Visibility Prior

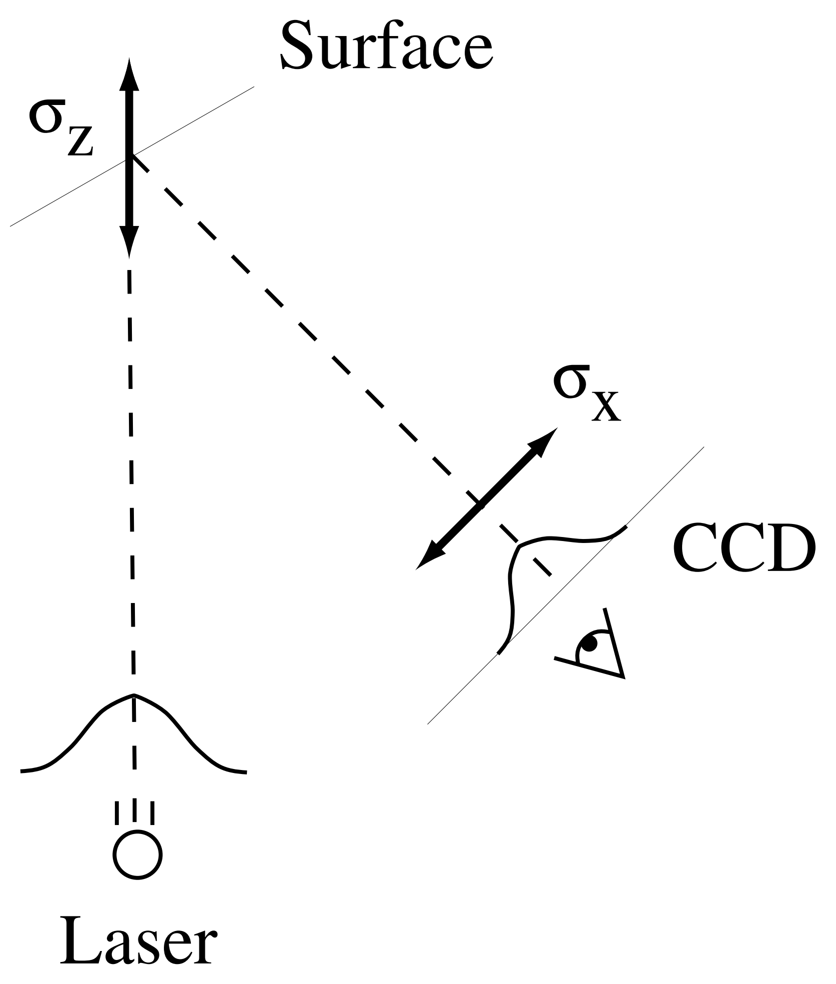
- Use exterior space to help in reconstruction
- Output: surface mesh, optionally watertight
- Class shape: general
- Main approaches:
    - Scanner visibility
    - Point set visibility
    - Parity

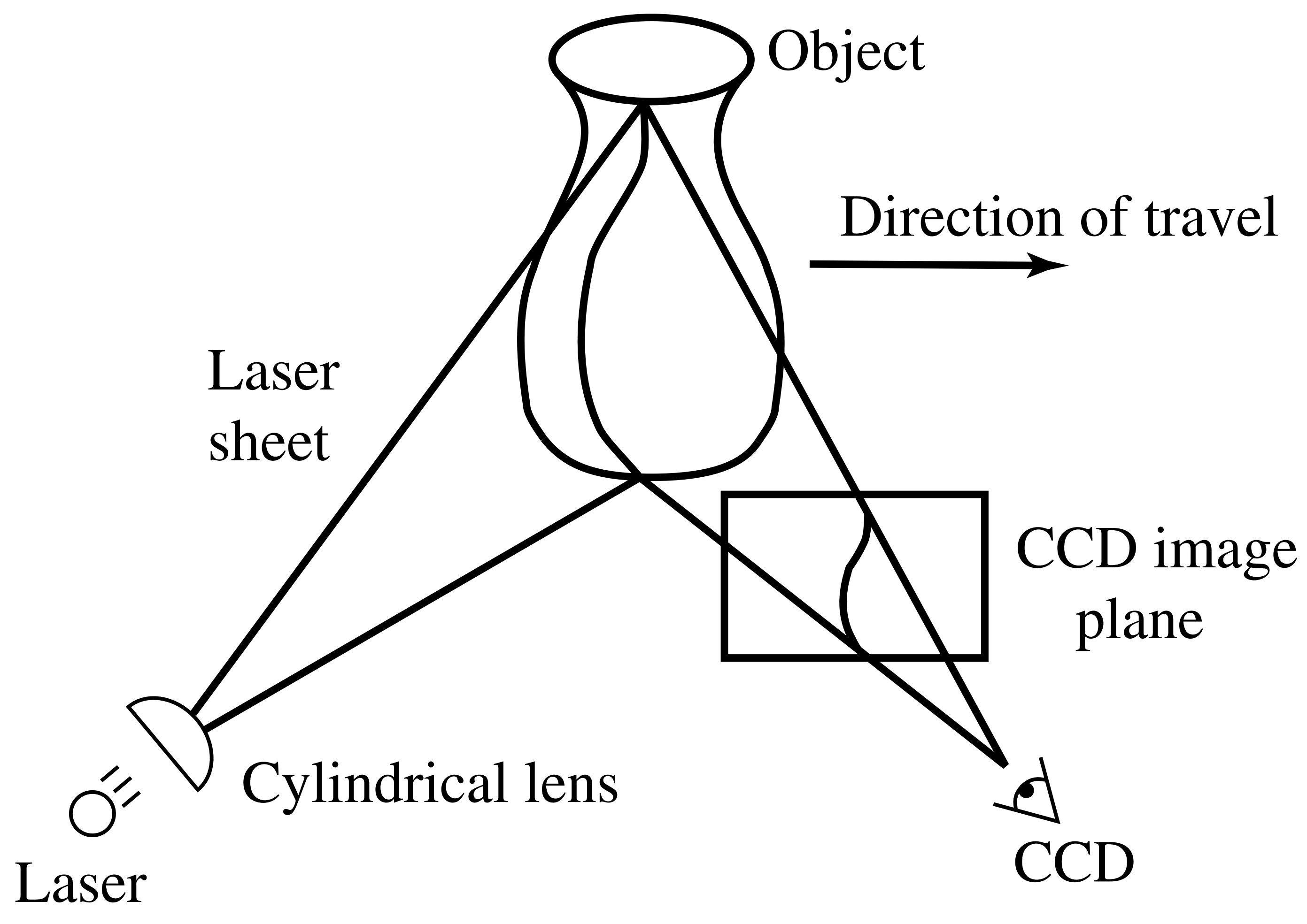
VRIP

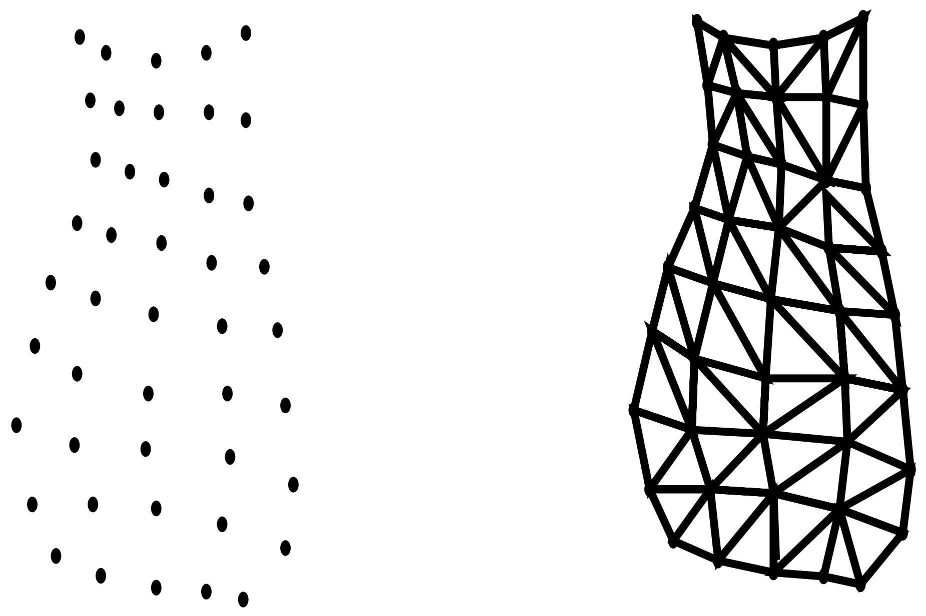
- Sequential integration of range scans

[Curless & Levoy SIG'96]

- Take advantage of acquisition:
  - confidence, space carving



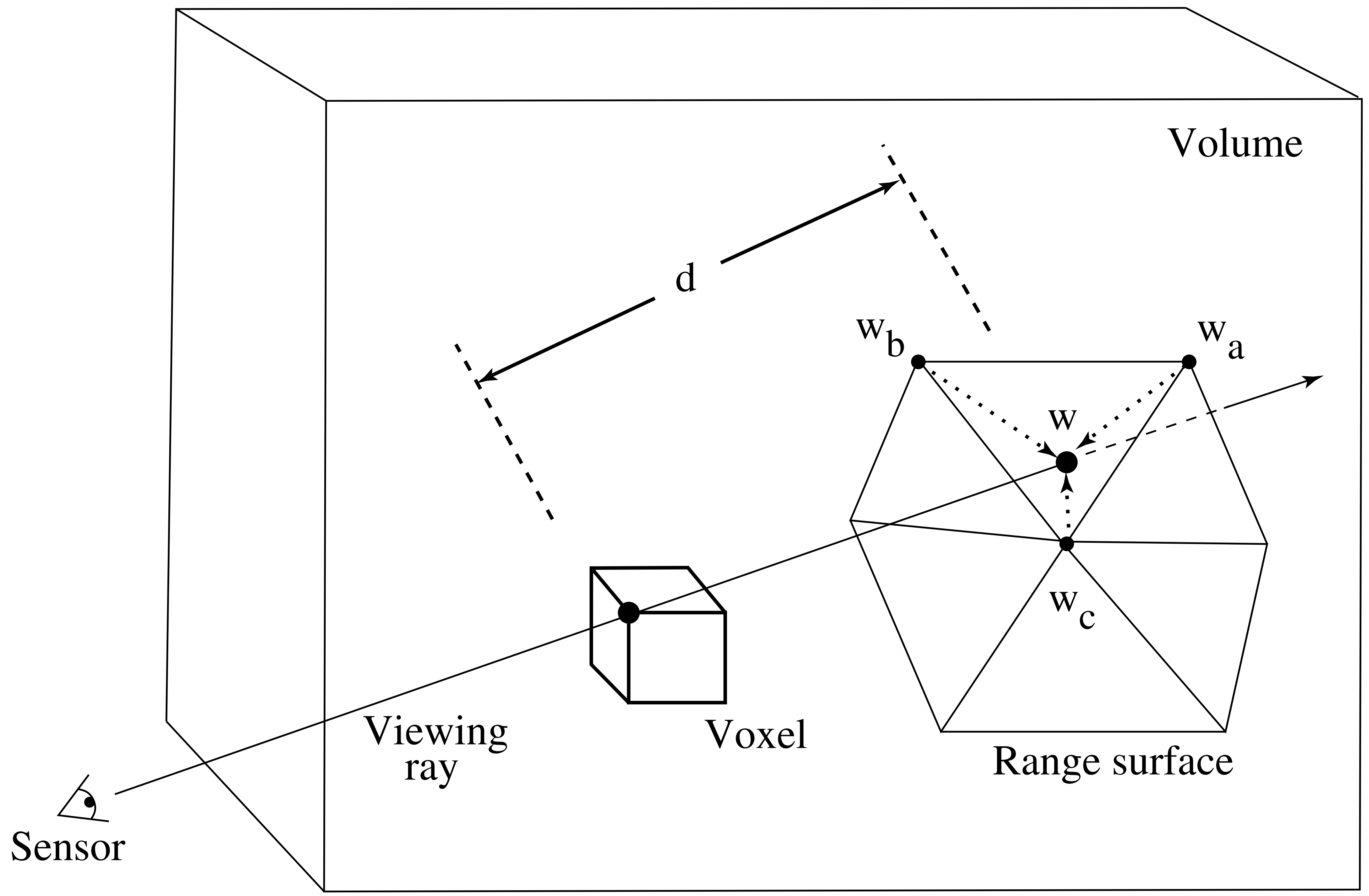




VRIP

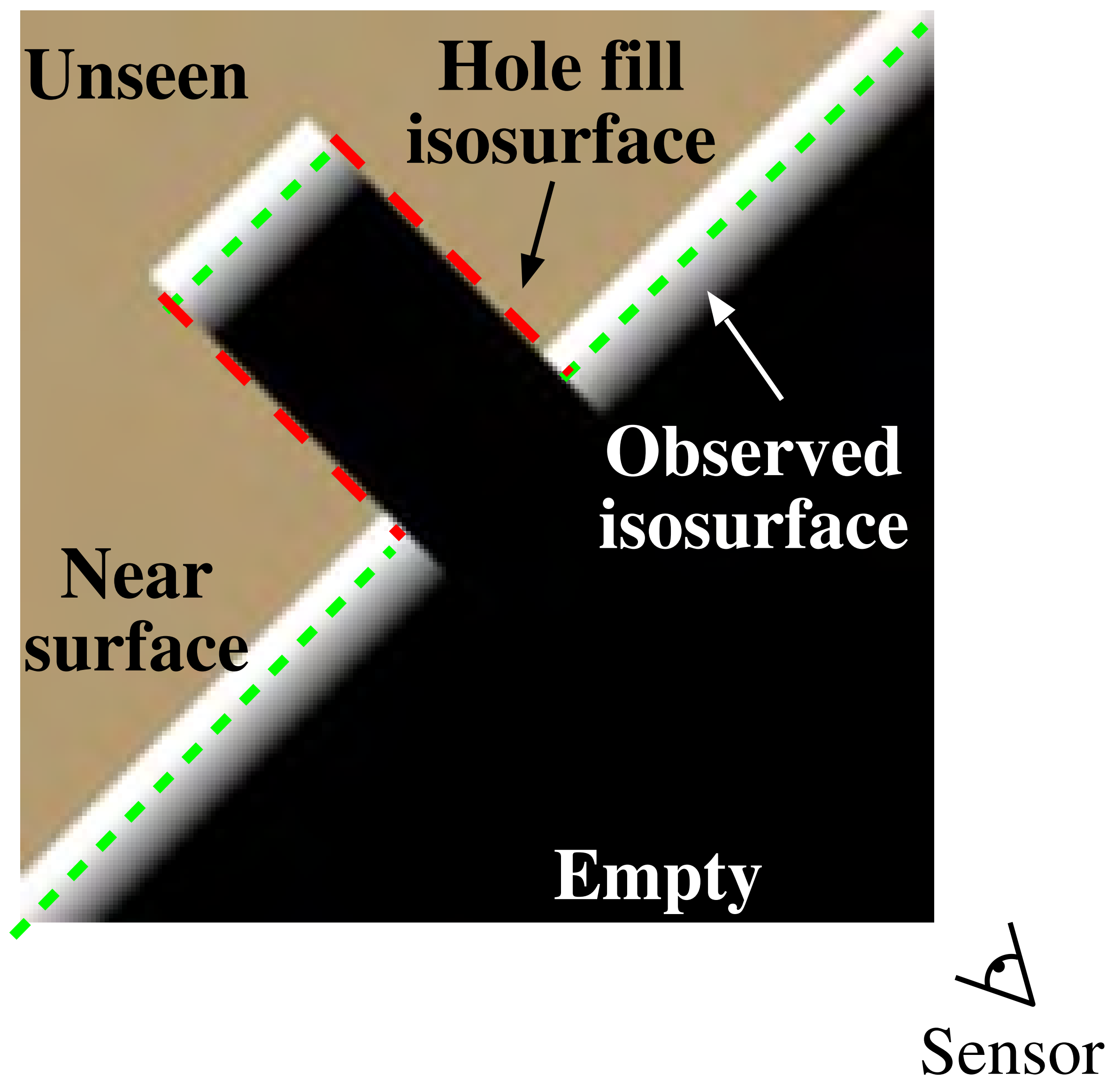
- Construction of signed distance field

[Curless & Levoy SIG'96]



VRIP

- Handling missing data: space carving [Curless & Levoy SIG'96]
- Interface between empty and unseen space considered to be the surface



Total Variation-L1

- More robust means of range scan integration
- Minimize gradient magnitude of signed distance field
- Use l1 norm for data fitting

[Zach et al. ICCV07]

Total Variation-L1

- More robust means of range scan integration
- Minimize gradient magnitude of signed distance field
- Use l1 norm for data fitting

[Zach et al. ICCV07]

$$E = \int_{\Omega} \left\{ |\nabla u| + \lambda \sum_{i \in \mathcal{I}(\vec{x})} w_i(\vec{x}) |u - f_i| \right\} d\vec{x}$$

Total Variation-L1

- More robust means of range scan integration
- Minimize gradient magnitude of signed distance field
- Use l1 norm for data fitting

[Zach et al. ICCV07]

$$E = \int_{\Omega} \left\{ |\nabla u| + \lambda \sum_{i \in \mathcal{I}(\vec{x})} w_i(\vec{x}) |u - f_i| \right\} d\vec{x}$$

impose smoothness

Total Variation-L1

- More robust means of range scan integration
- Minimize gradient magnitude of signed distance field
- Use l1 norm for data fitting

[Zach et al. ICCV07]

$$E = \int_\Omega \left\{ |\nabla u| + \lambda \sum_{i \in \mathcal{I}(\vec{x})} w_i(\vec{x}) |u - f_i| \right\} d\vec{x}$$

impose smoothness

robustness to outliers

Total Variation-L1

- More robust means of range scan integration
- Minimize gradient magnitude of signed distance field
- Use l1 norm for data fitting

[Zach et al. ICCV07]

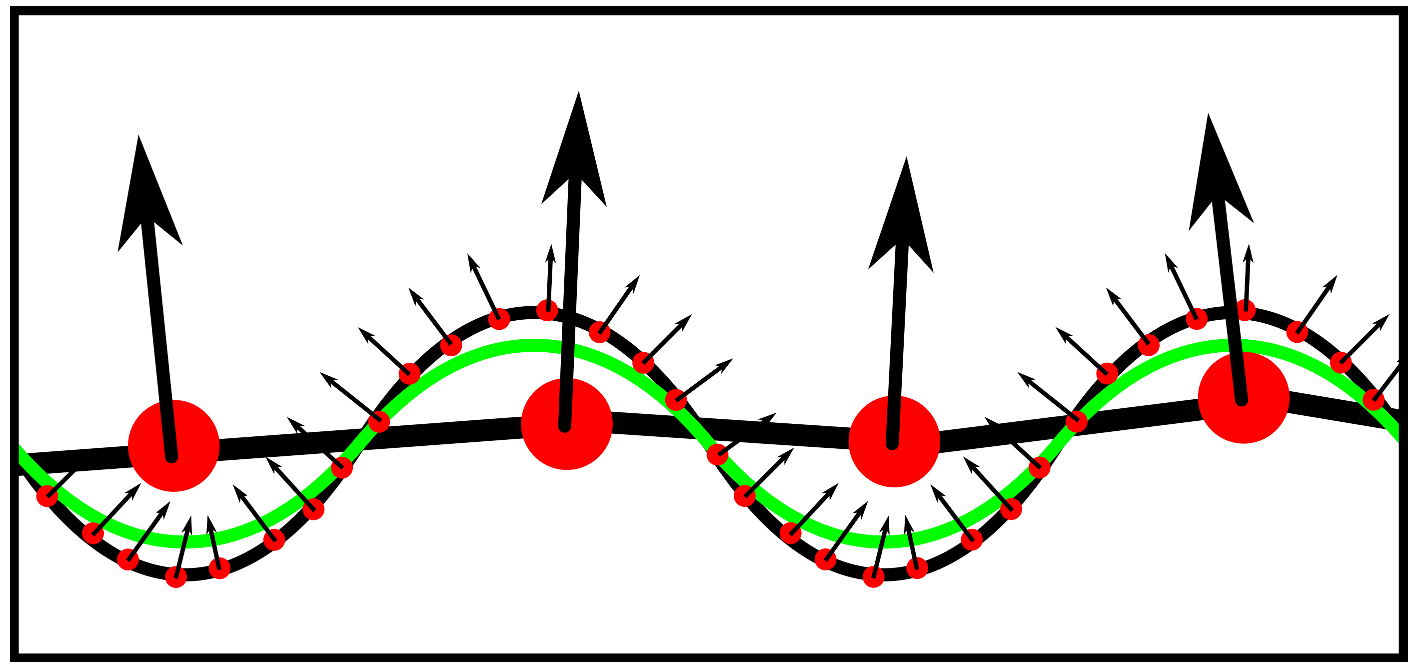
$$E = \int_{\Omega} \left\{ \underbrace{|\nabla u|}_{\text{impose smoothness}} + \lambda \sum_{i \in \mathcal{I}(\vec{x})} w_i(\vec{x}) \underbrace{|u - f_i|}_{\text{robustness to outliers}} \right\} d\vec{x}$$

Multi-resolution Scan Merging

- Handle merging of scans at different resolutions [Fuhrmann & Goesele SIGA'11]

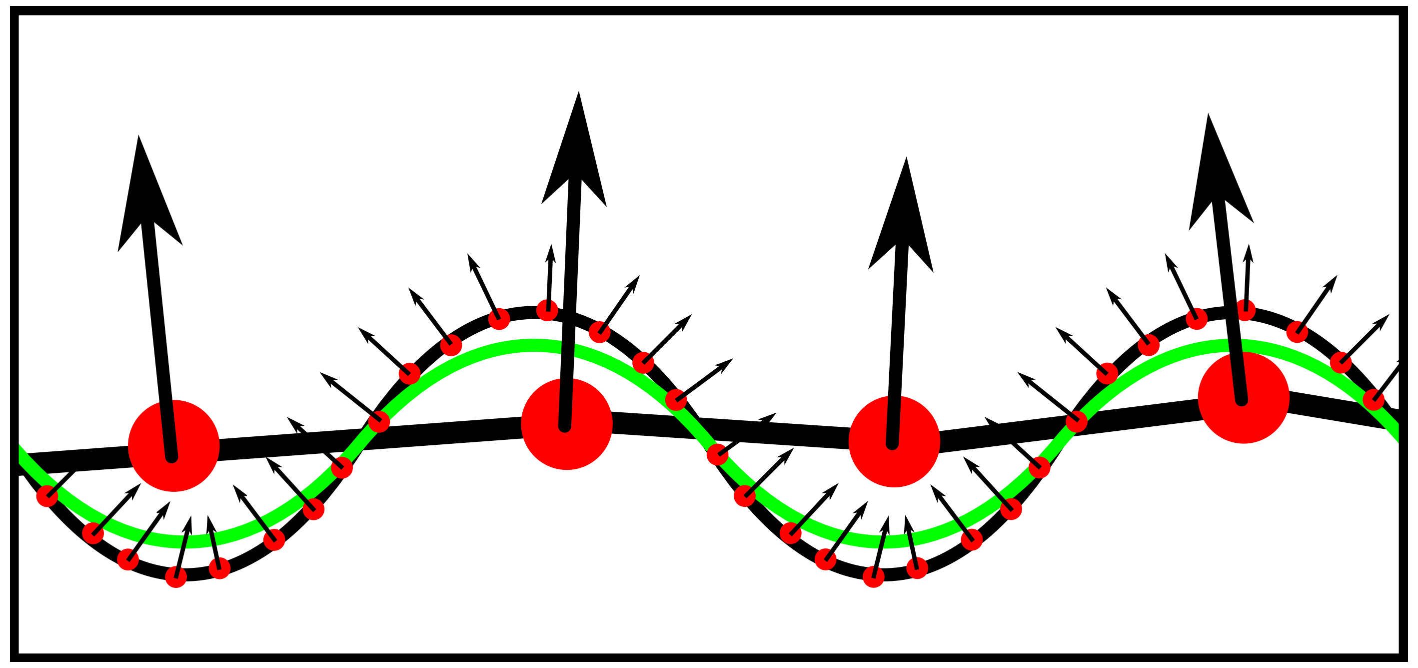
Multi-resolution Scan Merging

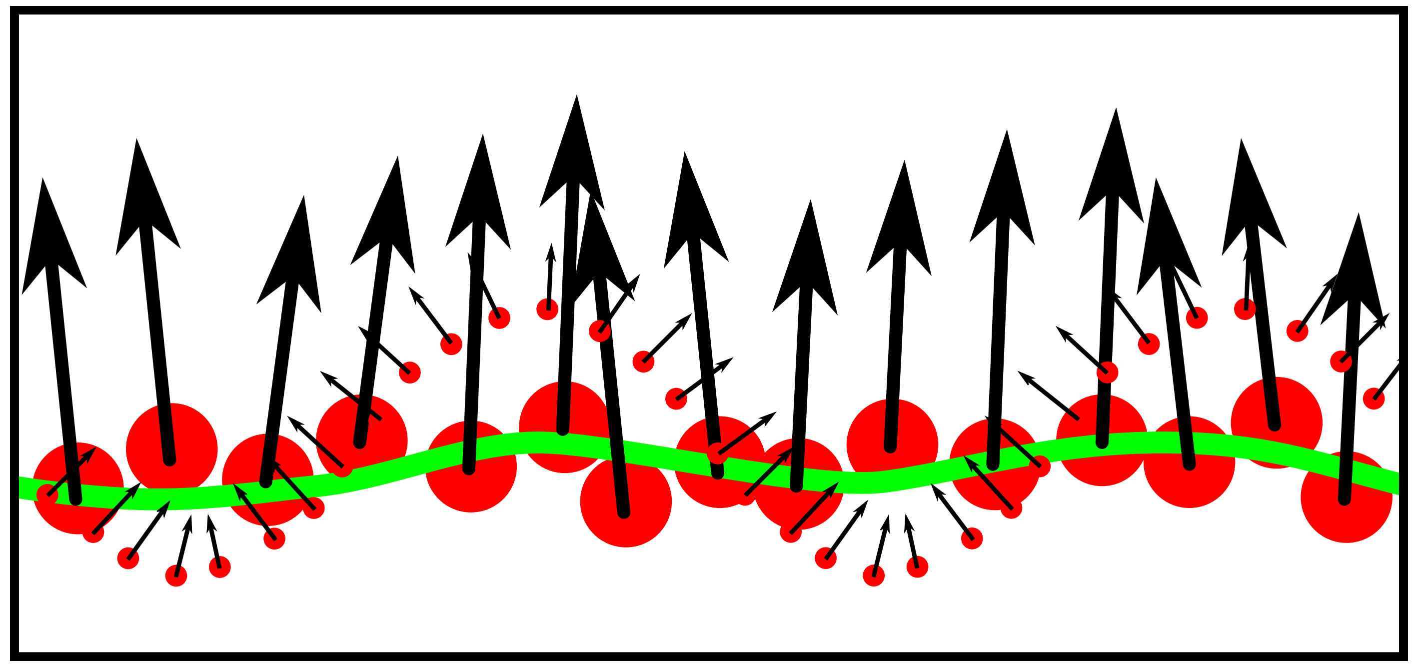
- Handle merging of scans at different resolutions [Fuhrmann & Goesele SIGA'11]



Multi-resolution Scan Merging

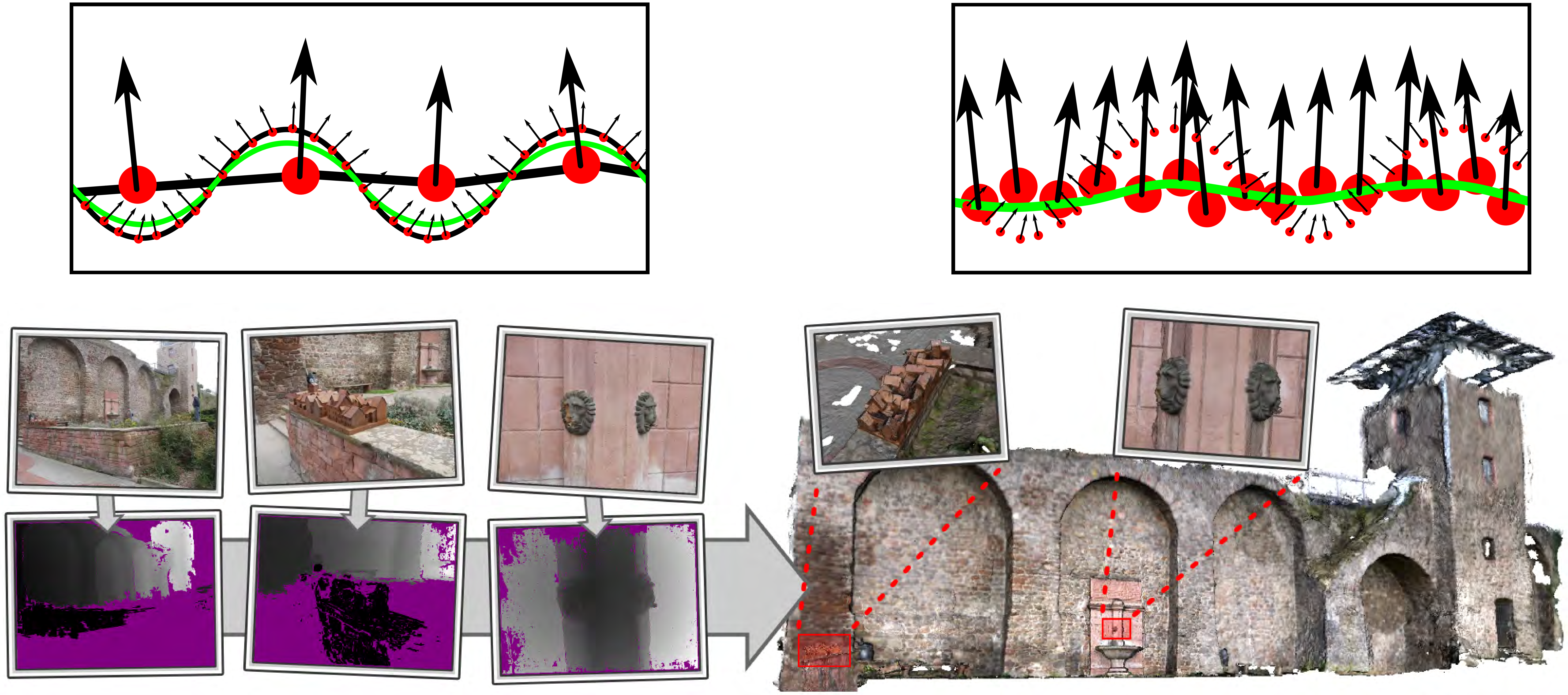
- Handle merging of scans at different resolutions [Fuhrmann & Goesele SIGA'11]





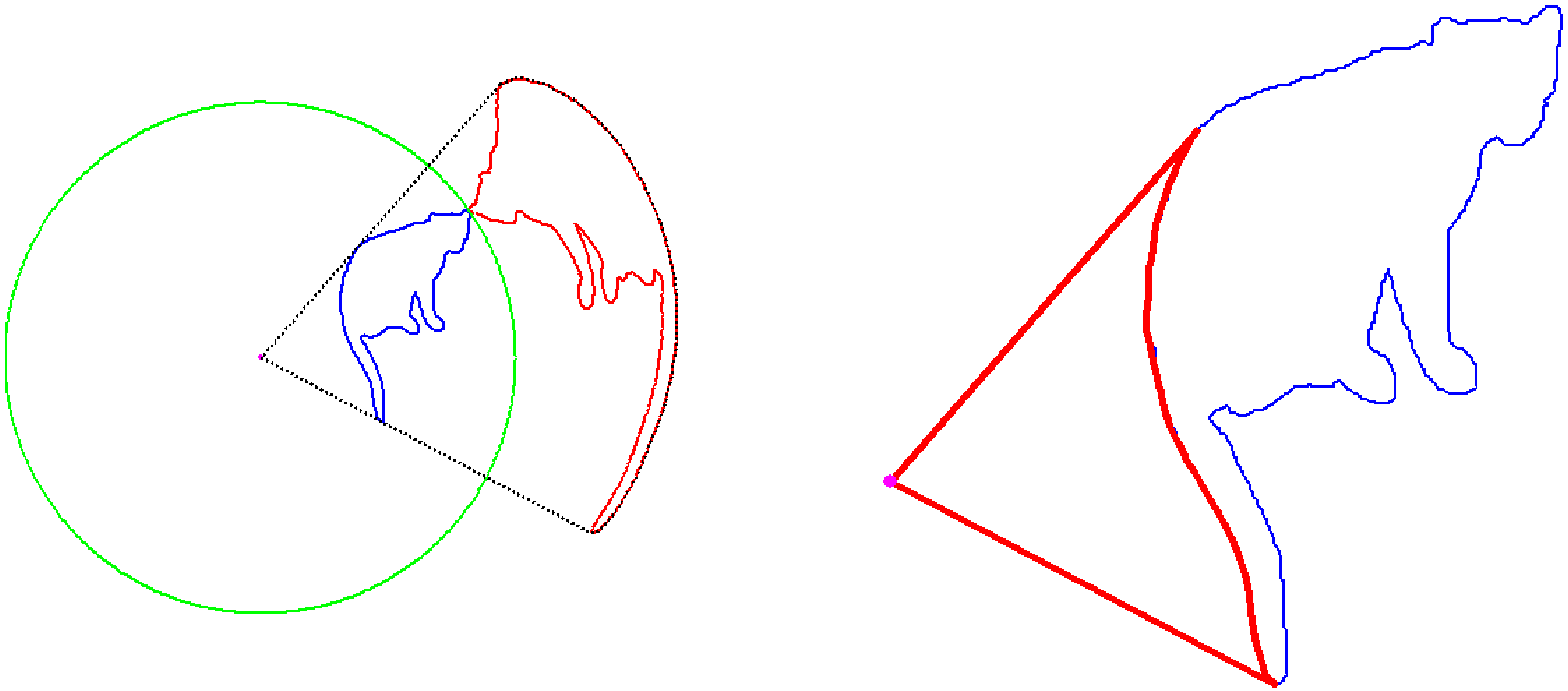
Multi-resolution Scan Merging

- Handle merging of scans at different resolutions [Fuhrmann & Goesele SIGA'11]



Point Set Visibility

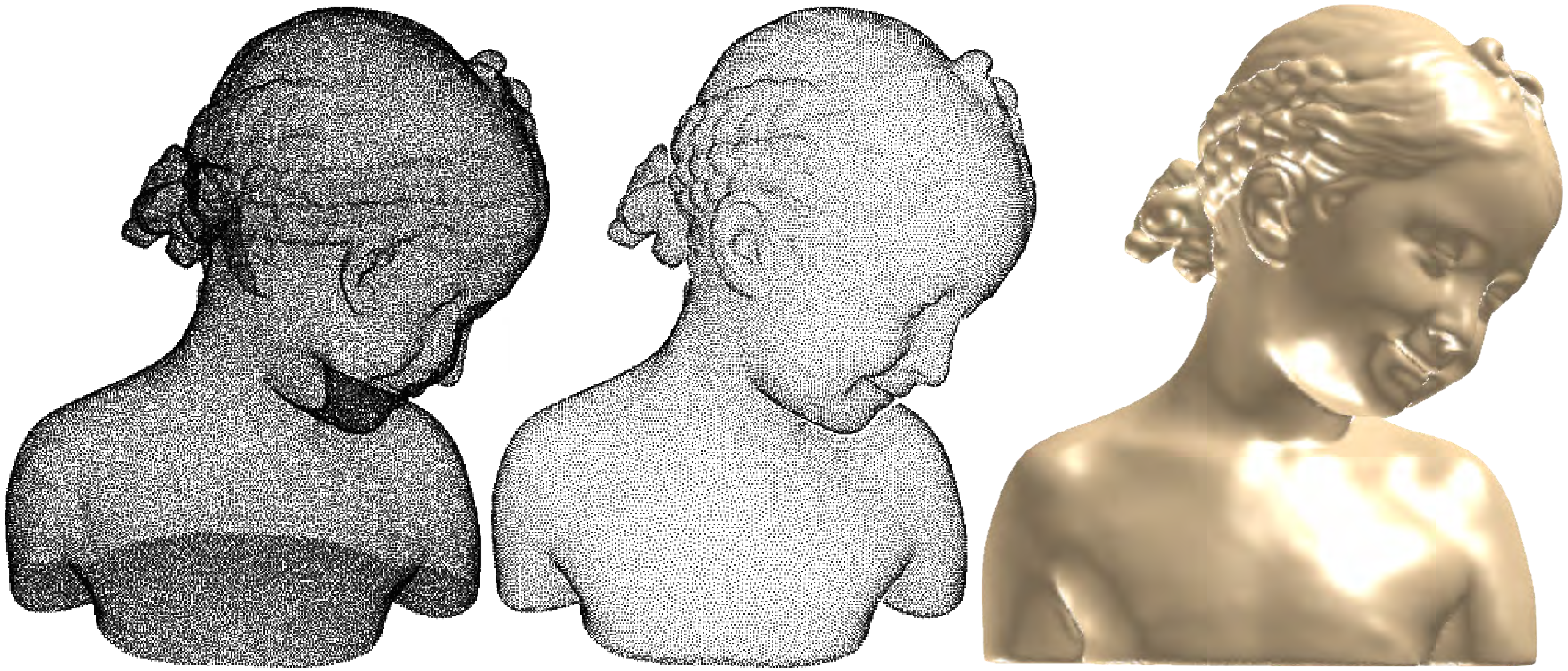
- Determine visible points from arbitrary views [Katz et al. SIG'07]
- Perform spherical inversion of point set from the given view
- Take convex hull of viewpoint & spherical inversion: points on the convex hull are considered visible



Point Set Visibility

- Use for on-the-fly reconstruction

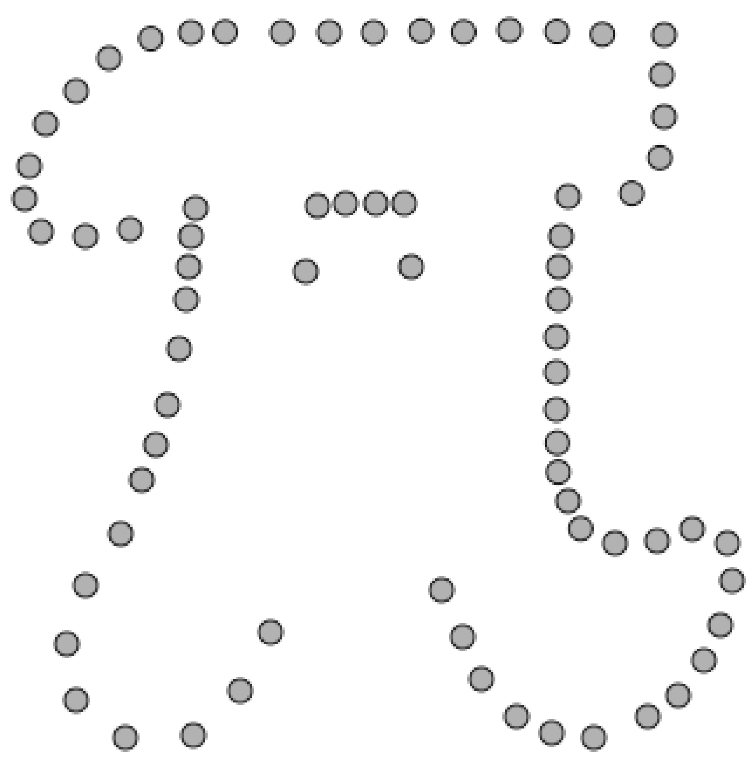
[Katz et al. SIG'07]



Cone Carving

- Use visibility cones centered at points on the surface

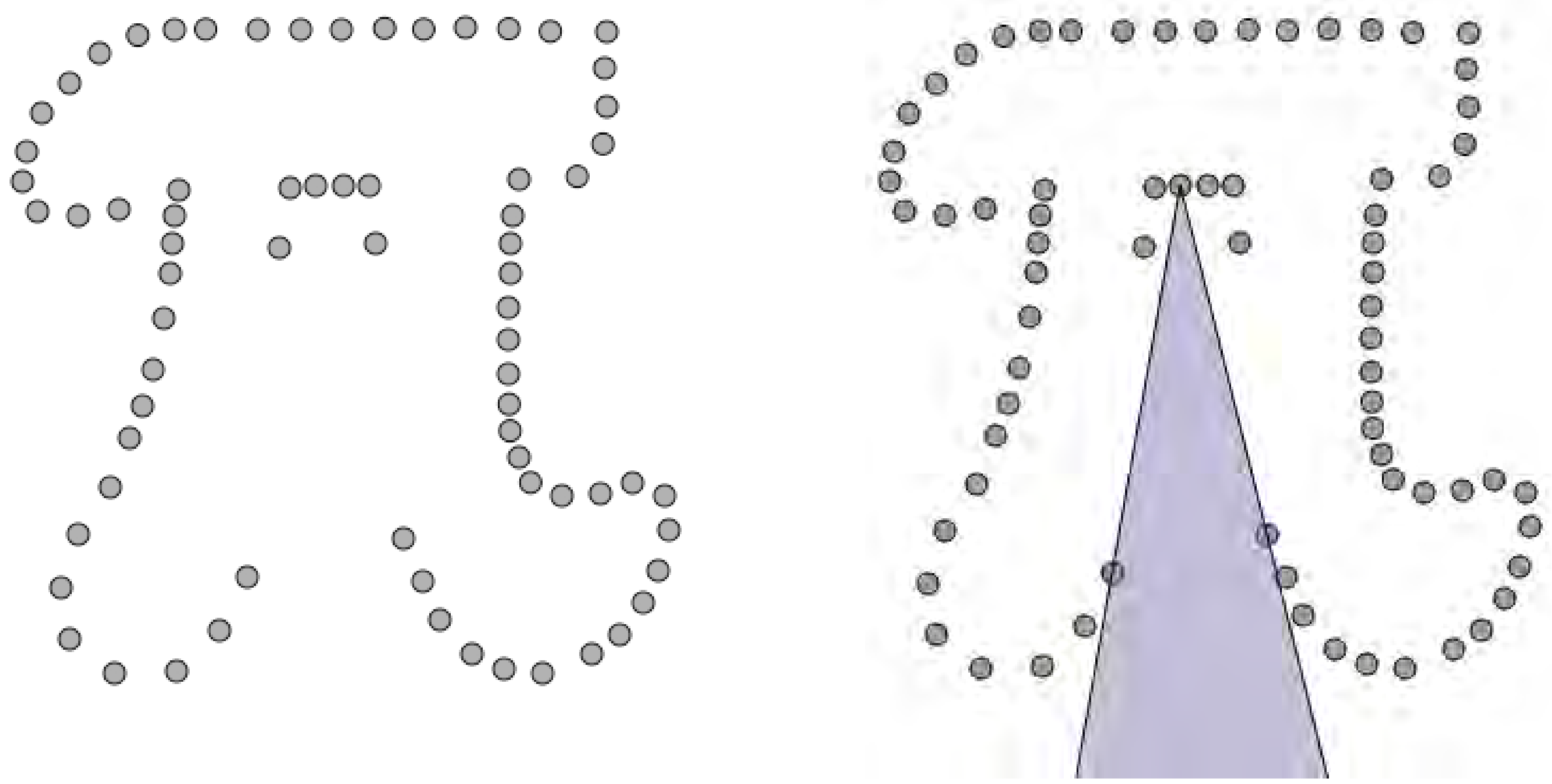
[Shalom et al. SIGA'10]



Cone Carving

- Use visibility cones centered at points on the surface

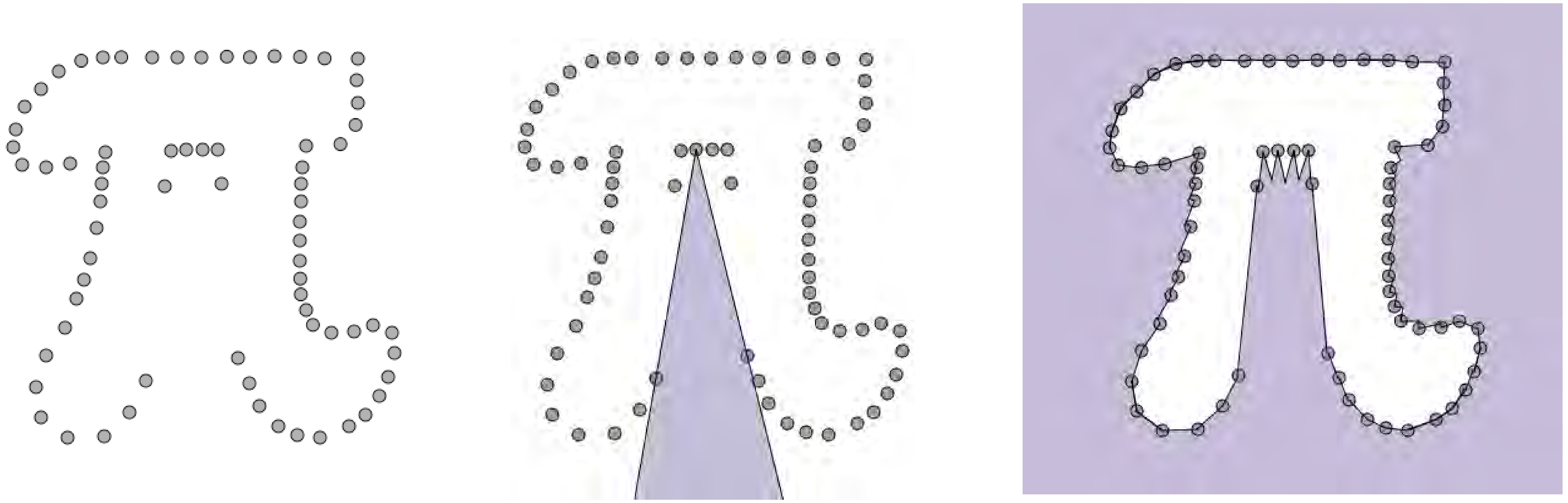
[Shalom et al. SIGA'10]



Cone Carving

- Use visibility cones centered at points on the surface

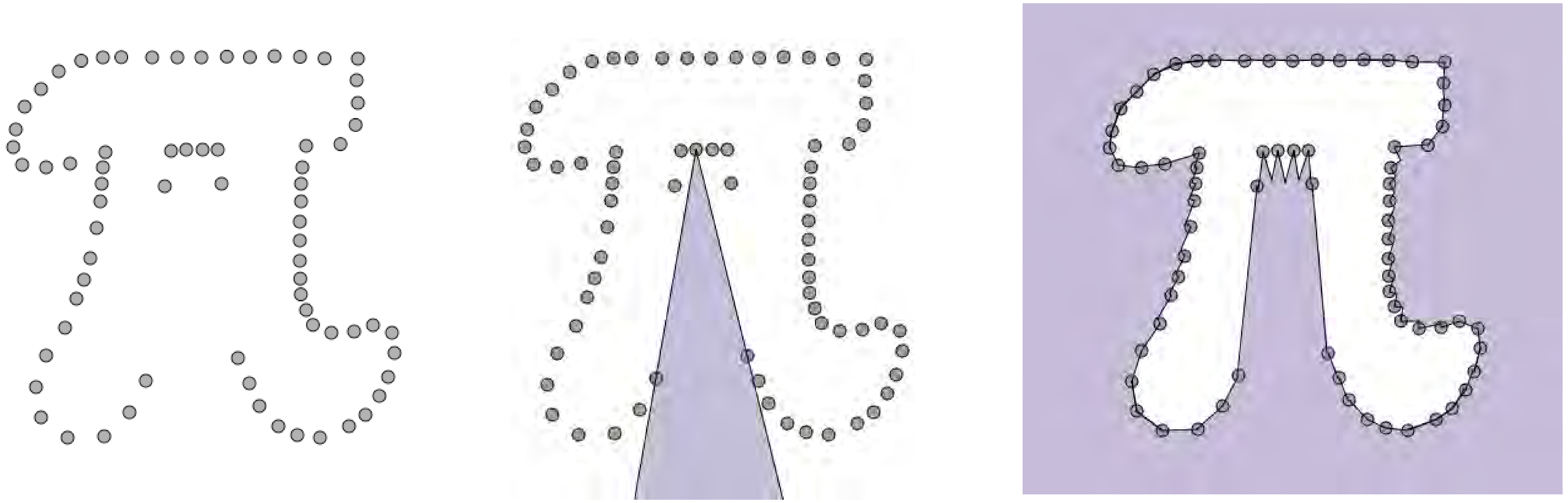
[Shalom et al. SIGA'10]



Cone Carving

- Use visibility cones centered at points on the surface

[Shalom et al. SIGA'10]

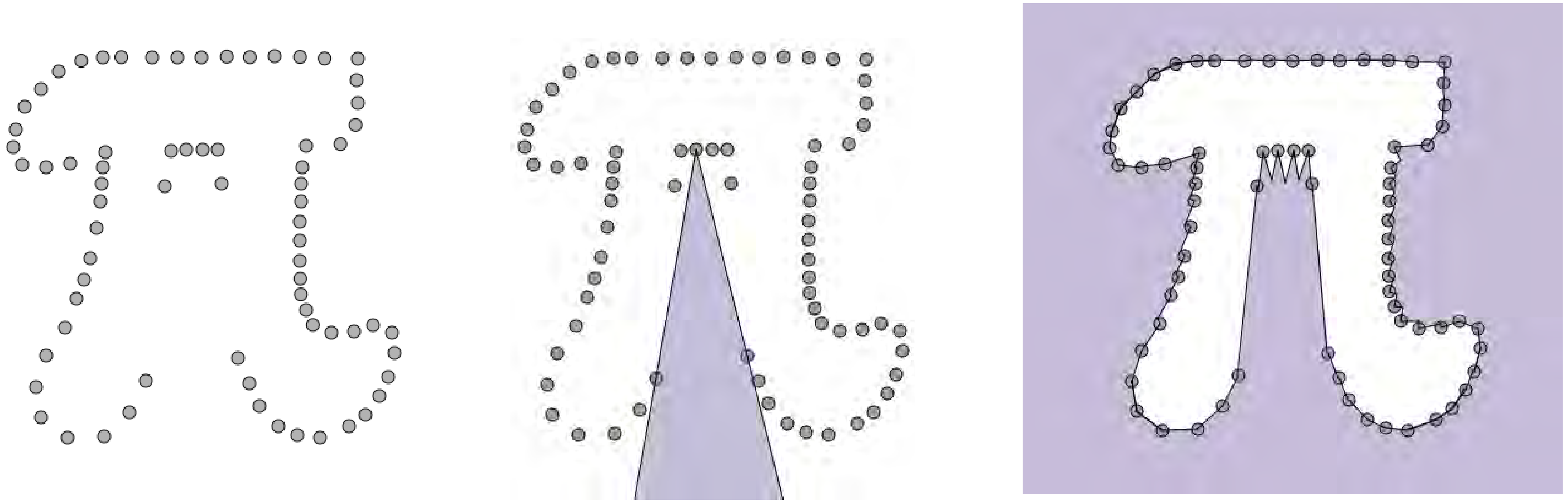


- Cones used to generate off-surface points for RBF reconstruction

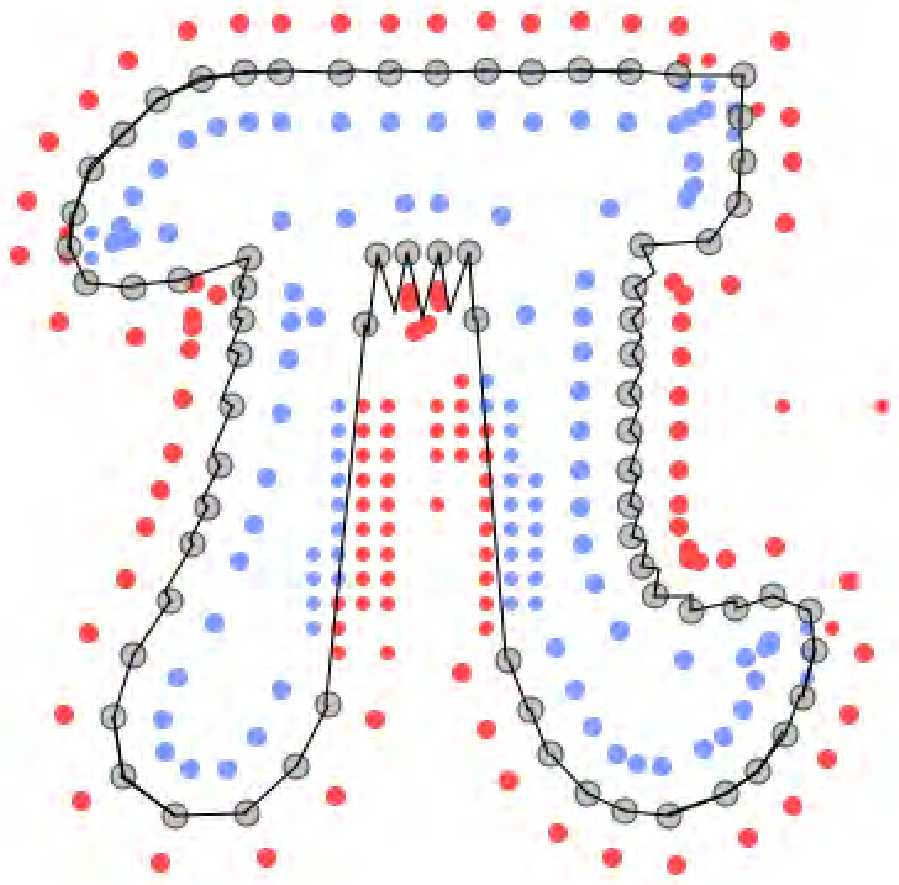
Cone Carving

- Use visibility cones centered at points on the surface

[Shalom et al. SIGA'10]



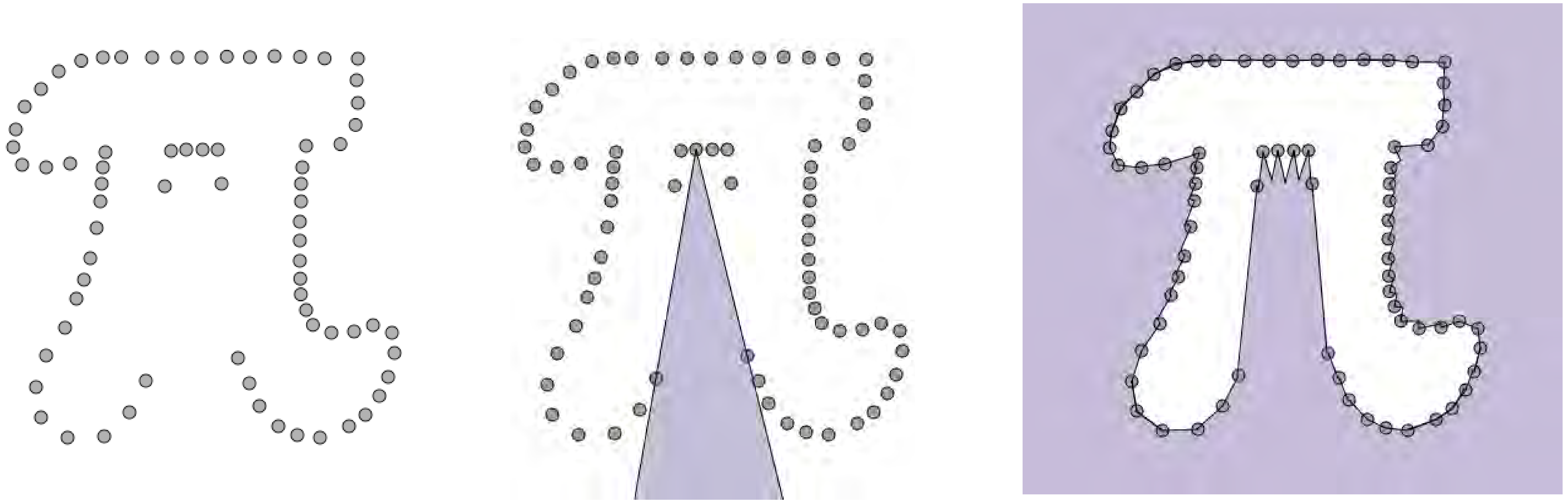
- Cones used to generate off-surface points for RBF reconstruction



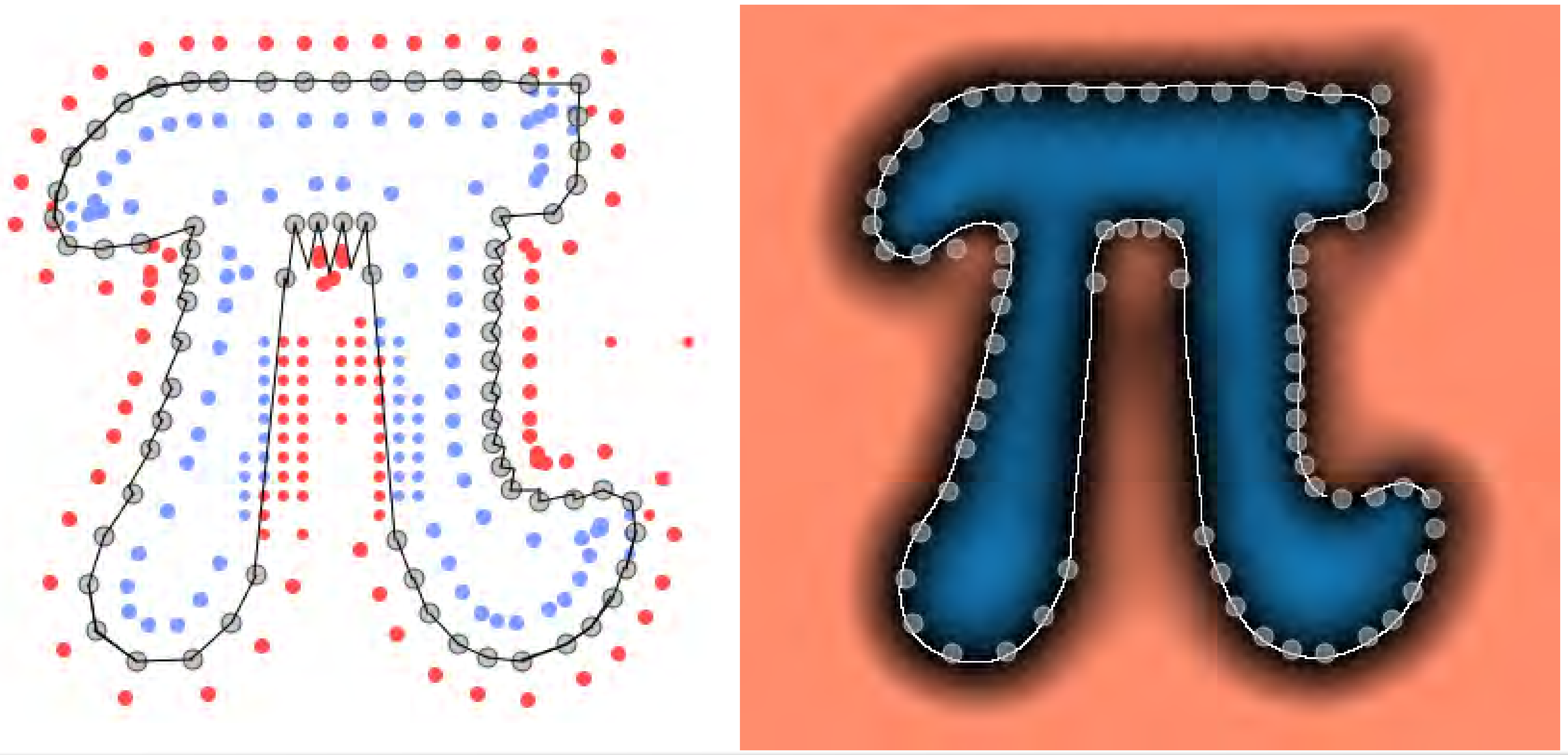
Cone Carving

- Use visibility cones centered at points on the surface

[Shalom et al. SIGA'10]



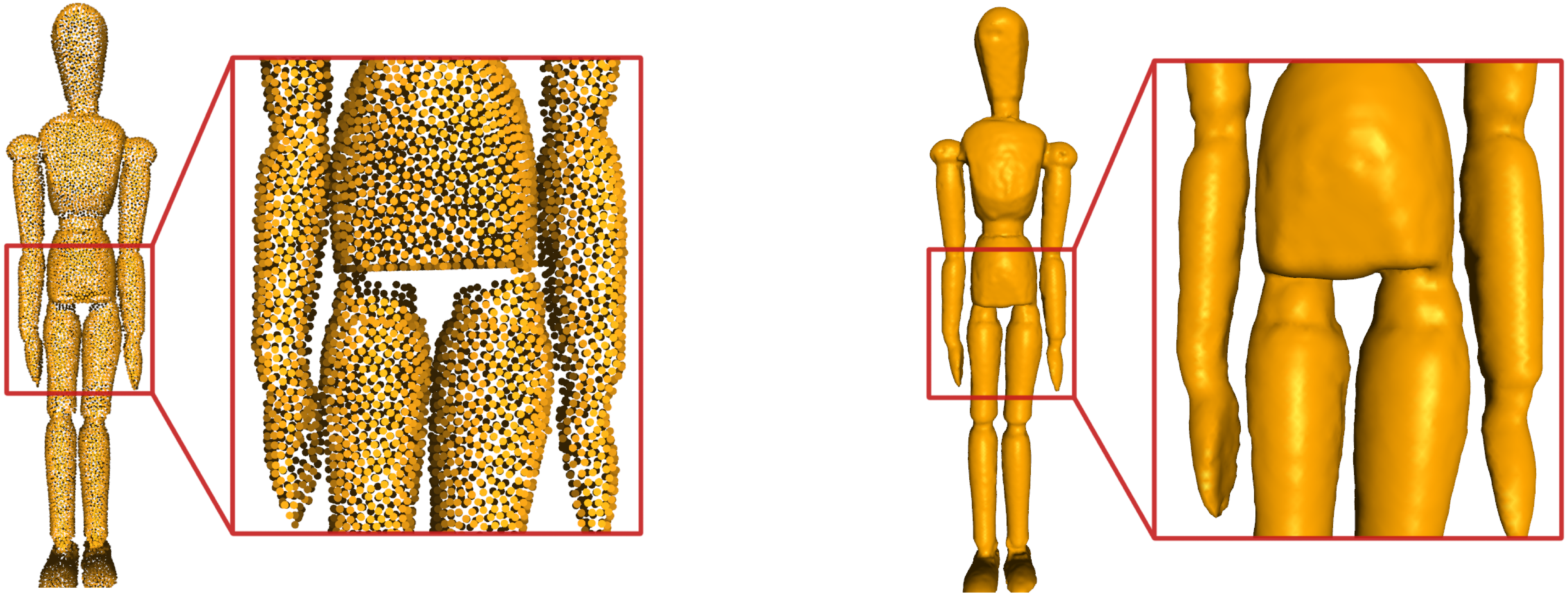
- Cones used to generate off-surface points for RBF reconstruction



Cone Carving

- Robust to missing data in challenging scenarios

[Shalom et al. SIGA'10]



Parity

- Use parity to assign confidence to exterior/interior labeling [Mullen et al. SGP'10]
- Robust unsigned distance function

Parity

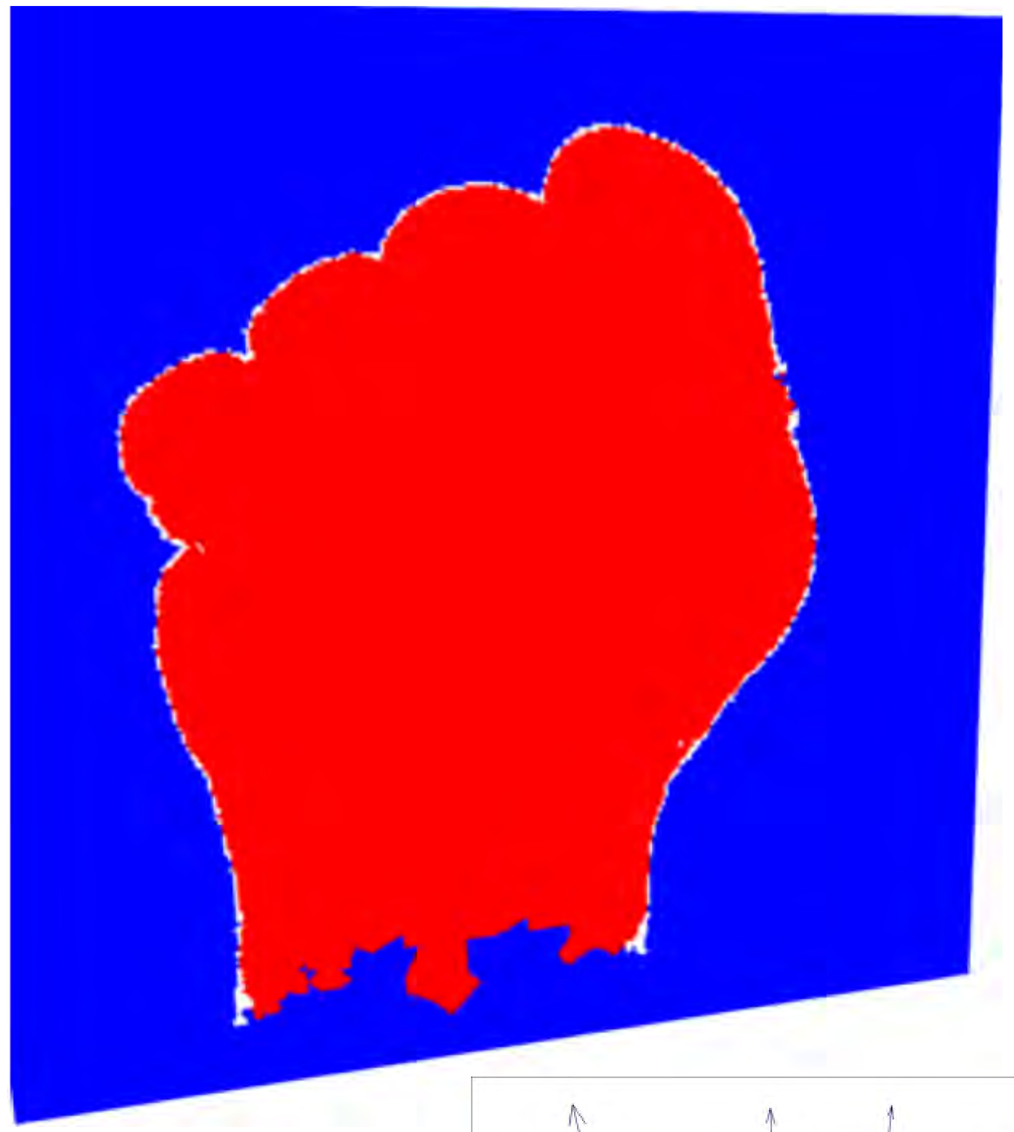
- Use parity to assign confidence to exterior/interior labeling [Mullen et al. SGP'10]
- Robust unsigned distance function

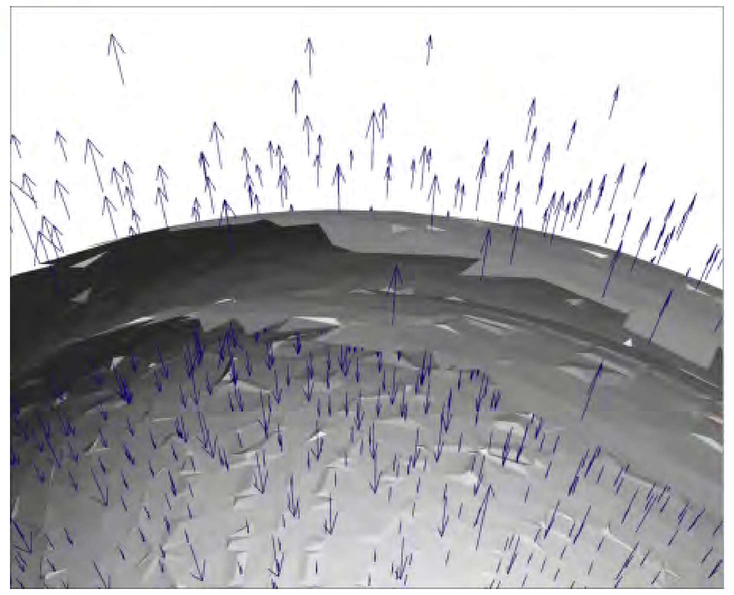
$$d_U(x) = \sqrt{\frac{1}{K} \sum_{p \in N_K(x)} ||x-p||^2}$$

Parity

- Use parity to assign confidence to exterior/interior labeling [Mullen et al. SGP'10]
- Robust unsigned distance function

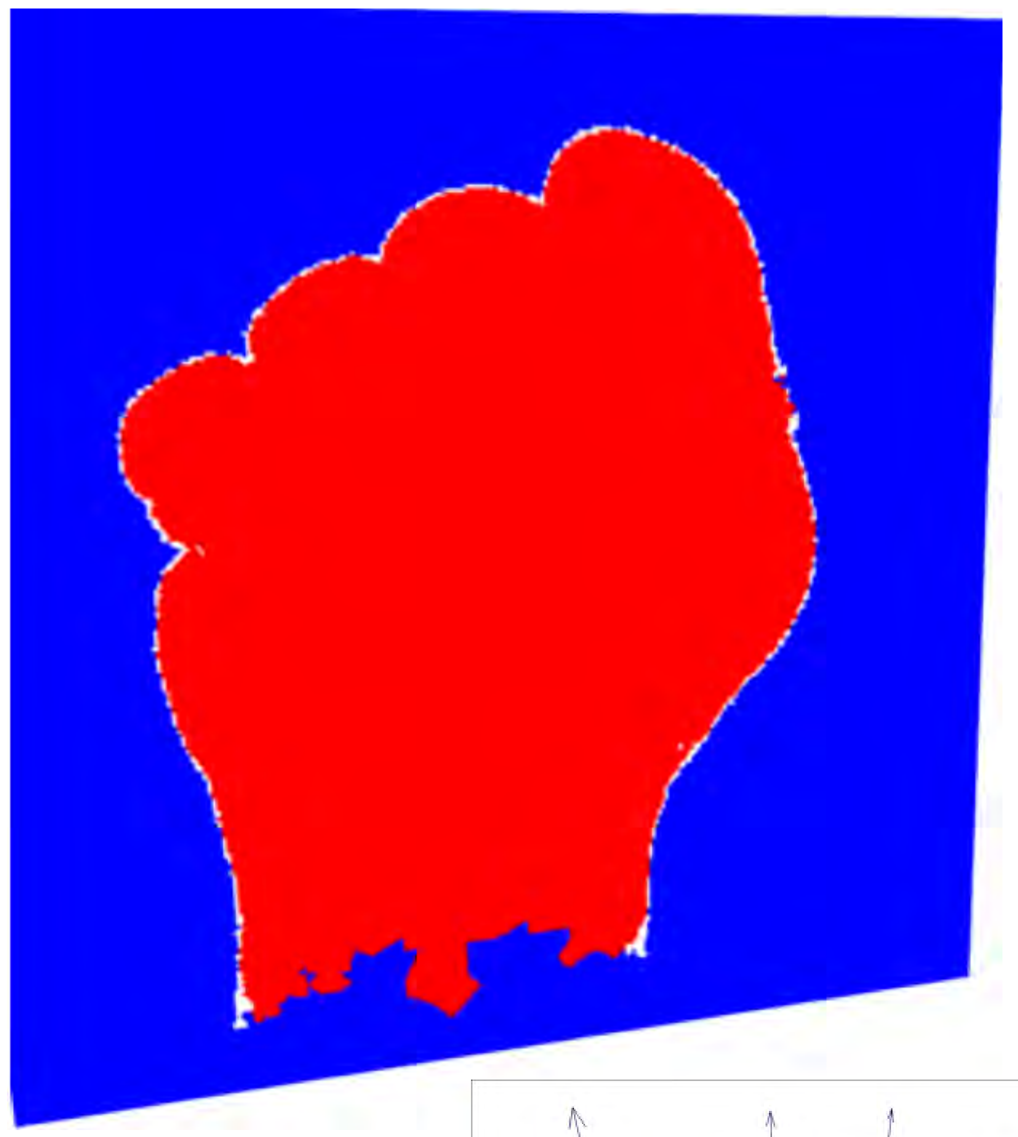
$$d_U(x) = \sqrt{\frac{1}{K}\sum_{p \in N_K(x)} ||x-p||^2}$$



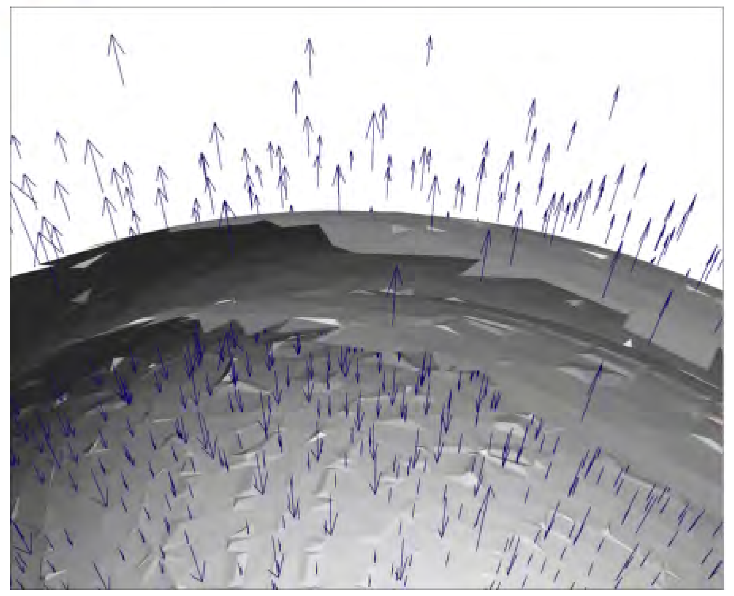


Parity

- Use parity to assign confidence to exterior/interior labeling [Mullen et al. SGP'10]
- Robust unsigned distance function



$$d_U(x) = \sqrt{\frac{1}{K} \sum_{p \in N_K(x)} ||x-p||^2}$$



$$E(f(x)) = \int_\Omega \left[\, |f(x)|_S^2 + W(c(x))(f(x) - \bar{\lambda}(x) d_U(x))^2 \, \right] dx$$

Volume Smoothness Prior

- Assume smoothness in the shape interior
- Output: skeleton curve, surface mesh, generally assumes watertightness
- Class shape: organic, man-made shapes
- Main approaches:
  - Skeletal curve prior
  - Smoothness on medial axis

ROSA

[Tagliasacchi et al. SIG'09]

- Produce skeletal curve from incomplete data
- Main assumption: local rotational symmetry

$$\mathbf{v}_i^{t+1} = \underset{\mathbf{v}\in\Re^3, \|\mathbf{v}\|=1}{argmin} \ \mathrm{var}\{\langle \mathbf{v}, \mathbf{n}(p_j)\rangle \ : \ p_j \in N_i^{(t)}\}, t \geq 0$$

ROSA

[Tagliasacchi et al. SIG'09]

- Produce skeletal curve from incomplete data
- Main assumption: local rotational symmetry

$$\mathbf{v}_i^{t+1} = \underset{\mathbf{v}\in\Re^3, \|\mathbf{v}\|=1}{argmin} \ \text{var}\{\langle \mathbf{v}, \mathbf{n}(p_j)\rangle : p_j \in N_i^{(t)}\}, t \geq 0$$

ensure plane representing rotational symmetry is orthogonal to point normals

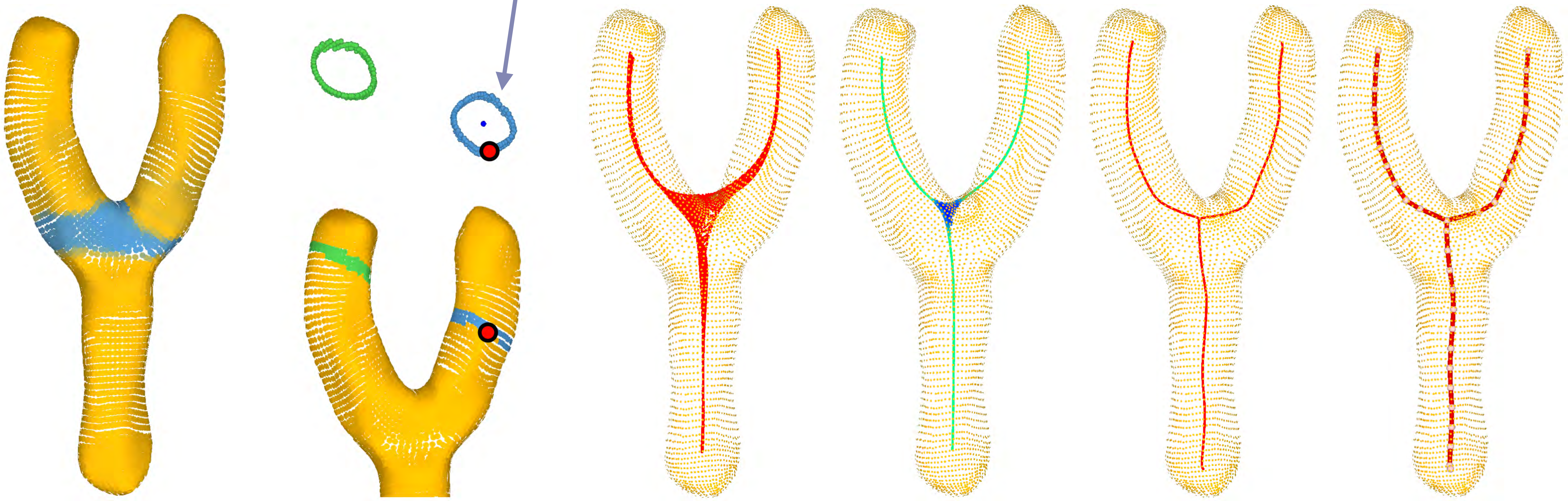
ROSA

[Tagliasacchi et al. SIG'09]

- Produce skeletal curve from incomplete data
- Main assumption: local rotational symmetry

$$\mathbf{v}_i^{t+1} = \underset{\mathbf{v}\in\Re^3, \|\mathbf{v}\|=1}{argmin} \quad \text{var}\left\{\langle \mathbf{v}, \mathbf{n}(p_j)\rangle \quad p_j \in N_i^{(t)}\right\}, t \geq 0$$

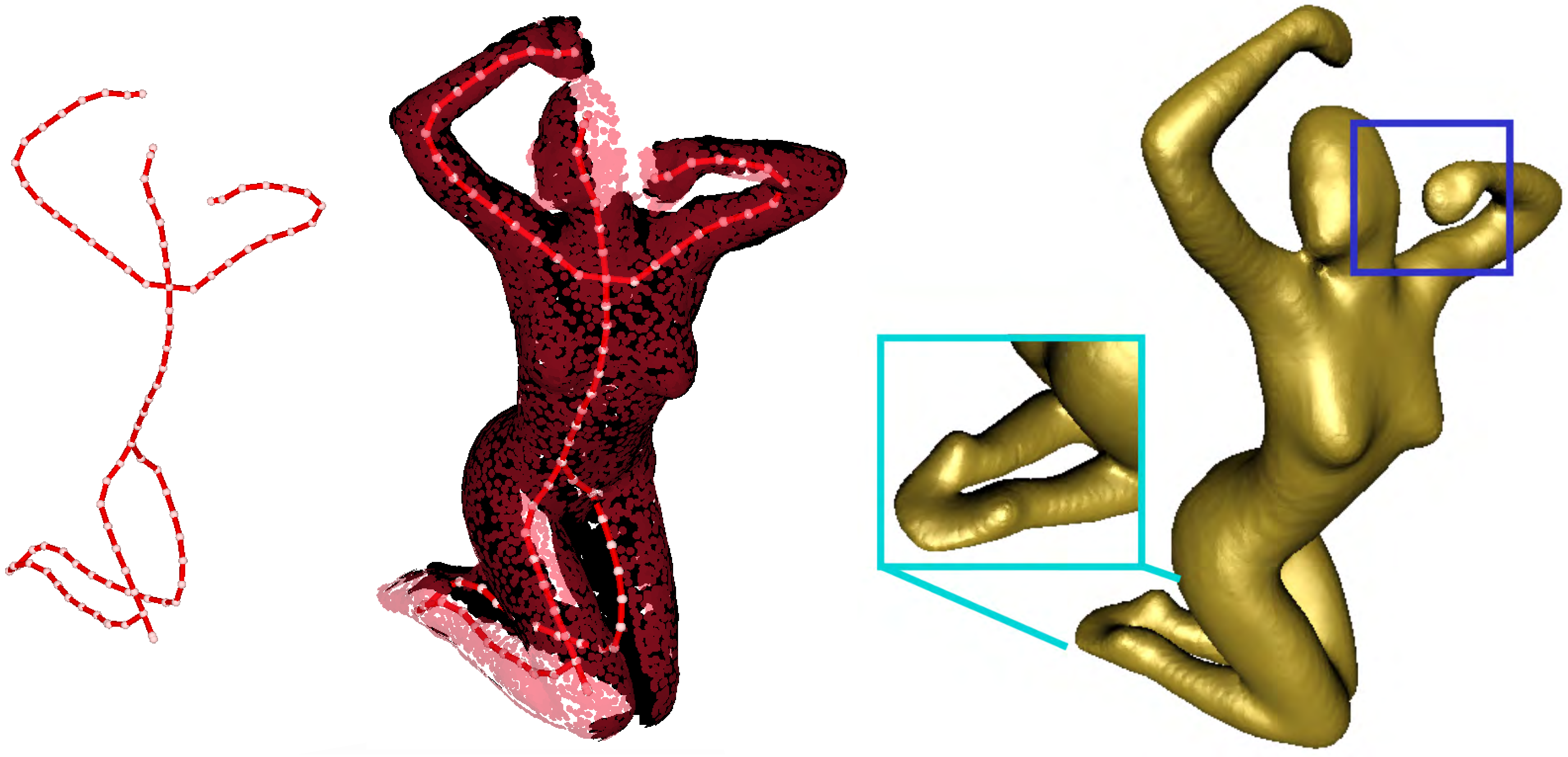
ensure plane representing rotational symmetry is orthogonal to point normals



ROSA

- Can also be used to resample surface for other reconstruction methods

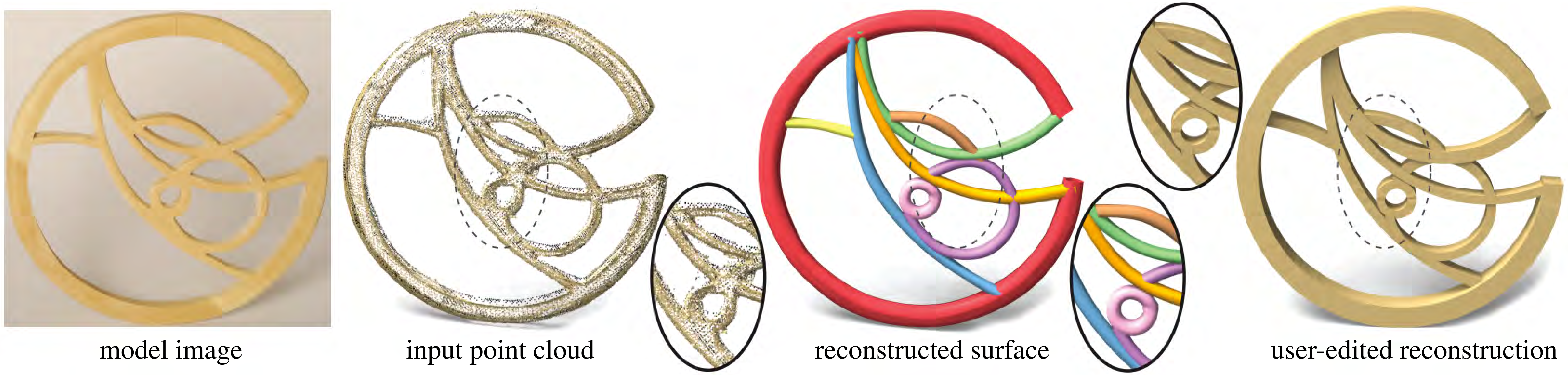
[Tagliasacchi et al. SIG'09]



Arterial Snakes

[Li et al. SIGA'10]

- Deformable model for extracting skeleton curve

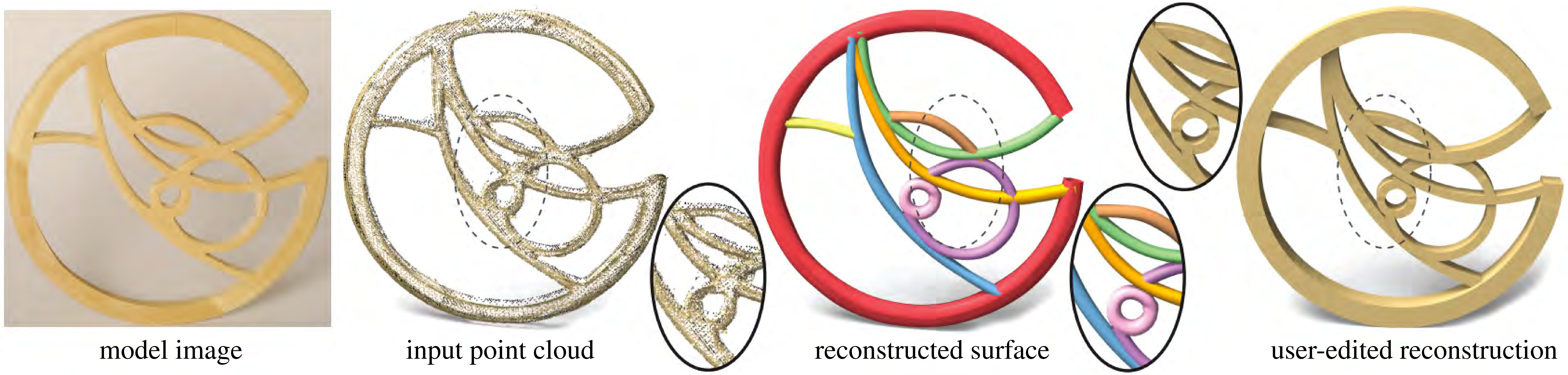


model image   input point cloud   reconstructed surface   user-edited reconstruction

Arterial Snakes

[Li et al. SIGA'10]

- Deformable model for extracting skeleton curve



model image   input point cloud   reconstructed surface   user-edited reconstruction

- Simultaneously grow, merge, and retract individual "snakelets"

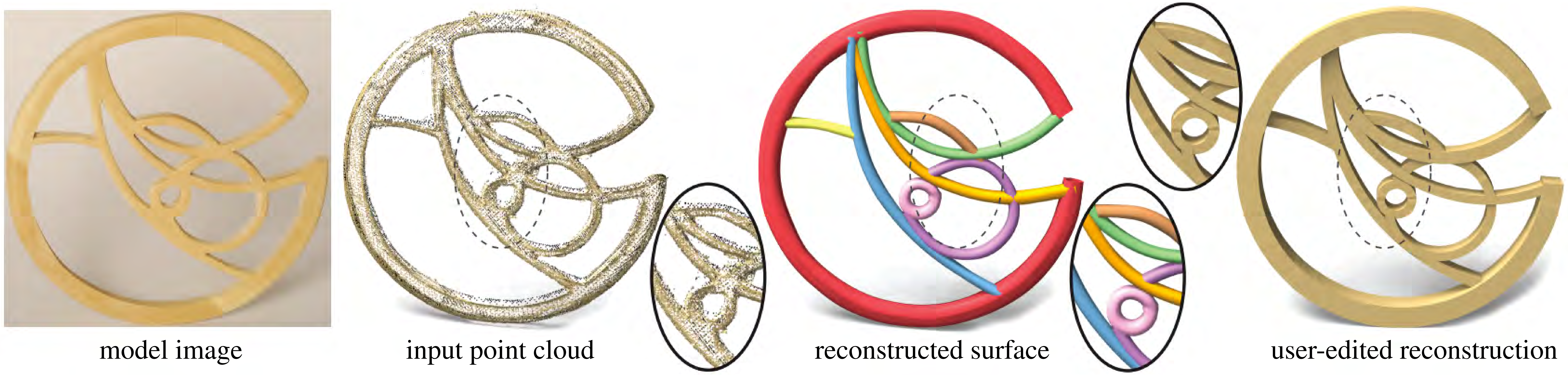
- Geometric optimization: satisfy data fitting, smoothness, fairness

$$E_d := \sum_{i,j}^{n,m} w_i^j \left\| \mathbf{A}_i^j G_i^j - R_i^j \right\|^2 \qquad E_s := \sum_{(i,j)} \|\mathbf{A}_i - \mathbf{A}_j\|_F^2,$$

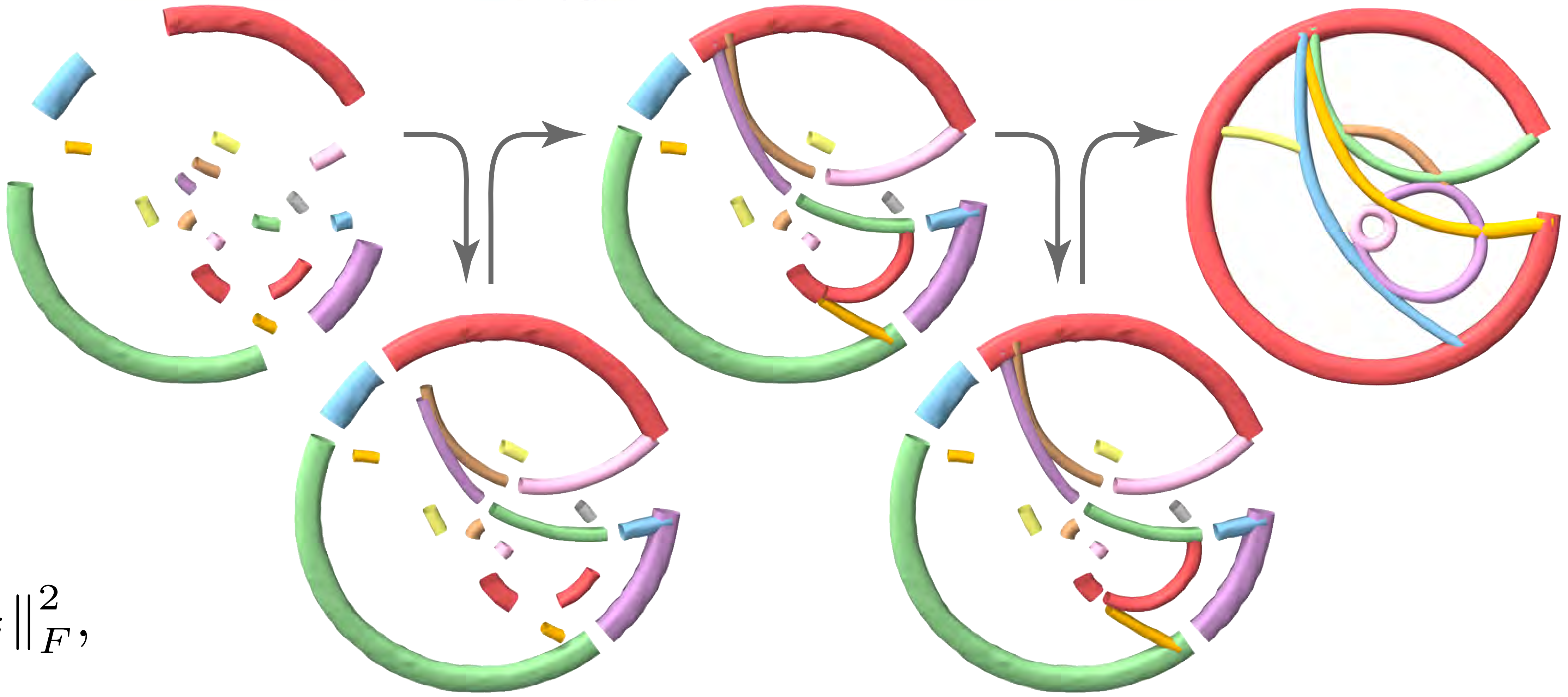
Arterial Snakes

[Li et al. SIGA'10]

- Deformable model for extracting skeleton curve



- Simultaneously grow, merge, and retract individual "snakelets"



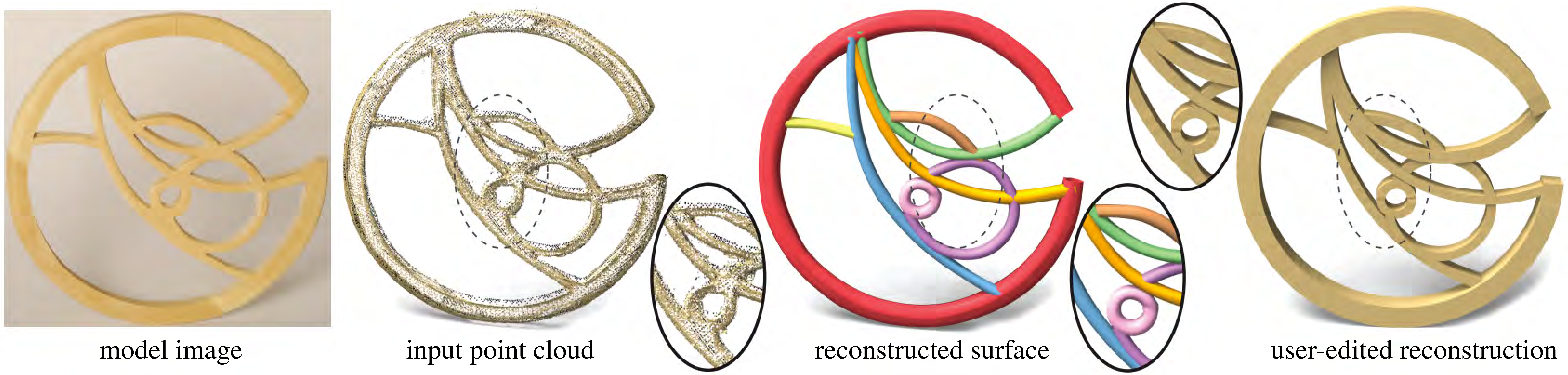
- Geometric optimization: satisfy data fitting, smoothness, fairness

$$E_d := \sum_{i,j}^{n,m} w_i^j \left\| \mathbf{A}_i^j G_i^j - R_i^j \right\|^2 \qquad E_s := \sum_{(i,j)} \|\mathbf{A}_i - \mathbf{A}_j\|_F^2,$$

Arterial Snakes

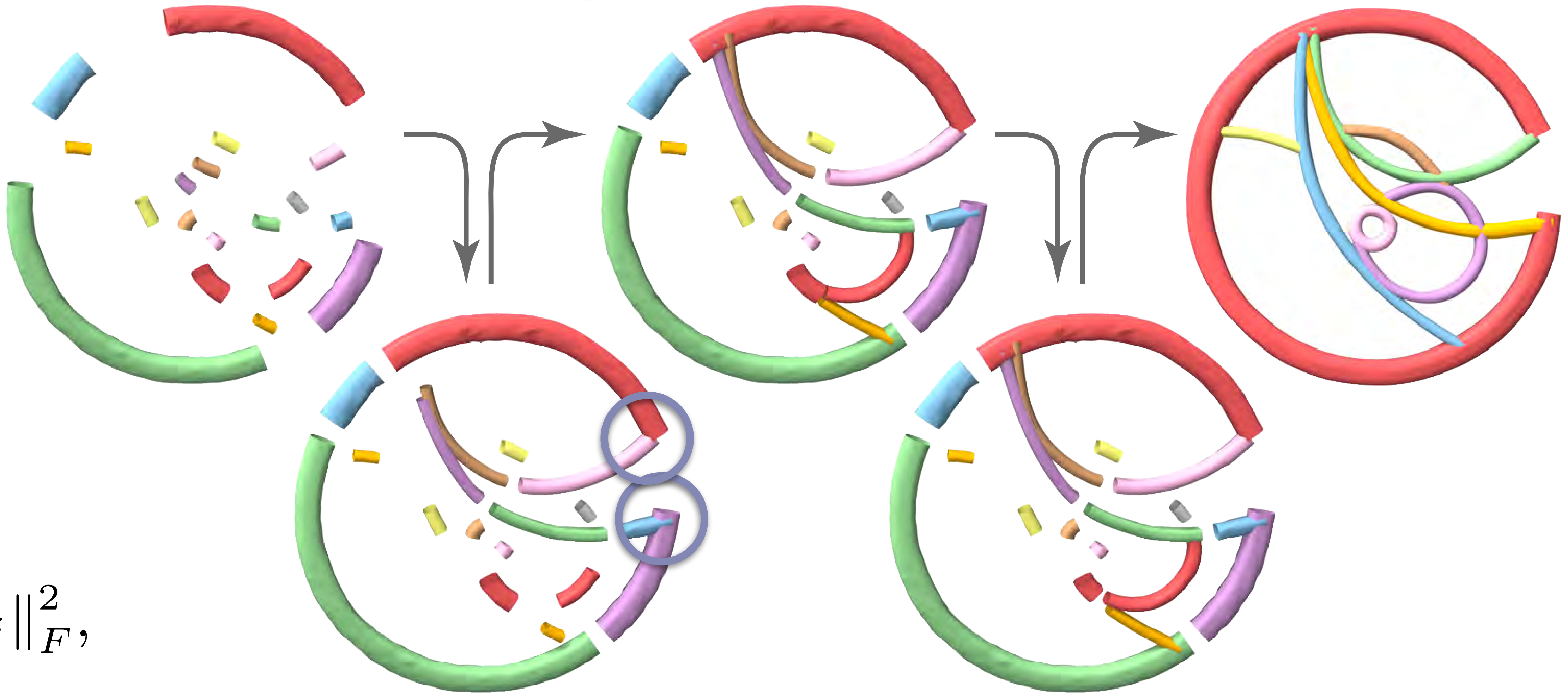
[Li et al. SIGA'10]

- Deformable model for extracting skeleton curve



model image   input point cloud   reconstructed surface   user-edited reconstruction

- Simultaneously grow, merge, and retract individual "snakelets"



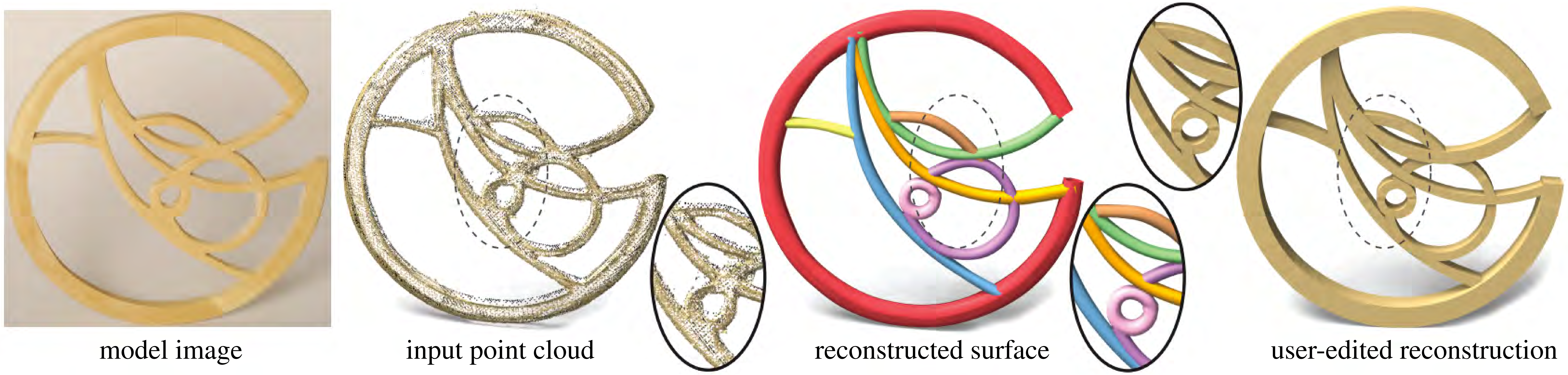
- Geometric optimization: satisfy data fitting, smoothness, fairness

$$E_d := \sum_{i,j}^{n,m} w_i^j \left\| \mathbf{A}_i^j G_i^j - R_i^j \right\|^2 \qquad E_s := \sum_{(i,j)} \|\mathbf{A}_i - \mathbf{A}_j\|_F^2,$$

Arterial Snakes

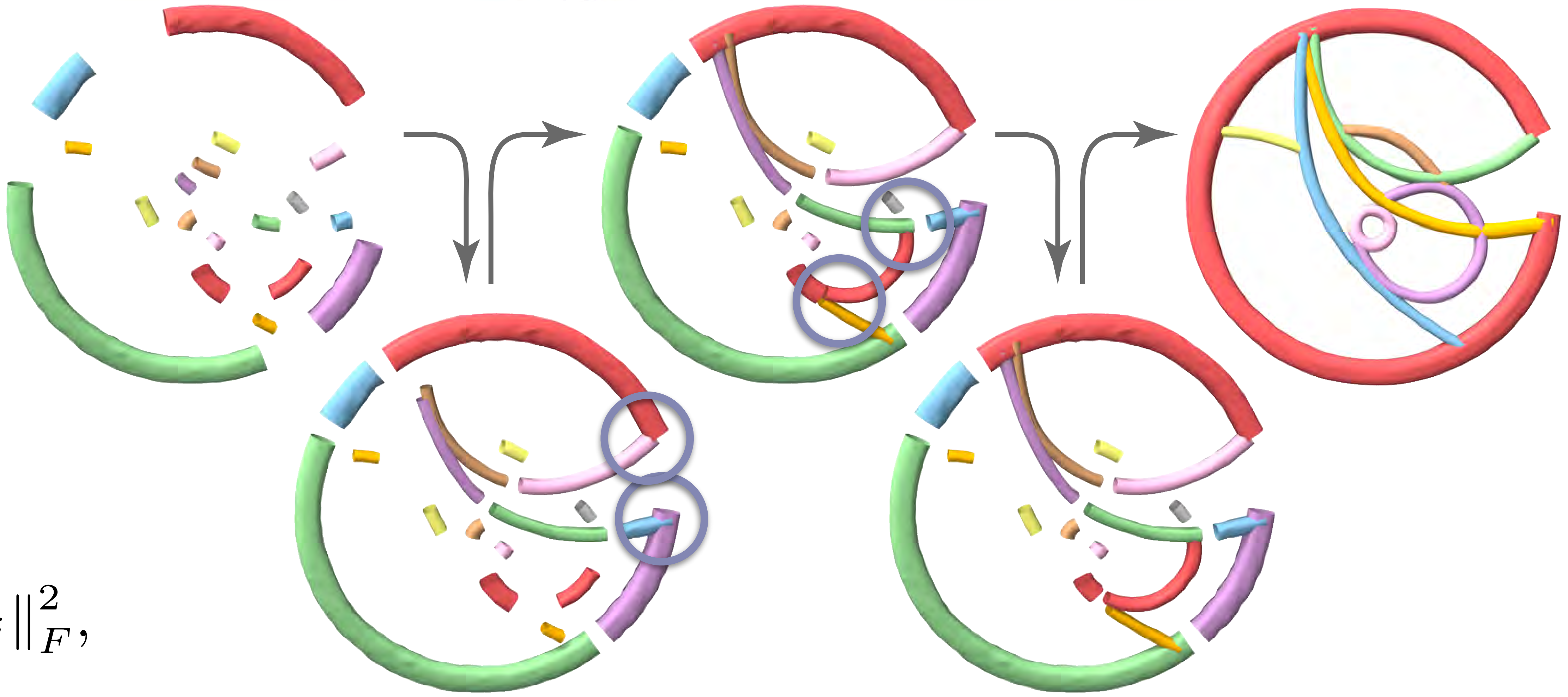
[Li et al. SIGA'10]

- Deformable model for extracting skeleton curve


model image   input point cloud   reconstructed surface   user-edited reconstruction

- Simultaneously grow, merge, and retract individual "snakelets"

- Geometric optimization: satisfy data fitting, smoothness, fairness

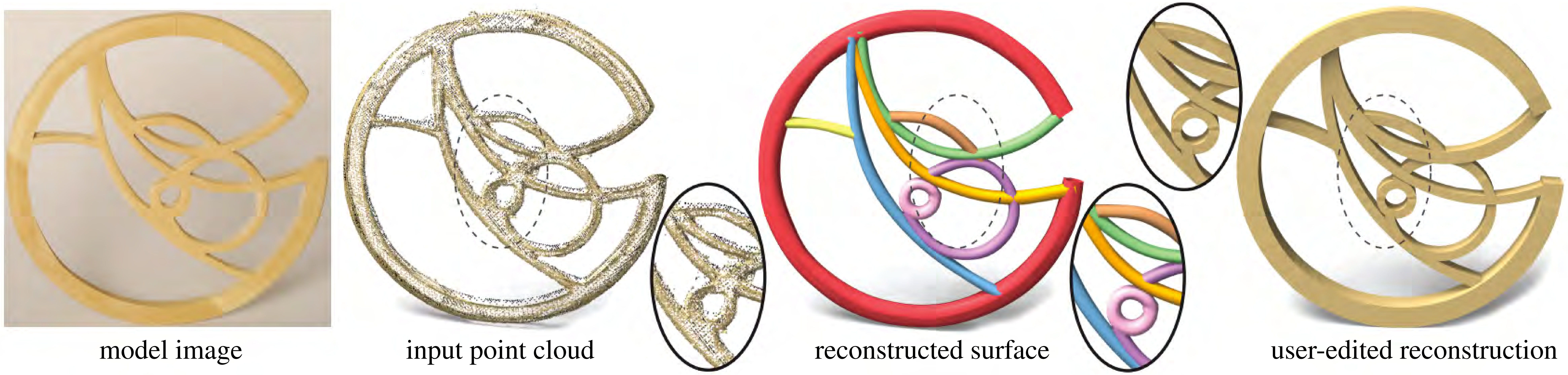


$$E_d := \sum_{i,j}^{n,m} w_i^j \left\| \mathbf{A}_i^j G_i^j - R_i^j \right\|^2 \qquad E_s := \sum_{(i,j)} \|\mathbf{A}_i - \mathbf{A}_j\|_F^2,$$

Arterial Snakes

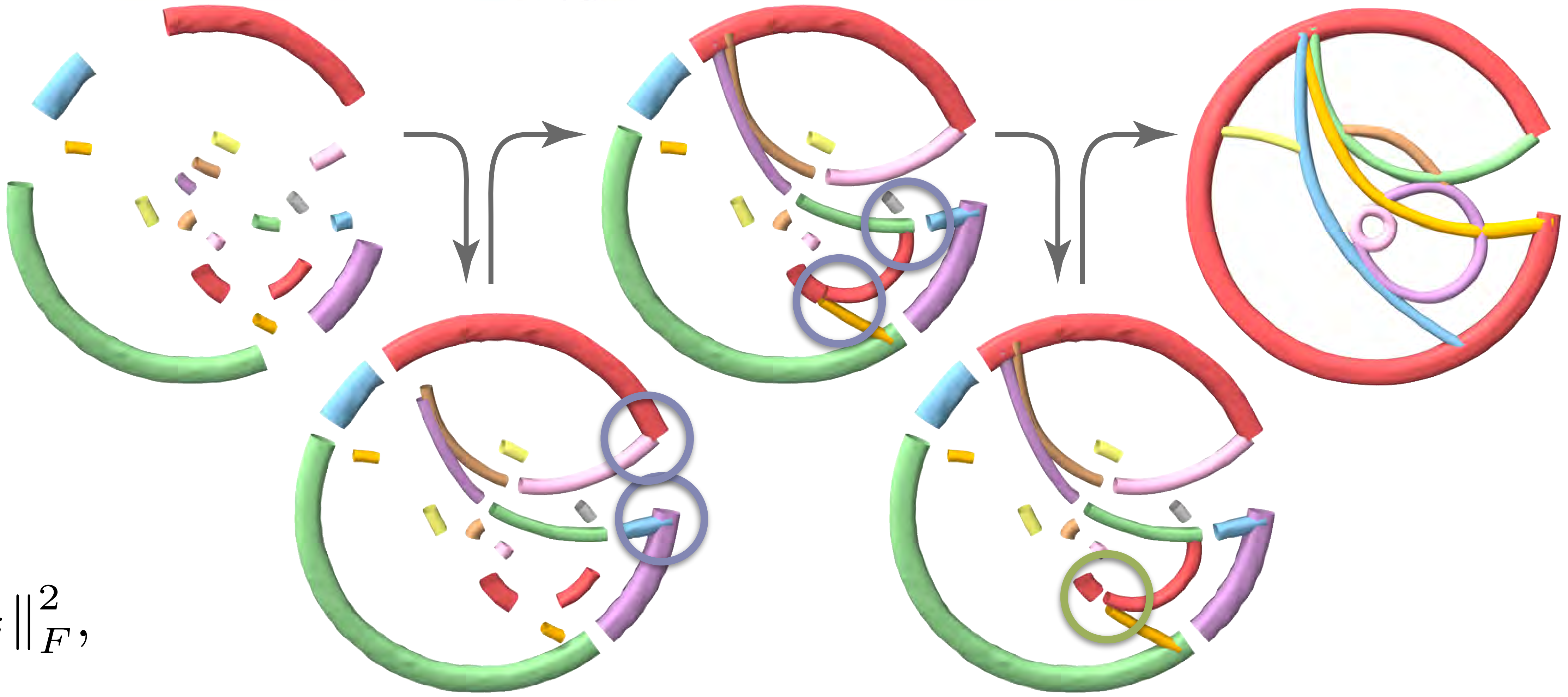
[Li et al. SIGA'10]

- Deformable model for extracting skeleton curve



- Simultaneously grow, merge, and retract individual "snakelets"

- Geometric optimization: satisfy data fitting, smoothness, fairness



$$E_d := \sum_{i,j}^{n,m} w_i^j \left\| \mathbf{A}_i^j G_i^j - R_i^j \right\|^2 \qquad E_s := \sum_{(i,j)} \|\mathbf{A}_i - \mathbf{A}_j\|_F^2,$$

VASE

[Tagliasacchi et al. SGP'11]

- Surface evolution method combining surface smoothness, visibility constraints, and medial axis smoothness

VASE

[Tagliasacchi et al. SGP'11]

- Surface evolution method combining surface smoothness, visibility constraints, and medial axis smoothness

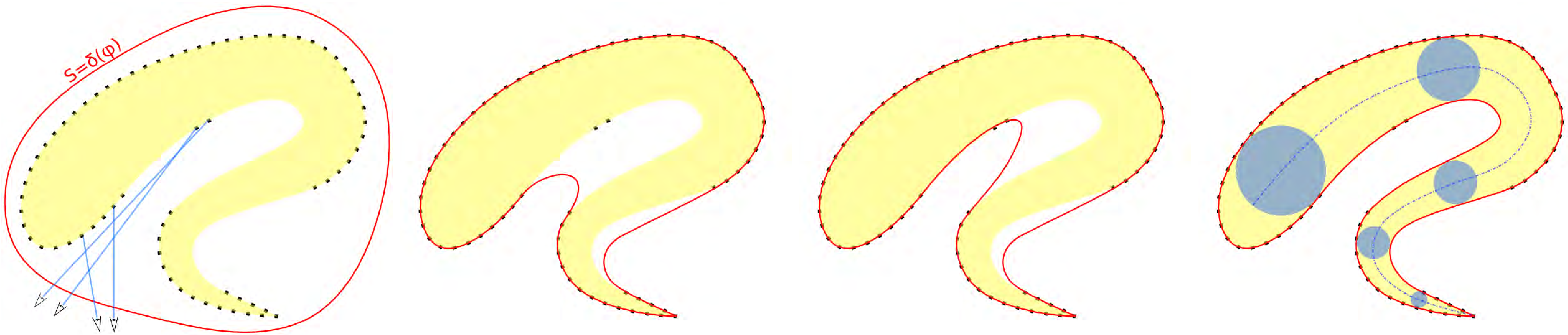
$$\dot{\phi} + \omega_1 F_{\text{fit}} + \omega_2 F_{\text{smooth}} + \omega_3 F_{\text{vol}} = 0$$

VASE

[Tagliasacchi et al. SGP'11]

- Surface evolution method combining surface smoothness, visibility constraints, and medial axis smoothness

$$\dot{\phi} + \omega_1 F_{\text{fit}} + \omega_2 F_{\text{smooth}} + \omega_3 F_{\text{vol}} = 0$$

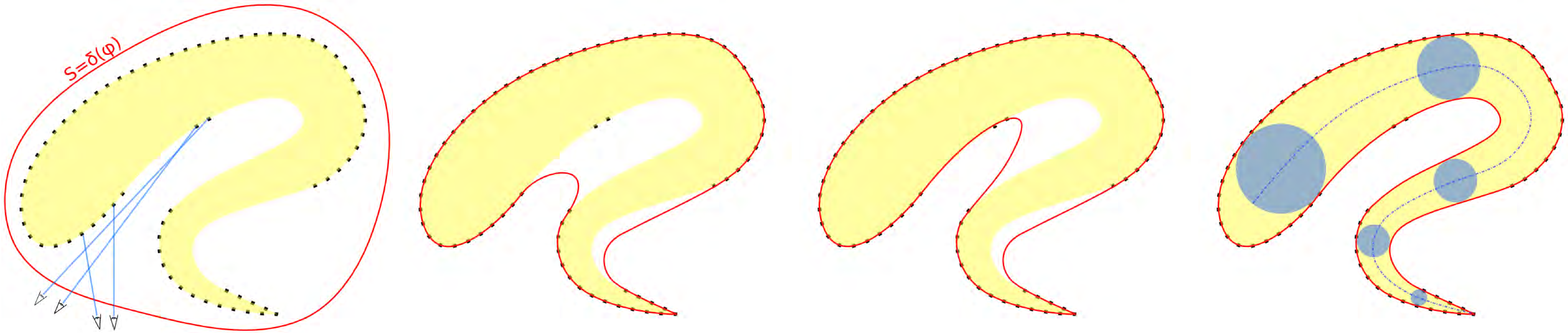


VASE

- Surface evolution method combining surface smoothness, visibility constraints, and medial axis smoothness

[Tagliasacchi et al. SGP'11]

$$\dot{\phi} + \omega_1 F_{\text{fit}} + \omega_2 F_{\text{smooth}} + \omega_3 F_{\text{vol}} = 0$$



Surface Smoothness

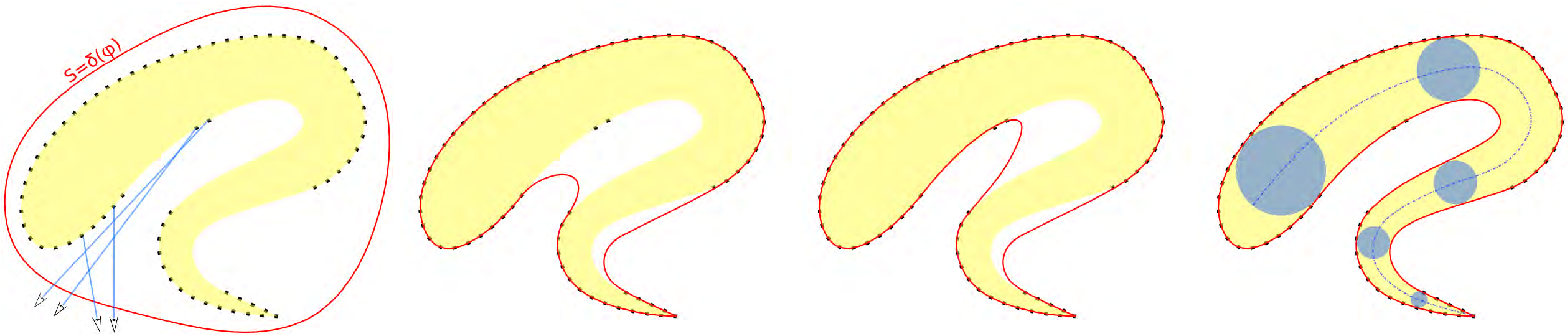
$$E_{\text{smooth}} = \int \|\Delta_{\mathcal{S}}^2 S\|^2$$

VASE

[Tagliasacchi et al. SGP'11]

- Surface evolution method combining surface smoothness, visibility constraints, and medial axis smoothness

$$\dot{\phi} + \omega_1 F_{\text{fit}} + \omega_2 F_{\text{smooth}} + \omega_3 F_{\text{vol}} = 0$$



Surface Smoothness

$$E_{\text{smooth}} = \int \|\Delta^2_{\mathcal{S}} S\|^2$$

Visibility

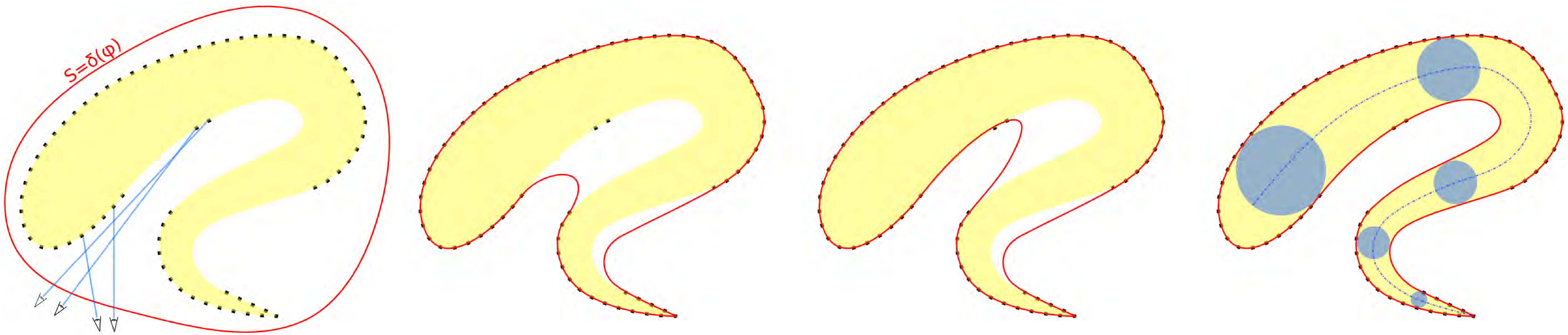
$$E_{\text{fit}} = \sum_{i=1}^{N} \|s_i^t - p_i\|^2$$

VASE

- Surface evolution method combining surface smoothness, visibility constraints, and medial axis smoothness

[Tagliasacchi et al. SGP'11]

$$\dot{\phi} + \omega_1 F_{\text{fit}} + \omega_2 F_{\text{smooth}} + \omega_3 F_{\text{vol}} = 0$$



Surface Smoothness

$$E_{\text{smooth}} = \int \|\Delta^2_{\mathcal{S}} S\|^2$$

Visibility

$$E_{\text{fit}} = \sum_{i=1}^{N} \|s_i^t - p_i\|^2$$

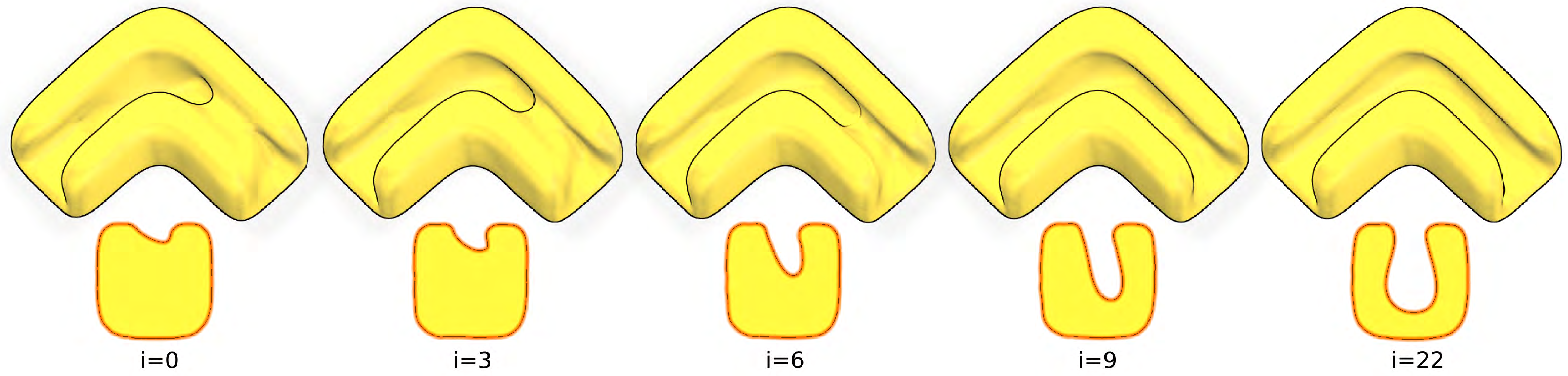
Medial Axis Smoothness

$$E_{\text{vol}} = \int \|\Delta_{\mathcal{M}} \mathcal{R}\|^2$$

VASE

- Surface evolution over time

[Tagliasacchi et al. SGP'11]



Primitive Prior

- Shape composed of simple geometric primitives
- Output: surface mesh, set of primitives
- Class shape: CAD models, architectural models
- Main approaches:
  - Surface primitives
  - Volume primitives

Efficient RANSAC

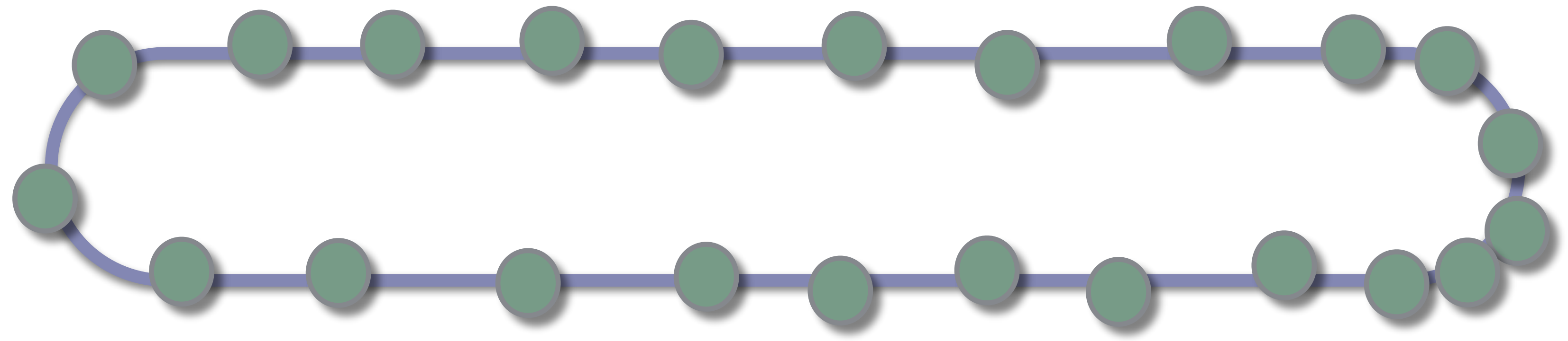
[Schnabel et al. CGF'07]

- Detection of simple surface primitives
  - Planes, spheres, cylinders, torii, etc..

Efficient RANSAC

[Schnabel et al. CGF'07]

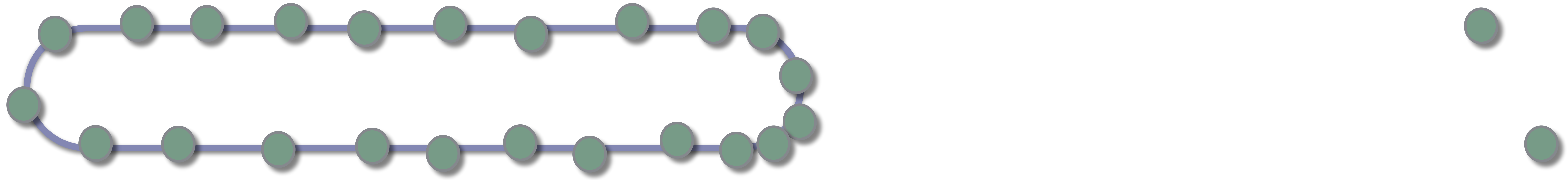
- Detection of simple surface primitives
  - Planes, spheres, cylinders, torii, etc..



Efficient RANSAC

[Schnabel et al. CGF’07]

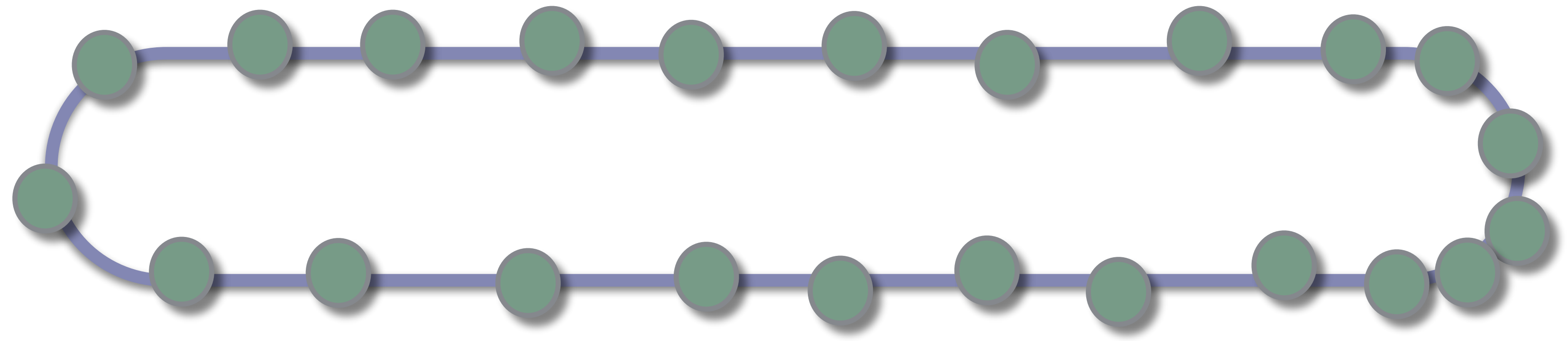
- Detection of simple surface primitives
  - Planes, spheres, cylinders, torii, etc..

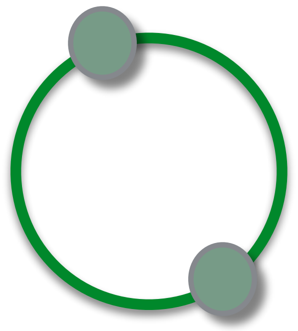


Efficient RANSAC

[Schnabel et al. CGF'07]

- Detection of simple surface primitives
  - Planes, spheres, cylinders, torii, etc..

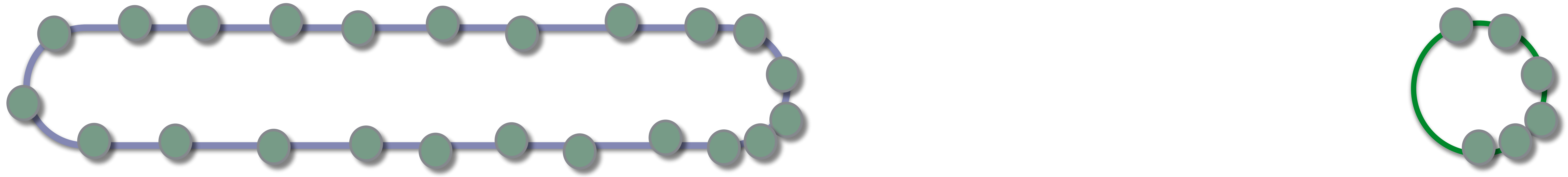




Efficient RANSAC

[Schnabel et al. CGF'07]

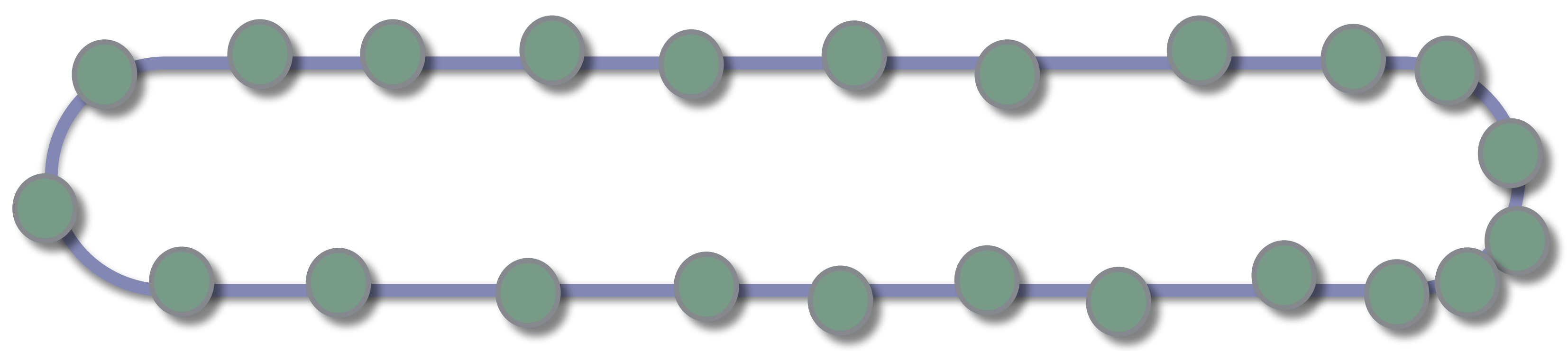
- Detection of simple surface primitives
  - Planes, spheres, cylinders, torii, etc..

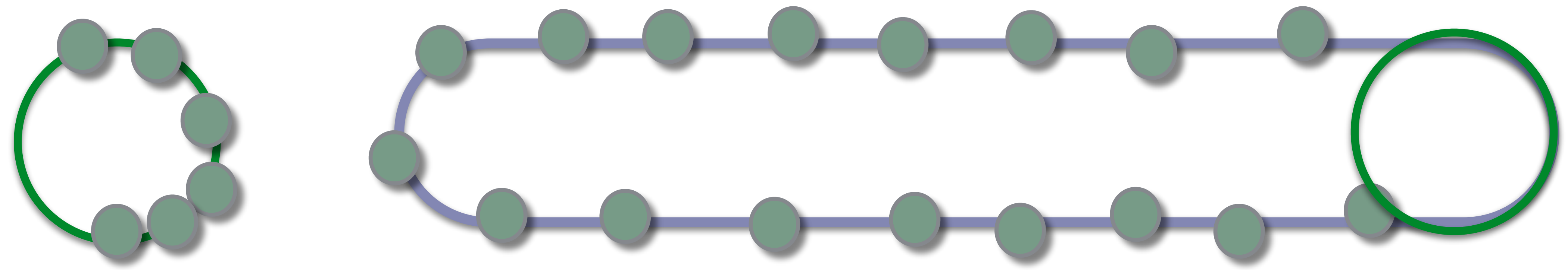


Efficient RANSAC

[Schnabel et al. CGF'07]

- Detection of simple surface primitives
    - Planes, spheres, cylinders, torii, etc..

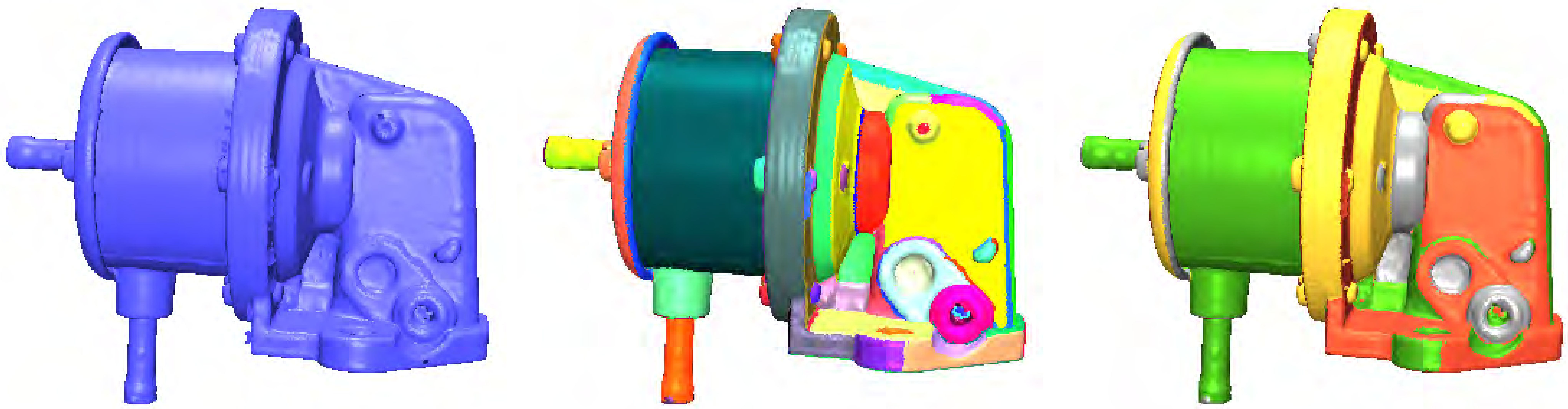




Efficient RANSAC

[Schnabel et al. CGF'07]

- Detection of simple surface primitives
    - Planes, spheres, cylinders, torii, etc..



Primitive Completion

[Schnabel et al. EG’09]

- Extrapolate primitives into the rest of the volume for surface completion

Primitive Completion

[Schnabel et al. EG'09]

- Extrapolate primitives into the rest of the volume for surface completion

$$E(\mathcal{S}) = E_a(\mathcal{S}) - E_p(\mathcal{S}) + E_c(\mathcal{S})$$
$$= \int_{\mathcal{S}} dA - \int_{\mathcal{S}} \mathbf{H}(\langle n|v\rangle) dA + E_c(\mathcal{S})$$

Primitive Completion

[Schnabel et al. EG'09]

- Extrapolate primitives into the rest of the volume for surface completion

$$E(\mathcal{S}) = E_a(\mathcal{S}) - E_p(\mathcal{S}) + E_c(\mathcal{S})$$
$$= \int_{\mathcal{S}} dA - \int_{\mathcal{S}} \mathbf{H}(\langle n|v\rangle) dA + E_c(\mathcal{S})$$

minimal surface area

Primitive Completion

[Schnabel et al. EG'09]

- Extrapolate primitives into the rest of the volume for surface completion

$$E(\mathcal{S}) = E_a(\mathcal{S}) - E_p(\mathcal{S}) + E_c(\mathcal{S})$$

$$= \int_{\mathcal{S}} dA - \int_{\mathcal{S}} \mathbf{H}(\langle n|v\rangle)dA + E_c(\mathcal{S})$$

minimal surface area

input and primitive normals agree

Primitive Completion

[Schnabel et al. EG'09]

- Extrapolate primitives into the rest of the volume for surface completion

$$E(\mathcal{S}) = E_a(\mathcal{S}) - E_p(\mathcal{S}) + E_c(\mathcal{S})$$
$$= \int_{\mathcal{S}} dA - \int_{\mathcal{S}} \mathbf{H}(\langle n|v \rangle) dA + E_c(\mathcal{S})$$

minimal surface area

input and primitive normals agree

interior/exterior constraints

Primitive Completion

[Schnabel et al. EG'09]

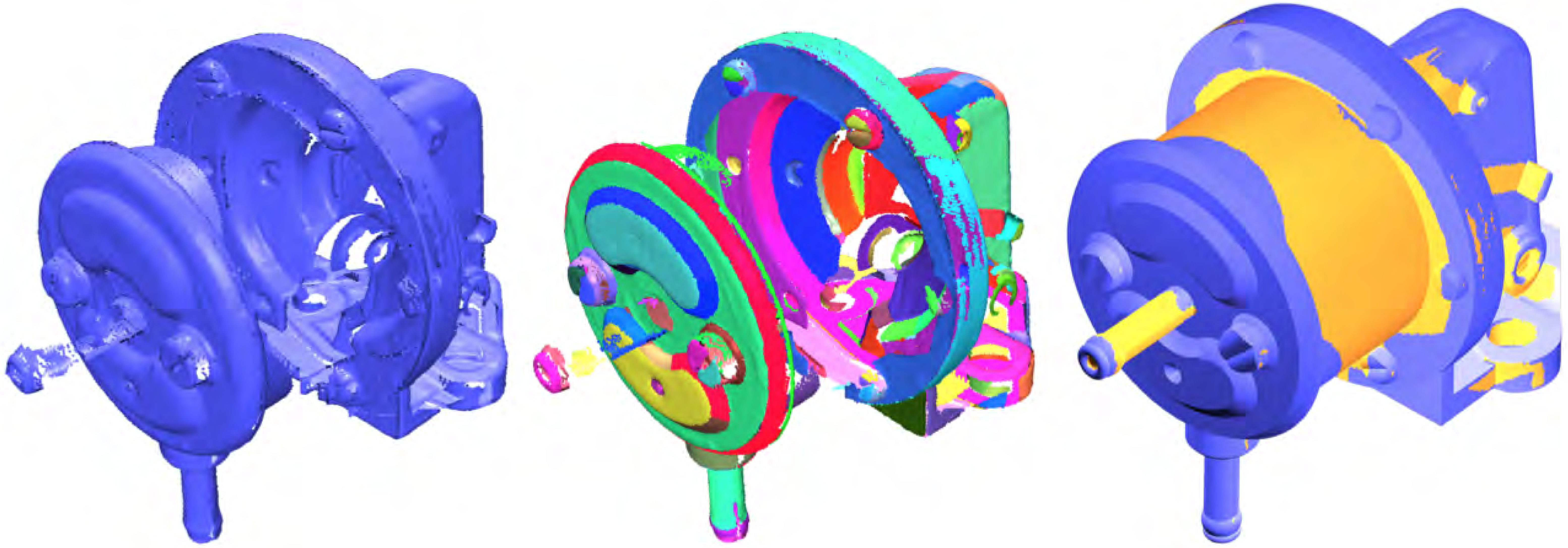
- Extrapolate primitives into the rest of the volume for surface completion

$$E(\mathcal{S}) = E_a(\mathcal{S}) - E_p(\mathcal{S}) + E_c(\mathcal{S})$$
$$= \int_{\mathcal{S}} dA - \int_{\mathcal{S}} \mathbf{H}(\langle n|v\rangle)dA + E_c(\mathcal{S})$$

minimal surface area

input and primitive normals agree

interior/exterior constraints



Hybrid Reconstruction

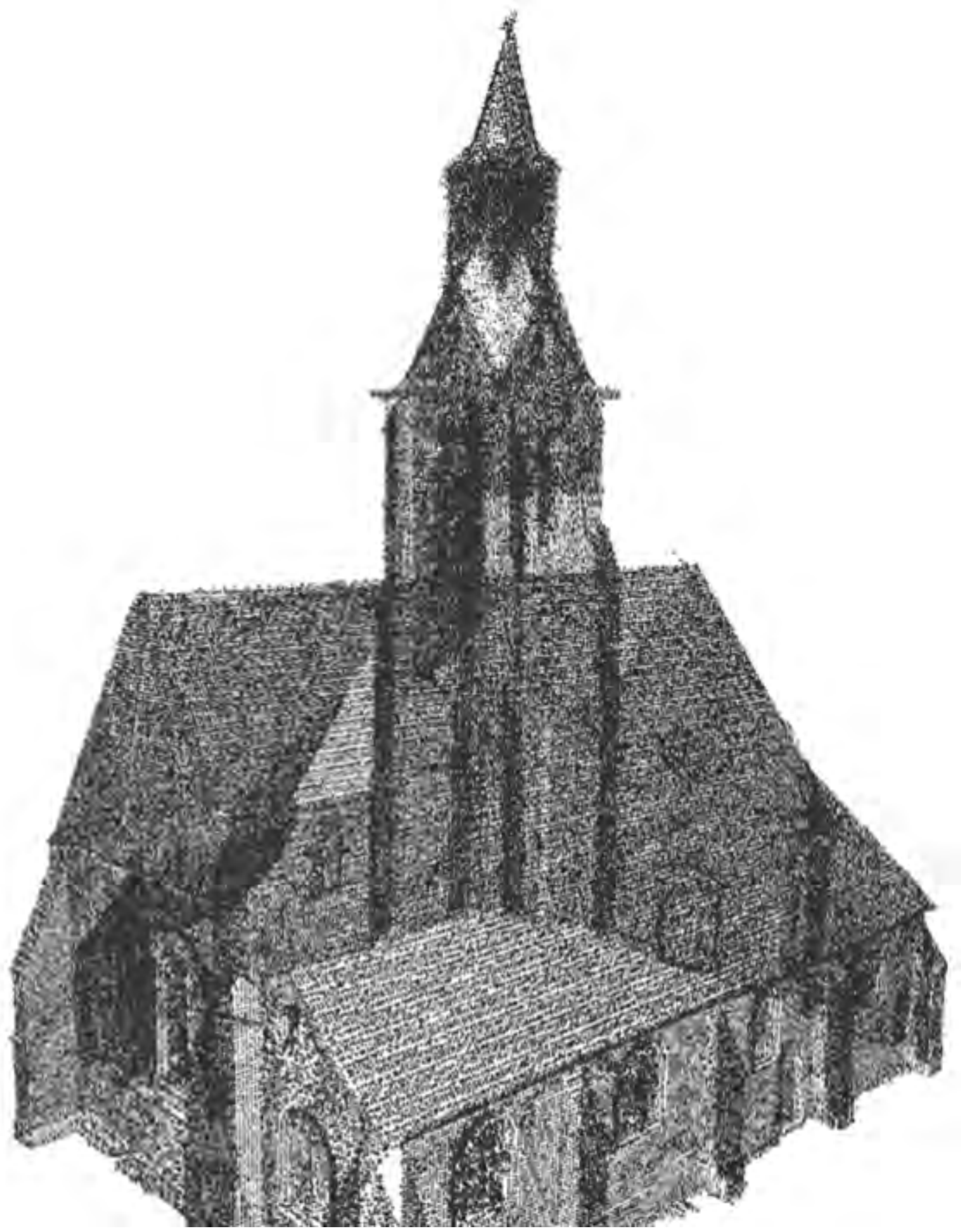
[Lafarge & Alliez EG'13]

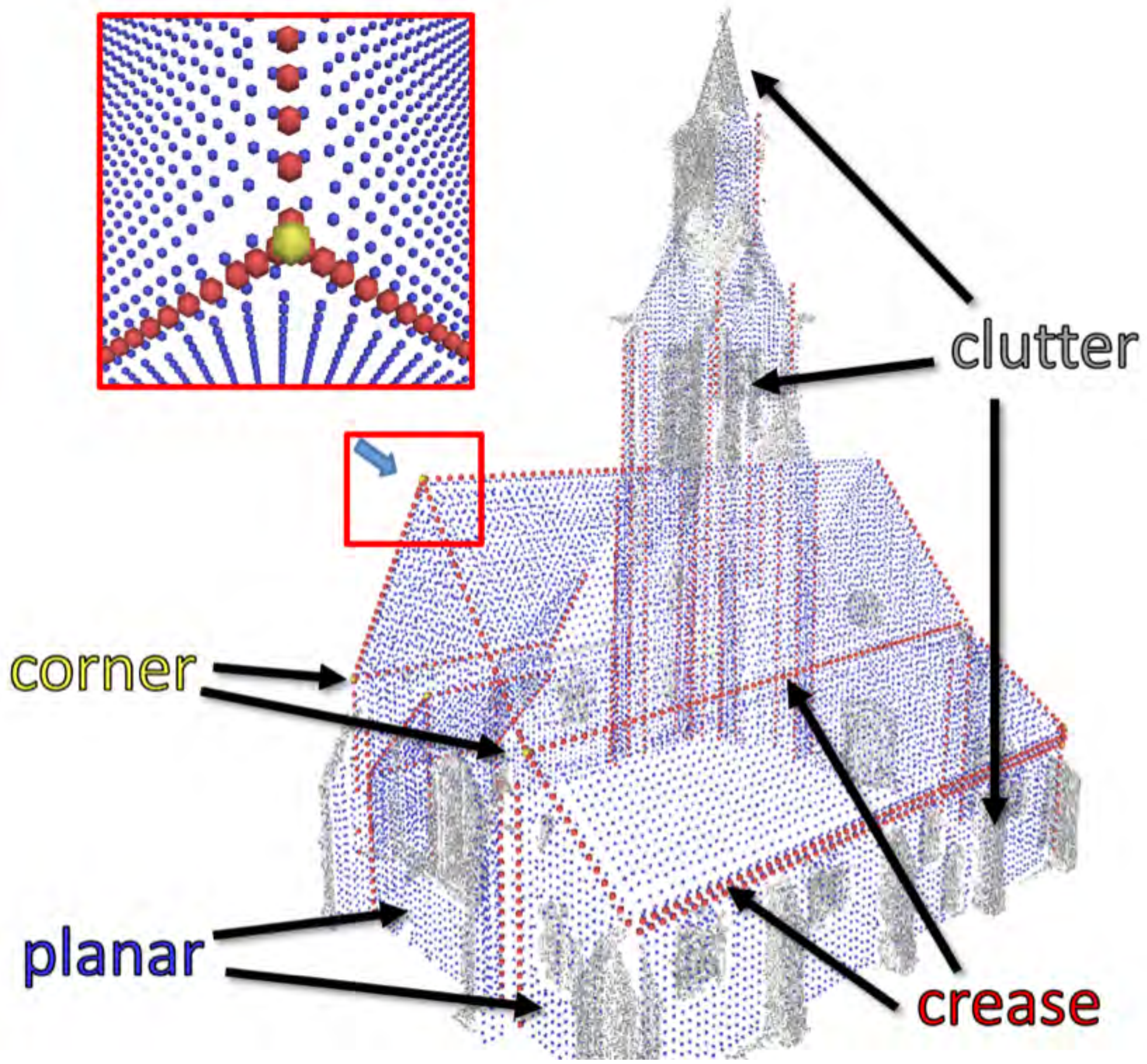
- Use primitives where there exists a good fit, otherwise use surface smoothness, by *structuring* the point set

Hybrid Reconstruction

- Use primitives where there exists a good fit, otherwise use surface smoothness, by *structuring* the point set

[Lafarge & Alliez EG'13]

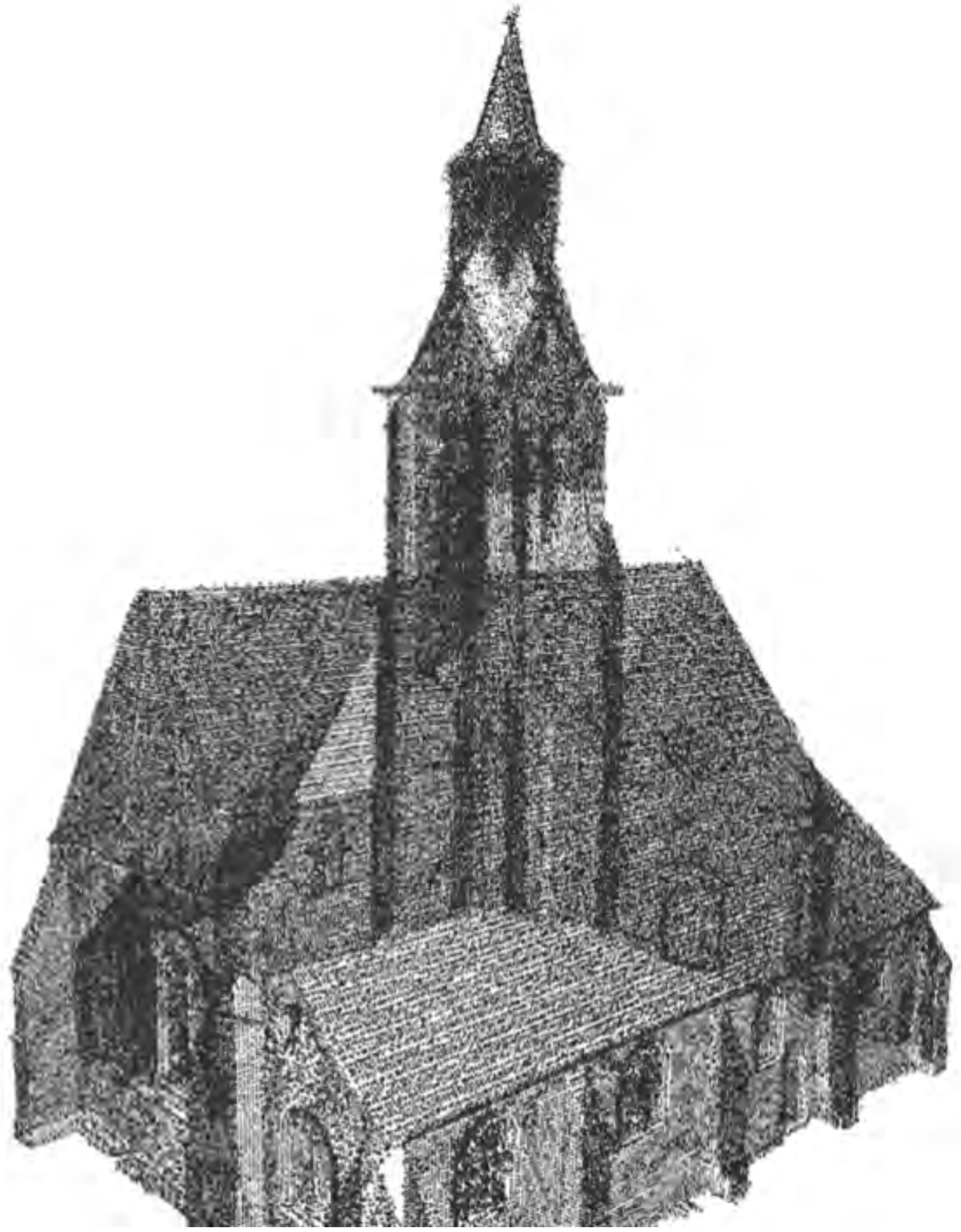


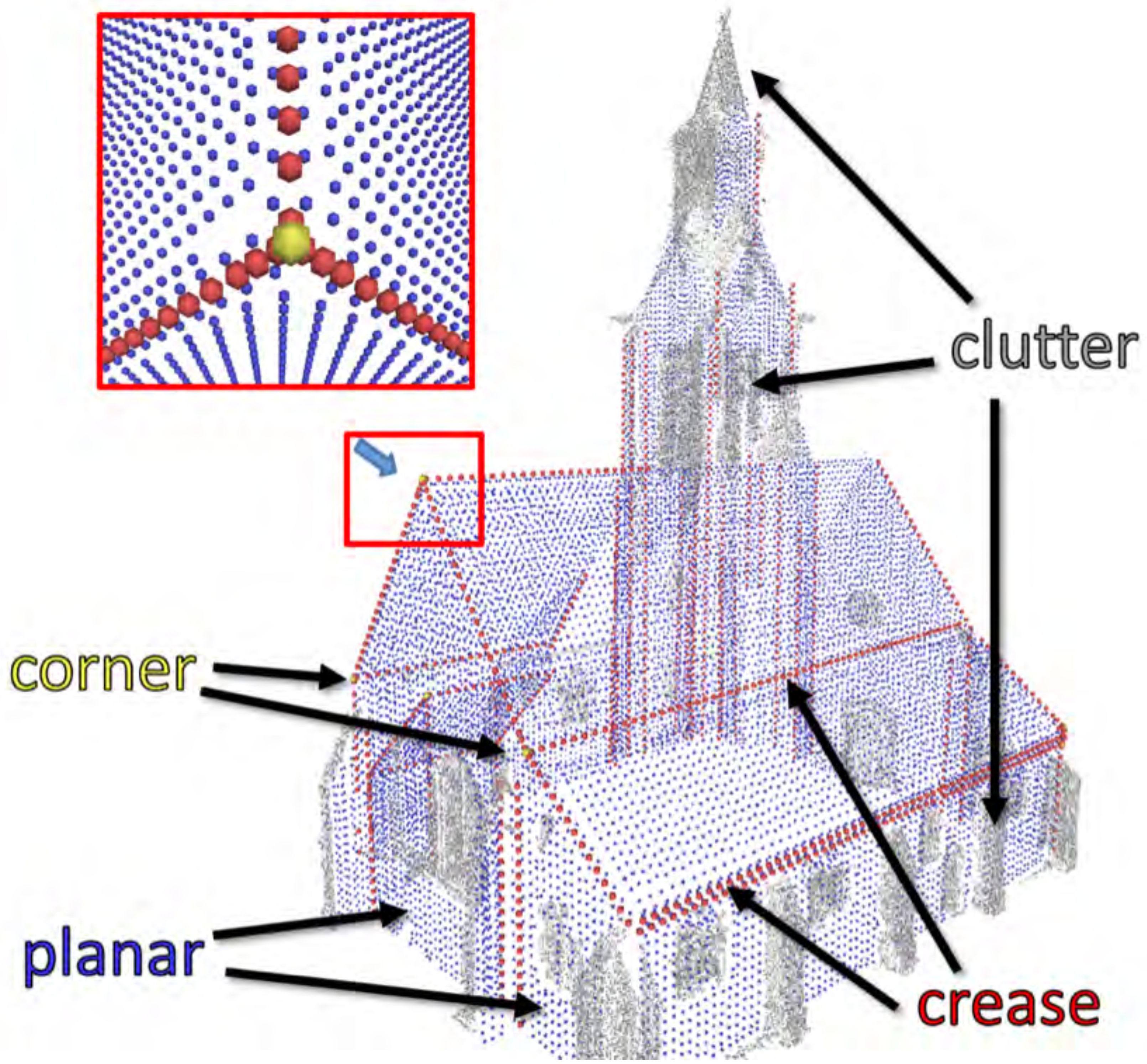


Hybrid Reconstruction

- Use primitives where there exists a good fit, otherwise use surface smoothness, by *structuring* the point set

[Lafarge & Alliez EG'13]





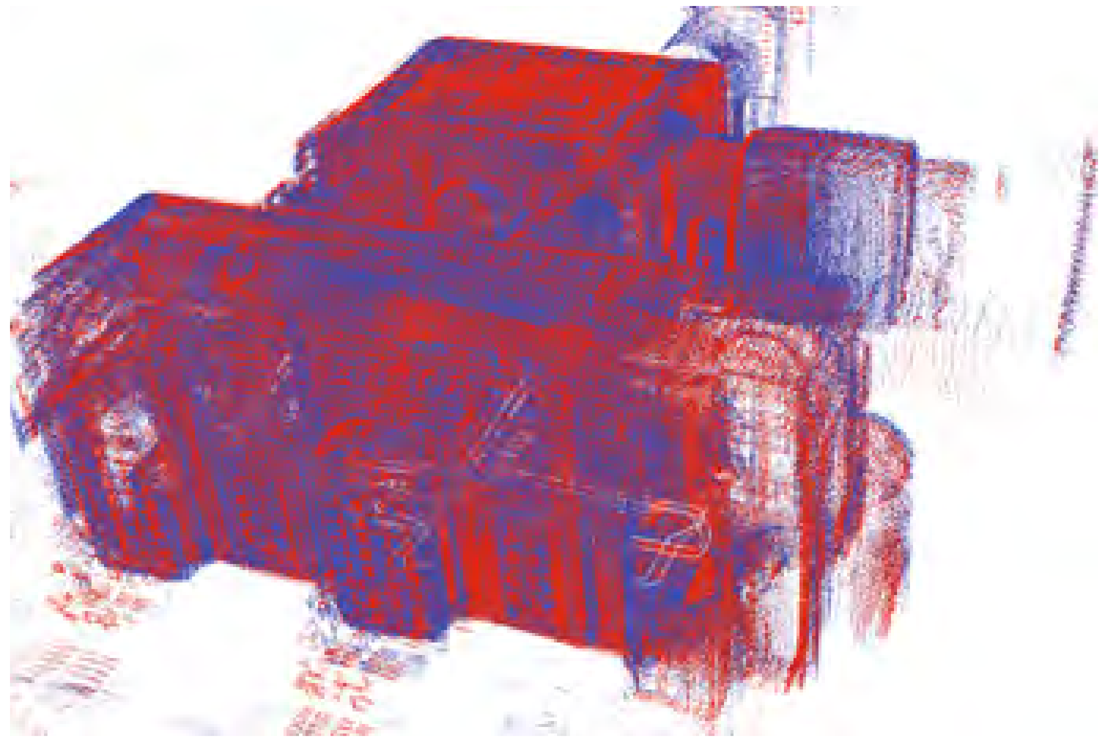
Volume Primitives

- Extract 2D CSG models, extrude to volumetric 3D primitives [Xiao et al. ECCV'12]

Volume Primitives

- Extract 2D CSG models, extrude to volumetric 3D primitives

[Xiao et al. ECCV'12]

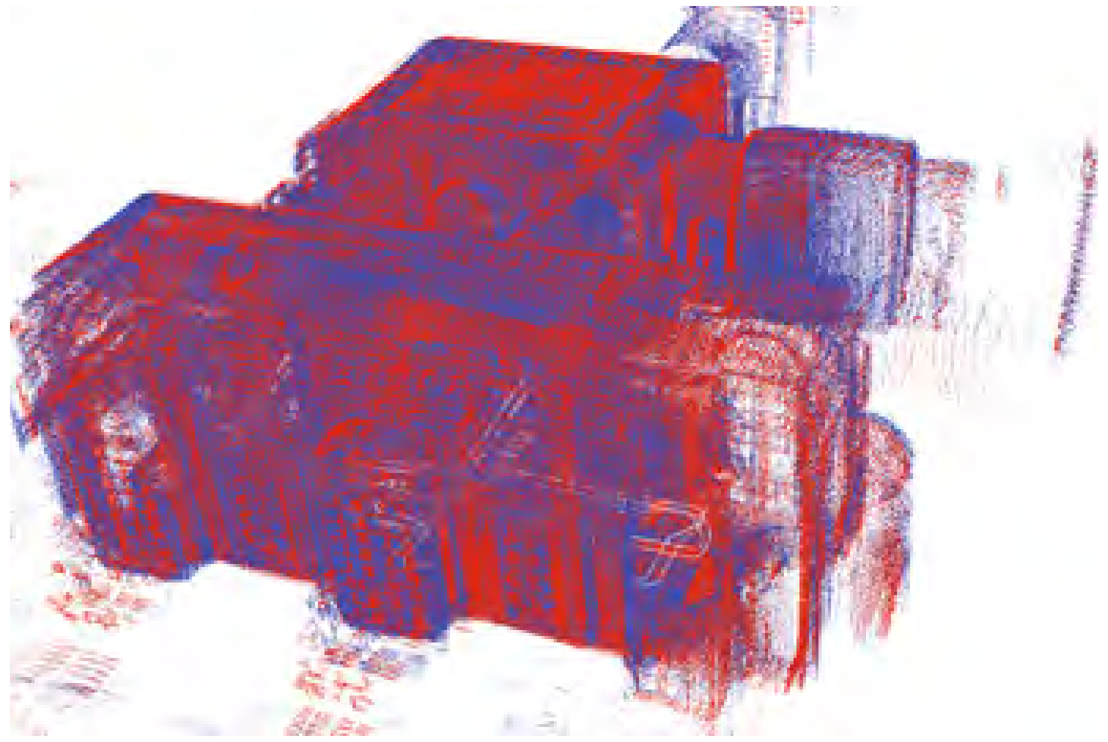


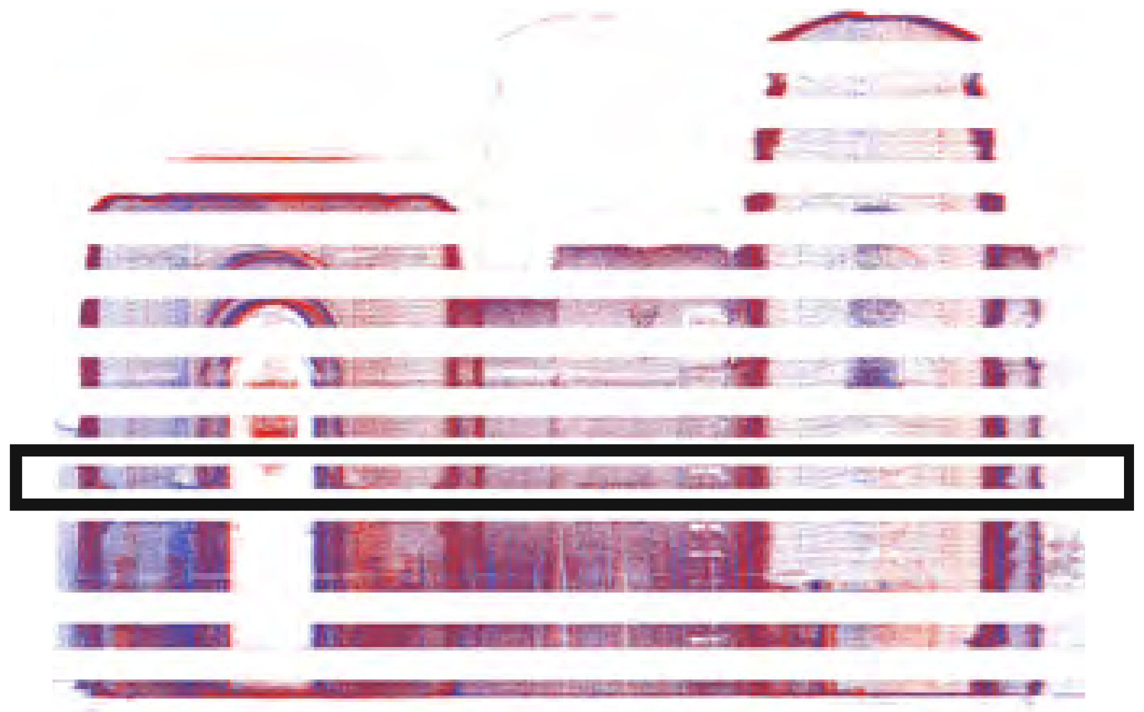
Volume Primitives

- Extract 2D CSG models, extrude to volumetric 3D primitives [Xiao et al. ECCV'12]

Volume Primitives

- Extract 2D CSG models, extrude to volumetric 3D primitives [Xiao et al. ECCV'12]

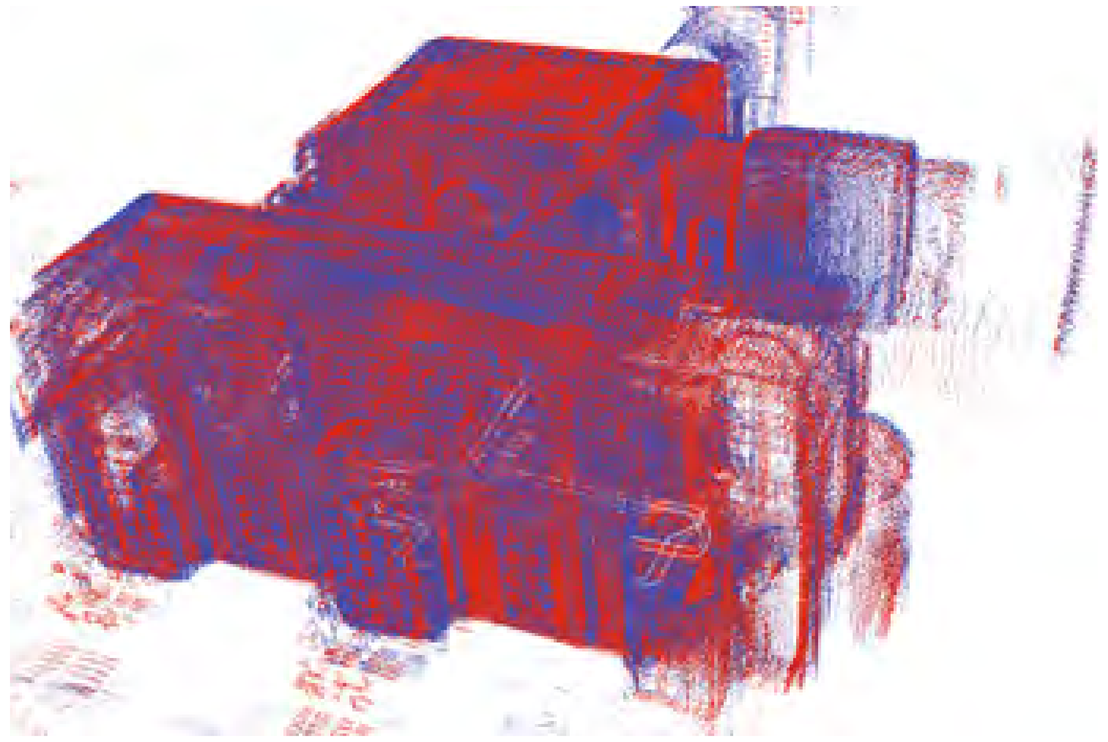


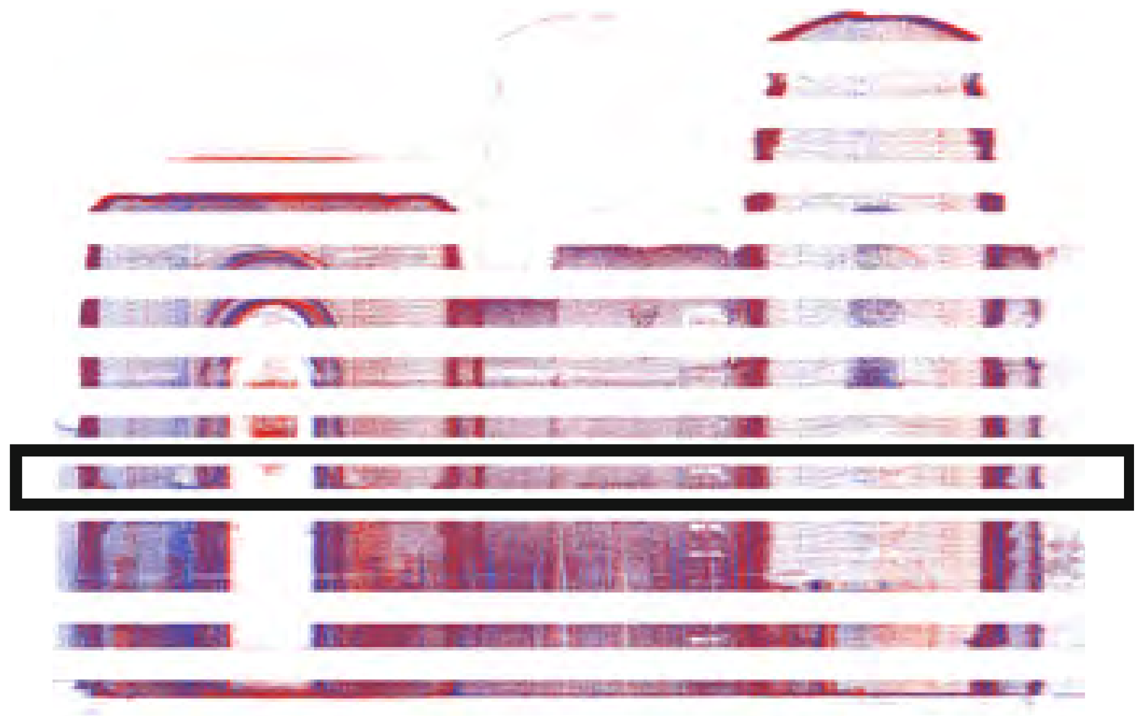


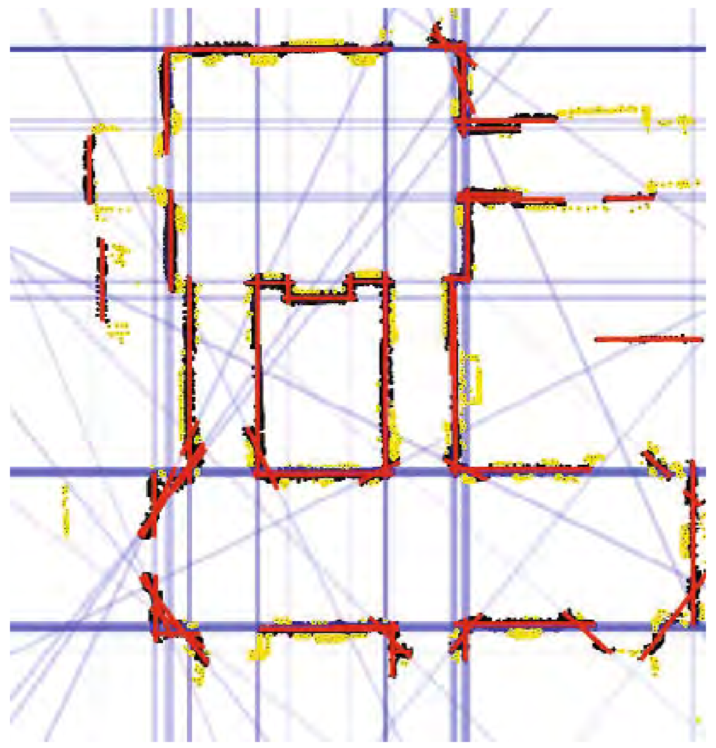
Volume Primitives

- Extract 2D CSG models, extrude to volumetric 3D primitives

[Xiao et al. ECCV'12]



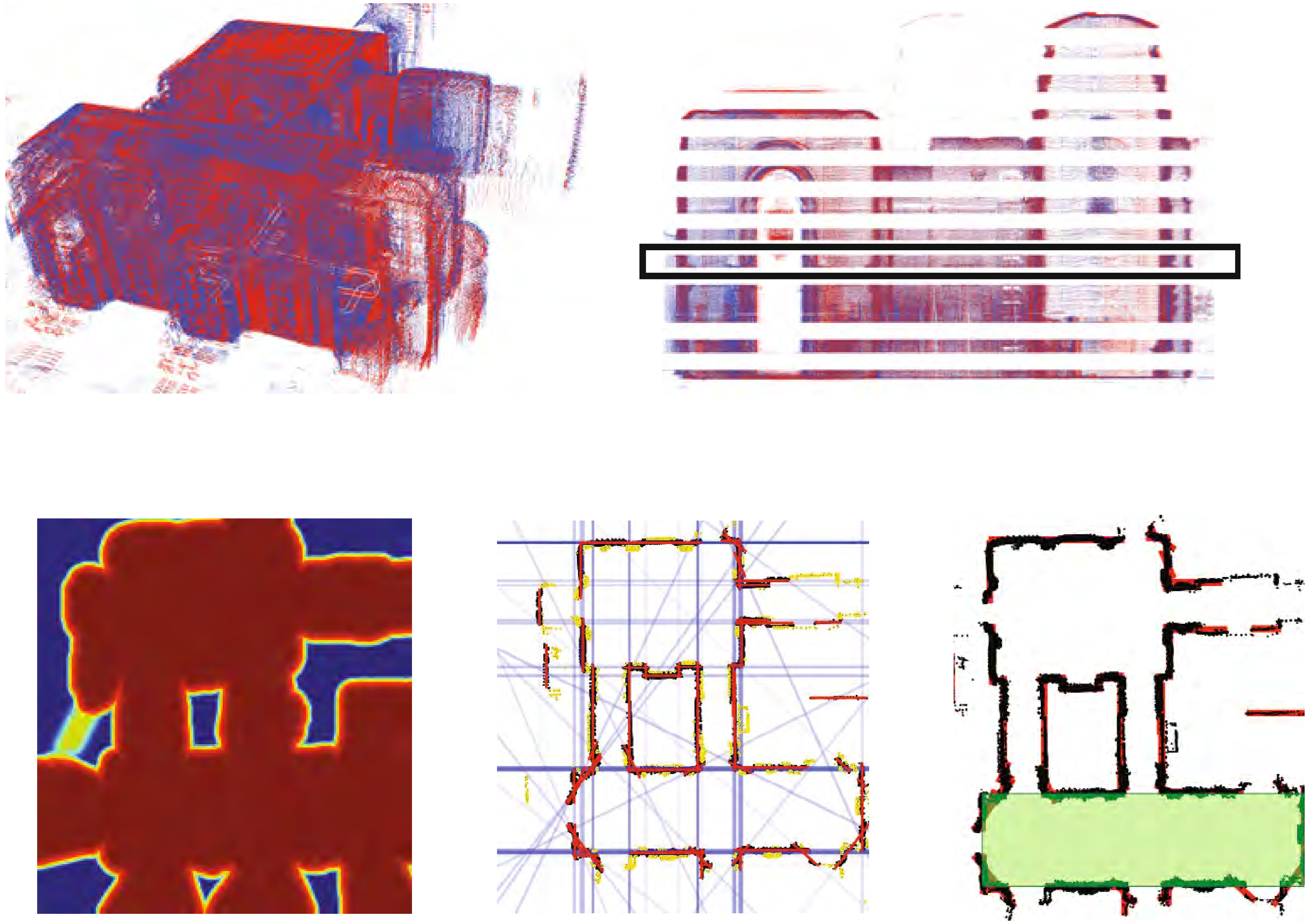




Volume Primitives

- Extract 2D CSG models, extrude to volumetric 3D primitives

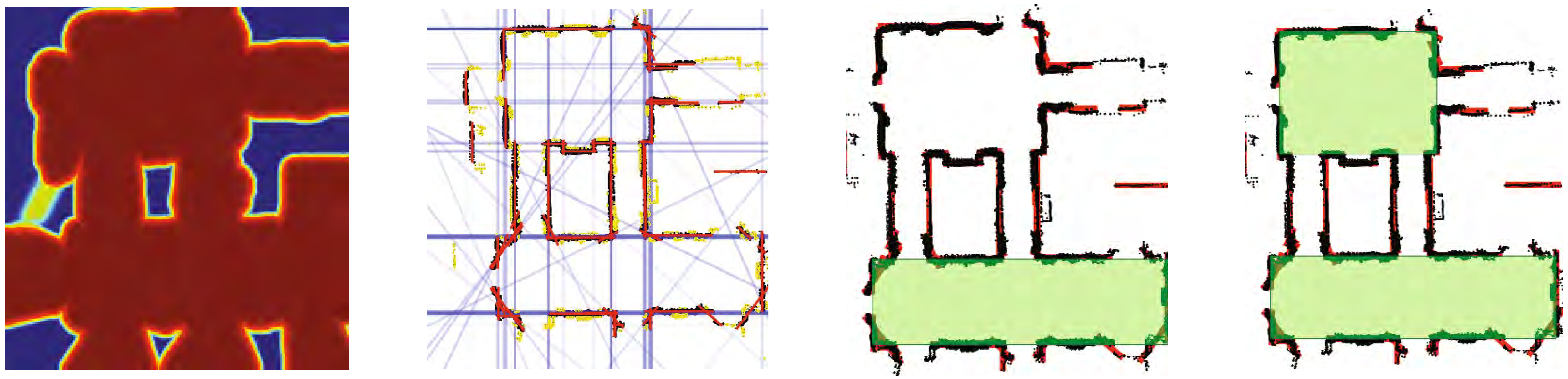
[Xiao et al. ECCV'12]



Volume Primitives

- Extract 2D CSG models, extrude to volumetric 3D primitives

[Xiao et al. ECCV'12]



Volume Primitives

- Extract 2D CSG models, extrude to volumetric 3D primitives

[Xiao et al. ECCV'12]

Volume Primitives

- Extract 2D CSG models, extrude to volumetric 3D primitives

[Xiao et al. ECCV'12]