3D Scattered Data Approximation with Adaptive Compactly Supported RBFs

Diego Salume

Authors: Ohtake, Belyaev, Seidel.

Overview

- Input: A set of points scattered over a piecewise smooth surface with oriented normals.
- Output: An implicit function whose zero level set is an approximation of the surface (robust to noise).
- Use compactly supported radial basis functions (RBFs) centered at randomly chosen points in the set.
- Support size for each RBF is adapted depending surface geometry around that RBF center.
- Takes into account confidence values for each sample point.

RBFs

- Given a set of N points p_i with normals n_i and confidence values v_i .
- Want to find implicit function $y=f(x)$, such that its zero level set approximates points p_i .
- Given M approximation centers c_i, such that M < N construct

$$\begin{aligned} f(x) &= \sum_{c_i \in \mathcal{C}} [g_i(x) + \lambda_i] \phi_{\sigma_i}(||x - c_i||) \\ &= \sum_{c_i \in \mathcal{C}} g_i(x) \phi_{\sigma_i}(||x - c_i||) + \sum_{c_i \in \mathcal{C}} \lambda_i \phi_{\sigma_i}(||x - c_i||) \end{aligned}$$

Base approximation

Local details

$$\phi_\sigma(r) = \phi\left(\frac{r}{\sigma}\right)$$

$$\phi(r) = (1 - r)_+^4 (4r + 1)$$

$$f(x) = \sum_{c_i \in \mathcal{C}} [g_i(x) + \lambda_i]\phi_{\sigma_i}(||x - c_i||)$$
$$= \sum_{c_i \in \mathcal{C}} g_i(x)\phi_{\sigma_i}(||x - c_i||) + \sum_{c_i \in \mathcal{C}} \lambda_i\phi_{\sigma_i}(||x - c_i||)$$

$$\phi_\sigma(r) = \phi\left(\frac{r}{\sigma}\right)$$
$$\phi(r) = (1 - r)_+^4(4r + 1)$$

RBFs

- Our unknowns are $g_i(x)$ and λ_i.
  - g is a local quadratic estimation of our points in $\{||x - c_i|| < \sigma_i\}$ (region of influence)
  - Coefficients λ_i are determined from M interpolation conditions

$$f(c_i) = 0, i = 1, ..., M$$

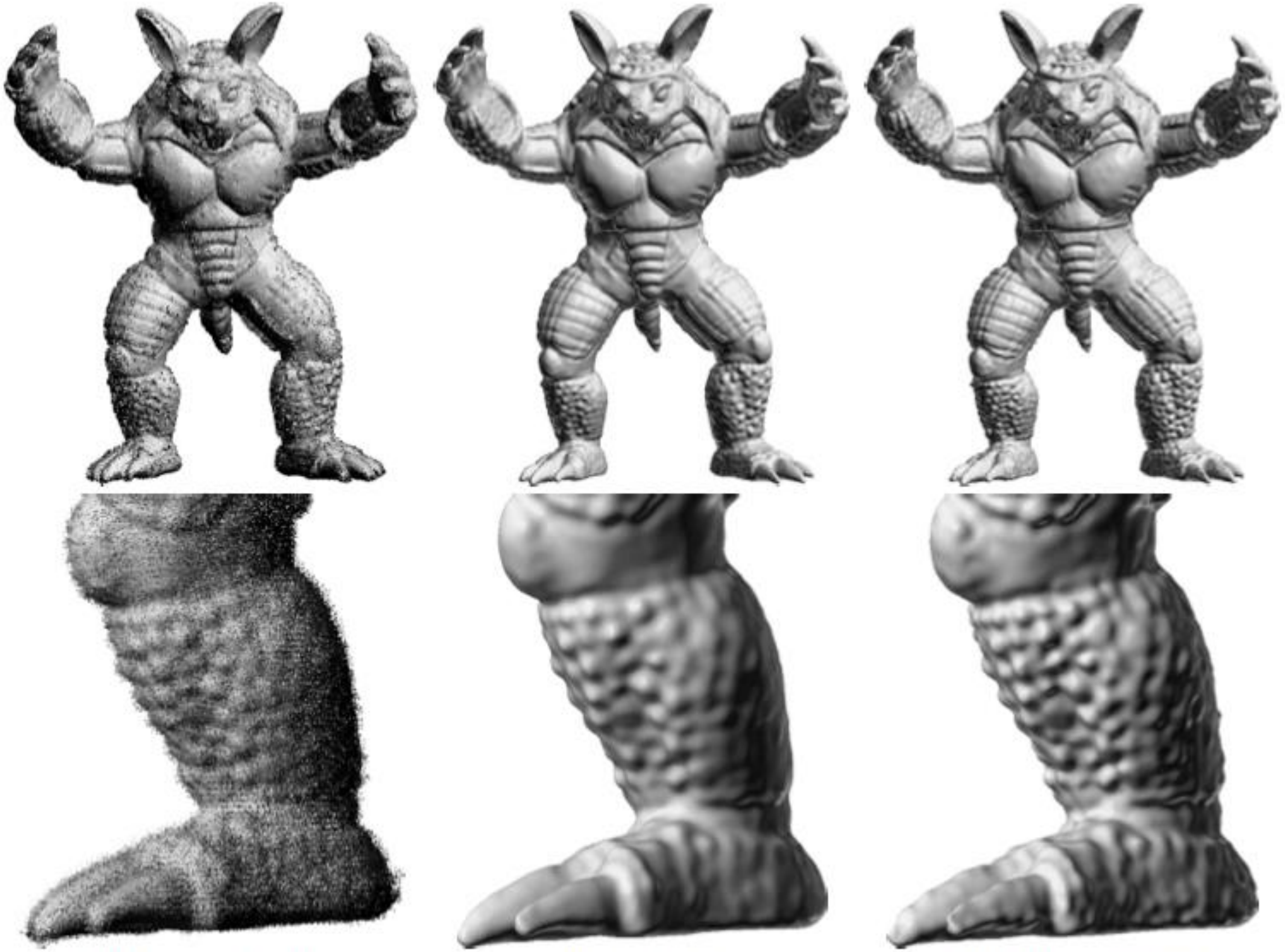
  - Base approximation term has the same zero level set as partition of unity approximations that use normalized RBFs, so we rewrite *f*

$$f(x) = \sum_{c_i \in \mathcal{C}} g_i(x)\Phi_{\sigma_i}(||x - c_i||) + \sum_{c_i \in \mathcal{C}} \lambda_i\Phi_{\sigma_i}(||x - c_i||)$$

Adaptive PU

Normalized RBF (refine PU approx)

$$\Phi_{\sigma_i}(||x - c_i||) = \frac{\phi_{\sigma_i}(||x - c_i||)}{\sum_j \phi_{\sigma_j}(||x - c_j||)}$$



Initial noisy point dataset | PU approximation | PU + RBF approximation

$$f(x) = \sum_{c_i \in \mathcal{C}} g_i(x)\Phi_{\sigma_i}(||x - c_i||)$$

$$f(x) = \sum_{c_i \in \mathcal{C}} g_i(x)\Phi_{\sigma_i}(||x - c_i||) + \sum_{c_i \in \mathcal{C}} \lambda_i \Phi_{\sigma_i}(||x - c_i||)$$

Algorithm Outline

- Adaptive PU Approximation
  - Select approximation centers c_i
  - Assign influence parameters $\sigma_{i.}$
  - Estimate function g_i.
- Least Squares RBF Approximation
  - Find coefficients λ_i

$$
\begin{aligned}
f(x) &= \sum_{c_i \in \mathcal{C}} [g_i(x) + \lambda_i] \phi_{\sigma_i}(||x - c_i||) \\
&= \sum_{c_i \in \mathcal{C}} g_i(x) \phi_{\sigma_i}(||x - c_i||) + \sum_{c_i \in \mathcal{C}} \lambda_i \phi_{\sigma_i}(||x - c_i||)
\end{aligned}
$$

Algorithm Outline

- Adaptive PU Approximation
  - Select approximation centers c_i
  - Assign influence parameters $\sigma_{i.}$
  - Estimate function g_i.
- Least Squares RBF Approximation
  - Find coefficients λ_i

$$
\begin{aligned}
f(x) &= \sum_{c_i \in \mathcal{C}} [g_i(x) + \lambda_i]\phi_{\sigma_i}(||x - c_i||) \\
&= \sum_{c_i \in \mathcal{C}} g_i(x)\phi_{\sigma_i}(||x - c_i||) + \sum_{c_i \in \mathcal{C}} \lambda_i\phi_{\sigma_i}(||x - c_i||)
\end{aligned}
$$

Adaptive PU Approximation: Finding Local Quadratic Estimate *g*

- To account for density irregularities, weight each point by the distance to its neighbors.

$$d_i = v_i \sum_{j=1}^{K} ||p_i - p_j||^2$$

- Define local coordinate system (u,v,w) at center c_i such that the positive w axis is the weighted average of the normals of the center's σ-neighborhood

$$\sum_j d_j \phi_\sigma(||p_j - c_i||) n_j$$

- Define local fitting function *h*

$$w = h(u, v) \equiv Au^2 + 2Buv + Cv^2 + Du + Ev + F$$

- Found by minimizing $\sum d_j \phi_\sigma(||p_j - c_i||) g_i(p_j)^2 \rightarrow min$

$$g_i(x) = w - h(u, v), \text{ where } (u, v, w) \text{ are local coordinates of } x$$

Algorithm Outline

- Adaptive PU Approximation
  - Select approximation centers c_i
  - Assign influence parameters σ_i.
  - Estimate function g_i.
- Least Squares RBF Approximation
  - Find coefficients λ_i

$$
\begin{aligned}
f(x) &= \sum_{c_i \in \mathcal{C}} [g_i(x) + \lambda_i] \phi_{\sigma_i}(||x - c_i||) \\
&= \sum_{c_i \in \mathcal{C}} g_i(x) \phi_{\sigma_i}(||x - c_i||) + \sum_{c_i \in \mathcal{C}} \lambda_i \phi_{\sigma_i}(||x - c_i||)
\end{aligned}
$$

Adaptive PU Approximation: Finding Influence Parameters σ

- To find optimal influence parameter σ_i associated with a center c_i , we define error function

$$E_{\text{local}}(\sigma) = \frac{1}{L}\sqrt{\frac{\sum_j d_j \phi_\sigma(\|\mathbf{p}_j - \mathbf{c}_i\|)\left(\frac{g_i(\mathbf{p}_j)}{\|\nabla g_i(\mathbf{p}_j)\|}\right)^2}{\sum_j d_j \phi_\sigma(\|\mathbf{p}_j - \mathbf{c}_i\|)}}$$

L: main diagonal of the bounding box of sample points
1/L: for scale-independence

- Assume this function is monotonically decreasing to zero as σ approaches zero.
- Use Rissanen's minimum description length (MDL) principle:

> *"From several alternative models, the best one gives the minimum length of combined description of the model and the residuals"*

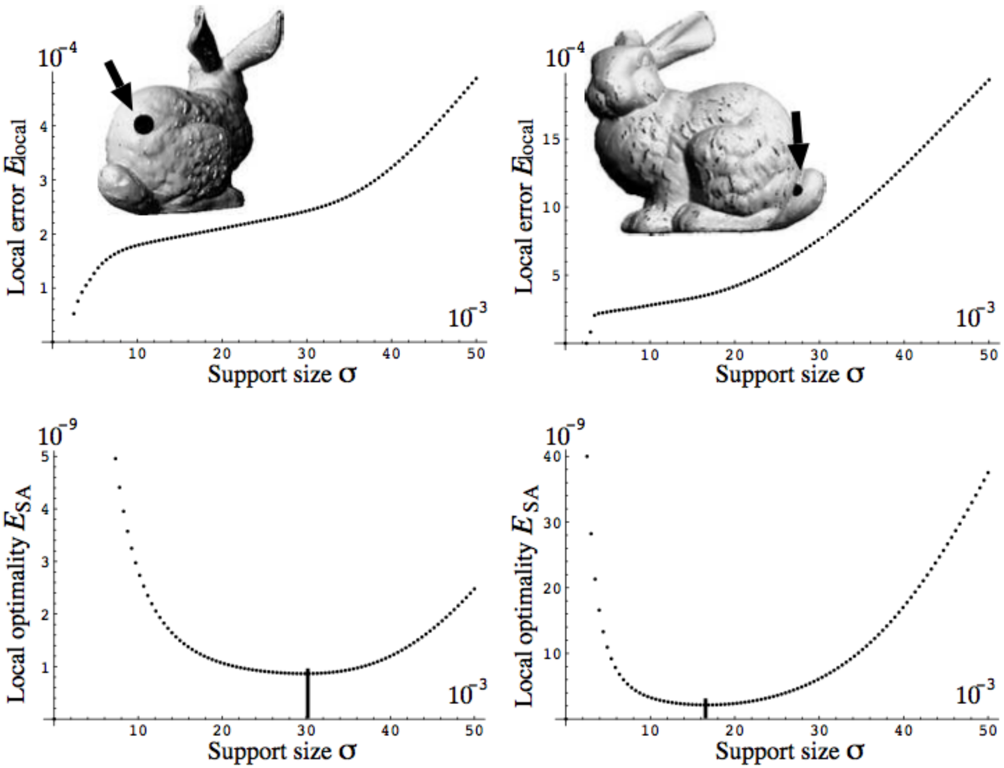
Adaptive PU Approximation: Finding Influence Parameters σ

- Find the approximation of a noisy signal that is the linear combination of the smallest number of approximants in a given collection.
- The distance from p to $g_{i(x)} = 0$ is approx $\frac{g(p)}{||\nabla g(p)||}$
- $E_{local}(\sigma)^2$ is proportional to the negative logarithm ⇔ number of bits required to describe points near c_i.
- Choose σ that minimizes $E_{SA}(\sigma) = E_{\text{local}}(\sigma)^2 + \frac{C}{\sigma^2}$
- *C* is a positive constant that controls the trade off between sparsity and approximation, and the smoothness of the reconstruction.

$$E_{SA}(\sigma) = E_{local}(\sigma)^2 + \frac{C}{\sigma^2}$$

Adaptive PU Approximation: Finding Influence Parameters σ

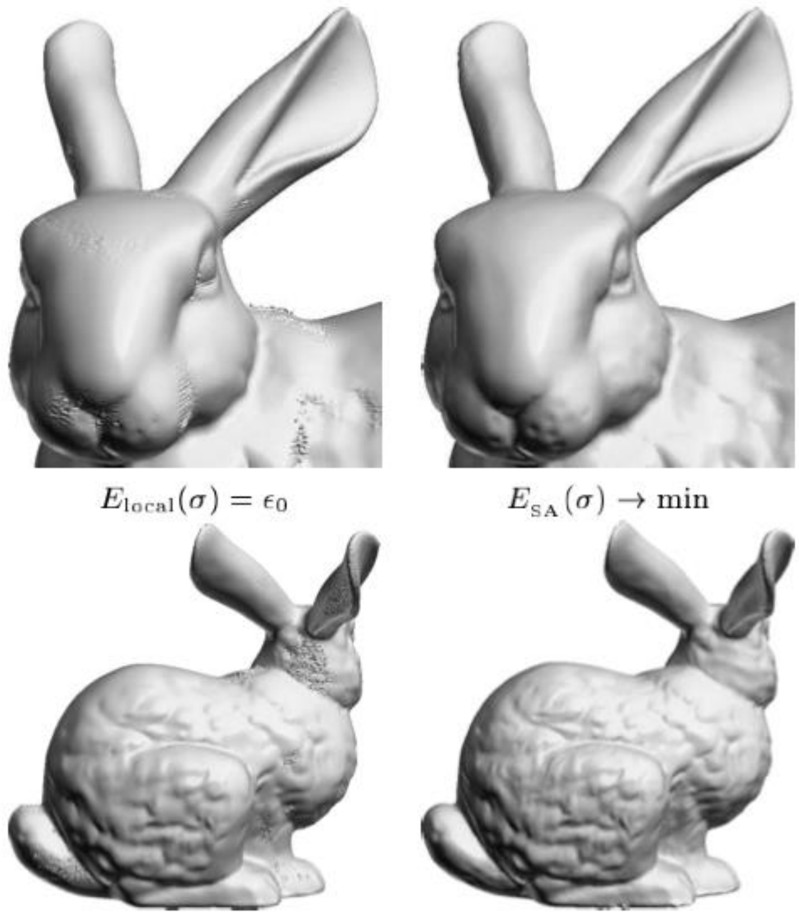
- If σ is large, then the number of approximation centers is small and local error $E_{local}(\sigma)$ is large.
  - Thus, $E_{SA}(\sigma)$ grows drastically as σ goes to infinity.
- For small σ, the number of approximation centers is large, since the zero level-set must reproduce noise.



$$E_{SA}(\sigma) = E_{local}(\sigma)^2 + \frac{C}{\sigma^2}$$

Adaptive PU Approximation: Finding Influence Parameters σ

- The value of σ_i reflects the surface complexity at c_i.
  - Bigger complexity ⇒ smaller σ_i.
- Two approaches for selecting σ_i:
  - Minimize $E_{SA}(\sigma)$: one-dimensional problem. Penalize number of local approximations.
  - Solve equation $E_{local}(\sigma) = \epsilon_0$ for a user-specified accuracy.
    - Small ϵ_0 ⇒ reconstruction of noise.
    - Large ϵ_0 ⇒ oversmoothing.



Algorithm Outline

- Adaptive PU Approximation
  - Select approximation centers c_i
  - Assign influence parameters $\sigma_{i.}$
  - Estimate function g_i.
- Least Squares RBF Approximation
  - Find coefficients λ_i

$$
\begin{aligned}
f(x) &= \sum_{c_i \in \mathcal{C}} [g_i(x) + \lambda_i] \phi_{\sigma_i}(||x - c_i||) \\
&= \sum_{c_i \in \mathcal{C}} g_i(x) \phi_{\sigma_i}(||x - c_i||) + \sum_{c_i \in \mathcal{C}} \lambda_i \phi_{\sigma_i}(||x - c_i||)
\end{aligned}
$$

Adaptive PU Approximation: Selecting the centers c_i

- Choose centers so that their corresponding balls cover all the sample points with overlap greater than a threshold.
- The cover for center c_i is a ball of radius σ_i centered at c_i, defined by $\text{supp}\ \phi_{\sigma_i}(||x - c_i||)$.
- Measure overlap at point p_j as

$$v_j = \sum_{i=1}^{M} \phi_{\sigma_i}(||\mathbf{p}_j - \mathbf{c}_i||)$$

Adaptive PU Approximation: Selecting the centers c_i

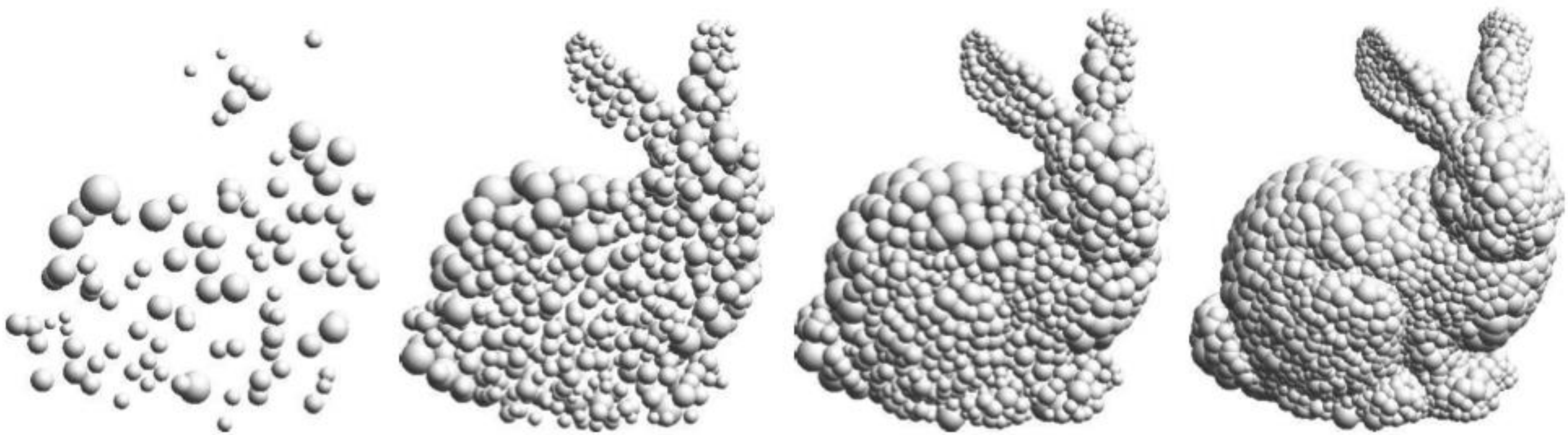
$$v_j = \sum_{i=1}^{M} \phi_{\sigma_i}(||\mathbf{p}_j - \mathbf{c}_i||)$$

- Algorithm for selecting c_i with $T_{overlap}$:
  1. Assign $v_j = 0$ for each sample point p_j.
  2. Choose m random sample points with $v < T_{overlap.}$
  3. Select the point with the minimum value of v.
  4. Choose that point as an approx center c_k and set $v_k = T_{overlap.}$
  5. Find support size σ_k and quadratic approximation $g_{i.}$
  6. Update overlap value v_j for all sample points not selected as centers by adding
  $$v_j += \phi_{\sigma_k}(||p_j - c_k||), \text{ for } p_j \in \mathcal{P} \backslash \mathcal{C}$$
  7. If there are points with $v < T_{overlap,}$ go to step 2.

Adaptive PU Approximation: Selecting the centers c_i

$$v_j = \sum_{i=1}^{M} \phi_{\sigma_i}(\|\mathbf{p}_j - \mathbf{c}_i\|)$$

- Algorithm for selecting c with T

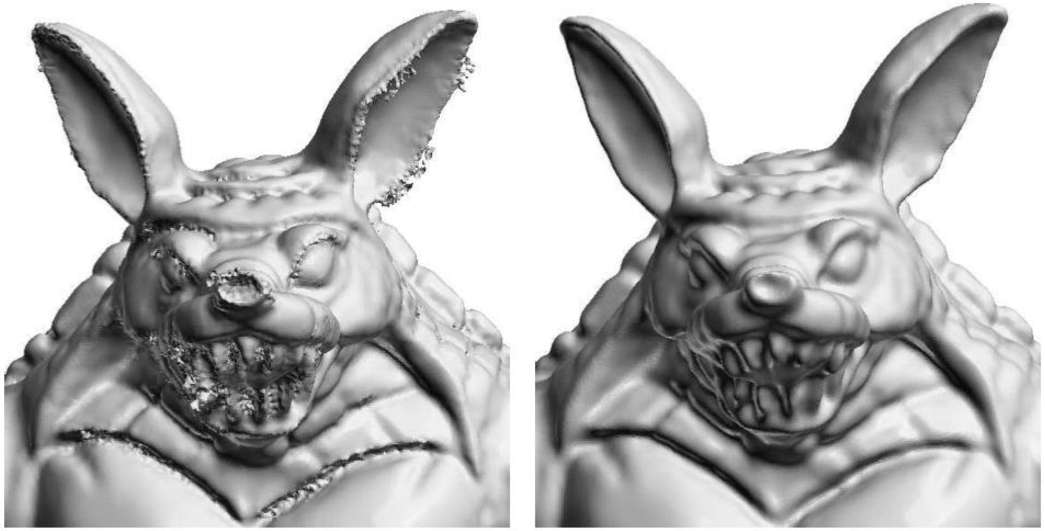


7. If there are points with $v < T_{overlap}$, go to step 2.

Adaptive PU Approximation: Noisy Data

- Extra zero level-set surfaces appear on noisy datasets.
- To avoid artifacts, prevent influence parameter σ from being too small.
  - Modify $E_{local}(\sigma)$ for cases with small influence parameters:

$$E_{\mathrm{local}}(\sigma) = L \quad \text{if} \quad \sigma < \sigma_{\min}$$



$\sigma_{\min} = 0, M = 134K$ $\sigma_{\min} = L/100, M = 42K$

Algorithm Outline

- Adaptive PU Approximation
  - Select approximation centers c_i
  - Assign influence parameters $\sigma_{i.}$
  - Estimate function g_i.
- Least Squares RBF Approximation
  - Find coefficients λ_i

$$
\begin{aligned}
f(x) &= \sum_{c_i \in \mathcal{C}} [g_i(x) + \lambda_i] \phi_{\sigma_i}(||x - c_i||) \\
&= \sum_{c_i \in \mathcal{C}} g_i(x) \phi_{\sigma_i}(||x - c_i||) + \sum_{c_i \in \mathcal{C}} \lambda_i \phi_{\sigma_i}(||x - c_i||)
\end{aligned}
$$

$$d_i = v_i \sum_{j=1}^{K} ||p_i - p_j||^2$$

$$f(x) = \sum_{c_i \in \mathcal{C}} [g_i(x) + \lambda_i] \phi_{\sigma_i}(||x - c_i||) = \sum_{c_i \in \mathcal{C}} g_i(x) \phi_{\sigma_i}(||x - c_i||) + \sum_{c_i \in \mathcal{C}} \lambda_i \phi_{\sigma_i}(||x - c_i||)$$

Least-Squares RBF Approximation: Determining RBF weights λ_i

- Define global L^2-error metric:

$$E_{\text{global}}(\boldsymbol{\lambda}) = \frac{1}{L} \sqrt{\frac{\sum_{j=1}^{N} d_j \, f(\mathbf{p}_j)^2}{\sum_{j=1}^{N} d_j}}$$

- Want to minimize global error metric to obtain RBF weights, but this produces overfitting.
  - Use a regularization approach to suppress oscillations. Modification:

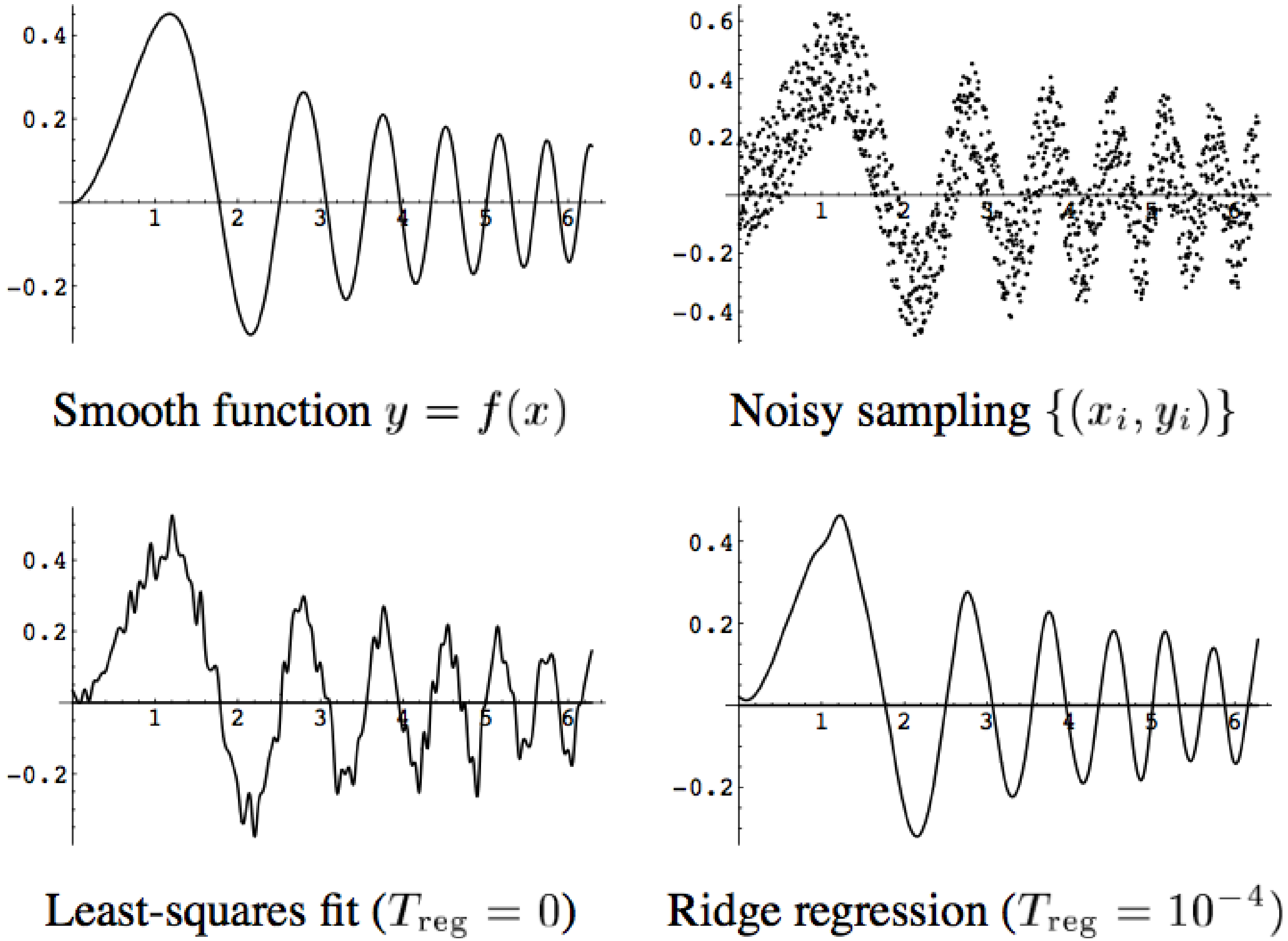
$$E_{\text{reg}}(\boldsymbol{\lambda}) = E_{\text{global}}(\boldsymbol{\lambda})^2 + T_{\text{reg}} \, ||\boldsymbol{\lambda}||^2 \rightarrow \min \qquad ||\boldsymbol{\lambda}|| = \sqrt{\frac{1}{M} \sum_{i=1}^{M} \left(\frac{\lambda_i}{\sigma_i} \right)^2}$$

  - This is a quadratic min problem so $\frac{\partial E_{\text{reg}}(\boldsymbol{\lambda})}{\partial \boldsymbol{\lambda}} = \mathbf{0} \iff (\mathbf{A} + T_{\text{reg}} \mathbf{D}) \boldsymbol{\lambda} = \mathbf{b}$

$$\begin{cases} A_{ij} = \dfrac{\sum_{k=1}^{N} d_k \, \Phi_{\sigma_i}(||\mathbf{p}_k - \mathbf{c}_i||) \, \Phi_{\sigma_j}(||\mathbf{p}_k - \mathbf{c}_j||)}{L^2 \sum_{k=1}^{N} d_k} \\ D_{ii} = \dfrac{1}{M} \left(\dfrac{1}{\sigma_i} \right)^2, \\ b_i = \dfrac{\sum_{k=1}^{N} d_k \, \Phi_{\sigma_i}(||\mathbf{p}_k - \mathbf{c}_i||) \left(-f(\mathbf{p}_k)|_{\boldsymbol{\lambda}=\mathbf{0}} \right)}{L^2 \sum_{k=1}^{N} d_k} \end{cases}$$

$$d_i = v_i \sum_{j=1}^{K} ||p_i - p_j||^2$$

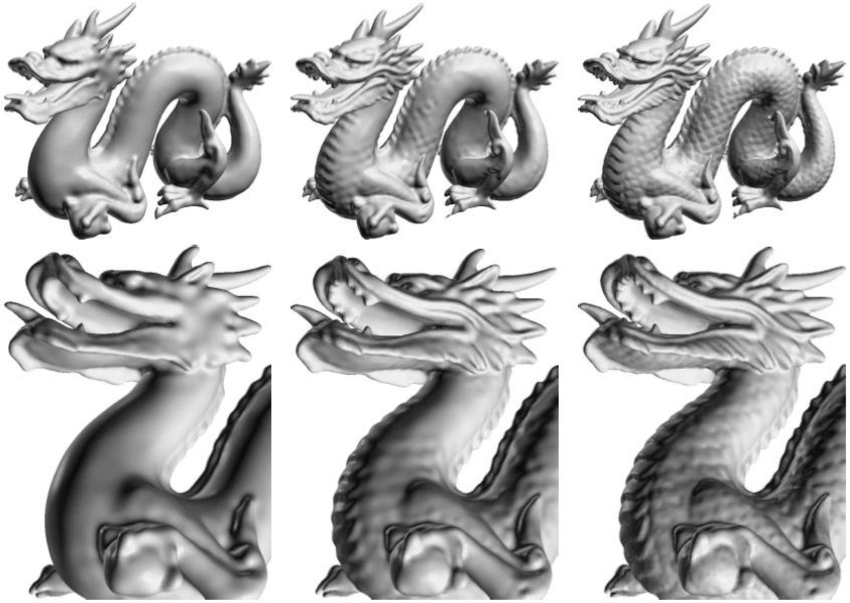
$$f(x) = \sum_{c_i \in \mathcal{C}} [g_i(x) + \lambda_i] \phi_{\sigma_i}(||x - c_i||)$$
$$= \sum_{c_i \in \mathcal{C}} g_i(x) \phi_{\sigma_i}(||x - c_i||) + \sum_{c_i \in \mathcal{C}} \lambda_i \phi_{\sigma_i}(||x - c_i||)$$



$$b_i = \frac{\sum_{k=1}^{N} d_k \, \Phi_{\sigma_i}(||\mathbf{p}_k - \mathbf{c}_i||) \, (-f(\mathbf{p}_k)|_{\boldsymbol{\lambda}=\mathbf{0}})}{L^2 \sum_{k=1}^{N} d_k}$$

Discussion

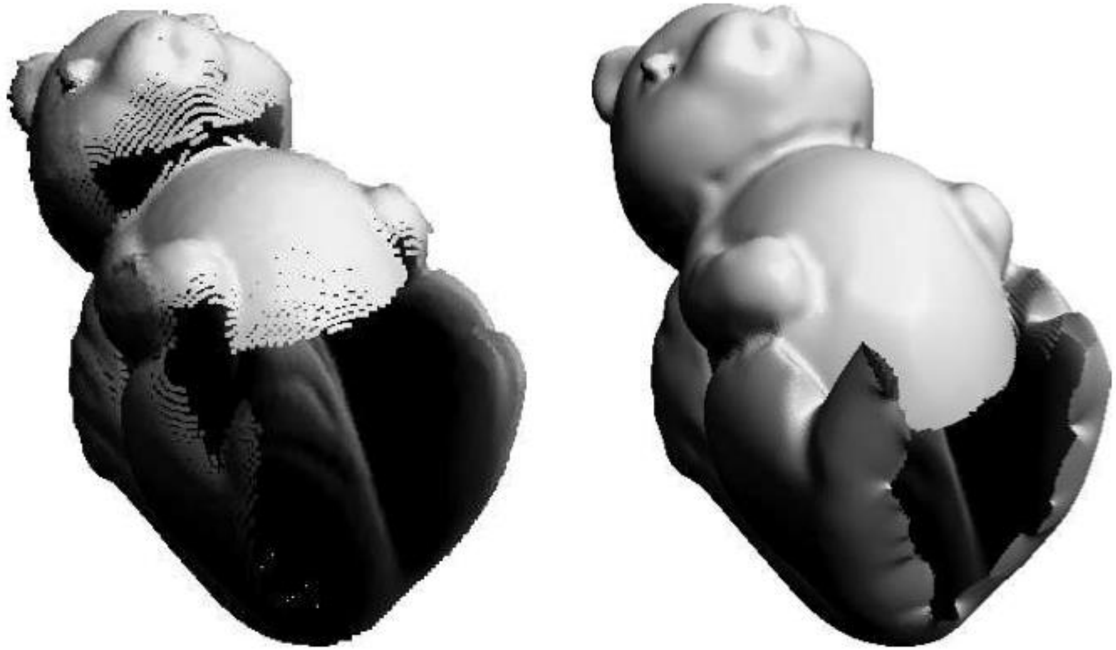
- Parameter selection
  - T_{SA} controls the smoothness of the model
  - $T_{overlap}$ and T_{reg} are fixed.
  - $\sigma_{min} = 0$ for low noise.
  - $\sigma_{min} = L/100$ for noisy data.



$T_{SA} = 5 \times 10^{-5}$ $T_{SA} = 1 \times 10^{-5}$ $T_{SA} = 2 \times 10^{-6}$

Discussion

- Hole filling



Discussion

- Performance
  - To evaluate function *f(x)*, we need to find all center c_i such that *x* belongs to their areas of influence.
    - Use a range searching octree-based data structure.
  - For visualization, use Bloomenthal's polygonizer.
    - One linear interpolation pass required to find *f(x) = 0* since *f(x)* mimics the distance function to the level set when close to zero.
  - Time complexity and number of approximation centers depends on size of the dataset and on the geometric complexity.
- Future work
  - Selecting approximation centers that are not restricted to sample points.
  - Combine method with a multi-scale approach.