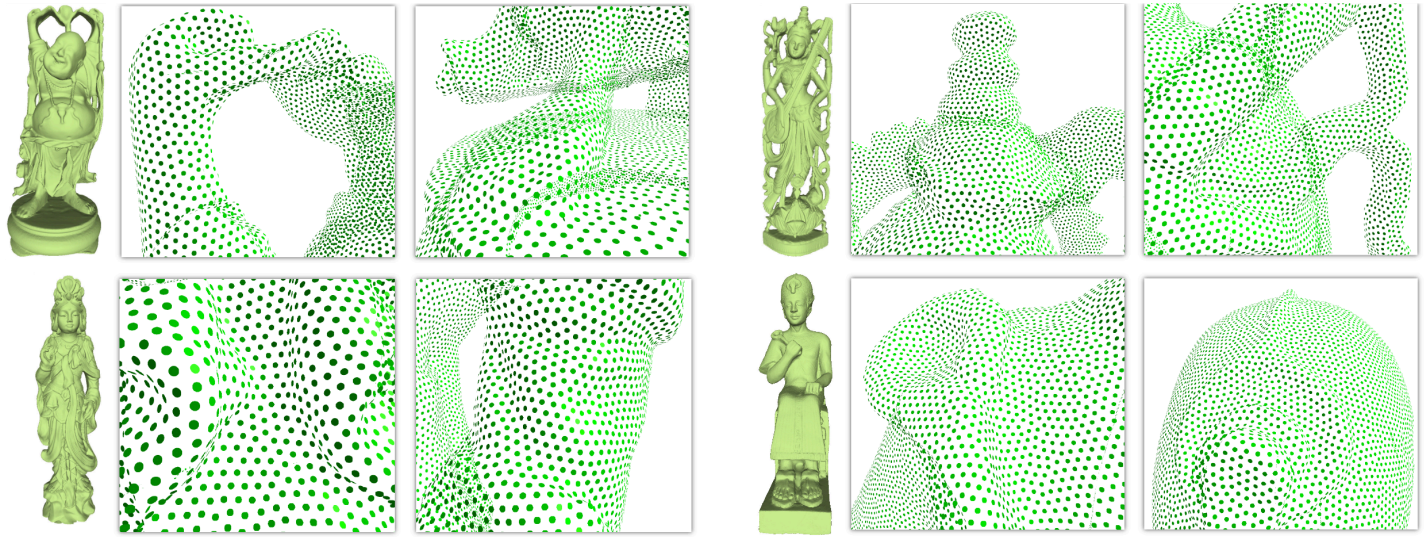
Moving Least Squares

- Sampling conditions



Moving Least Squares

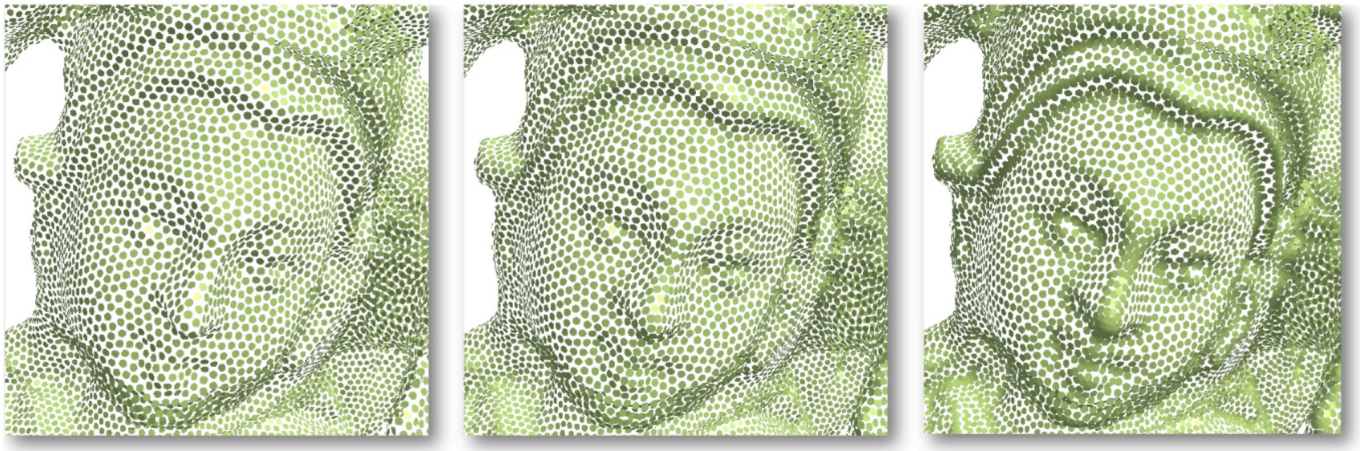
- Sampling should follow ϕ



Spectral sampling of manifolds, SIGGRAPH Asia 2010

Moving Least Squares

- Sampling should follow ϕ



Implicit MLS Surfaces

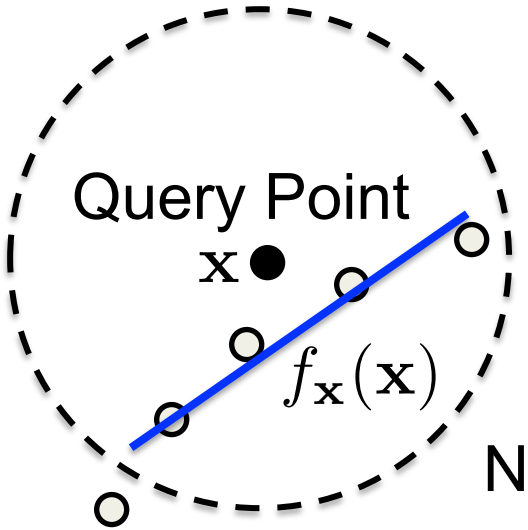
- Basic problem
  - Given sample points & attributes
  - Compute a function

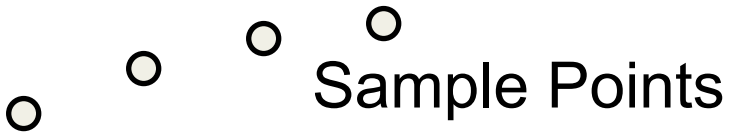
$$f(\mathbf{x}) : \mathbb{R}^2 \text{ or } \mathbb{R}^3 \rightarrow \mathbb{R}$$

  - such that the curve/surface is given by

$$\mathcal{S} = \{\mathbf{x} | f(\mathbf{x}) = 0,\ \nabla f(\mathbf{x}) \neq \mathbf{0}\}$$

Implicit MLS Surfaces



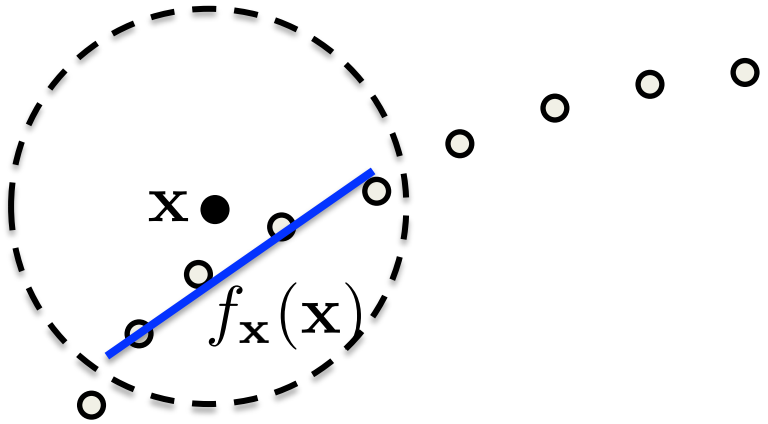


Neighborhood defined by $\phi(||\mathbf{x} - \cdot||)$

$$f(\mathbf{x}) = f_{\mathbf{x}}(\mathbf{x})$$

Implicit MLS Surfaces

Example $m = 1, d = 2$



$$f_{\mathbf{x}}(\mathbf{x}) = c_0(\mathbf{x}) + c_1(\mathbf{x})x + c_2(\mathbf{x})y$$

Implicit MLS Surfaces

How can we avoid the trivial solution

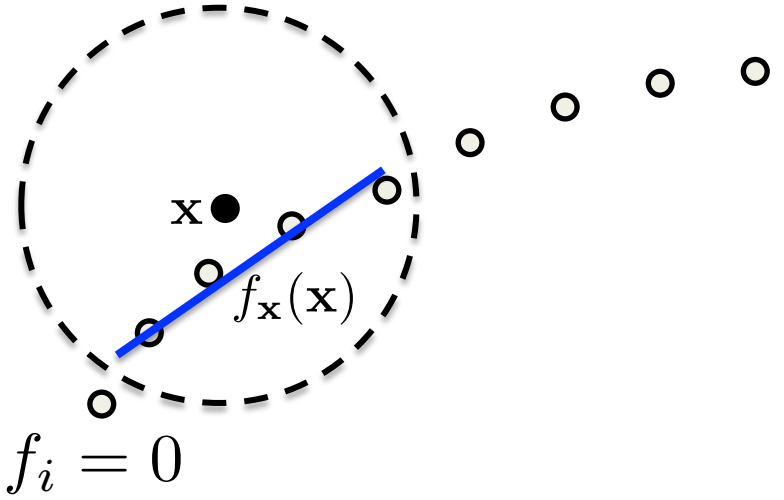
$$f(\mathbf{x}) = 0 \; \forall \mathbf{x}$$

Gradient constraints

$$||\nabla f_{\mathbf{x}}(\mathbf{x})|| = 1 \quad \nabla f(\mathbf{x}_i) = \mathbf{n}_i$$

Reproduce local functions

$$f_i(\mathbf{x}) = \mathbf{n}_i^T(\mathbf{x} - \mathbf{x}_i)$$

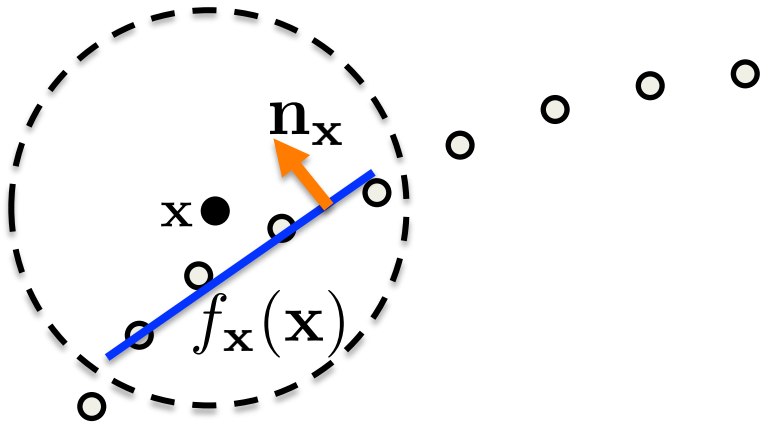


Implicit MLS Surfaces

- Example

$$m = 1, d = 2$$

$$f_{\mathbf{x}}(\mathbf{x}) = \mathbf{n}_{\mathbf{x}}^T \mathbf{x} + o_{\mathbf{x}} \qquad ||\mathbf{n}_{\mathbf{x}}|| = 1$$



$$(\mathbf{n}_{\mathbf{x}}, o_{\mathbf{x}}) = \text{argmin}_{\mathbf{n},o} \sum_i \phi_i(\mathbf{x}) \left(\mathbf{n}^T \mathbf{x}_i + o\right)^2 \qquad ||\mathbf{n}|| = 1$$

Implicit MLS Surfaces

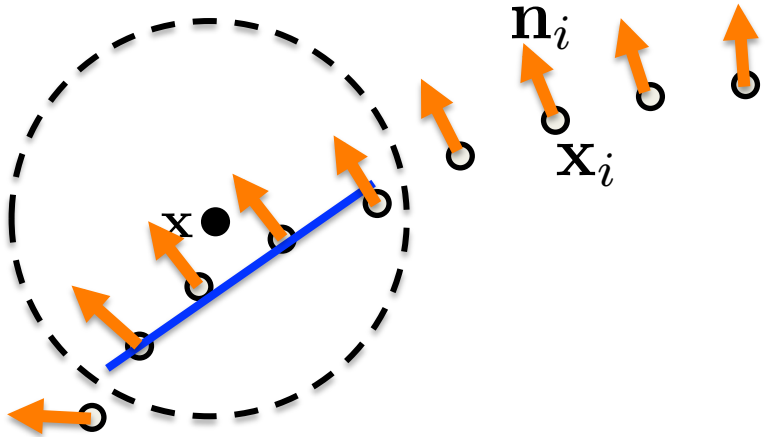
- Example

$$m = 0, d = 2$$

$$f_i(\mathbf{x}) = \mathbf{n}_i^T(\mathbf{x} - \mathbf{x}_i)$$

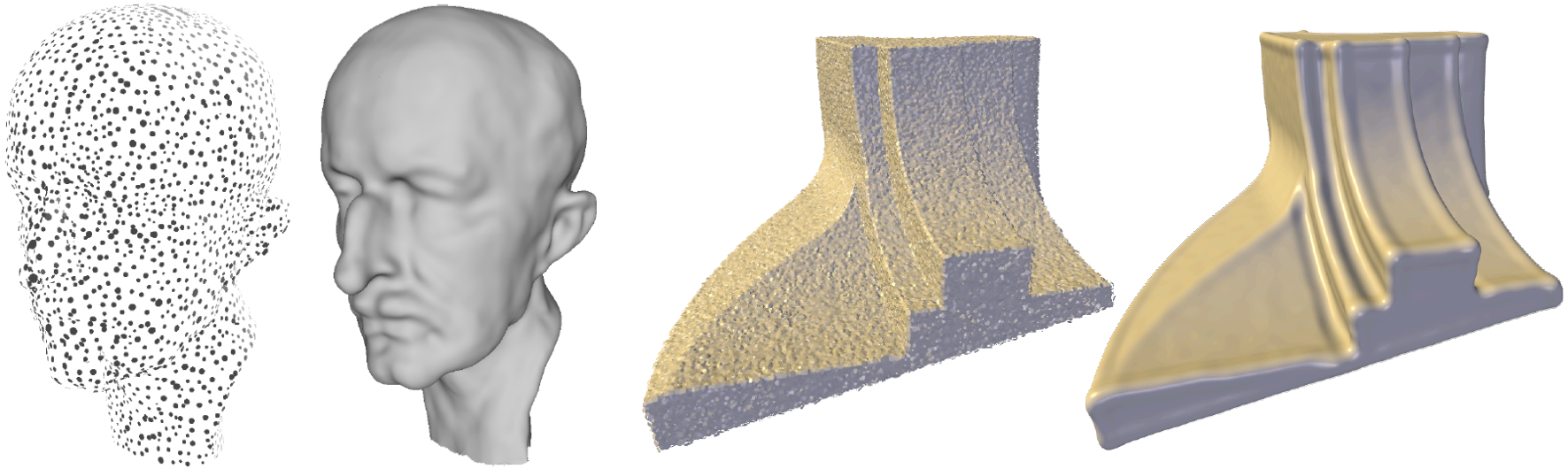
$$f(\mathbf{x}) = c_0$$

$$\text{argmin}_{c_0} \sum_i \phi_i(\mathbf{x})(c_0 - f_i(\mathbf{x}))^2$$



Implicit MLS Surfaces

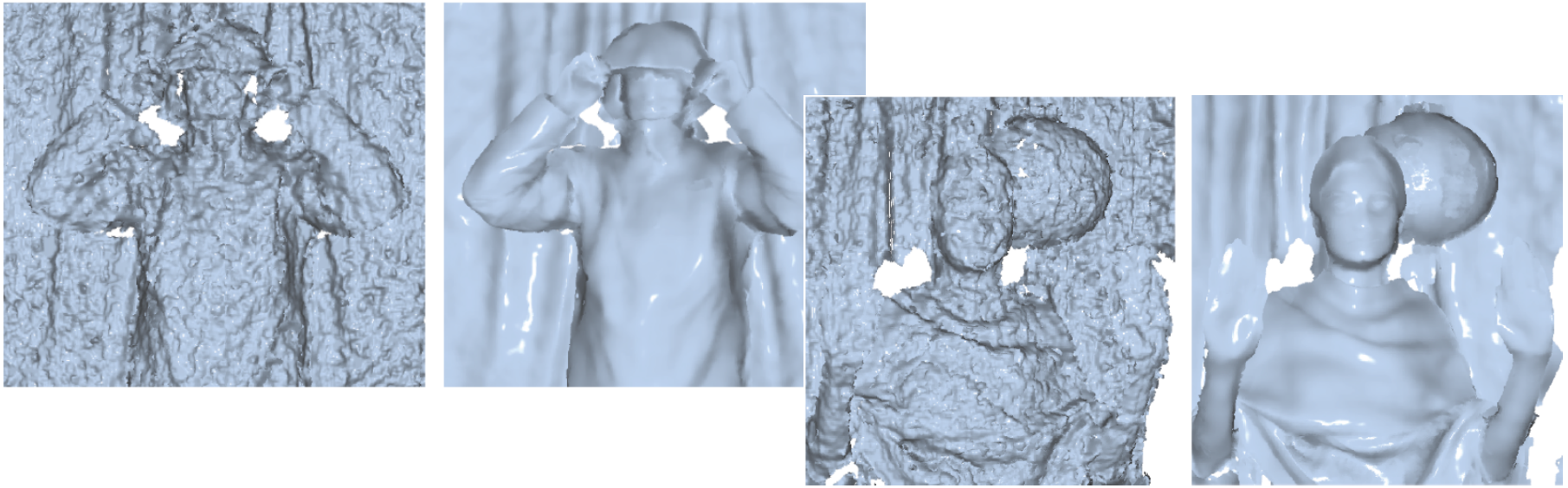
- Examples in 3D



Feature Preserving Point Set Surfaces based on Non-Linear Kernel Regression, Eurographics 2009

Implicit MLS Surfaces

- Examples in 3D



Spatio-Temporal Geometry Fusion for Multiple Hybrid Cameras using Moving Least Squares Surfaces, Eurographics 2014

MLS Image Deformation

- Given correspondences, get the new image

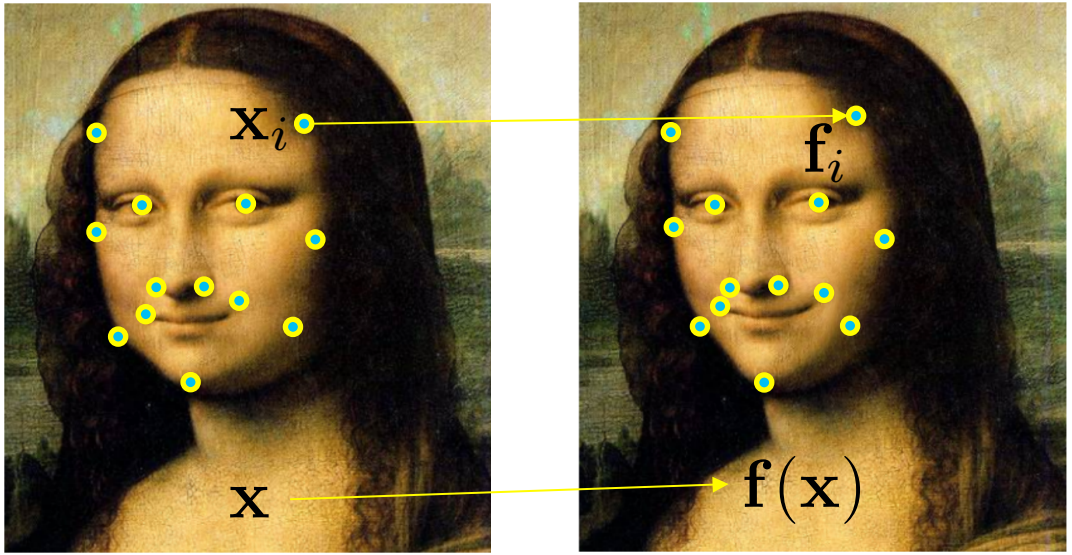
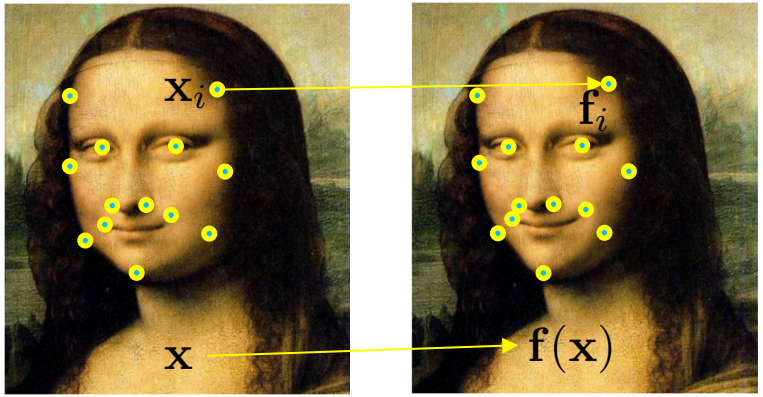


Image Deformation Using Moving Least Squares, SIGGRAPH 2006

MLS Image Deformation

- Given correspondences, get the new image

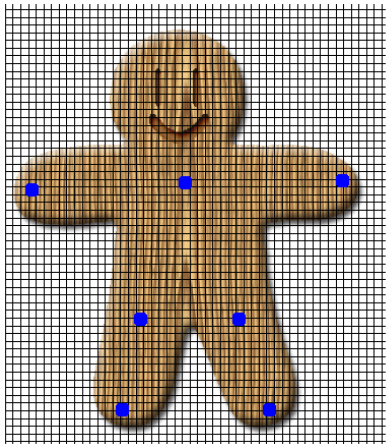


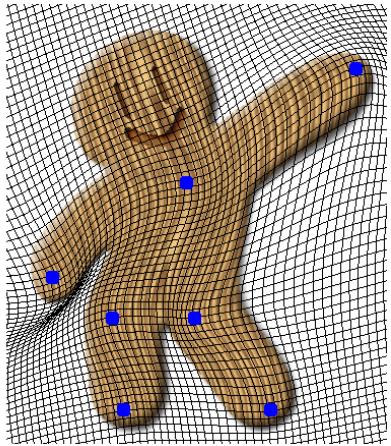
$$\mathbf{f}(\mathbf{x}) = \mathbf{f}_{\mathbf{x}}(\mathbf{x}) = \mathbf{R}_{\mathbf{x}}\mathbf{x} + \mathbf{t}_{\mathbf{x}}$$

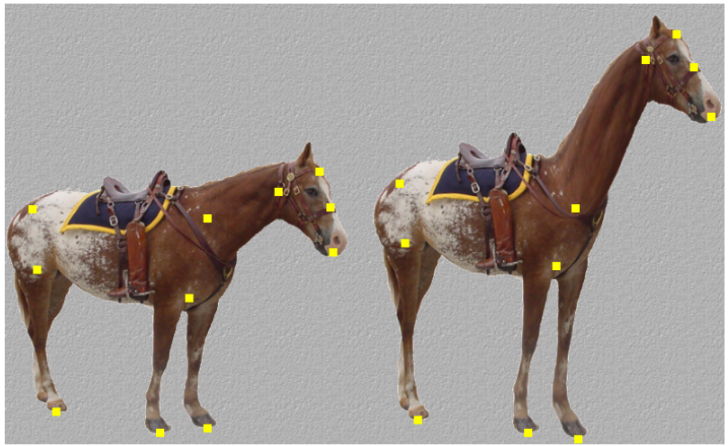
$$\min_{\mathbf{R}_{\mathbf{x}},\mathbf{t}_{\mathbf{x}}} \sum_i \phi(||\mathbf{x} - \mathbf{x}_i||)||\mathbf{f}_{\mathbf{x}}(\mathbf{x}_i) - \mathbf{f}_i||^2$$

MLS Image Deformation

- Examples







MLS Image Color Transfer

- Adjust colors of an image based on another

Input

Reference

Output

Color Transfer Using Probabilistic Moving Least Squares, CVPR 2014

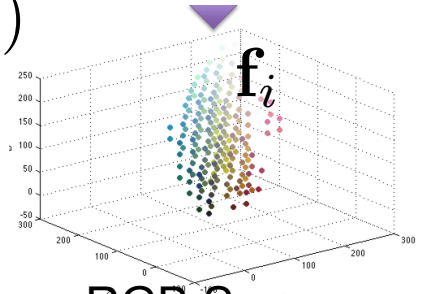
MLS Image Color Transfer

- Adjust colors of an image based on another

Input Image

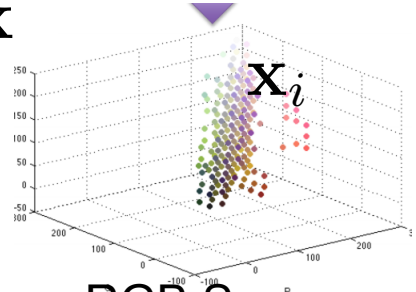
Reference Image

$\mathbf{f}(\mathbf{x})$



RGB Space

$\mathbf{x}$



RGB Space

ETH zürich

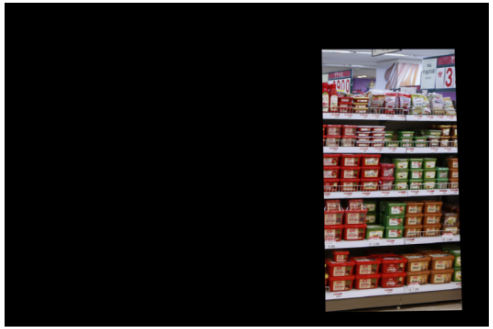
Disney Research

cgl

MLS Image Color Transfer

- Example

Input

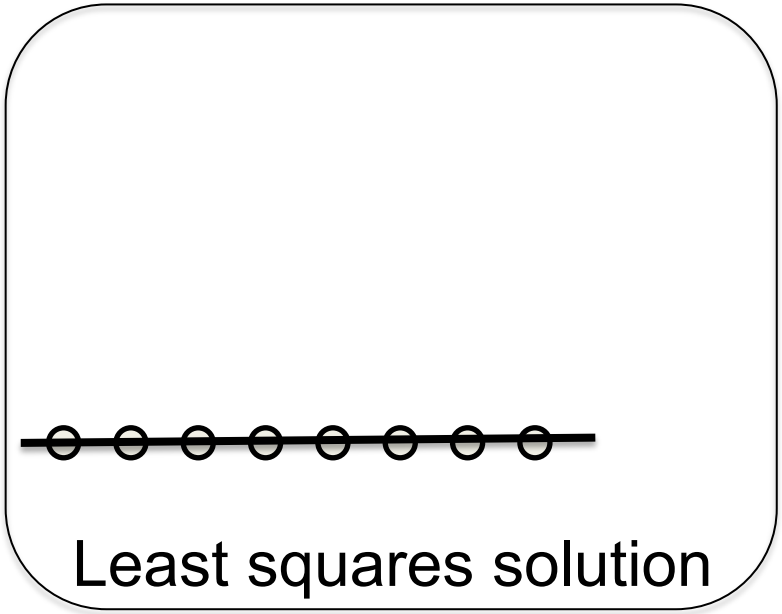


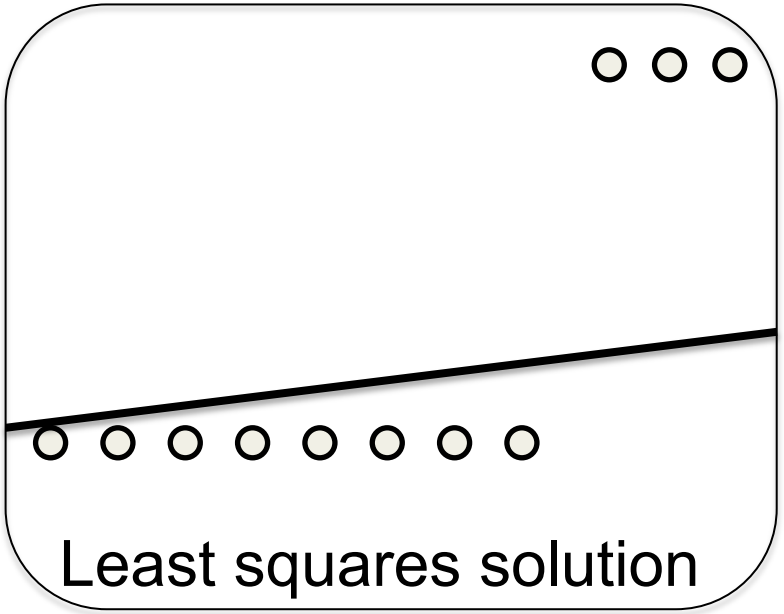
Reference

Output

Robust Approximation

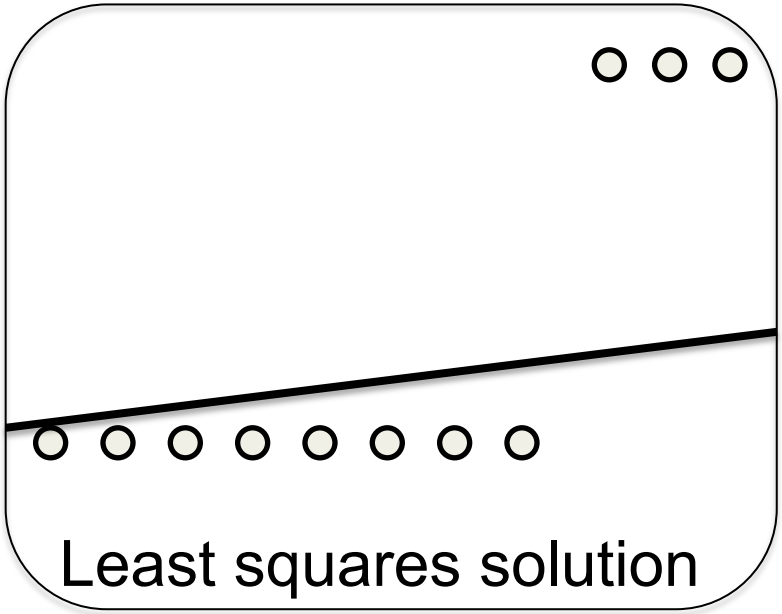
- Problem: outliers

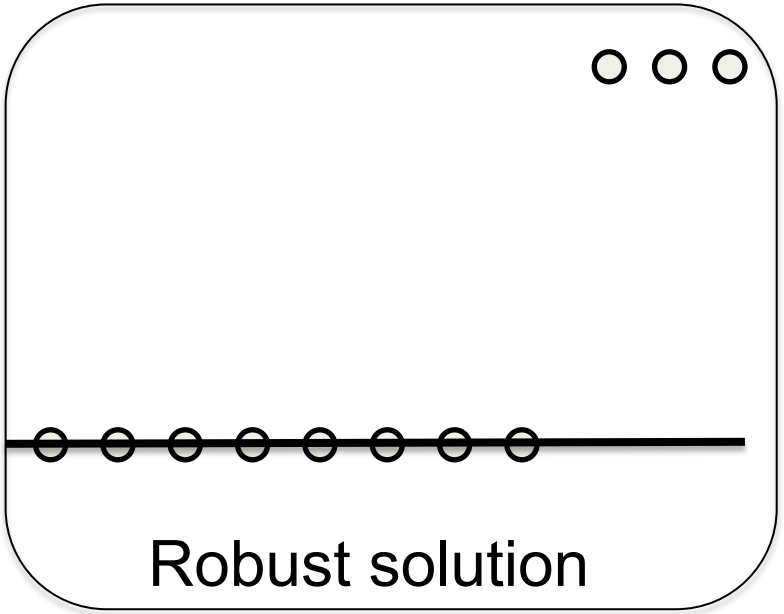




Robust Approximation

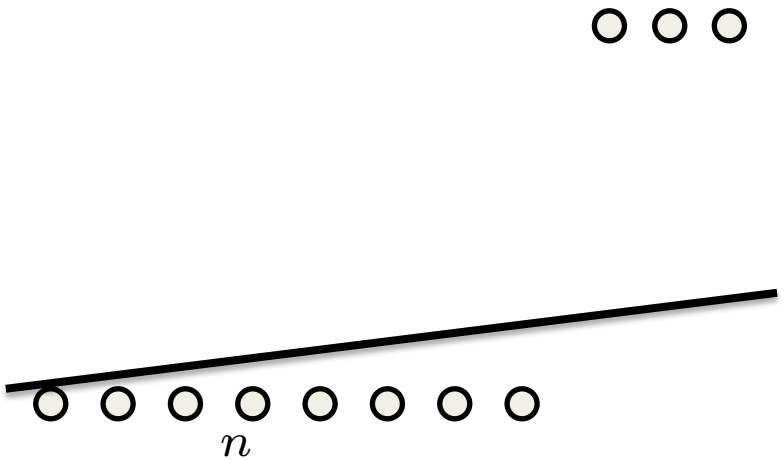
- Problem: outliers





Robust Approximation

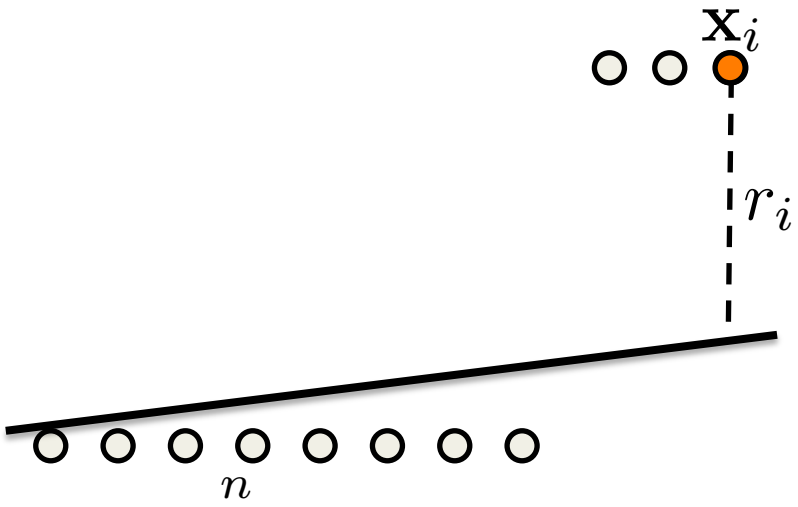
- Problem: squared norm is too sensitive to outliers



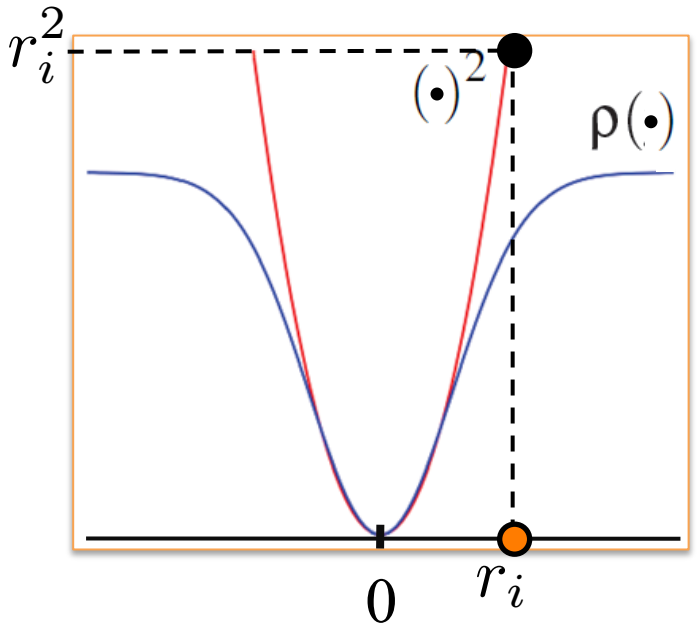
$$\min_{a,b} \sum_{i=1}^{n} (ax_i + b - y_i)^2$$

Robust Approximation

- Problem: squared norm is too sensitive to outliers

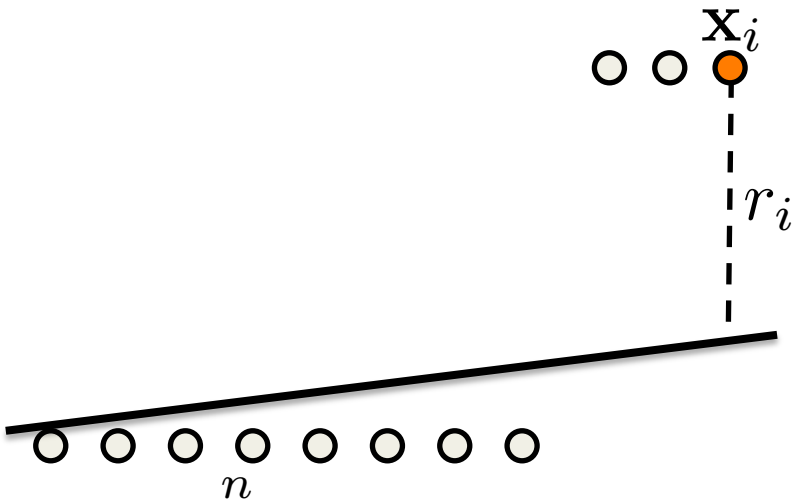


$$\min_{a,b} \sum_{i=1}^{n} \underbrace{(ax_i + b - y_i)^2}_{r_i}$$

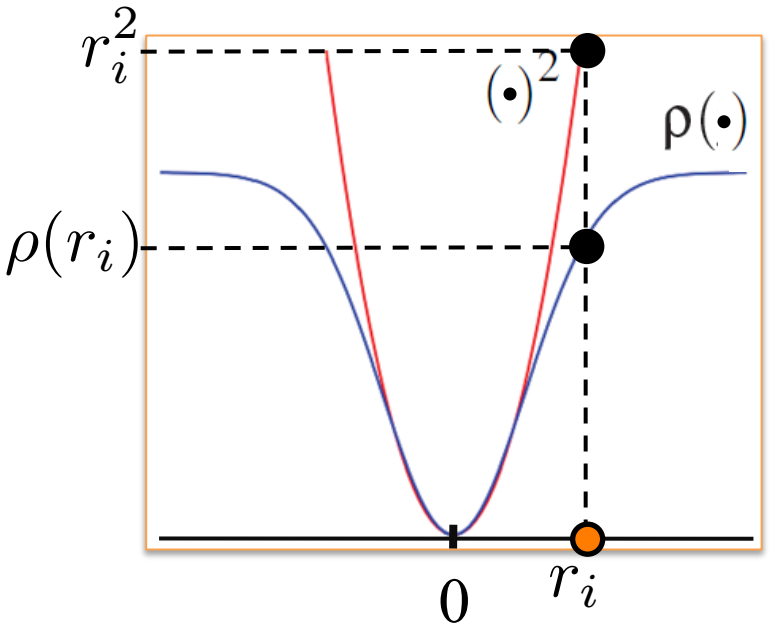


Robust Approximation

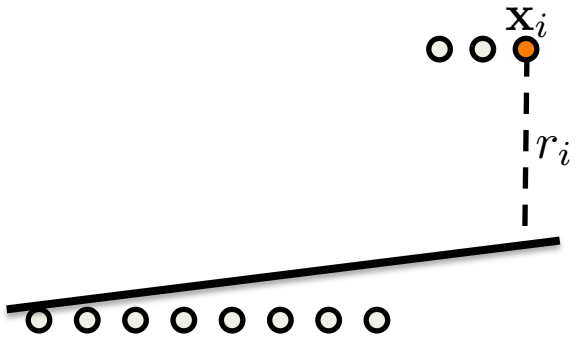
- Problem: squared norm is too sensitive to outliers

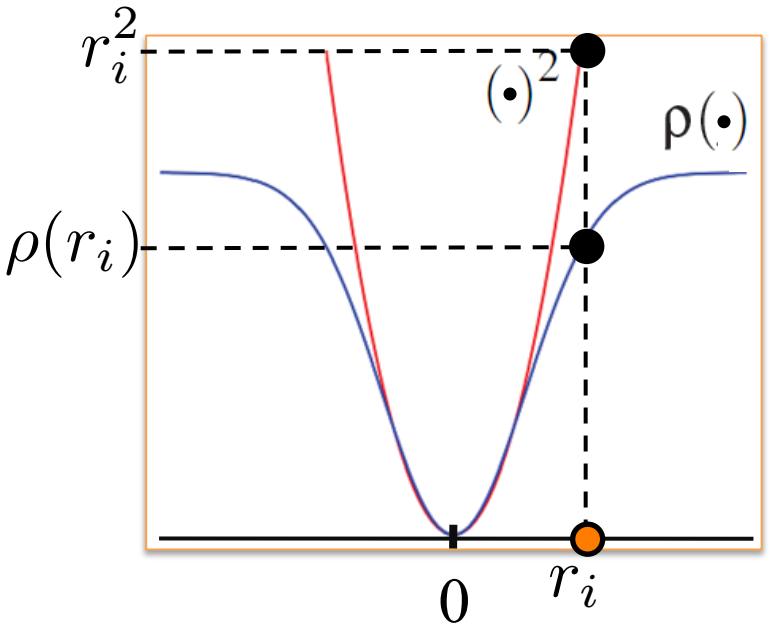


$$\min_{a,b} \sum_{i=1}^{n} \rho(\underbrace{ax_i + b - y_i}_{r_i})$$



Robust Approximation





$$\min_{a,b} \sum_i (r_i)^2 \qquad \min_{a,b} \sum_i \rho(r_i)$$

$$r_i = ax_i + b - y_i$$

Robust Approximation

- Solving the robust optimization

$$\min_{\mathbf{c}} E(\mathbf{c}) \qquad E(\mathbf{c}) = \sum_i \rho\left(\mathbf{b}(\mathbf{x}_i)^T\mathbf{c} - f_i\right)$$

$$r_i = \mathbf{b}(\mathbf{x}_i)^T\mathbf{c} - f_i$$

$$\frac{\partial E(\mathbf{c})}{\partial \mathbf{c}} = \frac{\partial \sum_i \rho(r_i)}{\partial \mathbf{c}} = \sum_i w(r_i) r_i \frac{\partial r_i}{\partial \mathbf{c}} \qquad w(x) = \frac{\frac{\partial \rho}{\partial x}}{x}$$

Robust Approximation

- Solving the robust optimization

$$\frac{\partial E(\mathbf{c})}{\partial \mathbf{c}} = \frac{\partial \sum_i \rho(r_i)}{\partial \mathbf{c}} = \sum_i w(r_i) r_i \frac{\partial r_i}{\partial \mathbf{c}} \qquad w(x) = \frac{\frac{\partial \rho}{\partial x}}{x}$$

Consider the following with fixed weights $w(r_i)$: $\sum_i w(r_i) r_i^2$

$$\frac{\partial \sum_i w(r_i) r_i^2}{\partial \mathbf{c}} = \sum_i w(r_i) r_i \frac{\partial r_i}{\partial \mathbf{c}}$$

Robust Approximation

- Solution: Iteratively reweighted least squares

Iterate

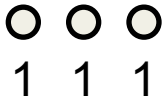
1. Compute weights $w(r_i)$ based on the current residuals
2. Solve the weighted least squares system $\sum_i w(r_i) r_i^2$
3. Update the residuals r_i

Initialize: $w(r_i) = 1$

Robust Approximation

- Iteratively reweighted least squares

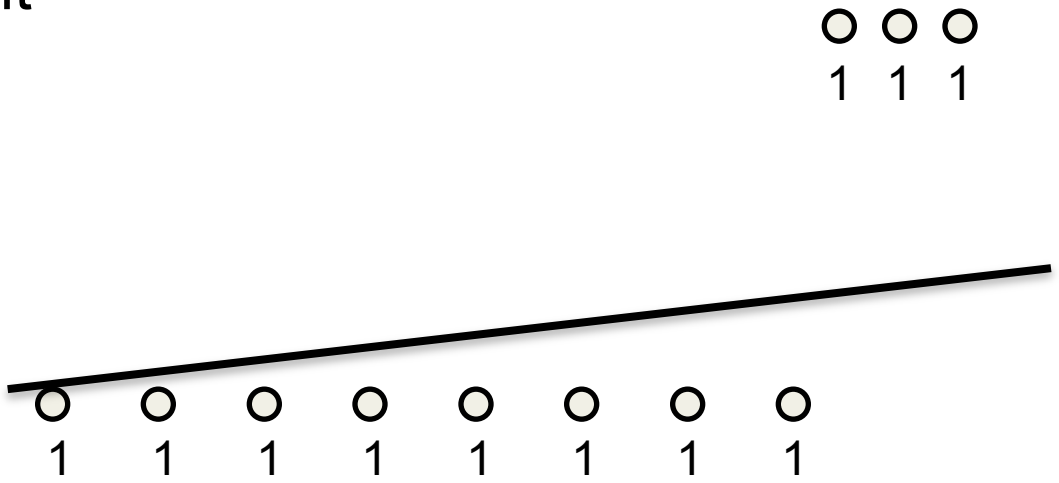
Initial weights



Robust Approximation

- Iteratively reweighted least squares

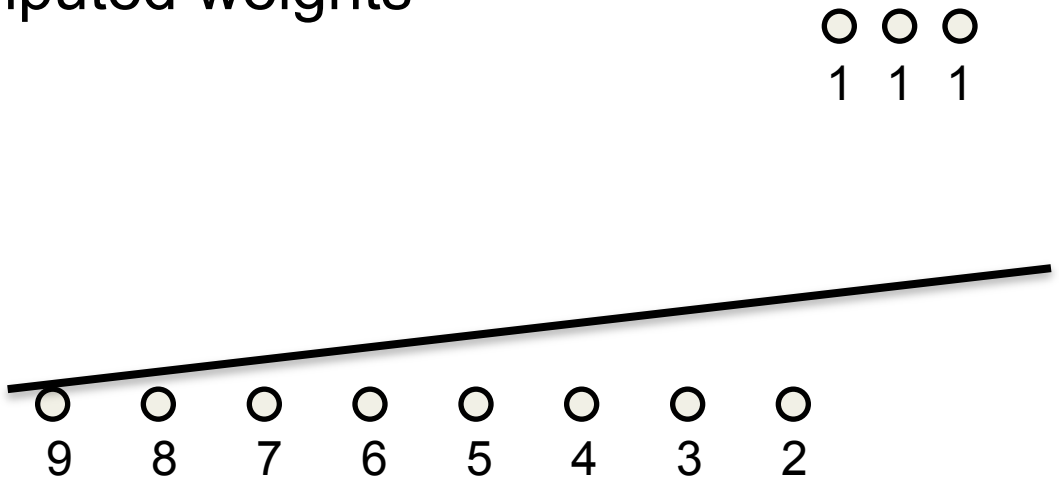
Initial fit



Robust Approximation

- Iteratively reweighted least squares

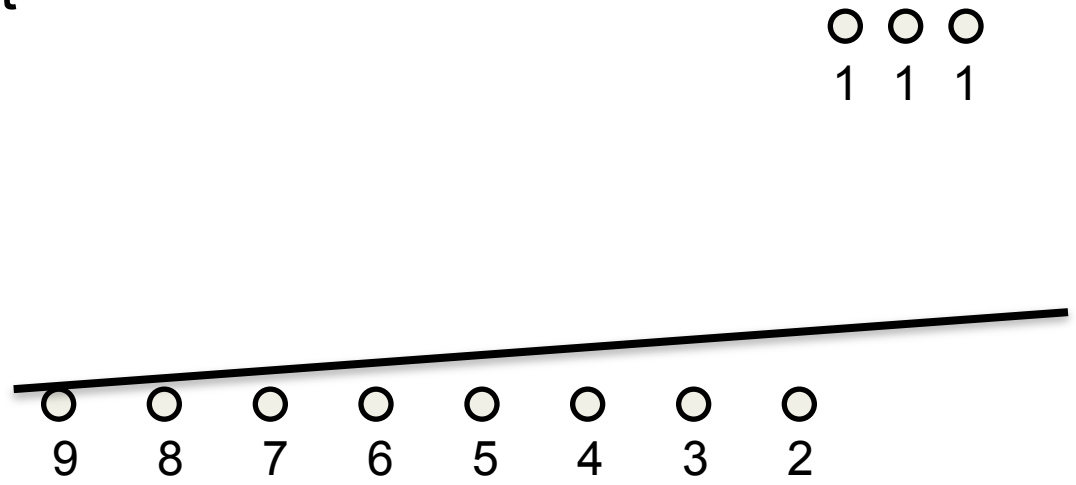
Recomputed weights



Robust Approximation

- Iteratively reweighted least squares

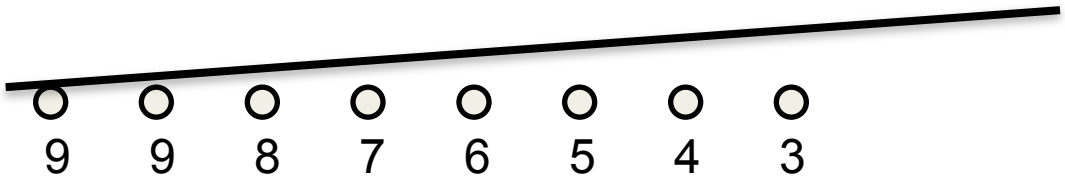
New fit



Robust Approximation

- Iteratively reweighted least squares

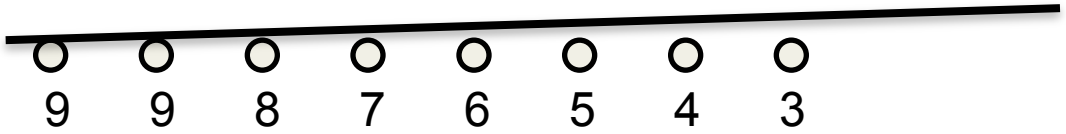
Recomputed weights



Robust Approximation

- Iteratively reweighted least squares

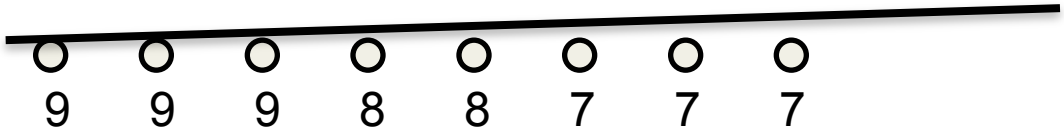
New fit



Robust Approximation

- Iteratively reweighted least squares

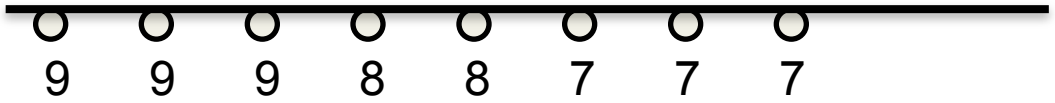
Recomputed weights



Robust Approximation

- Iteratively reweighted least squares

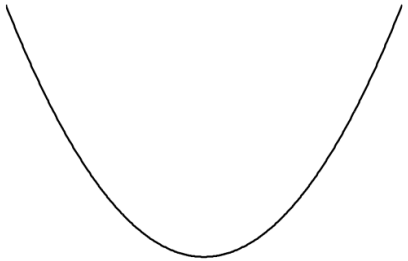
New fit



Robust Approximation

- Iteratively reweighted least squares – weights functions

Least squares

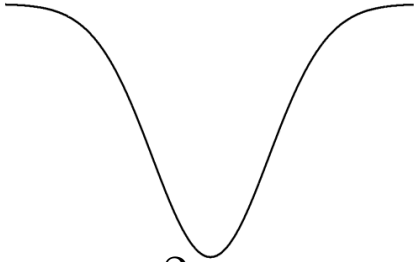


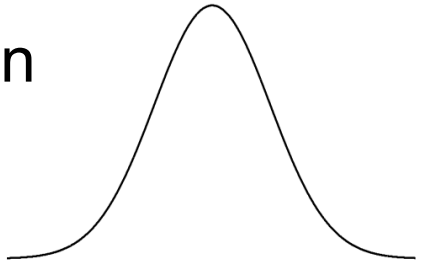
$$\rho(x) = x^2 \quad w(x) = \frac{\partial \rho(x)}{\partial x} / x = 2$$

Robust Approximation

- Iteratively reweighted least squares – weights functions

Robust fitting
Welsch's function





$$\rho(x) = \frac{h^2}{2}\left(1 - e^{-\left(\frac{x}{h}\right)^2}\right) \qquad w(x) = \frac{\partial \rho(x)}{\partial x} / x = e^{-\left(\frac{x}{h}\right)^2}$$

Robust Approximation

- Robust version of MLS

MLS

$$\mathbf{c}(\mathbf{x}) = \operatorname{argmin}_{\mathbf{c}} E_{\mathbf{x}}(\mathbf{c}) = \sum_i \phi_i(\mathbf{x}) \left(\mathbf{b}(\mathbf{x}_i)^T \mathbf{c} - f_i\right)^2$$

$$f(\mathbf{x}) = \mathbf{b}(\mathbf{x})^T \mathbf{c}(\mathbf{x})$$

Robust "MLS"

$$\mathbf{c}(\mathbf{x}) = \operatorname{argmin}_{\mathbf{c}} E_{\mathbf{x}}(\mathbf{c}) = \sum_i \phi_i(\mathbf{x}) \rho\left(\mathbf{b}(\mathbf{x}_i)^T \mathbf{c} - f_i\right)$$

$$f(\mathbf{x}) = \mathbf{b}(\mathbf{x})^T \mathbf{c}(\mathbf{x})$$

Robust MLS

- Example: Bilateral filtering of images

$$m = 0, d = 2$$

$$f(\mathbf{x}) = \frac{\sum_i y_i \phi_i(\mathbf{x}) w(f(\mathbf{x}) - y_i)}{\sum_i \phi_i(\mathbf{x}) w(f(\mathbf{x}) - y_i)}$$

$\mathbf{x}, \mathbf{x}_i$: pixel location

f, y_i : pixel color

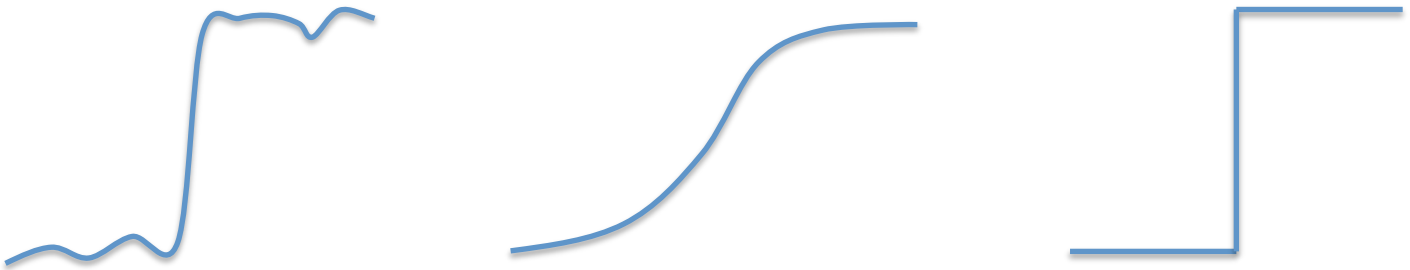
$$f(\mathbf{x}_k) = \frac{\sum_i y_i e^{-||\mathbf{x}-\mathbf{x}_i||^2/\sigma_1^2} e^{-(y_k-y_i)^2/\sigma_2^2}}{\sum_i e^{-||\mathbf{x}-\mathbf{x}_i||^2/\sigma_1^2} e^{-(y_k-y_i)^2/\sigma_2^2}}$$

Bilateral filtering for gray and color images, ICCV 1998
A Gentle Introduction to Bilateral Filtering and its Applications, SIGGRAPH 2008

Robust MLS

- Example: Bilateral filtering of images

$$f(\mathbf{x}_k) = \frac{\sum_i y_i e^{-||\mathbf{x}-\mathbf{x}_i||^2/\sigma_1^2} e^{-(y_k-y_i)^2/\sigma_2^2}}{\sum_i e^{-||\mathbf{x}-\mathbf{x}_i||^2/\sigma_1^2} e^{-(y_k-y_i)^2/\sigma_2^2}}$$



Input MLS Bilateral

Robust MLS

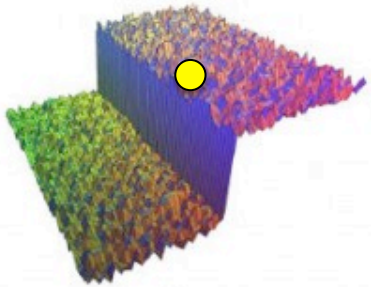
- Example: Bilateral filtering of images

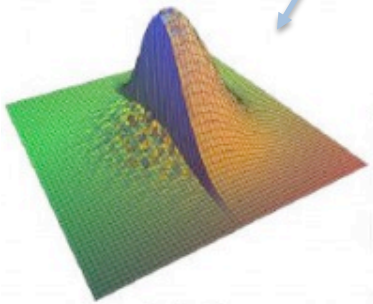
$$f(\mathbf{x}_k) = \frac{\sum_i y_i e^{-||\mathbf{x}-\mathbf{x}_i||^2/\sigma_1^2} e^{-(y_k-y_i)^2/\sigma_2^2}}{\sum_i e^{-||\mathbf{x}-\mathbf{x}_i||^2/\sigma_1^2} e^{-(y_k-y_i)^2/\sigma_2^2}}$$

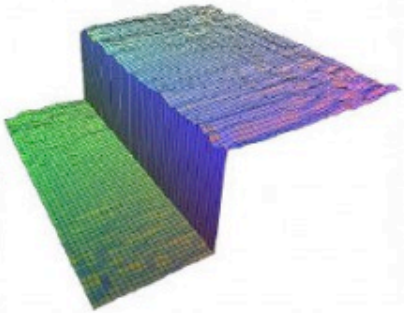
Robust MLS

- Example: Bilateral filtering of images

$$f(\mathbf{x}_k) = \frac{\sum_i y_i e^{-||\mathbf{x}-\mathbf{x}_i||^2/\sigma_1^2} e^{-(y_k-y_i)^2/\sigma_2^2}}{\sum_i e^{-||\mathbf{x}-\mathbf{x}_i||^2/\sigma_1^2} e^{-(y_k-y_i)^2/\sigma_2^2}}$$







Input

Output

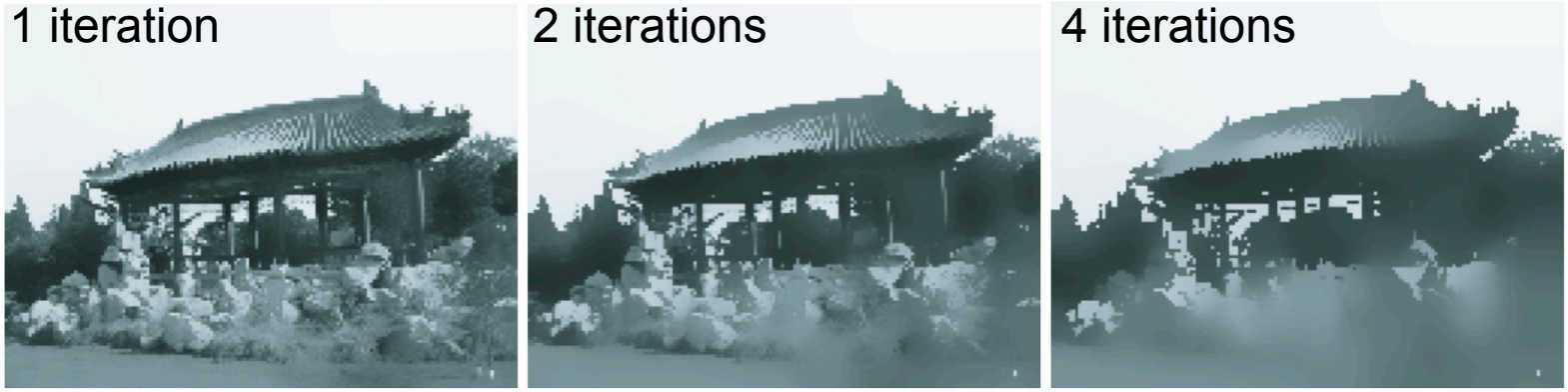
Robust MLS

- Example: Bilateral filtering of images

Robust MLS

- Example: Bilateral filtering of images

$$f(\mathbf{x}_k) = \frac{\sum_i y_i e^{-||\mathbf{x}-\mathbf{x}_i||^2/\sigma_1^2} e^{-(y_k-y_i)^2/\sigma_2^2}}{\sum_i e^{-||\mathbf{x}-\mathbf{x}_i||^2/\sigma_1^2} e^{-(y_k-y_i)^2/\sigma_2^2}}$$



Robust MLS

- Example: Bilateral filtering of images

$$f(\mathbf{x}_k) = \frac{\sum_i y_i e^{-||\mathbf{x}-\mathbf{x}_i||^2/\sigma_1^2} e^{-(y_k-y_i)^2/\sigma_2^2}}{\sum_i e^{-||\mathbf{x}-\mathbf{x}_i||^2/\sigma_1^2} e^{-(y_k-y_i)^2/\sigma_2^2}}$$

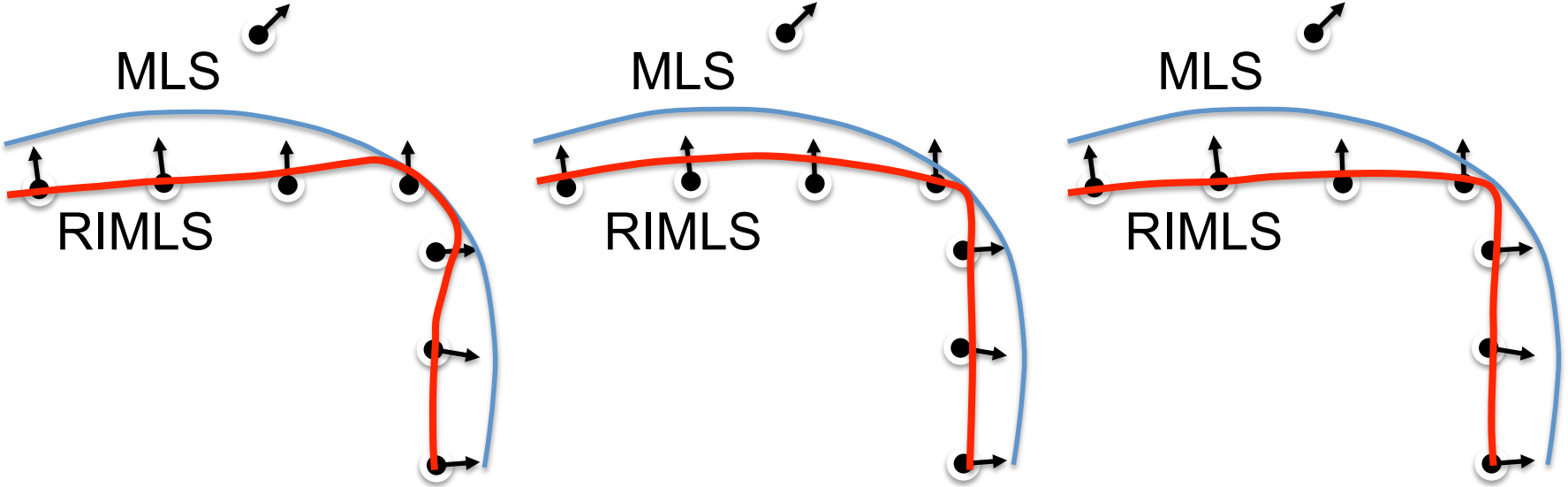
Can be interpreted as linear filtering in a higher dimensional space

$$e^{-||\mathbf{x}_k-\mathbf{x}_i||^2/\sigma_1^2-(y_k-y_i)^2/\sigma_2^2} = e^{-||\mathbf{p}_k-\mathbf{p}_i||^2}$$

$$\mathbf{p}_i = \begin{pmatrix} \mathbf{x}_i/\sigma_1 \\ y_i/\sigma_2 \end{pmatrix}$$

Robust MLS

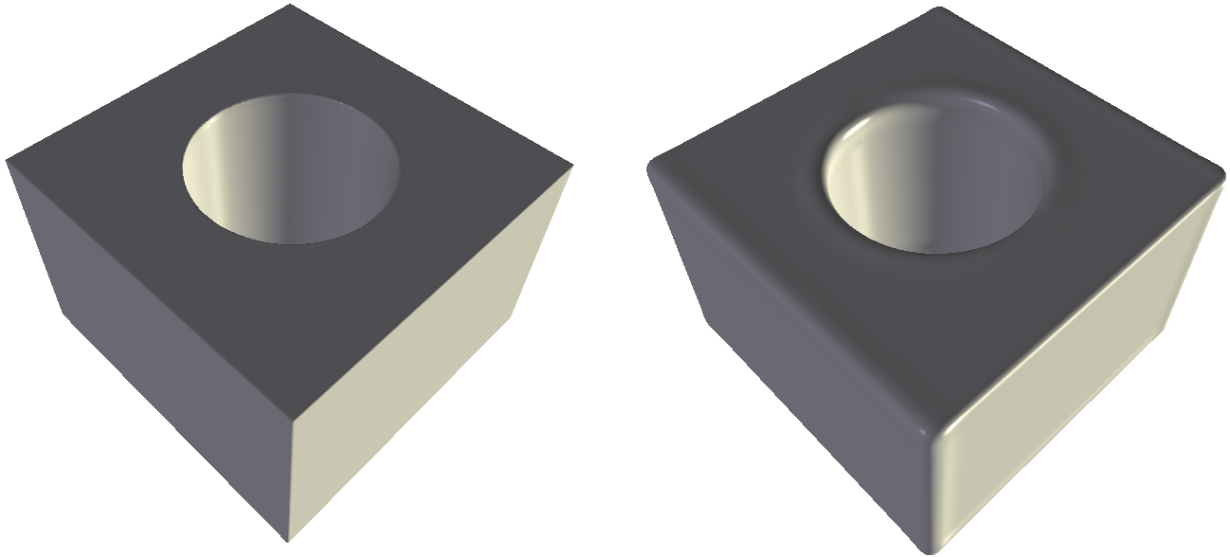
- Example: Sharp feature preserving surfaces



Spatial robustness Normal robustness Both terms

Robust MLS

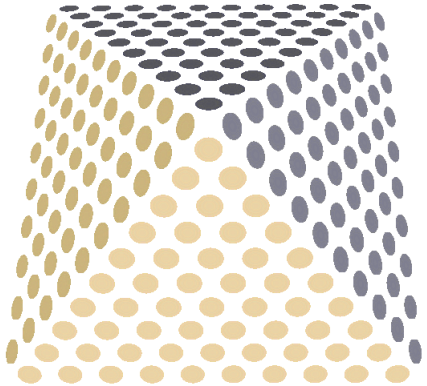
- Example: Sharp feature preserving surfaces



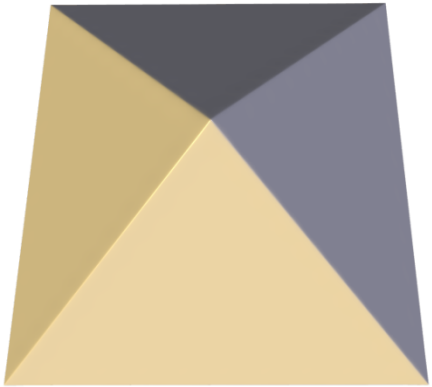
Robust Non-robust

Robust MLS

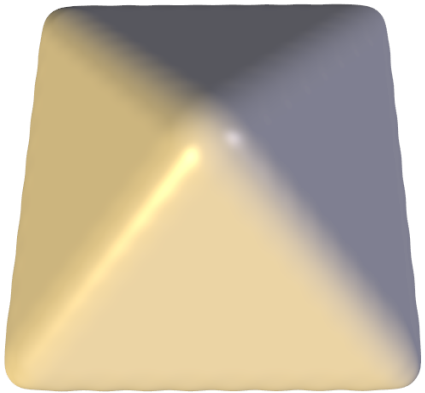
- Example: Sharp feature preserving surfaces



Input



Robust



Non-robust

Final remarks

- Approximation is everywhere
- Smoothly approximate with MLS
- Robust approximation to
  - Decrease the effect of outliers
  - Preserve important properties of the function