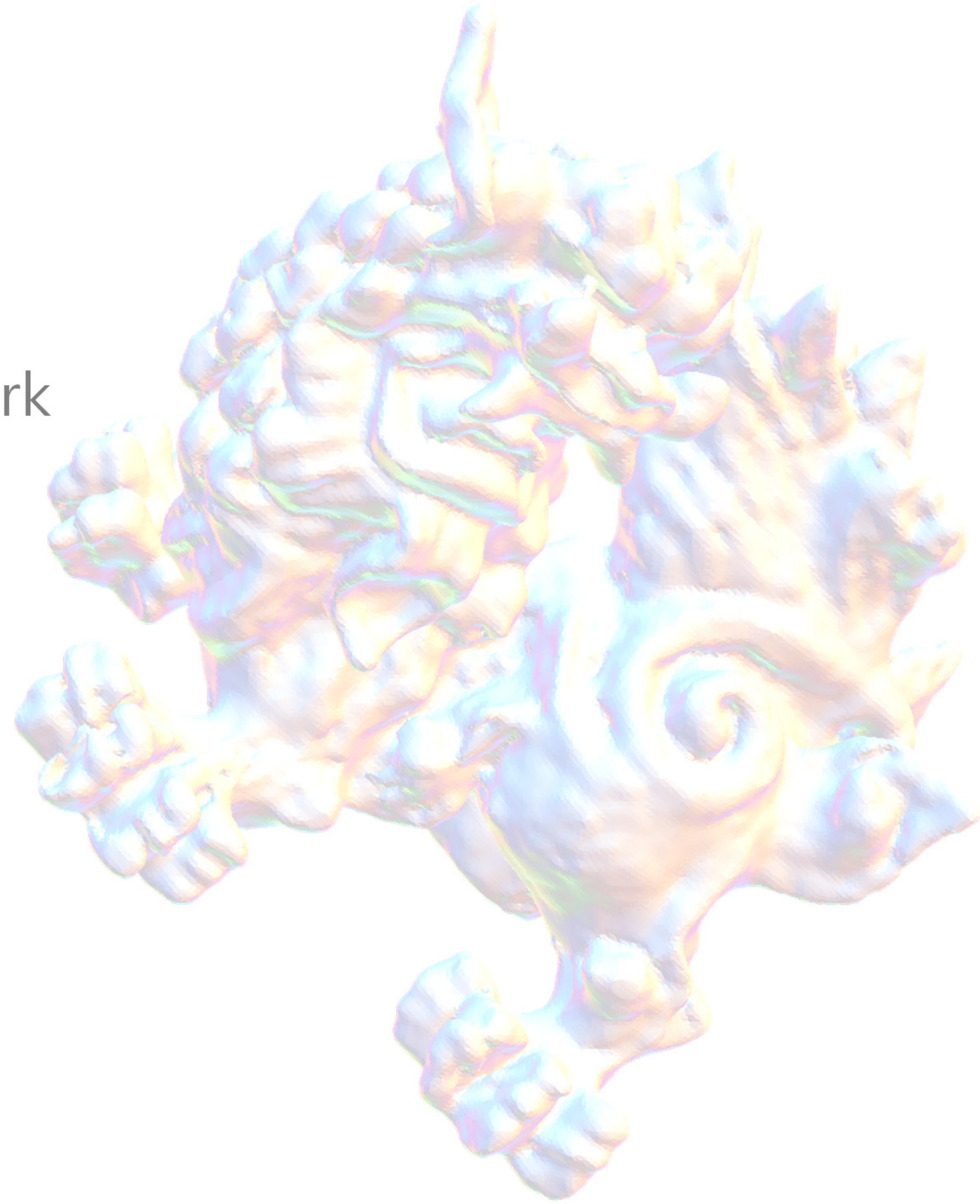
Presentation at ACDDE-2010, 25th - 28th August 2010, at Jeju island, Korea.

Outlier/Noise-Robust Partition of Unity Implicit Surface Reconstruction

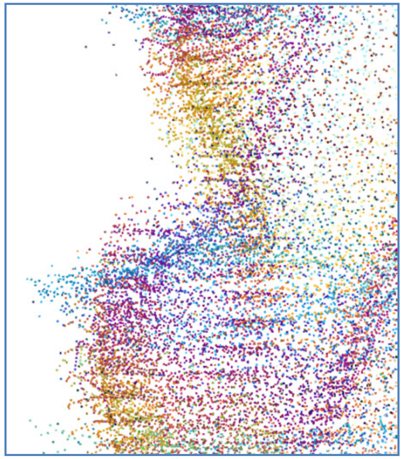
Yukie Nagai*	(RIKEN)
Yutaka Ohtake	(The university of Tokyo)
Hiromasa Suzuki	(The university of Tokyo)
Hideo Yokota	(RIKEN)

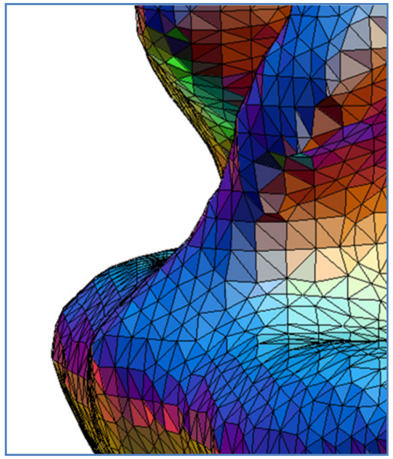
Overview

- ➢ Introduction
- Algorithm
- Results
- Conclusion & Future work



Surface Reconstruction





Point cloud → Surface mesh

Explicit

- Delaunay based method
  - Power Crust, Robust Cocone, Eigencrust

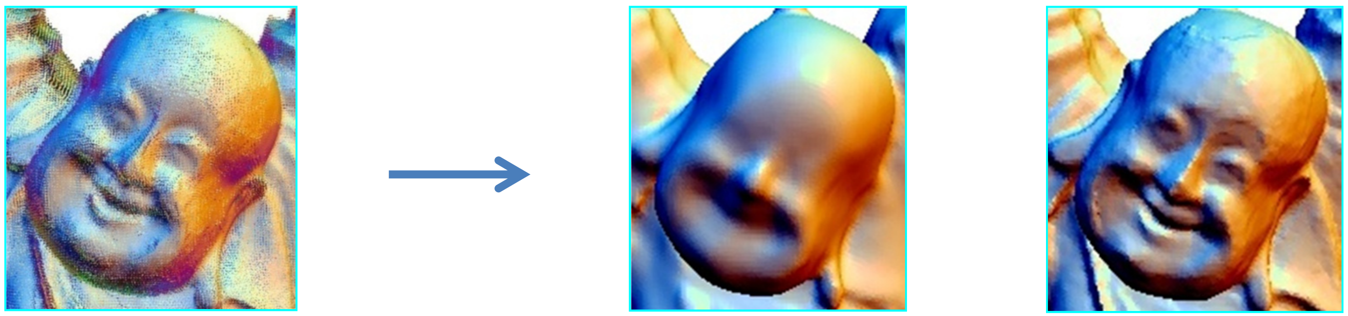
Implicit

- Global
  - RBF, Graph-cut, FEM, Poisson surface reconstruction
- Local
  - VRIP, MLS, MPU

Advantages of PU

- Processing nD data
- Representation with high order functions

- Easy control of the accuracy of approximation



3D points Large supports Small supports

- Handling topological changes
- Fast & memory efficient computation

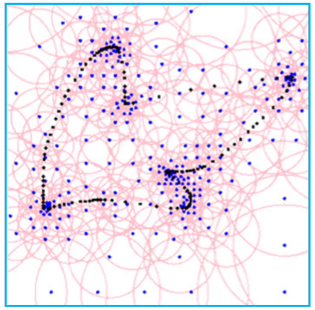
PU Spherical Cover

- Covering a space with spherical supports
- Representing a continuous function
  as a weighted average of local approximations

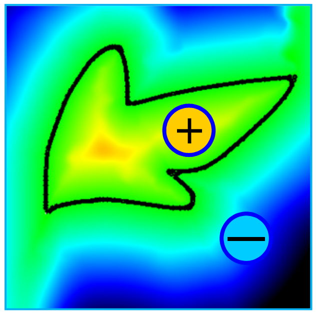
$$\boxed{f(\boldsymbol{x}) = \frac{\sum_i w_i(\boldsymbol{x}) g_i(\boldsymbol{x})}{\sum_i w_i(\boldsymbol{x})}}$$

- For surface reconstruction,
  $f(\boldsymbol{x})$ approximates the signed distance from a surface

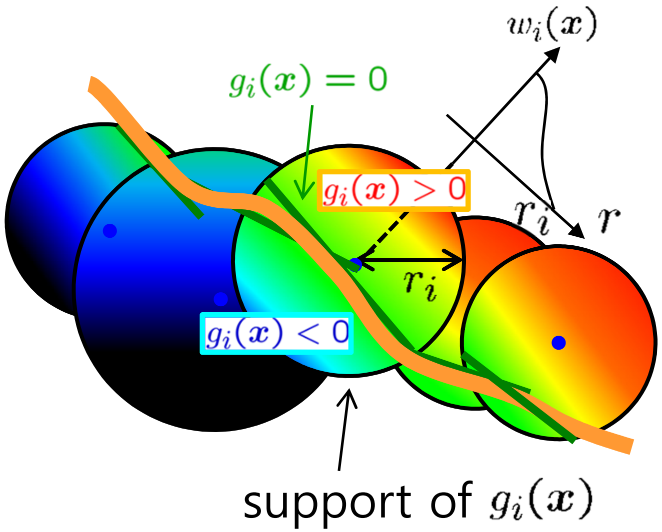
[Ohtake *et al.* 03]



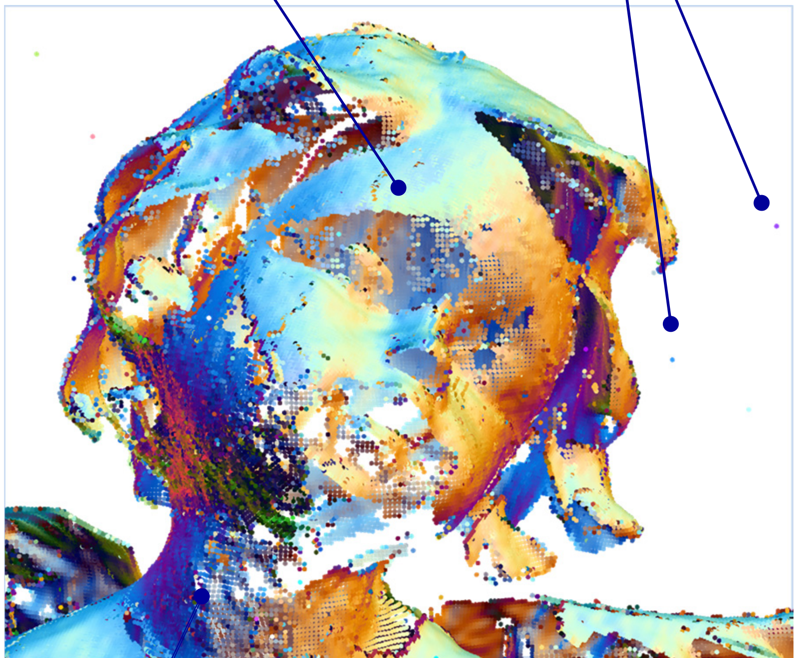
Points sampled
from a surface



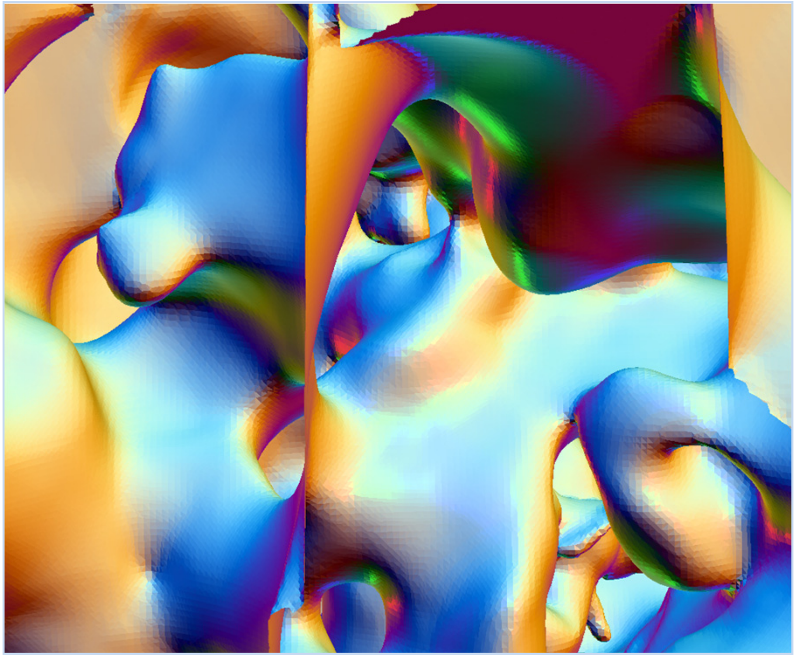
0-level set
~ surface



Problems of Local Fitting

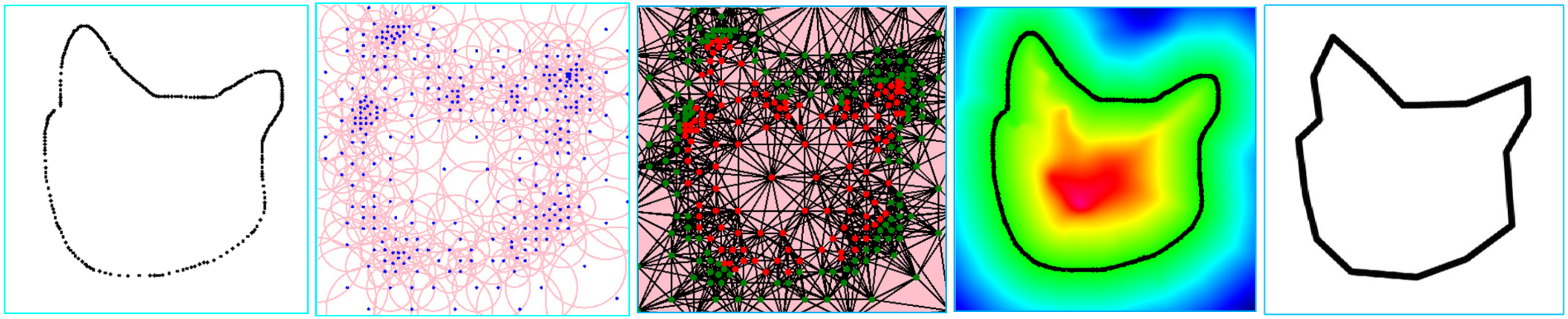


Sampling points



PU

Our Approach

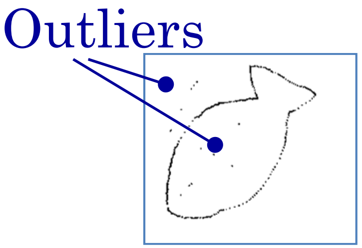


1. Generating implicit functions
2. Graph-cut (outlier & lack)
3. Smoothing approximation functions (noise & lack)
4. Polygonizing 0-level set

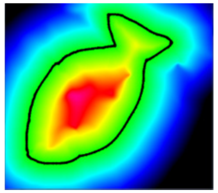
Outlier: Graph-cut

- **Locality** of PU and outliers (**local** error) cause in/out-classification error

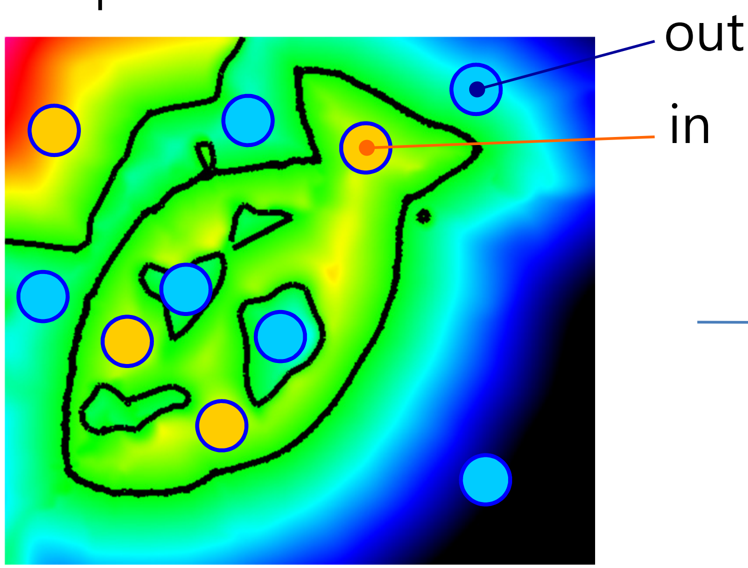
  → Using a **global** method (Graph-cut)



Input

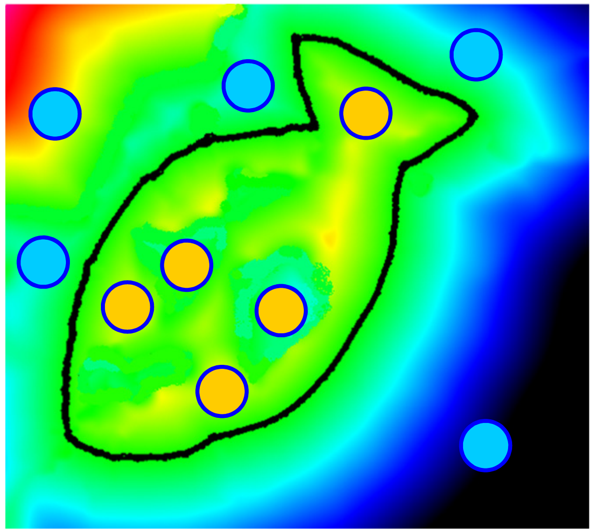


Ideal result



PU

→

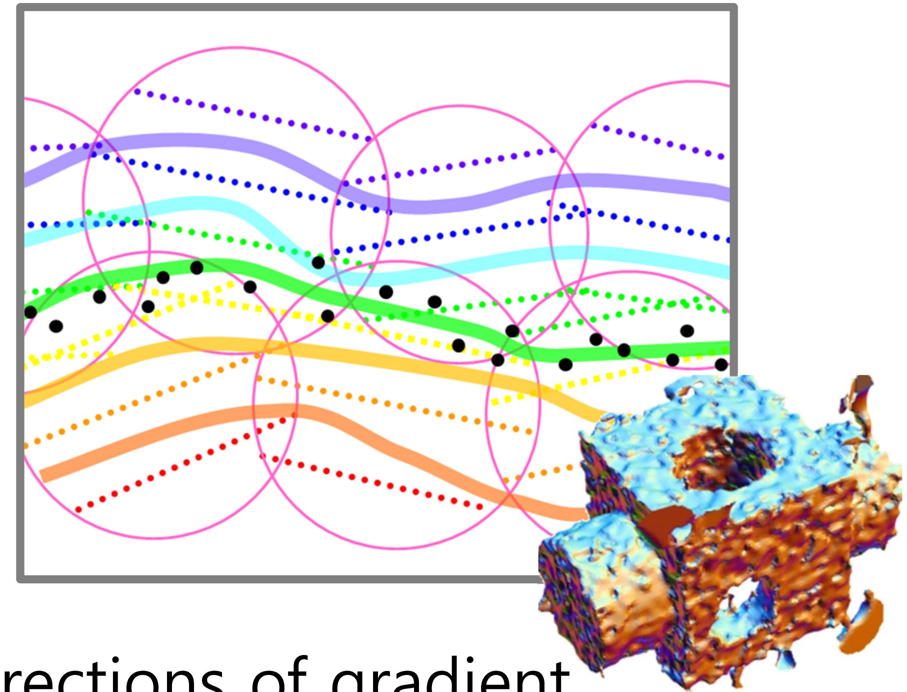


PU + Graph-cut

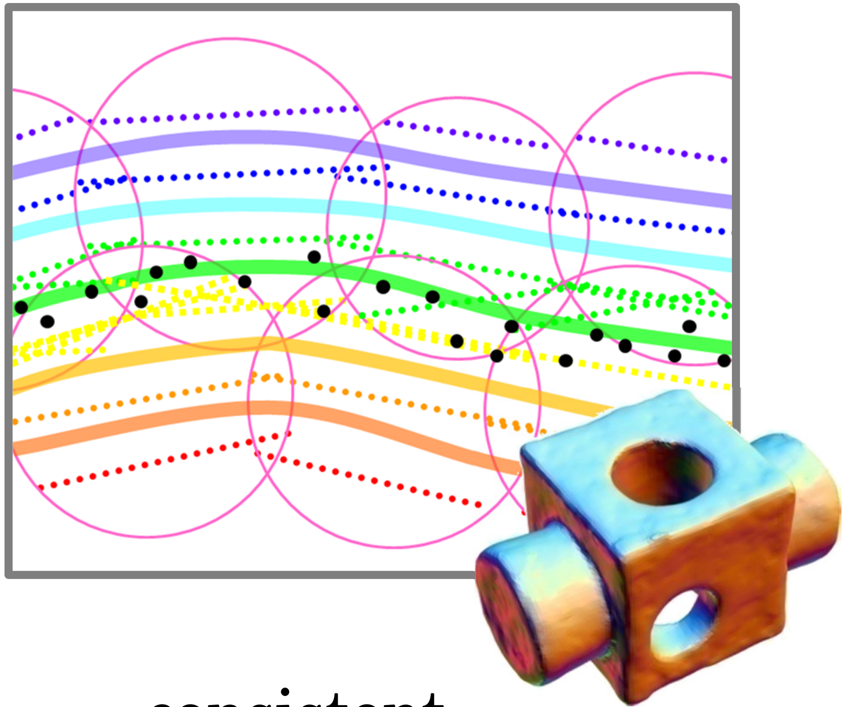
Noise: Smoothing Local Approximations

Smoothing the gradient field [SGP 09]

- Similar to tetrahedral mesh version [Tong *et al.* 03]



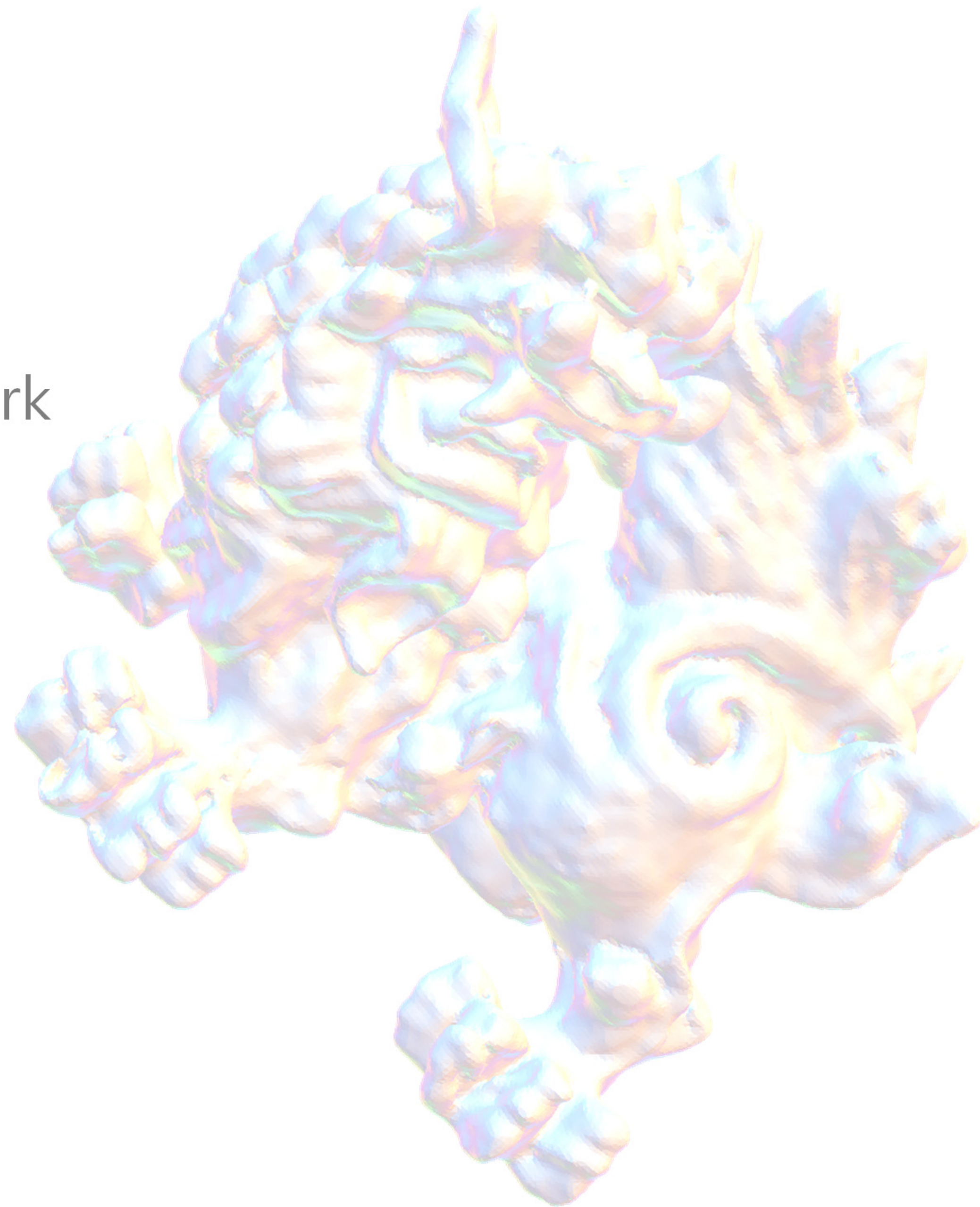
Directions of gradient
and isosurfaces are **different**



consistent

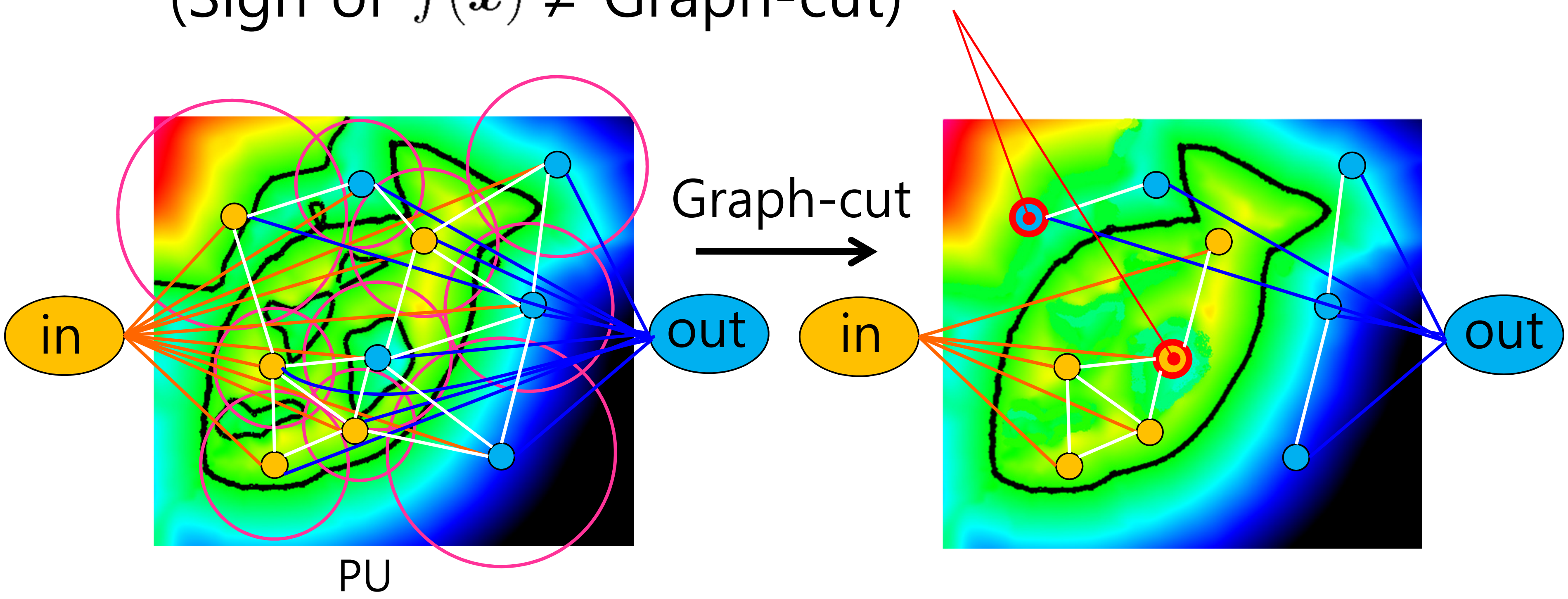
Overview

- Introduction
- ➢ Algorithm
- Results
- Conclusion & Future work



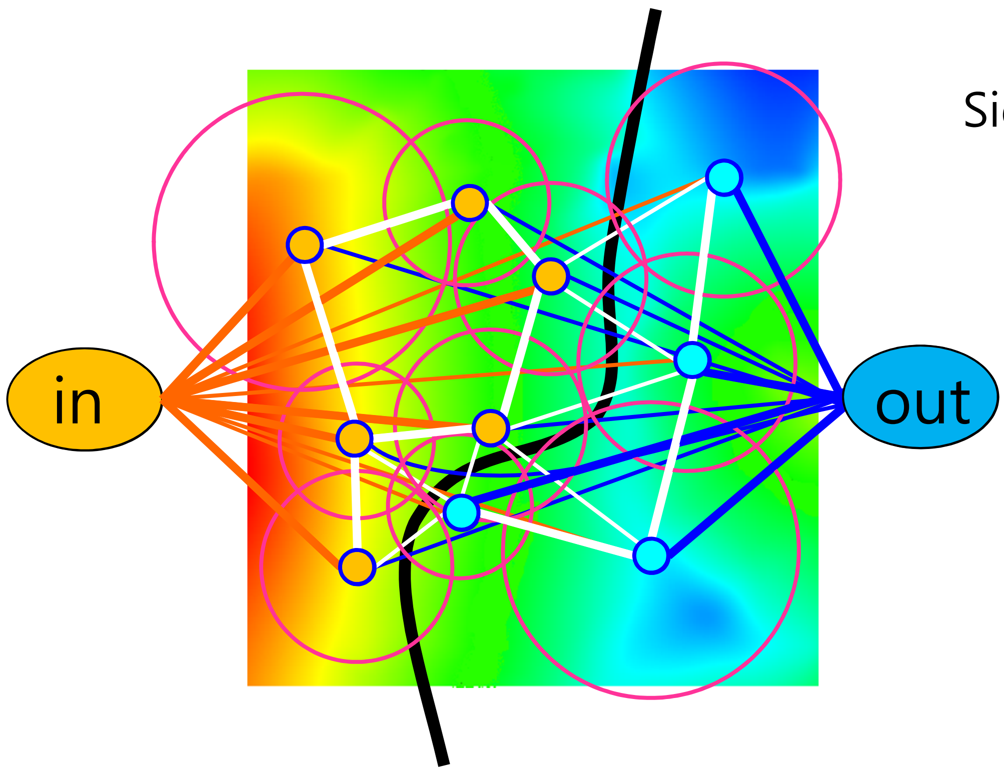
Graph-cut

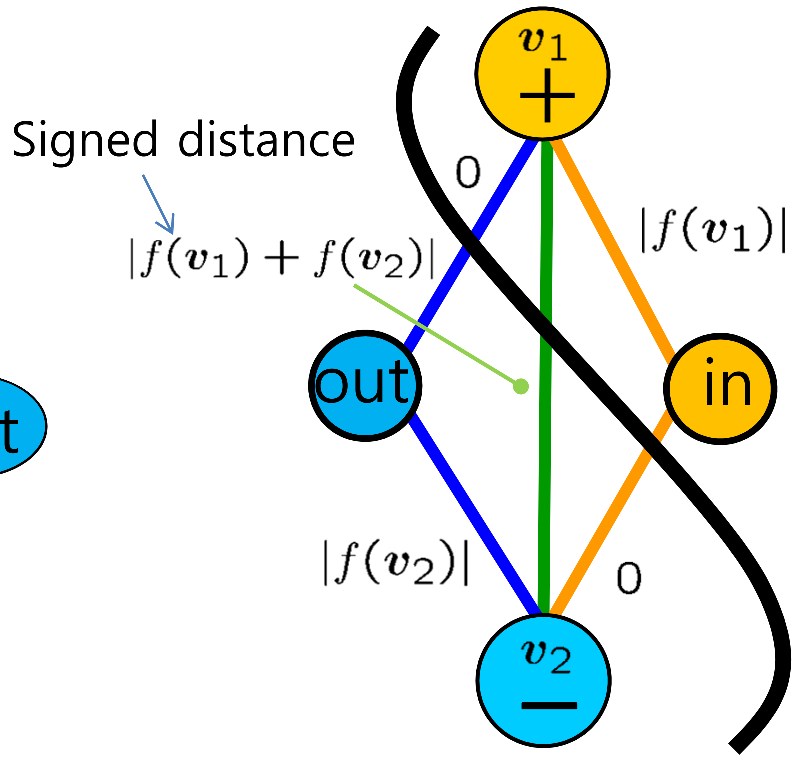
- Correcting the in/out-classification of the supports
  (Minimal cut ~ surface)
- Detecting supports affected by outliers
  (Sign of $f(\boldsymbol{x}) \neq$ Graph-cut)



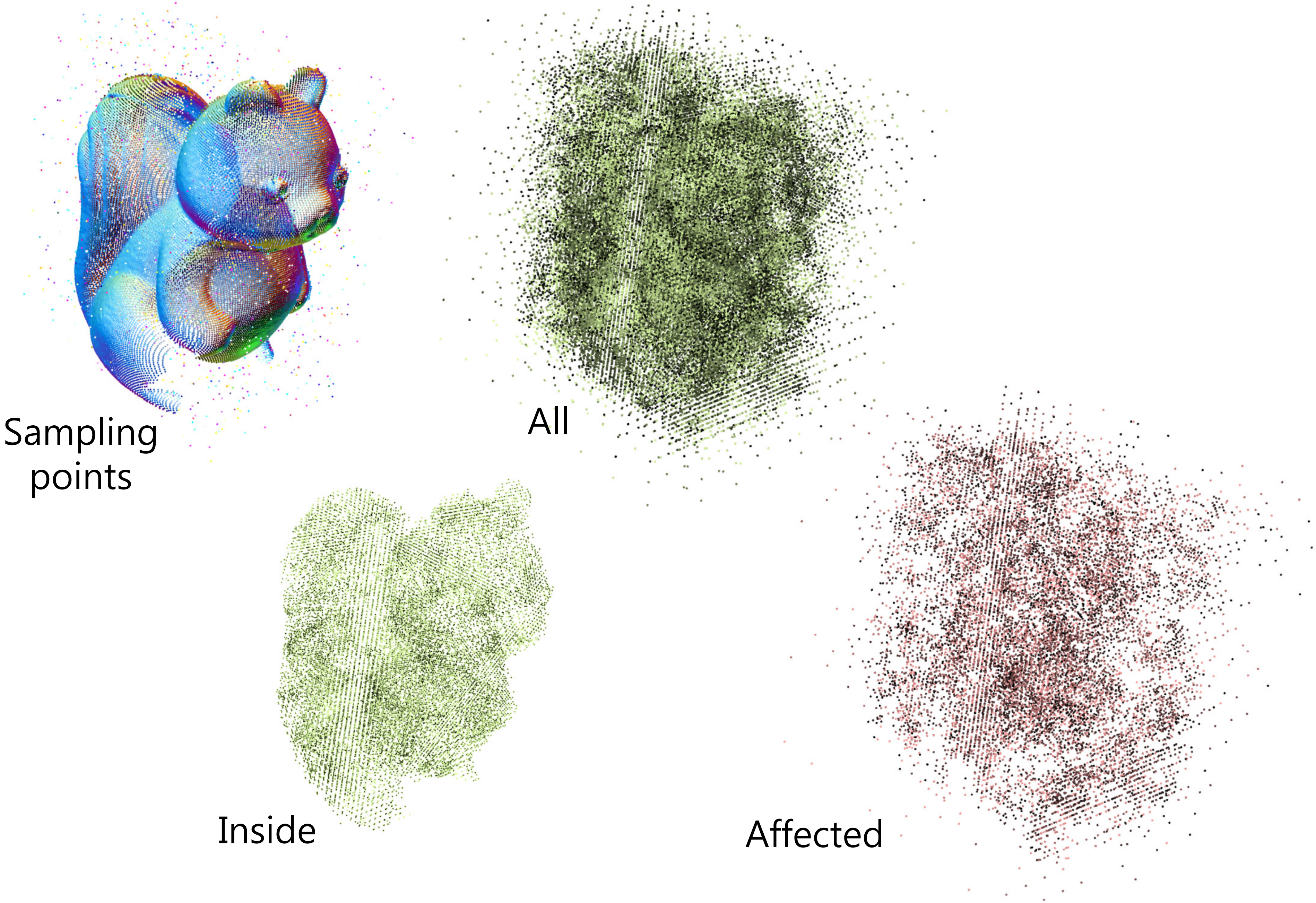
Graph Generation

- Connecting: sphere intersection graph
- Weighting: small value for a crossing edge

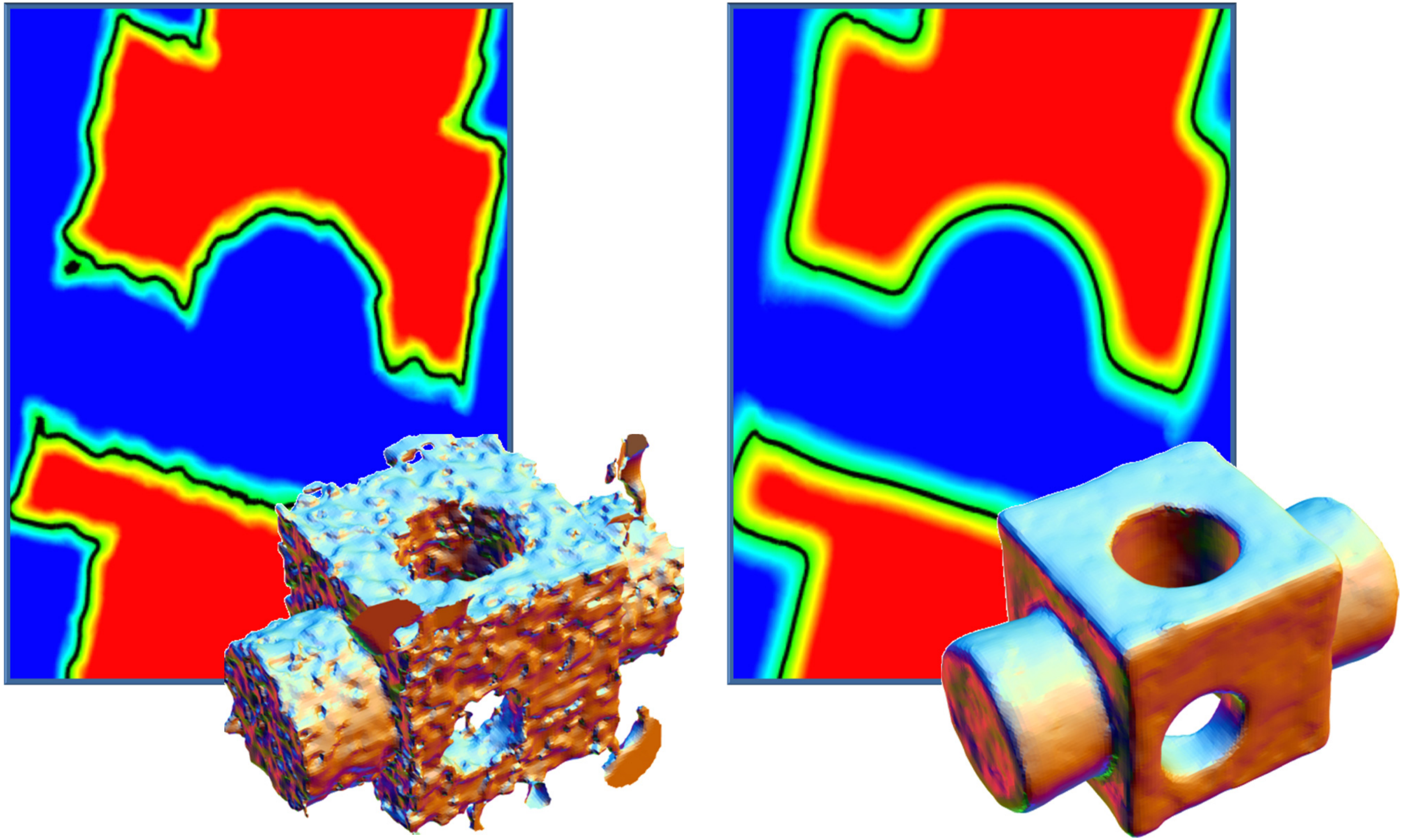




Classified Centers



Smoothing Local Approximations



Smoothing Local Approximations

- Local approximation: $g_i(\boldsymbol{x}) = \boldsymbol{\alpha}_i \cdot (\boldsymbol{x} - \boldsymbol{c}_i) + \beta_i$

Initialize gradient field

$\boldsymbol{v}(\boldsymbol{c}_i) \equiv \nabla f(\boldsymbol{c}_i)$

Smooth gradients

Updating $\{\boldsymbol{\alpha}_i\}$

Laplacian smoothing

$\text{Lap}(\hat{\boldsymbol{v}}_i) = 0$

$\hat{\boldsymbol{\alpha}}_i \equiv \hat{\boldsymbol{v}}_i$

Make isosurfaces consistent

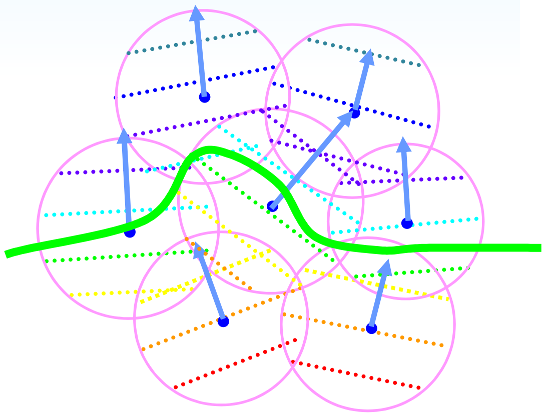
Updating $\{\beta_i\}$

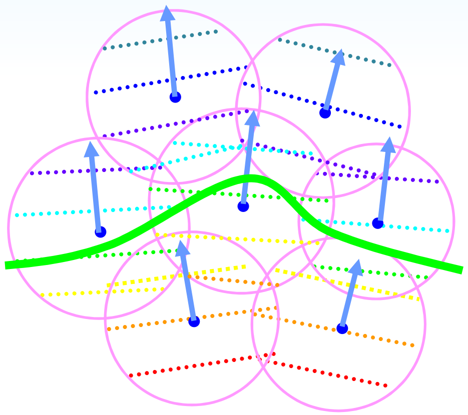
Poisson equation

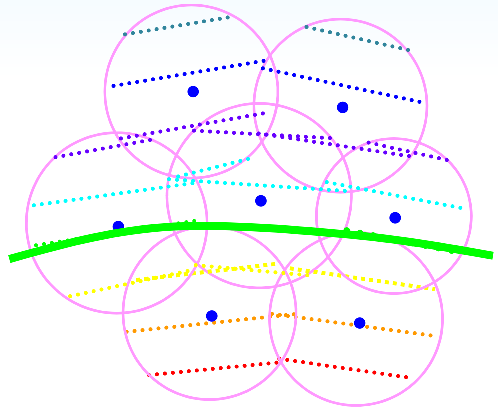
$\text{Grad}\,(f(\boldsymbol{c}_i)) = \hat{\boldsymbol{v}}_i$

weak formulation

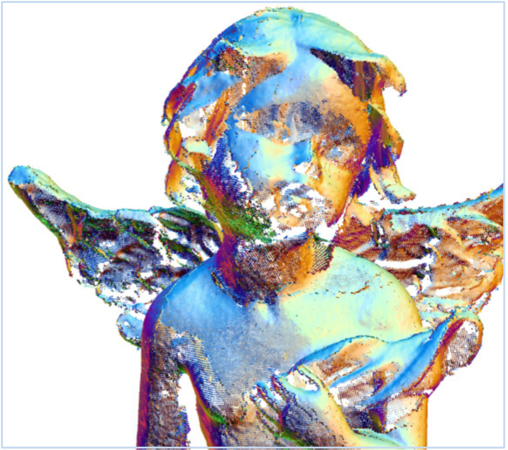
$\text{Lap}(f(\boldsymbol{c}_i)) = \text{Div}(\hat{\boldsymbol{v}}_i)$



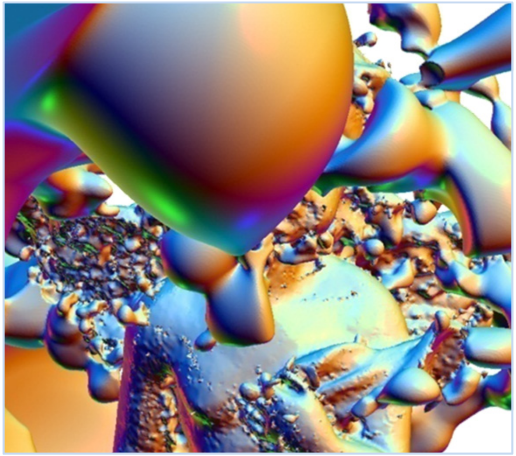




Effects of Smoothing



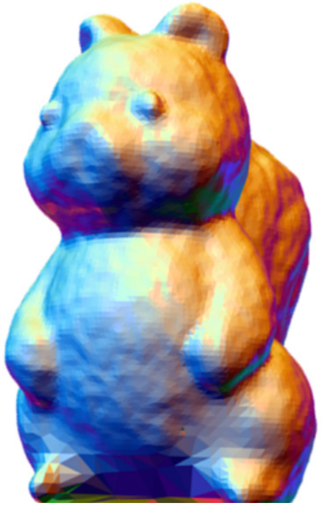
Noisy points (No outliers)



PU

PU+Smoothing

Noisy normals

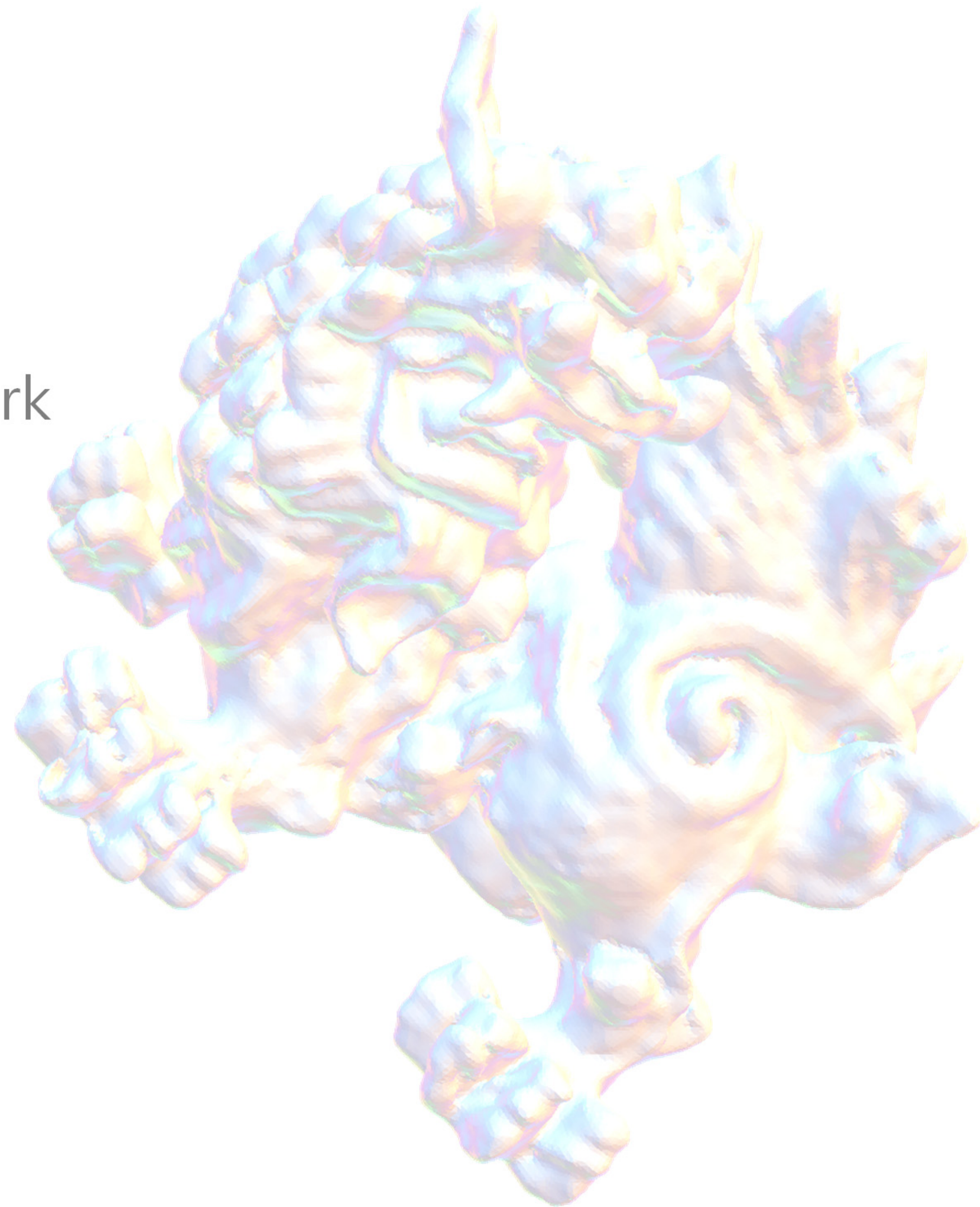


Poisson surface reconstruction
[Kazhdan *et al.* 06]

PU+Smoothing

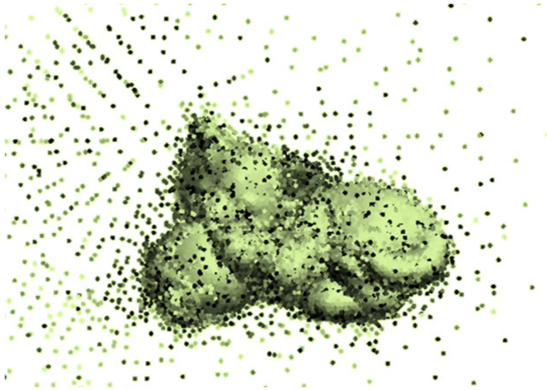
Overview

- Introduction
- Algorithm
- ➢ Results
- Conclusion & Future work

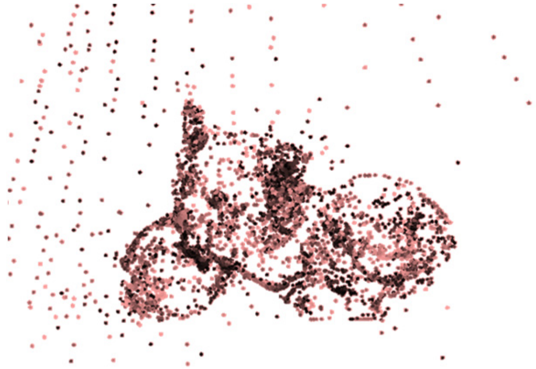


Results: noise & outliers

Sampling points

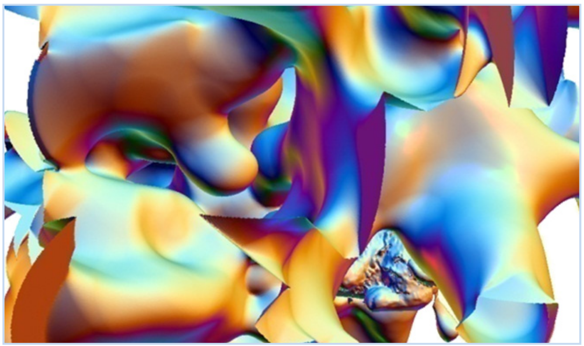


All centers



Outlier-affected centers

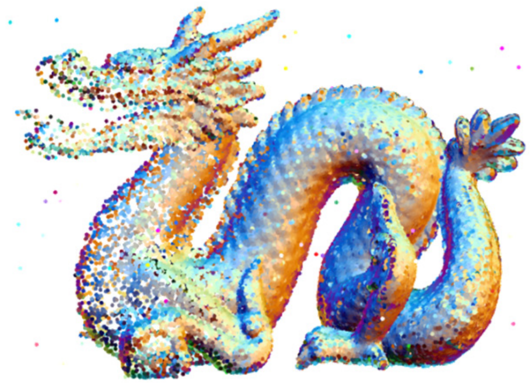
Original object



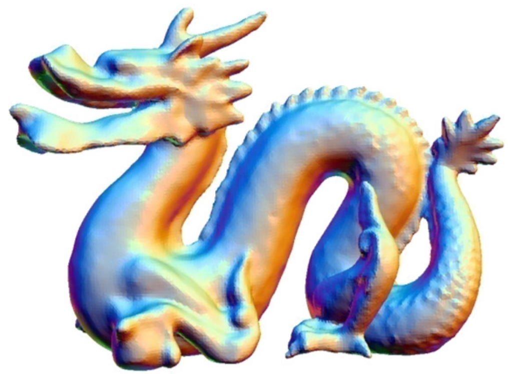
PU

Graph-cut+Smoothing

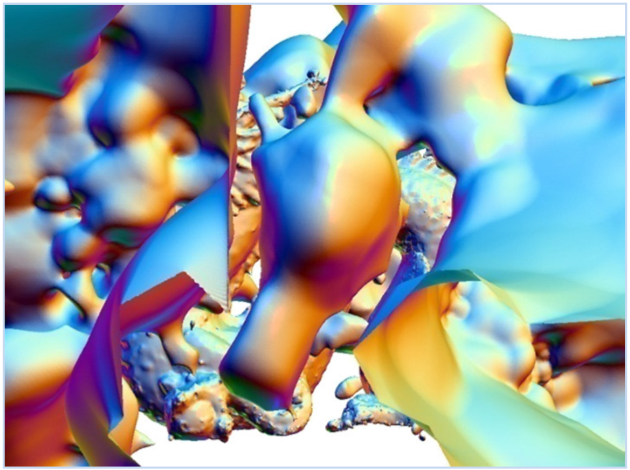
Comparison: sparse & outliers



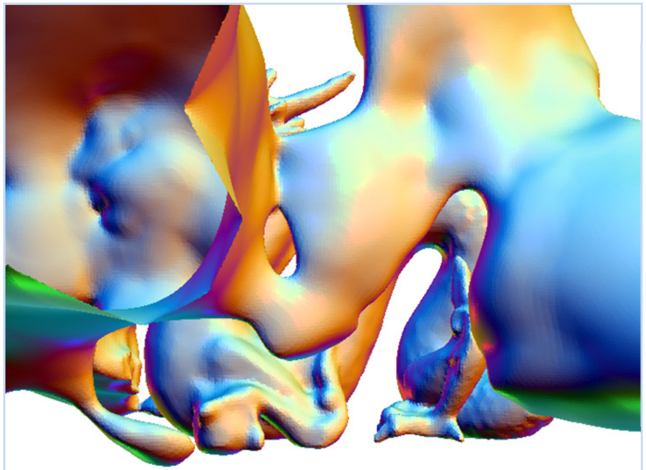
Sampling points



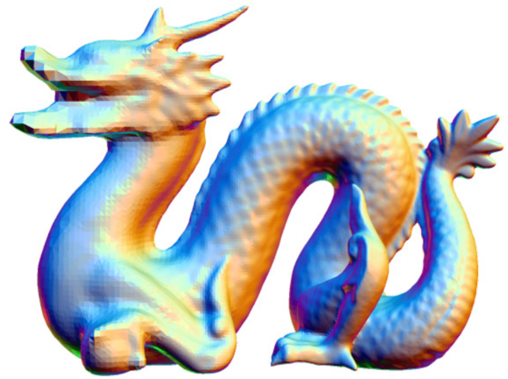
Graph-cut+Smoothing



PU



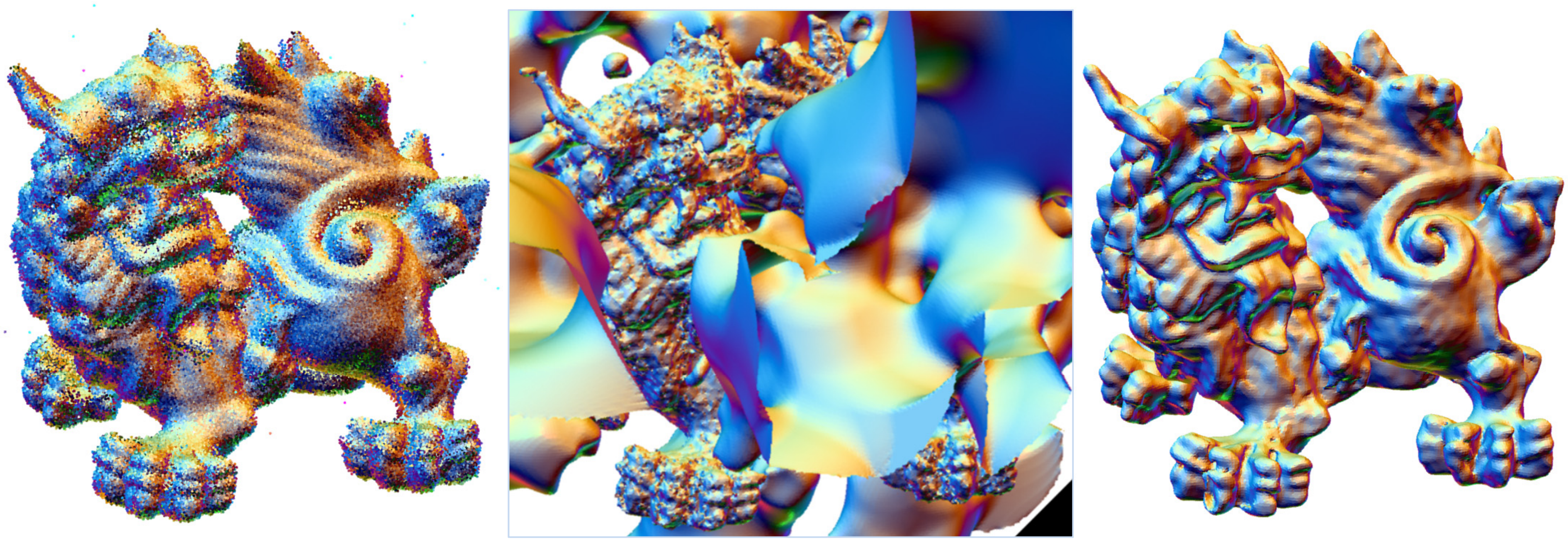
Smoothing
[SGP 09]



Poisson surface reconstruction
[Kazhdan *et al.* 06]

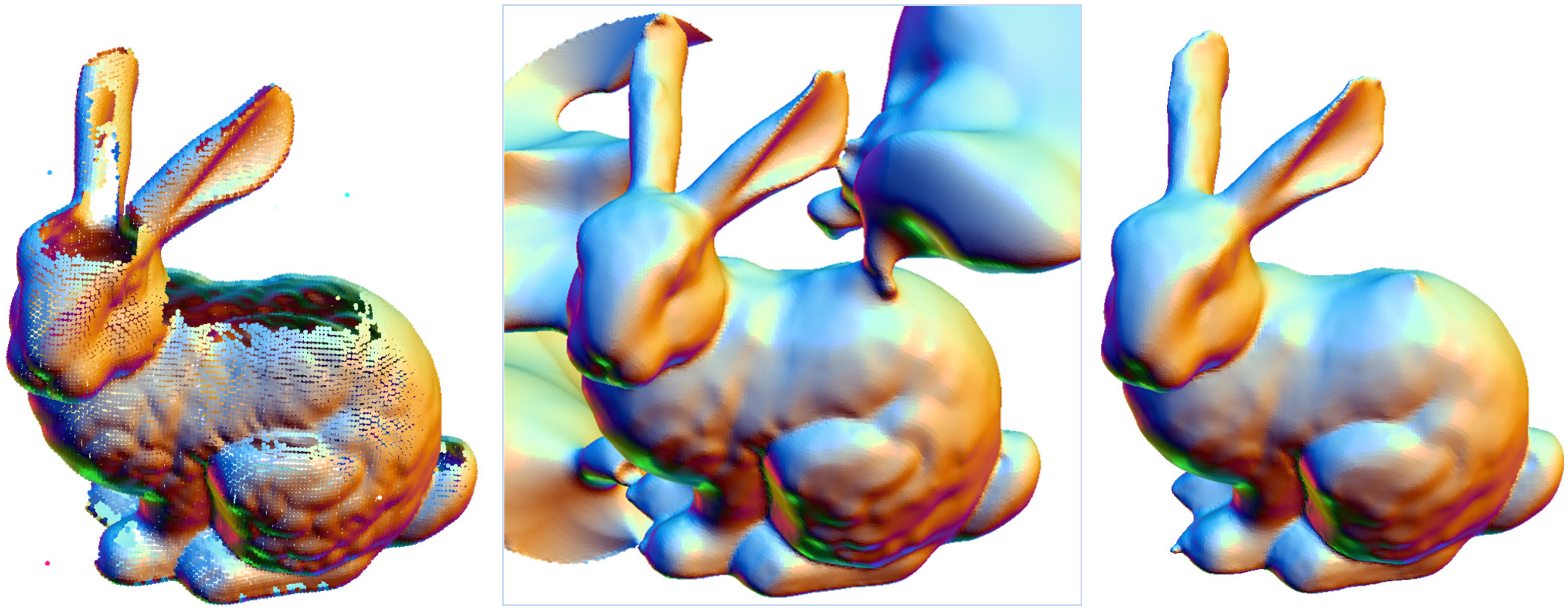
Conclusion

- Outlier/noise-robust and detail preserving surface reconstruction is achieved with the combination of Graph-cut and smoothing



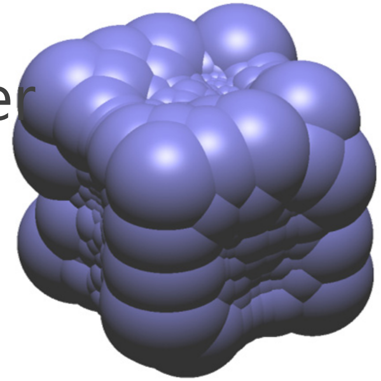
Future work

- Trade-off between detail preservation and extra surface-free reconstruction
- Tuning parameters adaptively may give better results

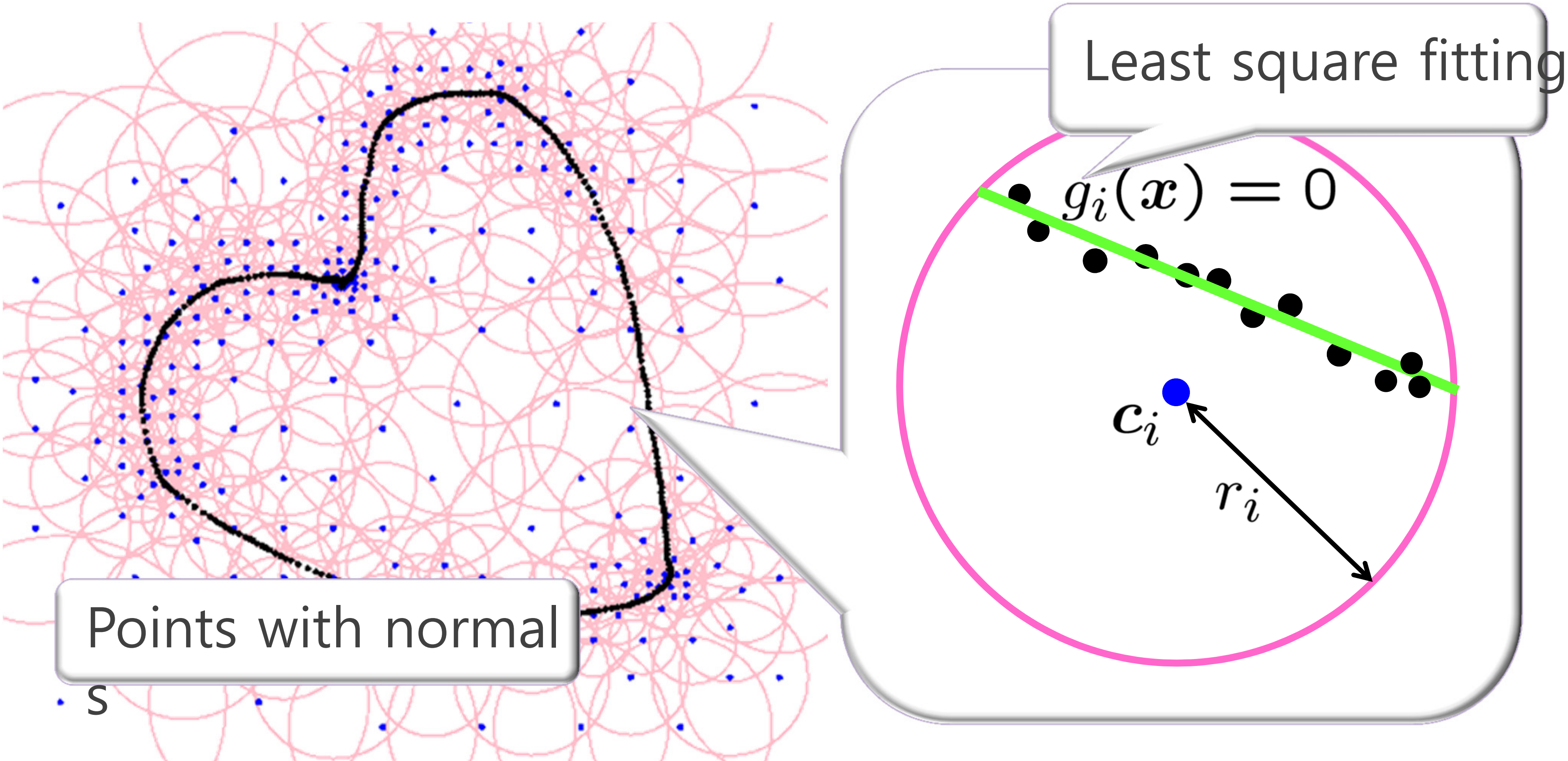


Partition of Unity

3D spherical cover



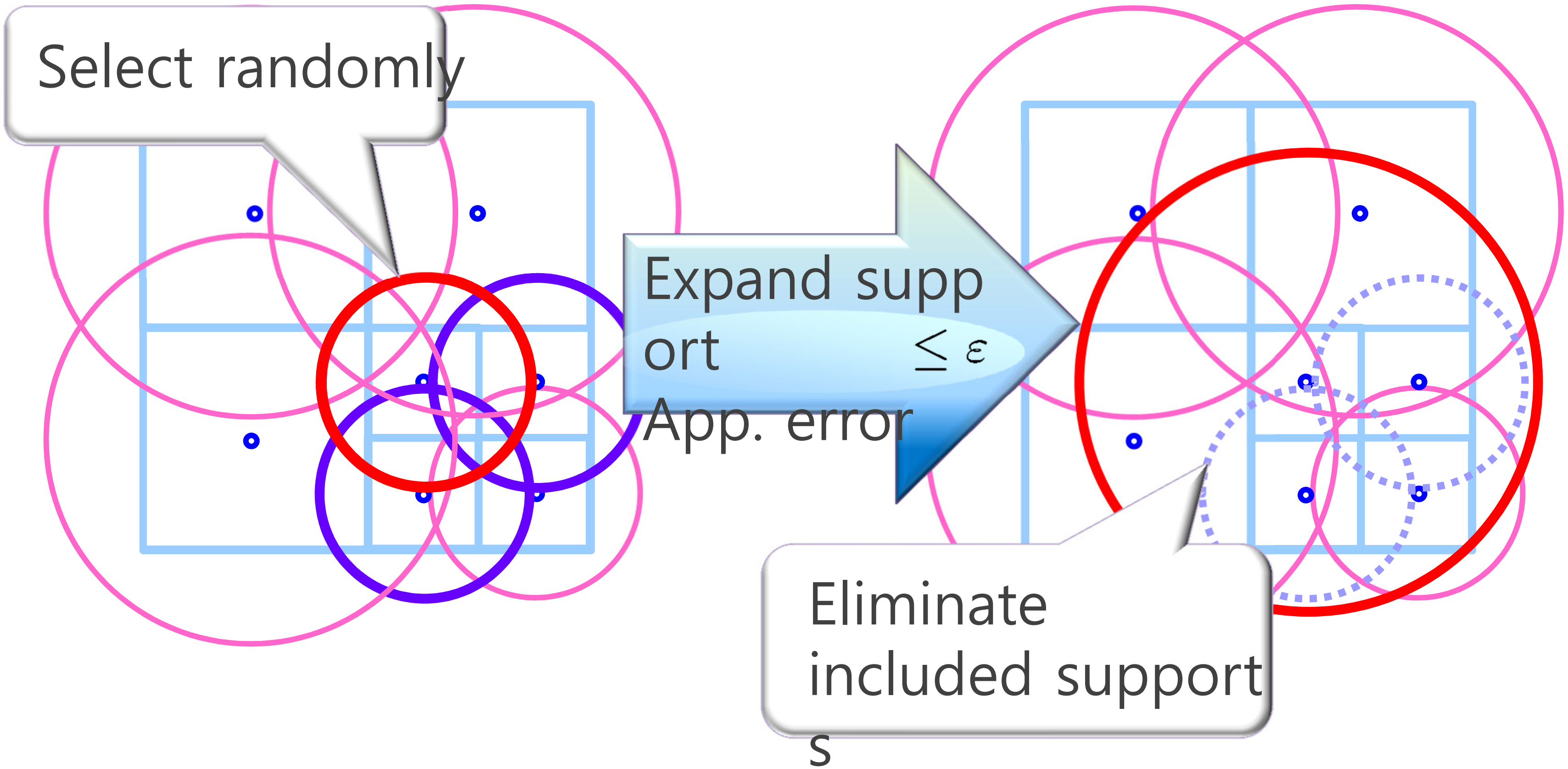
- Covering a space with spheres
- Each sphere is associated with a function $g_i(\boldsymbol{x}) = \boldsymbol{\alpha}_i \cdot (\boldsymbol{x} - \boldsymbol{c}_i) + \beta_i$



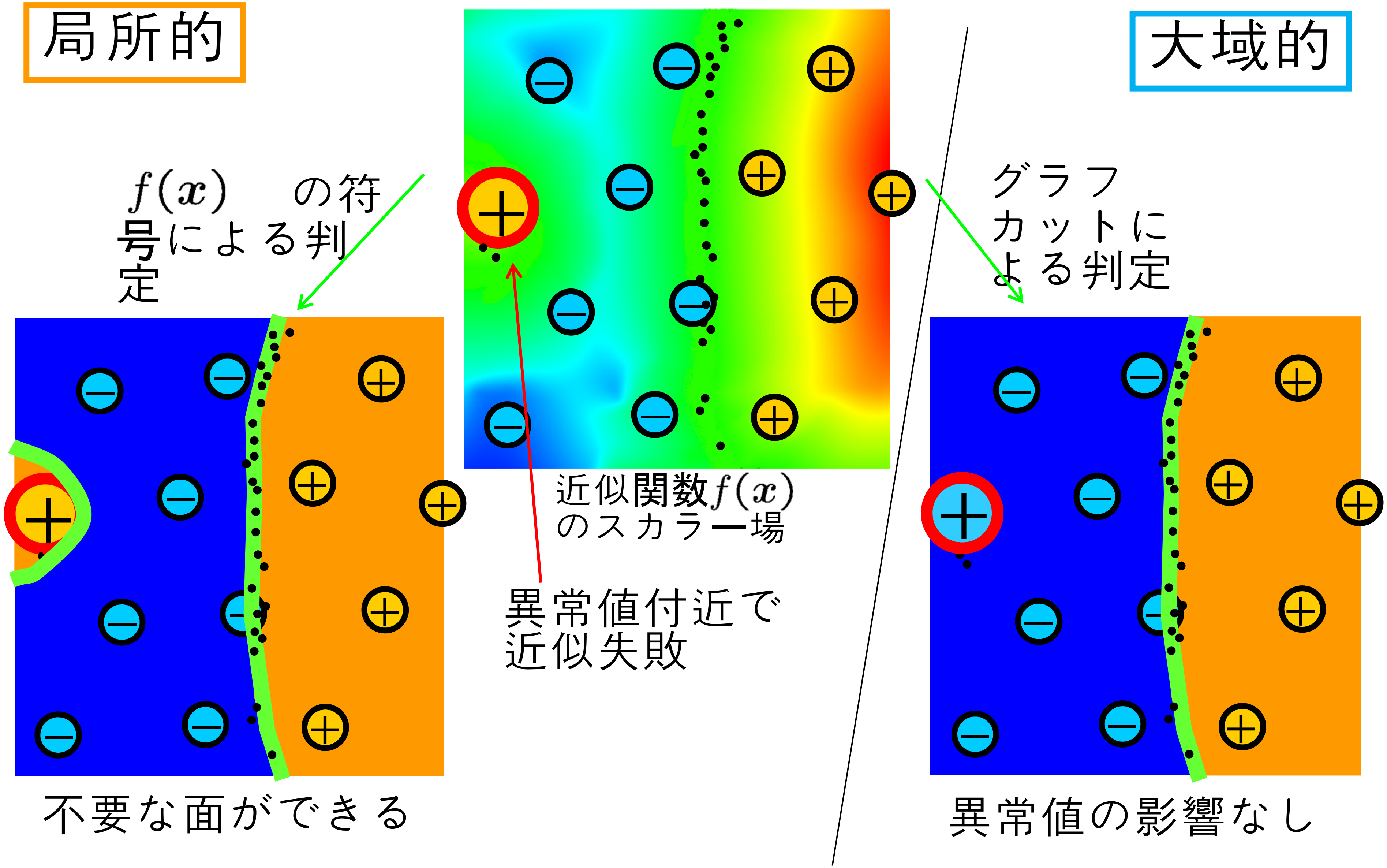
Spherical Covering

- Octree-based ex. MPU [Ohtake *et al.* 03]

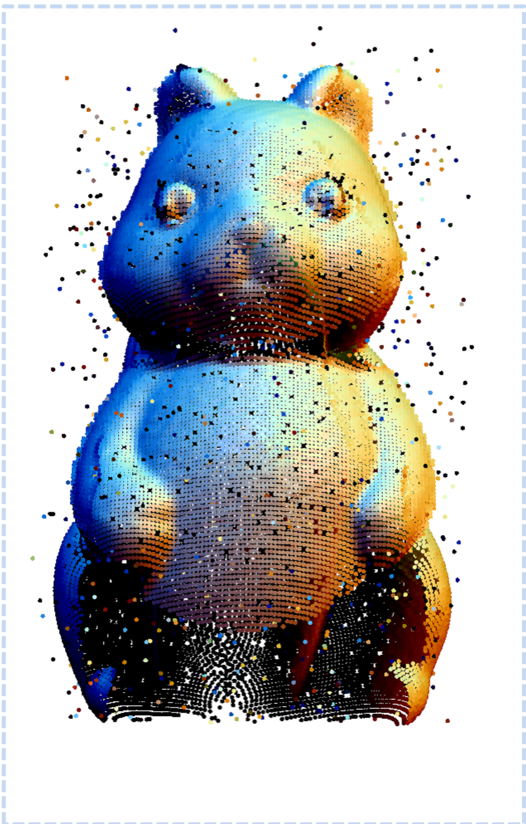
1. One support for one octant
2. Support expansion and elimination



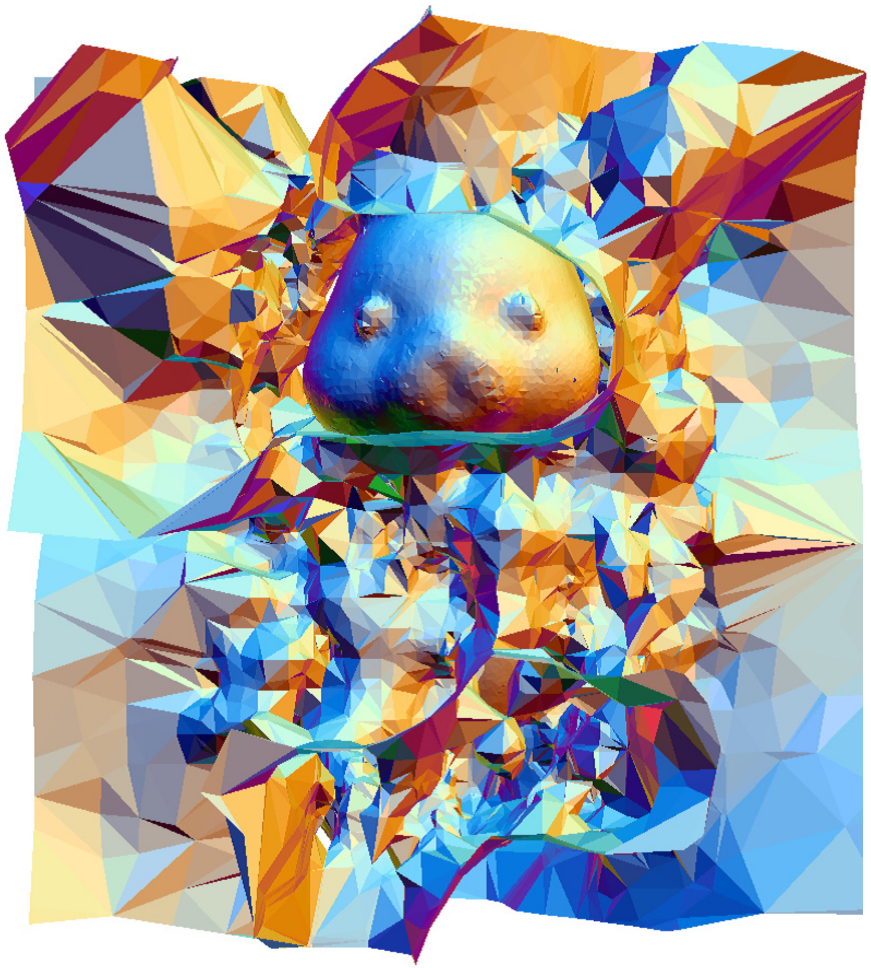
局所的判定と大域的判定



Effects of Graph-cut



Sampling points



PU

PU+ Graph-cut

Differential Operators for PU

- Vector at point $\boldsymbol{x}$:

$$\boldsymbol{v}(\boldsymbol{x}) = \frac{\sum_i w_i(\boldsymbol{x})\boldsymbol{v}_i}{\sum_i w_i(\boldsymbol{x})}$$

- $\frac{4\pi r_i^3}{3}\text{Div}\,(\boldsymbol{v}_i) \equiv \int_{\boldsymbol{x}\in s_i} \nabla\cdot\boldsymbol{v}(\boldsymbol{x})\mathrm{d}\boldsymbol{x}$

divergence theorem

$$= \int_{\boldsymbol{x}\in\partial s_i} \boldsymbol{v}(\boldsymbol{x})\cdot\boldsymbol{n}(\boldsymbol{x})\mathrm{d}\boldsymbol{x}$$

$$\approx \frac{4\pi r_i^2}{\boxed{\sum_j \int_{\boldsymbol{x}\in\delta_{i,j}} d\boldsymbol{x}}} \sum_j \int_{\boldsymbol{x}\in\delta_{i,j}} \boldsymbol{v}_j\cdot\boldsymbol{n}(\boldsymbol{x})\mathrm{d}\boldsymbol{x}$$

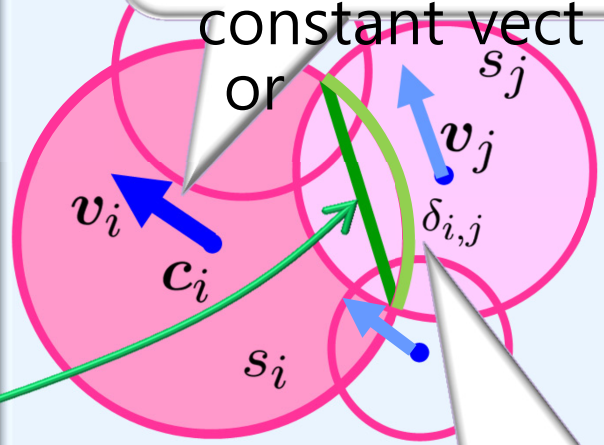
$$\varphi_{i,j} = \frac{\boxed{D_{i,j}}}{\|\boldsymbol{c}_j - \boldsymbol{c}_i\|}$$

$$\text{Div}(\boldsymbol{v}_i) = \frac{3}{r_i\boxed{S_i}}\sum_j \varphi_{i,j}\ \boldsymbol{v}_j\cdot(\boldsymbol{c}_j - \boldsymbol{c}_i)$$

at center $\boldsymbol{c}_i$

Assumption 1

support s_i has a constant vector $\boldsymbol{v}_i$



Assumption 2

$\boxed{\delta_{i,j}}$ is covered only by s_i and s_j

Differential Operators for PU

$$\text{Div}(\boldsymbol{v}_i) = \frac{3}{r_i S_i} \sum_j \varphi_{i,j} \; \boldsymbol{v}_j \cdot (\boldsymbol{c}_j - \boldsymbol{c}_i)$$

$$\text{Lap}(\boldsymbol{v}_i) \equiv \text{Div}(\text{Grad}(\boldsymbol{v}_i))$$

Replace $\boldsymbol{v}$ *with* $\text{Grad}(\boldsymbol{v})$

$$\text{Lap}(\boldsymbol{v}_i) = \frac{3}{r_i S_i} \sum_j \varphi_{i,j} \text{Grad}(\boldsymbol{v}_j) \cdot (\boldsymbol{c}_j - \boldsymbol{c}_i)$$

Taylor expansion

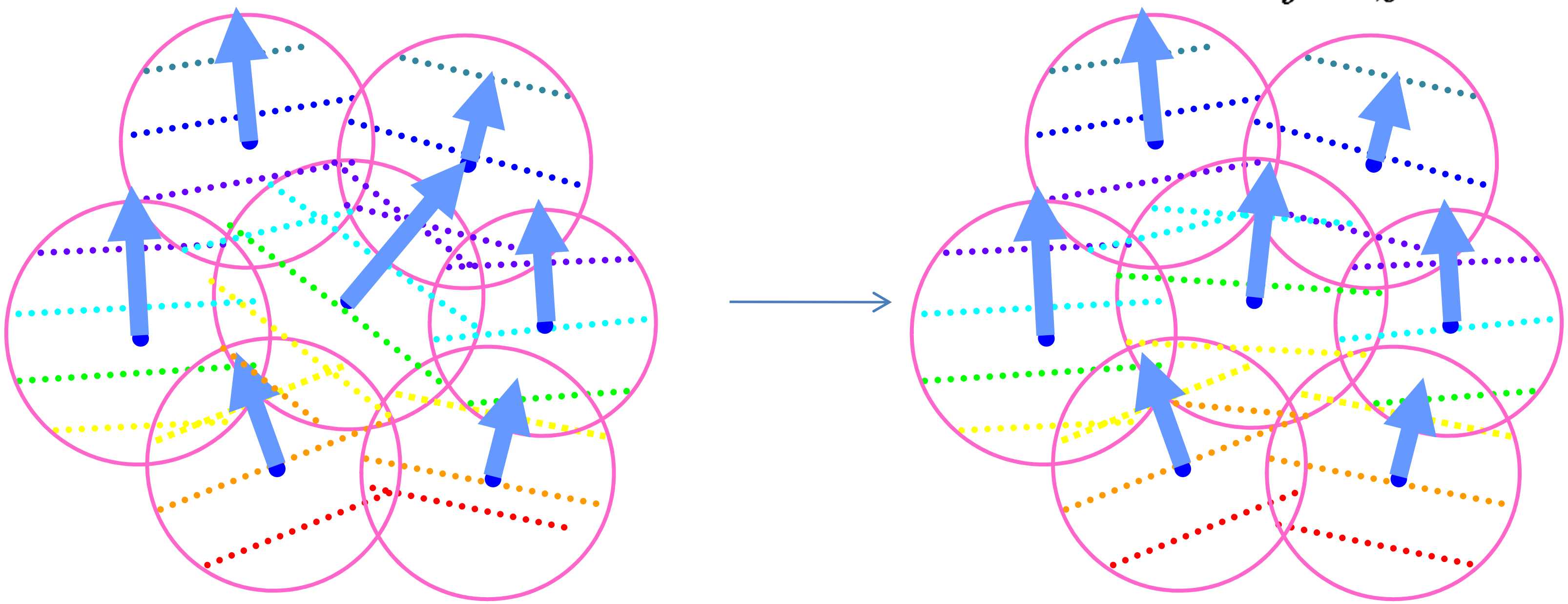
$$\boldsymbol{v}_i \approx \boldsymbol{v}_j + \text{Grad}(\boldsymbol{v}_j) \cdot (\boldsymbol{c}_i - \boldsymbol{c}_j)$$

$$\text{Lap}(\boldsymbol{v}_i) = \frac{3}{r_i S_i} \sum_j \varphi_{i,j} (\boldsymbol{v}_j - \boldsymbol{v}_i)$$

$$\text{Lap}(f(\boldsymbol{c}_i)) = \frac{3}{r_i S_i} \sum_j \varphi_{i,j} (f(\boldsymbol{c}_j) - f(\boldsymbol{c}_i))$$

Update of $\{\boldsymbol{\alpha}_i\}$

- Smoothing the gradients
- $\mathrm{Lap}(\hat{\boldsymbol{v}}_i) = 0$
- Update $\boldsymbol{\alpha}_i$ to a weighted average of the gradient vector $\hat{\boldsymbol{\alpha}}_i = \dfrac{\sum_j \varphi_{i,j} \boldsymbol{v}_j}{\sum_j \varphi_{i,j}}$



Update of $\{\beta_i\}$

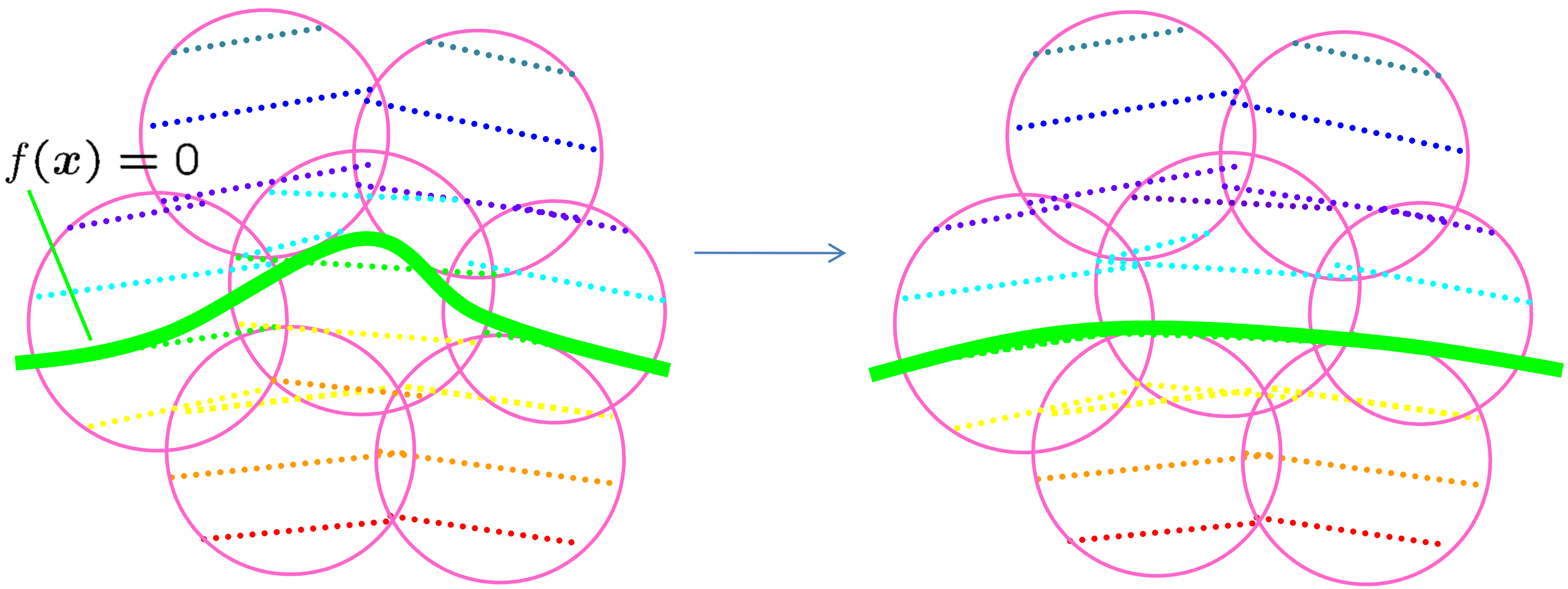
- Make isosurfaces consistent
- $\text{Grad}\left(f(\boldsymbol{c}_i)\right) = \widehat{\boldsymbol{v}}_i$

  $\text{Lap}\left(f(\boldsymbol{c}_i)\right) = \text{Div}(\widehat{\boldsymbol{v}}_i)$

weak formulation

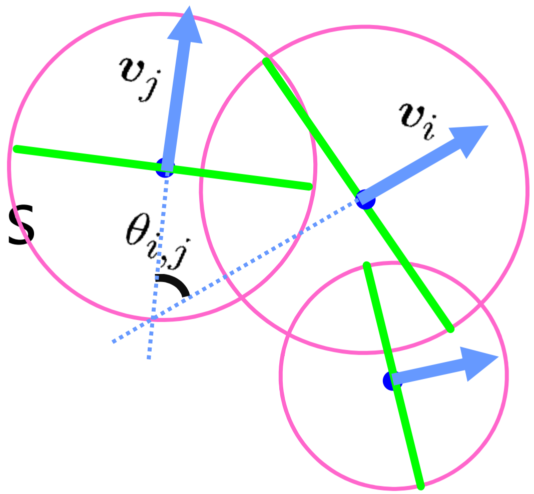
$$\widehat{\beta}_i = \frac{\sum_j \varphi_{i,j}\left(\widehat{\boldsymbol{\alpha}}_j \cdot (\boldsymbol{c}_i - \boldsymbol{c}_j) + \beta_j\right)}{\sum_j \varphi_{i,j}}$$

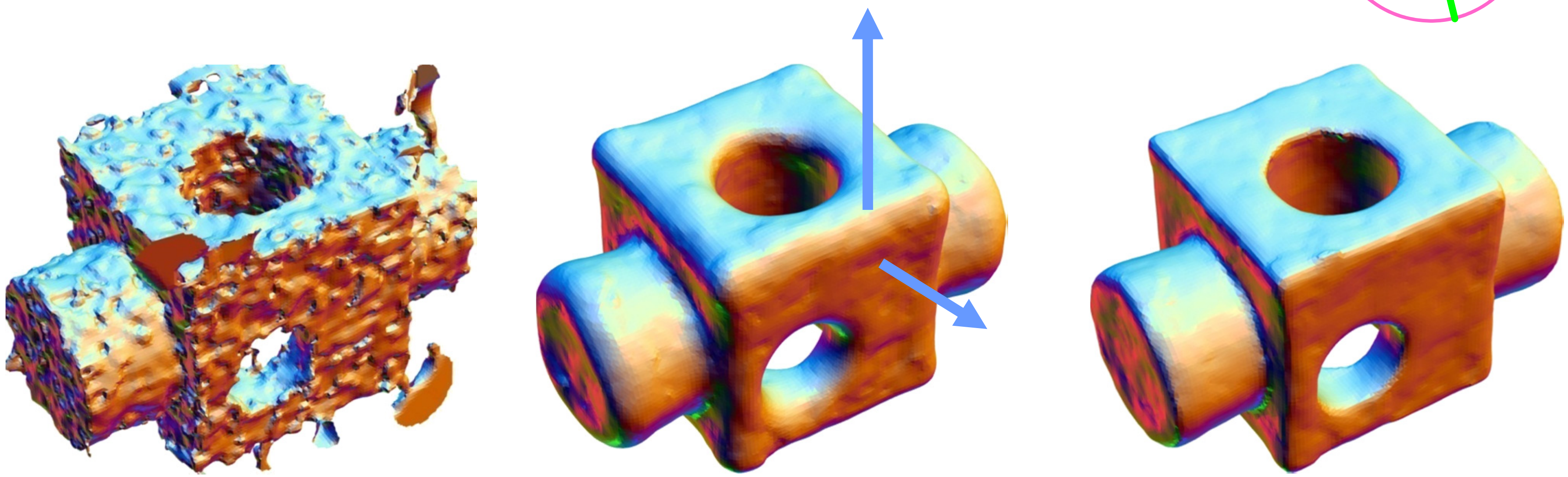
(Assumption: $f(\boldsymbol{c}_i) = \beta_i$)



Anisotropic Smoothing

- To avoid loss of edges
- Taking anisotropic factor $\psi_{i,j} = \dfrac{1}{1+\theta_{i,j}^2}$ into updating of local functions
  - Difference of gradients

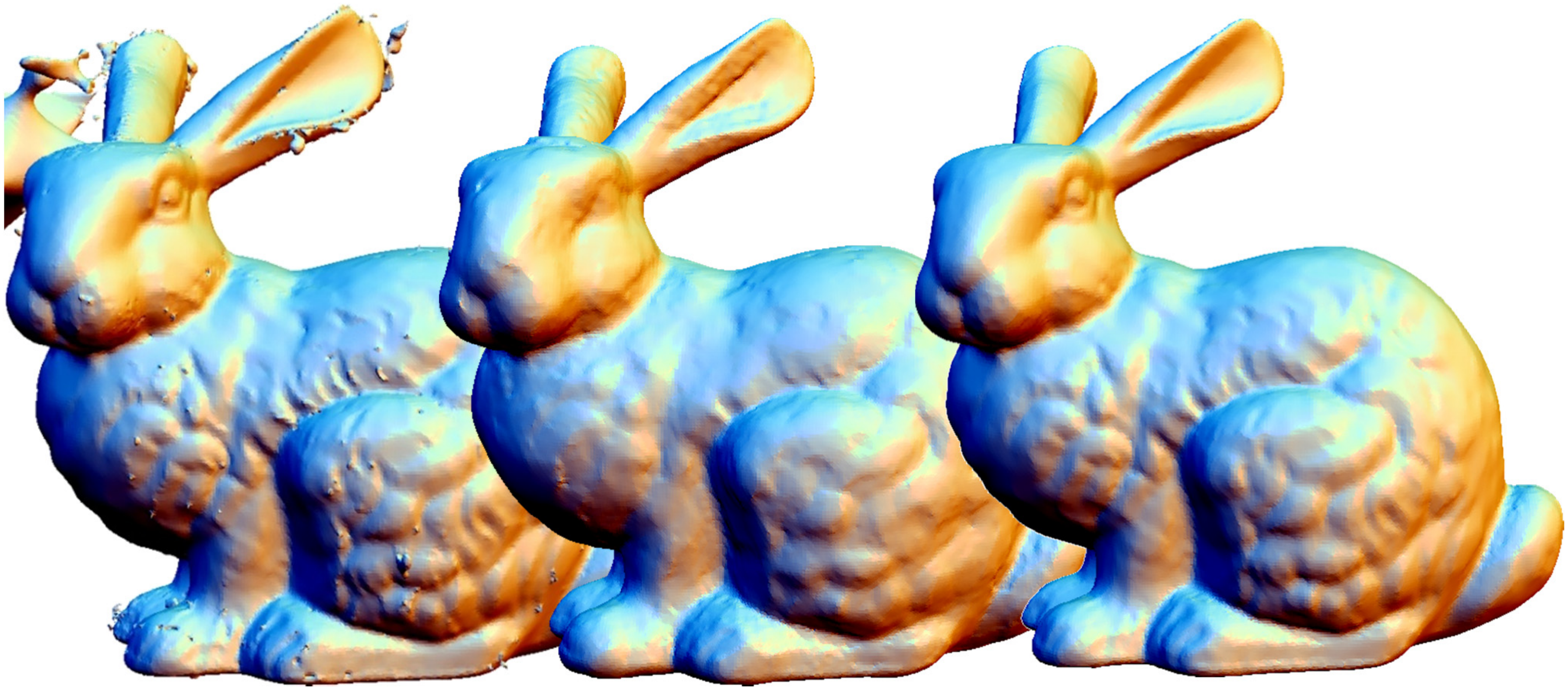




PU + Smoothing + Anisotropy

Detail Preservation

- To avoid oversmoothing
- Fit the surface to the sampling points



PU

Oversmoothe
d

Detail preserv
ed

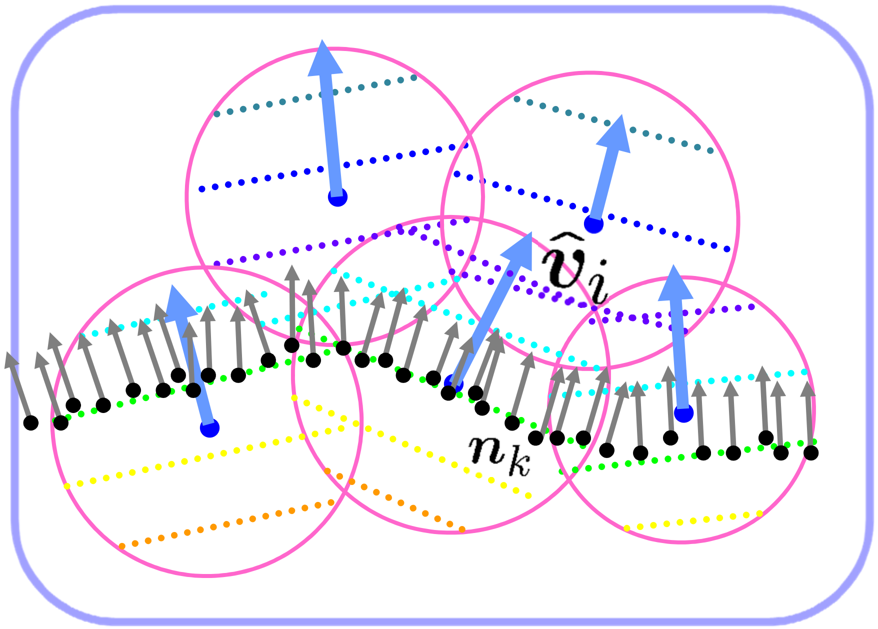
Detail Preservation

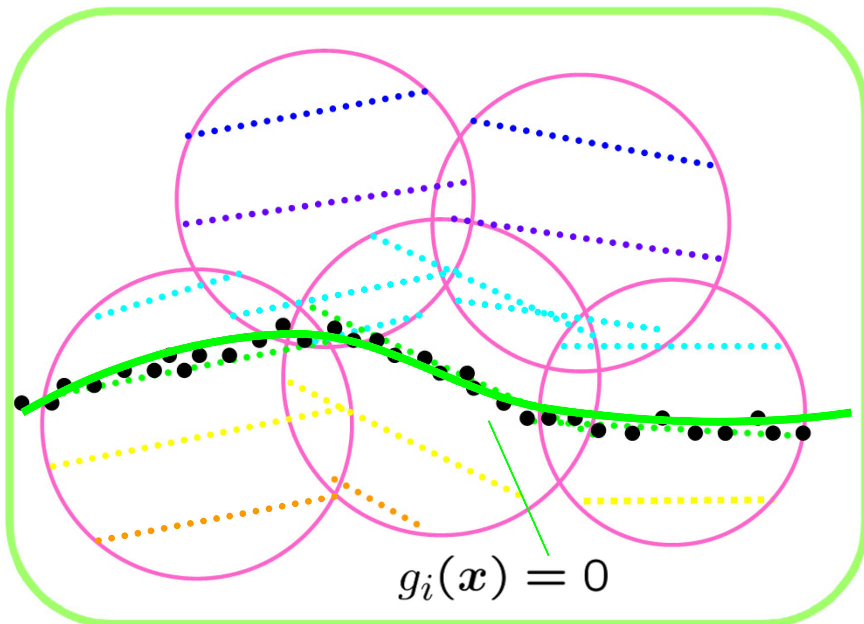
- Smoothing terms + Fitting terms
  - Update of $\{\boldsymbol{\alpha}_i\}$: Fit the gradient to normals

$$\|\mathrm{Lap}(\widehat{\boldsymbol{v}}_i)\|^2 + \lambda_n \sum_k w_i(\boldsymbol{p}_k)\,\|\widehat{\boldsymbol{v}}_i - \boldsymbol{n}_k\|^2 \to \min$$

  - Update of $\{\beta_i\}$: Fit the 0-level set to sampling points

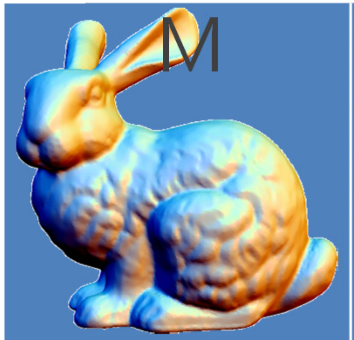
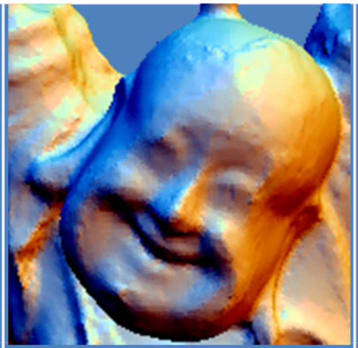
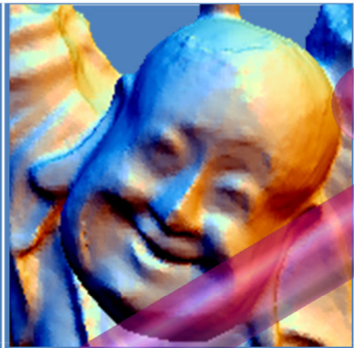
$$(\mathrm{Lap}(f(\boldsymbol{c}_i)) - \mathrm{Div}(\widehat{\boldsymbol{v}}_i))^2 + \lambda_p \sum_k w_i(\boldsymbol{p}_k)\, g_i^2(\boldsymbol{p}_k) \to \min$$





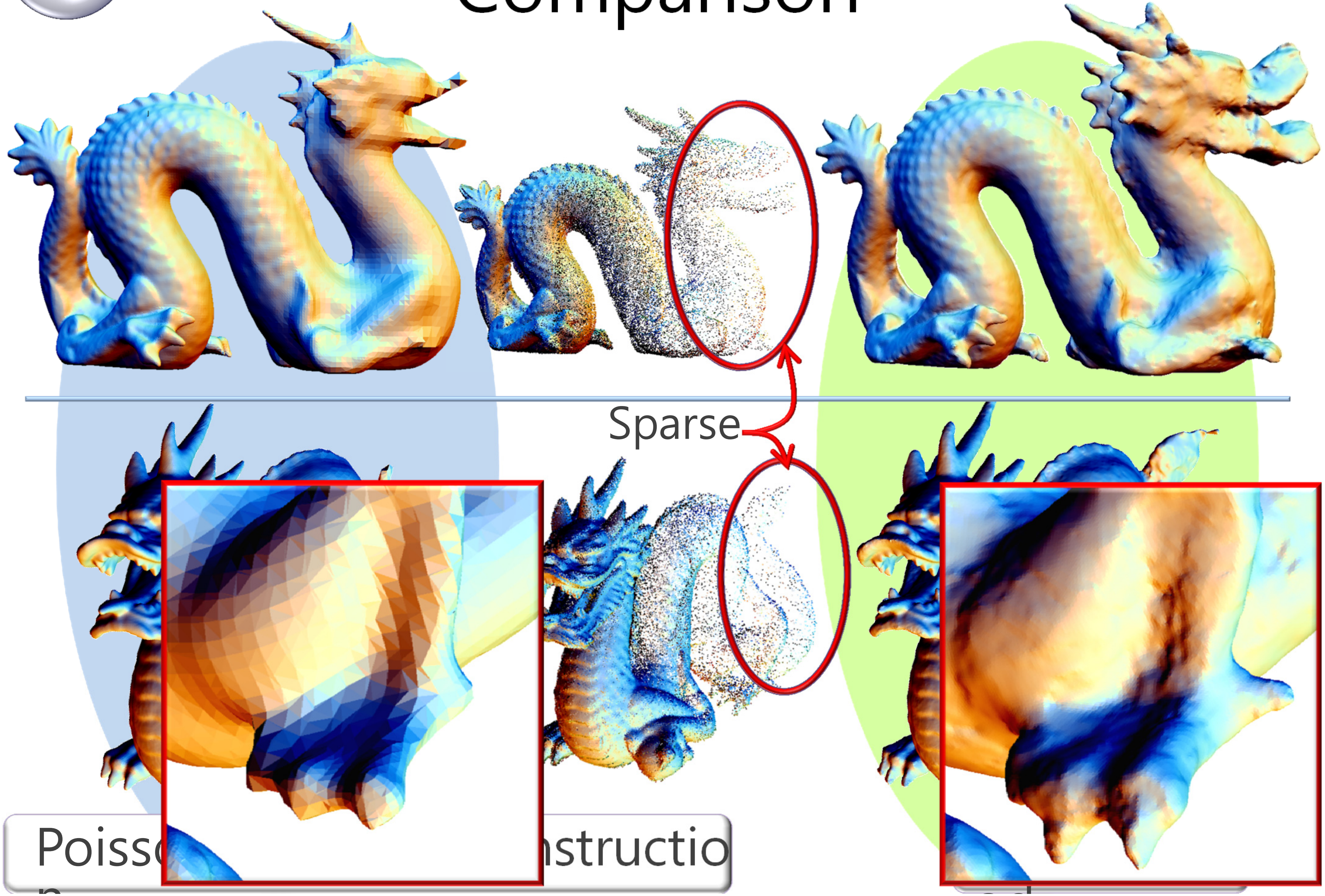
Performance

Intel Core2Duo 3.0GHz with 8.0GB RA
M

Model # points ε $(\times 10^{-3})$	 Bunny 362K 2.5	 Buddha rough 3.2M 2.5	 Buddha precis e 3.2M 1.25	 Armadillo 2.3M 1.0
# supports	176K	540K	2.94M	7.35M
Time [min:sec]	0:51	3:29	17:84	45:52
Peak RAM	74M	234M	1.23G	3.24G

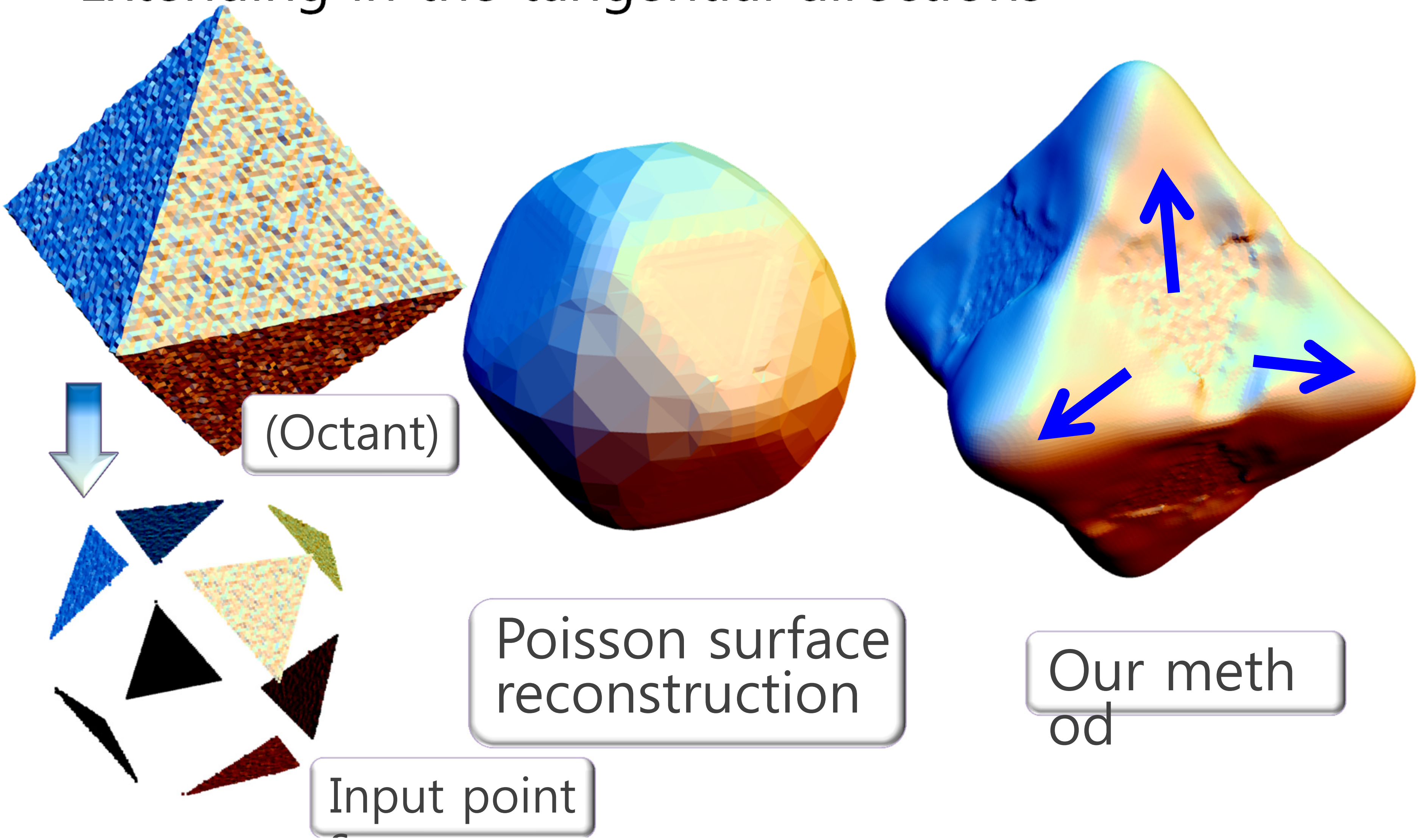
orts

Comparison



Surfaces' Properties

- Extending in the tangential directions



Polygonization of Isosurfaces

- Marching Cubes [Lorensen and Cline 87]