3D RECONSTRUCTION

Real-time 3D reconstruction in-door scene
using moving Kinect

Jiakai Zhang, Hao Liu, YuXu

Reconstruction method

- Reconstruction from images
- Reconstruction from video

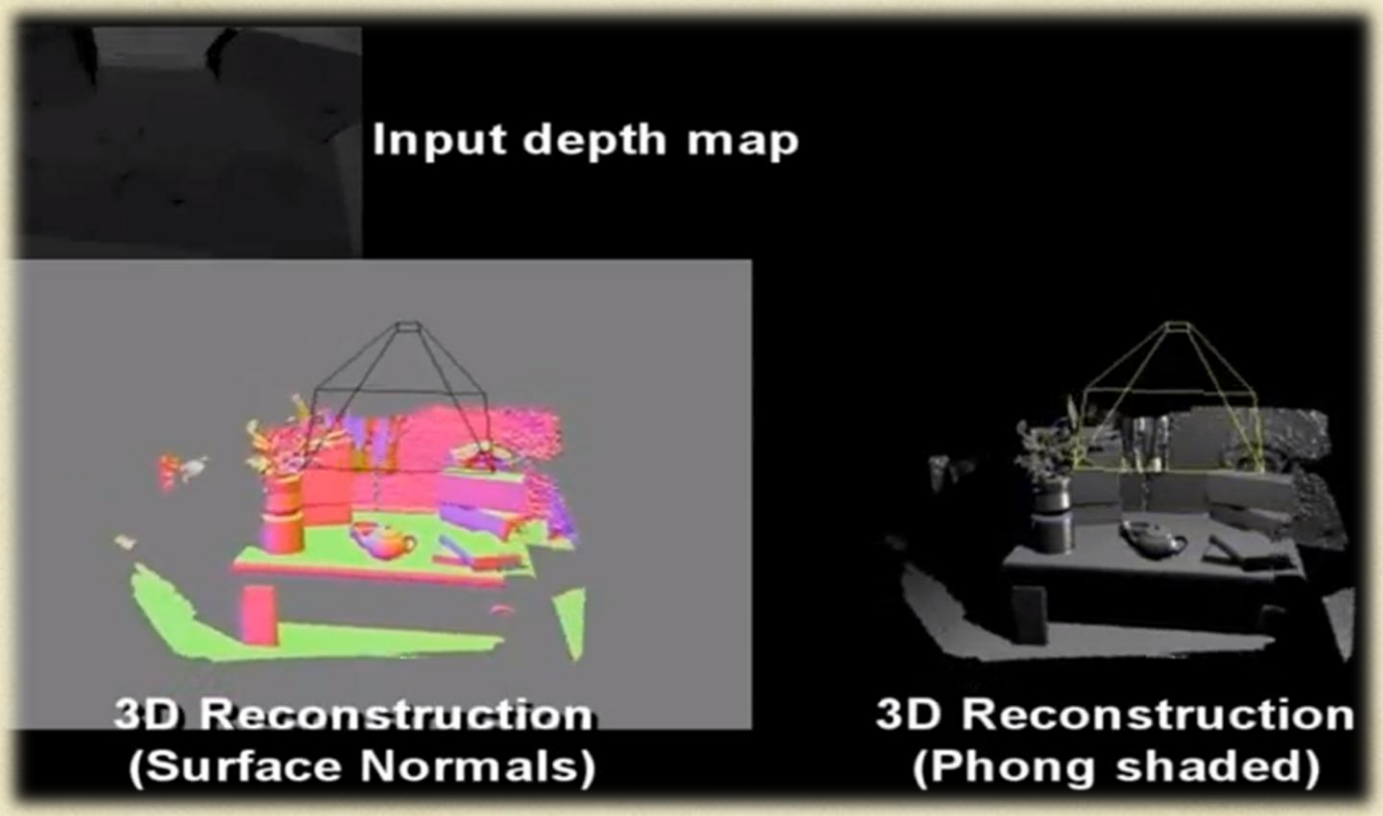
Using Kinect

Raw Depth Image

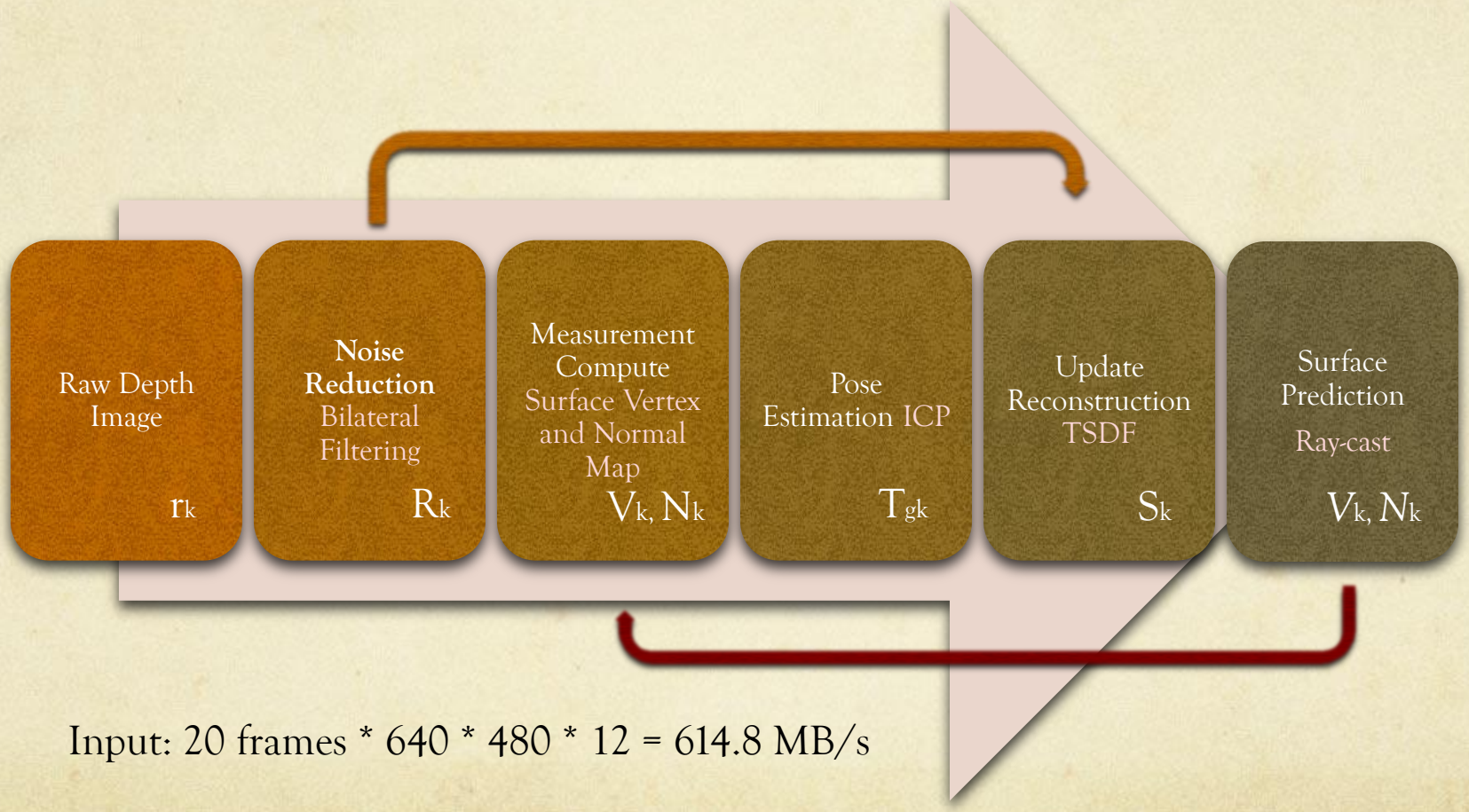
- Infrared laser projector
- Monochrome CMOS sensor

Demo
Kinect Raw data

Real-time Reconstruction

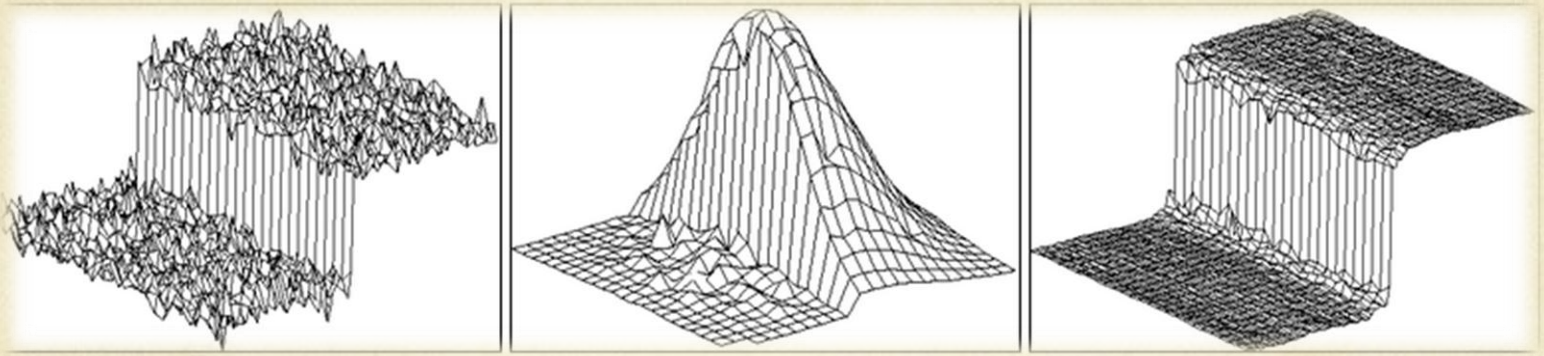


Pipeline



Input: 20 frames * 640 * 480 * 12 = 614.8 MB/s

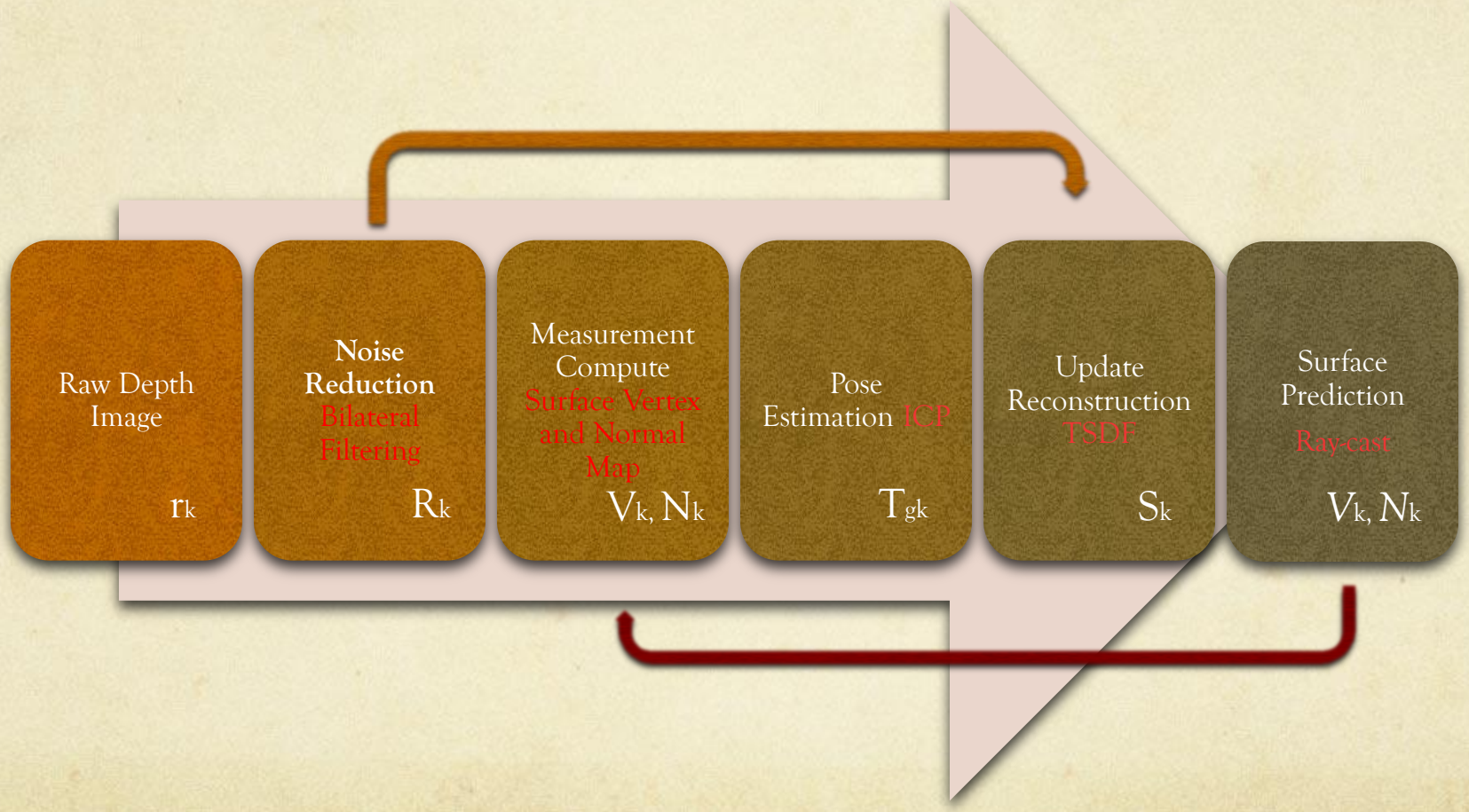
Bilateral Filtering



- $$\mathbf{h}(\mathbf{x}) = k^{-1} \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \mathbf{f}(\xi) c(\xi - \mathbf{x}) s(\mathbf{f}(\xi) - \mathbf{f}(\mathbf{x})) d\xi$$

Demo

Pipeline


Raw Depth Image
r_k
Noise Reduction
Bilateral Filtering
R_k
Measurement Compute
Surface Vertex and Normal Map
V_k, N_k
Pose Estimation ICP
T_{gk}
Update Reconstruction
TSDF
S_k
Surface Prediction
Ray-cast
V_k, N_k

ICP 3D shape alignment

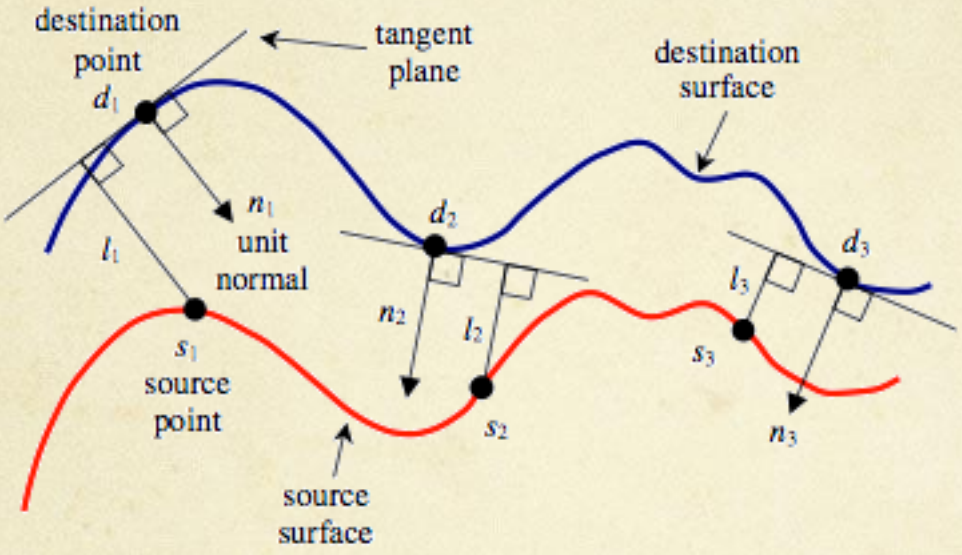


***Figure 1**: Point-to-plane error between two surfaces.*

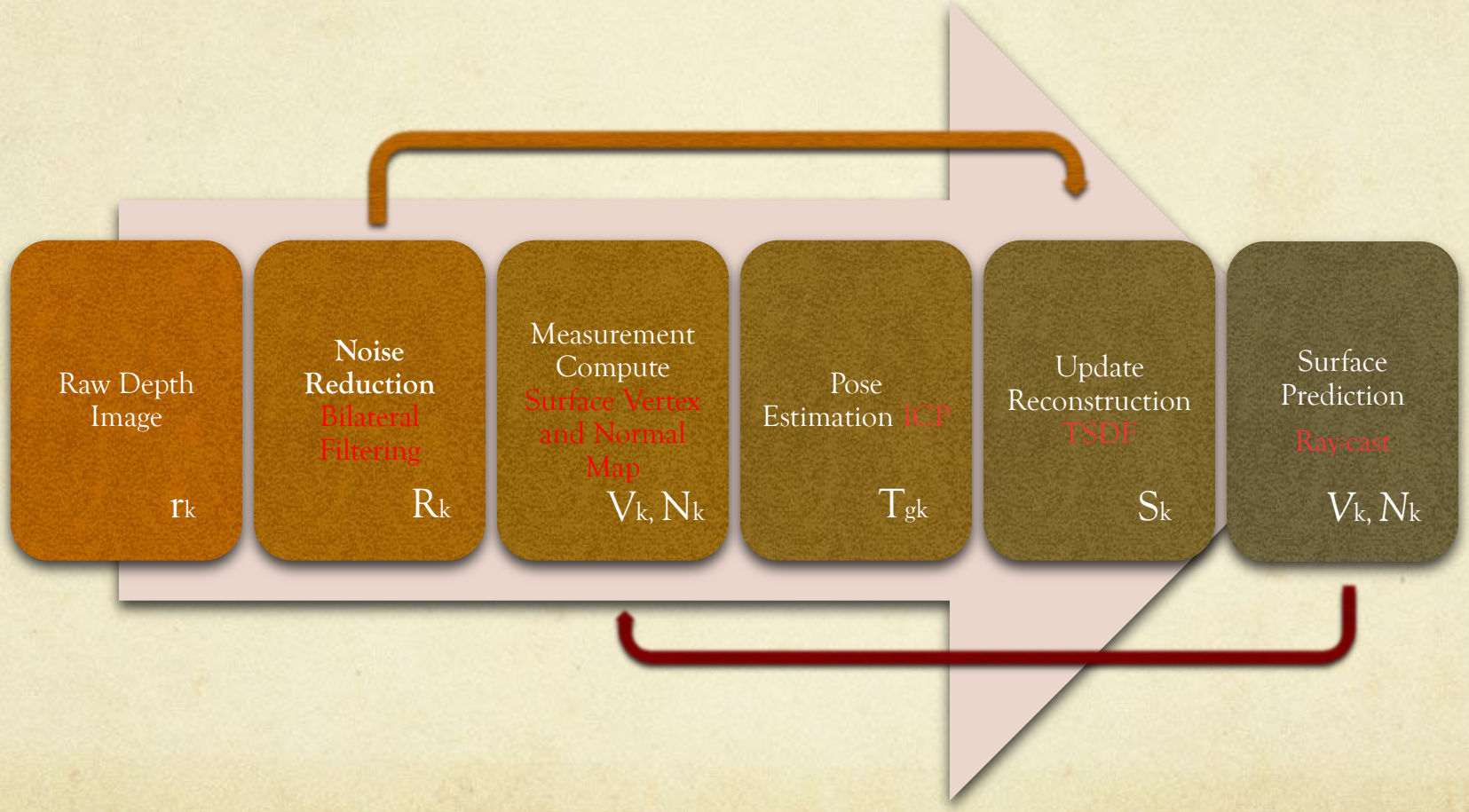
$$\hat{\mathbf{M}} = \mathbf{T}(t_x, t_y, t_z) \cdot \hat{\mathbf{R}}(\alpha, \beta, \gamma) = \begin{pmatrix} 1 & -\gamma & \beta & t_x \\ \gamma & 1 & -\alpha & t_y \\ -\beta & \alpha & 1 & t_z \\ 0 & 0 & 0 & 1 \end{pmatrix}.$$

- $\mathbf{M}_{opt} = \arg\min_{\mathbf{M}} \sum_i ((\mathbf{M} \cdot \mathbf{s}_i - \mathbf{d}_i) \bullet \mathbf{n}_i)$ $\quad \mathbf{M} = \mathbf{T}(t_x, t_y, t_z) \cdot \mathbf{R}(\alpha, \beta, \gamma)$
- $\min_{\hat{\mathbf{M}}} \sum_i ((\hat{\mathbf{M}} \cdot \mathbf{s}_i - \mathbf{d}_i) \bullet \mathbf{n}_i)^2 = \min_{\mathbf{x}} |\mathbf{A}\mathbf{x} - \mathbf{b}|^2.$ ➡ SVD

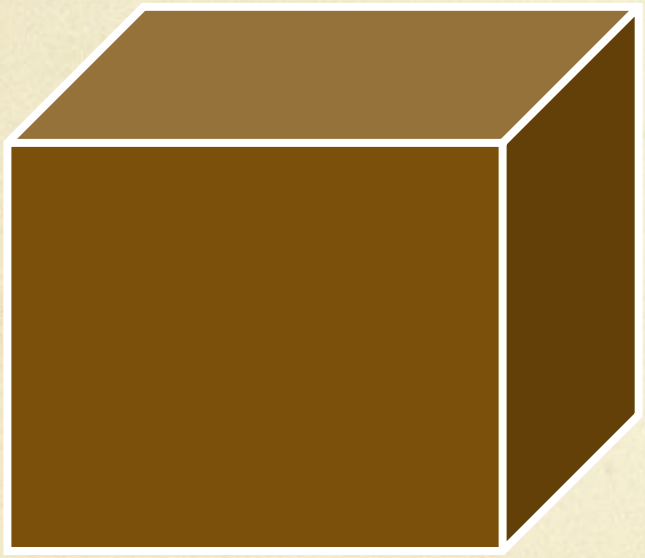
ICP 3D shape alignment

Demo

Pipeline


Raw Depth Image
r_k
Noise Reduction
Bilateral Filtering
R_k
Measurement Compute
Surface Vertex and Normal Map
V_k, N_k
Pose Estimation ICP
T_{gk}
Update Reconstruction
TSDF
S_k
Surface Prediction
Ray-cast
V_k, N_k

TSDF

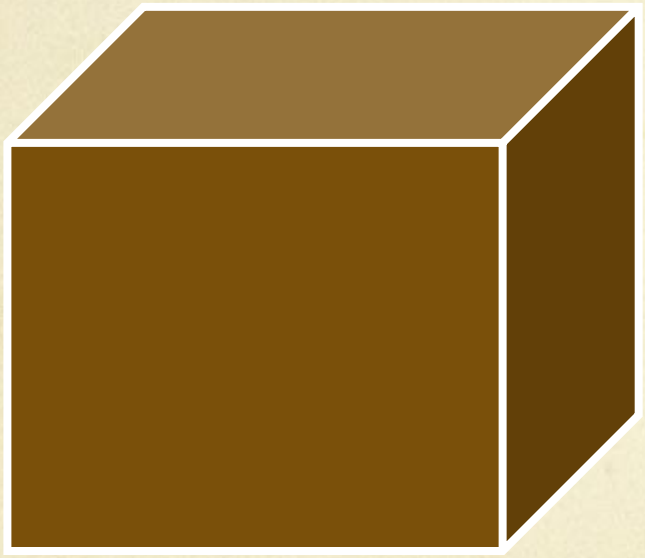


T S D F

- Signed Distance Function

The value in the cube corresponds to the signed distance to the closest zero crossing(surface).

TSDF

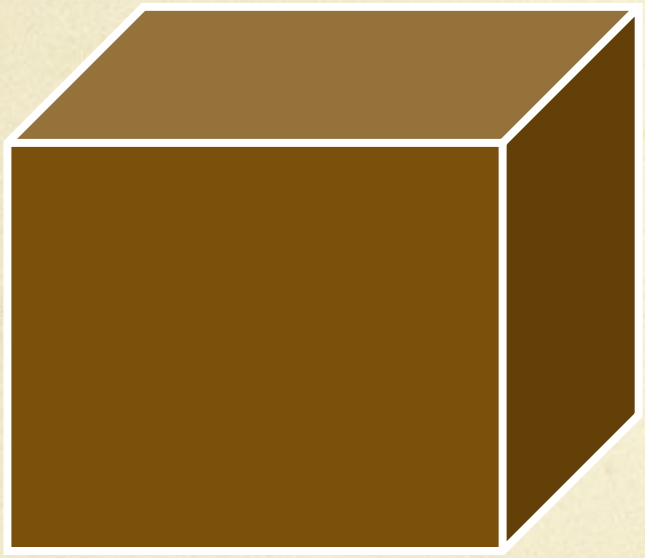


T S D F

- Signed Distance Function
- Truncated Signed Distance Function

$$
\begin{aligned}
F_{R_k}(\mathbf{p}) &= \Psi\left(\lambda^{-1}\|(\mathbf{t}_{g,k} - \mathbf{p}\|_2 - R_k(\mathbf{x})\right), \\
\lambda &= \|K^{-1}\dot{\mathbf{x}}\|_2, \\
\mathbf{x} &= \left\lfloor \pi\left(K T_{g,k}^{-1}\mathbf{p}\right)\right\rfloor, \\
\Psi(\eta) &= \begin{cases} \min\left(1, \frac{\eta}{\mu}\right)\operatorname{sgn}(\eta) & \text{iff } \eta \geq -\mu \\ \textit{null} & \textit{otherwise} \end{cases}
\end{aligned}
$$

TSDF



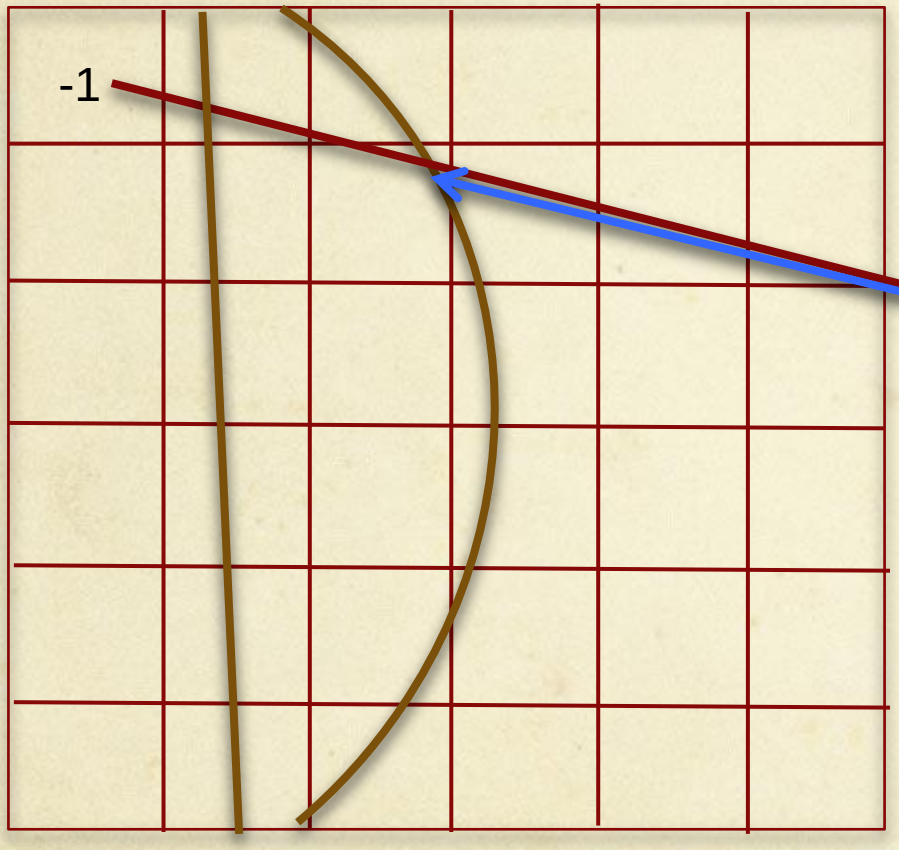
T S D F

- Signed Distance Function
- Truncated Signed Distance Function
- Integrate the cubes from different position.

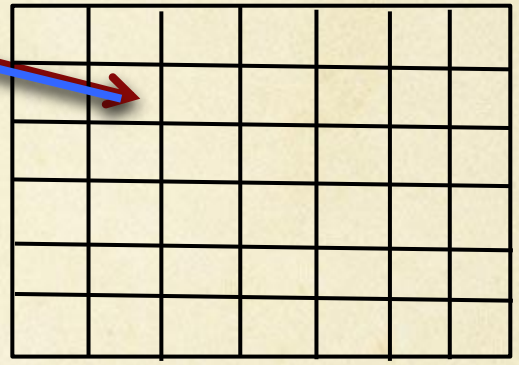
$$F_k(\mathbf{p}) = \frac{W_{k-1}(\mathbf{p})F_{k-1}(\mathbf{p}) + W_{R_k}(\mathbf{p})F_{R_k}(\mathbf{p})}{W_{k-1}(\mathbf{p}) + W_{R_k}(\mathbf{p})}$$

$$W_k(\mathbf{p}) = W_{k-1}(\mathbf{p}) + W_{R_k}(\mathbf{p})$$

TSDF

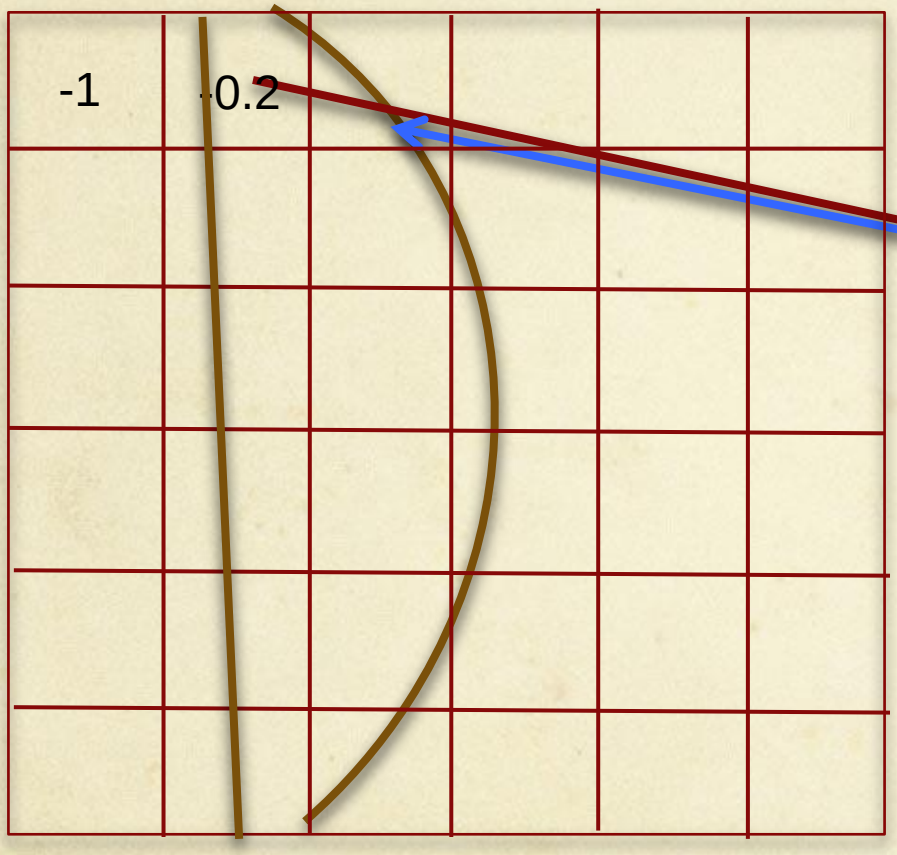


2D Sample

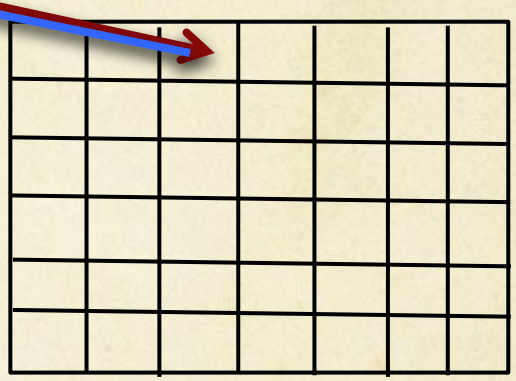


Depth Map from Kinect

TSDF

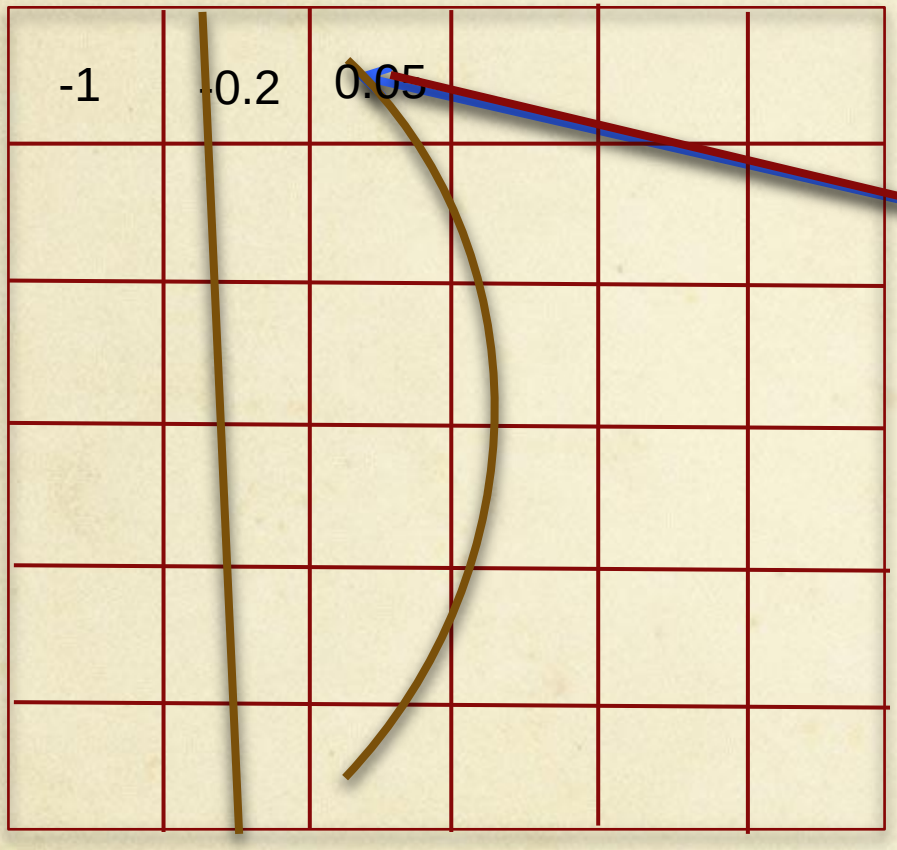


2D Sample

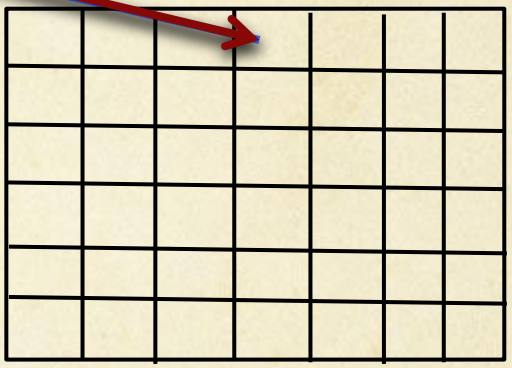


Depth Map from Kinect

TSDF

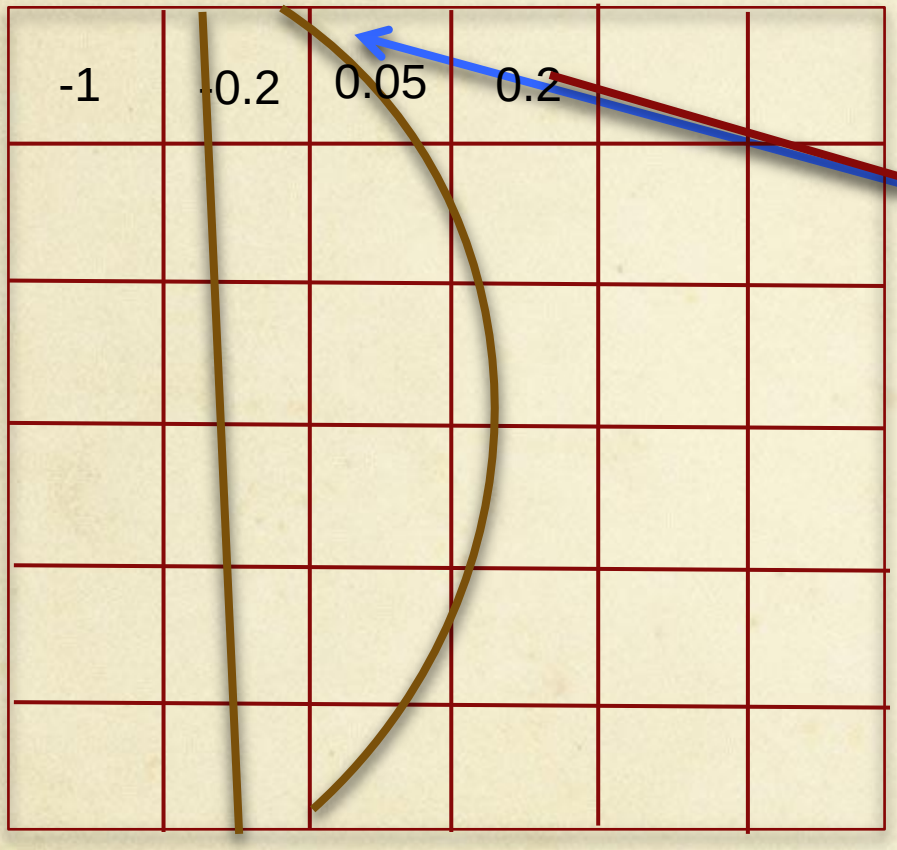


2D Sample

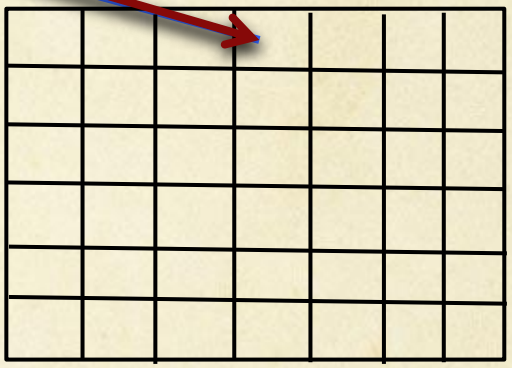


Depth Map from Kinect

TSDF

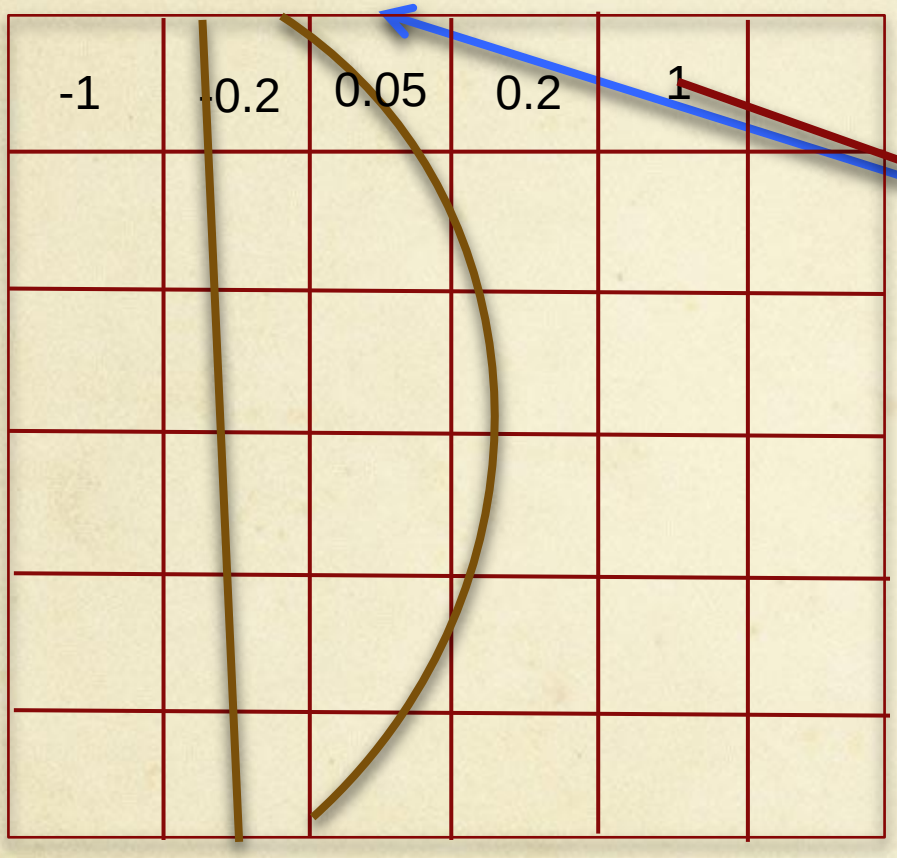


2D Sample

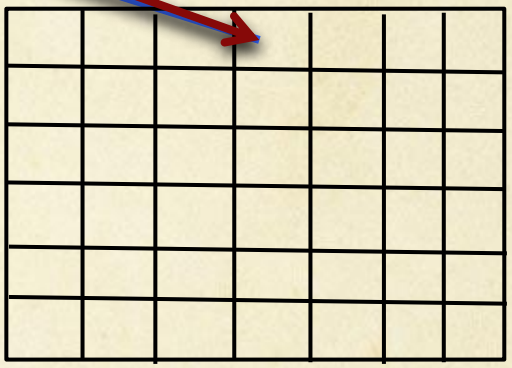


Depth Map from Kinect

TSDF

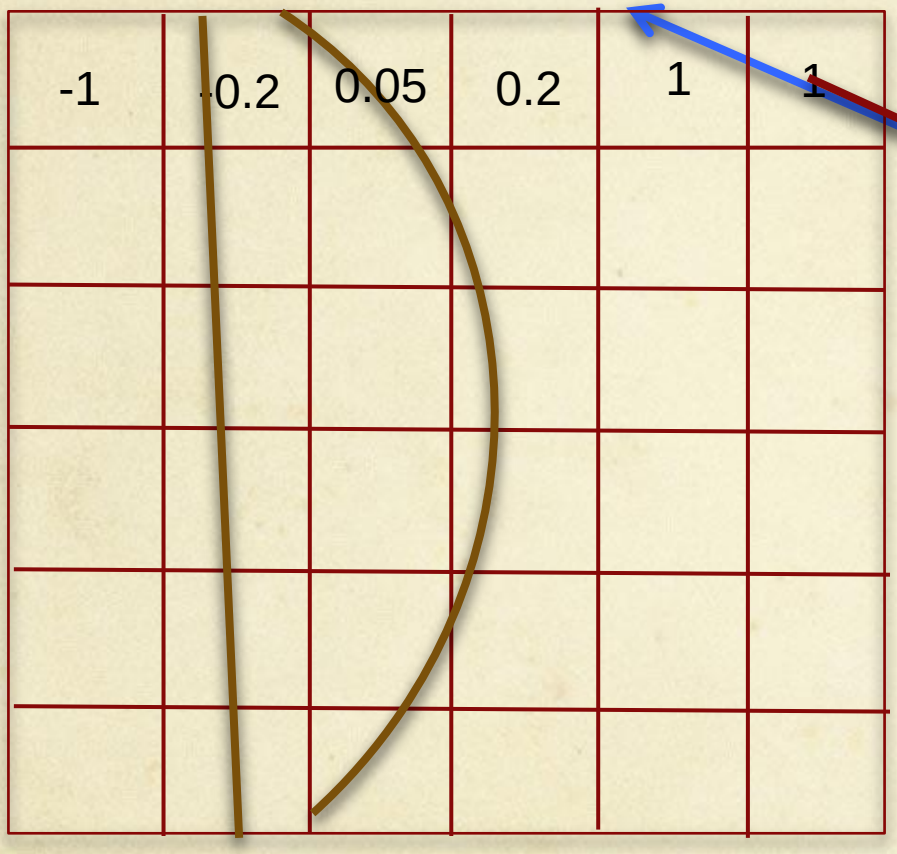


2D Sample

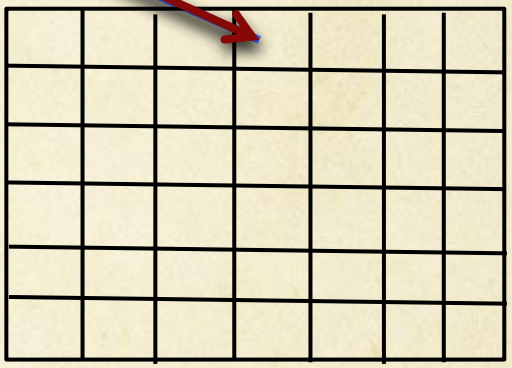


Depth Map from Kinect

TSDF

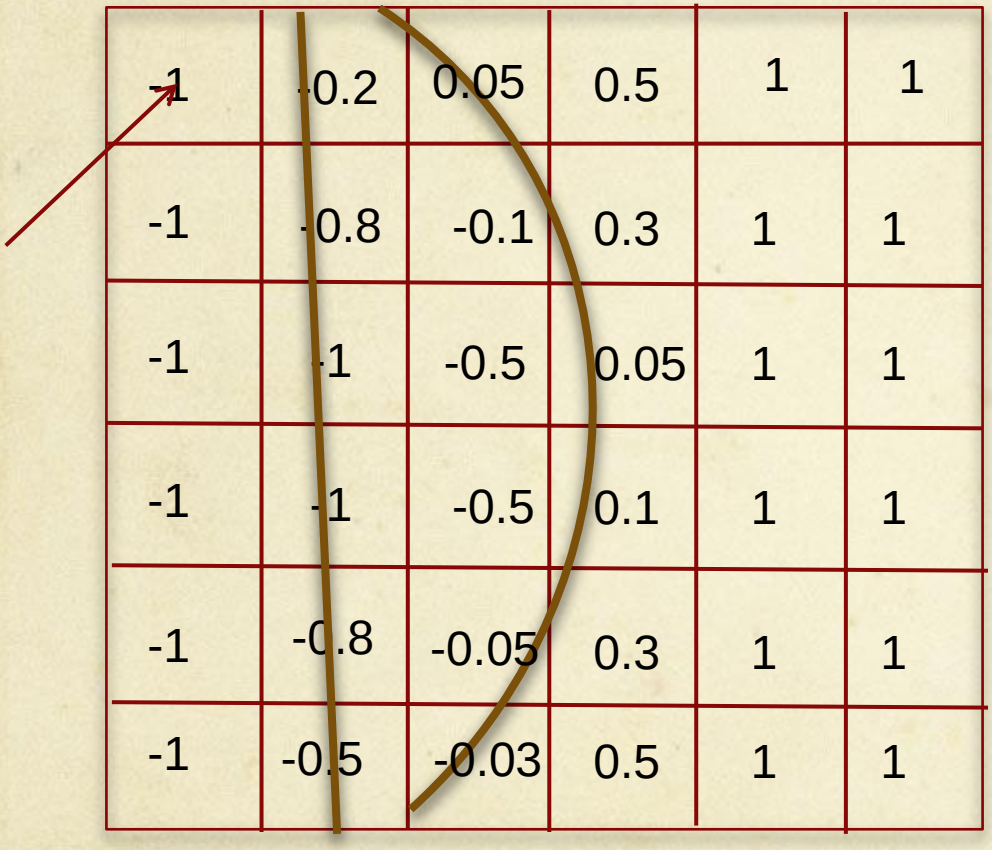


2D Sample



Depth Map from Kinect

TSDF

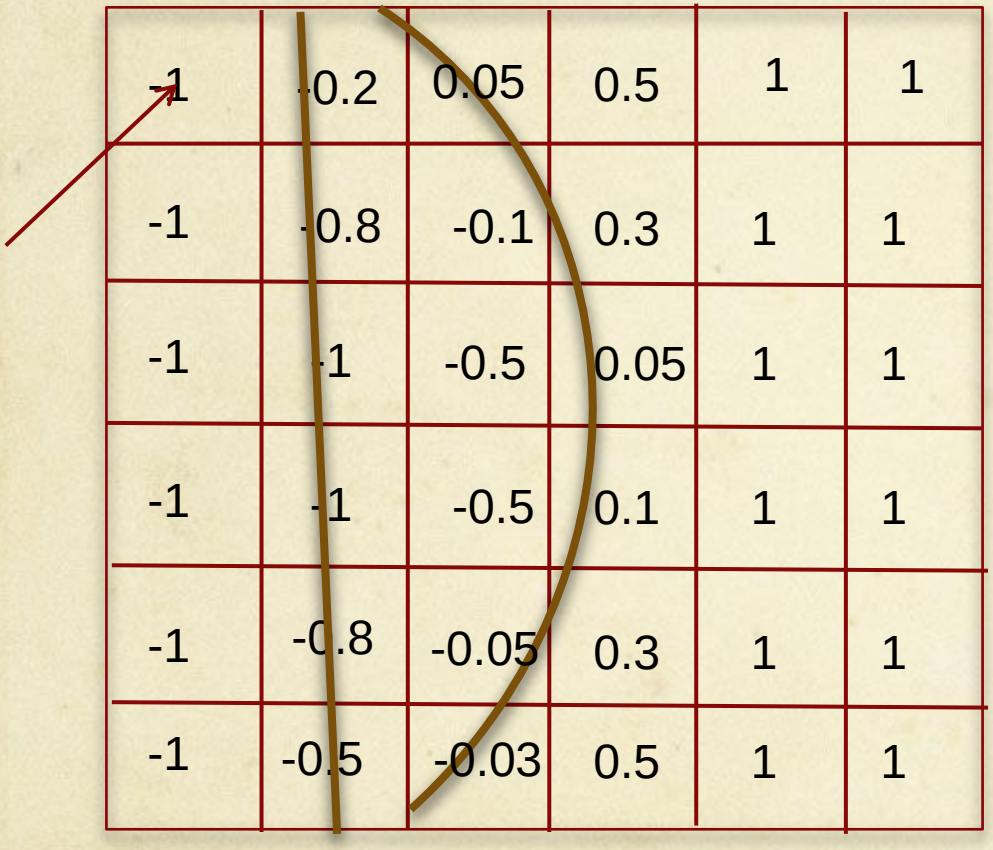


-1	-0.2	0.05	0.5	1	1
-1	-0.8	-0.1	0.3	1	1
-1	-1	-0.5	0.05	1	1
-1	-1	-0.5	0.1	1	1
-1	-0.8	-0.05	0.3	1	1
-1	-0.5	-0.03	0.5	1	1

Integration

- We have depth maps from different camera positions, how can we integrate them together ?
- Integration? or update?
- Weighted? or add up?
- What makes integration possible ?

TSDF

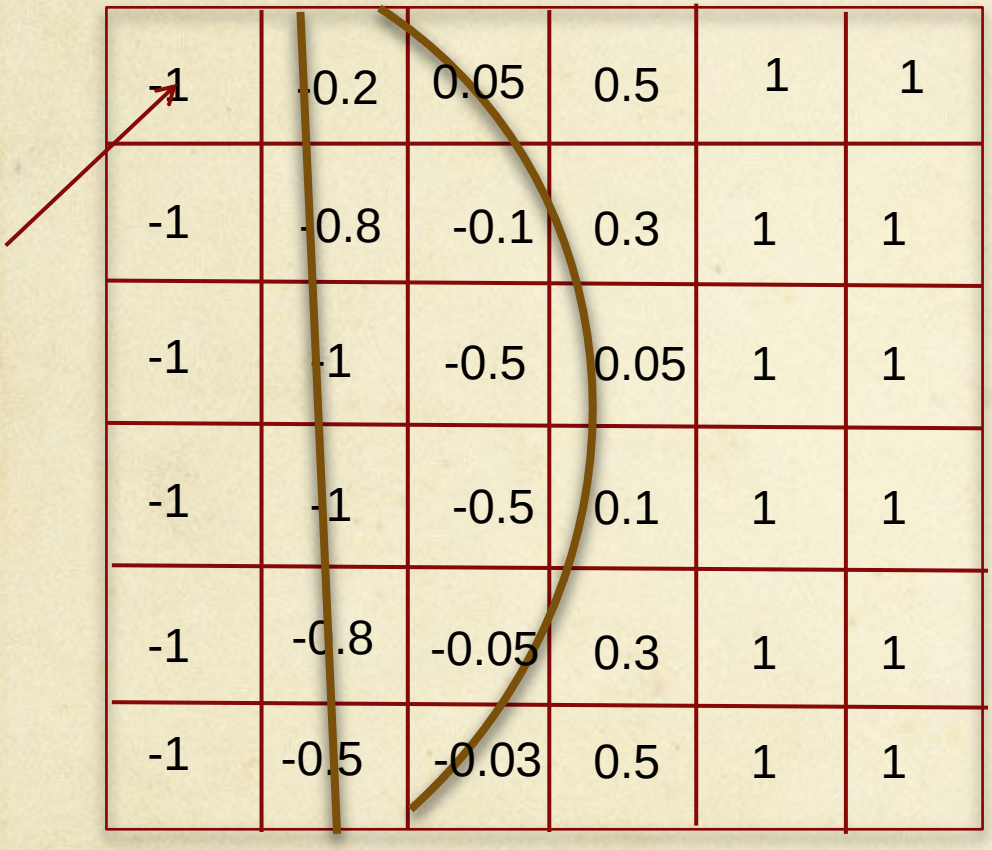


-1	-0.2	0.05	0.5	1	1
-1	-0.8	-0.1	0.3	1	1
-1	-1	-0.5	0.05	1	1
-1	-1	-0.5	0.1	1	1
-1	-0.8	-0.05	0.3	1	1
-1	-0.5	-0.03	0.5	1	1

Integration

- We have depth maps from different camera positions, how can we integrate them together ?
- Integration? or update?
- Weighted? or add up?
- What makes integration possible ?

TSDF

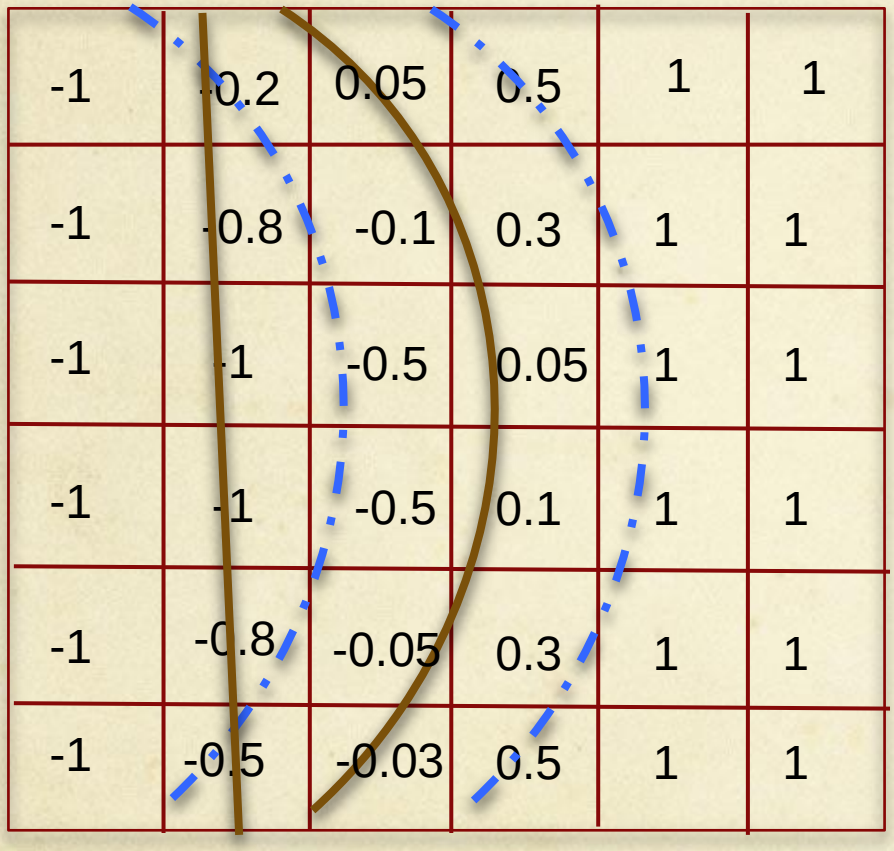


-1	-0.2	0.05	0.5	1	1
-1	-0.8	-0.1	0.3	1	1
-1	-1	-0.5	0.05	1	1
-1	-1	-0.5	0.1	1	1
-1	-0.8	-0.05	0.3	1	1
-1	-0.5	-0.03	0.5	1	1

Integration

- We have depth maps from different camera positions, how can we integrate them together ?
- Integration? or update?
- Weighted? or add up?
- What makes integration possible ?

TSDF

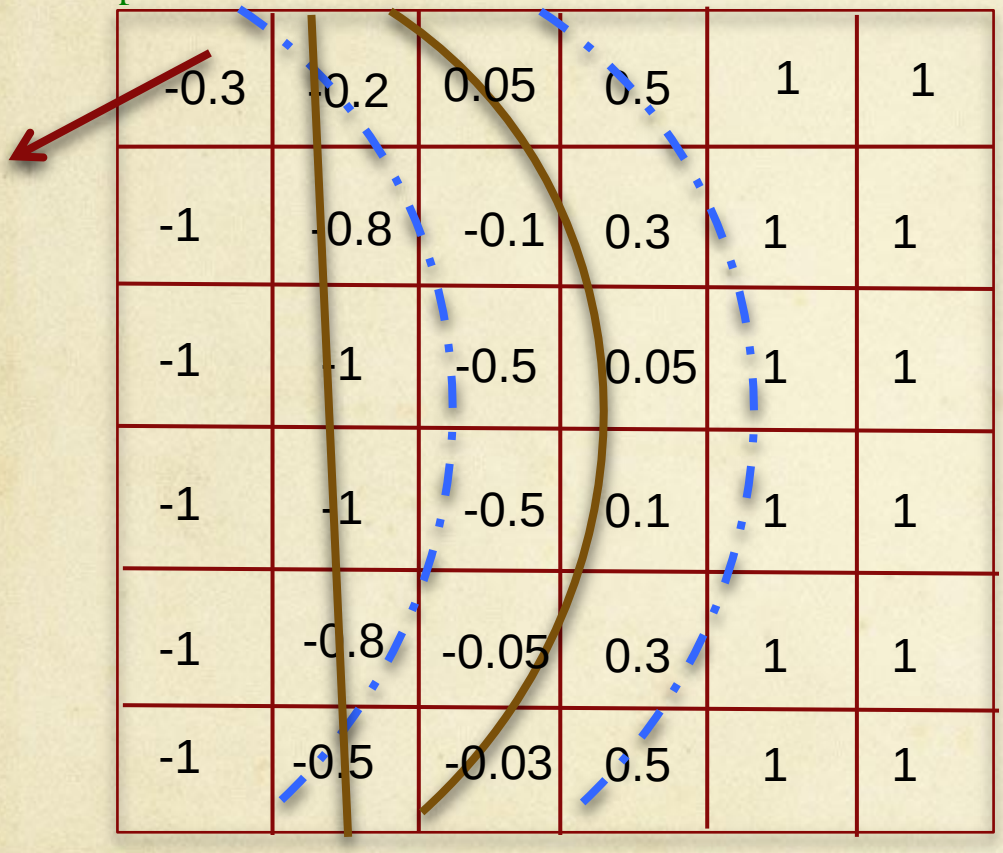


TRUNCATION

- To get the surface behind the surface. The camera is moving!
- Only part of distance data is needed, so we can truncate the distance.

TSDF

1 time update !

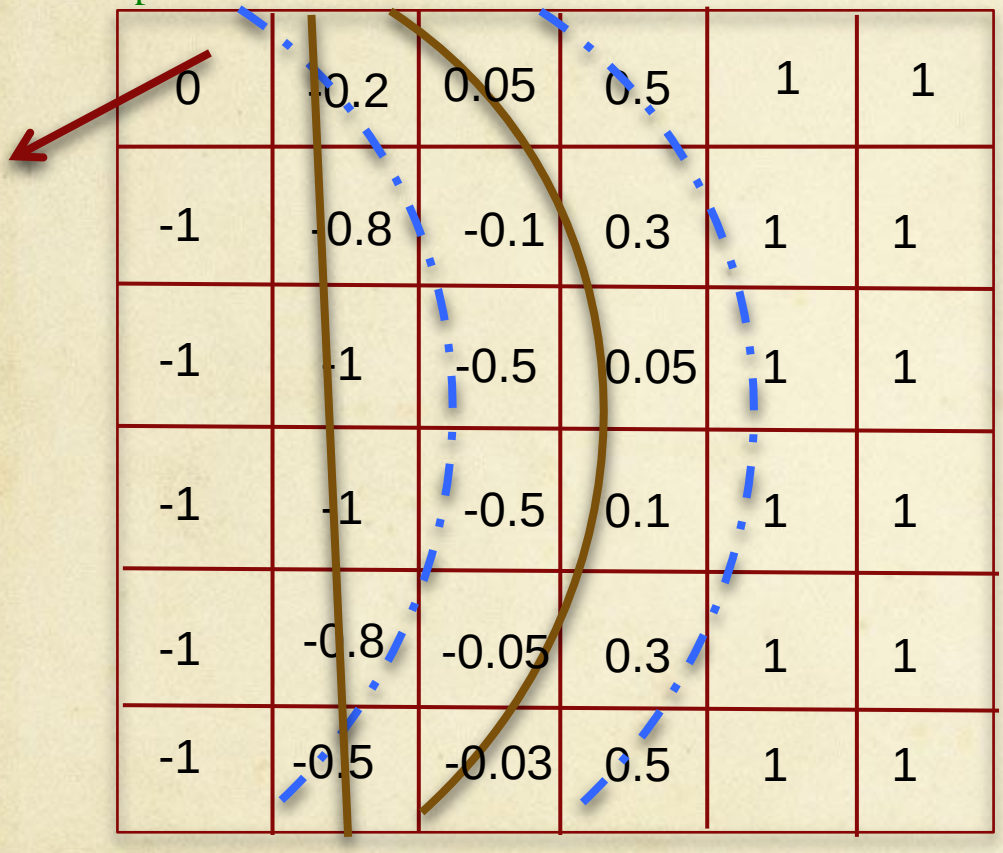


TRUNCATION

- To get the surface behind the surface. The camera is moving!
- Only part of distance data is needed to represent the object, so we can truncate the distance.

TSDF

2 times update !

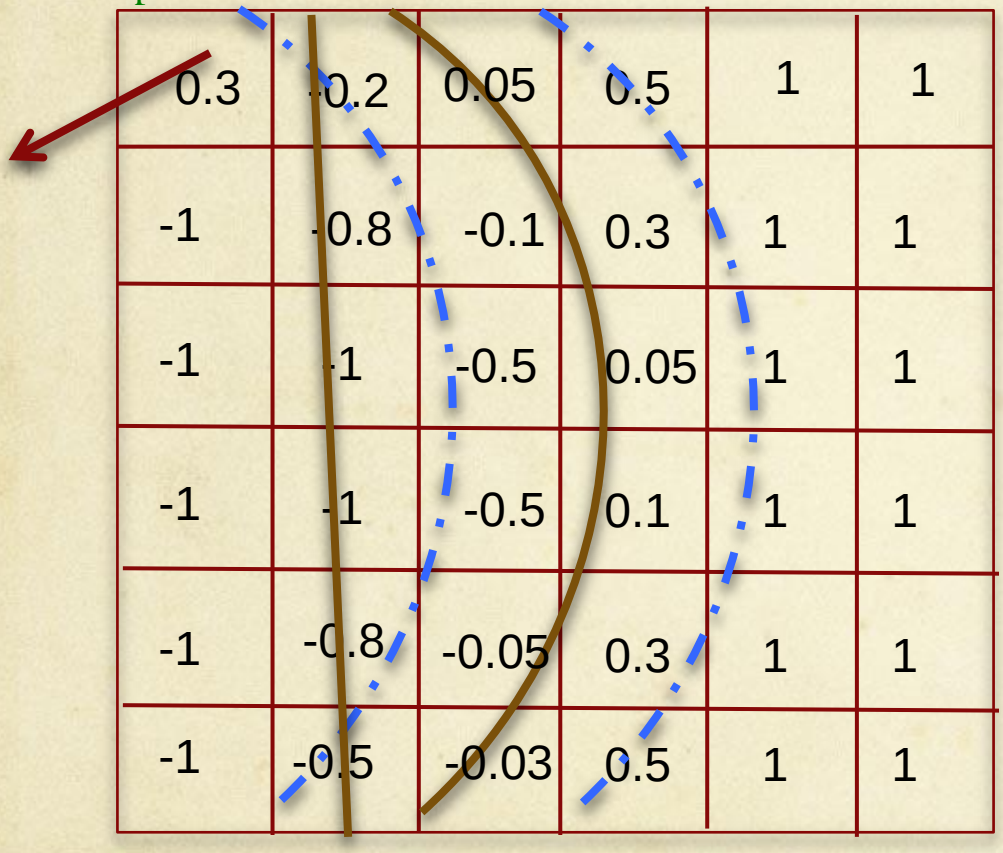


TRUNCATION

- To get the surface behind the surface. The camera is moving!
- Only part of distance data is needed to represent the object, so we can truncate the distance.

TSDF

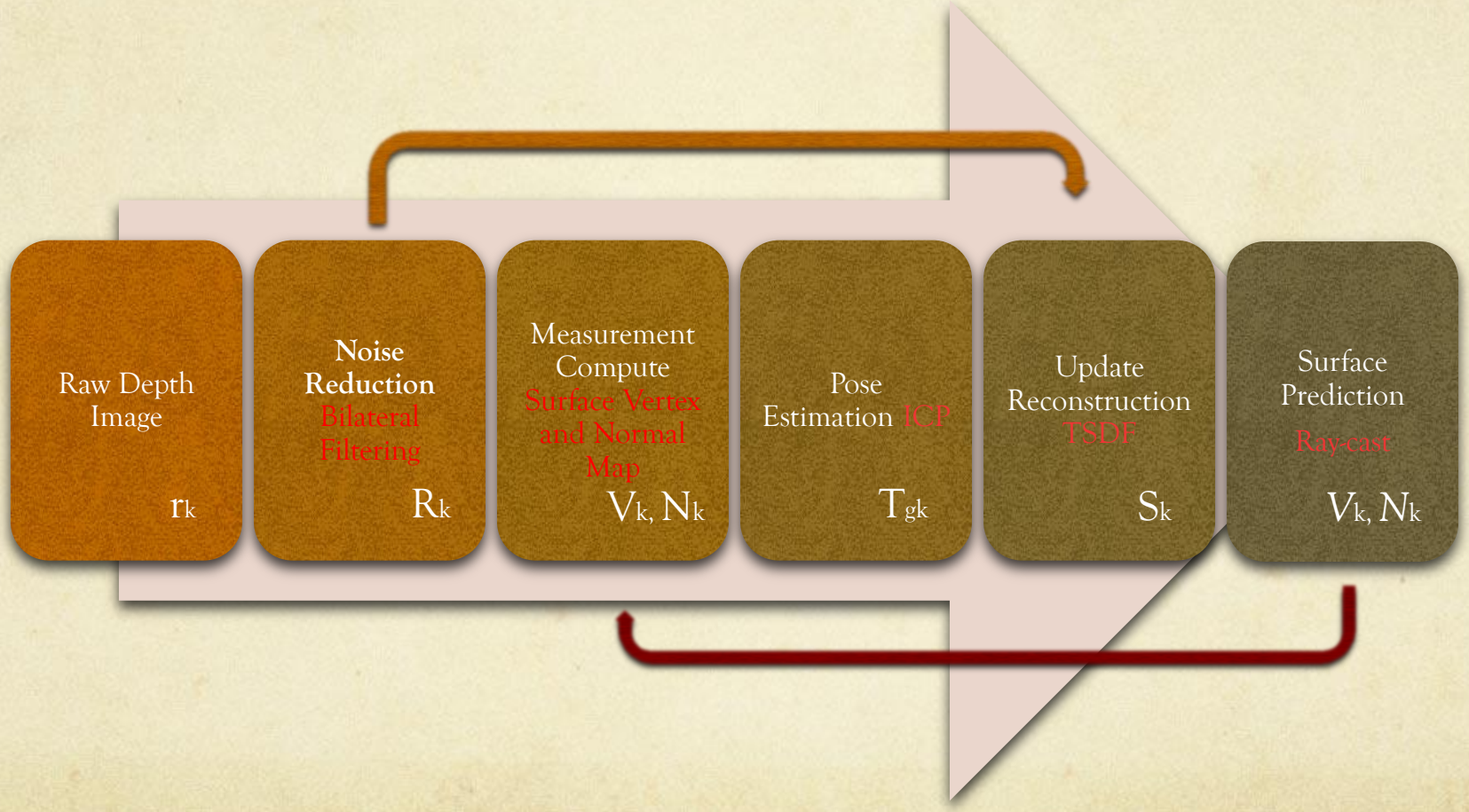
3 times update !



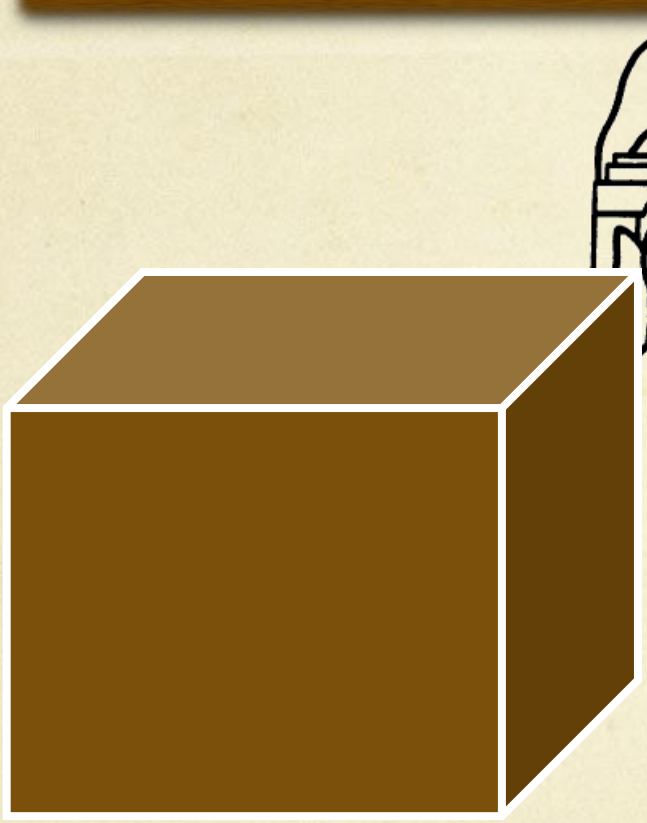
TRUNCATION

- To get the surface behind the surface. The camera is moving!
- Only part of distance data is needed to represent the object, so we can truncate the distance.

Pipeline


Raw Depth Image
r_k
Noise Reduction
Bilateral Filtering
R_k
Measurement Compute
Surface Vertex and Normal Map
V_k, N_k
Pose Estimation ICP
T_{gk}
Update Reconstruction
TSDF
S_k
Surface Prediction
Ray-cast
V_k, N_k

RAY CASTING

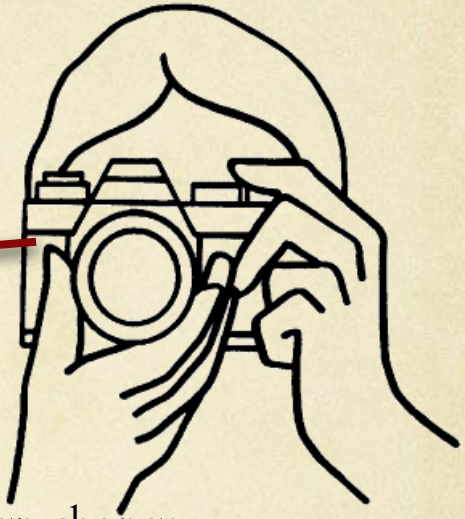


RAY CASTING

- Cast only, no chasing.
- Transfer the TSDF cube in to some thing the computer can understand, Vertex fusion.
- Take a photo using X-ray.

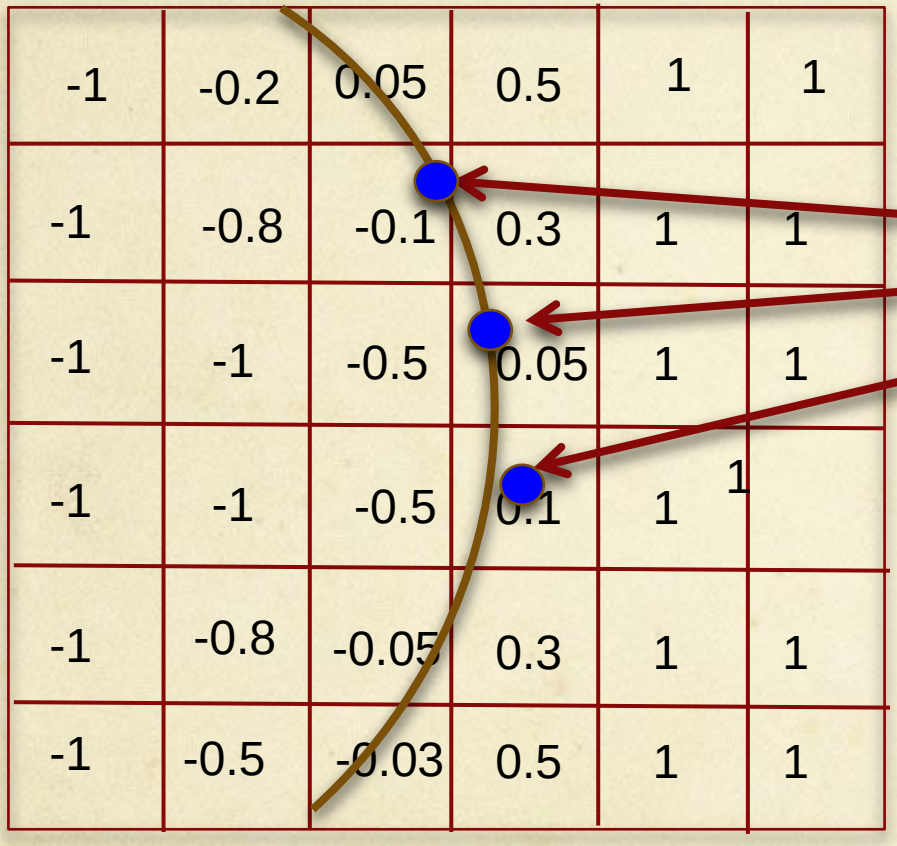
RAY CASTING

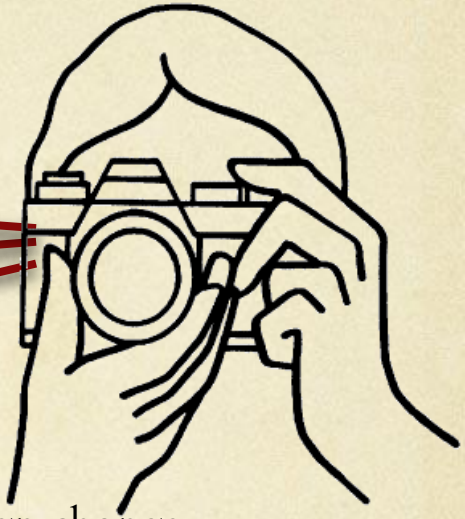
-1	-0.2	0.05	0.5	1	1
-1	-0.8	-0.1	0.3	1	1
-1	-1	-0.5	0.05	1	1
-1	-1	-0.5	0.1	1	1
-1	-0.8	-0.05	0.3	1	1
-1	-0.5	-0.03	0.5	1	1



- Detect the sign change.
- Two scales search
- Linear regression

RAY CASTING





- Detect the sign change.
- Two scales search
- Linear regression
- Normal Vectors

Real-time Reconstruction

Demo

Reference

[1] KinectFusion: Real-Time Dense Surface Mapping and Tracking. *Microsoft Research*

[2] B. Curless and M. Levoy. A volumetric method for building complex models from range images.

[3] M. Harris, S. Sengupta, and J. D. Owens. Parallel prefix sum (scan) with CUDA. In H. Nguyen, editor, GPU Gems 3, chapter 39, pages 851–876. Addison Wesley, August 2007. 3.5

[4] C. Tomasi and R. Manduchi. Bilateral filtering for gray and color images. In Proceedings of the ICCV, 1998.

[5] C. Rasch and T. Satzger. Remarks on the O(N) implementation of the fast marching method.

[6] Y. Chen and G. Medioni. Object modeling by registration of multiple range images. Image and Vision Computing (IVC), 10(3):145–155,1992

[7] Kok-Lim Low Linear Least-Squares Optimization for Point-to-Plane ICP Surface Registration

Thanks !!!