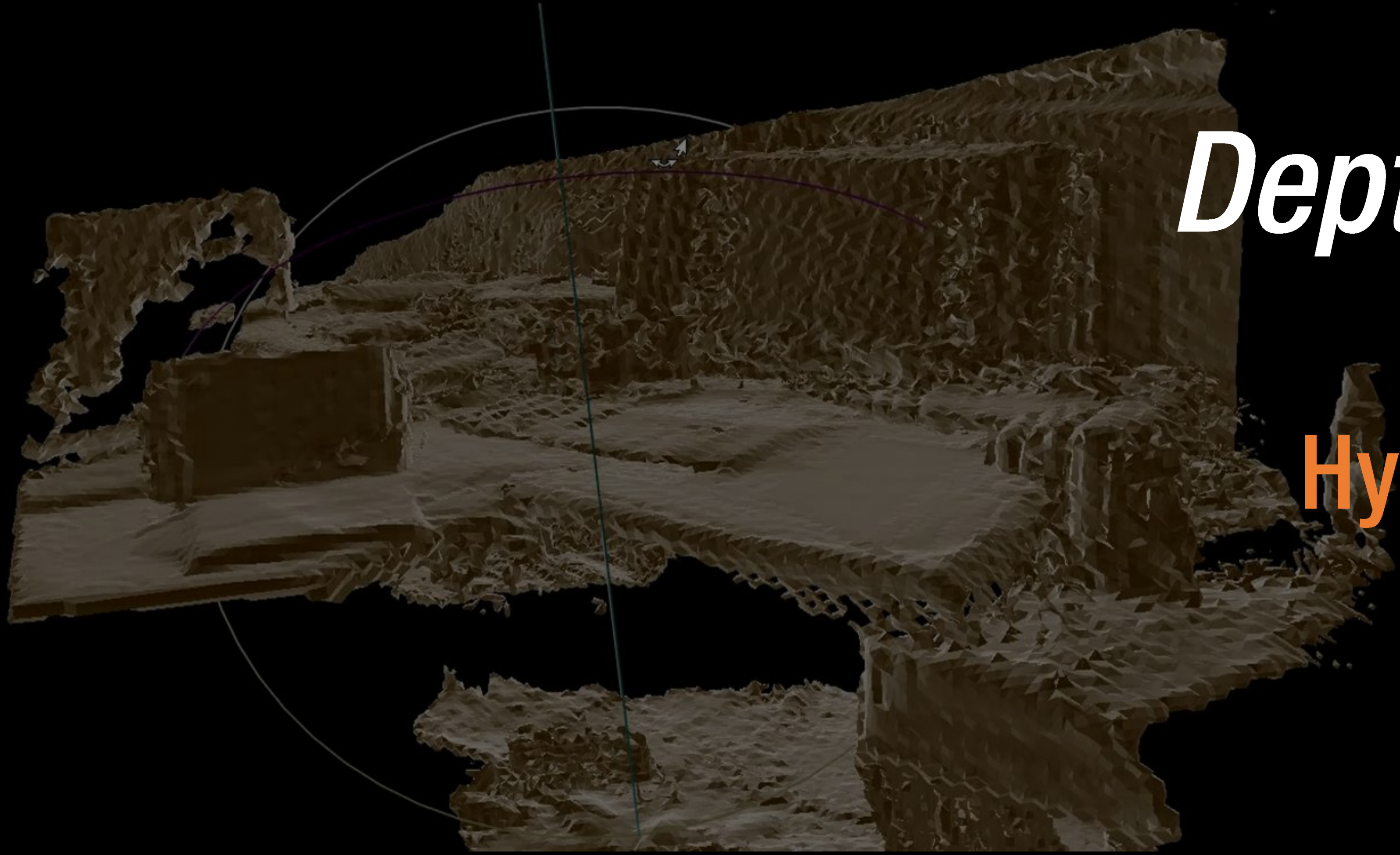
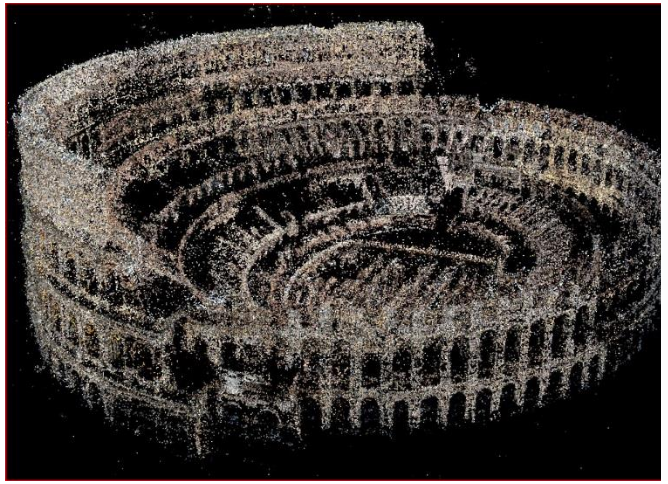
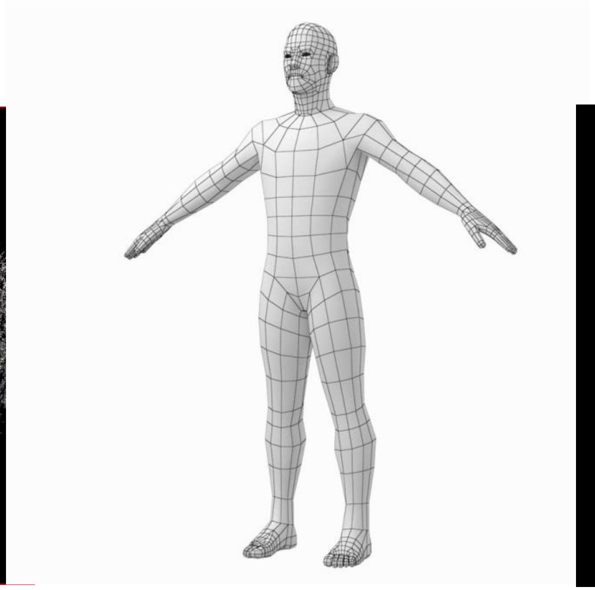


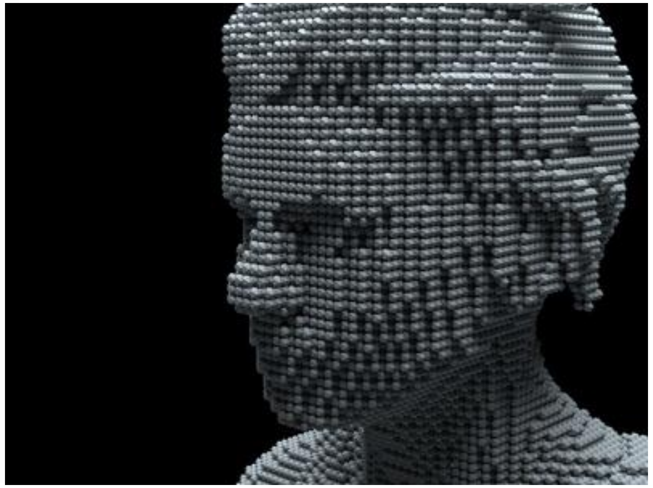
Depth Fusion

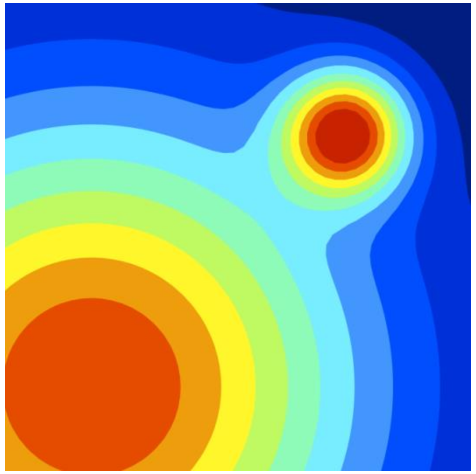
Hyun Soo Park

3D Representation

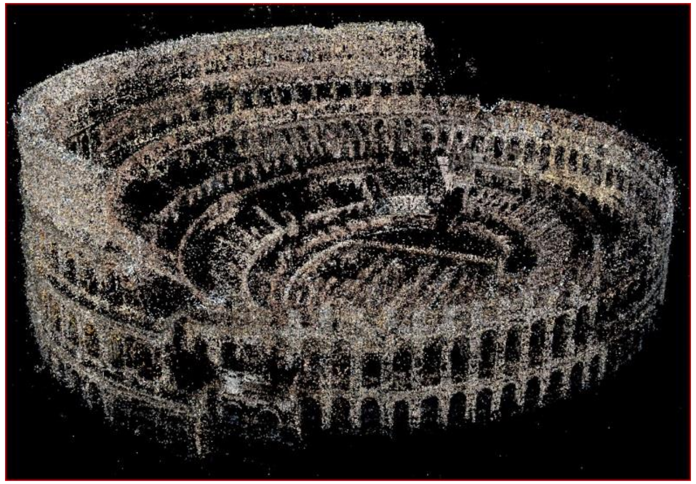

Point cloud

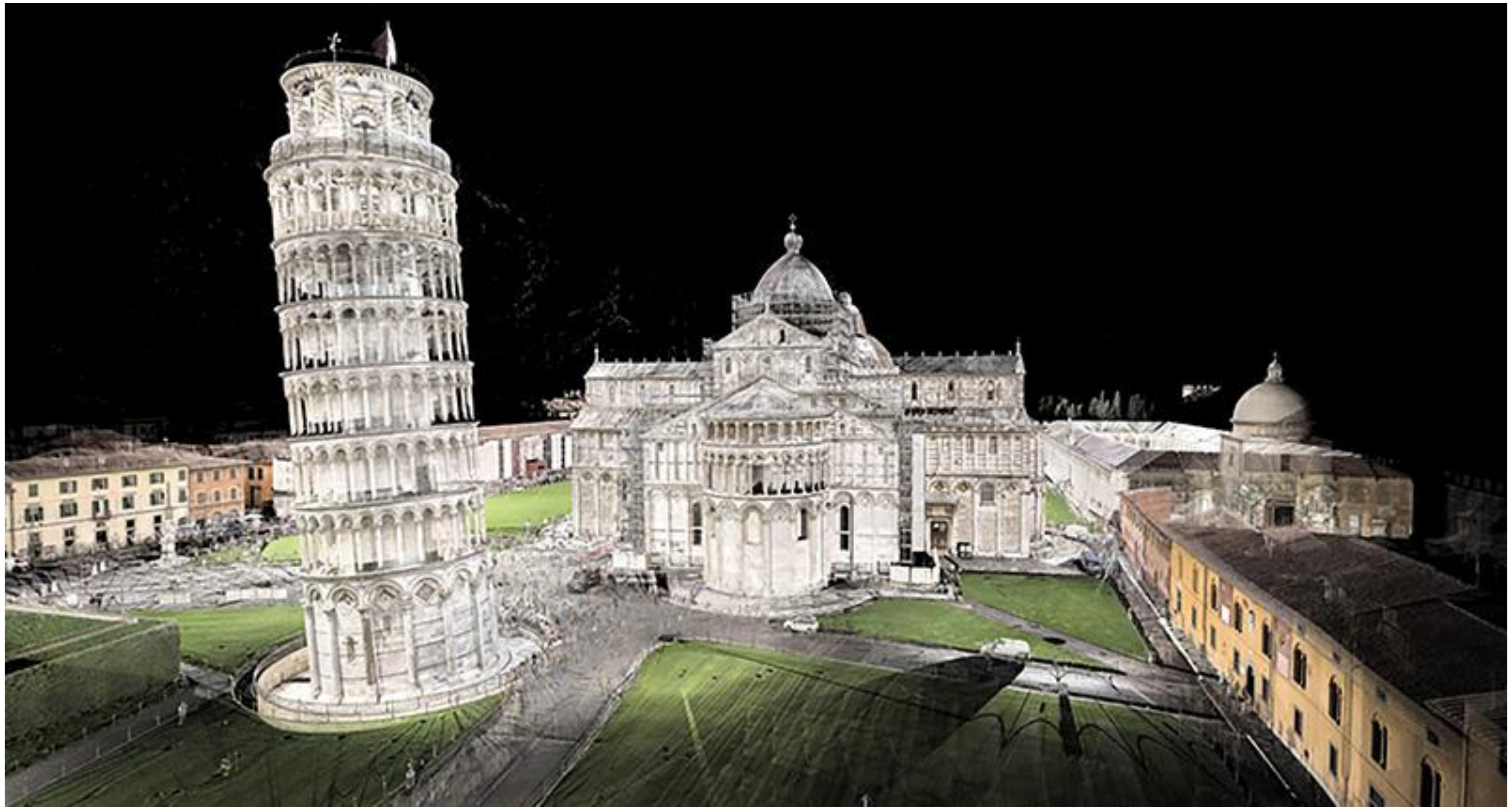

Mesh


Voxel


Implicit function

3D Representation



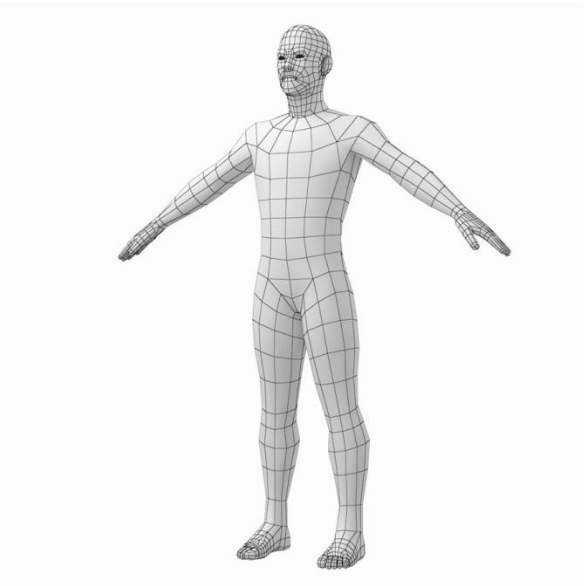


Point cloud

(x, y, z)

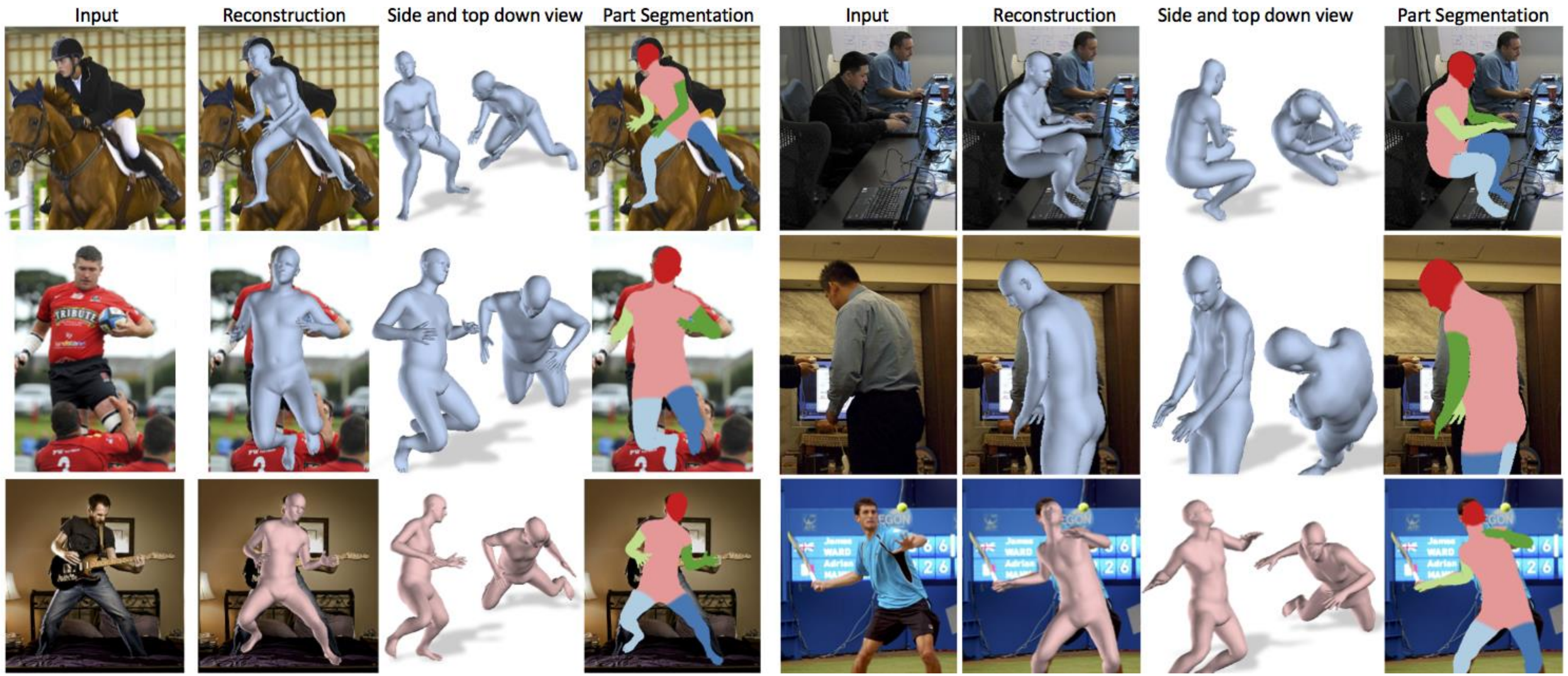
- Non-parametric
- Non-volumetric
- Topology-agnostic

3D Representation

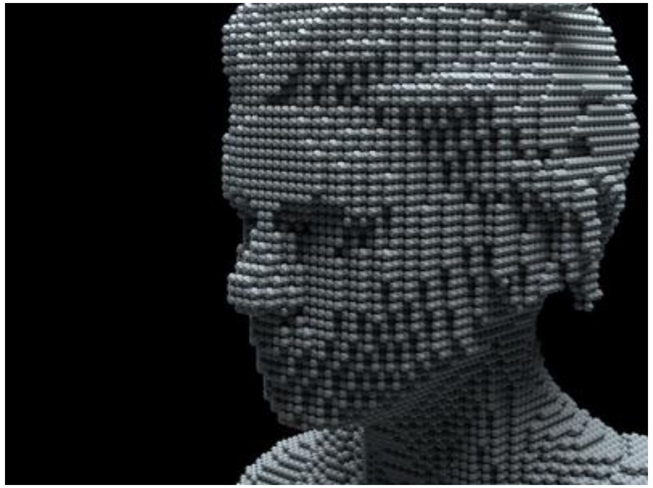


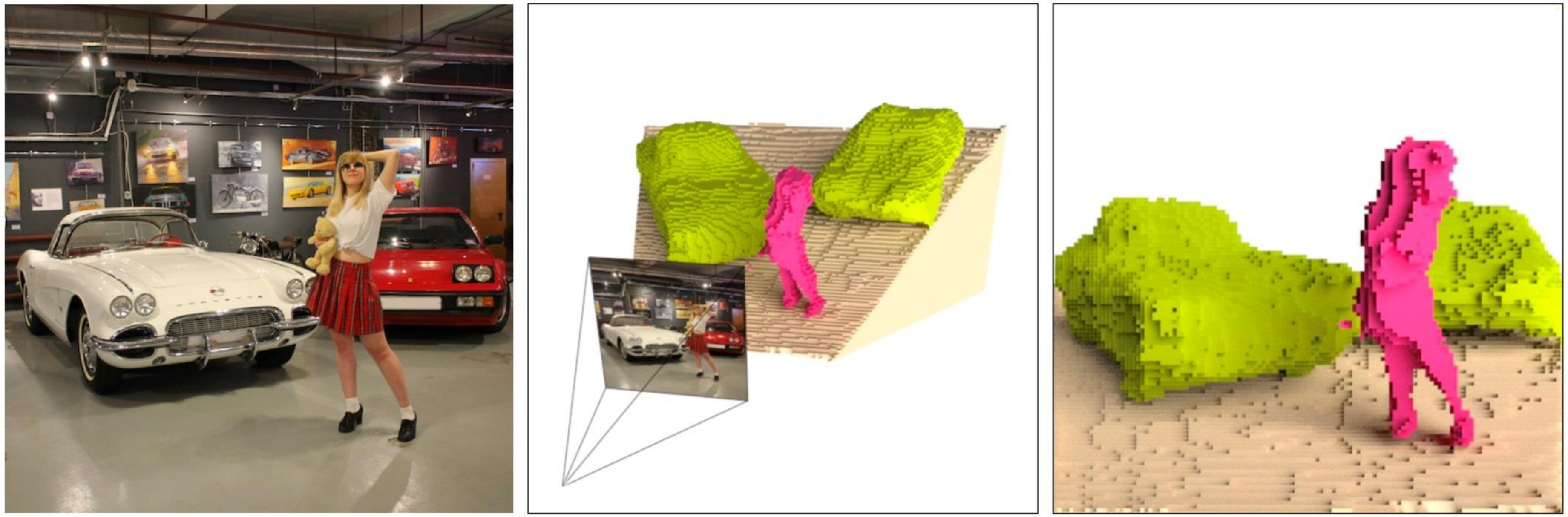
Mesh $G(E,V)$

- Parametric
- Controllable
- Volumetric
- Compact
- Fixed topology



3D Representation



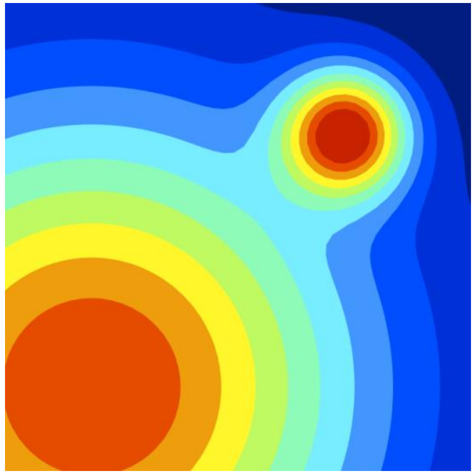


Voxel

$$V_{ijk} = \begin{cases} 1 & \text{if occupied} \\ 0 & \text{if empty} \end{cases}$$

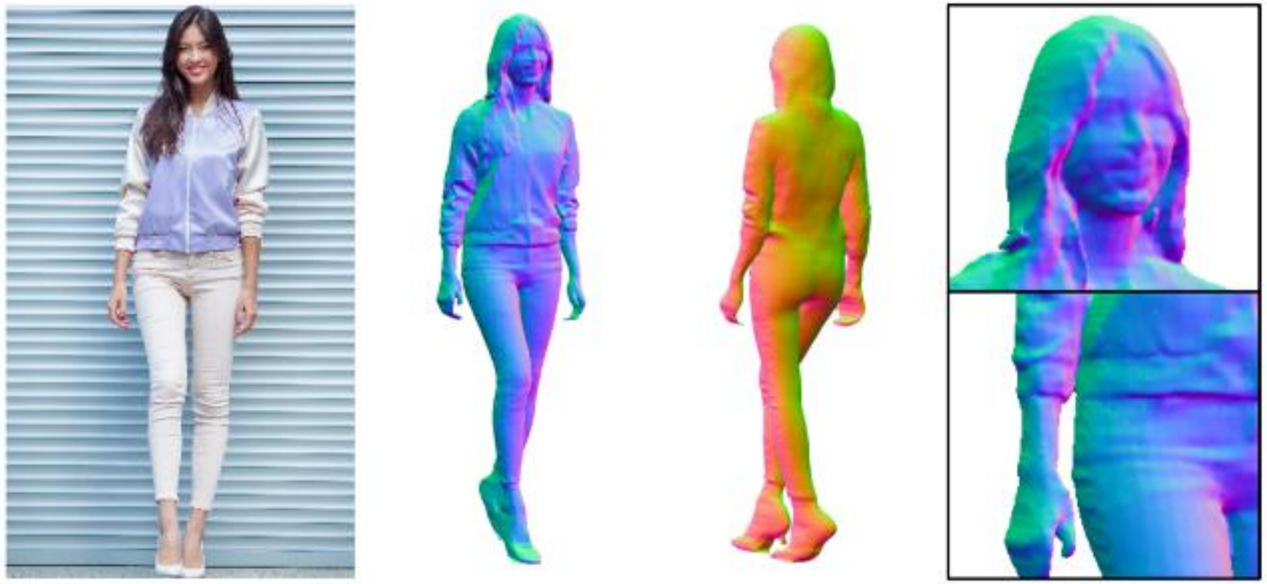
- Volumetric
- Non-parametric
- Topology agnostic
- Cubic order of memory

3D Representation



Implicit function

$$f(x) = c$$



- Non-parametric
- Volumetric
- Topology agnostic
- Compact

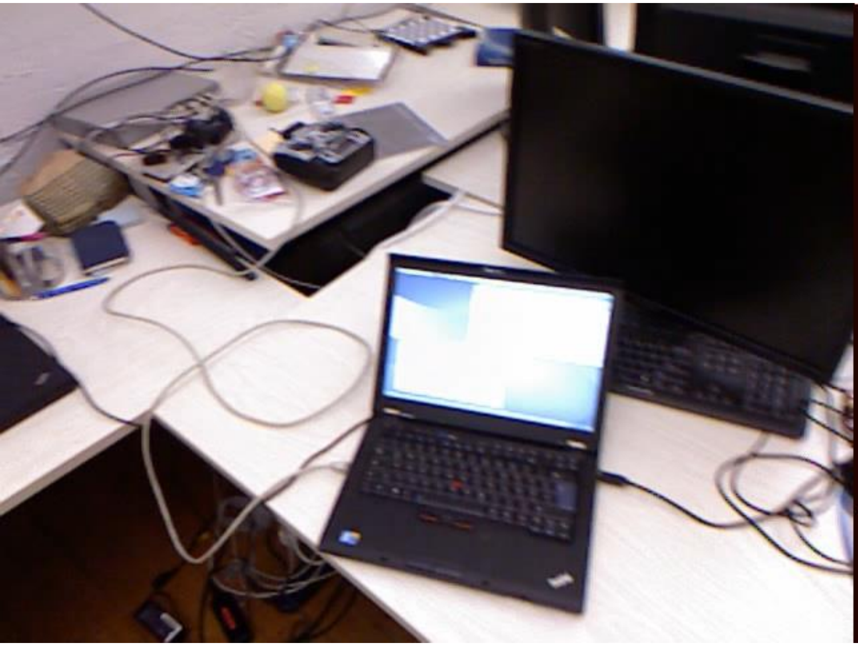
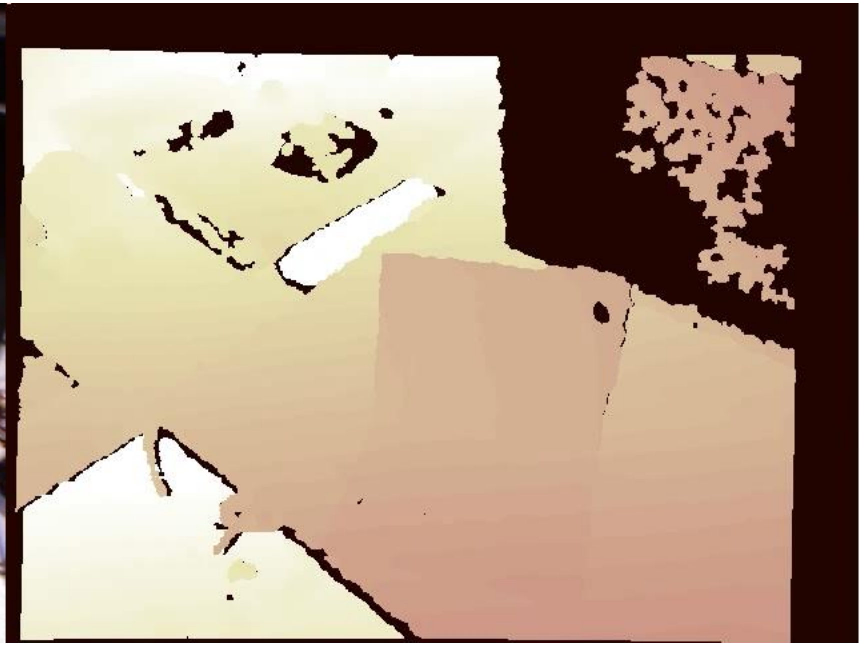
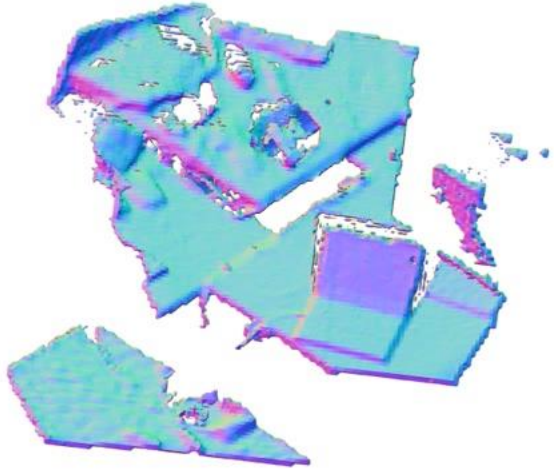
SIGGRAPH Talks 2011

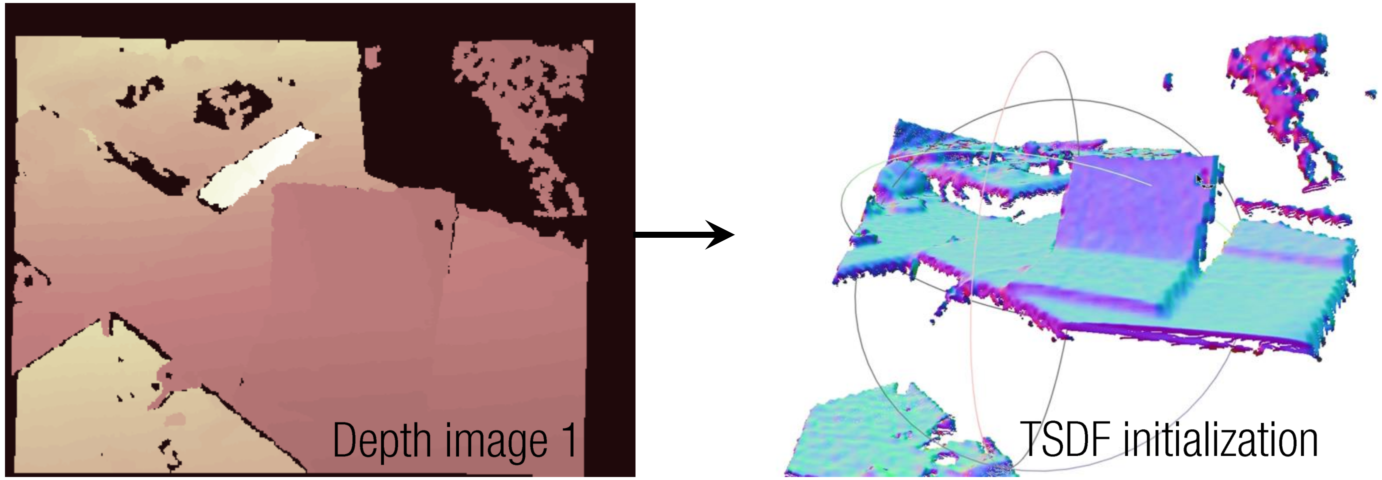
KinectFusion:

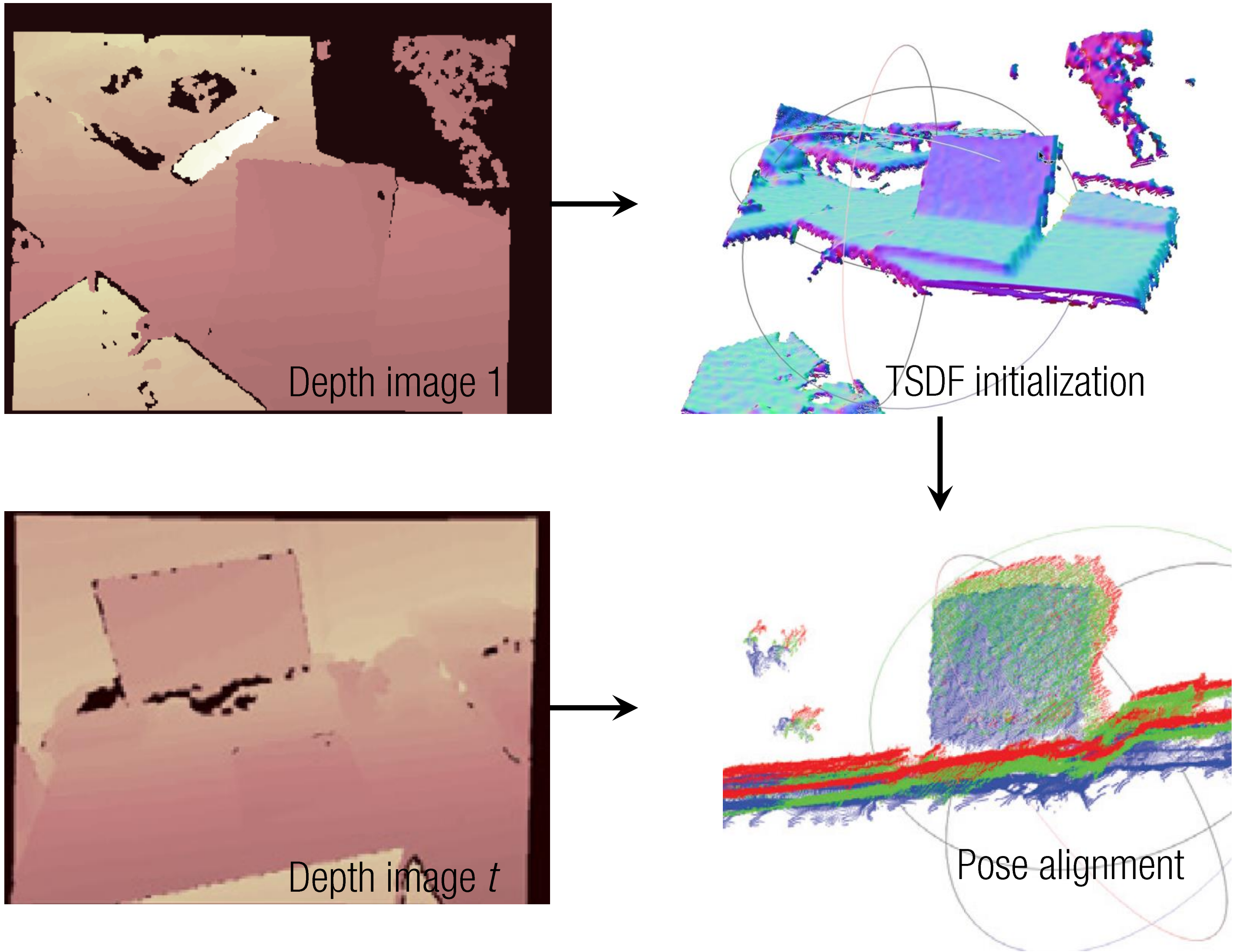
Real-Time Dynamic 3D Surface Reconstruction and Interaction

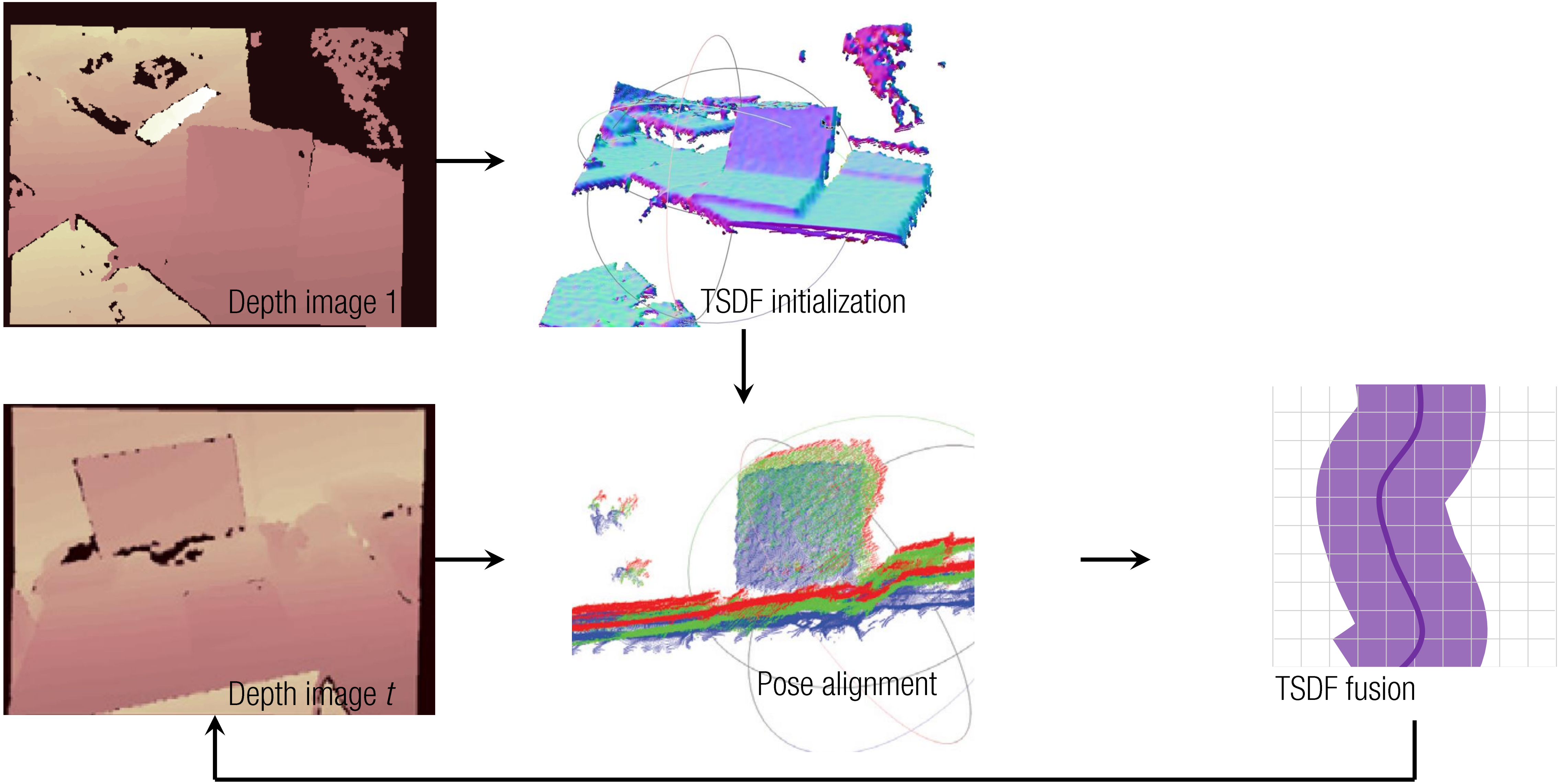
Shahram Izadi 1, Richard Newcombe 2, David Kim 1,3, Otmar Hilliges 1, David Molyneaux 1,4, Pushmeet Kohli 1, Jamie Shotton 1, Steve Hodges 1, Dustin Freeman 5, Andrew Davison 2, Andrew Fitzgibbon 1

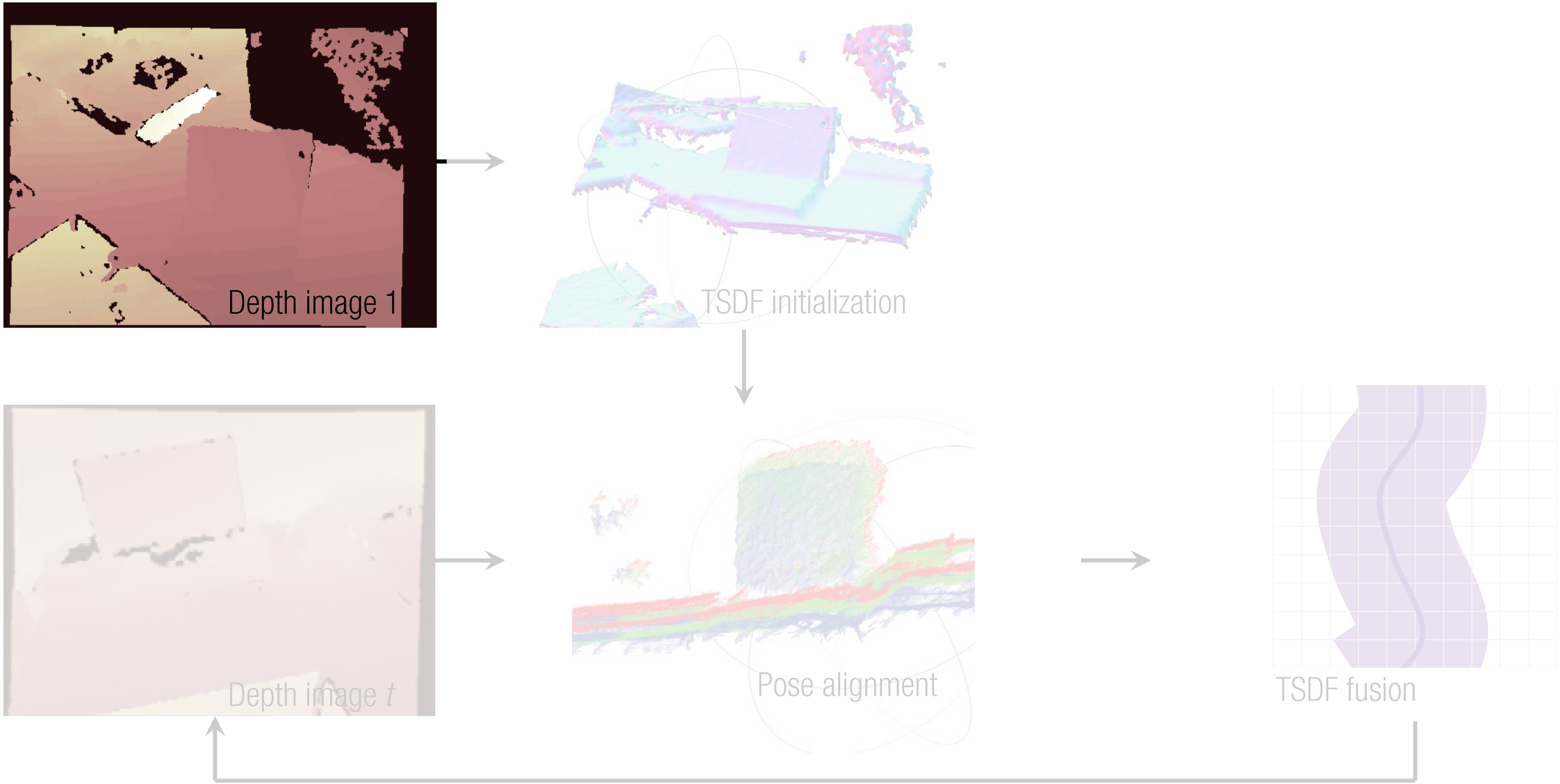
1 Microsoft Research Cambridge 2 Imperial College London
3 Newcastle University 4 Lancaster University
5 University of Toronto

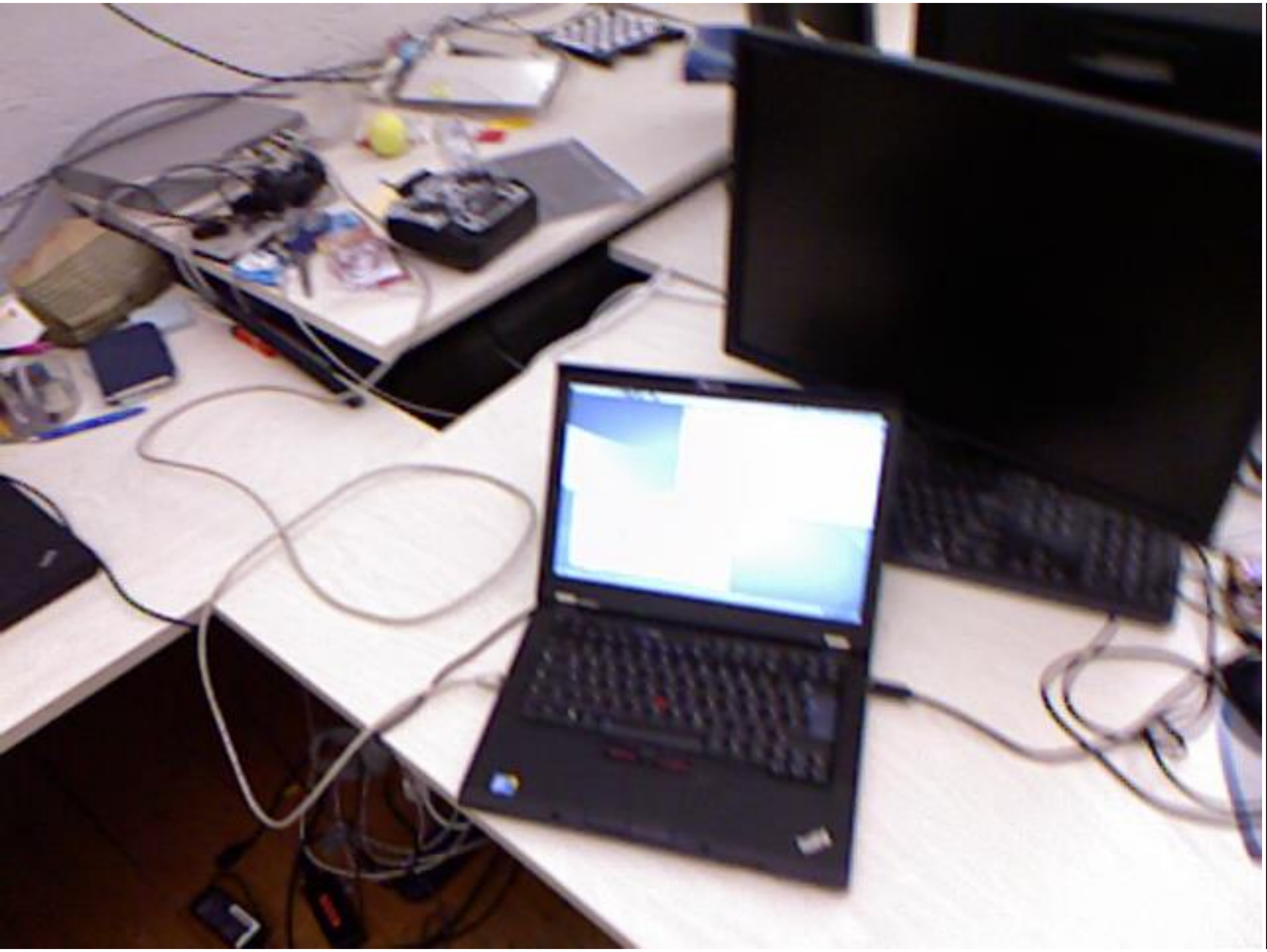
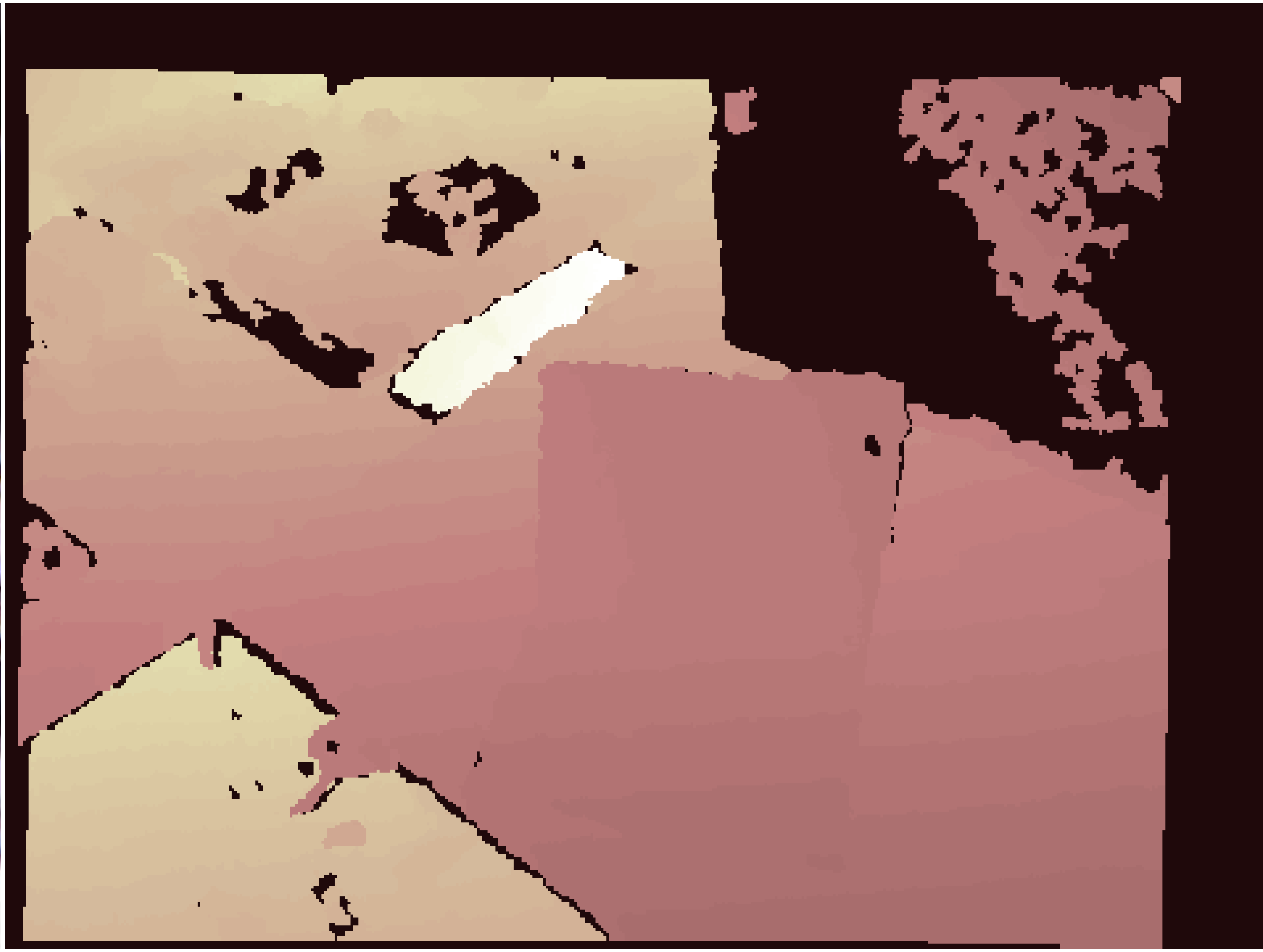
  


Depth image 1
TSDF initialization

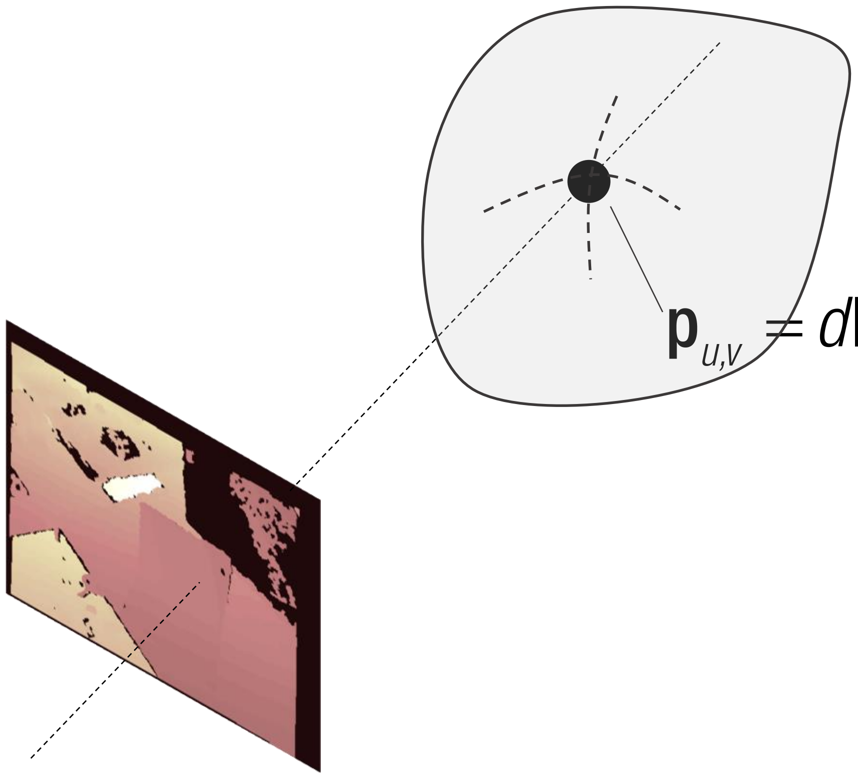

Depth image 1
TSDF initialization
Depth image t
Pose alignment


Depth image 1
TSDF initialization
Depth image t
Pose alignment
TSDF fusion


Depth image 1
TSDF initialization
Depth image *t*
Pose alignment
TSDF fusion

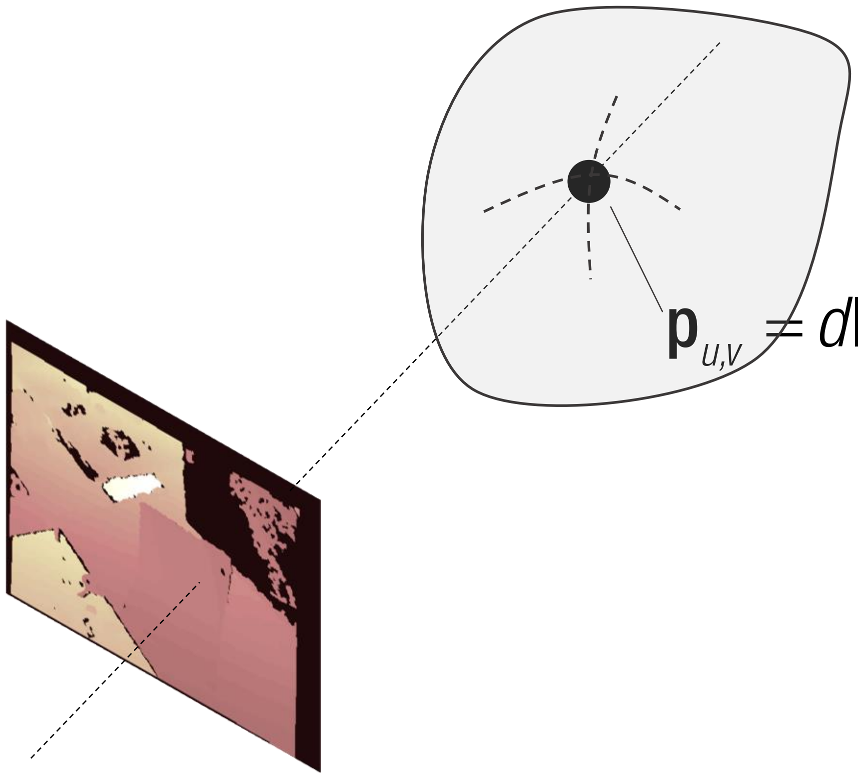



3D Reconstruction from Depth

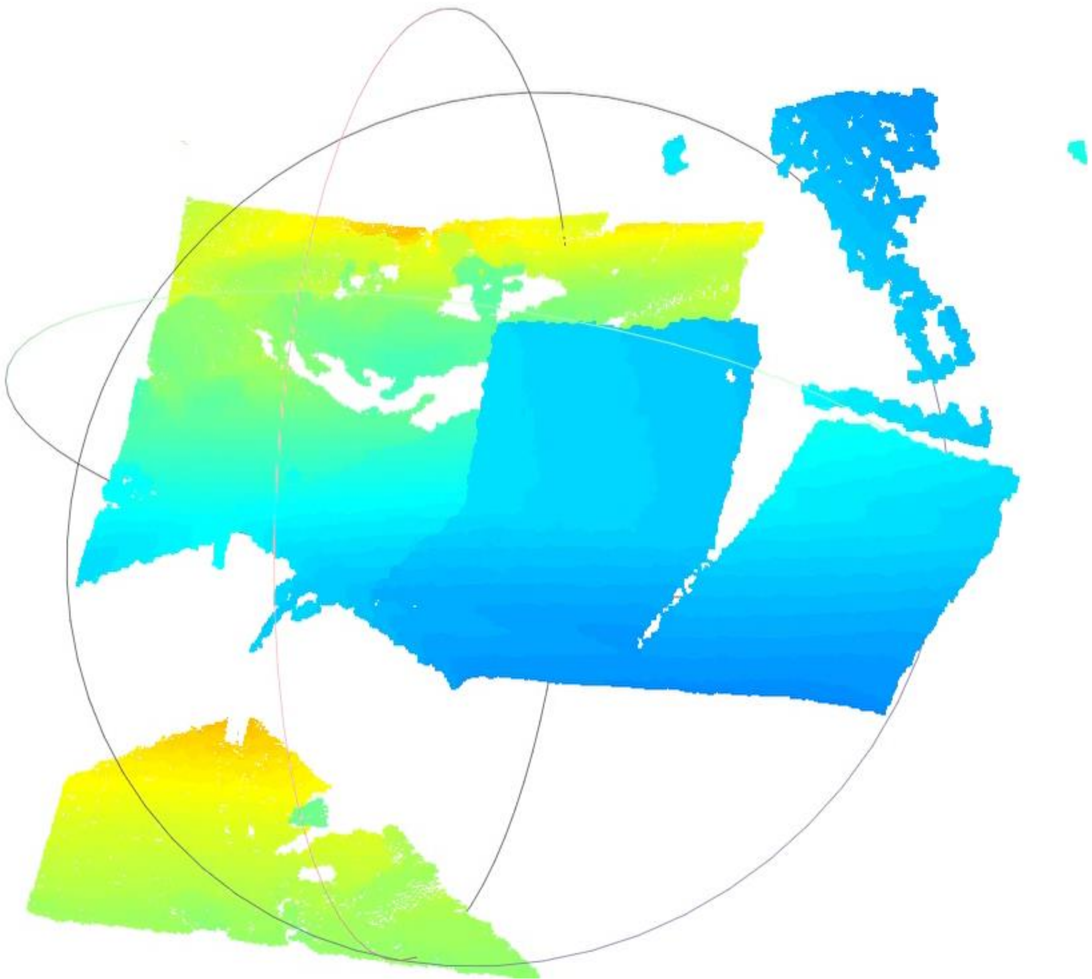


$$\mathbf{p}_{u,v} = d\mathbf{K}^{-1}\begin{bmatrix} u \\ v \\ 1 \end{bmatrix}$$

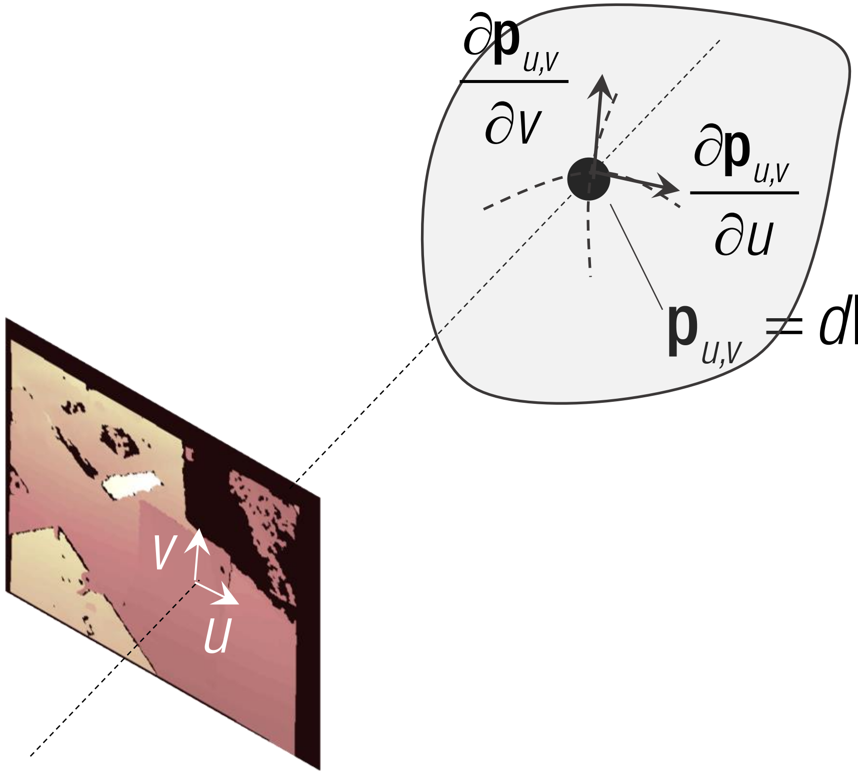
3D Reconstruction from Depth



$$\mathbf{p}_{u,v} = d\mathbf{K}^{-1}\begin{bmatrix} u \\ v \\ 1 \end{bmatrix}$$



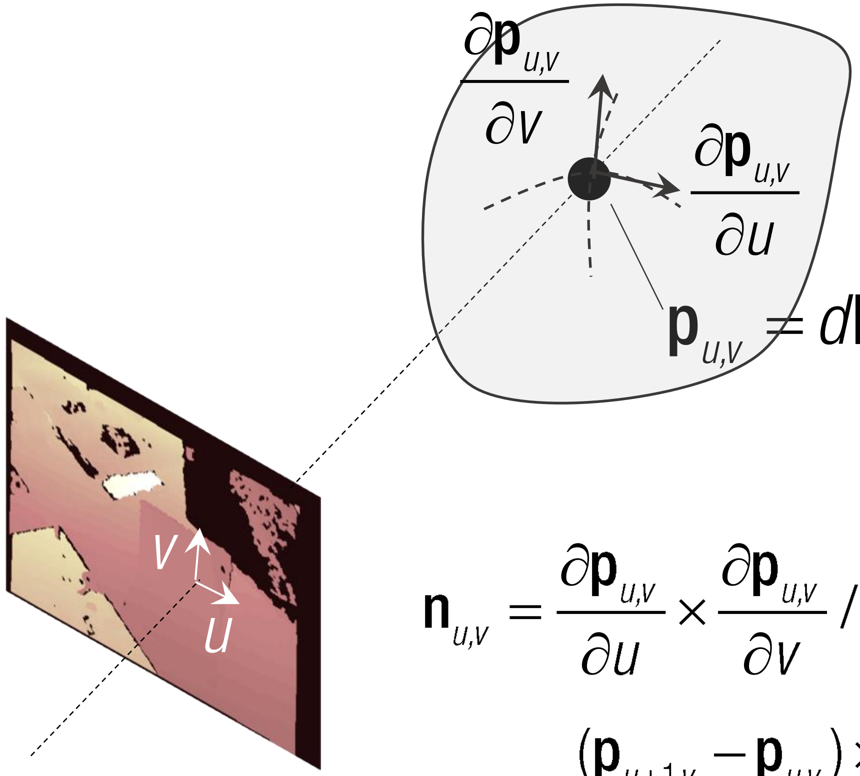
Surface Normals from Depths



$$\frac{\partial \mathbf{p}_{u,v}}{\partial v} \quad \frac{\partial \mathbf{p}_{u,v}}{\partial u}$$

$$\mathbf{p}_{u,v} = d\mathbf{K}^{-1}\begin{bmatrix} u \\ v \\ 1 \end{bmatrix}$$

Surface Normals from Depths

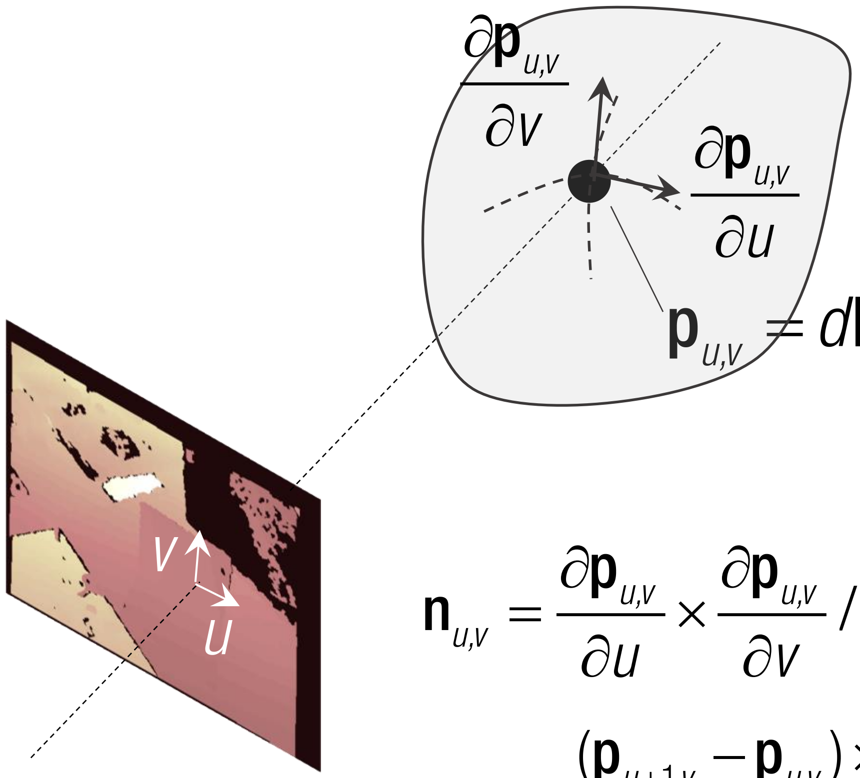


$$\mathbf{p}_{u,v} = d\mathbf{K}^{-1}\begin{bmatrix} u \\ v \\ 1 \end{bmatrix}$$

$$\mathbf{n}_{u,v} = \frac{\partial \mathbf{p}_{u,v}}{\partial u} \times \frac{\partial \mathbf{p}_{u,v}}{\partial v} \Big/ \left\| \frac{\partial \mathbf{p}_{u,v}}{\partial u} \times \frac{\partial \mathbf{p}_{u,v}}{\partial v} \right\|$$

$$= \frac{(\mathbf{p}_{u+1,v} - \mathbf{p}_{u,v}) \times (\mathbf{p}_{u,v+1} - \mathbf{p}_{u,v})}{\left\| (\mathbf{p}_{u+1,v} - \mathbf{p}_{u,v}) \times (\mathbf{p}_{u,v+1} - \mathbf{p}_{u,v}) \right\|}$$

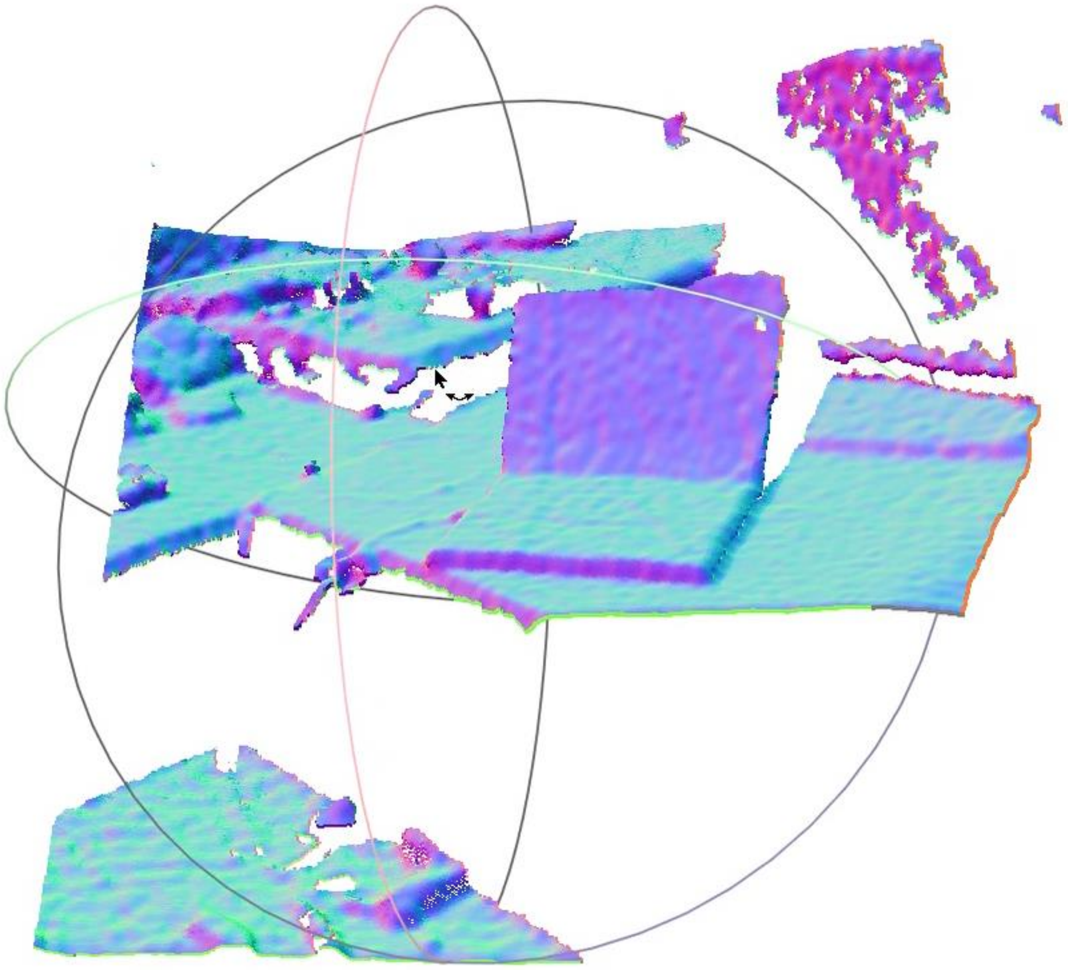
Surface Normals from Depths

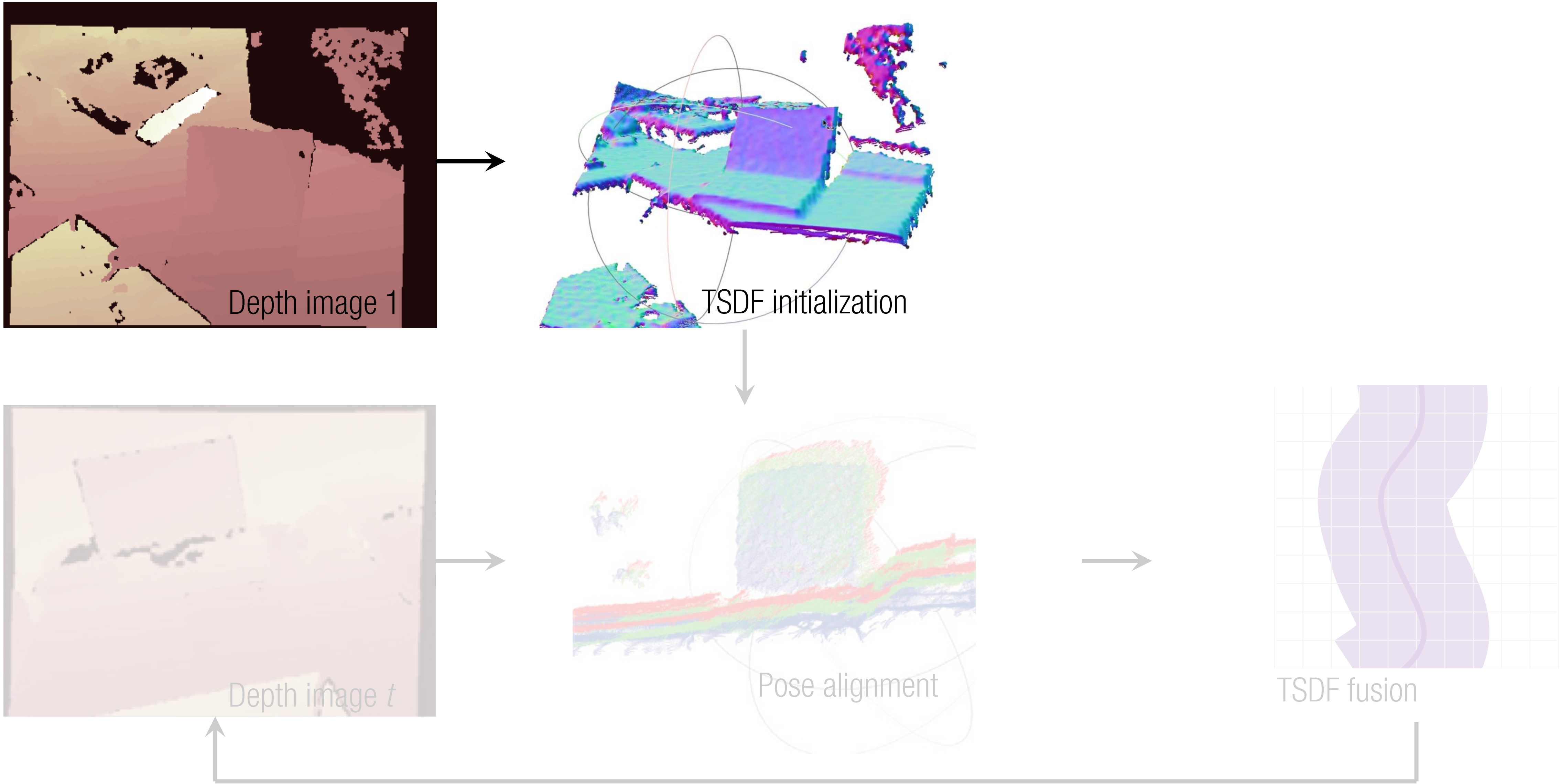


$$\mathbf{p}_{u,v} = d\mathbf{K}^{-1}\begin{bmatrix} u \\ v \\ 1 \end{bmatrix}$$

$$\mathbf{n}_{u,v} = \frac{\partial \mathbf{p}_{u,v}}{\partial u} \times \frac{\partial \mathbf{p}_{u,v}}{\partial v} \Big/ \left\| \frac{\partial \mathbf{p}_{u,v}}{\partial u} \times \frac{\partial \mathbf{p}_{u,v}}{\partial v} \right\|$$

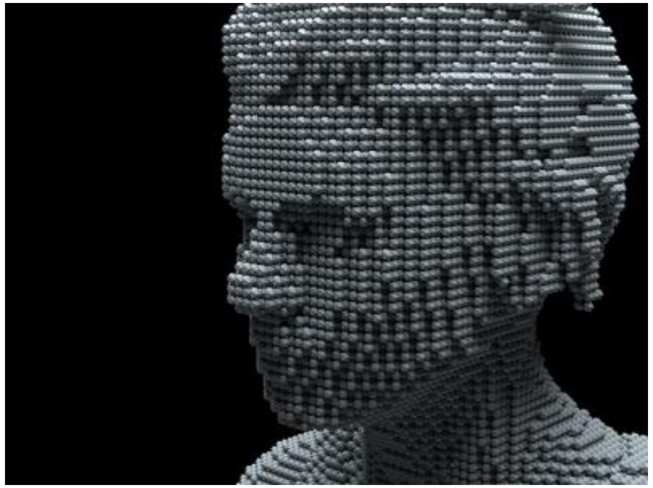
$$= \frac{(\mathbf{p}_{u+1,v} - \mathbf{p}_{u,v}) \times (\mathbf{p}_{u,v+1} - \mathbf{p}_{u,v})}{\left\| (\mathbf{p}_{u+1,v} - \mathbf{p}_{u,v}) \times (\mathbf{p}_{u,v+1} - \mathbf{p}_{u,v}) \right\|}$$





Depth image 1
TSDF initialization
Depth image t
Pose alignment
TSDF fusion

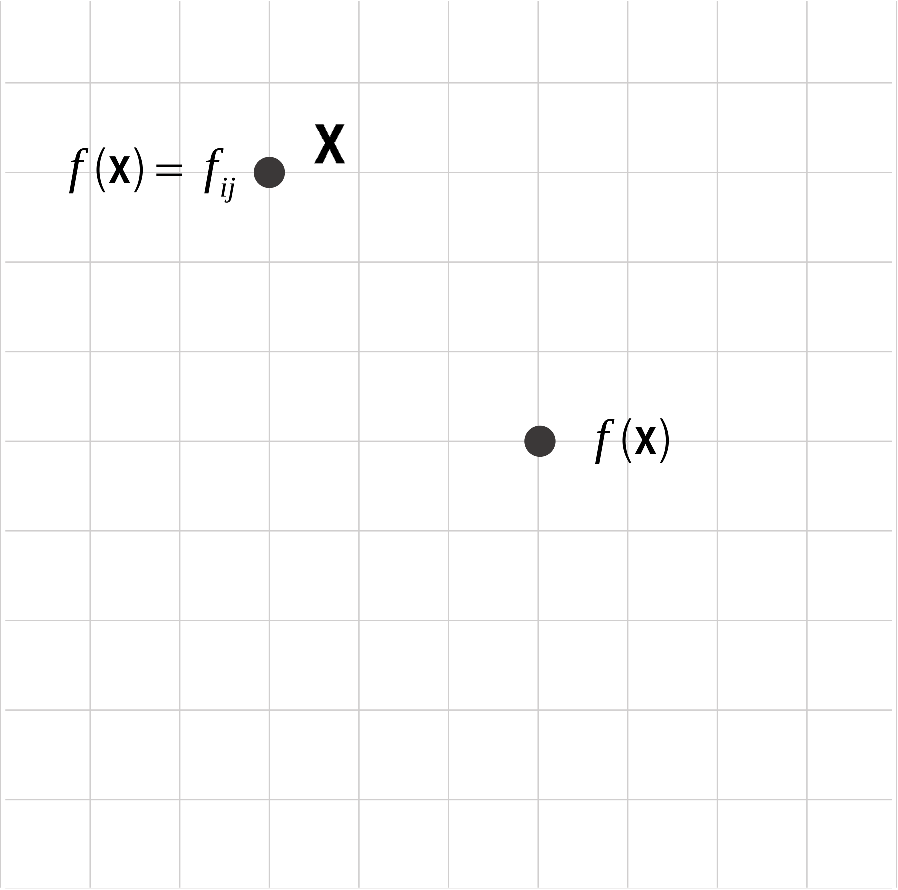
Distance Field



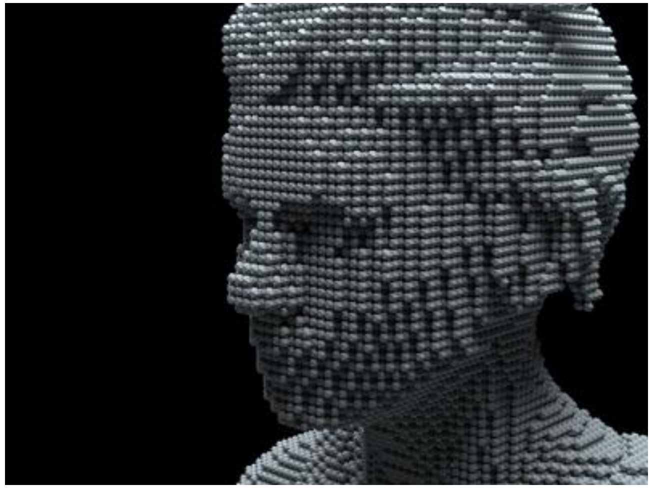
Voxel

$$V_{ijk} = \begin{cases} 1 & \text{if occupied} \\ 0 & \text{if empty} \end{cases}$$

- Volumetric
- Non-parametric
- Topology agnostic
- Cubic order of memory



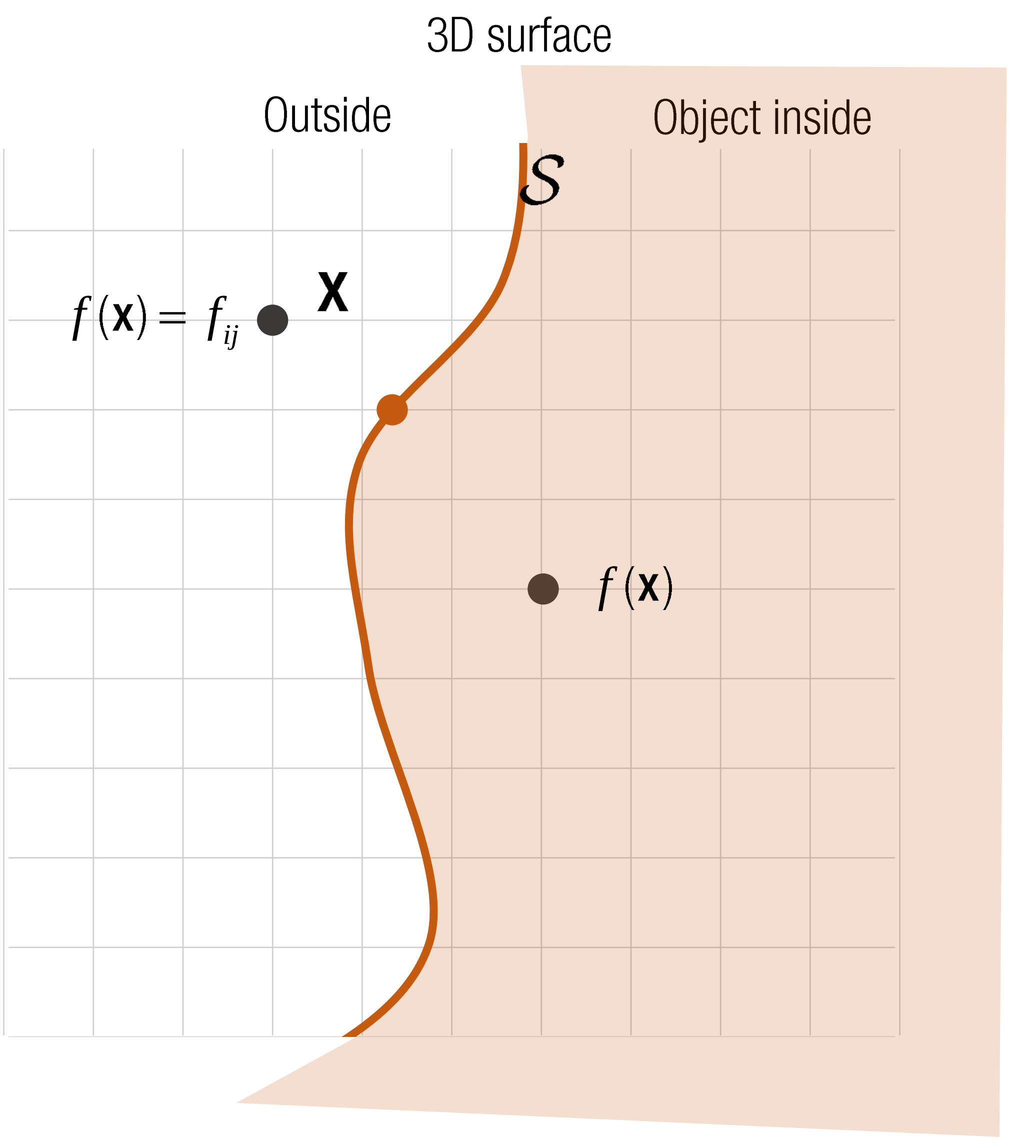
Distance Field



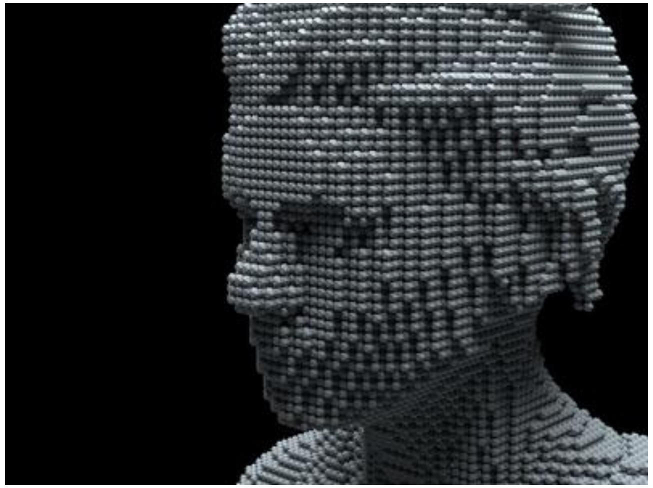
Voxel

$$V_{ijk} = \begin{cases} 1 & \text{if occupied} \\ 0 & \text{if empty} \end{cases}$$

- Volumetric
- Non-parametric
- Topology agnostic
- Cubic order of memory



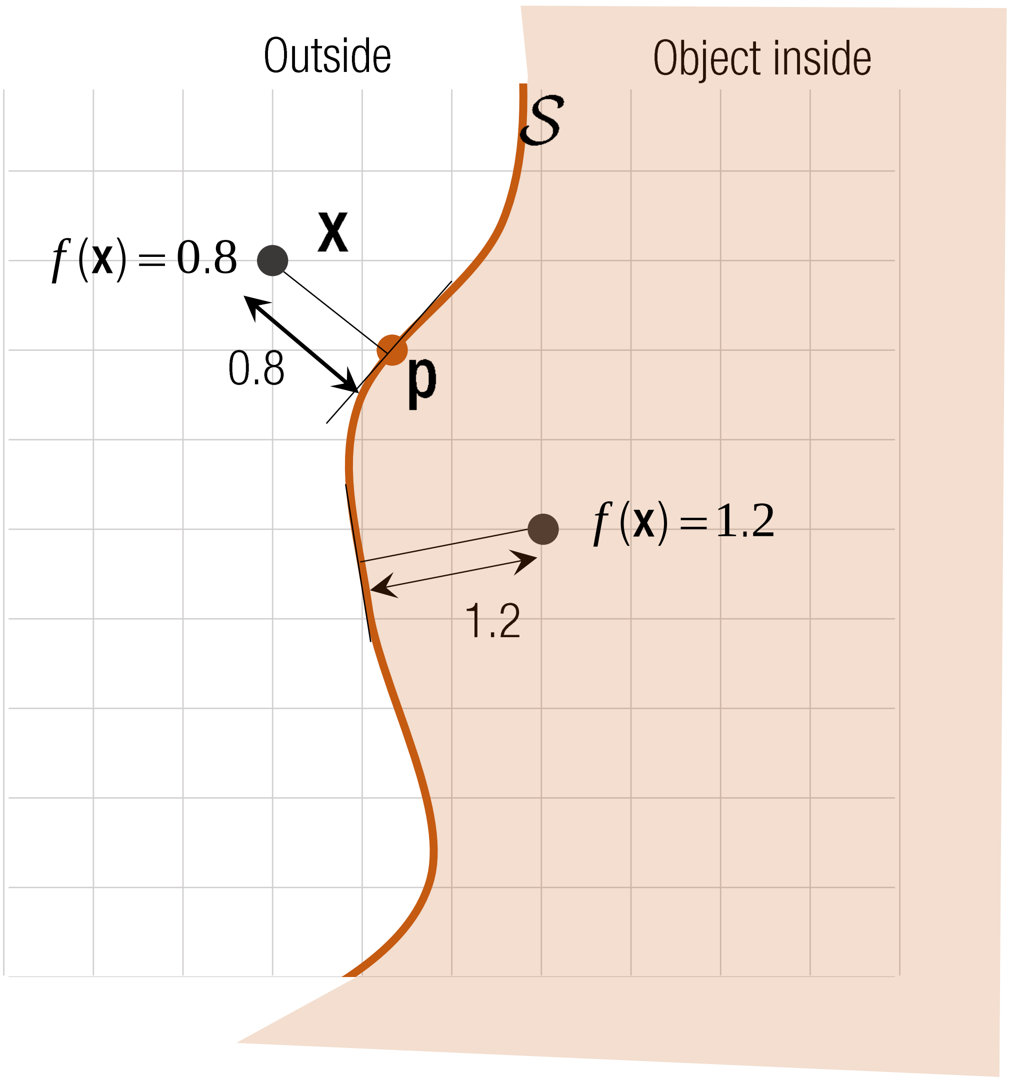
Distance Field



Voxel

$$V_{ijk} = \begin{cases} 1 & \text{if occupied} \\ 0 & \text{if empty} \end{cases}$$

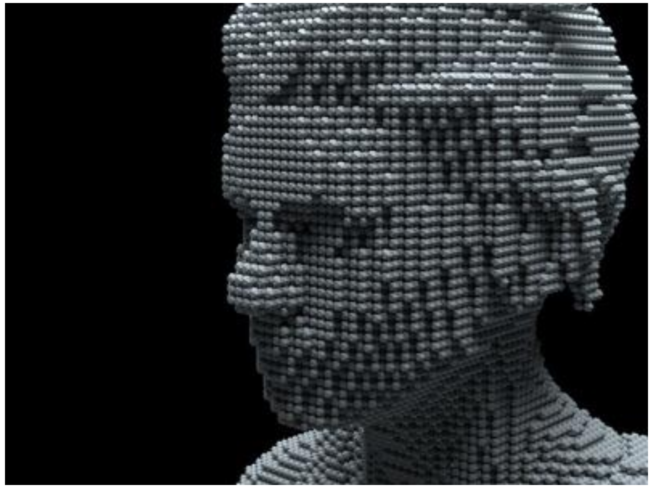
- Volumetric
- Non-parametric
- Topology agnostic
- Cubic order of memory



$$f(\mathbf{x}) = d(\mathbf{x}, \mathcal{S})$$

$$d(\mathbf{x}, \mathcal{S}) = \min_{\mathbf{p} \in \mathcal{S}} \|\mathbf{x} - \mathbf{p}\|$$

Distance Field

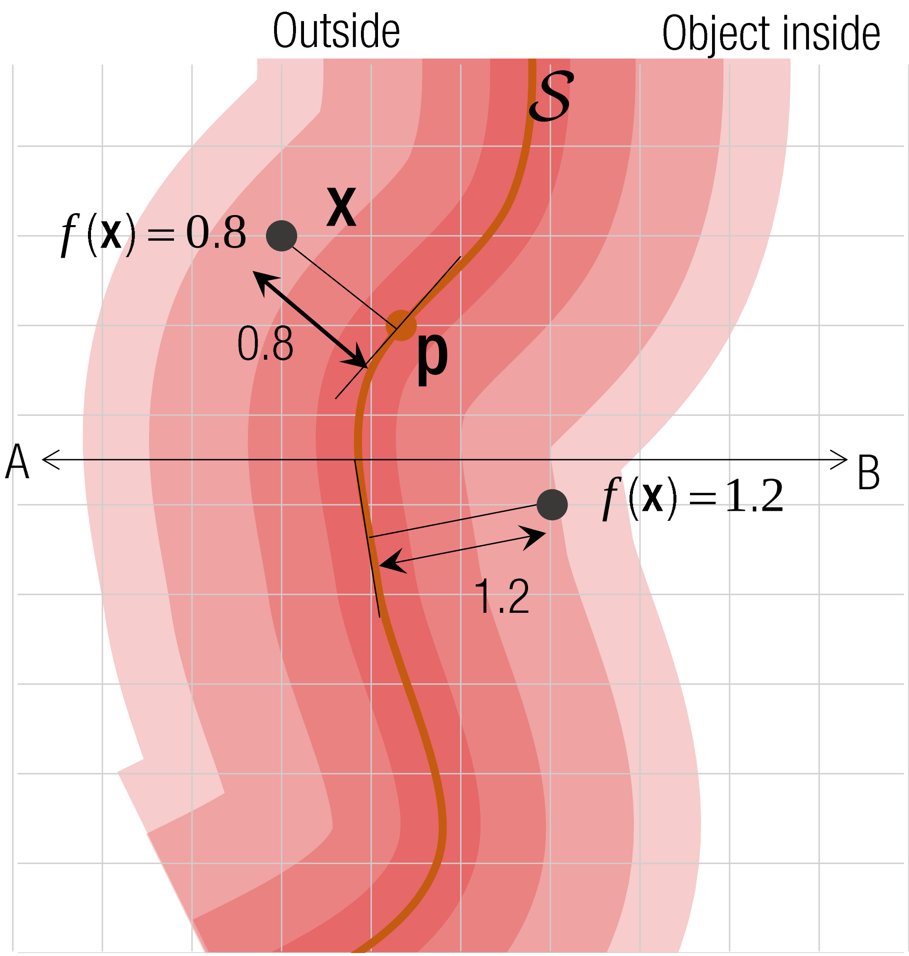


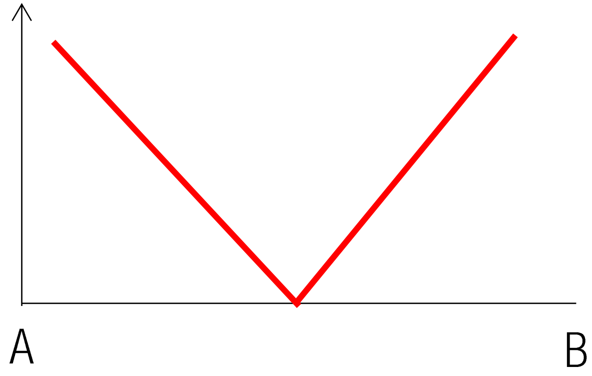
Voxel

$$V_{ijk} = \begin{cases} 1 & \text{if occupied} \\ 0 & \text{if empty} \end{cases}$$

- Volumetric
- Non-parametric
- Topology agnostic
- Cubic order of memory

3D surface

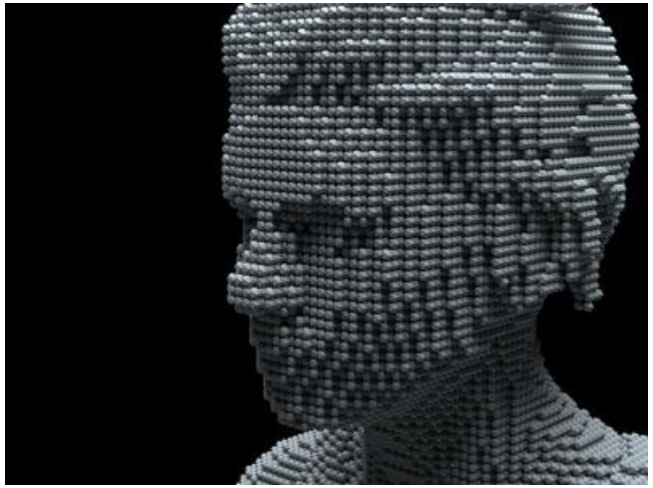




$$f(\mathbf{x}) = d(\mathbf{x}, \mathcal{S})$$

$$d(\mathbf{x}, \mathcal{S}) = \min_{\mathbf{p} \in \mathcal{S}} \|\mathbf{x} - \mathbf{p}\|$$

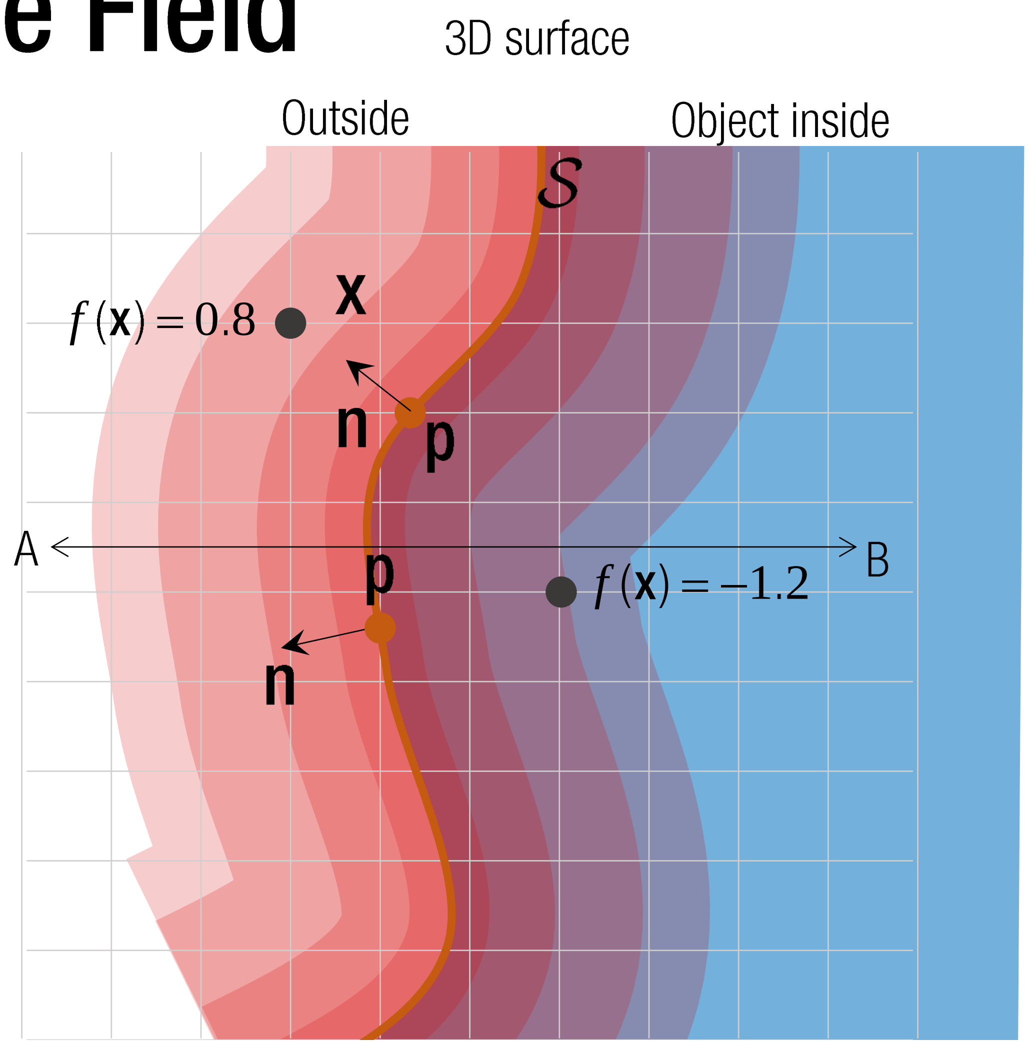
Signed Distance Field

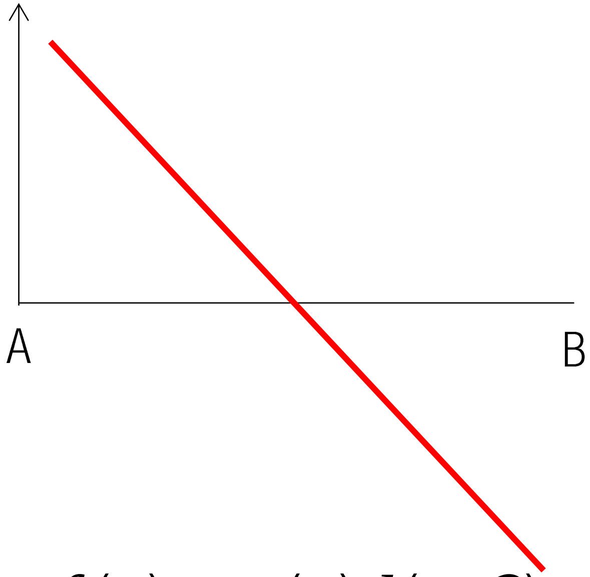


Voxel

$$V_{ijk} = \begin{cases} 1 & \text{if occupied} \\ 0 & \text{if empty} \end{cases}$$

- Volumetric
- Non-parametric
- Topology agnostic
- Cubic order of memory

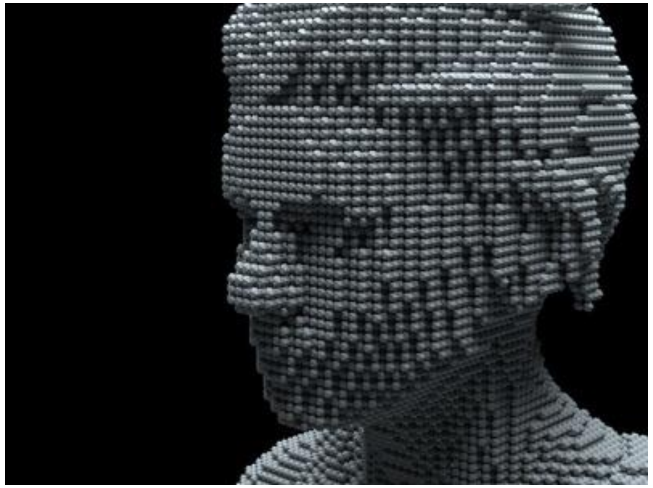




$$f(\mathbf{x}) = \sigma(\mathbf{x}) d(\mathbf{x}, \mathcal{S})$$

$$\sigma(\mathbf{x}) = \begin{cases} 1 & \text{if } \mathbf{n}^{\top}(\mathbf{x} - \mathbf{p}) \geq 0 \\ -1 & \text{otherwise} \end{cases}$$

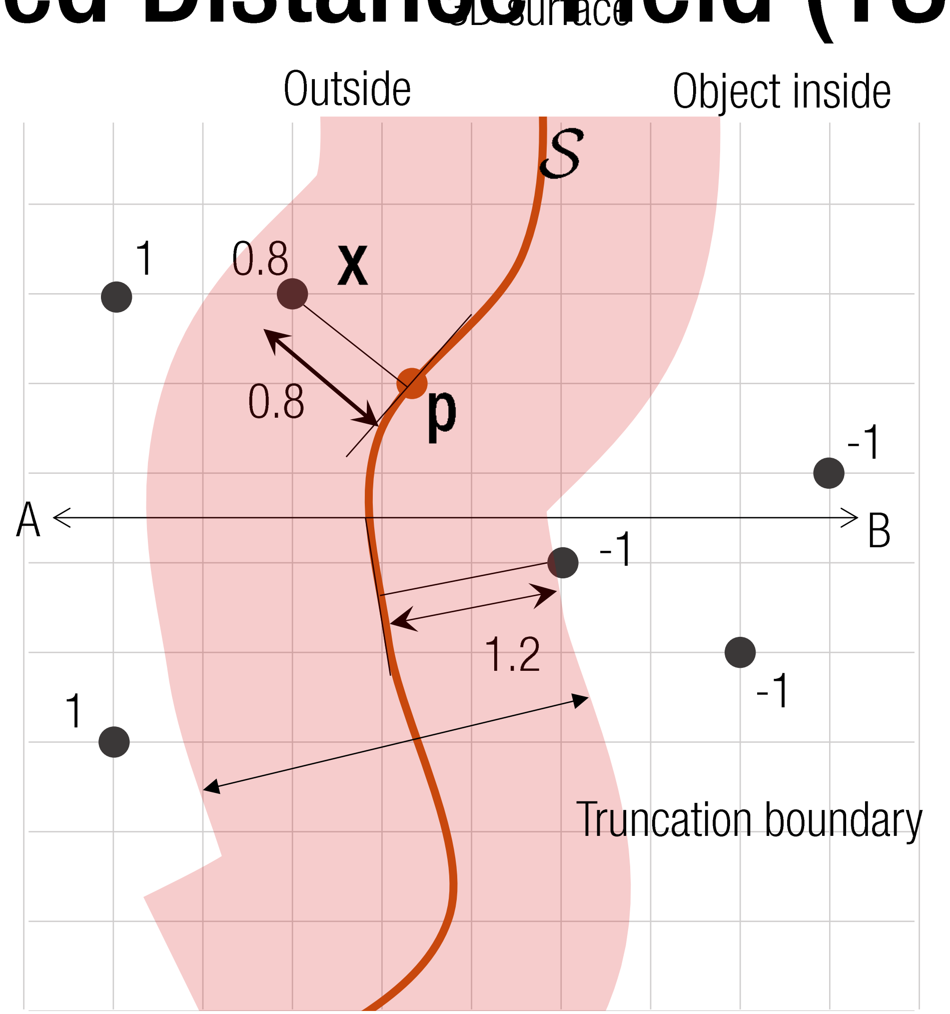
Truncated Signed Distance Field (TSDF)

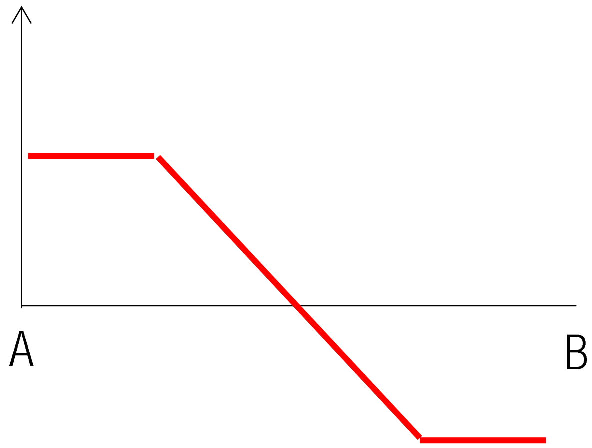


Voxel

$$V_{ijk} = \begin{cases} 1 & \text{if occupied} \\ 0 & \text{if empty} \end{cases}$$

- Volumetric
- Non-parametric
- Topology agnostic
- Cubic order of memory

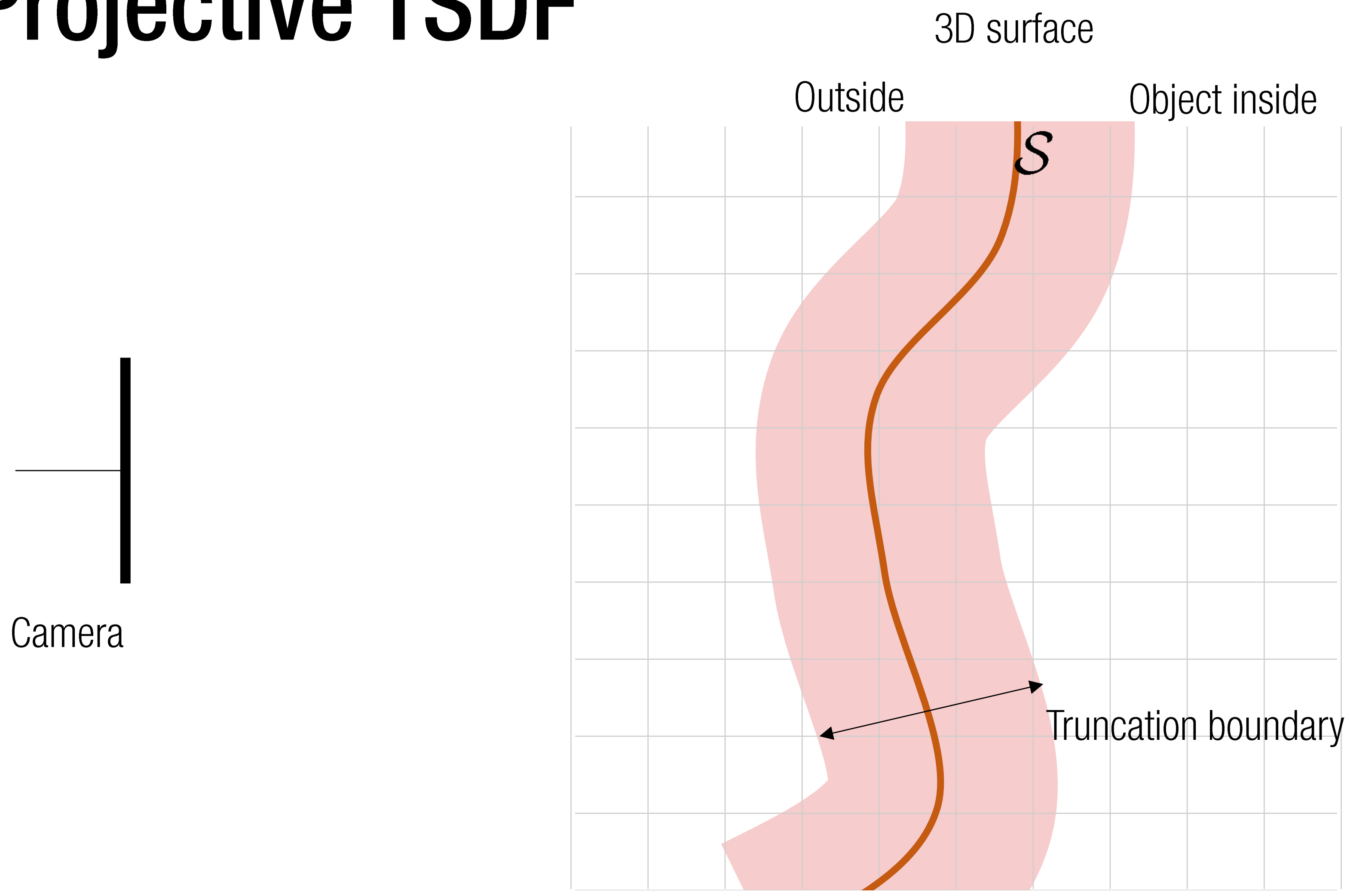




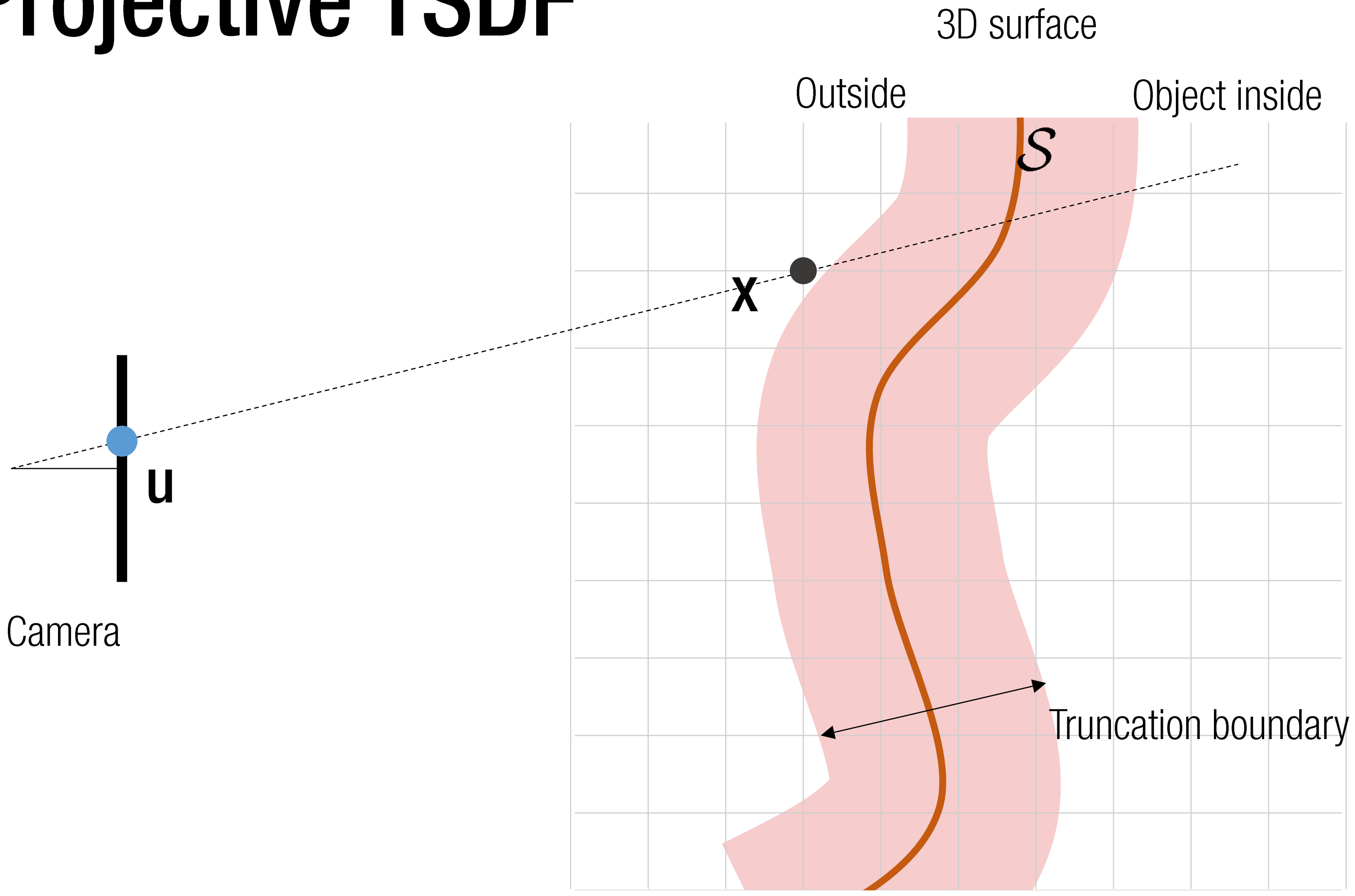
$$f(\mathbf{x}) = \begin{cases} \sigma(\mathbf{x}) d(\mathbf{x}, \mathcal{S}) & \text{if } d(\mathbf{x}, \mathcal{S}) \leq 1 \\ \sigma(\mathbf{x}) & \text{otherwise} \end{cases}$$

$$\sigma(\mathbf{x}) = \begin{cases} 1 & \text{if } \mathbf{n}^{\mathsf{T}}(\mathbf{x} - \mathbf{p}) \geq 0 \\ -1 & \text{otherwise} \end{cases}$$

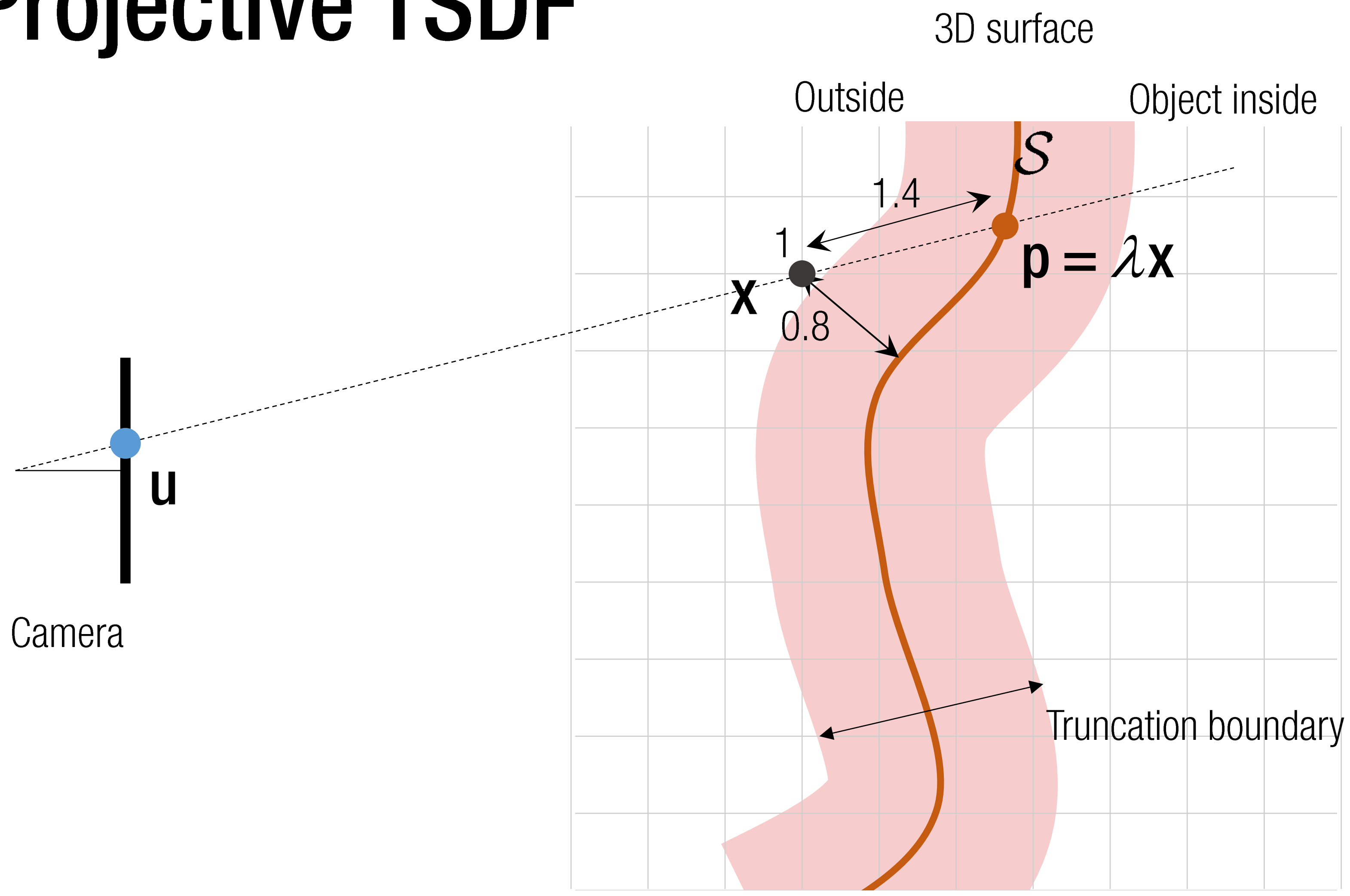
Projective TSDF



Projective TSDF



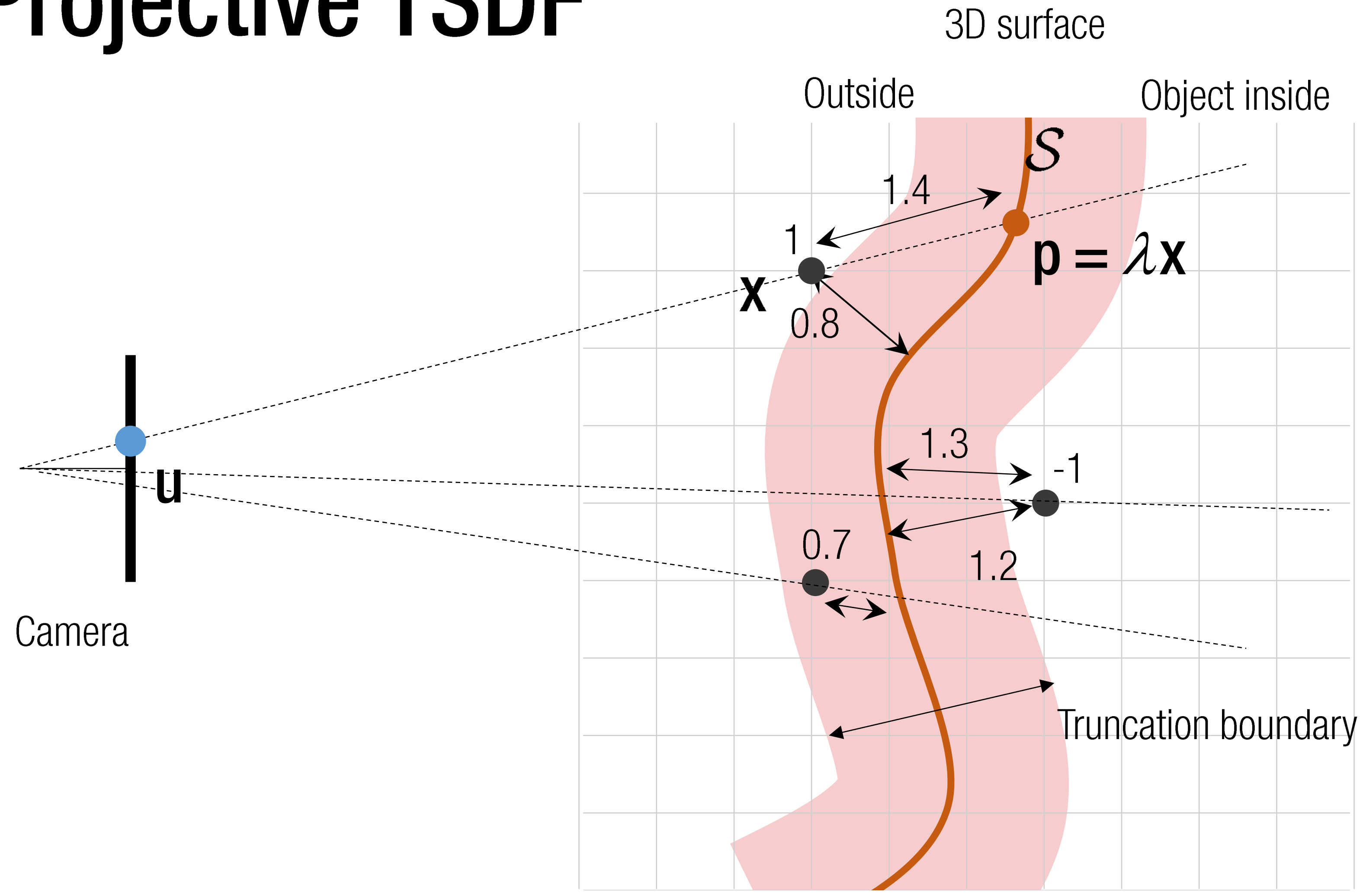
Projective TSDF



$$d(\mathbf{x}, \mathcal{S}) = \min_{\mathbf{p} \in \mathcal{S}} \|\mathbf{x} - \mathbf{p}\|$$

$$\text{s.t. } \mathbf{p} = \lambda \mathbf{K} \mathbf{u} = \lambda \mathbf{x}$$

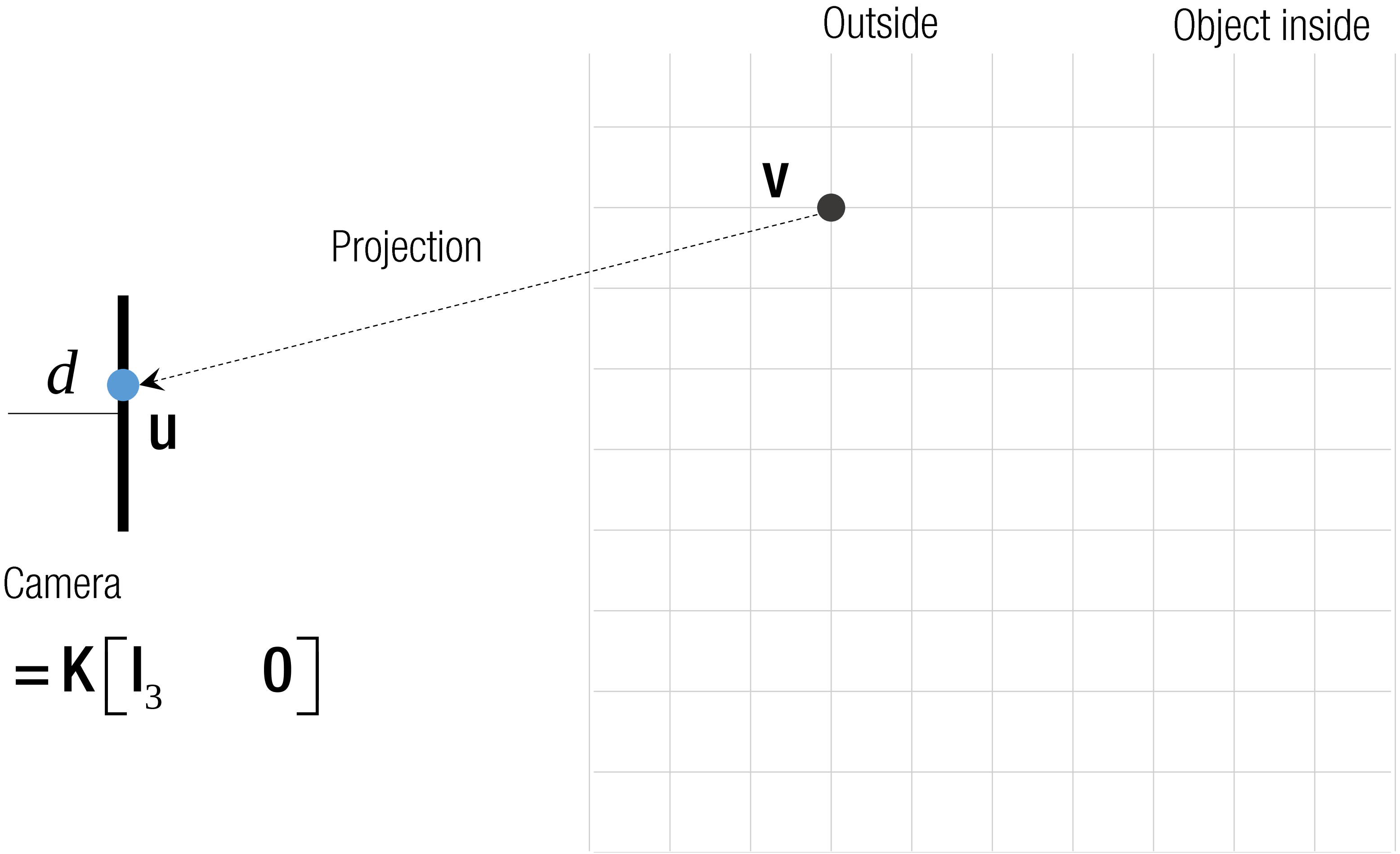
Projective TSDF



$$d(\mathbf{x}, \mathcal{S}) = \min_{\mathbf{p} \in \mathcal{S}} \|\mathbf{x} - \mathbf{p}\|$$

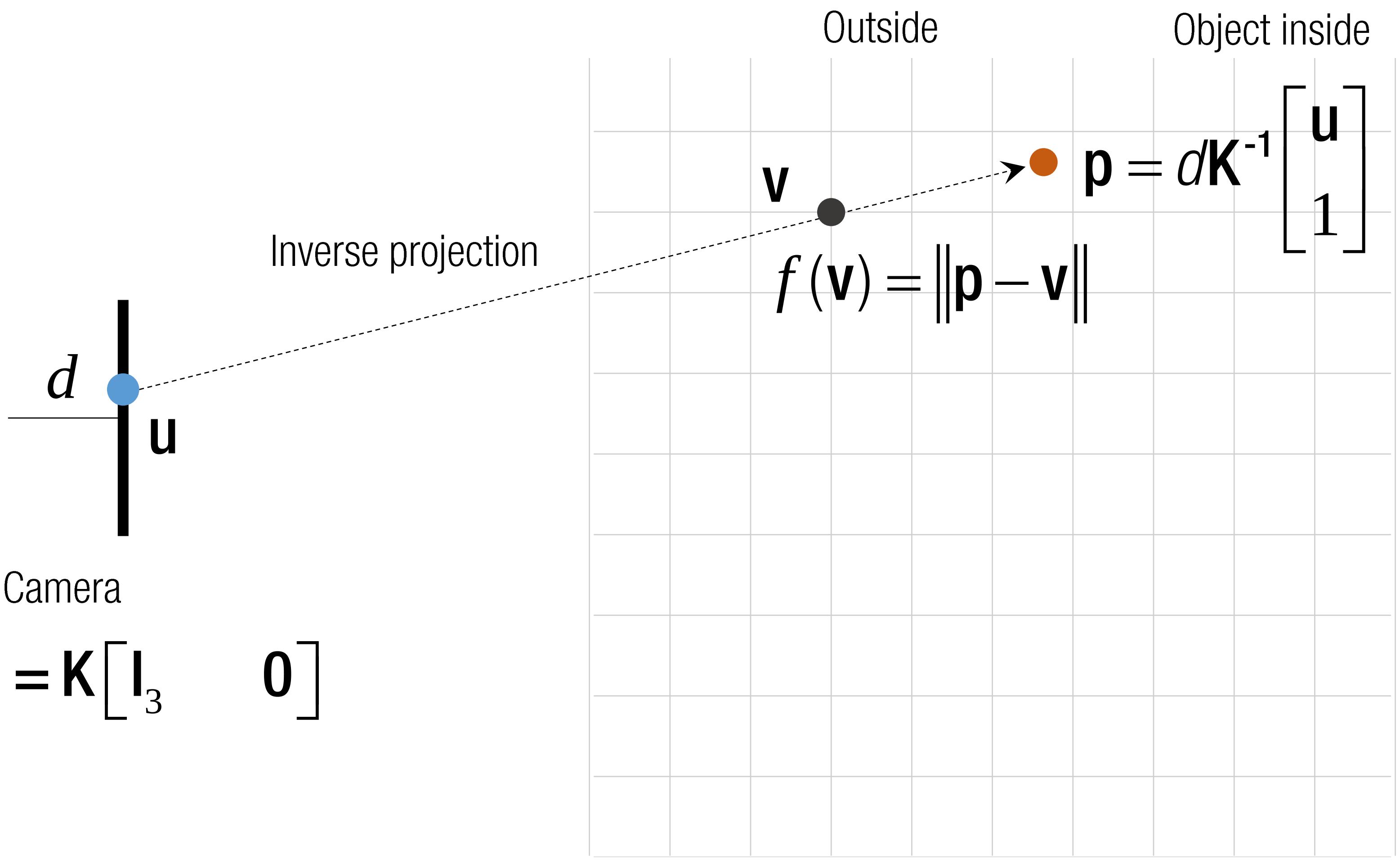
$$\text{s.t. } \mathbf{p} = \lambda \mathbf{K} \mathbf{u} = \lambda \mathbf{x}$$

Constructing TSDF from Depth



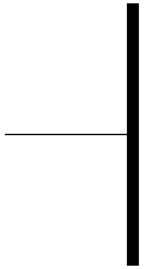
$$\mathbf{P} = \mathbf{K}\begin{bmatrix} \mathbf{I}_3 & \mathbf{0} \end{bmatrix}$$

Constructing TSDF from Depth



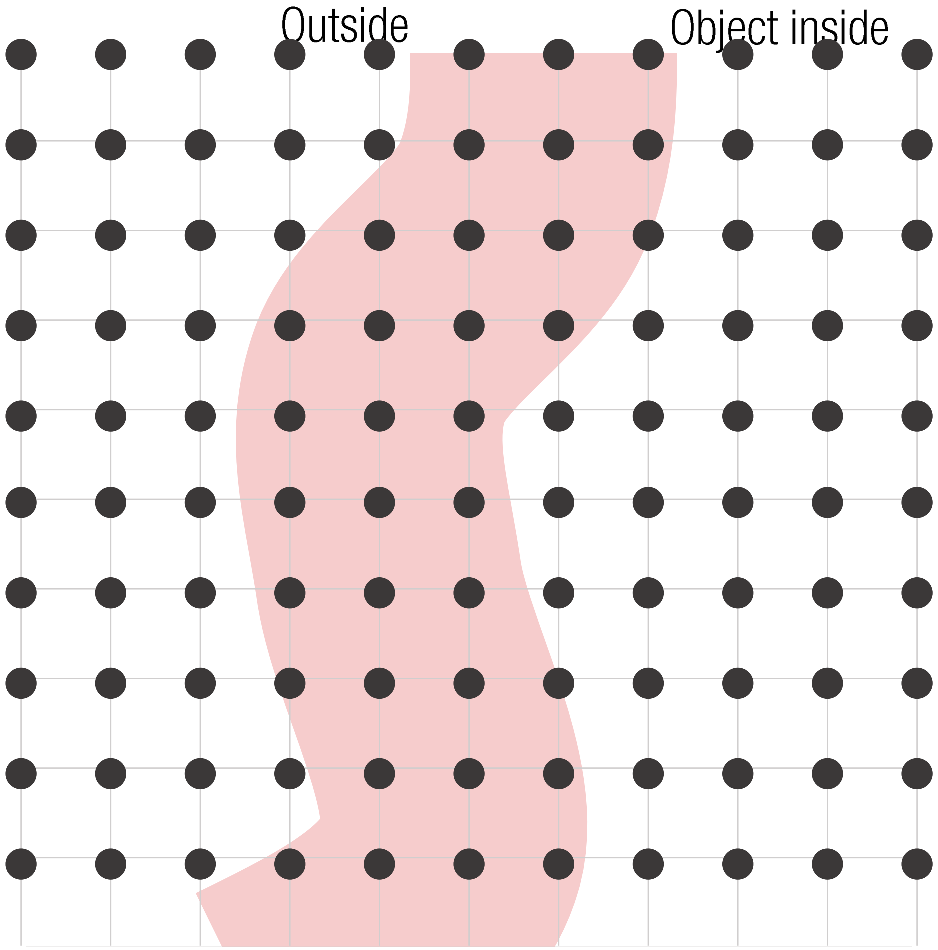
$$\mathbf{P} = \mathbf{K}\begin{bmatrix}\mathbf{I}_3 & \mathbf{0}\end{bmatrix}$$

Constructing TSDF from Depth



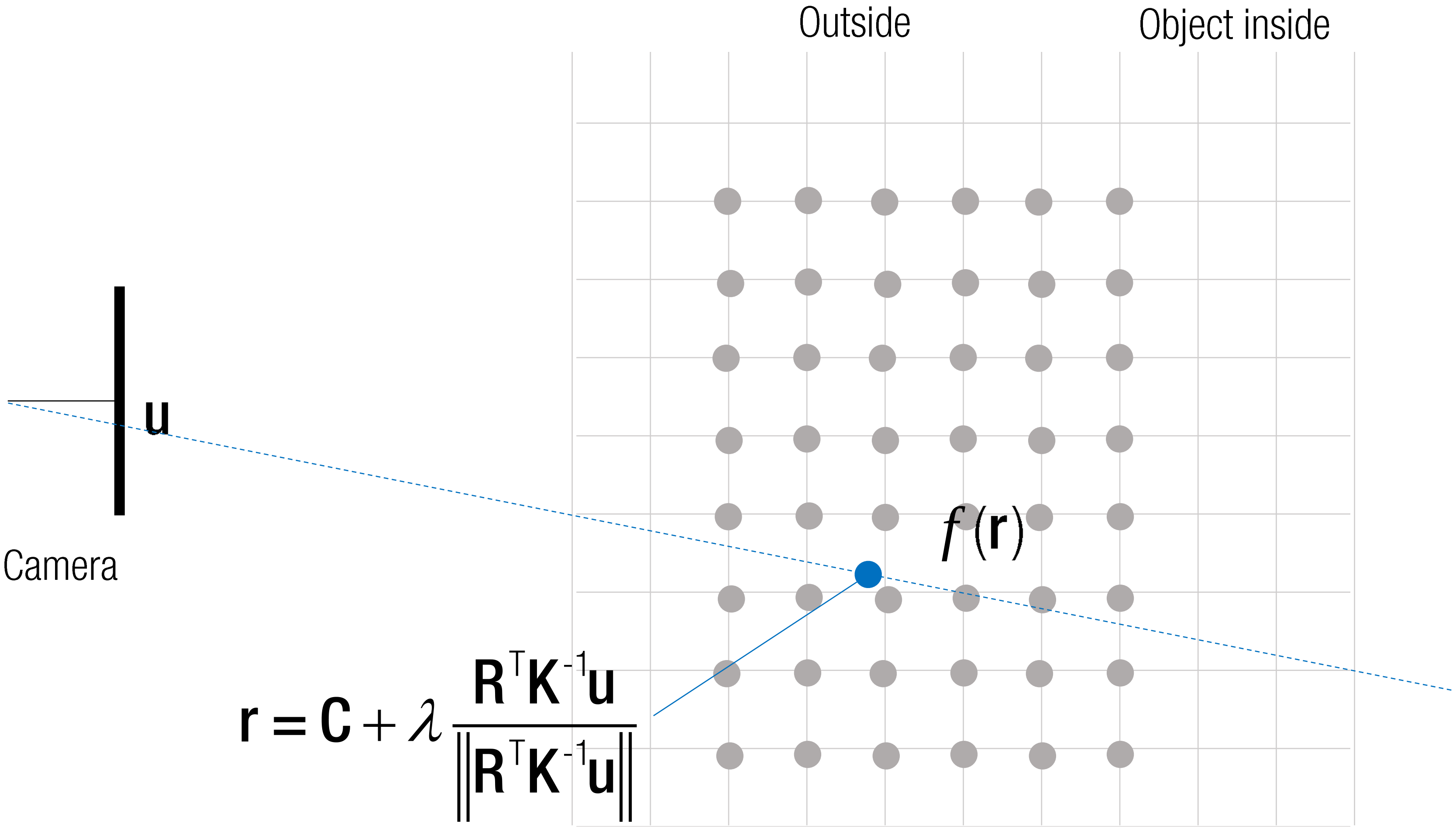
Camera

$$P = K\begin{bmatrix} I_3 & 0 \end{bmatrix}$$



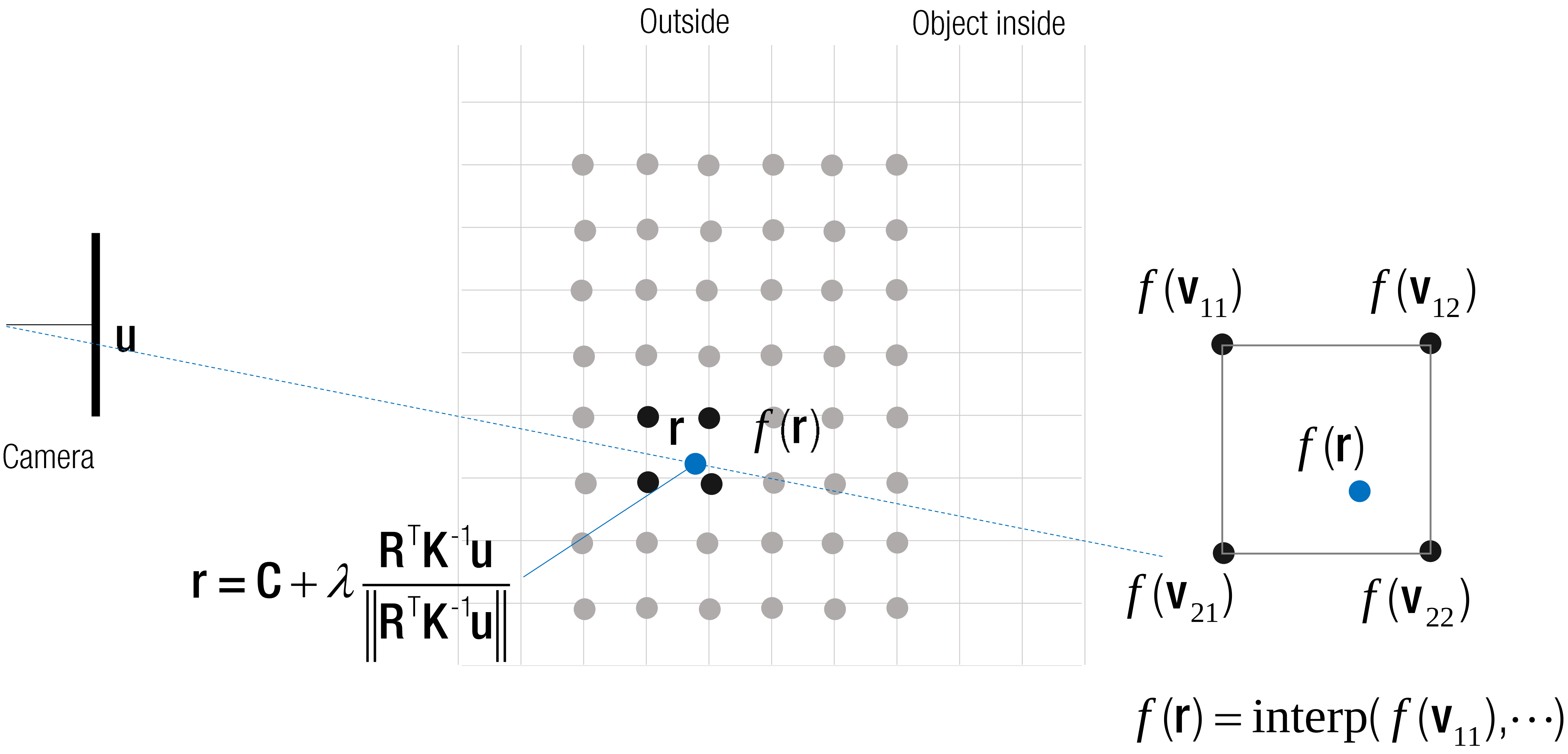
$$f(\mathbf{v})$$

Surface Reconstruction from TSDF

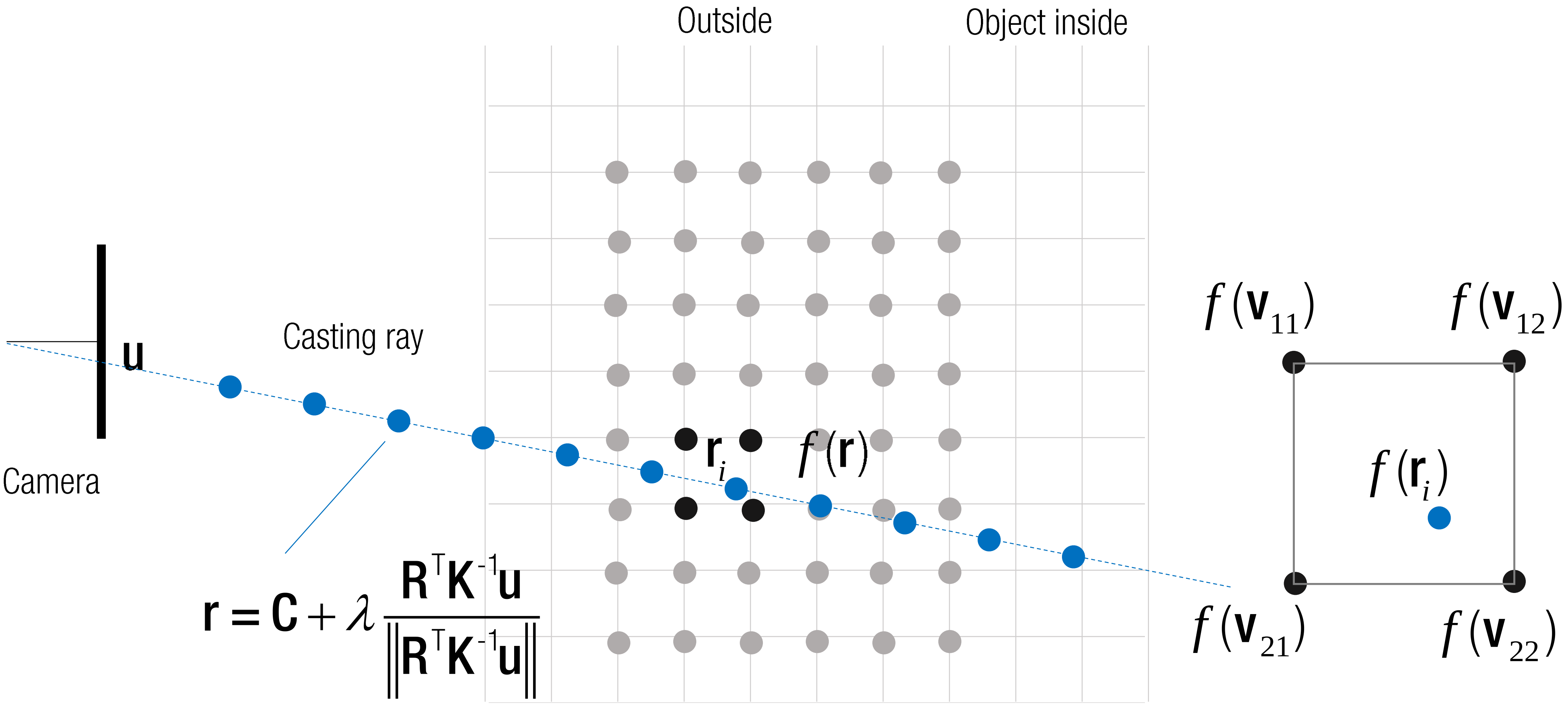


$$\mathbf{r} = \mathbf{C} + \lambda \frac{\mathbf{R}^{\mathrm{T}}\mathbf{K}^{-1}\mathbf{u}}{\left\|\mathbf{R}^{\mathrm{T}}\mathbf{K}^{-1}\mathbf{u}\right\|}$$

Surface Reconstruction from TSDF

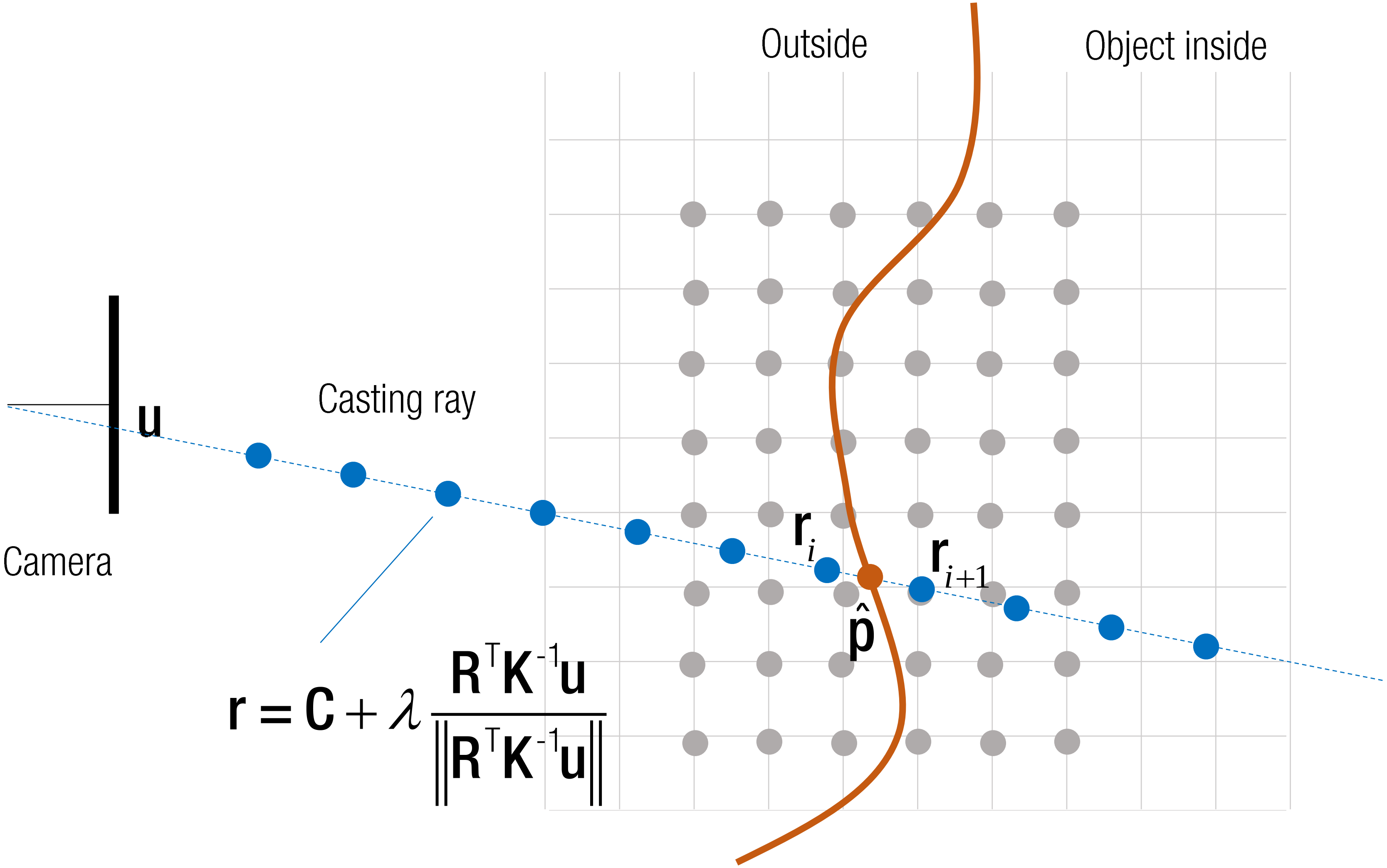


Surface Reconstruction from TSDF



$$f(\mathbf{r}_i) = \mathrm{interp}(f(\mathbf{v}_{11}), \cdots)$$

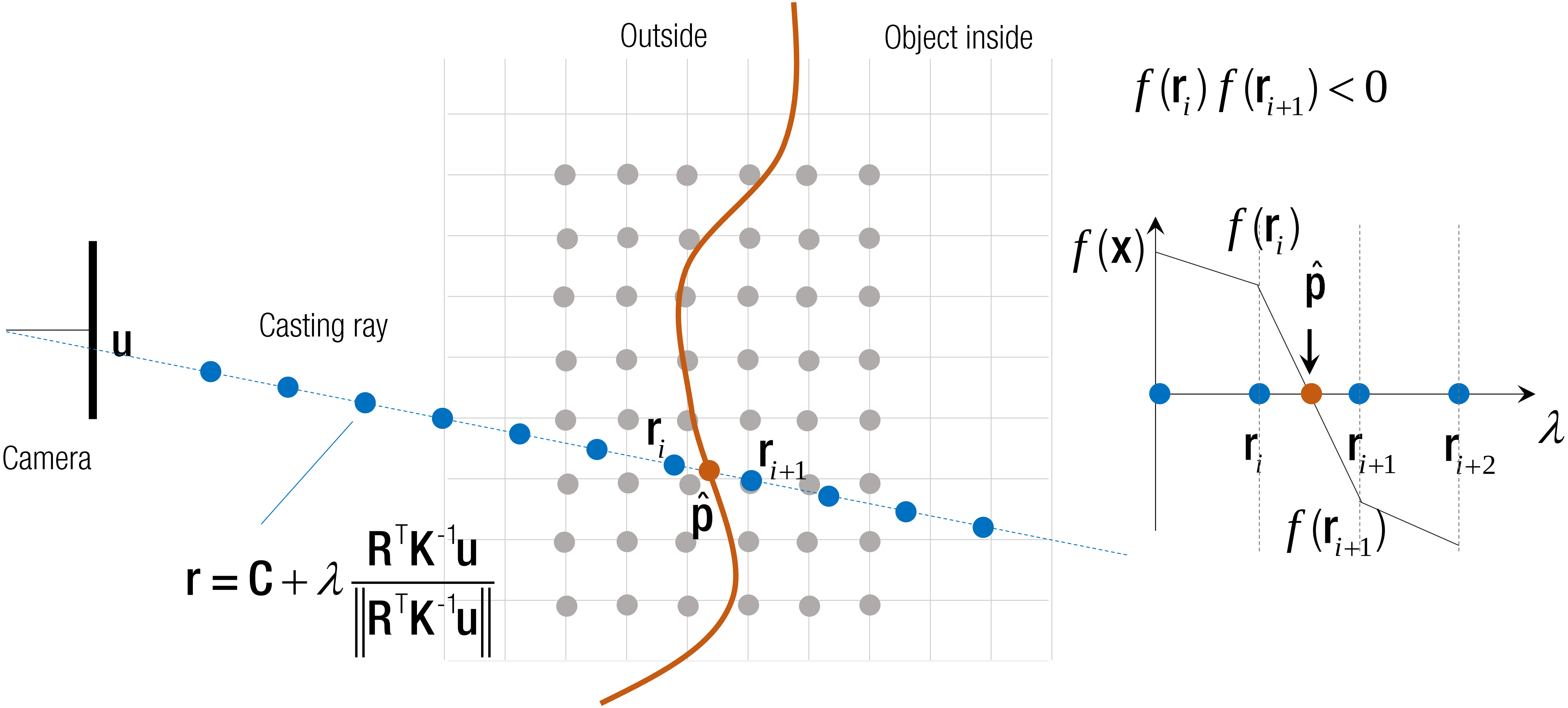
Surface Reconstruction from TSDF



$$f(\mathbf{r}_i)\, f(\mathbf{r}_{i+1}) < 0$$

$$\mathbf{r} = \mathbf{C} + \lambda \frac{\mathbf{R}^T\mathbf{K}^{-1}\mathbf{u}}{\left\|\mathbf{R}^T\mathbf{K}^{-1}\mathbf{u}\right\|}$$

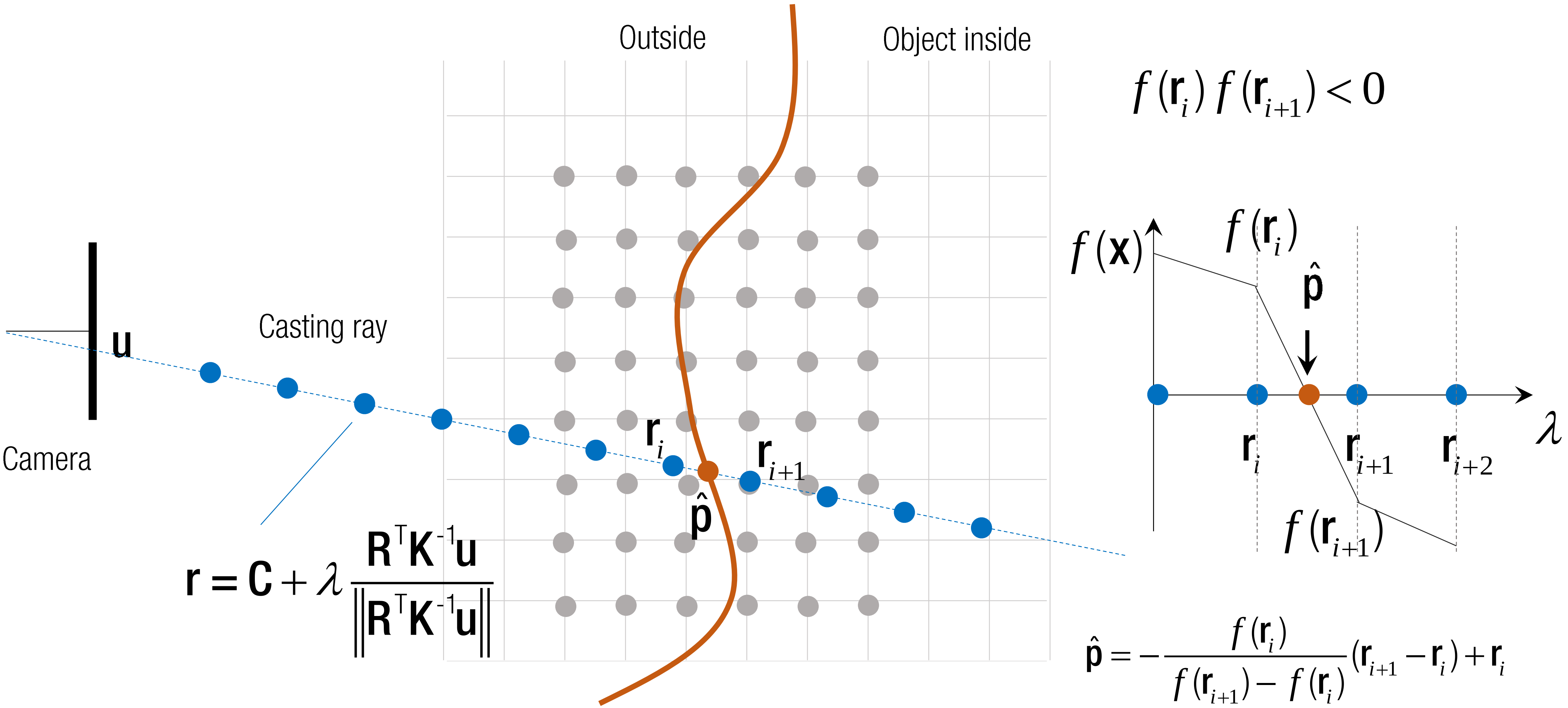
Surface Reconstruction from TSDF



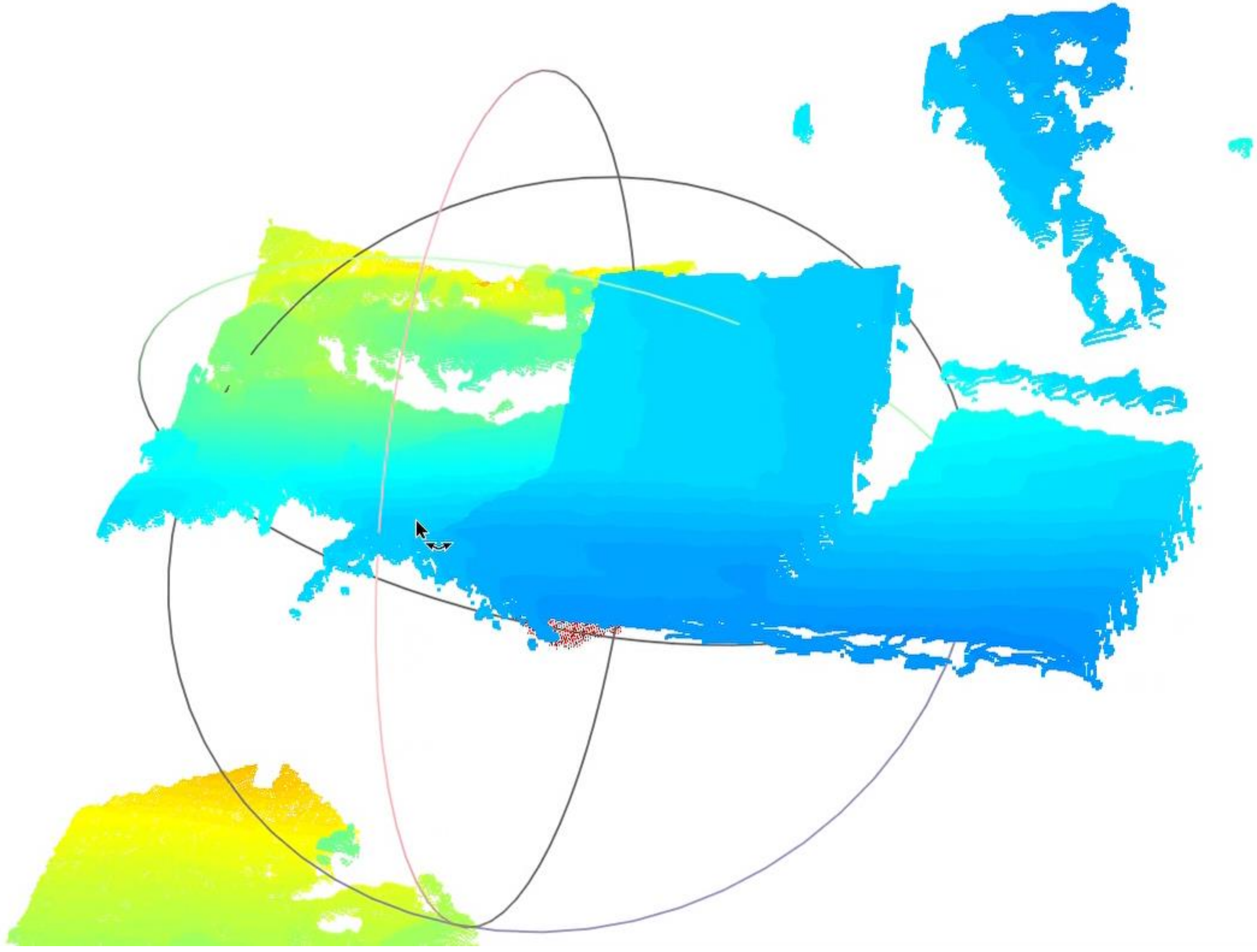
$$f(\mathbf{r}_i)\,f(\mathbf{r}_{i+1}) < 0$$

$$\mathbf{r} = \mathbf{C} + \lambda \frac{\mathbf{R}^T\mathbf{K}^{-1}\mathbf{u}}{\left\|\mathbf{R}^T\mathbf{K}^{-1}\mathbf{u}\right\|}$$

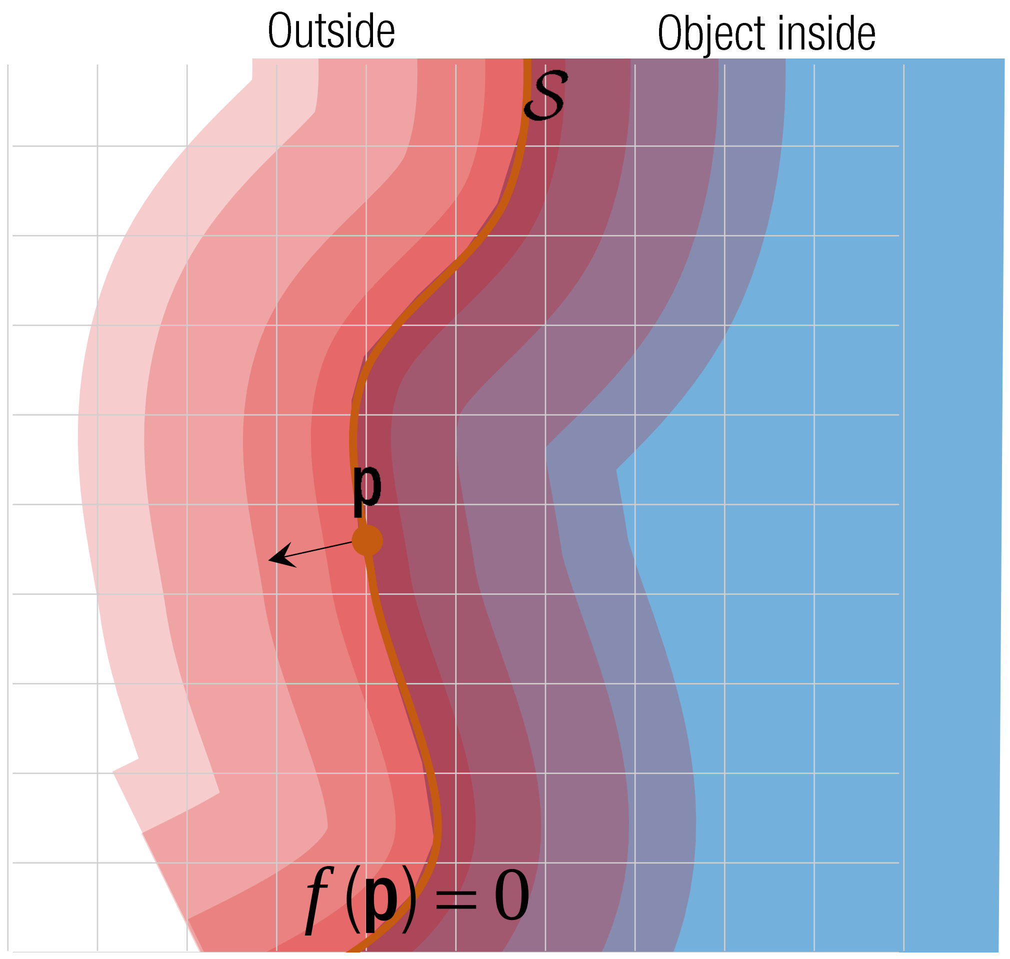
Surface Reconstruction from TSDF



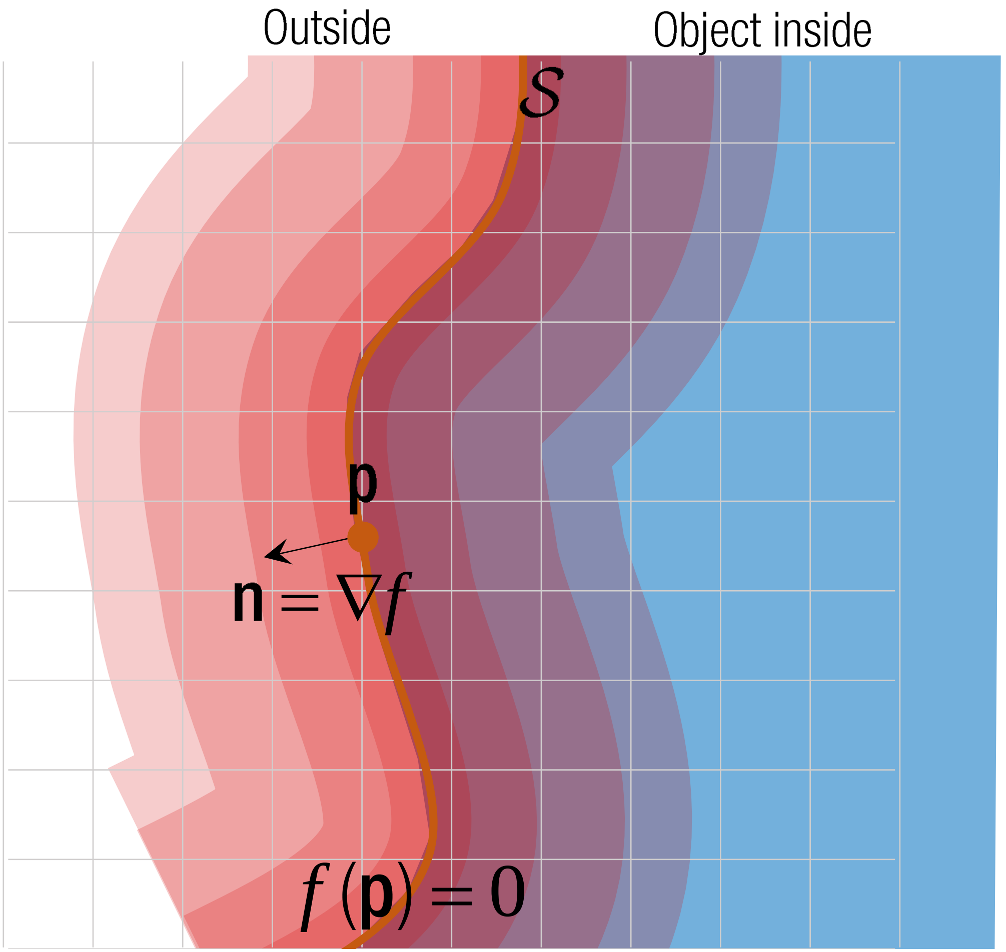
Surface Reconstruction from TSDF



Surface Normals from TSDF



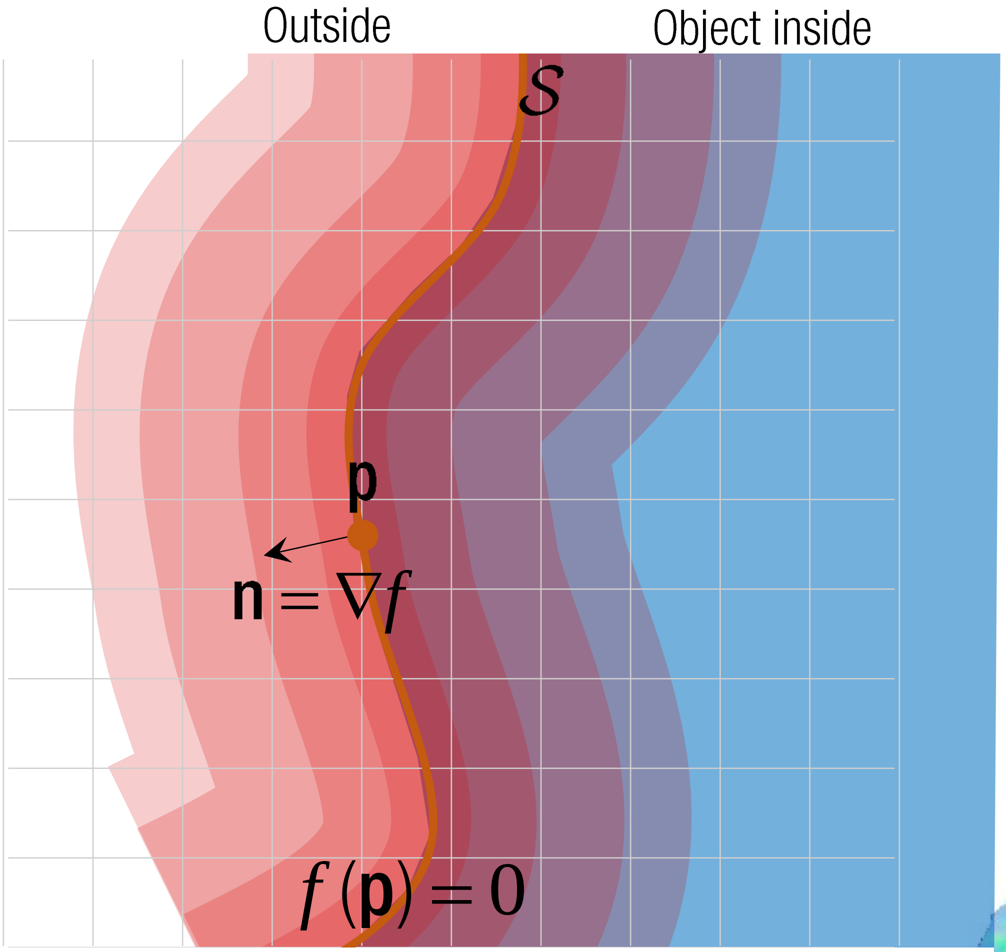
Surface Normals from TSDF

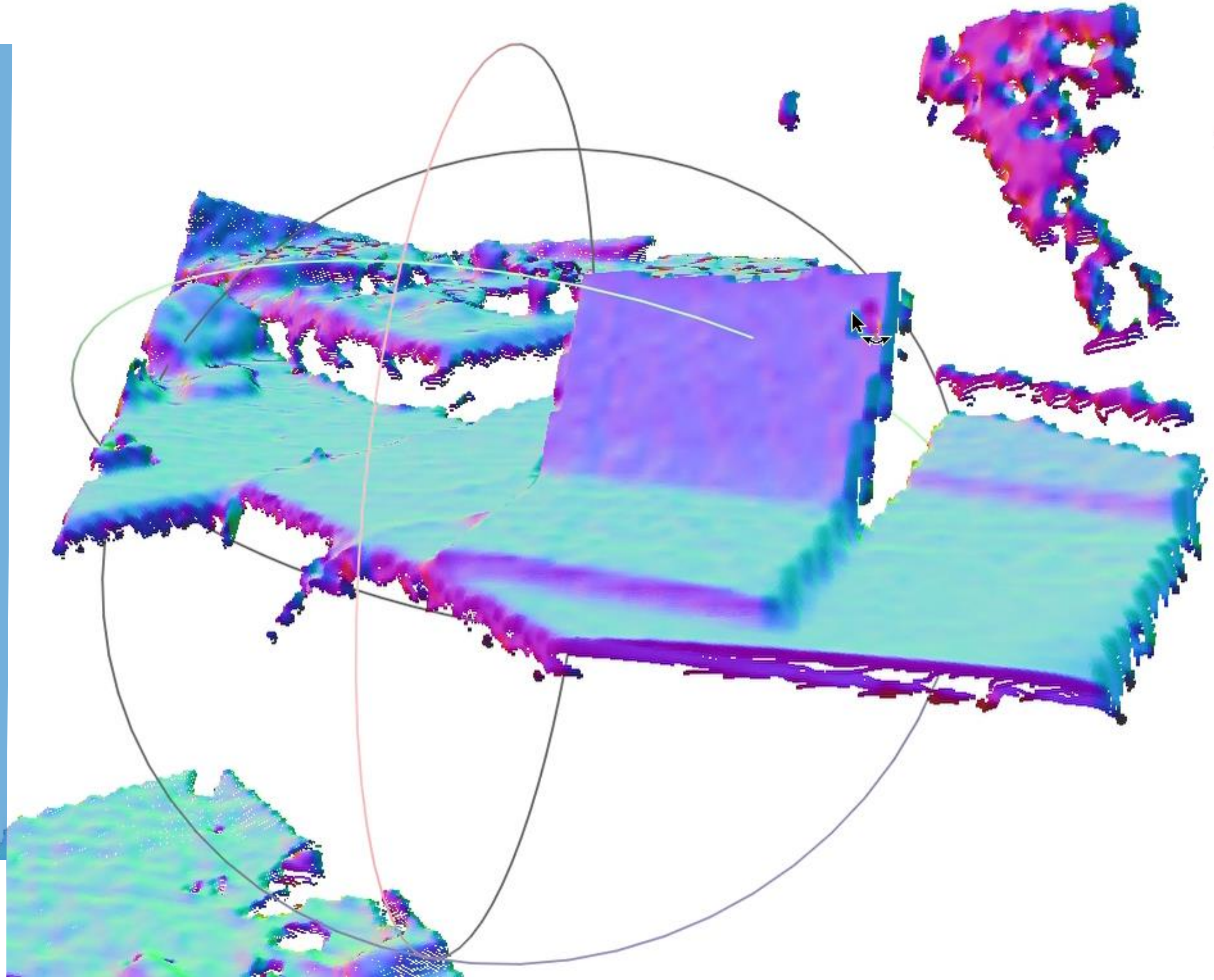


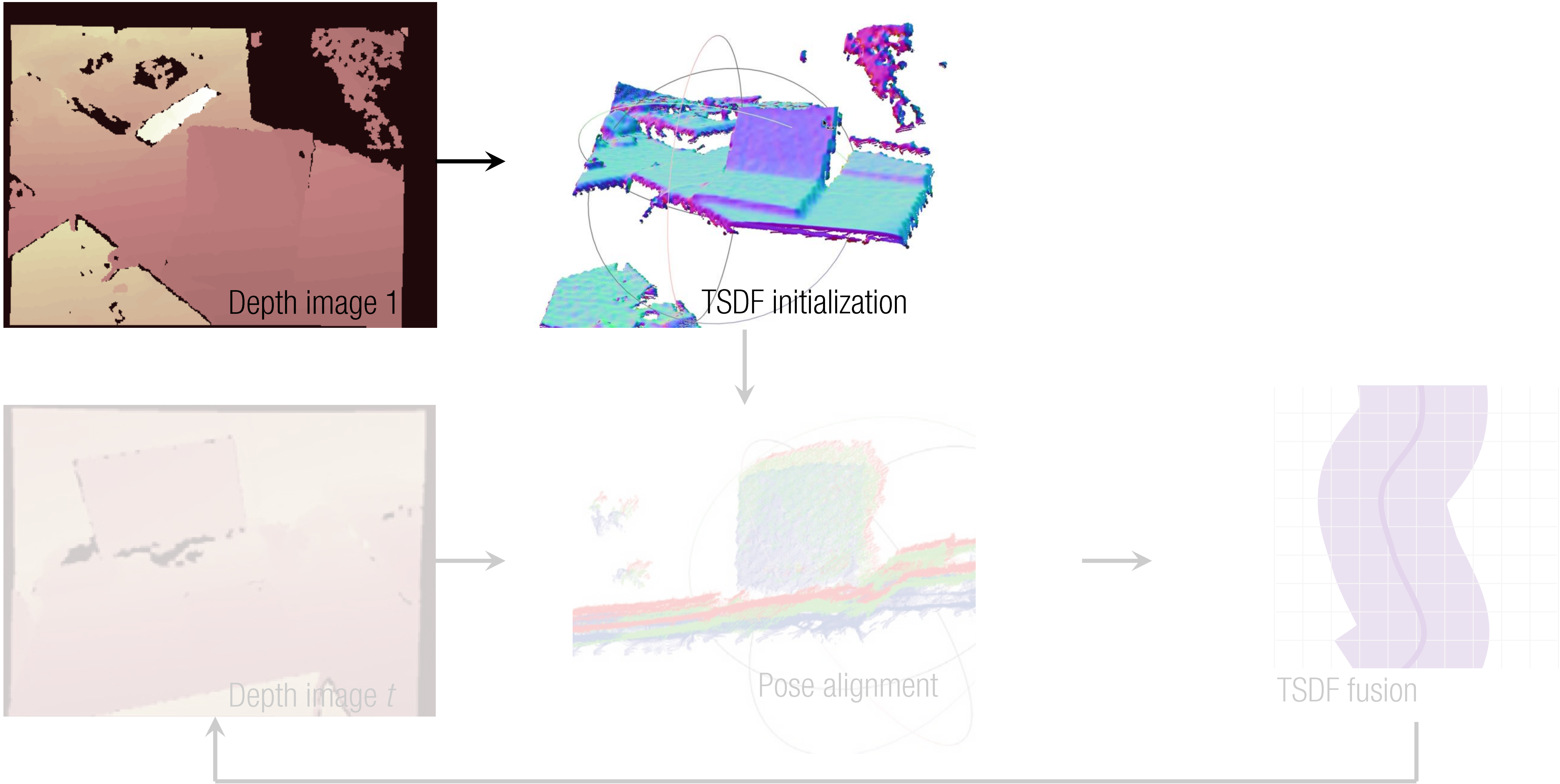
$$\mathbf{n} = \frac{\nabla f}{\|\nabla f\|}$$

$$= \frac{f(\mathbf{p} + \Delta\mathbf{p}) - f(\mathbf{p})}{\| f(\mathbf{p} + \Delta\mathbf{p}) - f(\mathbf{p}) \|}$$

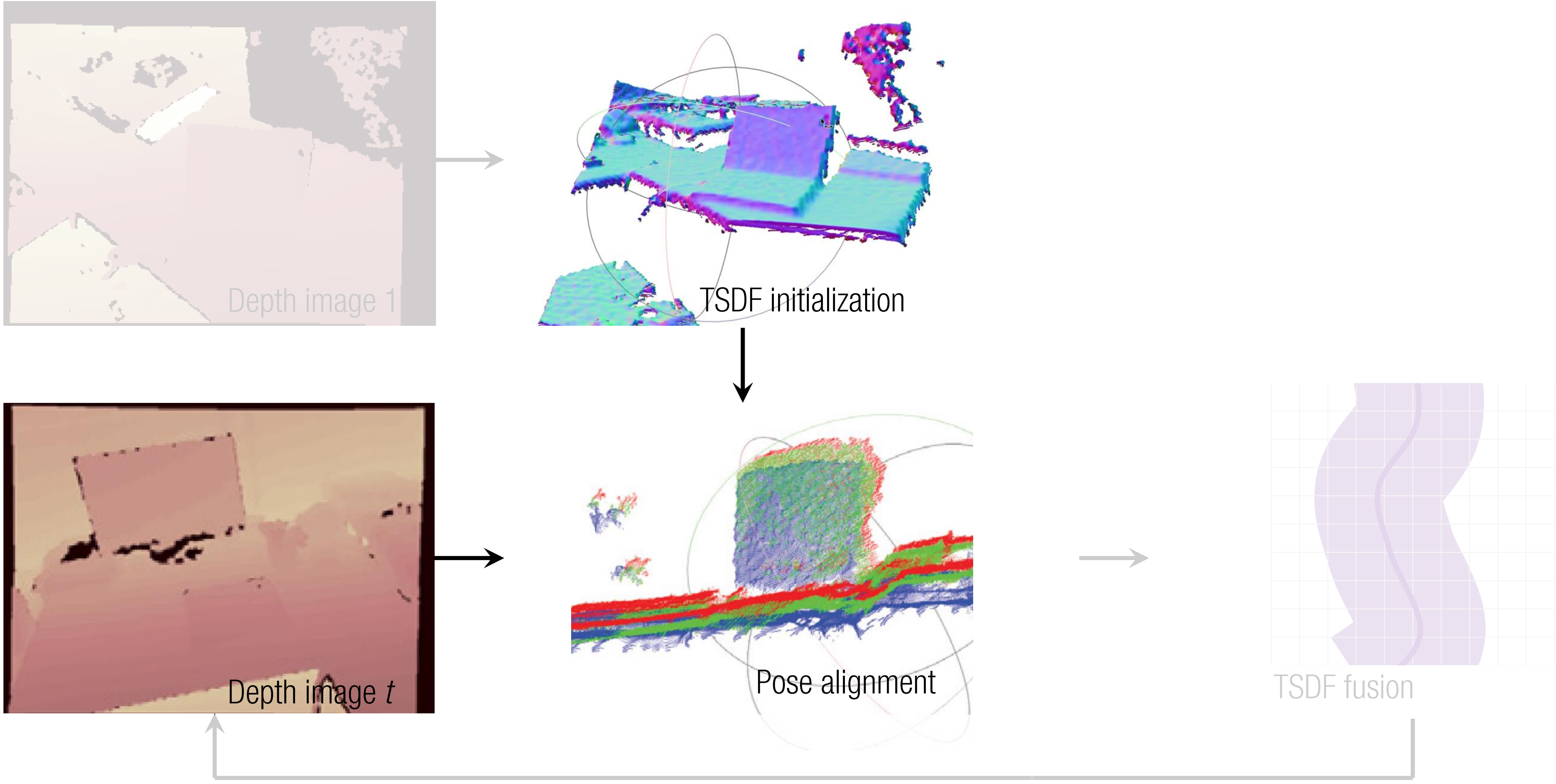
Surface Normals from TSDF





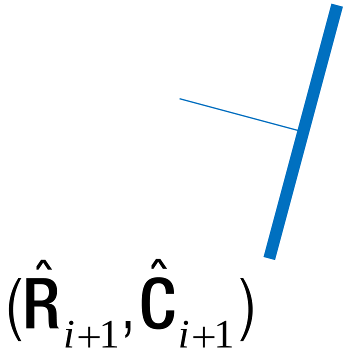


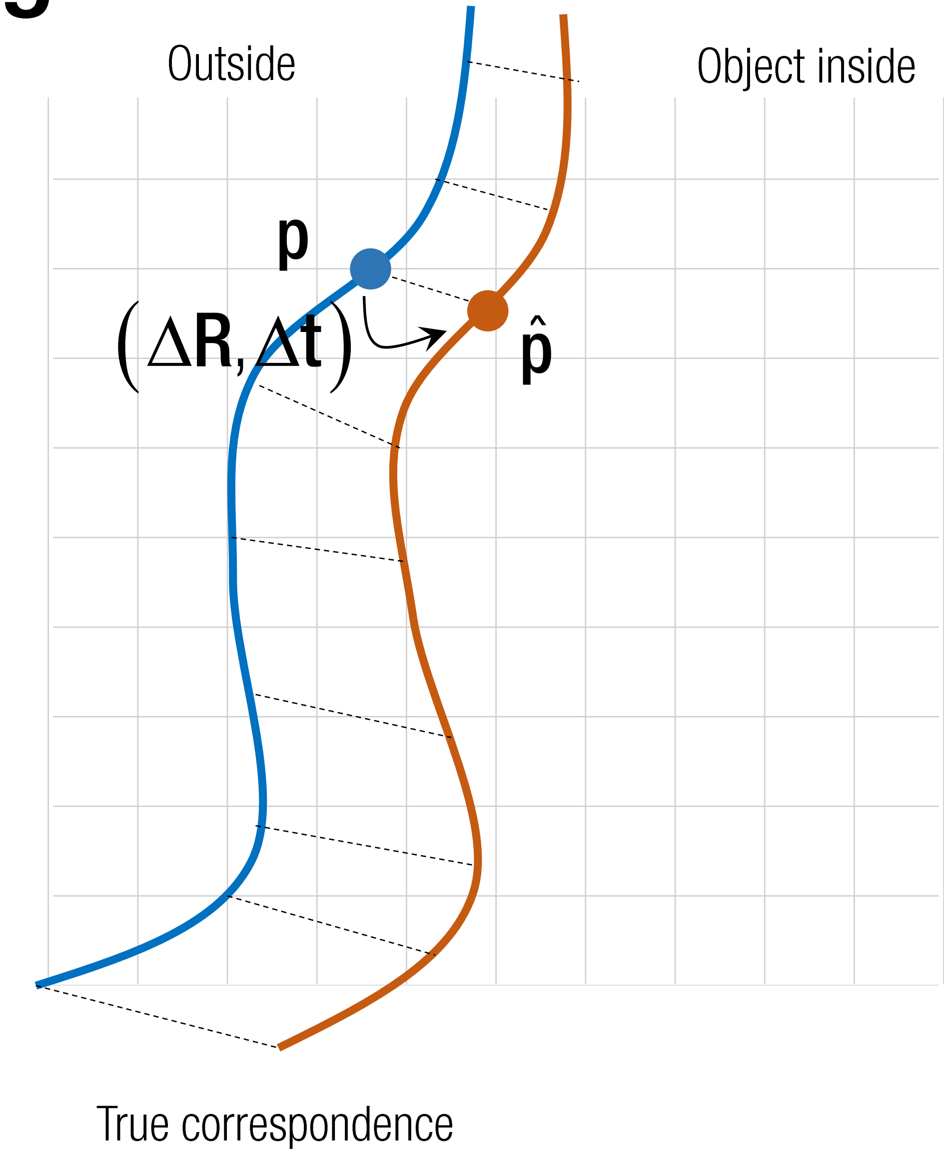
Depth image 1
TSDF initialization
Depth image *t*
Pose alignment
TSDF fusion



Depth image 1
TSDF initialization
Depth image t
Pose alignment
TSDF fusion

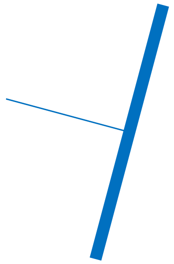
Point Cloud Alignment





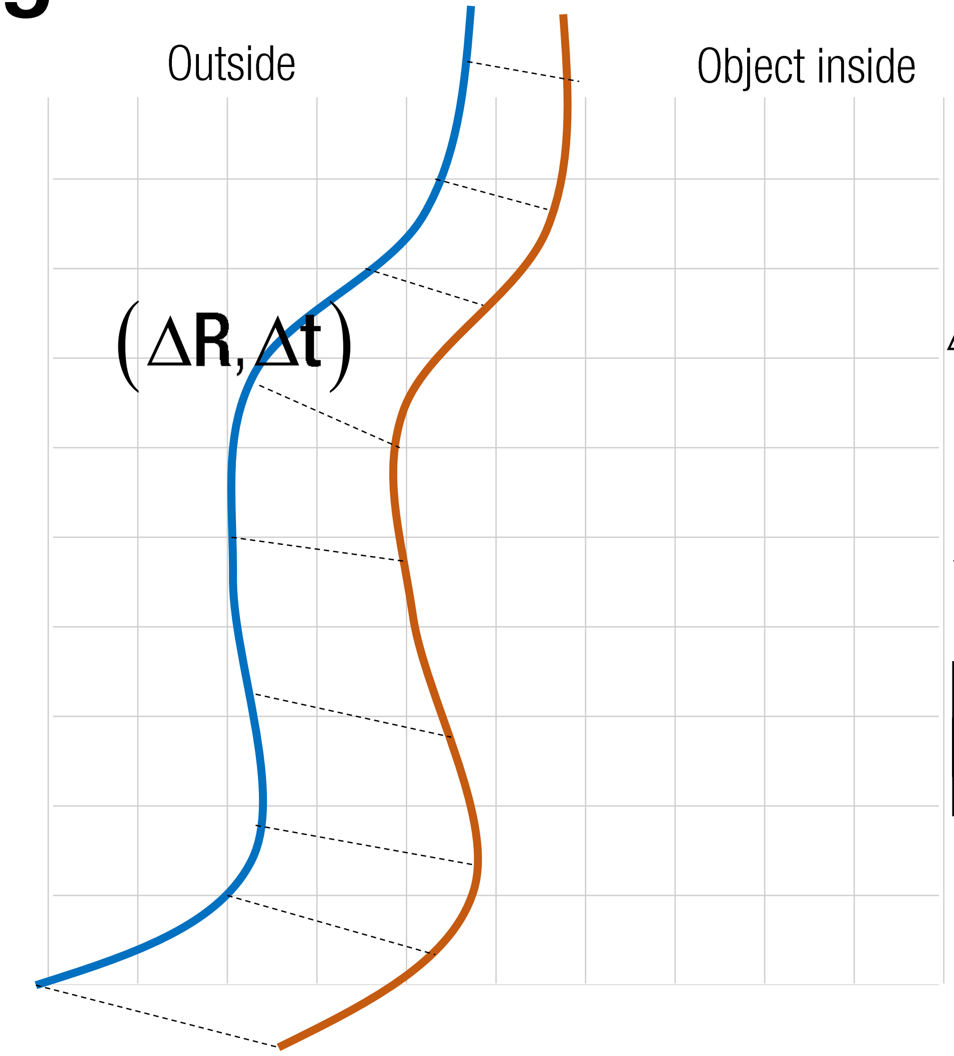
$$\Delta \mathbf{R}, \Delta \mathbf{t} = \underset{\Delta \mathbf{R}, \Delta \mathbf{t}}{\text{minimize}} \left\| \Delta \mathbf{R} \mathbf{p} + \Delta \mathbf{t} - \hat{\mathbf{p}} \right\|$$

Point Cloud Alignment



$$(\hat{\mathbf{R}}_{i+1}, \hat{\mathbf{C}}_{i+1})$$

$$(\mathbf{R}_{i+1}, \mathbf{C}_{i+1})$$

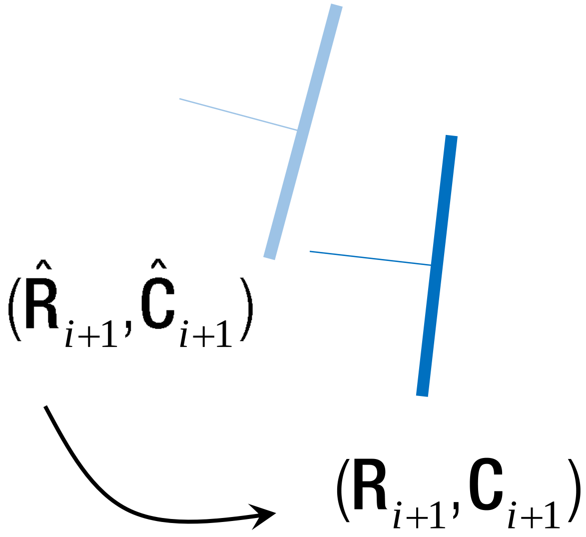


$$\Delta\mathbf{R}, \Delta\mathbf{t} = \underset{\Delta\mathbf{R}, \Delta\mathbf{t}}{\text{minimize}} \left\| \Delta\mathbf{R}\mathbf{p} + \Delta\mathbf{t} - \hat{\mathbf{p}} \right\|$$

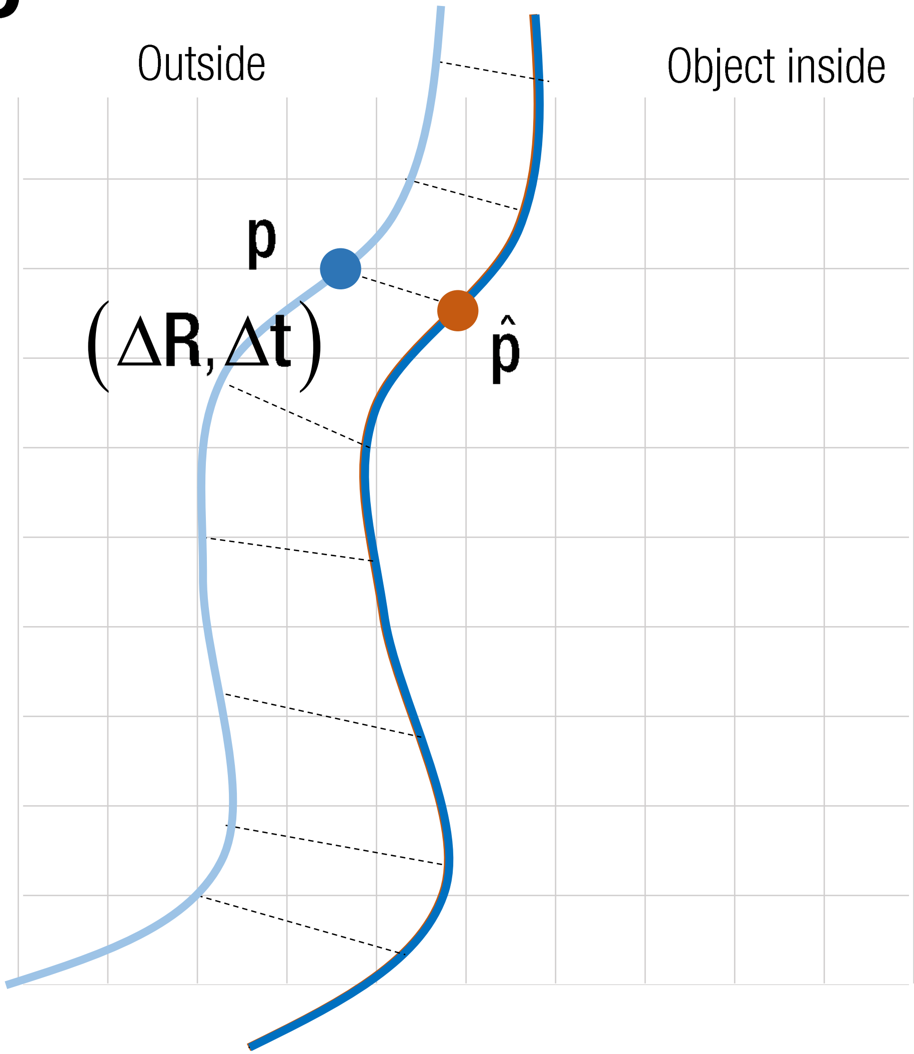
$${}^{w}\mathbf{T}_i = {}^{w}\mathbf{T}_{w_0}\, {}^{w_0}\mathbf{T}_i = \Delta\mathbf{T}_i\, {}^{w_0}\mathbf{T}_i$$

$$\begin{bmatrix} \mathbf{R} & \mathbf{t} \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} \Delta\mathbf{R} & \Delta\mathbf{t} \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \mathbf{R}_i^{\top} & \mathbf{C}_i \\ 0 & 1 \end{bmatrix}$$

Point Cloud Alignment



$$\Delta T = (\Delta R, \Delta t)$$



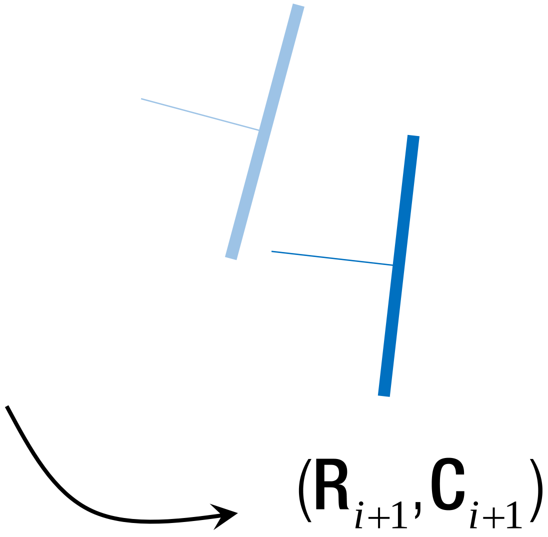
True correspondence

$$\Delta R, \Delta t = \underset{\Delta R, \Delta t}{\text{minimize}} \left\| \Delta R p + \Delta t - \hat{p} \right\|$$

$$^{w}T_i = {}^{w}T_{w_0}\,{}^{w_0}T_i = \Delta T_i\,{}^{w_0}T_i$$

$$\begin{bmatrix} R & t \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} \Delta R & \Delta t \\ 0 & 1 \end{bmatrix} \begin{bmatrix} R_i^{T} & C_i \\ 0 & 1 \end{bmatrix}$$

Point Cloud Alignment



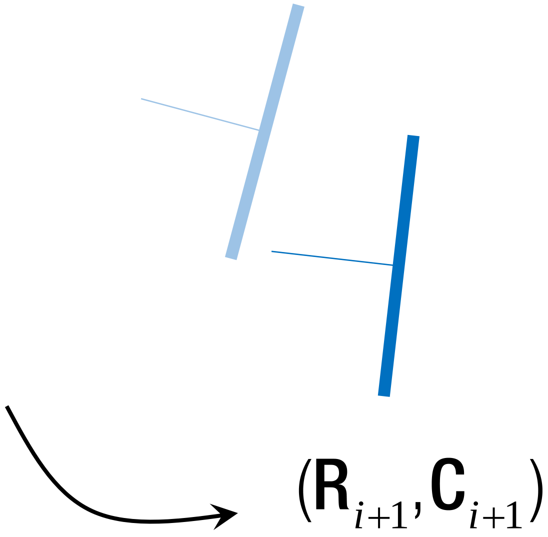
$$\Delta\mathbf{T} = \left(\Delta\mathbf{R}, \Delta\mathbf{t}\right)$$

Lie Algebra of Euclidean transformation:

$$\Delta\mathbf{T}_i = \begin{bmatrix} \Delta\mathbf{R} & \Delta\mathbf{t} \\ \mathbf{0} & 1 \end{bmatrix} = \mathbf{I} + \mathbf{Z} + \frac{\mathbf{Z}^2}{2!} + \frac{\mathbf{Z}^3}{3!} + \cdots$$

where $$\mathbf{Z}(\beta, \gamma, \alpha, t_x, t_y, t_z) = \begin{bmatrix} 0 & \alpha & -\gamma & t_x \\ -\alpha & 0 & \beta & t_x \\ \gamma & -\beta & 0 & t_x \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

Point Cloud Alignment



$$\Delta\mathrm{T} = (\Delta\mathrm{R}, \Delta\mathrm{t})$$

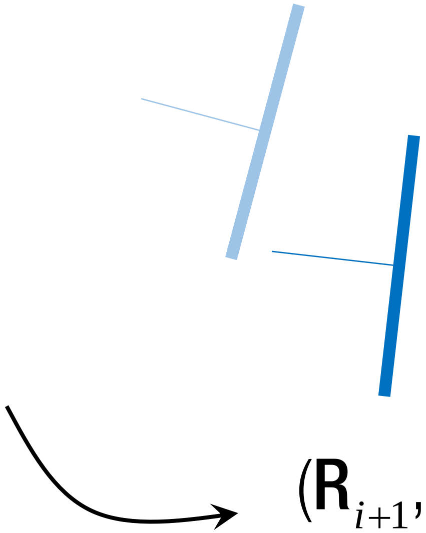
First order approximation:

$$\Delta\mathrm{T}_i = \begin{bmatrix} \Delta\mathrm{R} & \Delta\mathrm{t} \\ \mathbf{0} & 1 \end{bmatrix} \approx \mathrm{I} + \mathrm{Z}$$

where $\mathbf{Z}(\beta,\gamma,\alpha,t_x,t_y,t_z) = \begin{bmatrix} 0 & \alpha & -\gamma & t_x \\ -\alpha & 0 & \beta & t_x \\ \gamma & -\beta & 0 & t_x \\ 0 & 0 & 0 & 0 \end{bmatrix}$

$$= \begin{bmatrix} [\xi]_\times & \Delta\mathrm{t} \\ \mathbf{0} & 0 \end{bmatrix}$$

Point Cloud Alignment



$(\mathbf{R}_{i+1}, \mathbf{C}_{i+1})$

$$\Delta\mathbf{T} = \left(\Delta\mathbf{R}, \Delta\mathbf{t}\right)$$

First order approximation:

$$\Delta\mathbf{T}_i = \begin{bmatrix} \Delta\mathbf{R} & \Delta\mathbf{t} \\ \mathbf{0} & 1 \end{bmatrix} \approx \mathbf{I} + \mathbf{Z}$$

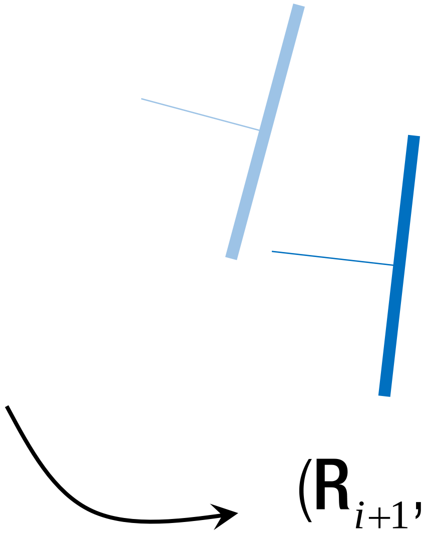
where

$$\mathbf{Z}(\beta, \gamma, \alpha, t_x, t_y, t_z) = \begin{bmatrix} 0 & \alpha & -\gamma & t_x \\ -\alpha & 0 & \beta & t_x \\ \gamma & -\beta & 0 & t_x \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

$$= \begin{bmatrix} [\xi]_\times & \Delta\mathbf{t} \\ \mathbf{0} & 0 \end{bmatrix}$$

$$\Delta\mathbf{R}, \Delta\mathbf{t} = \underset{\Delta\mathbf{R}, \Delta\mathbf{t}}{\text{minimize}} \left\| \Delta\mathbf{R}\mathbf{p} + \Delta\mathbf{t} - \hat{\mathbf{p}} \right\|$$

Point Cloud Alignment



$$(\mathbf{R}_{i+1}, \mathbf{C}_{i+1})$$

$$\Delta \mathbf{T} = \left(\Delta \mathbf{R}, \Delta \mathbf{t} \right)$$

First order approximation:

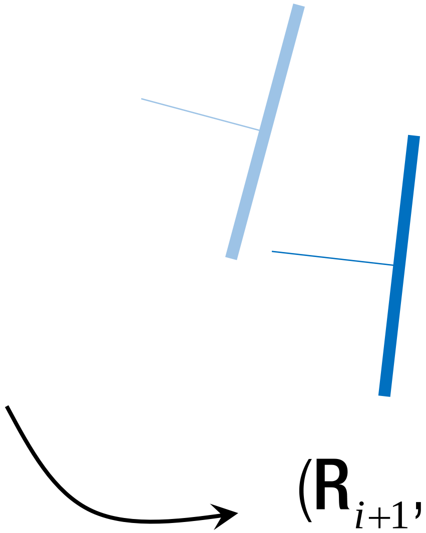
$$\Delta \mathbf{T}_i = \begin{bmatrix} \Delta \mathbf{R} & \Delta \mathbf{t} \\ \mathbf{0} & 1 \end{bmatrix} \approx \mathbf{I} + \mathbf{Z} \qquad \text{where} \quad \mathbf{Z}(\beta, \gamma, \alpha, t_x, t_y, t_z) = \begin{bmatrix} 0 & \alpha & -\gamma & t_x \\ -\alpha & 0 & \beta & t_x \\ \gamma & -\beta & 0 & t_x \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

$$= \begin{bmatrix} [\xi]_\times & \Delta \mathbf{t} \\ \mathbf{0} & 0 \end{bmatrix}$$

$$\Delta \mathbf{R}, \Delta \mathbf{t} = \underset{\Delta \mathbf{R}, \Delta \mathbf{t}}{\text{minimize}} \left\| \Delta \mathbf{R}\mathbf{p} + \Delta \mathbf{t} - \hat{\mathbf{p}} \right\|$$

$$\longrightarrow \quad \Delta \mathbf{R}\mathbf{p} + \Delta \mathbf{t} = \hat{\mathbf{p}}$$

Point Cloud Alignment



$(\mathbf{R}_{i+1}, \mathbf{C}_{i+1})$

$$\Delta\mathbf{T} = \left(\Delta\mathbf{R}, \Delta\mathbf{t}\right)$$

First order approximation:

$$\Delta\mathbf{T}_i = \begin{bmatrix} \Delta\mathbf{R} & \Delta\mathbf{t} \\ \mathbf{0} & 1 \end{bmatrix} \approx \mathbf{I} + \mathbf{Z} \quad \text{where} \quad \mathbf{Z}(\beta,\gamma,\alpha,t_x,t_y,t_z) = \begin{bmatrix} 0 & \alpha & -\gamma & t_x \\ -\alpha & 0 & \beta & t_x \\ \gamma & -\beta & 0 & t_x \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

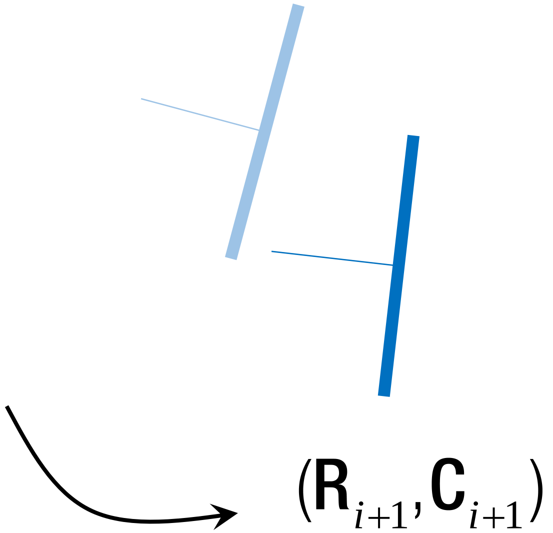
$$= \begin{bmatrix} [\xi]_\times & \Delta\mathbf{t} \\ \mathbf{0} & 0 \end{bmatrix}$$

$$\Delta\mathbf{R}, \Delta\mathbf{t} = \underset{\Delta\mathbf{R},\Delta\mathbf{t}}{\text{minimize}} \left\| \Delta\mathbf{R}\mathbf{p} + \Delta\mathbf{t} - \hat{\mathbf{p}} \right\|$$

$$\longrightarrow \quad \Delta\mathbf{R}\mathbf{p} + \Delta\mathbf{t} = \hat{\mathbf{p}}$$

$$\longrightarrow \quad (\mathbf{I} + [\xi]_\times)\mathbf{p} + \Delta\mathbf{t} = \hat{\mathbf{p}}$$

Point Cloud Alignment



$$\Delta \mathrm{T} = \left(\Delta \mathrm{R}, \Delta \mathrm{t}\right)$$

First order approximation:

$$\Delta \mathbf{T}_i = \begin{bmatrix} \Delta \mathbf{R} & \Delta \mathbf{t} \\ \mathbf{0} & 1 \end{bmatrix} \approx \mathbf{I} + \mathbf{Z} \quad \text{where} \quad \mathbf{Z}(\beta,\gamma,\alpha,t_x,t_y,t_z) = \begin{bmatrix} 0 & \alpha & -\gamma & t_x \\ -\alpha & 0 & \beta & t_x \\ \gamma & -\beta & 0 & t_x \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

$$= \begin{bmatrix} [\xi]_\times & \Delta \mathbf{t} \\ \mathbf{0} & 0 \end{bmatrix}$$

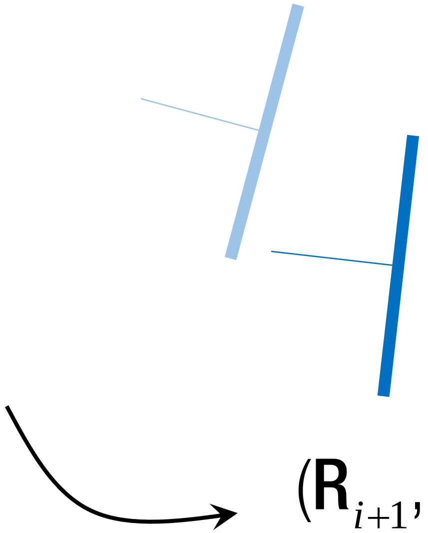
$$\Delta \mathbf{R}, \Delta \mathbf{t} = \underset{\Delta \mathbf{R}, \Delta \mathbf{t}}{\text{minimize}} \left\| \Delta \mathbf{R} \mathbf{p} + \Delta \mathbf{t} - \hat{\mathbf{p}} \right\|$$

$$\longrightarrow \quad \Delta \mathbf{R} \mathbf{p} + \Delta \mathbf{t} = \hat{\mathbf{p}}$$

$$\longrightarrow \quad (\mathbf{I} + [\xi]_\times)\mathbf{p} + \Delta \mathbf{t} = \hat{\mathbf{p}}$$

$$\longrightarrow \quad [\mathbf{p}]_\times \xi + \Delta \mathbf{t} = \hat{\mathbf{p}} - \mathbf{p}$$

Point Cloud Alignment



$$(\mathbf{R}_{i+1}, \mathbf{C}_{i+1})$$

$$\Delta \mathbf{T} = \left(\Delta \mathbf{R}, \Delta \mathbf{t}\right)$$

First order approximation:

$$\Delta \mathbf{T}_i = \begin{bmatrix} \Delta \mathbf{R} & \Delta \mathbf{t} \\ \mathbf{0} & 1 \end{bmatrix} \approx \mathbf{I} + \mathbf{Z} \quad \text{where} \quad \mathbf{Z}(\beta, \gamma, \alpha, t_x, t_y, t_z) = \begin{bmatrix} 0 & \alpha & -\gamma & t_x \\ -\alpha & 0 & \beta & t_x \\ \gamma & -\beta & 0 & t_x \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

$$= \begin{bmatrix} [\xi]_\times & \Delta \mathbf{t} \\ \mathbf{0} & 0 \end{bmatrix}$$

$$\Delta \mathbf{R}, \Delta \mathbf{t} = \underset{\Delta \mathbf{R}, \Delta \mathbf{t}}{\text{minimize}} \left\| \Delta \mathbf{R} \mathbf{p} + \Delta \mathbf{t} - \hat{\mathbf{p}} \right\|$$

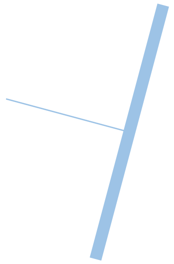
$$\longrightarrow \quad \Delta \mathbf{R} \mathbf{p} + \Delta \mathbf{t} = \hat{\mathbf{p}}$$

$$\longrightarrow \quad (\mathbf{I} + [\xi]_\times)\mathbf{p} + \Delta \mathbf{t} = \hat{\mathbf{p}}$$

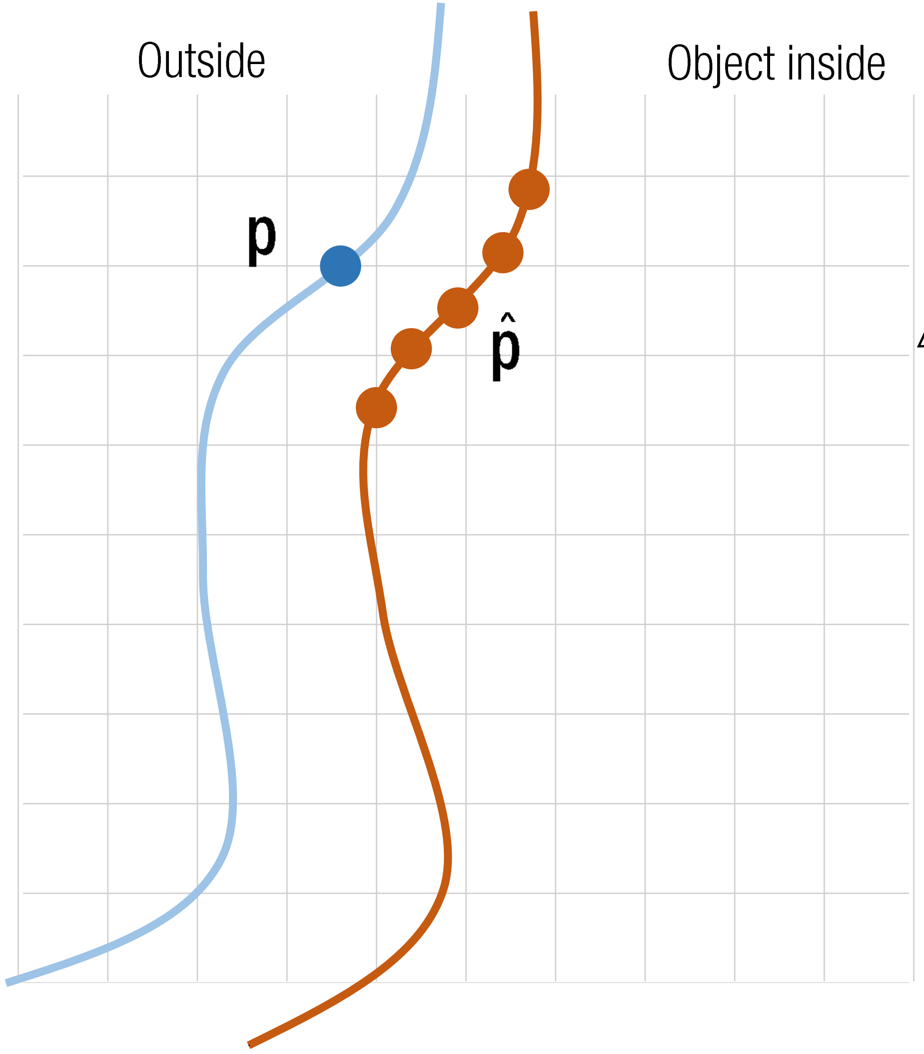
$$\longrightarrow \quad [\mathbf{p}]_\times \xi + \Delta \mathbf{t} = \hat{\mathbf{p}} - \mathbf{p}$$

$$\begin{bmatrix} [\mathbf{p}]_\times & \mathbf{I} \end{bmatrix} \begin{bmatrix} \xi \\ \Delta \mathbf{t} \end{bmatrix} = \hat{\mathbf{p}} - \mathbf{p}$$

Iterative Closest Point

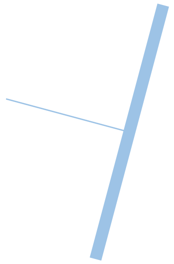


$(\hat{\mathbf{R}}_{i+1}, \hat{\mathbf{C}}_{i+1})$

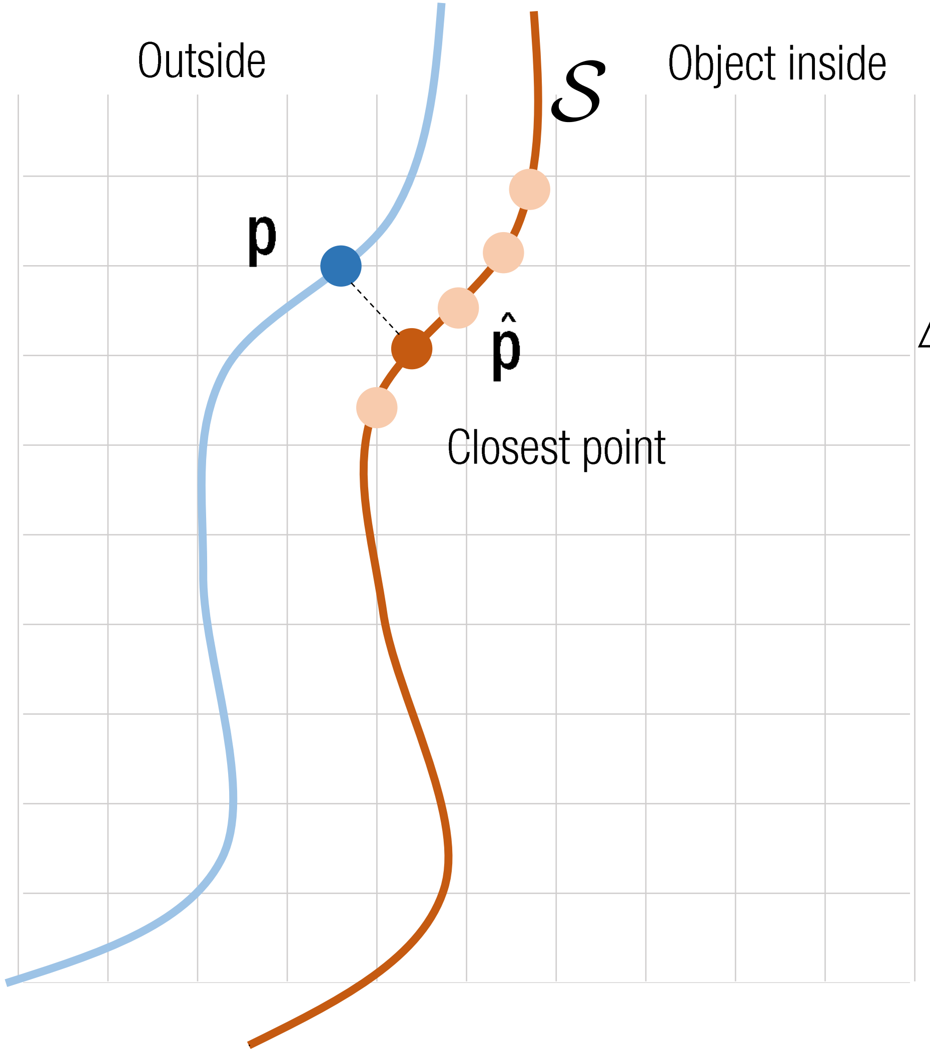


$$\Delta\mathbf{R}, \Delta\mathbf{t} = \underset{\Delta\mathbf{R},\Delta\mathbf{t}}{\text{minimize}} \left\| \Delta\mathbf{R}\mathbf{p} + \Delta\mathbf{t} - \hat{\mathbf{p}} \right\|$$

Iterative Closest Point



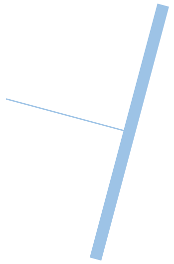
$(\hat{\mathbf{R}}_{i+1}, \hat{\mathbf{C}}_{i+1})$



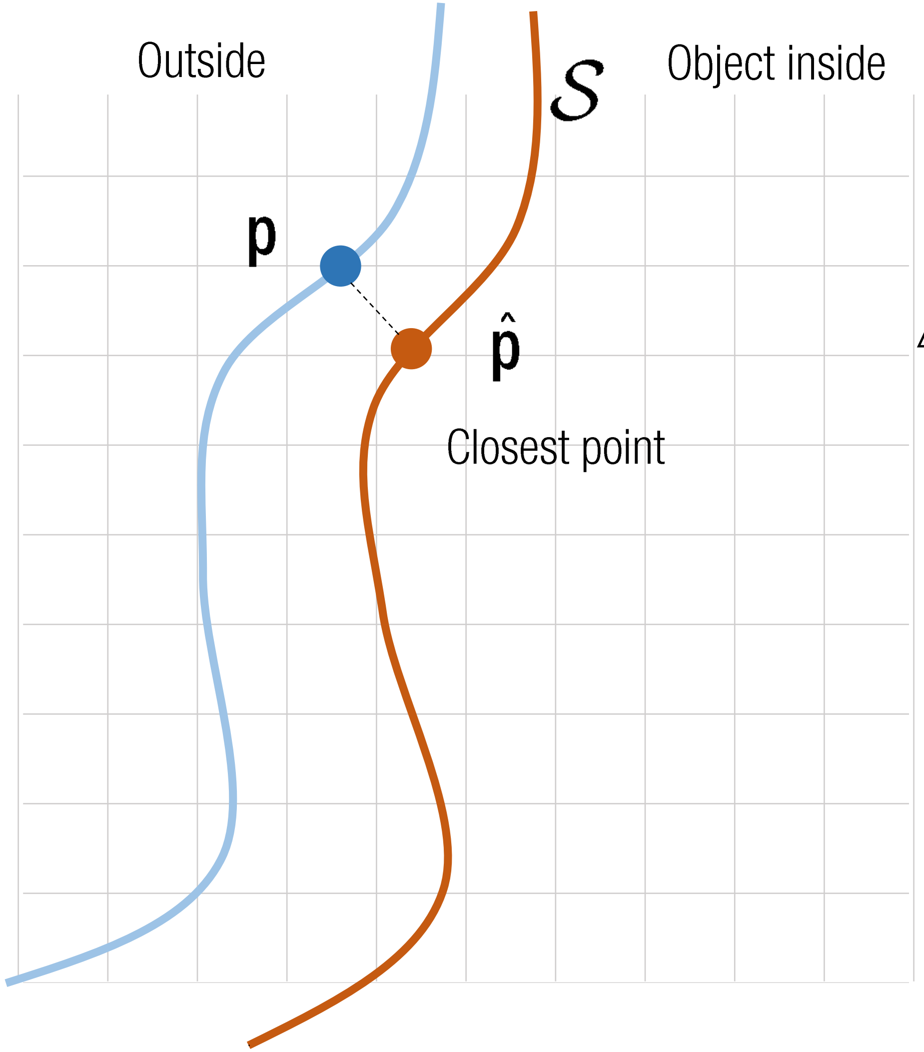
$$\Delta\mathbf{R}, \Delta\mathbf{t} = \underset{\Delta\mathbf{R}, \Delta\mathbf{t}}{\text{minimize}} \left\| \Delta\mathbf{R}\mathbf{p} + \Delta\mathbf{t} - \hat{\mathbf{p}} \right\|$$

where $\hat{\mathbf{p}} = \min_{\mathbf{x} \in \mathcal{S}} \left\| \Delta\mathbf{R}\mathbf{p} + \Delta\mathbf{t} - \mathbf{x} \right\|$

Iterative Closest Point



$(\hat{R}_{i+1}, \hat{C}_{i+1})$

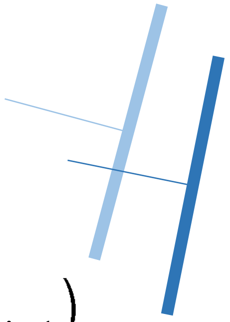


Unknown correspondence

$$\Delta R, \Delta t = \underset{\Delta R, \Delta t}{\text{minimize}} \left\| \Delta R p + \Delta t - \hat{p} \right\|$$

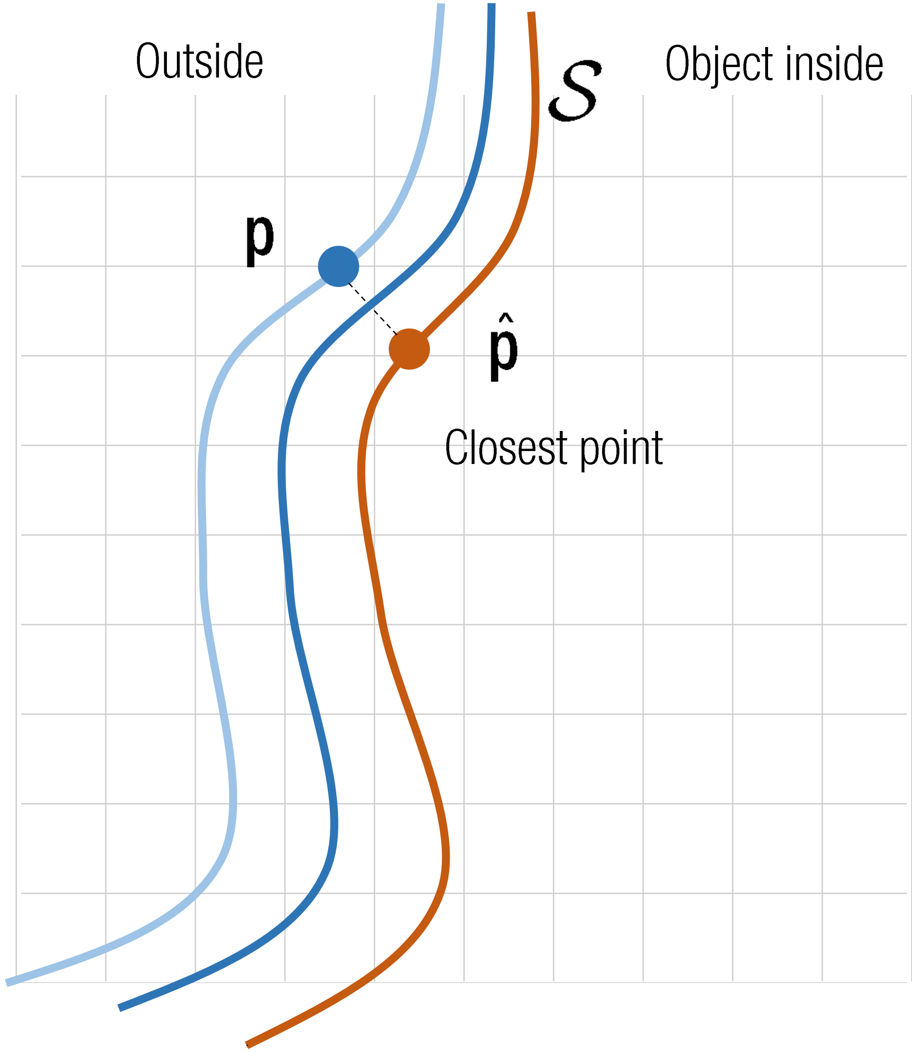
where $\hat{p} = \min_{x \in \mathcal{S}} \left\| \Delta R p + \Delta t - x \right\|$

Iterative Closest Point



$(\hat{\mathbf{R}}_{i+1}, \hat{\mathbf{C}}_{i+1})$

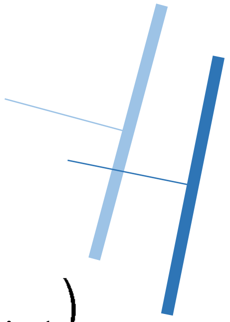
$(\hat{\mathbf{R}}^1_{i+1}, \hat{\mathbf{C}}^1_{i+1})$



$$\Delta\mathbf{R}, \Delta\mathbf{t} = \underset{\Delta\mathbf{R},\Delta\mathbf{t}}{\text{minimize}} \left\| \Delta\mathbf{R}\mathbf{p} + \Delta\mathbf{t} - \hat{\mathbf{p}} \right\|$$

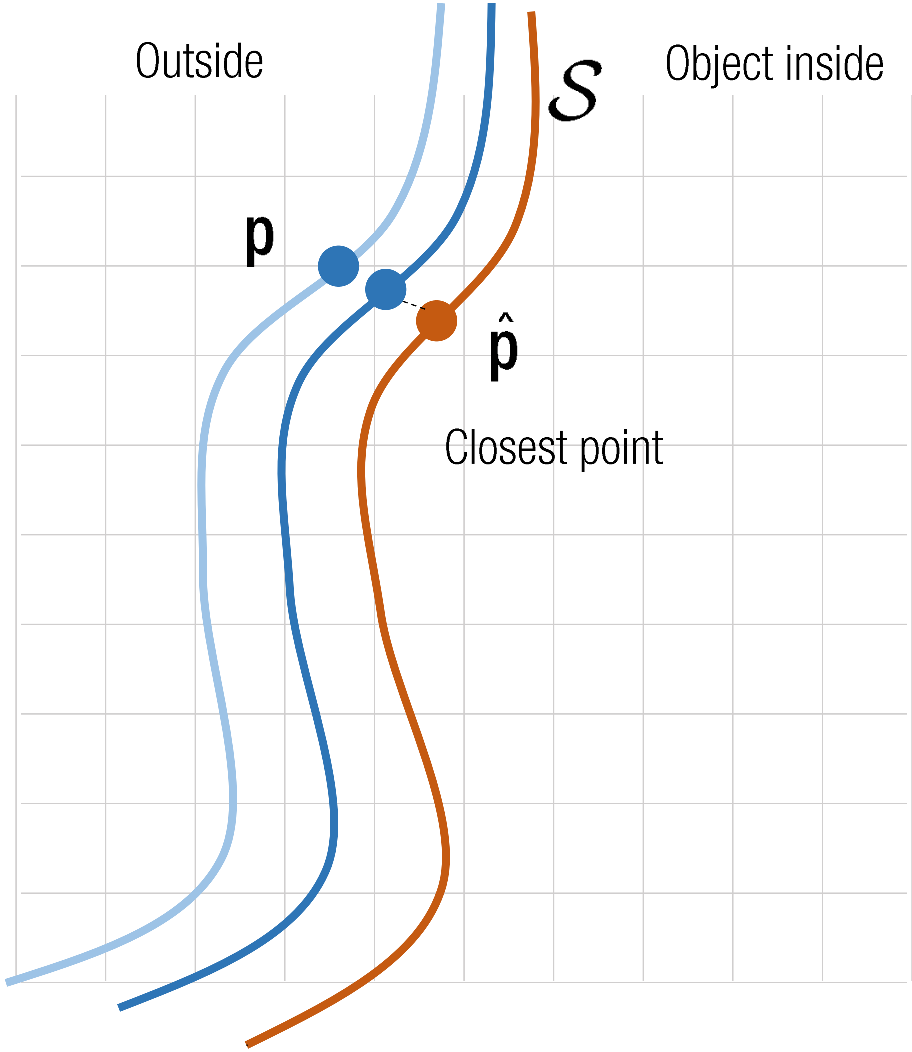
where $\hat{\mathbf{p}} = \min_{\mathbf{x}\in\mathcal{S}} \left\| \Delta\mathbf{R}\mathbf{p} + \Delta\mathbf{t} - \mathbf{x} \right\|$

Iterative Closest Point



$(\hat{\mathbf{R}}_{i+1}, \hat{\mathbf{C}}_{i+1})$

$(\hat{\mathbf{R}}^1_{i+1}, \hat{\mathbf{C}}^1_{i+1})$

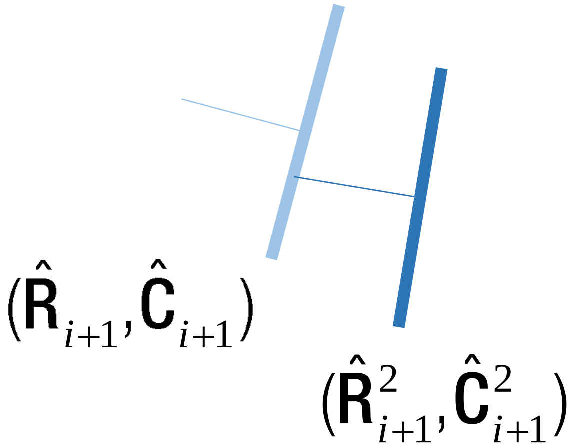


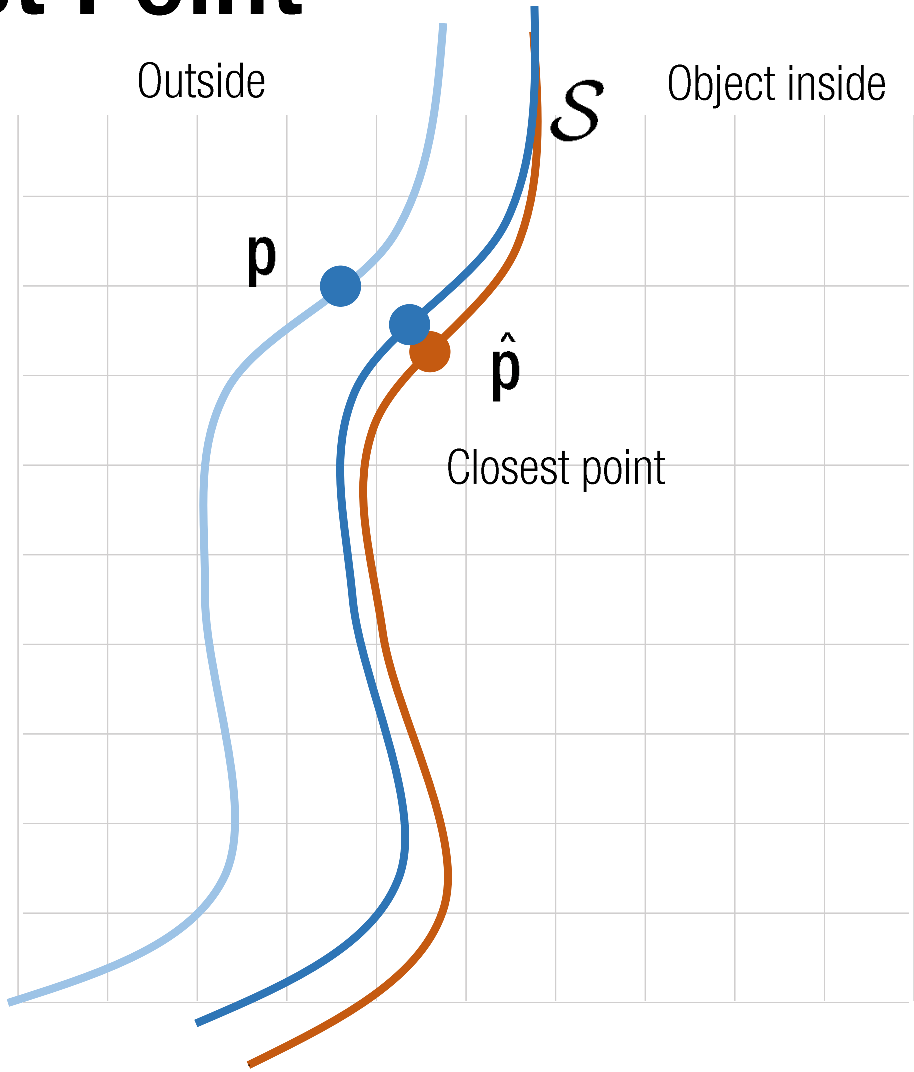
Unknown correspondence

$$\Delta\mathbf{R}, \Delta\mathbf{t} = \underset{\Delta\mathbf{R}, \Delta\mathbf{t}}{\text{minimize}} \left\| \Delta\mathbf{R}\mathbf{p} + \Delta\mathbf{t} - \hat{\mathbf{p}} \right\|$$

where $\hat{\mathbf{p}} = \min_{\mathbf{x} \in \mathcal{S}} \left\| \Delta\mathbf{R}\mathbf{p} + \Delta\mathbf{t} - \mathbf{x} \right\|$

Iterative Closest Point



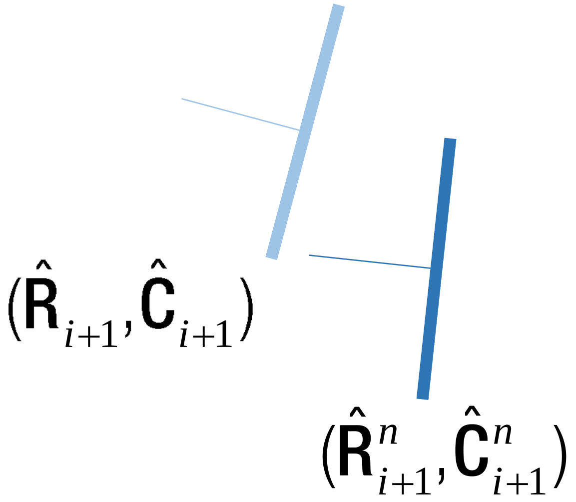


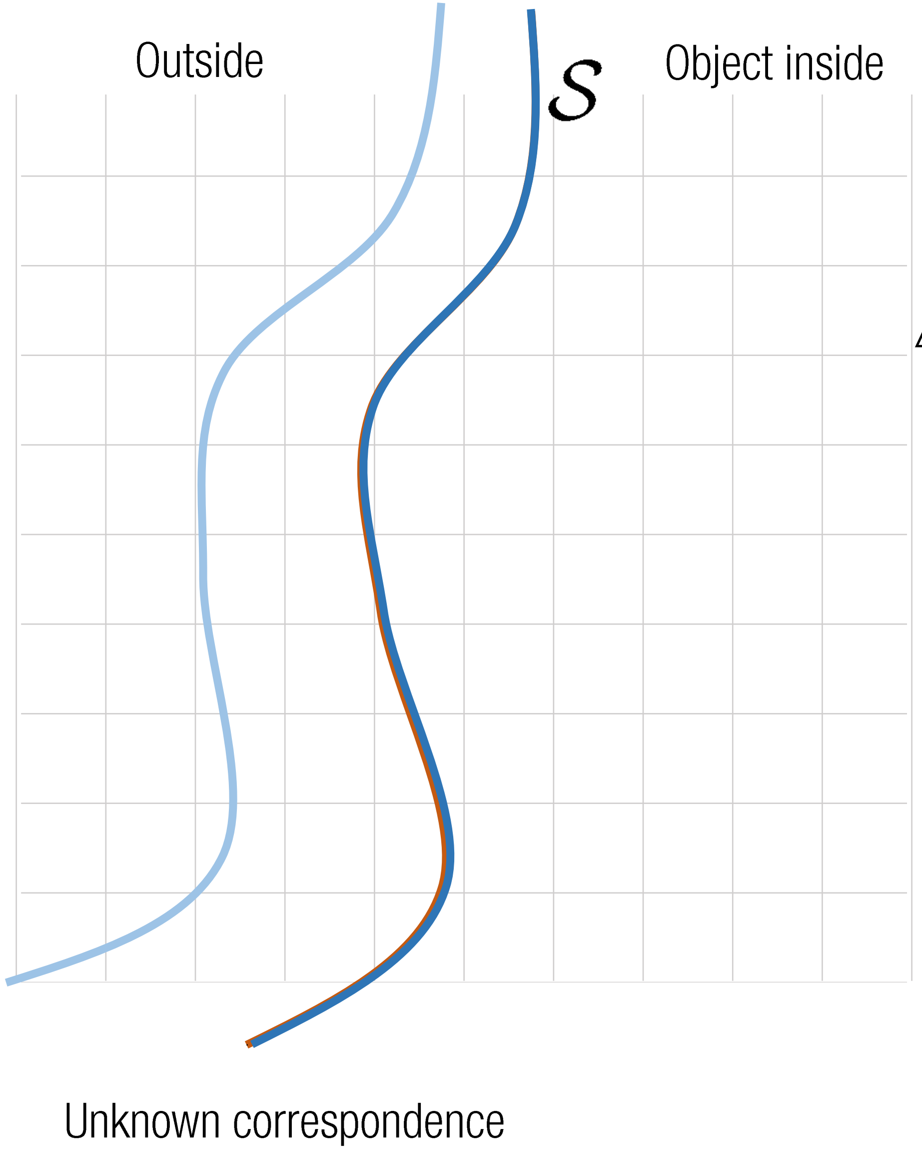
Unknown correspondence

$$\Delta R, \Delta t = \underset{\Delta R, \Delta t}{\text{minimize}} \left\| \Delta R p + \Delta t - \hat{p} \right\|$$

where $\hat{p} = \min_{x \in \mathcal{S}} \left\| \Delta R p + \Delta t - x \right\|$

Iterative Closest Point

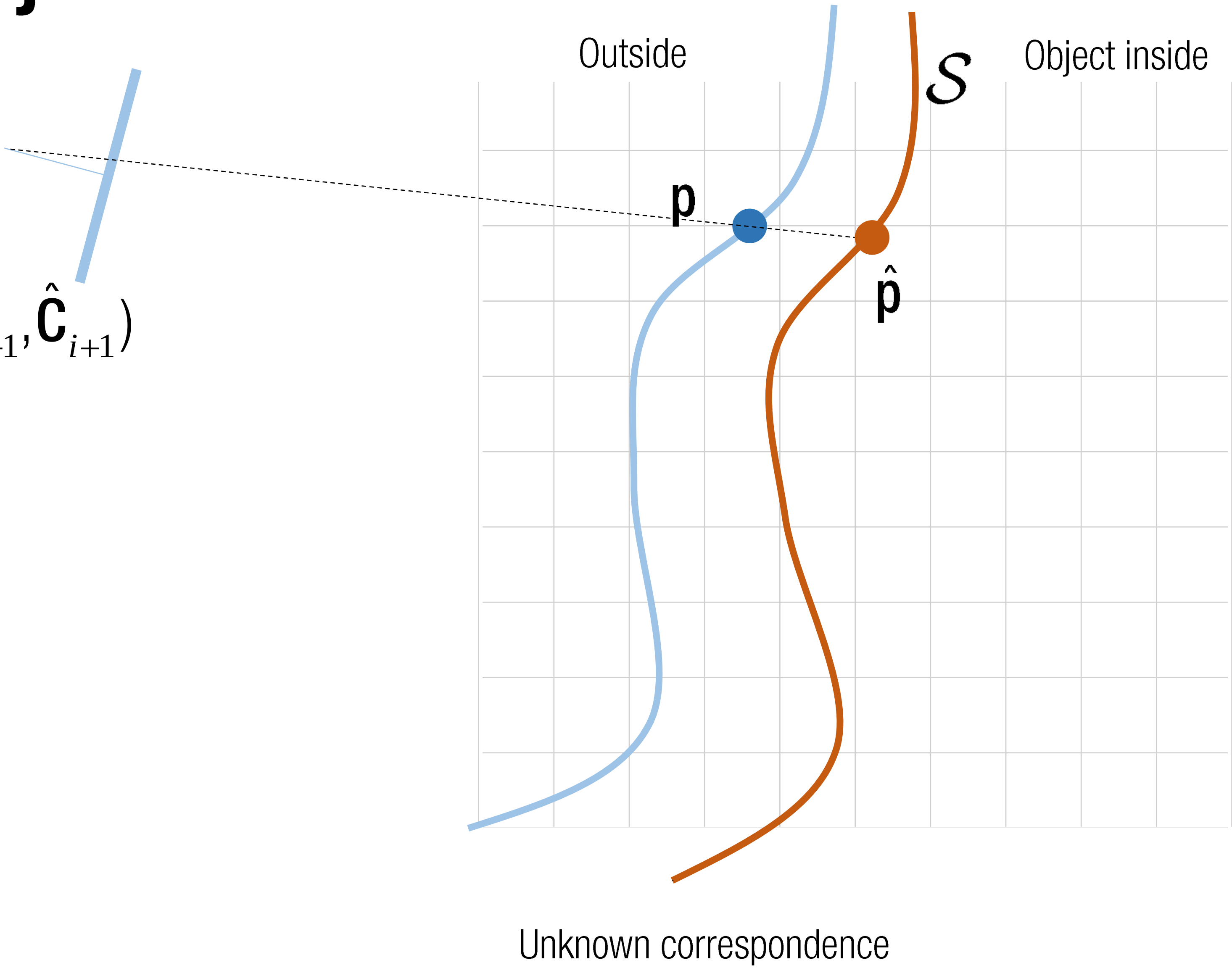




$$\Delta\mathbf{R}, \Delta\mathbf{t} = \underset{\Delta\mathbf{R},\Delta\mathbf{t}}{\text{minimize}} \left\| \Delta\mathbf{R}\mathbf{p} + \Delta\mathbf{t} - \hat{\mathbf{p}} \right\|$$

where $\hat{\mathbf{p}} = \min_{\mathbf{x}\in\mathcal{S}} \left\| \Delta\mathbf{R}\mathbf{p} + \Delta\mathbf{t} - \mathbf{x} \right\|$

Projective ICP



$$\mathbf{P}_{i+1} = (\hat{\mathbf{R}}_{i+1}, \hat{\mathbf{C}}_{i+1})$$

$$\Delta\mathbf{R}, \Delta\mathbf{t} = \underset{\Delta\mathbf{R},\Delta\mathbf{t}}{\text{minimize}} \left\| \Delta\mathbf{R}\mathbf{p} + \Delta\mathbf{t} - \hat{\mathbf{p}} \right\|$$

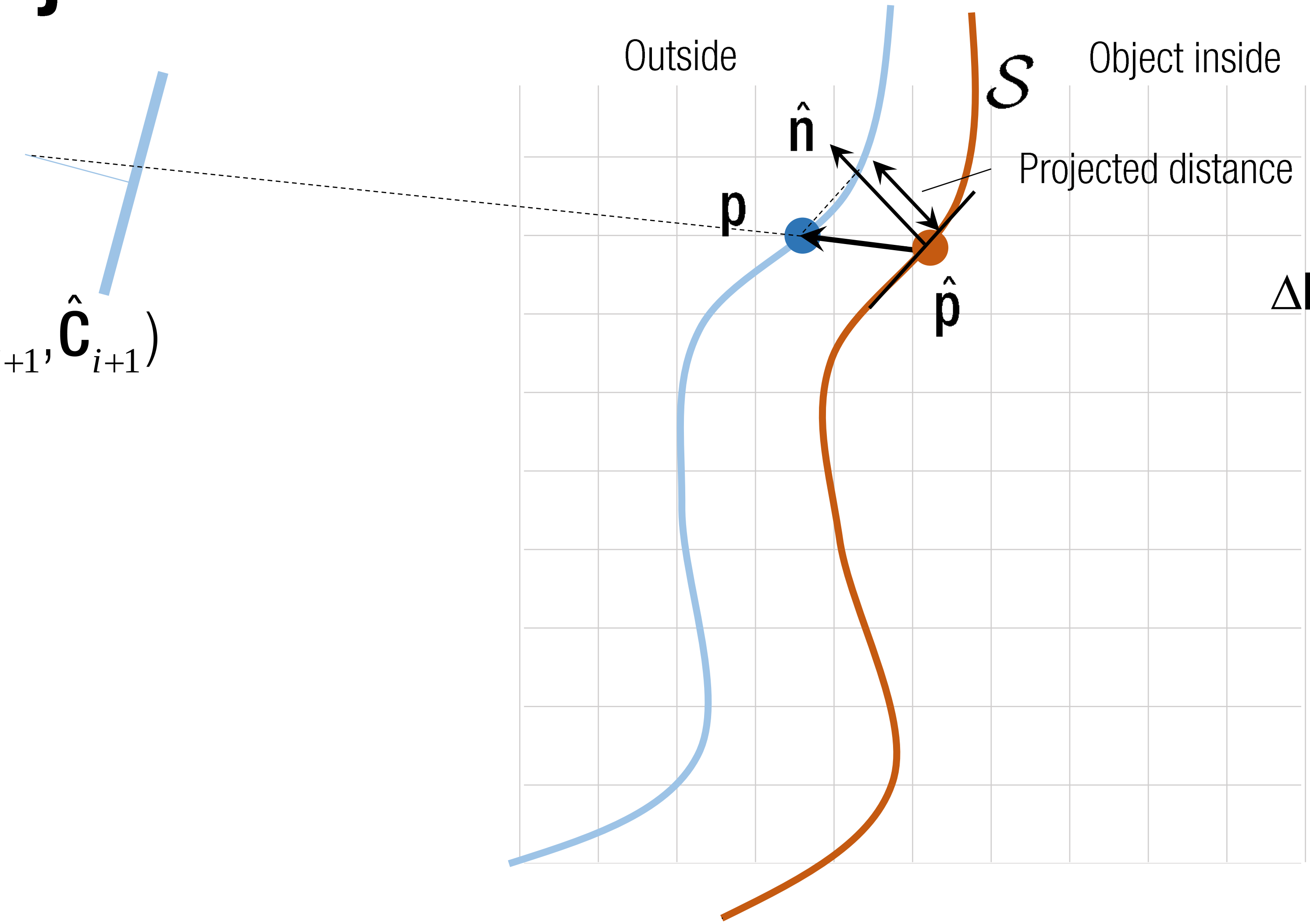
where $\hat{\mathbf{p}} = \min_{\mathbf{x}\in\mathcal{X}} \left\| \Delta\mathbf{R}\mathbf{p} + \Delta\mathbf{t} - \mathbf{x} \right\|$

$$\mathcal{X} = \left\{ \mathbf{x} \mid \mathbf{P}_{i+1}\Delta\mathbf{T}\mathbf{x} \propto \mathbf{P}_{i+1}\Delta\mathbf{T}\mathbf{p}, \mathbf{x} \in \mathcal{S} \right\}$$

Projections of two points must be the same.

Projective ICP

$$P_{i+1} = (\hat{R}_{i+1}, \hat{C}_{i+1})$$



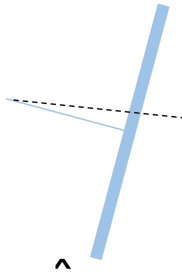
$$\Delta R, \Delta t = \underset{\Delta R, \Delta t}{\text{minimize}} \left\| (\Delta R \mathbf{p} + \Delta \mathbf{t} - \hat{\mathbf{p}})^{T} \mathbf{n} \right\|$$

where $\hat{\mathbf{p}} = \min_{\mathbf{x} \in \mathcal{X}} \left\| \Delta R \mathbf{p} + \Delta \mathbf{t} - \mathbf{x} \right\|$

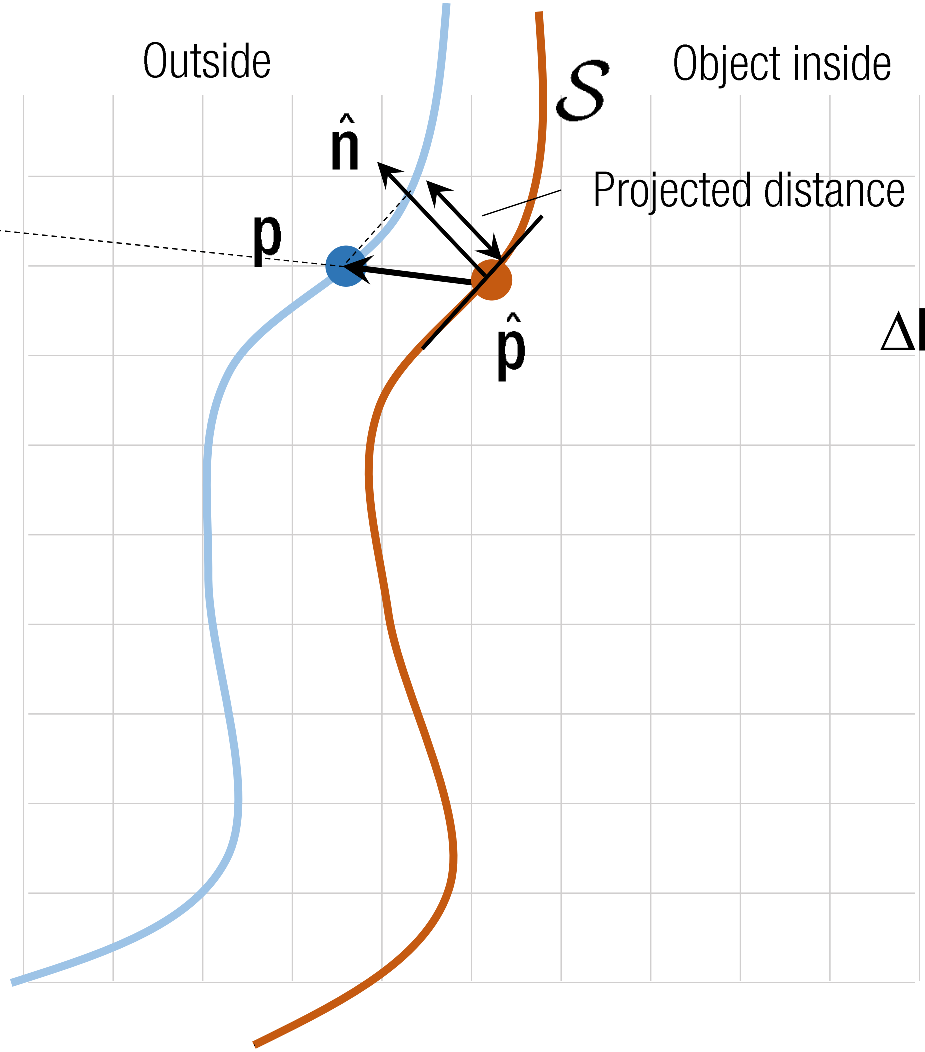
$$\mathcal{X} = \left\{ \mathbf{x} \mid P_{i+1} \Delta T \mathbf{x} \propto P_{i+1} \Delta T \mathbf{p}, \mathbf{x} \in \mathcal{S} \right\}$$

Projections of two points must be the same.

Projective ICP



$$\mathbf{P}_{i+1} = (\hat{\mathbf{R}}_{i+1}, \hat{\mathbf{C}}_{i+1})$$



Unknown correspondence

$$\Delta\mathbf{R},\Delta\mathbf{t} = \underset{\Delta\mathbf{R},\Delta\mathbf{t}}{\text{minimize}} \left\| (\Delta\mathbf{R}\mathbf{p}+\Delta\mathbf{t}-\hat{\mathbf{p}})^{\mathsf{T}}\mathbf{n}\right\|$$

where $\hat{\mathbf{p}} = \underset{\mathbf{x}\in\mathcal{X}}{\min}\left\|\Delta\mathbf{R}\mathbf{p}+\Delta\mathbf{t}-\mathbf{x}\right\|$

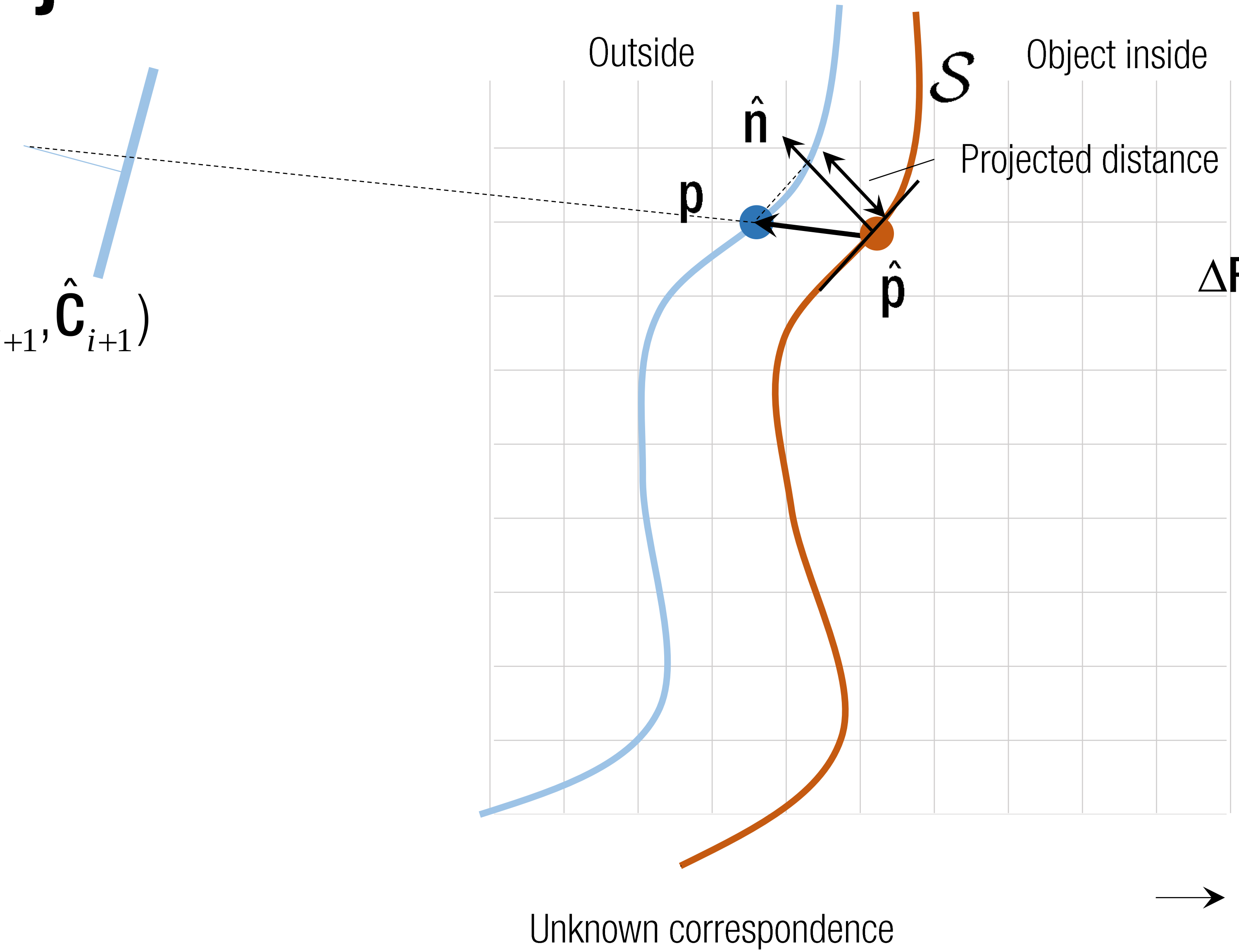
$$\mathcal{X} = \left\{\mathbf{x} \mid \mathbf{P}_{i+1}\Delta\mathbf{T}\mathbf{x} \propto \mathbf{P}_{i+1}\Delta\mathbf{T}\mathbf{p}, \mathbf{x}\in\mathcal{S}\right\}$$

Projections of two points must be the same.

$$\begin{bmatrix}[\mathbf{p}]_\times & \mathbf{I}\end{bmatrix}\begin{bmatrix}\xi \\ \Delta\mathbf{t}\end{bmatrix} = \hat{\mathbf{p}} - \mathbf{p}$$

Projective ICP

$$\mathbf{P}_{i+1} = (\hat{\mathbf{R}}_{i+1}, \hat{\mathbf{C}}_{i+1})$$



$$\Delta\mathbf{R}, \Delta\mathbf{t} = \underset{\Delta\mathbf{R},\Delta\mathbf{t}}{\text{minimize}} \left\| (\Delta\mathbf{R}\mathbf{p} + \Delta\mathbf{t} - \hat{\mathbf{p}})^{\mathsf{T}} \mathbf{n} \right\|$$

where $\hat{\mathbf{p}} = \min_{\mathbf{x}\in\mathcal{X}} \left\| \Delta\mathbf{R}\mathbf{p} + \Delta\mathbf{t} - \mathbf{x} \right\|$

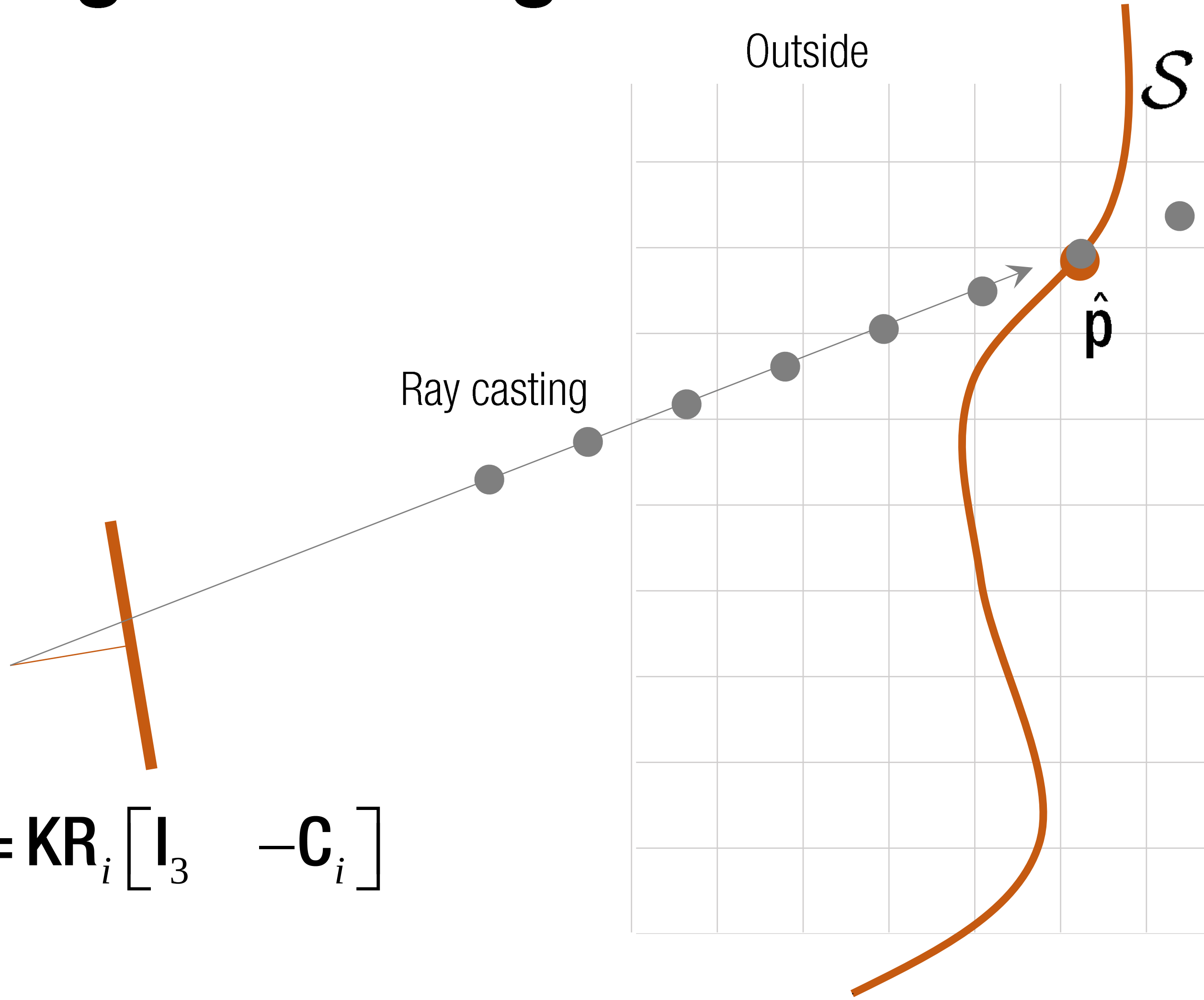
$$\mathcal{X} = \left\{ \mathbf{x} \mid \mathbf{P}_{i+1}\Delta\mathbf{T}\mathbf{x} \propto \mathbf{P}_{i+1}\Delta\mathbf{T}\mathbf{p}, \mathbf{x} \in \mathcal{S} \right\}$$

Projections of two points must be the same.

$$\begin{bmatrix} [\mathbf{p}]_\times & \mathbf{I} \end{bmatrix} \begin{bmatrix} \xi \\ \Delta\mathbf{t} \end{bmatrix} = \hat{\mathbf{p}} - \mathbf{p}$$

$$\rightarrow \mathbf{n}^{\mathsf{T}} \begin{bmatrix} [\mathbf{p}]_\times & \mathbf{I} \end{bmatrix} \begin{bmatrix} \xi \\ \Delta\mathbf{t} \end{bmatrix} = \mathbf{n}^{\mathsf{T}}(\hat{\mathbf{p}} - \mathbf{p})$$

Alignment Algorithm



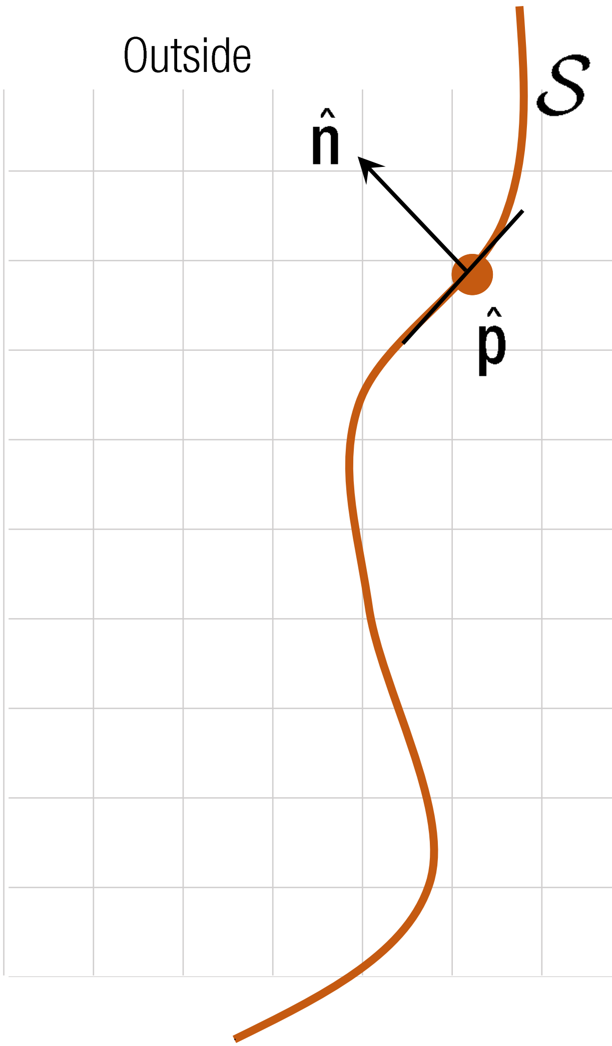
$$\mathbf{P}_i = \mathbf{K}\mathbf{R}_i \begin{bmatrix} \mathbf{I}_3 & -\mathbf{C}_i \end{bmatrix}$$

```
    Predict points from the TSDF by ray casting
    Predict normals from the predicted points
    Load the i^th depth image
    Compute 3D points and surface normals.
    for j^th iteration < n_iter do
        Find correspondences
        Compute ΔT
        Update T_curr and make sure det(R_curr) = 1
    end for
    Update transformation T = T_curr
```

For each pixel in the image *i*, find the surface point.

Note: ray casting is expensive operation. Use the surface predicted by image *i*.

Alignment Algorithm

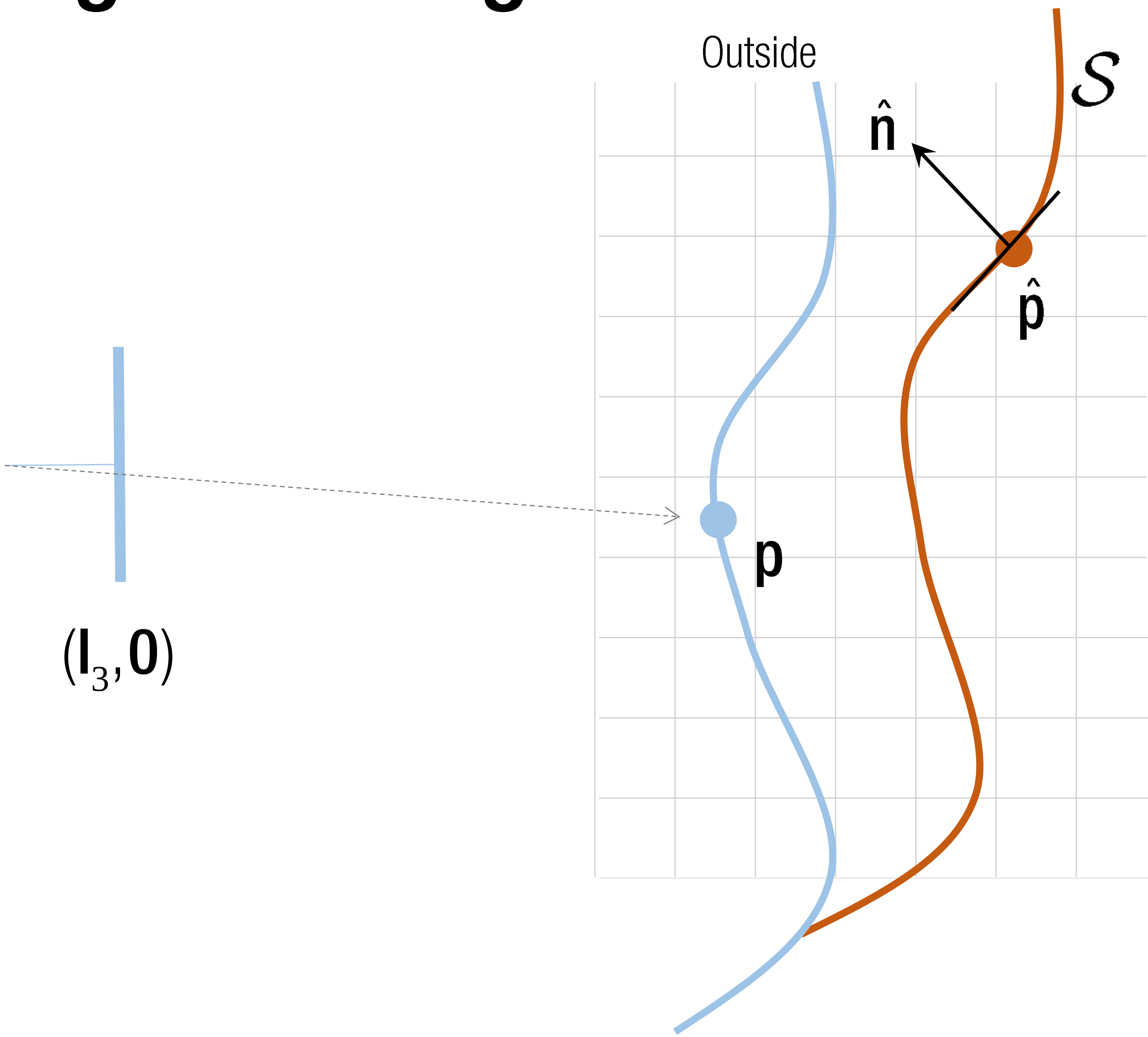



```
Predict points from the TSDF by ray casting
Predict normals from the predicted points
Load the i^th depth image
Compute 3D points and surface normals.
for j^th iteration < n_iter do
    Find correspondences
    Compute ΔT
    Update T_curr and make sure det(R_curr) = 1
end for
Update transformation T = T_curr
```

$$\mathbf{n} = \frac{\nabla f}{\|\nabla f\|}$$

$$= \frac{f(\mathbf{p} + \Delta\mathbf{p}) - f(\mathbf{p})}{\|f(\mathbf{p} + \Delta\mathbf{p}) - f(\mathbf{p})\|}$$

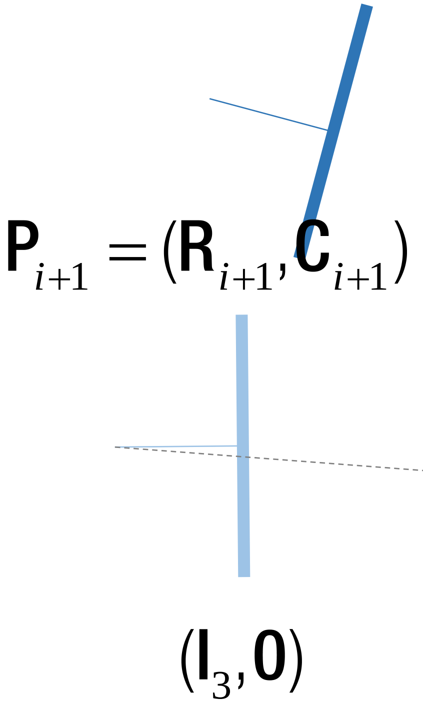
Alignment Algorithm

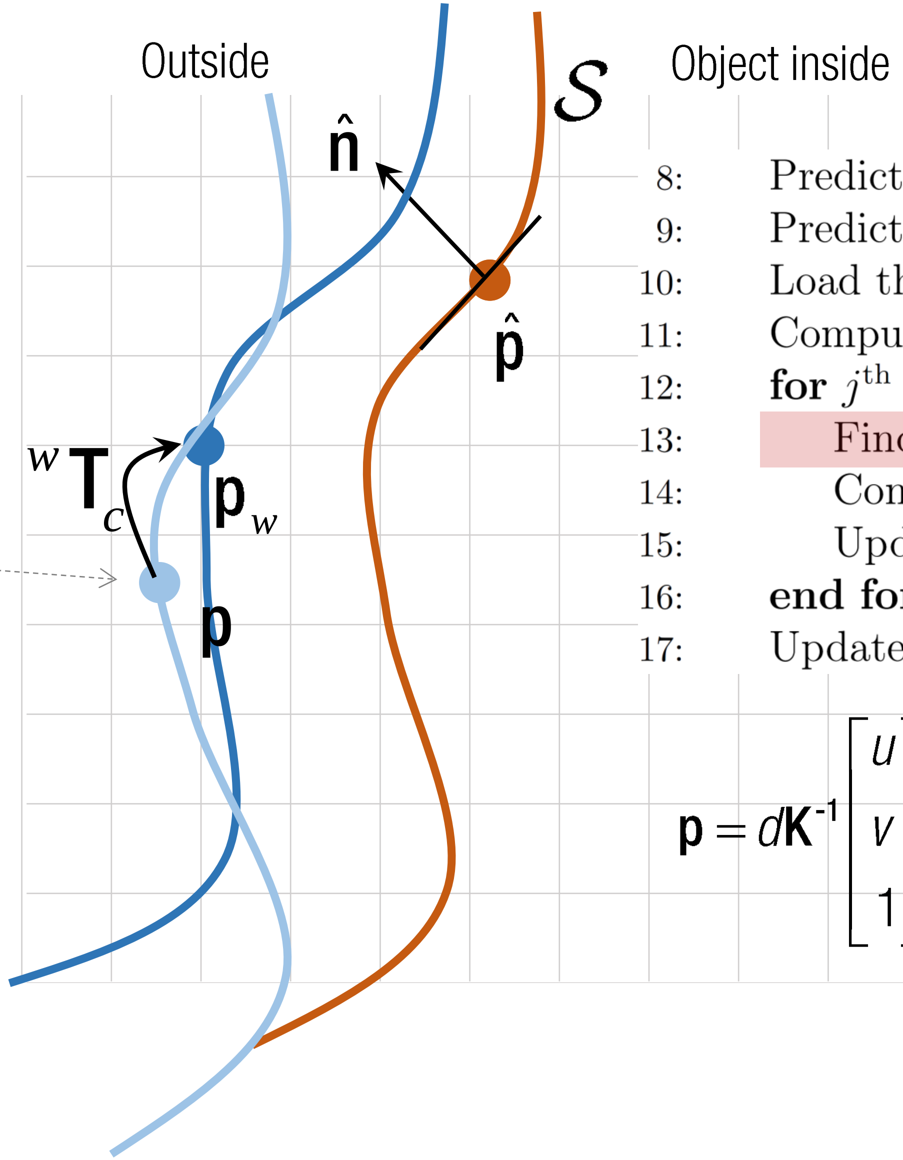


Predict points from the TSDF by ray casting
Predict normals from the predicted points
Load the i^{th} depth image
Compute 3D points and surface normals.
for j^{th} iteration $< n_{\text{iter}}$ **do**
  Find correspondences
  Compute ΔT
  Update $\mathbf{T}_{\text{curr}}$ and make sure $\det(\mathbf{R}_{\text{curr}}) = 1$
end for
Update transformation $\mathbf{T} = \mathbf{T}_{\text{curr}}$

$$\mathbf{p} = d\mathbf{K}^{-1}\begin{bmatrix} u \\ v \\ 1 \end{bmatrix}$$

Alignment Algorithm



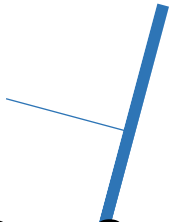


Predict points from the TSDF by ray casting
Predict normals from the predicted points
Load the i^{th} depth image
Compute 3D points and surface normals.
for j^{th} iteration $< n_{\text{iter}}$ **do**
    Find correspondences
    Compute ΔT
    Update $\mathbf{T}_{\text{curr}}$ and make sure $\det(\mathbf{R}_{\text{curr}}) = 1$
end for
Update transformation $\mathbf{T} = \mathbf{T}_{\text{curr}}$

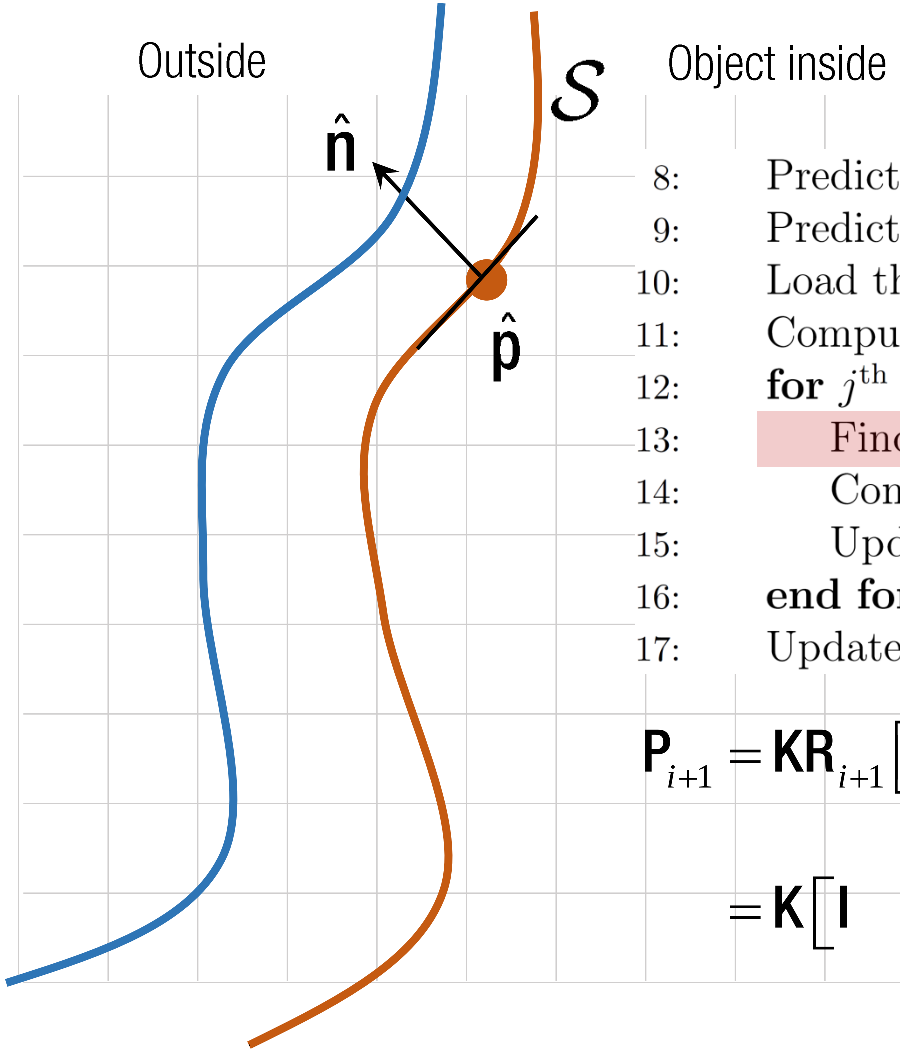
$$\mathbf{p} = d\mathbf{K}^{-1}\begin{bmatrix} u \\ v \\ 1 \end{bmatrix}$$

$${}^{w}\mathbf{T}_{c} = \begin{bmatrix} \mathbf{R} & \mathbf{t} \\ 0 & 1 \end{bmatrix}$$

Alignment Algorithm



$$P_{i+1} = (R_{i+1}, C_{i+1})$$




```
Predict points from the TSDF by ray casting
Predict normals from the predicted points
Load the i^th depth image
Compute 3D points and surface normals.
for j^th iteration < n_iter do
    Find correspondences
    Compute ΔT
    Update T_curr and make sure det(R_curr) = 1
end for
Update transformation T = T_curr
```

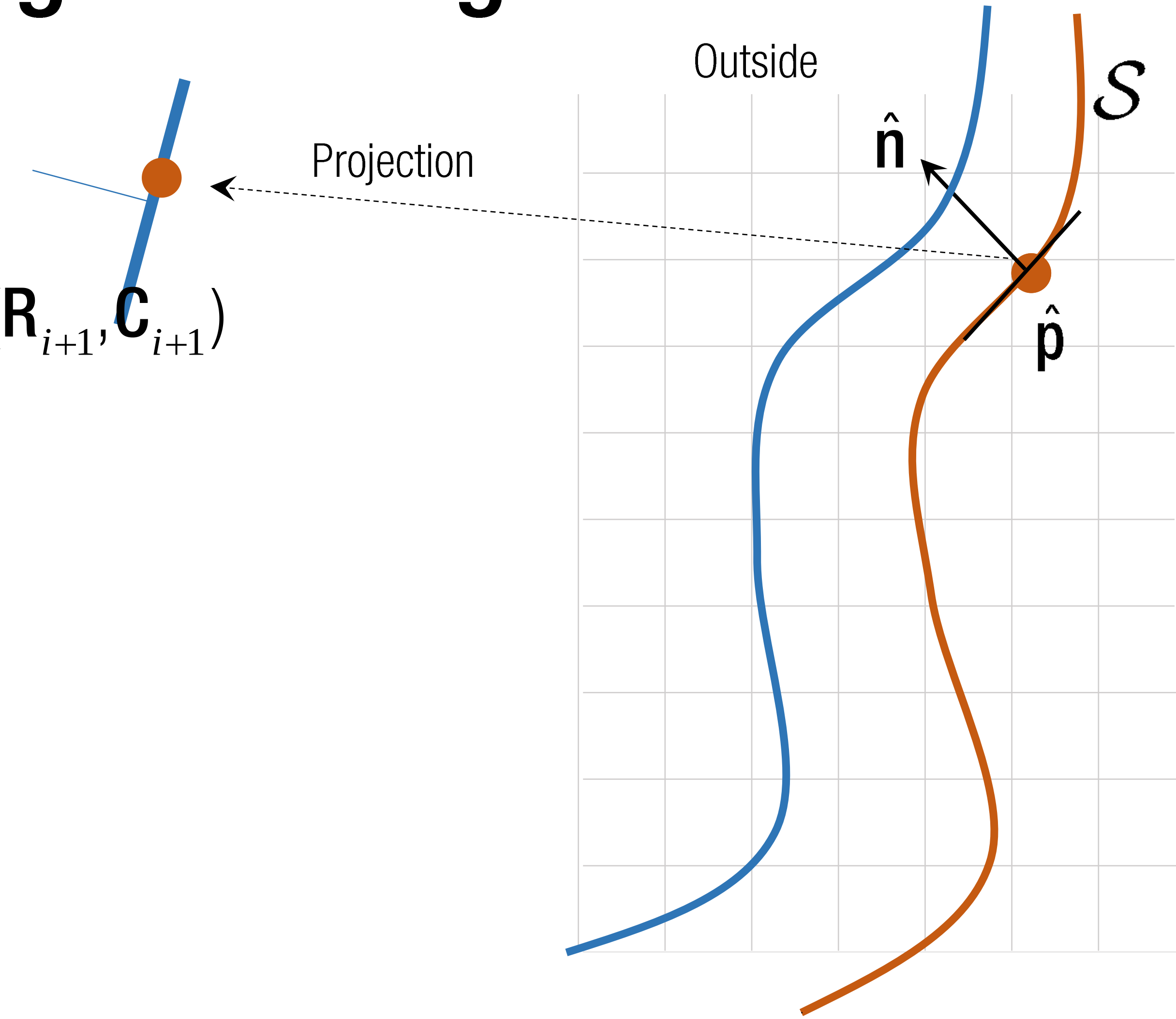
$$P_{i+1} = KR_{i+1}\begin{bmatrix} I & -C_{i+1} \end{bmatrix}$$

$$= K\begin{bmatrix} I & 0 \end{bmatrix}\begin{bmatrix} R & t \\ 0 & 1 \end{bmatrix}^{-1}$$

$$^{w}T_{c} = \begin{bmatrix} R & t \\ 0 & 1 \end{bmatrix}$$

$$\longrightarrow R_{i+1} = R^{\top}, C_{i+1} = t$$

Alignment Algorithm

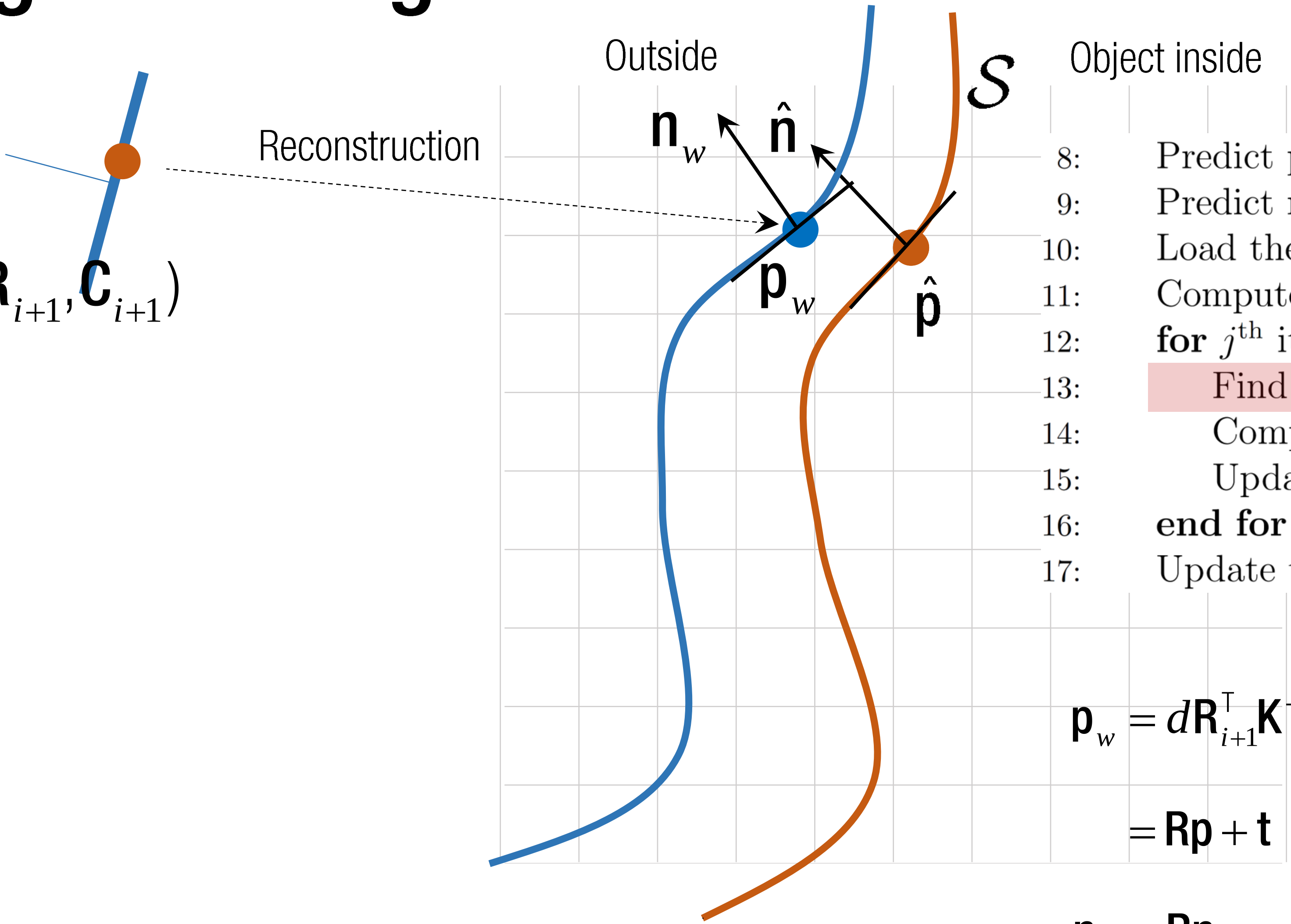


$$\mathbf{P}_{i+1} = (\mathbf{R}_{i+1}, \mathbf{C}_{i+1})$$

Predict points from the TSDF by ray casting
Predict normals from the predicted points
Load the i^{th} depth image
Compute 3D points and surface normals.
for j^{th} iteration $< n_{\text{iter}}$ **do**
    Find correspondences
    Compute ΔT
    Update $\mathbf{T}_{\text{curr}}$ and make sure $\det(\mathbf{R}_{\text{curr}}) = 1$
end for
Update transformation $\mathbf{T} = \mathbf{T}_{\text{curr}}$

$$\lambda \begin{bmatrix} \hat{\mathbf{u}} \\ 1 \end{bmatrix} = \mathbf{P}_{i+1} \begin{bmatrix} \hat{\mathbf{p}} \\ 1 \end{bmatrix}$$

Alignment Algorithm



$$\mathbf{P}_{i+1} = (\mathbf{R}_{i+1}, \mathbf{C}_{i+1})$$

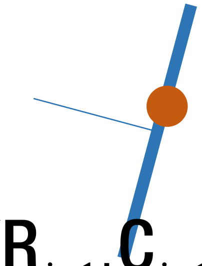
```
Predict points from the TSDF by ray casting
Predict normals from the predicted points
Load the i^th depth image
Compute 3D points and surface normals.
for j^th iteration < n_iter do
    Find correspondences
    Compute ΔT
    Update T_curr and make sure det(R_curr) = 1
end for
Update transformation T = T_curr
```

$$\mathbf{p}_w = d\mathbf{R}_{i+1}^{\top}\mathbf{K}^{-1}\begin{bmatrix}\hat{\mathbf{u}} \\ 1\end{bmatrix} + \mathbf{C}_{i+1}$$

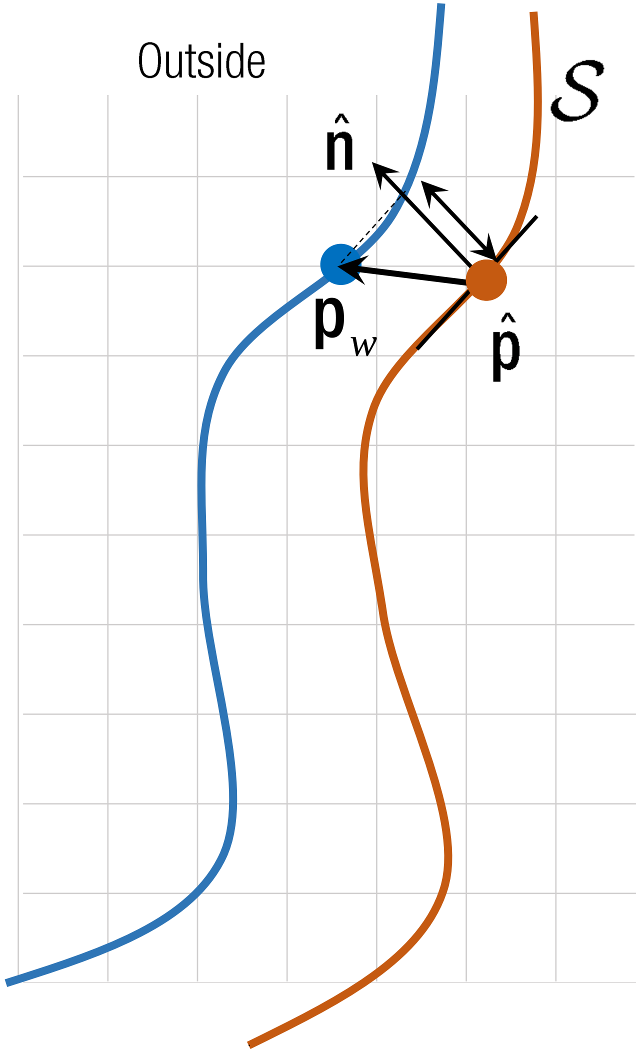
$$= \mathbf{R}\mathbf{p} + \mathbf{t}$$

$$\mathbf{n}_w = \mathbf{R}\mathbf{n}$$

Alignment Algorithm



$$\mathbf{P}_{i+1} = (\mathbf{R}_{i+1}, \mathbf{C}_{i+1})$$

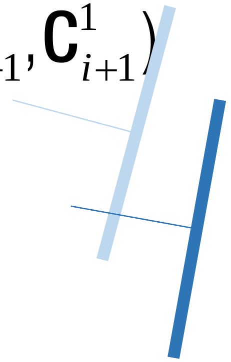



```
    Predict points from the TSDF by ray casting
    Predict normals from the predicted points
    Load the i^th depth image
    Compute 3D points and surface normals.
    for j^th iteration < n_iter do
        Find correspondences
        Compute ΔT
        Update T_curr and make sure det(R_curr) = 1
    end for
    Update transformation T = T_curr
```

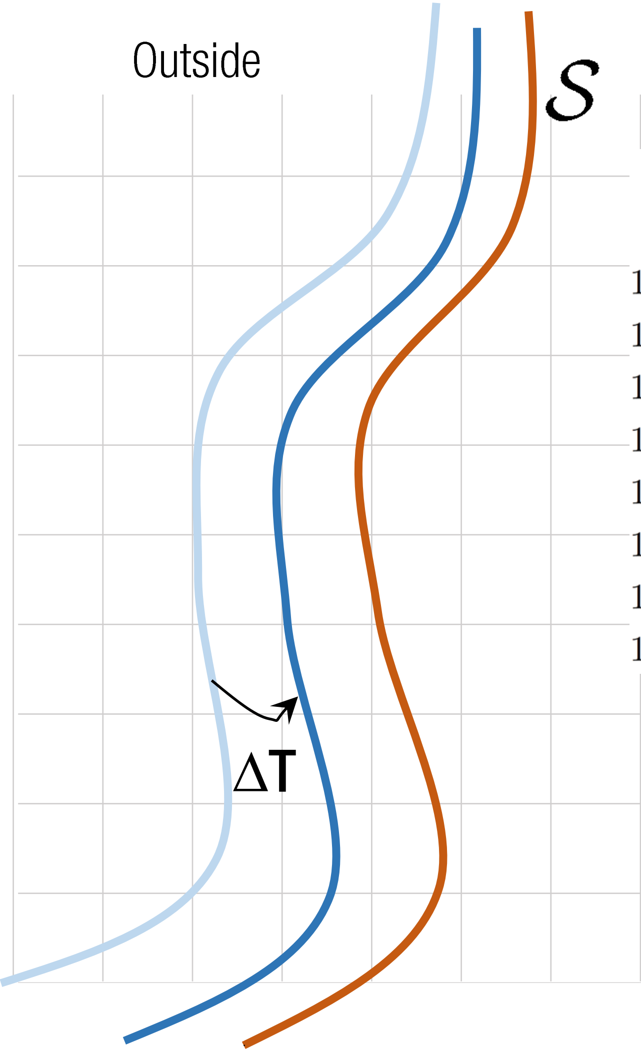
$$\hat{\mathbf{n}}^{\mathsf{T}} \begin{bmatrix} [\mathbf{p}]_\times & \mathbf{I} \end{bmatrix} \begin{bmatrix} \xi \\ \Delta \mathbf{t} \end{bmatrix} = \hat{\mathbf{n}}^{\mathsf{T}}(\hat{\mathbf{p}} - \mathbf{p})$$

Alignment Algorithm

$$\mathbf{P}^1_{i+1} = (\mathbf{R}^1_{i+1}, \mathbf{C}^1_{i+1})$$



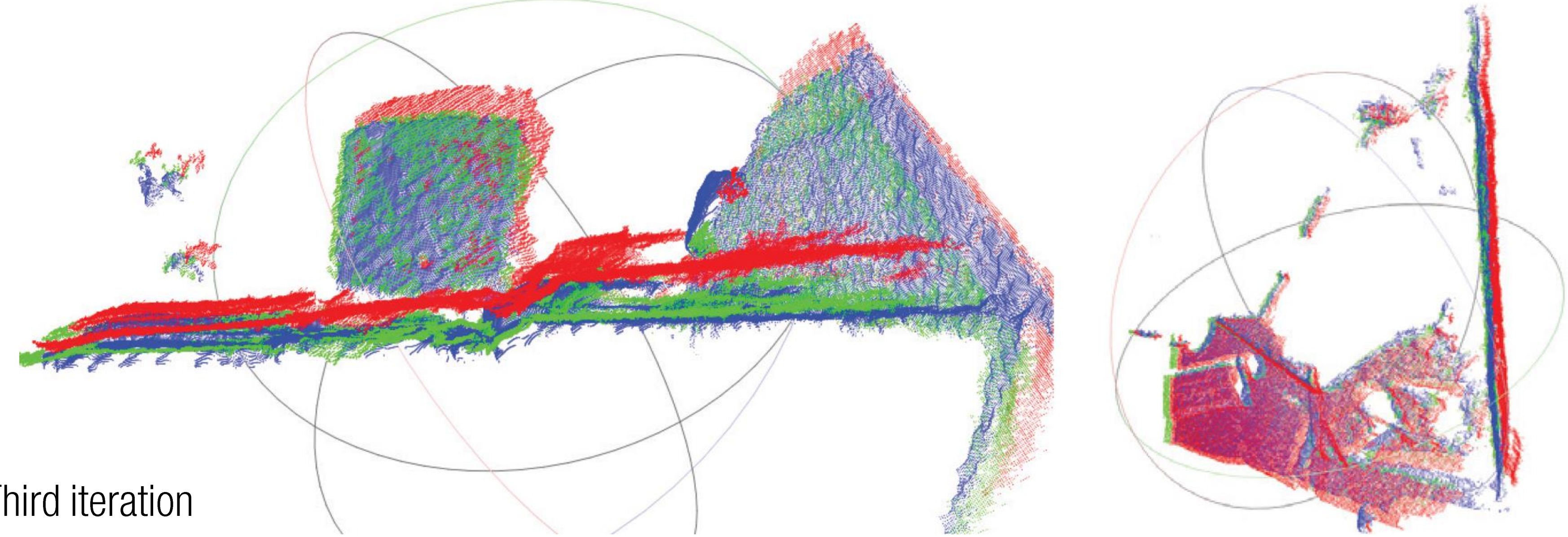
$$\mathbf{P}^2_{i+1} = (\mathbf{R}^2_{i+1}, \mathbf{C}^2_{i+1})$$



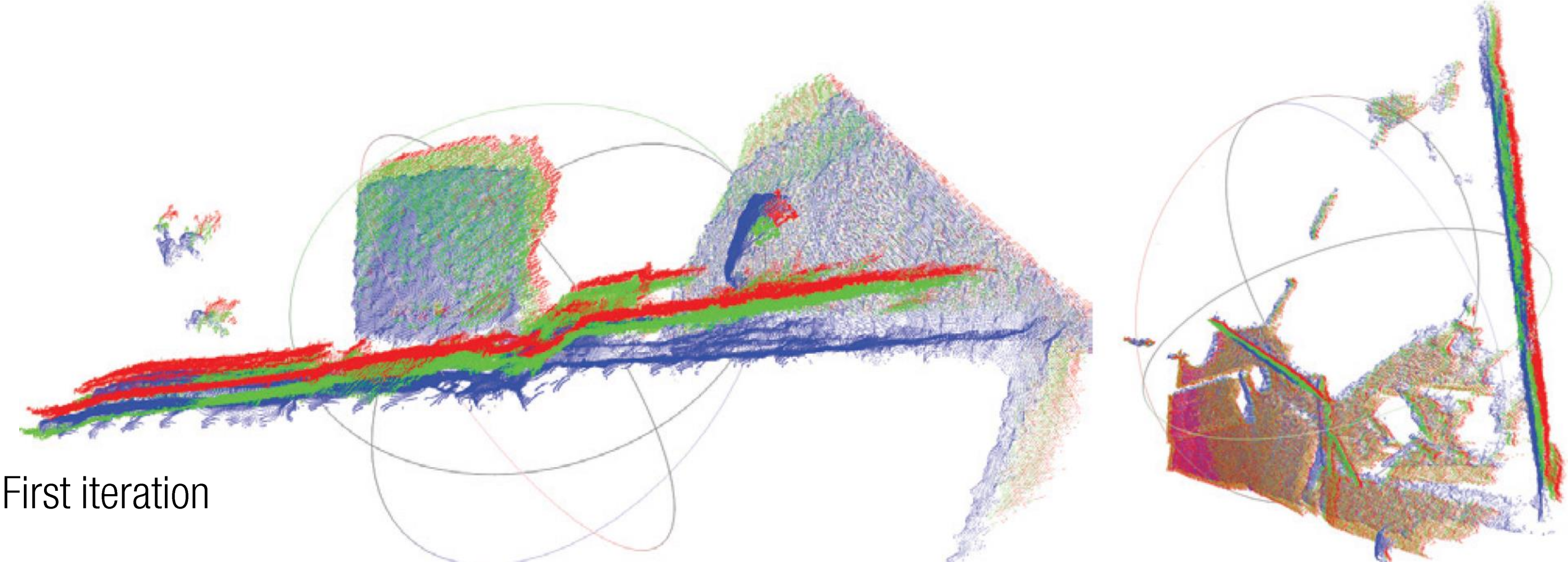

```
Predict points from the TSDF by ray casting
Predict normals from the predicted points
Load the i^th depth image
Compute 3D points and surface normals.
for j^th iteration < n_iter do
    Find correspondences
    Compute ΔT
    Update T_curr and make sure det(R_curr) = 1
end for
Update transformation T = T_curr
```

$$\Delta\mathbf{T} = \begin{bmatrix} \Delta\mathbf{R} & \Delta\mathbf{t} \\ \mathbf{0} & 1 \end{bmatrix} \approx \mathbf{I} + \mathbf{Z} \quad \text{where } \mathbf{Z} = \begin{bmatrix} [\xi]_\times & \Delta\mathbf{t} \\ \mathbf{0} & 0 \end{bmatrix}$$

$$\mathbf{T} = (\Delta\mathbf{T})\mathbf{T}$$

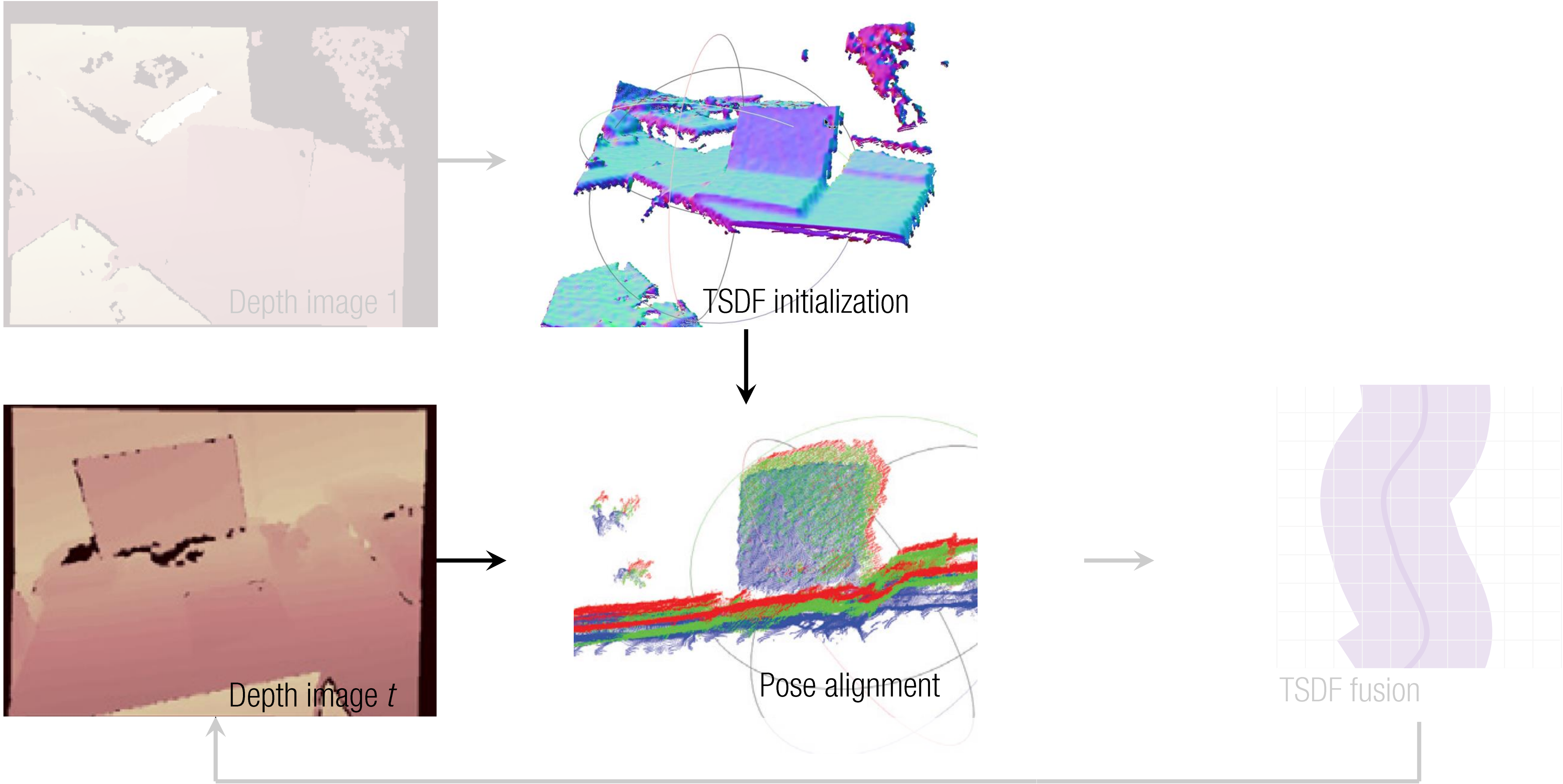


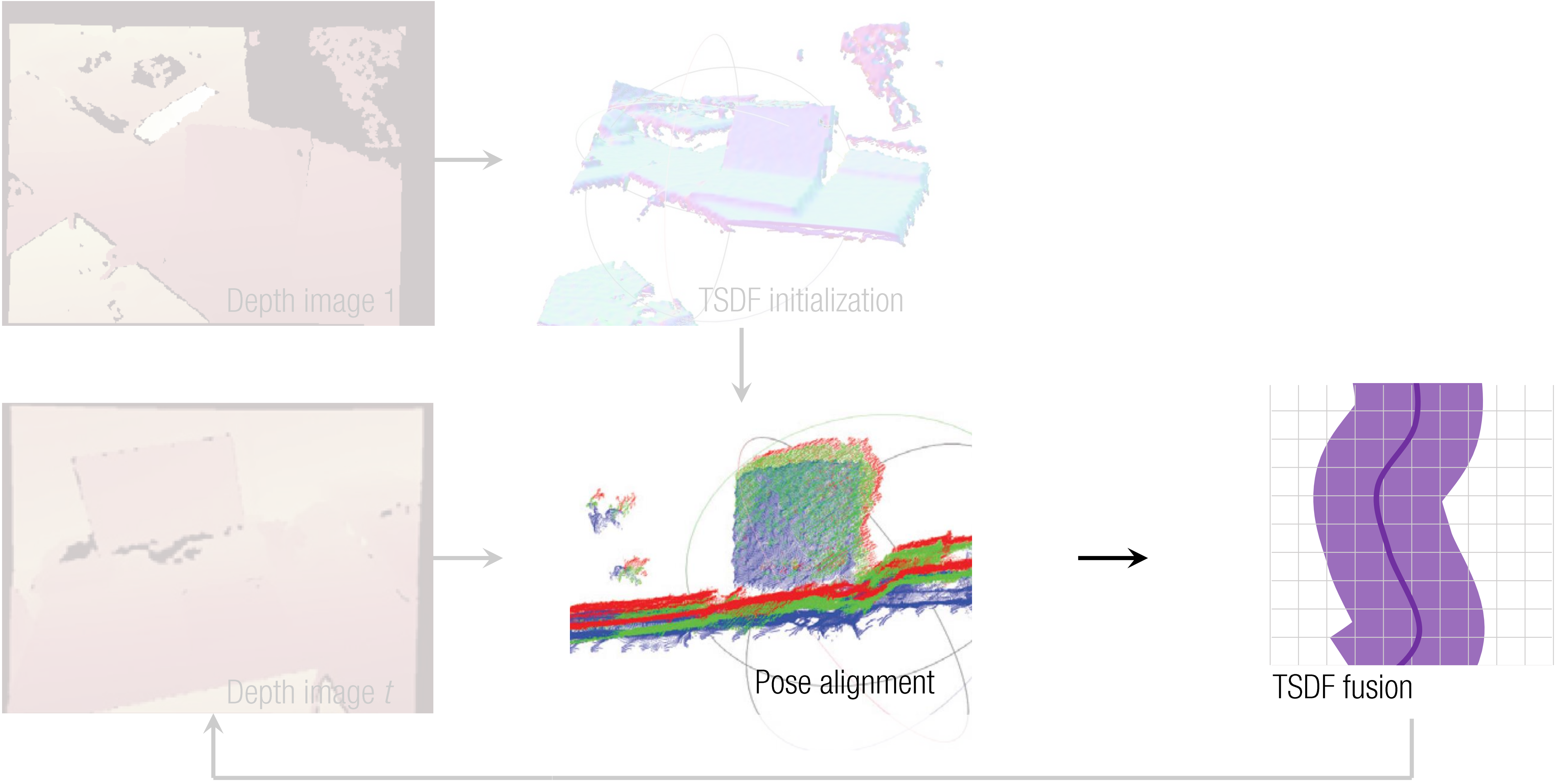
Third iteration



First iteration

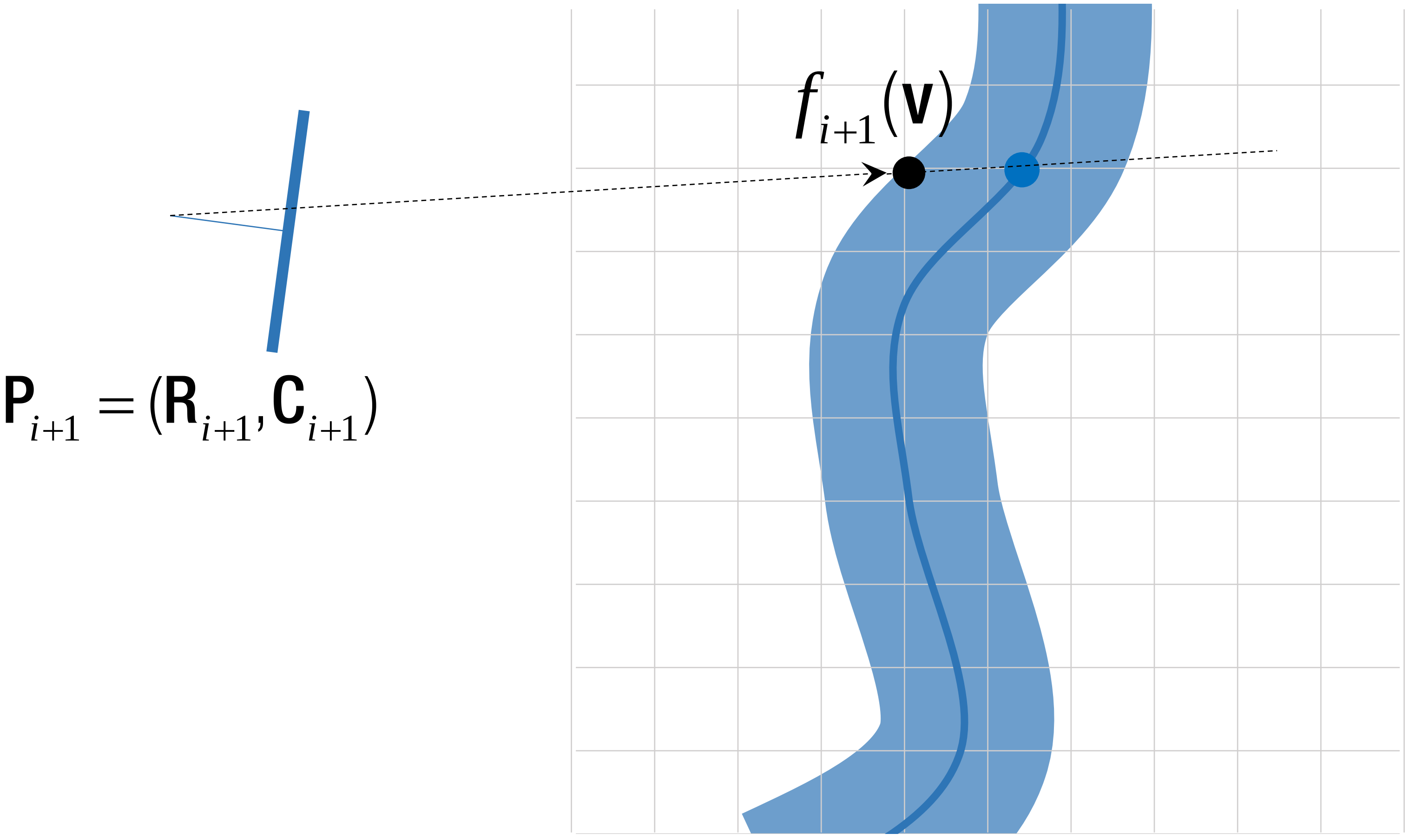
Blue: predicted points
Red: measured points
Green: optimized points


Depth image 1
TSDF initialization
Depth image t
Pose alignment
TSDF fusion



Depth image 1
TSDF initialization
Depth image *t*
Pose alignment
TSDF fusion

TSDF from New View

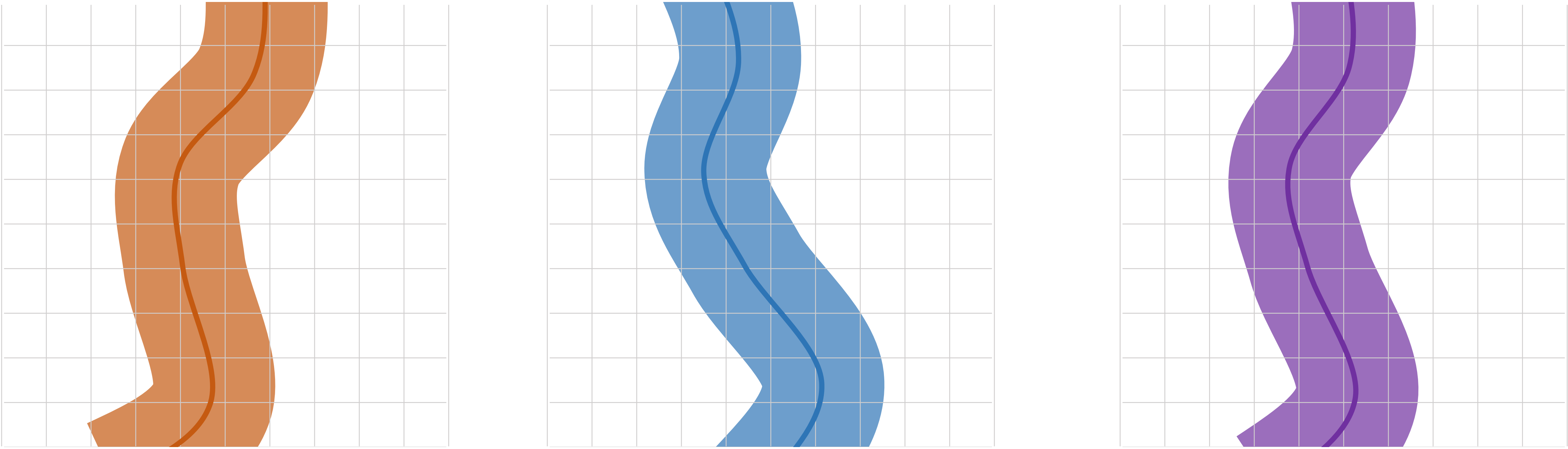


$${}^{w}\mathbf{T}_c = \begin{bmatrix} \mathbf{R} & \mathbf{t} \\ 0 & 1 \end{bmatrix}$$

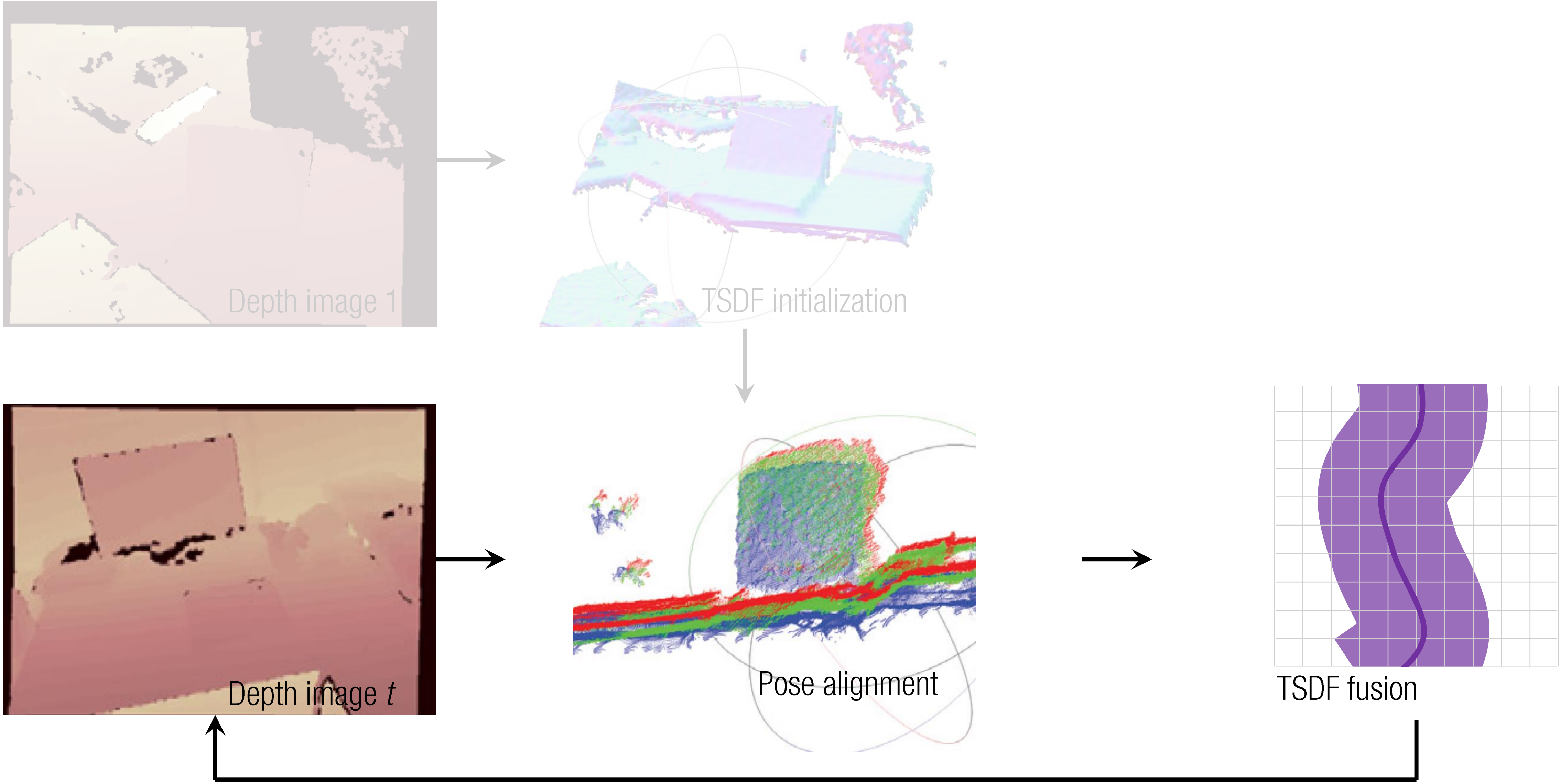
$$\mathbf{p}_w = d\mathbf{R}_{i+1}^{\top}\mathbf{K}^{-1}\begin{bmatrix} \hat{\mathbf{u}} \\ 1 \end{bmatrix} + \mathbf{C}_{i+1}$$

$$= \mathbf{R}\mathbf{p} + \mathbf{t}$$

TSDF Fusion



$$w f_{1:i}(\mathbf{v}) \quad + \quad (1-w)\, f_{i+1}(\mathbf{v}) \quad = \quad f_{1:i+1}(\mathbf{v})$$


Depth image 1
TSDF initialization
Depth image t
Pose alignment
TSDF fusion