Meshless Surfaces

presented by Niloy J. Mitra

An Nguyen

Outline

- Mesh-Independent Surface Interpolation
  D. Levin

Outline

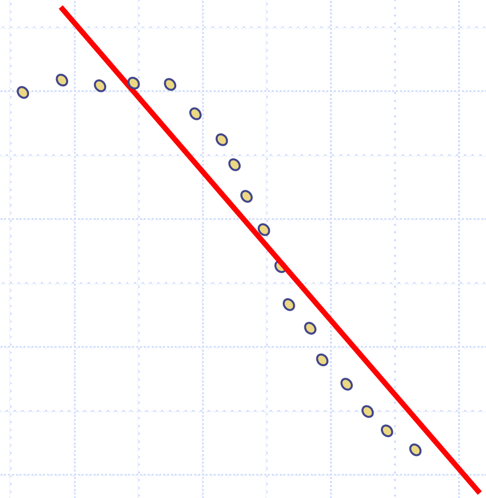
- Mesh-Independent Surface Interpolation
  D. Levin
- Point Set Surfaces
  M. Alexa, J. Behr, D. Cohen-Or, S. Fleishman, D. Levin, and C. T. Silva
- Progressive Point Set Surfaces
  S. Fleishman, D. Cohen-Or , M. Alexa, C. T. Silva

Outline

- Mesh-Independent Surface Interpolation
  D. Levin
- Point Set Surfaces
  M. Alexa, J. Behr, D. Cohen-Or, S. Fleishman, D. Levin, and C. T. Silva
- Progressive Point Set Surfaces
  S. Fleishman, D. Cohen-Or , M. Alexa, C. T. Silva
- Meshless Parametrization and Surface Reconstruction
  M. S. Floater, M. Reimers

Moving Least Square (MLS) Approximation

Fitting Functions



- Given: $\{\mathbf{x}_i, f_i\}$
- Goal: find p such that $\{\mathbf{x}_i, p(\mathbf{x}_i)\}$

$$\min_{p\in\Pi_m^{d-1}} \sum_i \underbrace{(p(\mathbf{x}_\mathbf{i}) - f_i)^2}_{\text{error}} \underbrace{\theta(\|\mathbf{x}_\mathbf{i}\|)}_{\text{weight}}$$

Motivation

- Given: PCD $\boldsymbol{R=\{r_i\}}$

Motivation

- Given: PCD $\boldsymbol{R=\{r_i\}}$

- Goal:
  - define a *projection operator*

$$\boldsymbol{P(P(x))=P(x)}$$

Motivation

- Given: PCD $\boldsymbol{R=\{r_i\}}$
- Goal:
  - define a *projection operator* ***P***
  - implies a unique manifold surface ***S***

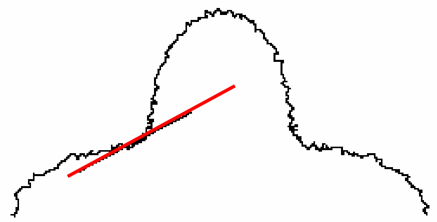
$$S \equiv \{x \mid P(x)=x\}$$

MLS Approach

- Step 1:

  define a *local* reference domain (something like a local tangent plane)

  gives a *local parameterization*



MLS Approach

- Step 1:
  define a local reference domain

- Step 2:
  construct a MLS *approximation*
  wrt a reference domain
  (a polynomial fitting step)

Notations so far

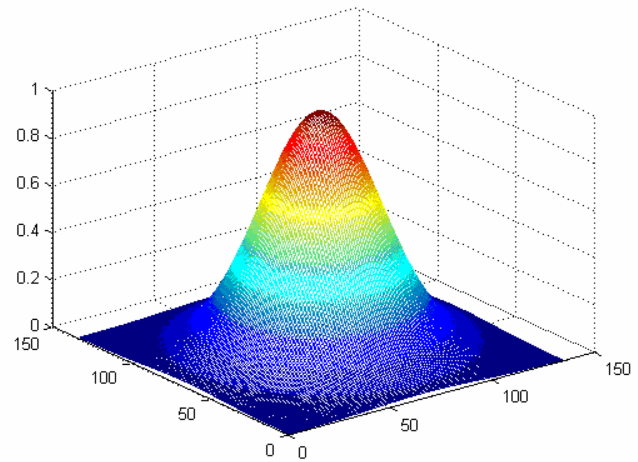
- ***$R=\{r_i\}$*** : input PCD
- ***S*** : (d-1)-dim manifold in $\Re^d$

Notations

- $R=\{r_i\}$: input PCD
- S : (d-1)-dim manifold in $\Re^d$
- $\| r_i - r \|$: Euclidean distance between r_i, r

θ : The Weight Function

- Non-negative decaying function
- Typical example
  - Gaussian $\theta(d) = \exp(-d^2 / h^2)$



Notations so far

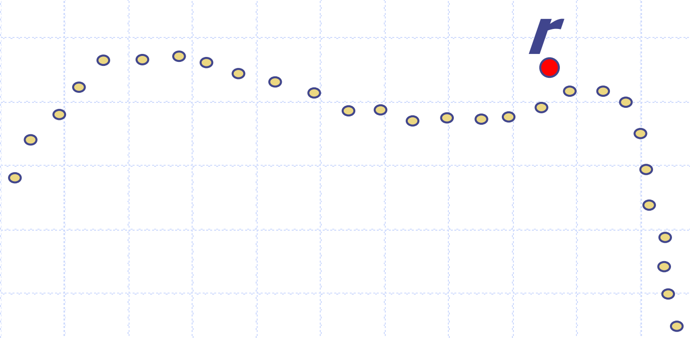
- $R=\{r_i\}$: input PCD
- S : (d-1)-dim manifold in $\Re^d$
- $\| r_i - r \|$: Euclidean distance between r_i, r
- θ : non-negative weight function

Basic MLS Procedure

- For a given point ***r*** near ***R***, *define* a local approximating hyper-plane

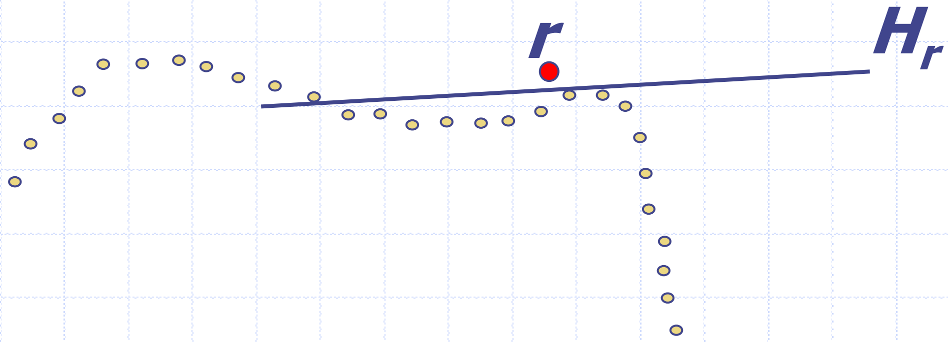
Basic MLS Procedure

- For a given point ***r*** near ***R***, *define* a local approximating hyper-plane



Basic MLS Procedure

- For a given point ***r*** near ***R***, *define* a local approximating hyper-plane

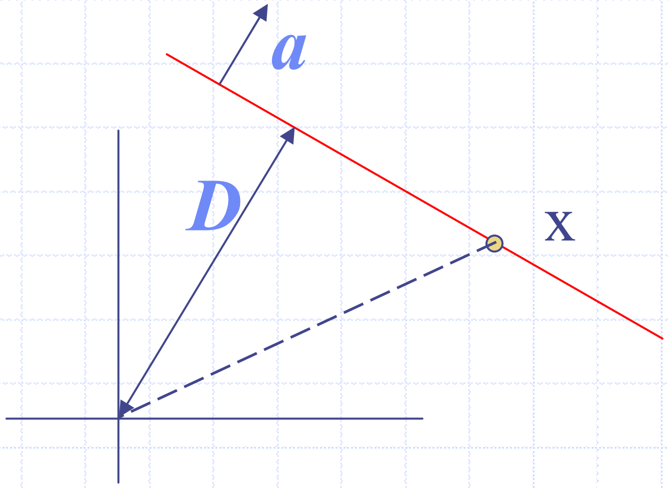


Notations so far

- $\boldsymbol{R=\{r_i\}}$: input PCD
- $\boldsymbol{S}$: (d-1)-dim manifold in $\Re^d$
- $\| r_i - r \|$: Euclidean distance between $\boldsymbol{r_i}$, $\boldsymbol{r}$
- θ : non-negative weight function
- $\boldsymbol{H_r}$: approximating hyper-plane at $\boldsymbol{r}$ using $\boldsymbol{R}$

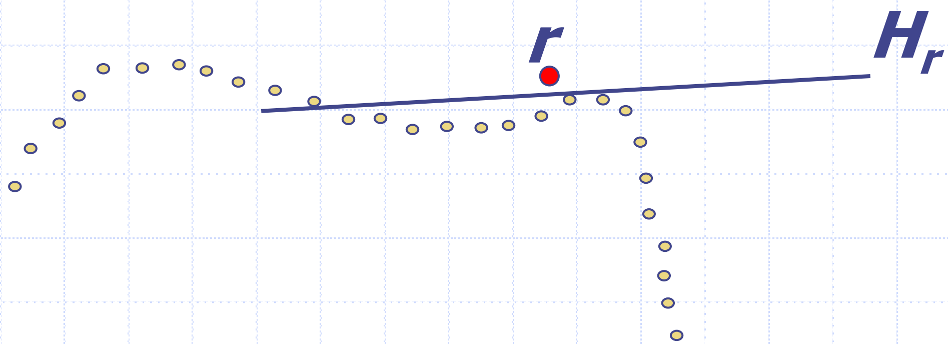
Equation of a Line

$$H = \{x \mid \langle a, x \rangle - D = 0, x \in \Re^d\}, a \in \Re^d, \|a\| = 1$$



Basic MLS Procedure : Step 1

- For a given point ***r*** near ***R***, define a local hyper-plane H_r
- Plane H_r defined by a least square formulation



Basic MLS Procedure : Step 1

- For a given point ***r*** near ***R***, define ***H*_r**

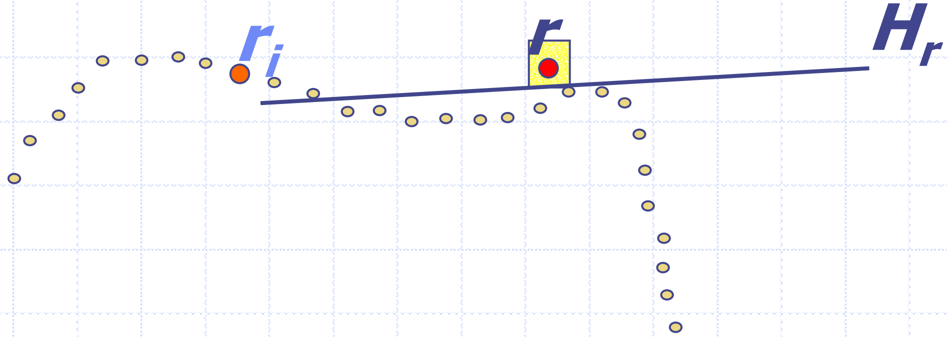
$$\min_{a,D} \sum_i (\langle a, r_i \rangle - D)^2 \theta(\| r_i - r \|)$$

- In case of multiple local minima, the plane closest to ***r*** is chosen

Basic MLS Procedure : Step 1

- For a given point ***r*** near ***R***, define H_r

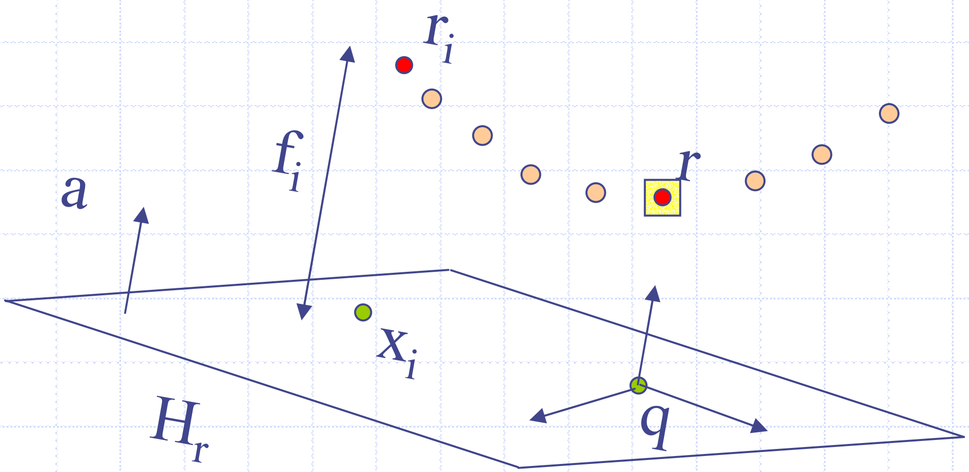
- $$\min_{a,D} \sum_i (\langle a, r_i \rangle - D)^2 \theta(\| r_i - r \|)$$



Basic MLS Procedure : Step 2

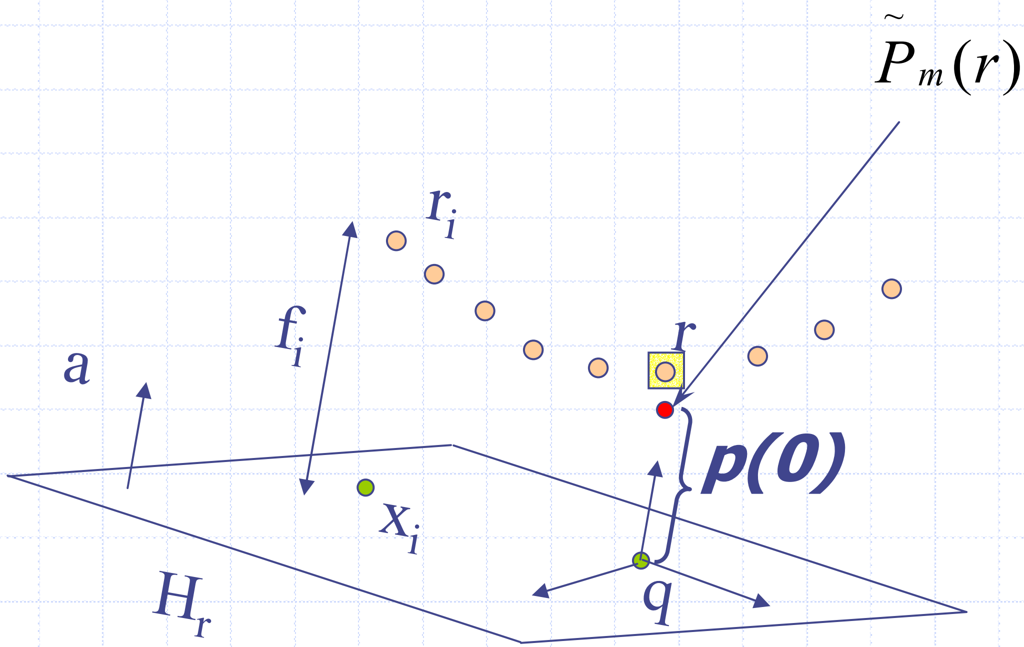
- ***H_r*** : local approximating plane
- Find a polynomial approx. of degree ***m***

$$\min_{p \in \Pi_m^{d-1}} \sum_i (p(x_i) - f_i)^2 \theta(\| r_i - r \|)$$

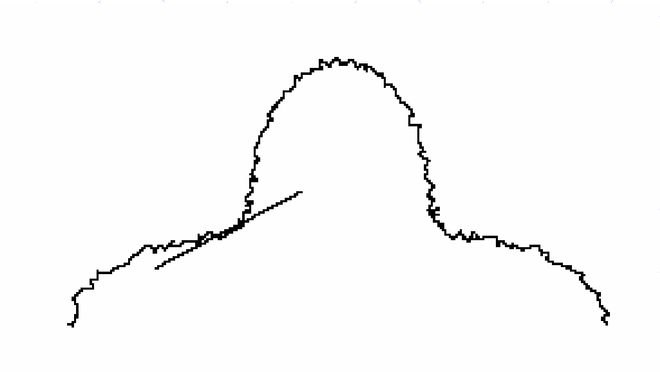


Basic MLS Procedure : Step 2

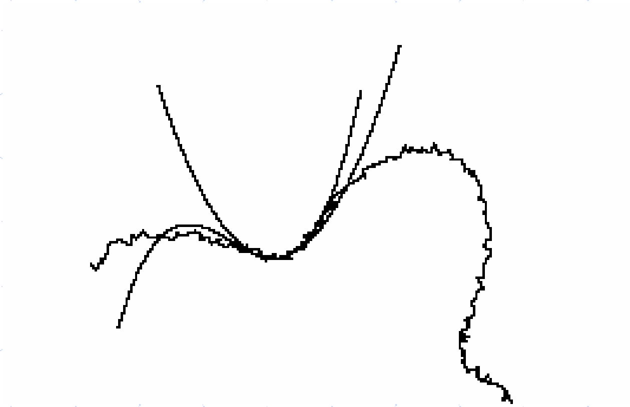
$$\min_{p \in \Pi_m^{d-1}} \sum_i (p(x_i) - f_i)^2 \theta(\| r_i - r \|)$$



Basic MLS : Curve Smoothing



Step 1

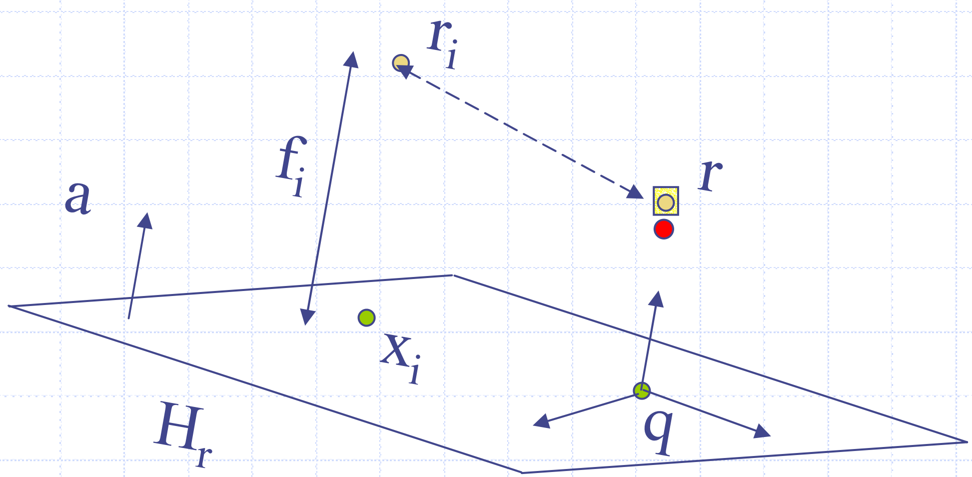


Step 2

But it’s **not** a projection

$$\tilde{P}_m(\tilde{P}_m(r)) \neq \tilde{P}_m(r)$$

Remember, $\theta(\| r_i - r \|)$

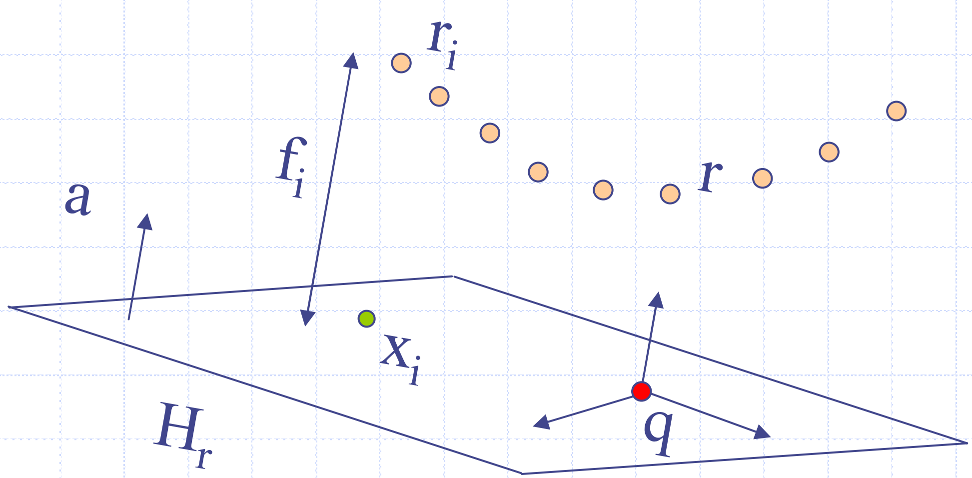


Problems with $\tilde{P}_m(r)$

- $\tilde{P}_m(\tilde{P}_m(r)) \neq \tilde{P}_m(r)$
- Basic MLS in $\Re^d$
  - Doesn't project points to a (d-1)-dim manifold
  - Doesn't define a surface

A Simple Fix

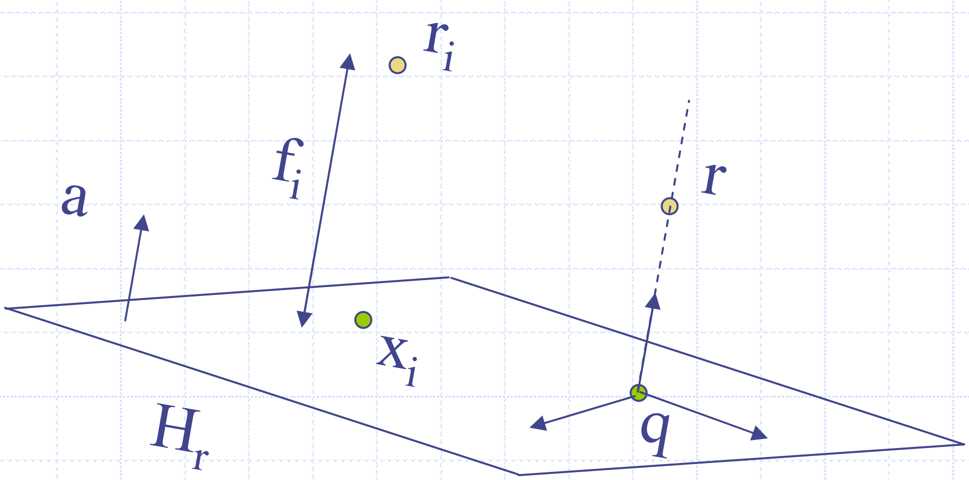
Replace $\theta(\| r_i - r \|)$ by $\theta(\| r_i - q \|)$



MLS Procedure : Step 1

- For a given point ***r*** near ***R***, define H_r

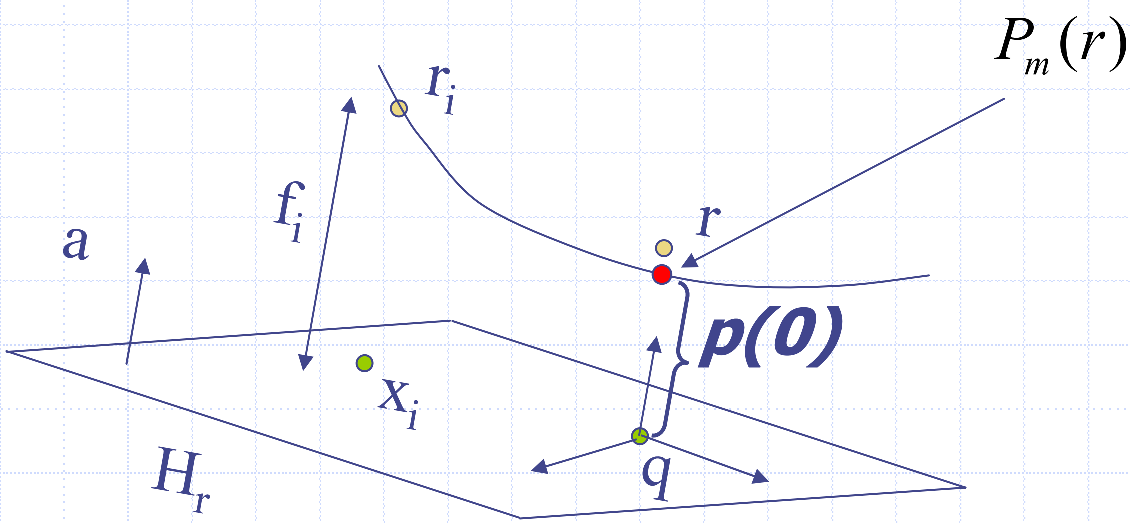
$$\min_{a,D} \sum_i (\langle a, r_i \rangle - D)^2 \theta(\| r_i - q \|)$$



MLS Projection : Step 2

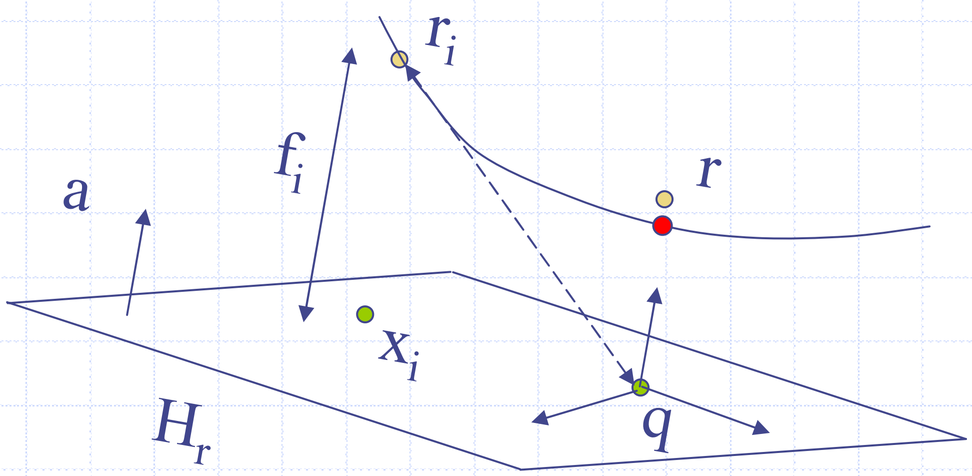
Given a local parameterization, ***H_r***

$$\min_{p \in \Pi_m^{d-1}} \sum_i (p(x_i) - f_i)^2 \theta(\| r_i - q \|)$$



Why does it work?

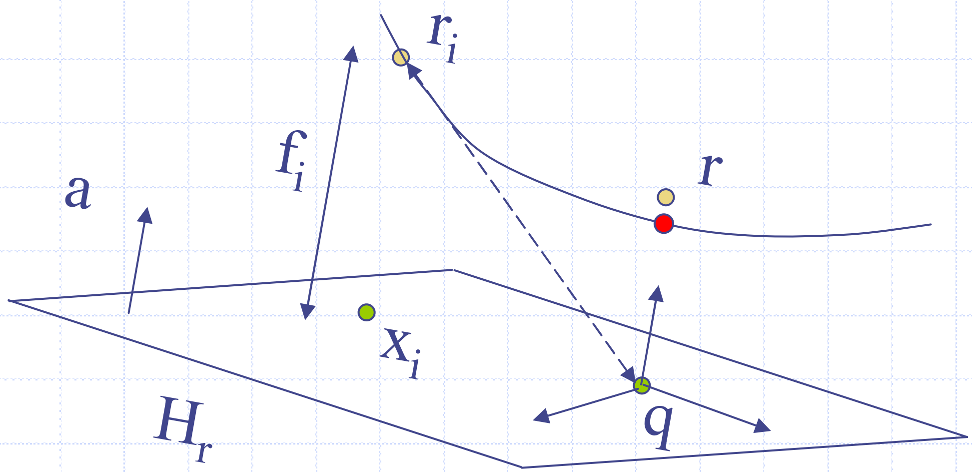
- $\theta(\| r_i - q \|)$ depends on the projection of ***r*** on $\boldsymbol{H_r}$



Why does it work?

- $\theta(\| r_i - q \|)$ depends on the projection of ***r*** on ***H$_r$***

$$\min_{a,D} \sum_i (< a, r_i > - D)^2 \theta(\| r_i - q \|)$$



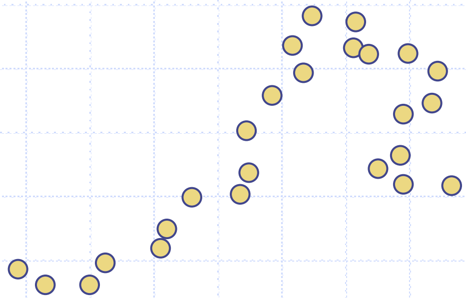
MLS defines (d-1)-dim manifold

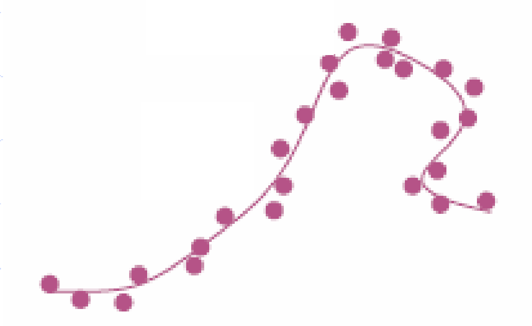
- $P_m(P_m(x))=P_m(x)$
- Implicit surface definition

$$S \equiv \{x \mid P_m(x) = x\}$$

- Conjecture:
  S infinitely smooth if $\theta \in C^\infty$

MLS defines Curve/Surface





Computing H_r and p(.)

- Computing hyper-plane $\boldsymbol{H_r}$
  - Non-linear optimization problem
  - Computed iteratively
  - Computing $\theta(.)$: time consuming step
    - O(N) for each iteration step
    - Approximate by doing a hierarchical clustering
- Fitting a polynomial ***p(.)***, given $\boldsymbol{H_r}$
  - Solve a linear system
    - Size depends on the order of approximation (***m***)

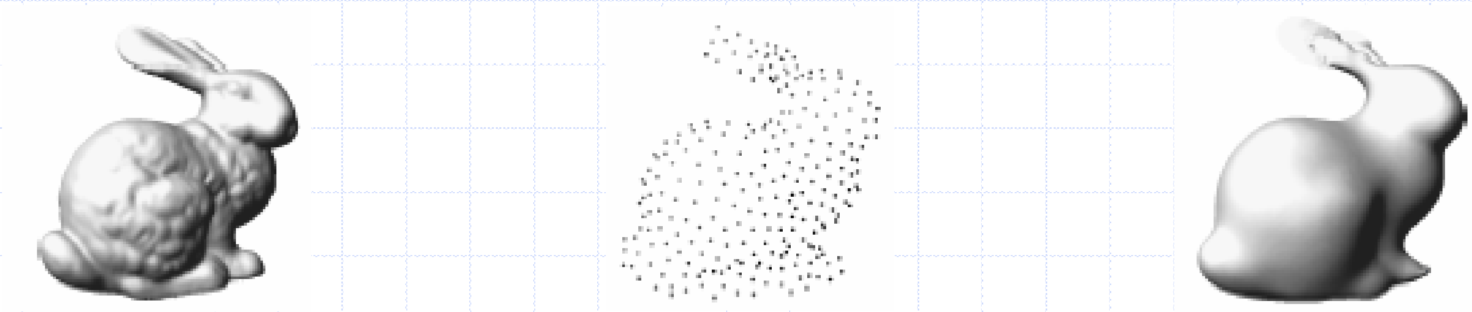
Weight Function θ

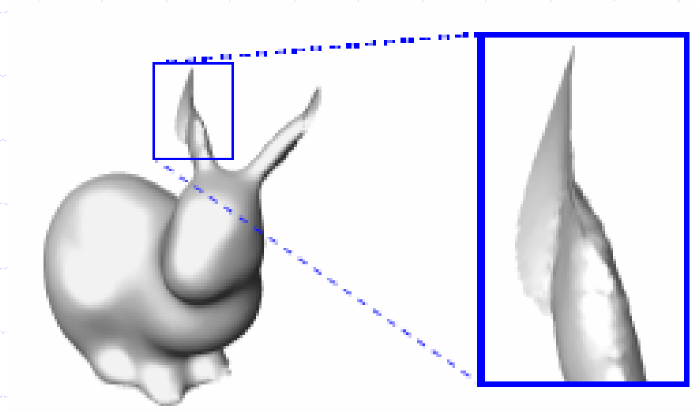
$$\theta(d) = \exp(-d^2 / h^2)$$

small h

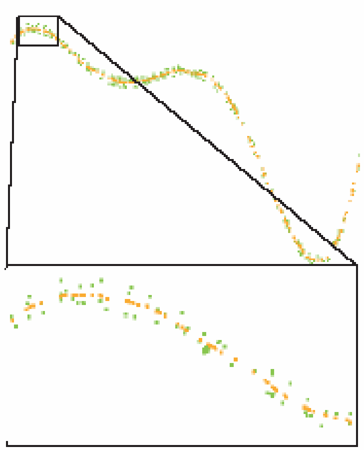
large h

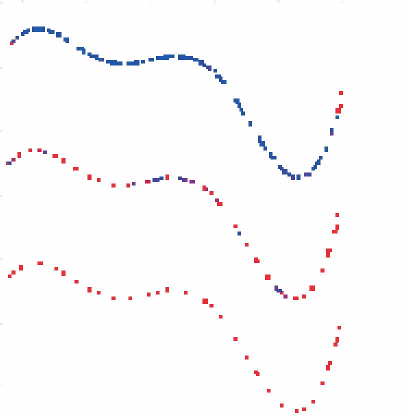
Sampling Condition?





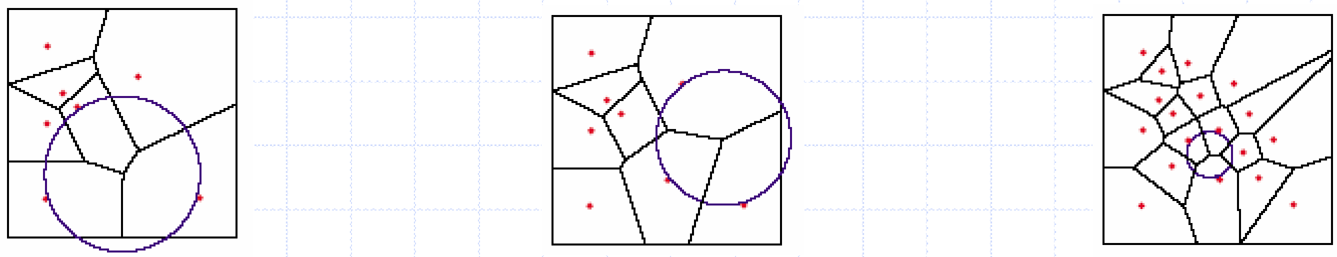
Simplification

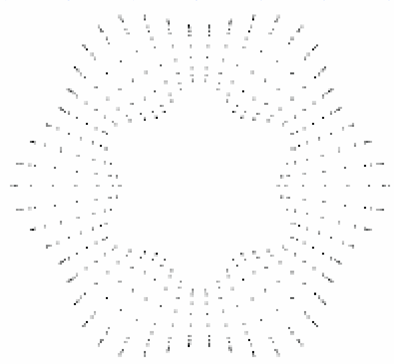




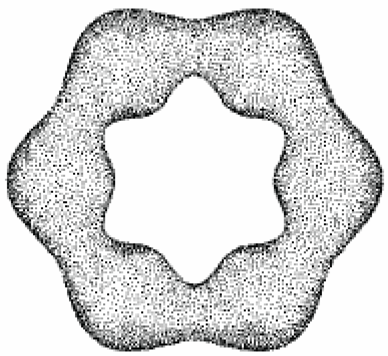
More simplification algorithms/results on Nov. 19

Up-sampling





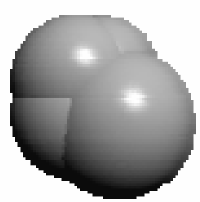
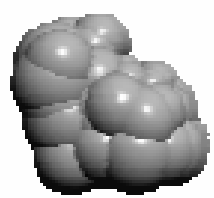
800

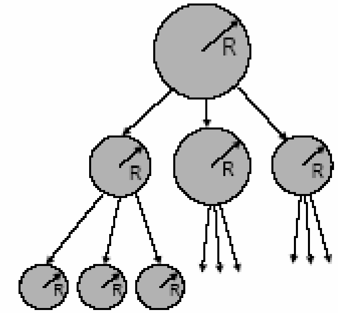


20,000

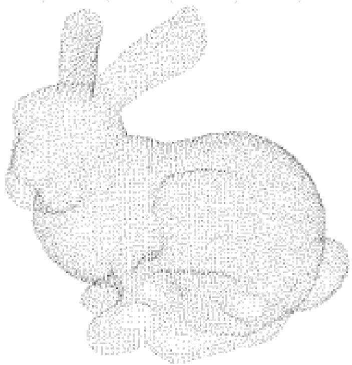
Rendering

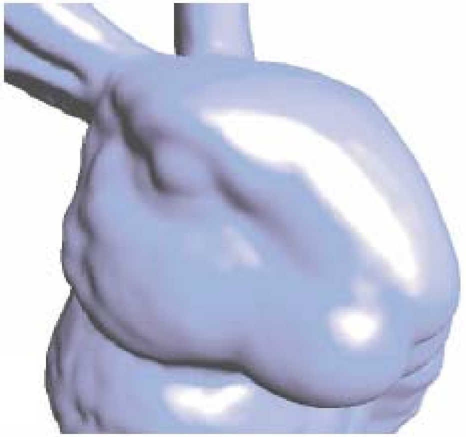
- Q-Splat
  - Rusinkiewicz et al., Siggraph 2000
  - hierarchical point rendering
  - based on **Bounding Sphere Hierarchy**



Rendering Results





Progressive Point Surface

- Extension of ***progressive mesh***
- Start with base domain $\boldsymbol{P}_0$
- Add points implicitly to get $\boldsymbol{P}_1$
- Predict positions and transmit error (Δ)
- Repeat

Progressive Point Surface

- Given point set $\boldsymbol{R=\{r_i\}}$
- Start with base point set $\boldsymbol{P_0}$
  - Get this by clustering method (Nov. 19)

Progressive Point Surface

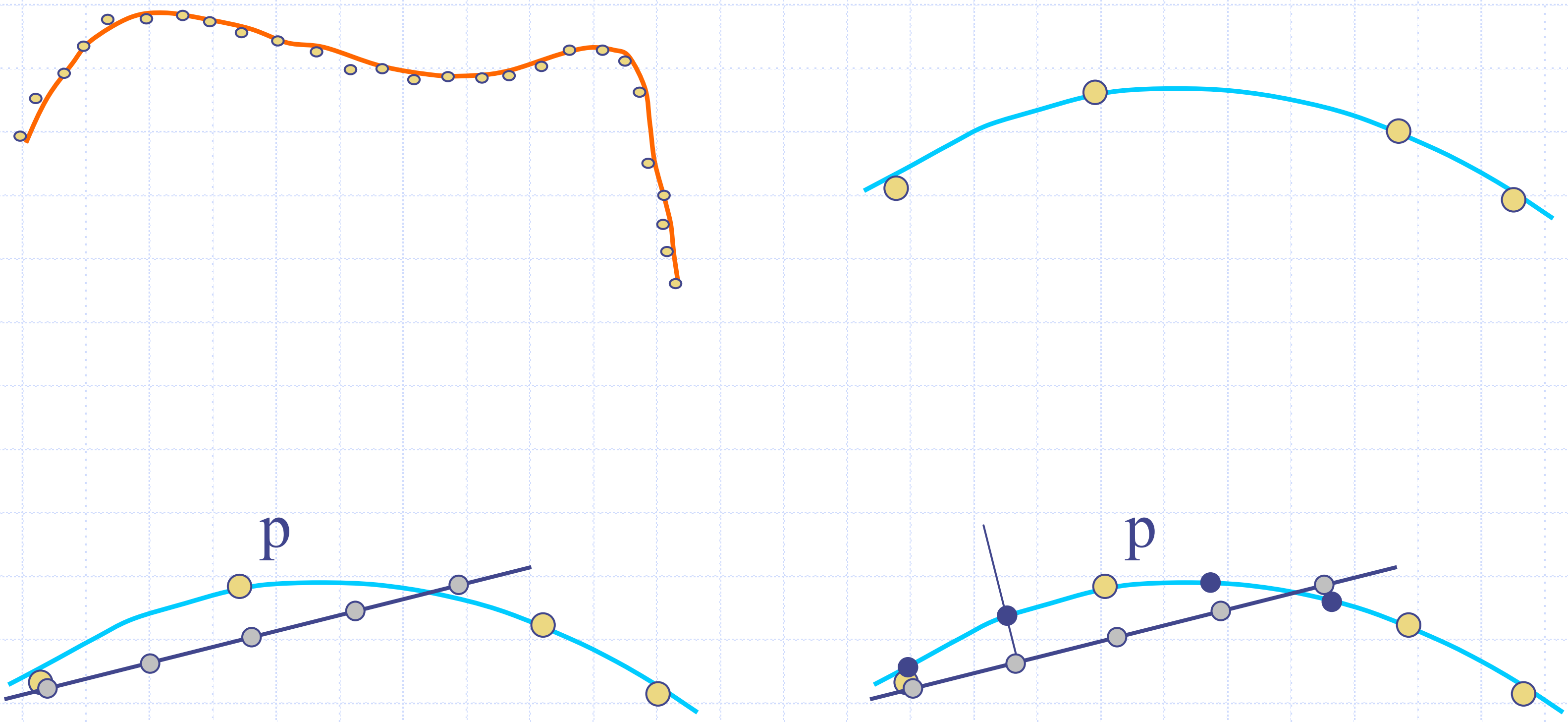
- Given point set $R=\{r_i\}$
- Start with base point set P_0
  - Get this by clustering method (Nov. 19)
- repeat
  - Refine P_i to get P_{i+1}

Progressive Point Surface

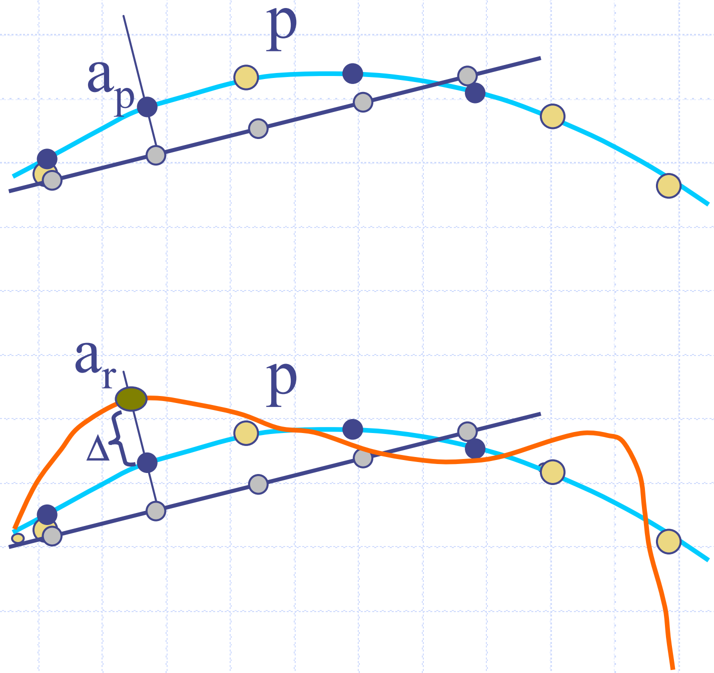
- Given point set $R=\{r_i\}$
- Start with base point set P_0
  - Get this by clustering method (Nov. 19)
- repeat
  - Refine P_i to get P_{i+1}
  - Displace new points s.t.

$$d(S_{P_{i+1}}, S_R) < d(S_{P_i}, S_R)$$

Refining P_i

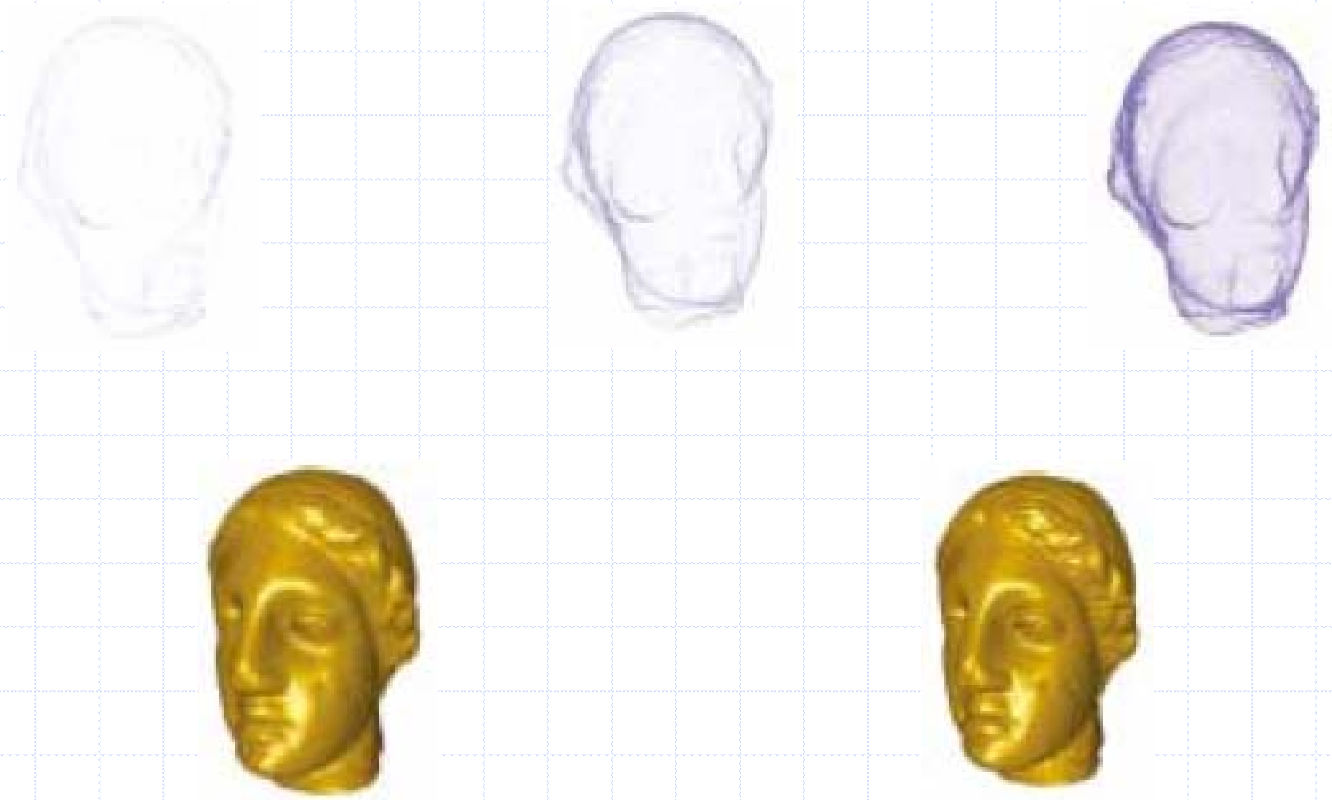


Computing Δ



- Transmit P_0, $\{\Delta_1\}$, $\{\Delta_2\}$, $\{\Delta_3\}$, …
- Do inverse process to reconstruct

Results



Conclusions

- MLS: Projection-based *surface definition*
- Surface is *smooth* and a *manifold*
- Surface may be bounded
- *Representation error* depends on point density
- Adjustable feature size ***h*** allows to smooth out noise

Adaptive MLS