Depth Camera, 3D Reconstruction, and Applications

Jeong Joon Park

Overview

1. Range Imaging -- Why use it? How does it work?
    a. Structured Light Sensor Mechanism
2. Depth Sensor-based 3D reconstruction
    a. Explicit vs Implicit Representation
    b. Signed Distance Function
    c. SDF Fusion
    d. Depth-based Tracking
3. Recent Developments
    a. DynamicFusion
    b. Appearance Reconstruction
    c. Learning on SDF

Depth Camera: Core of AR Technology

- Automonous Driving: https://www.youtube.com/watch?v=JC94Y063x58
- Magic Leap Demo: https://www.youtube.com/watch?v=kPMHcanq0xM

Depth Sensor

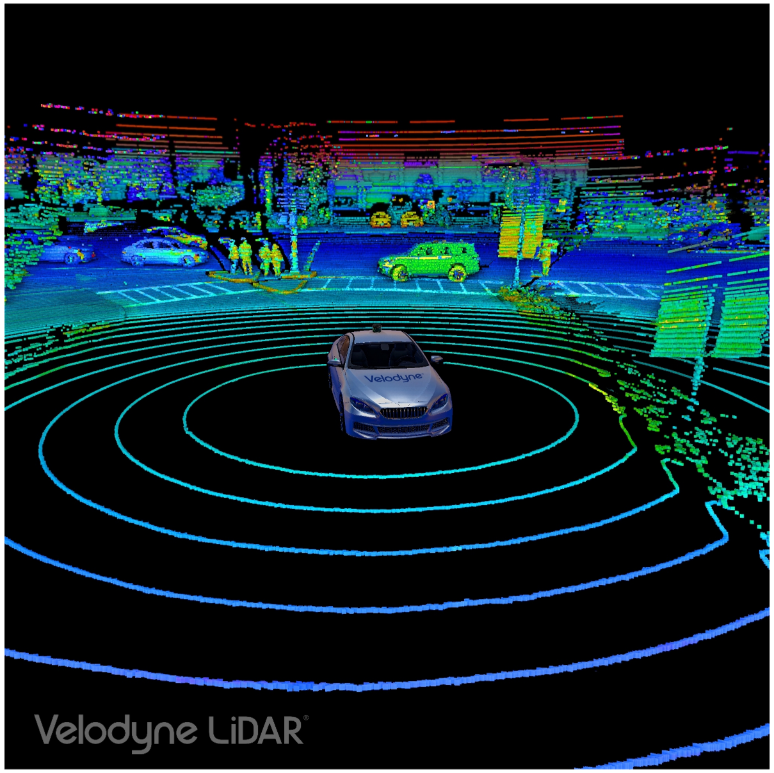
- Passive Depth -- Stereo
- ToF Camera
    - Phase Modulated, Lidar, etc
- Structured Light Sensor
    - Kinect

Depth Sensor

- Passive Depth -- Stereo
- ToF Camera
  - Phase Modulated, Lidar, etc
- Structured Light Sensor
  - Kinect

Depth Sensor

- Passive Depth -- Stereo
- ToF Camera
  - Phase Modulated, Lidar, etc
- Structured Light Sensor
  - Kinect



Depth Sensor

- Passive Depth -- Stereo
- ToF Camera
  - Phase Modulated, Lidar, etc
- Structured Light Sensor
  - Kinect, Hololens

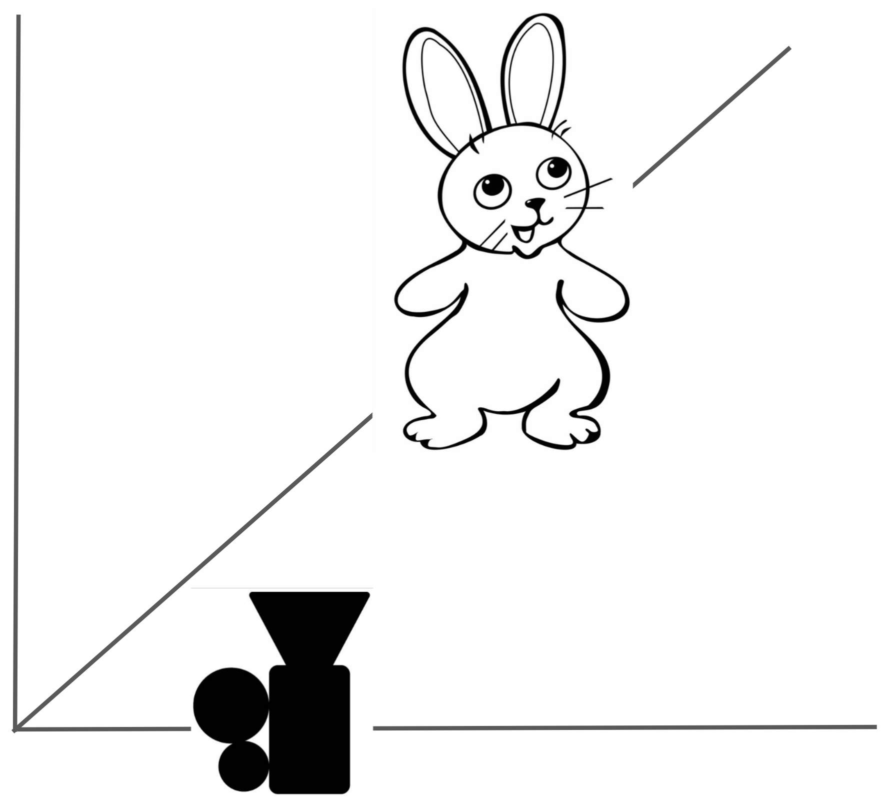
Depth Sensor

- Passive Depth -- Stereo
- ToF Camera
  - Phase Modulated, Lidar, etc
- Structured Light Sensor
  - Kinect, Hololens

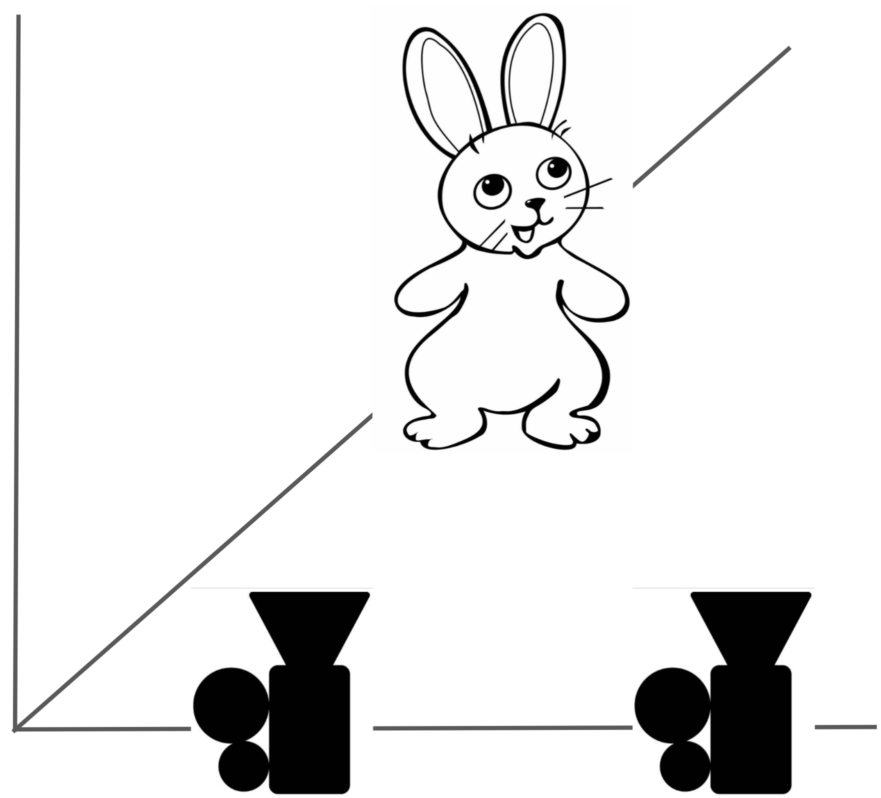
Depth Sensor

- Passive Depth -- Stereo
- ToF Camera
  - Phase Modulated, Lidar, etc
- Structured Light Sensor
  - Kinect, Hololens

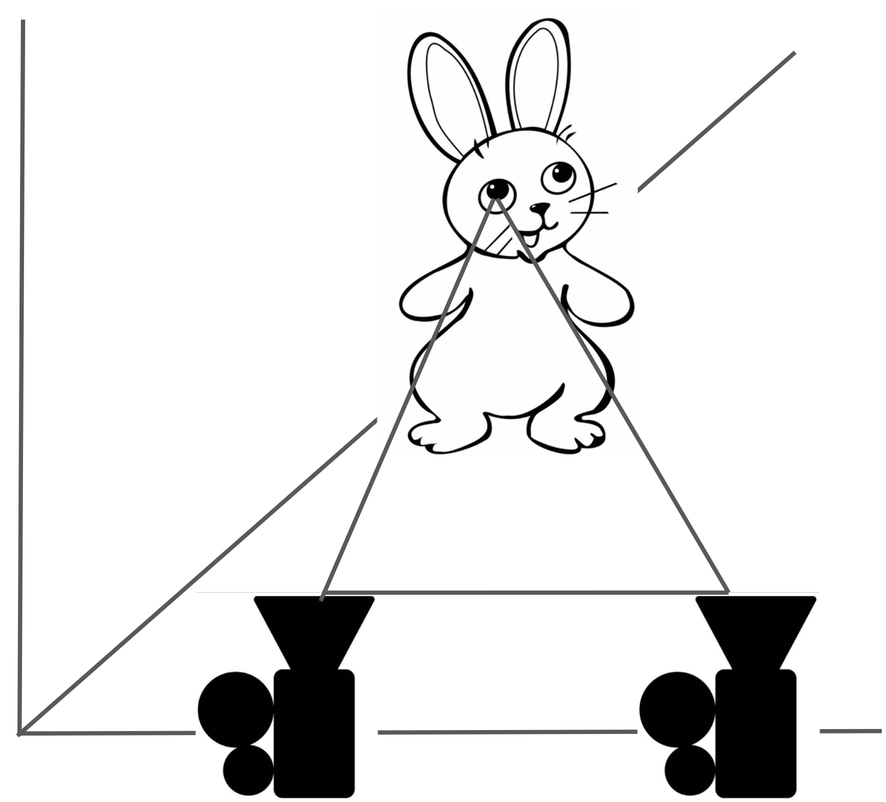
Stereo-Based Depth



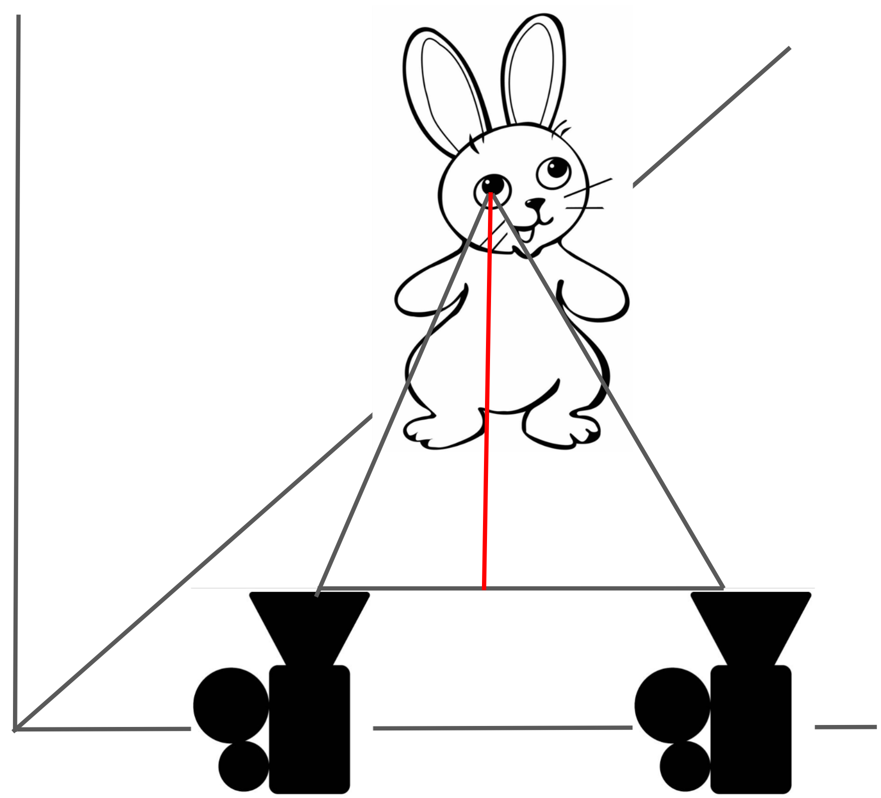
Stereo-Based Depth



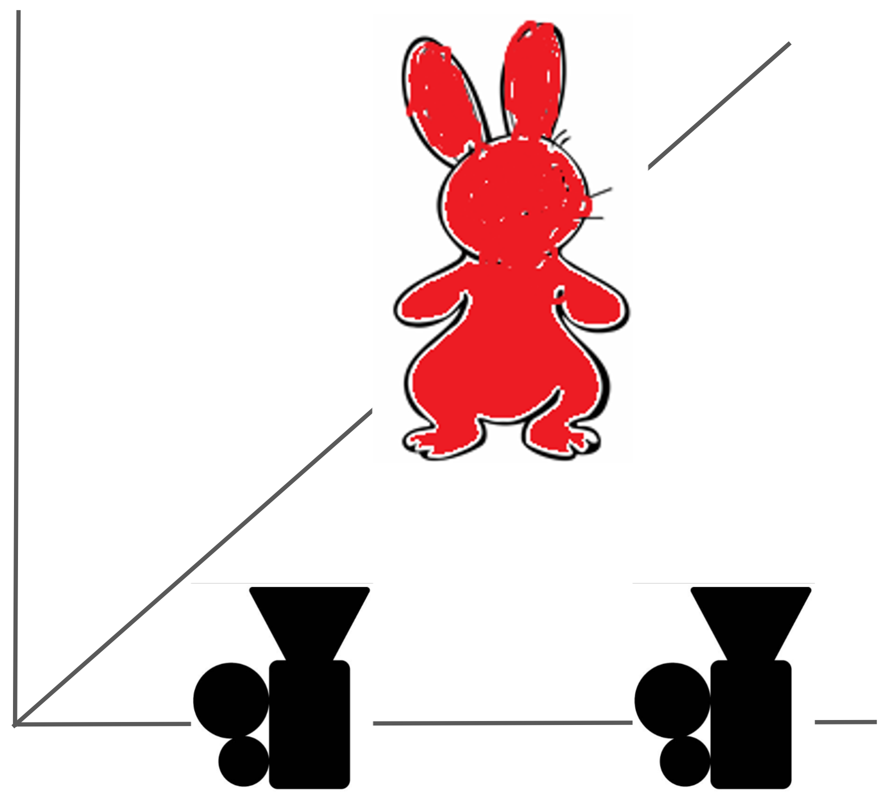
Stereo-Based Depth



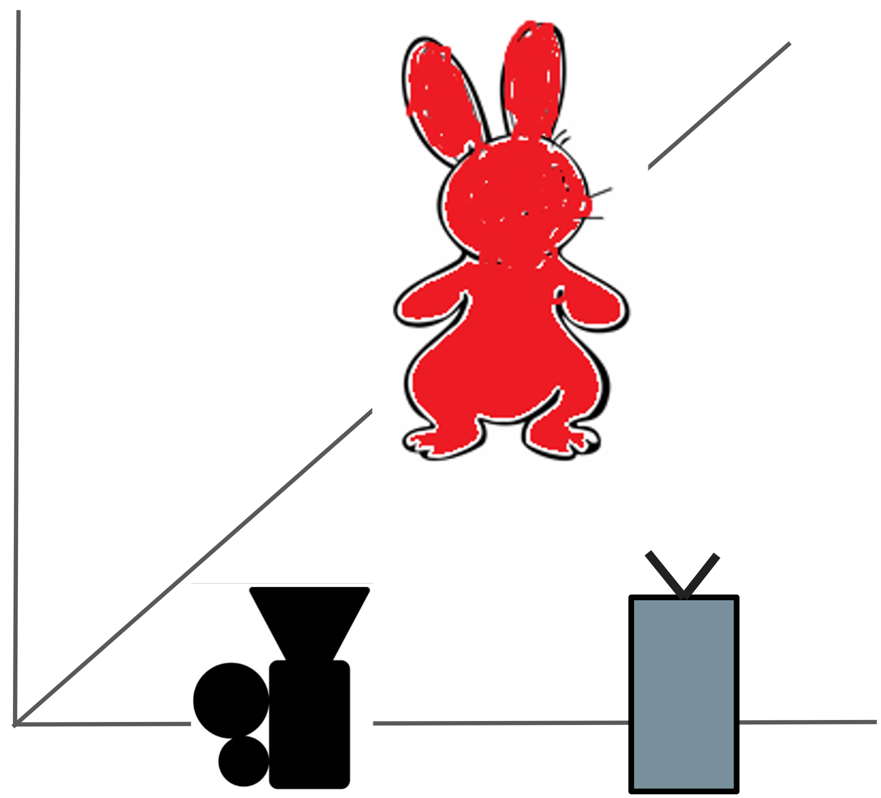
Stereo-Based Depth



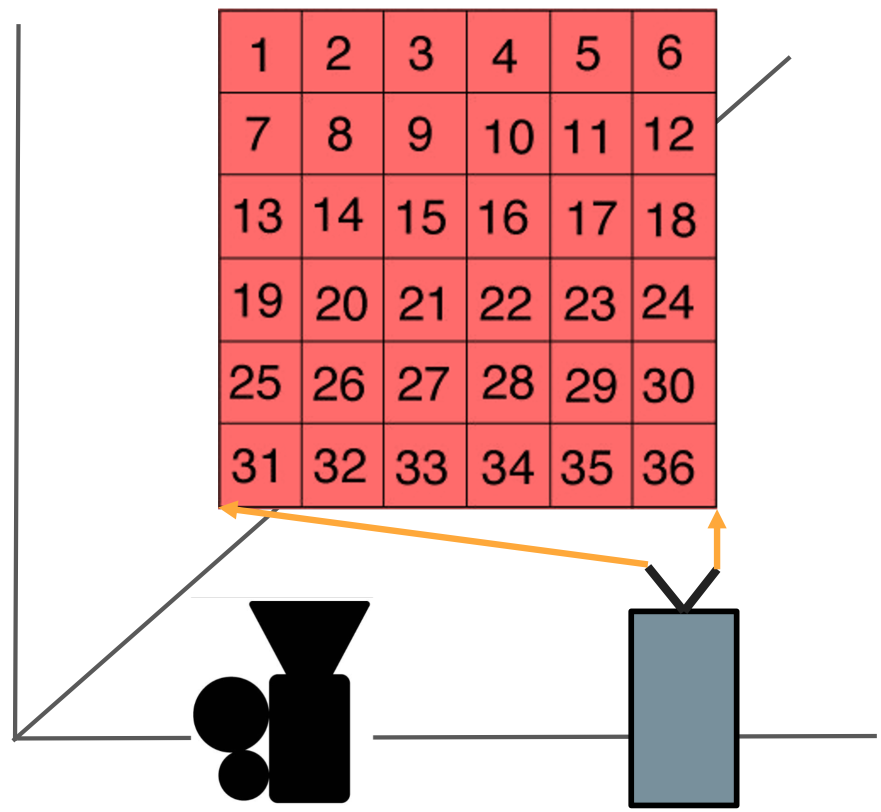
Stereo-Based Depth -- fails for textureless case



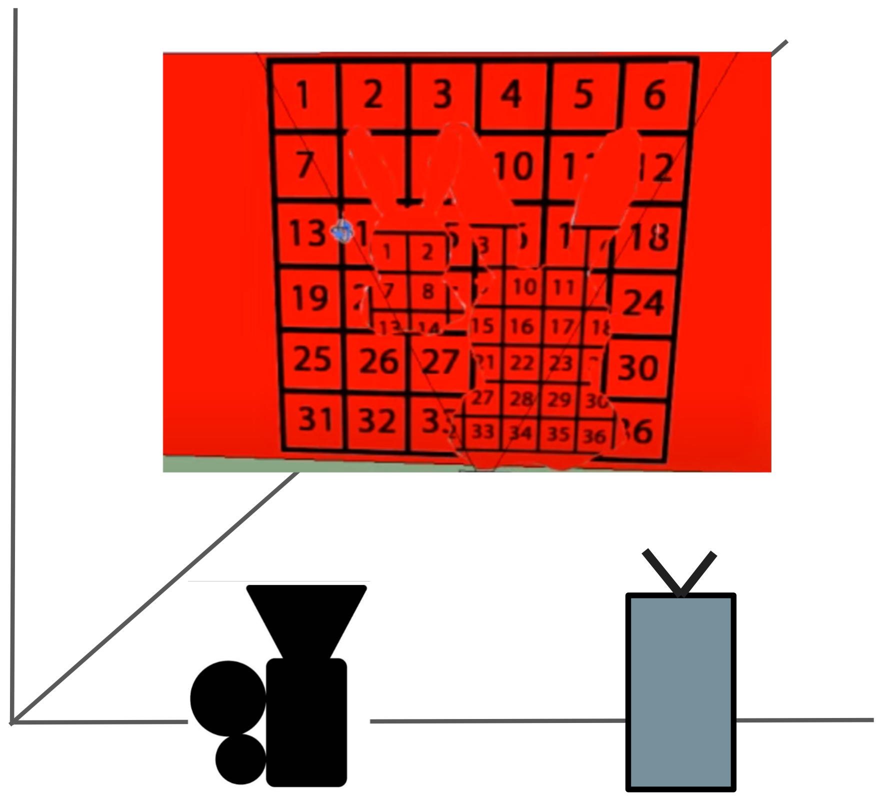
Stereo-Based Depth -- fails for textureless case



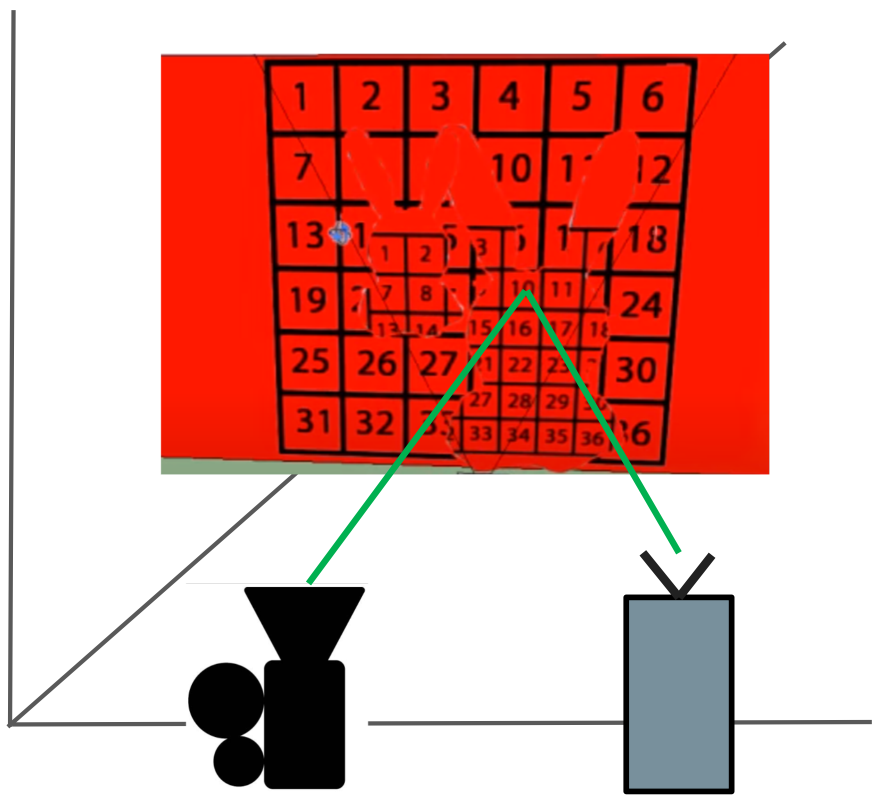
Active Depth Camera – IR Projector



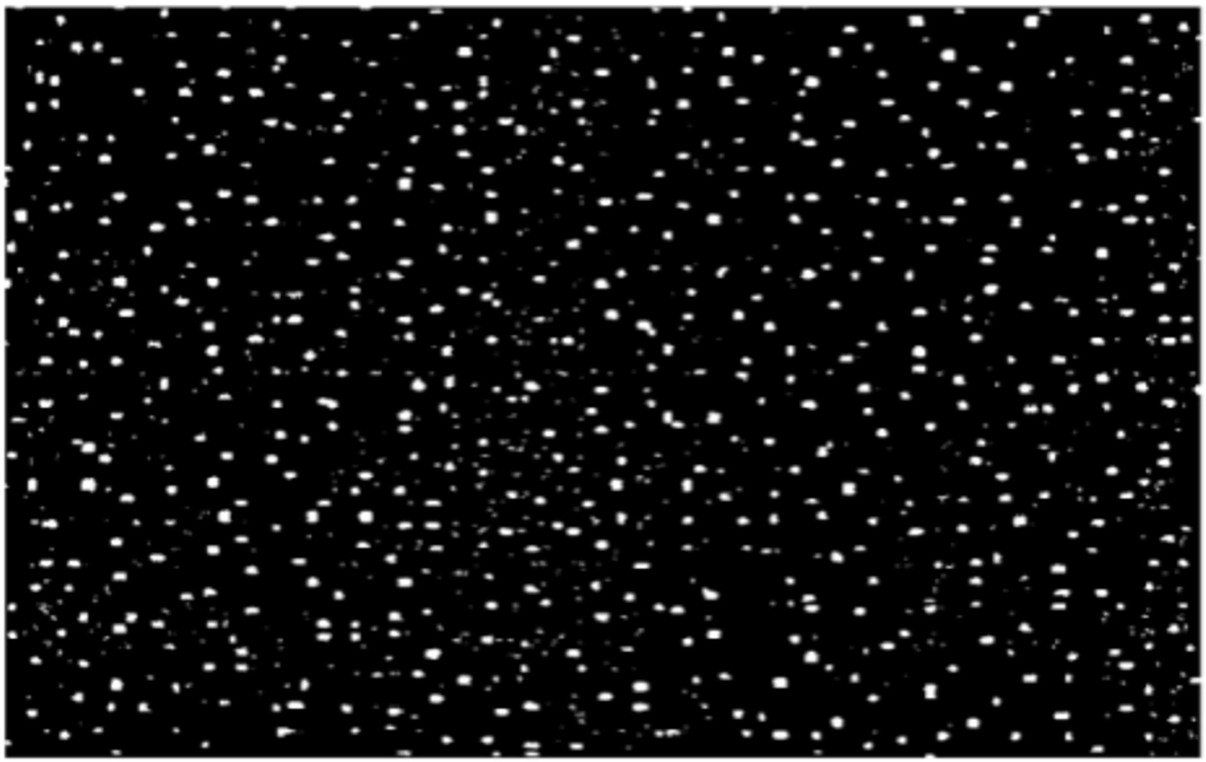
Active Depth Camera -- Projector



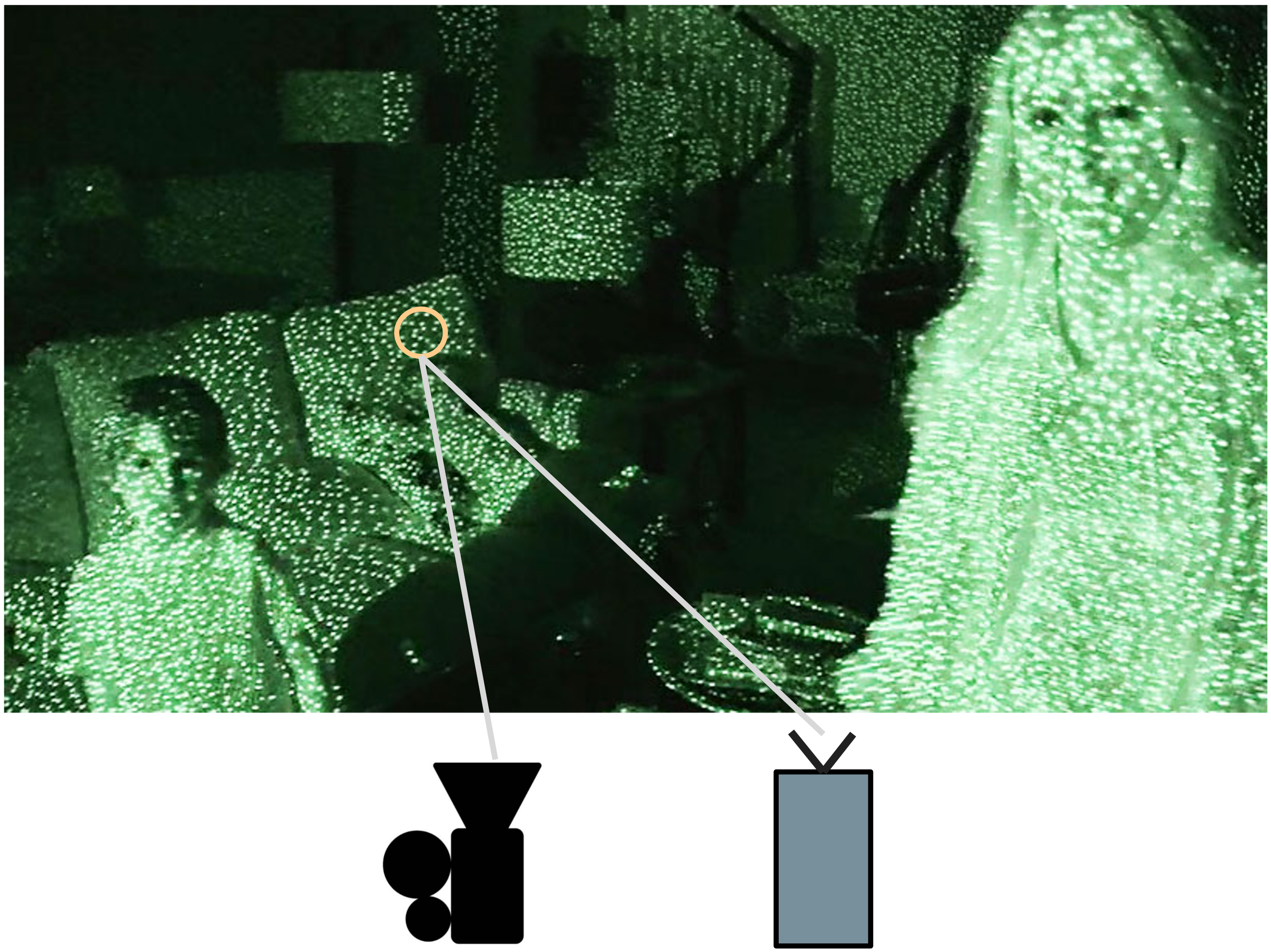
Active Depth Camera -- Projector



Active Depth Camera – Speckle Pattern

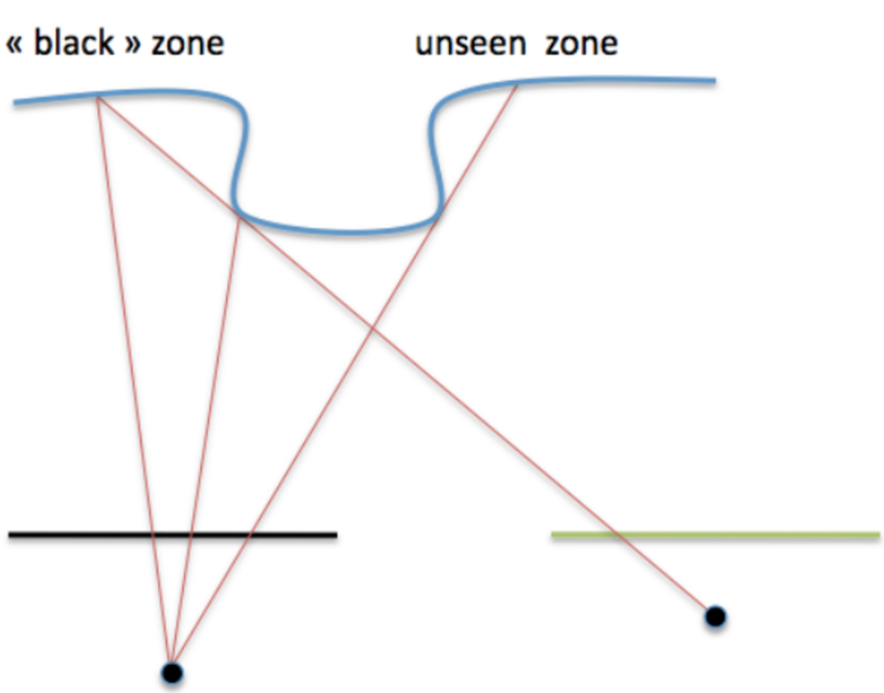


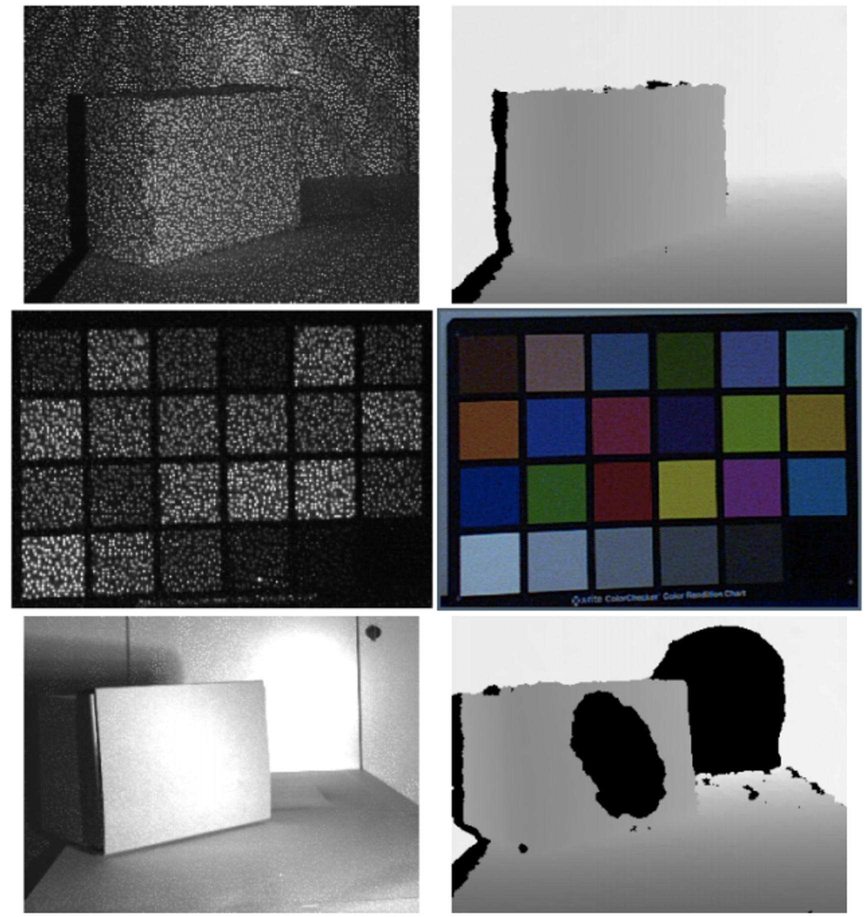
Shpunt et al, PrimeSense patent application
US 2008/0106746



Active Depth Camera

Failure Modes





KinectFusion Video

- https://www.youtube.com/watch?v=quGhaggn3cQ

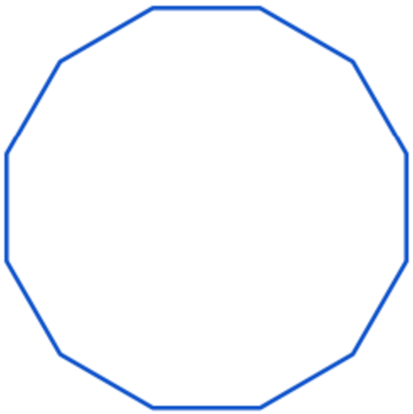
Newcombe, Richard A., et al. "KinectFusion: Real-time dense surface mapping and tracking." *2011 10th IEEE International Symposium on Mixed and Augmented Reality*. IEEE, 2011.

Dense 3D Reconstruction (KinectFusion 2011)

- Signed Distance Function (SDF)
- Raycasting
- SDF Fusion [Curless 1996]
- Tracking (Iterative Closest Point) [Rusinkiewicz 2001]

Explicit Surface Representation

- Explicitly Carry List of Vertices and Lines
  - E.g. Triangle Mesh

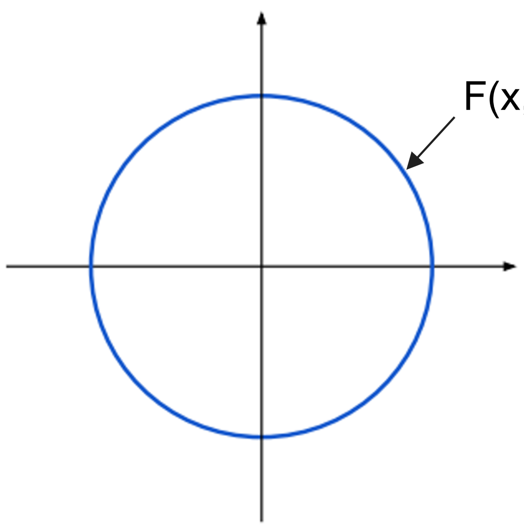


Vertices: [(x0, y0, z0), (x1, y1, z1), …, (xn, yn, zn)]

Indices: [(i0, i1), (i2, i3), …, (in-1, in)]

Implicit Surface Representation

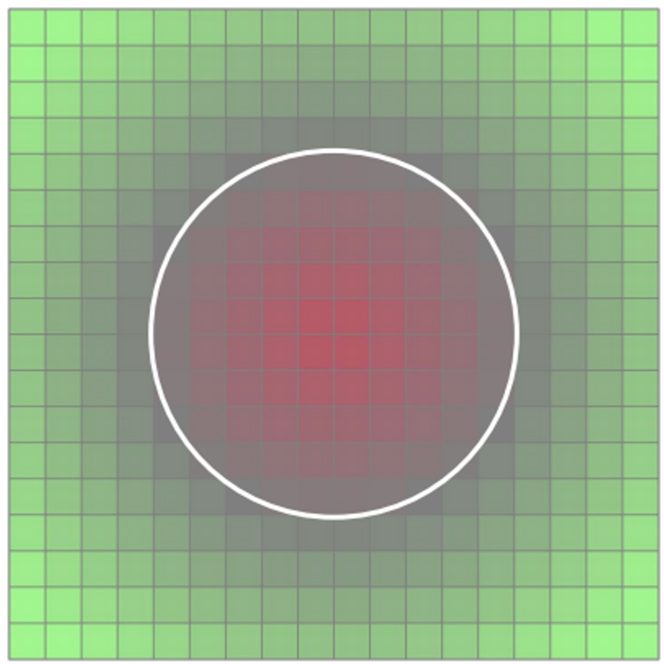
- Implicitly represent the surface through level set
    - E.g. parametric equation – zero level set!



$$f(x, y) = x^2 + y^2 - r^2$$

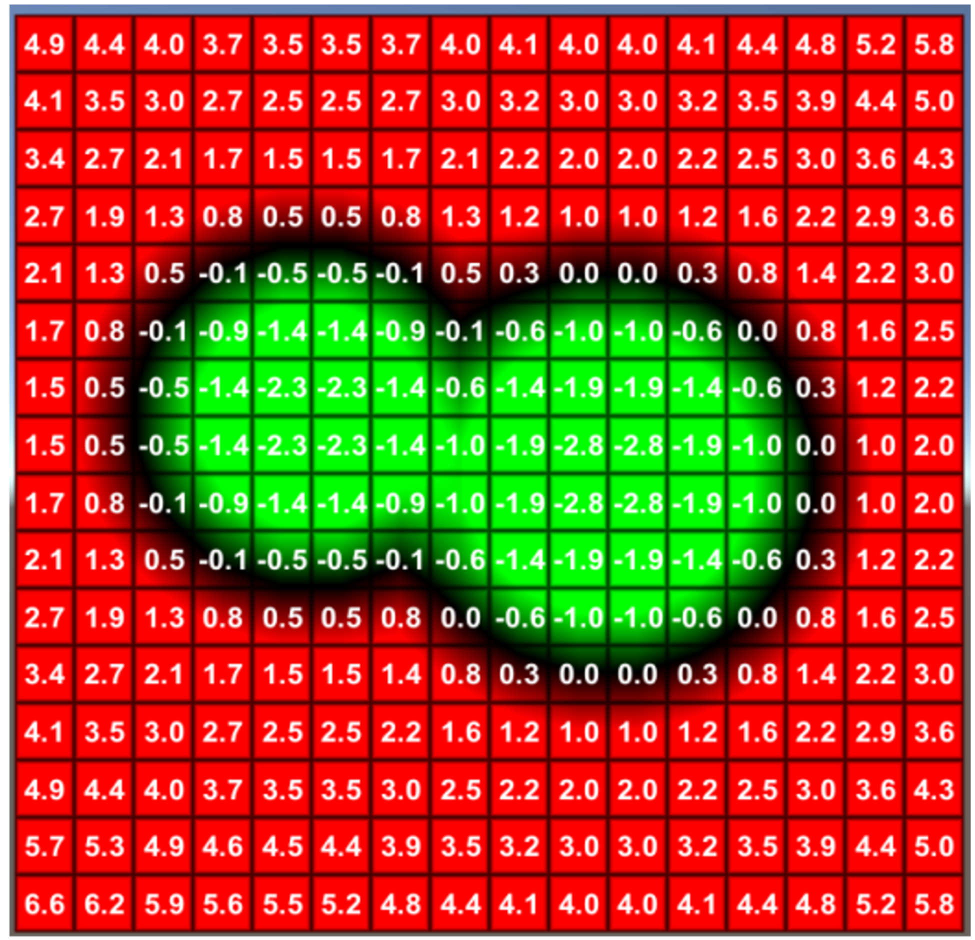
Implicit Surface Representation

- Implicitly represent the surface through level set
    - E.g. Non-parametric – zero level set!



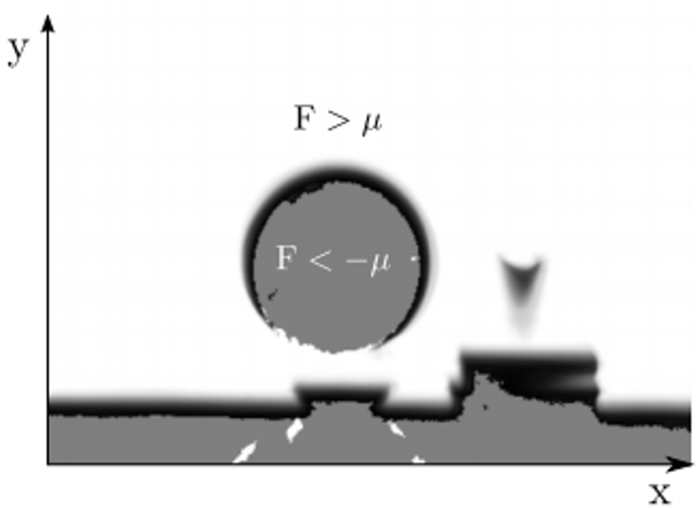
Signed Distance

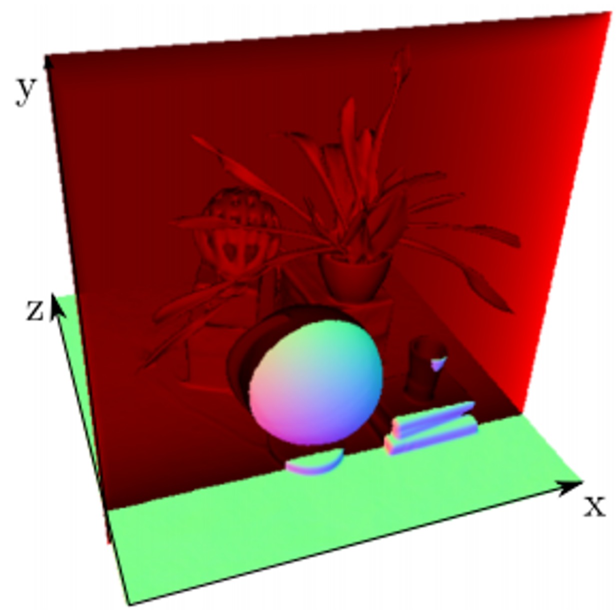
- Distance to Closest Surface



Implicit Surface Representation

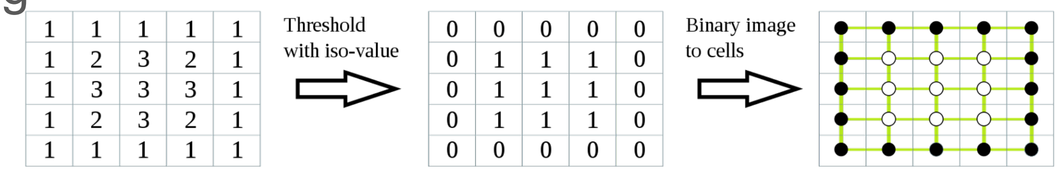
- Implicitly represent the surface through level set
  - Represent Arbitrary Topology!





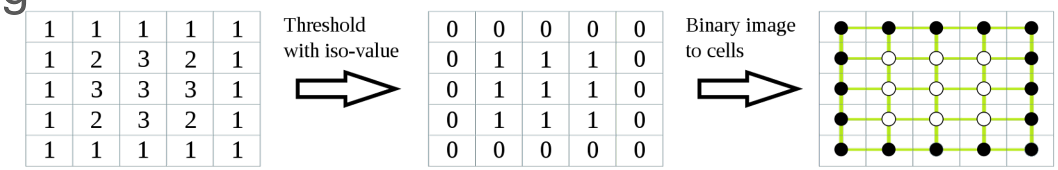
Implicit -> Explicit Conversion

- Marching Cubes / Squres
- 1. Thresholding



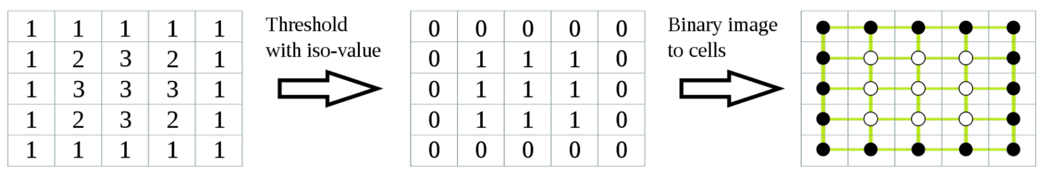
Implicit -> Explicit Conversion

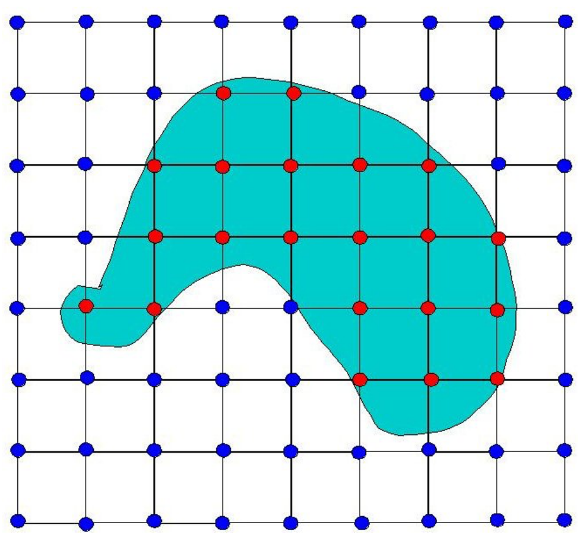
- Marching Cubes / Squres
- 1. Thresholding

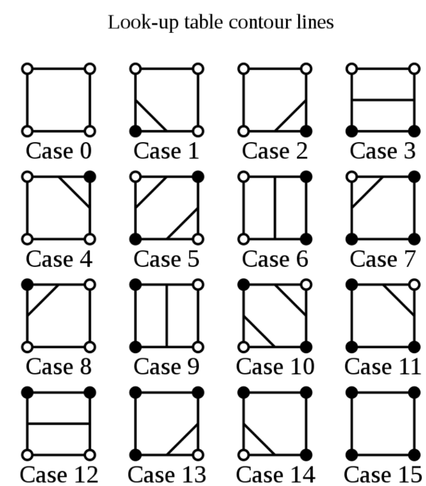


Implicit -> Explicit Conversion

- Lookup Codebook for Marching Cubes / Squres

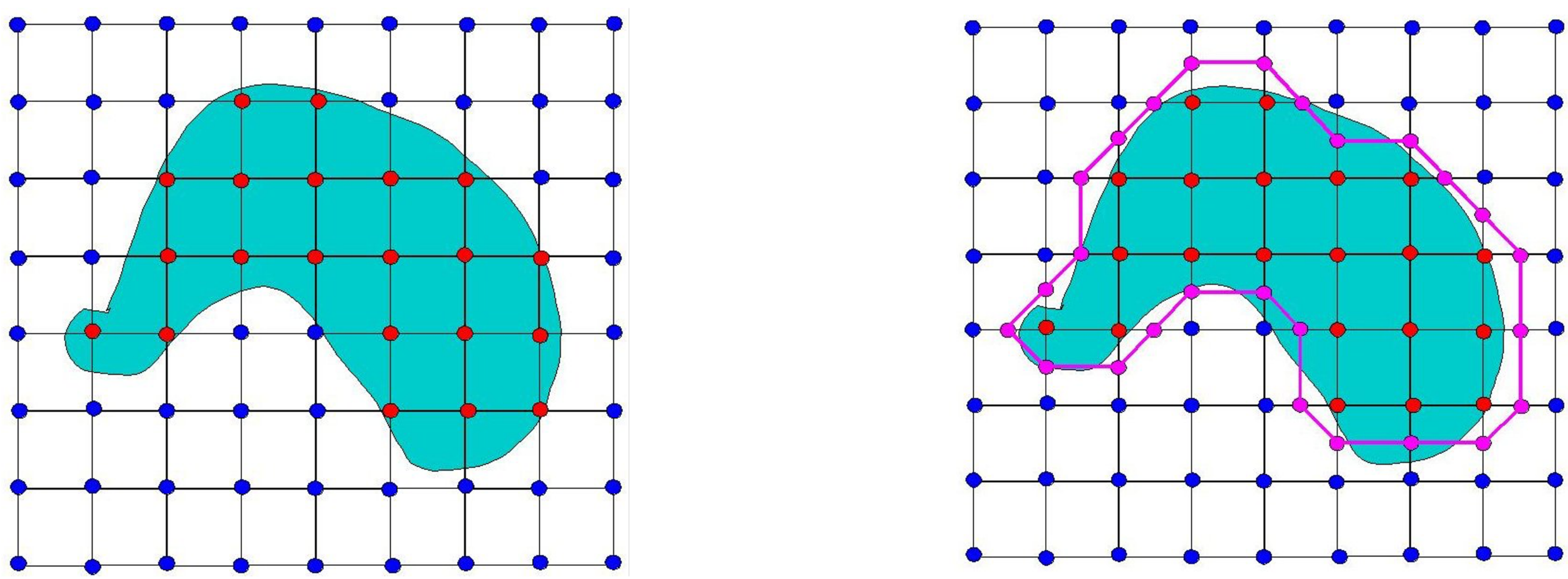






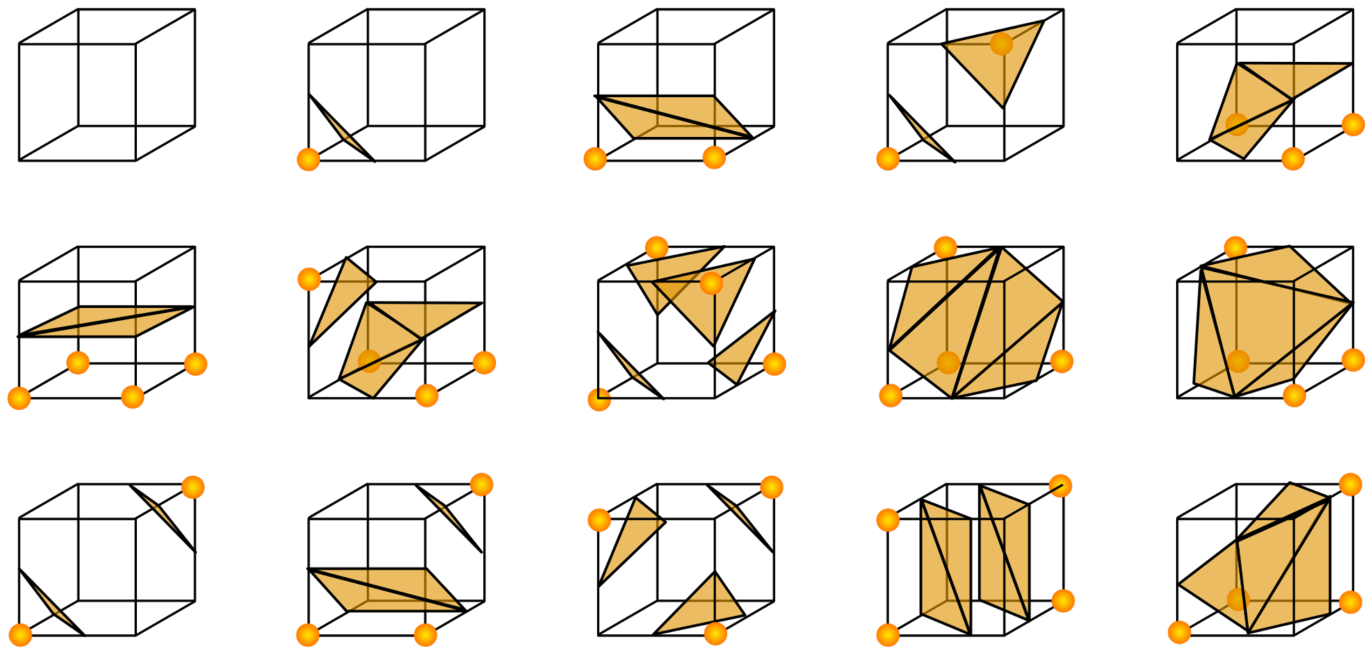
Implicit -> Explicit Conversion

- Lookup Codebook for Marching Cubes / Squres



Implicit -> Explicit Conversion

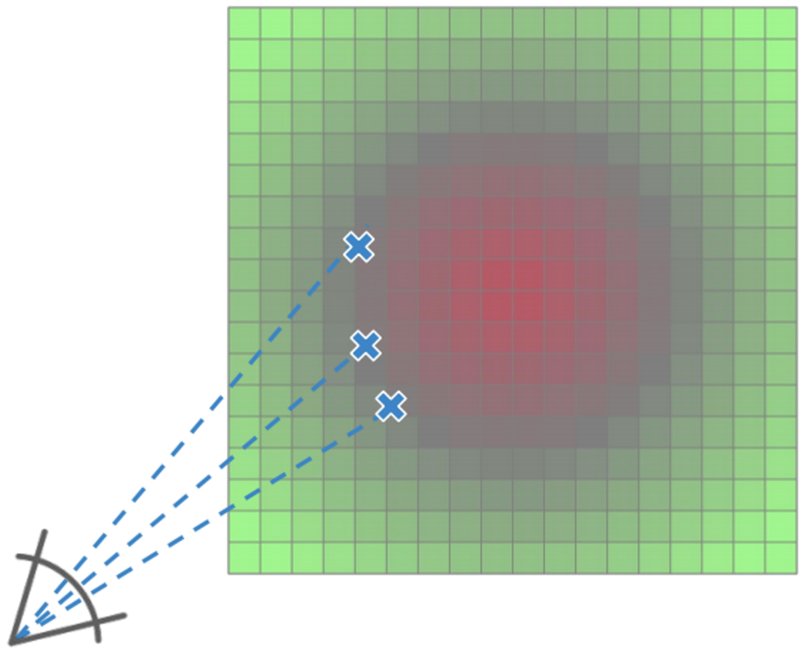
- 3D Marching Cubes



Lorensen, William E., and Harvey E. Cline. "Marching cubes: A high resolution 3D surface construction algorithm." *ACM siggraph computer graphics* 21.4 (1987): 163-169.

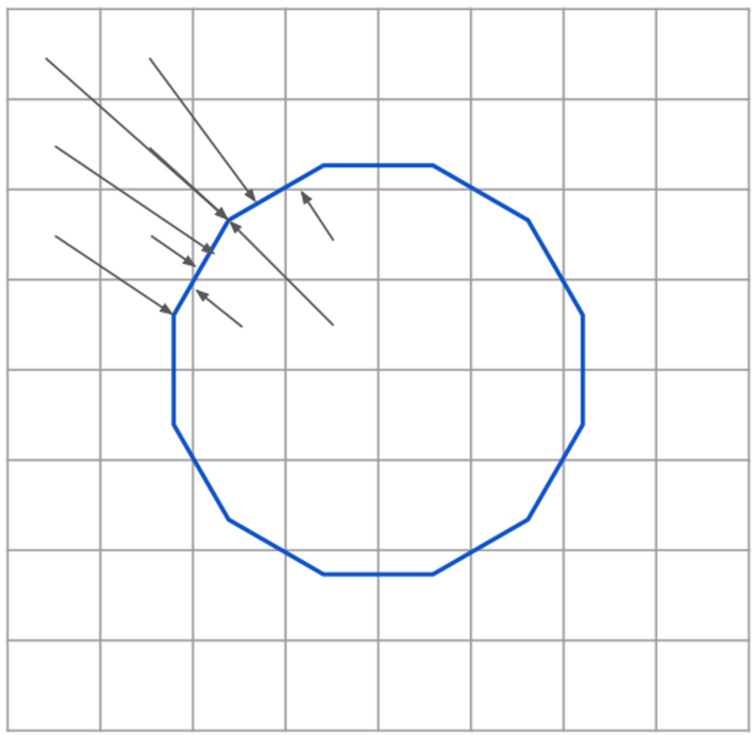
Implicit -> Explicit Conversion

- RayCasting → produce depth map



Implicit -> Explicit Conversion

- Raycasting: Arbitrary Accuracy



Dense 3D Reconstruction (KinectFusion 2011)

- Signed Distance Function (SDF)
- Marching Cubes, Raycasting
- **SDF Fusion (Curless 1996)**
- Tracking (Iterative Closest Point)

SDF Fusion

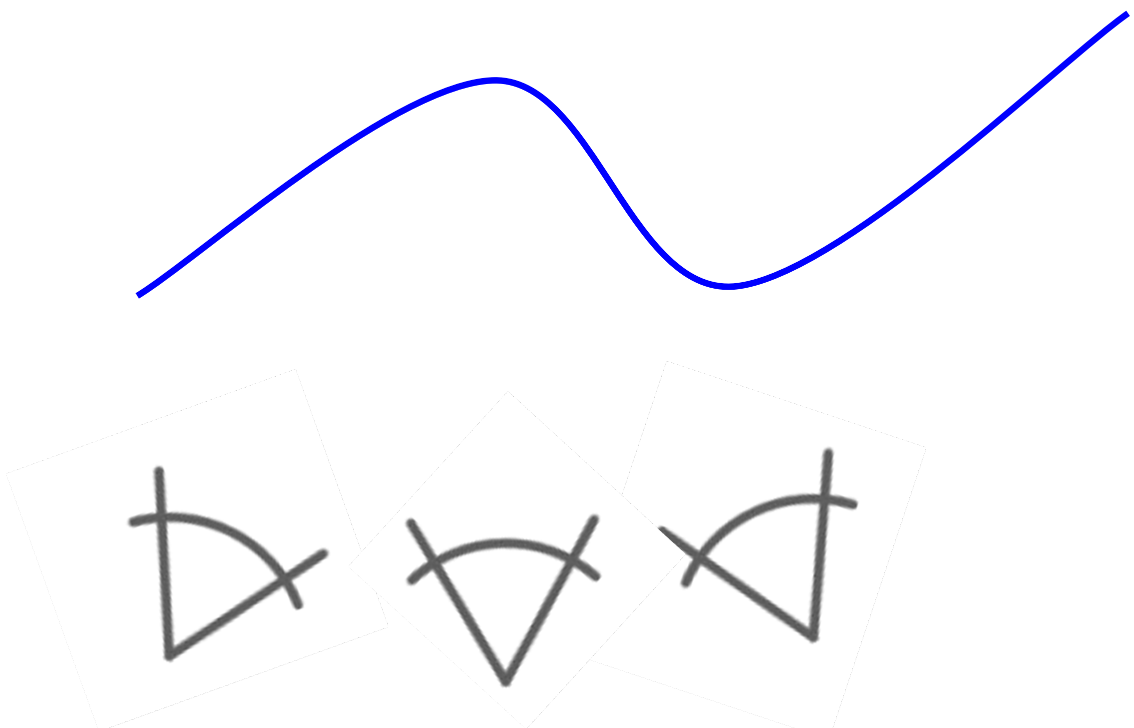
- Range Sensing to Volumetric SDFs
- Projective SDF vs True SDF
- SDF Averaging Fusion

Newcombe, Richard A., et al. "KinectFusion: Real-time dense surface mapping and tracking." *2011 10th IEEE International Symposium on Mixed and Augmented Reality*. IEEE, 2011.
Curless, Brian, and Marc Levoy. "A volumetric method for building complex models from range images." *Proceedings of the 23rd annual conference on Computer graphics and interactive techniques*. 1996.

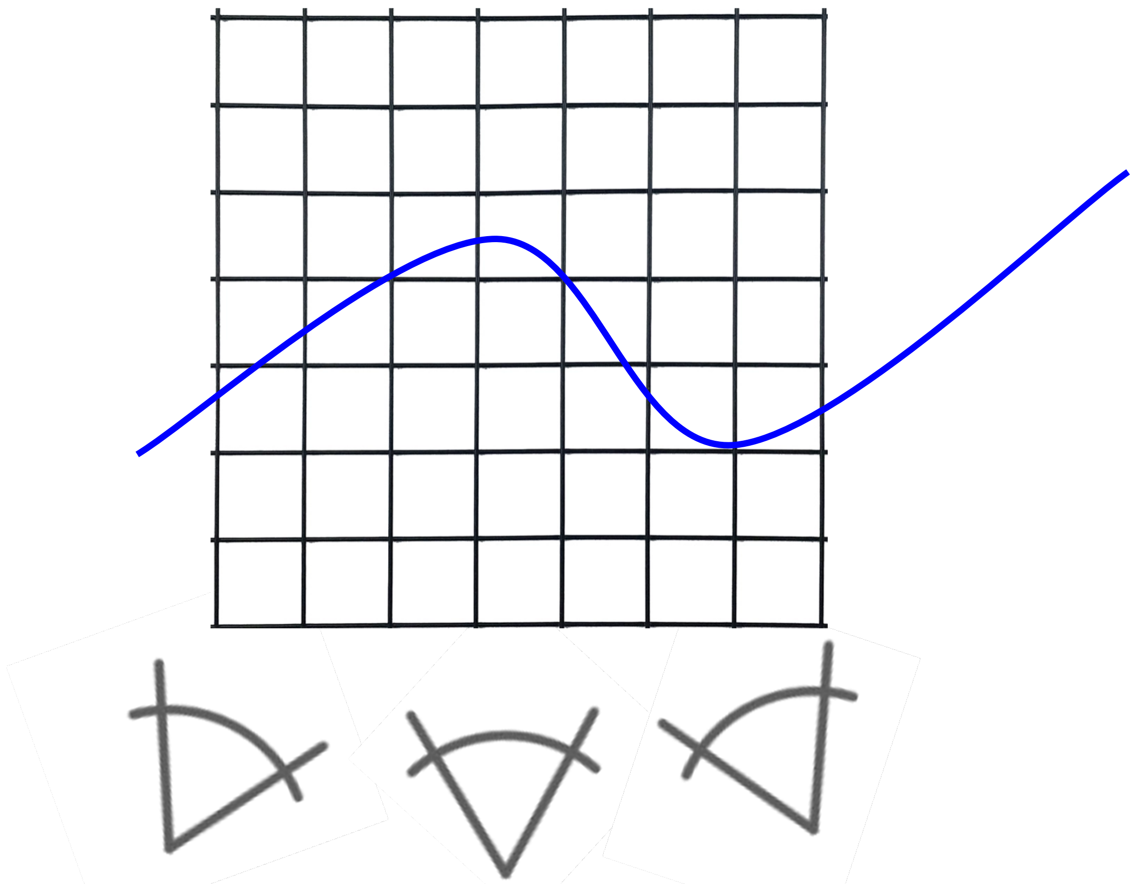
Fusing Multi-view Depth

Range Observations

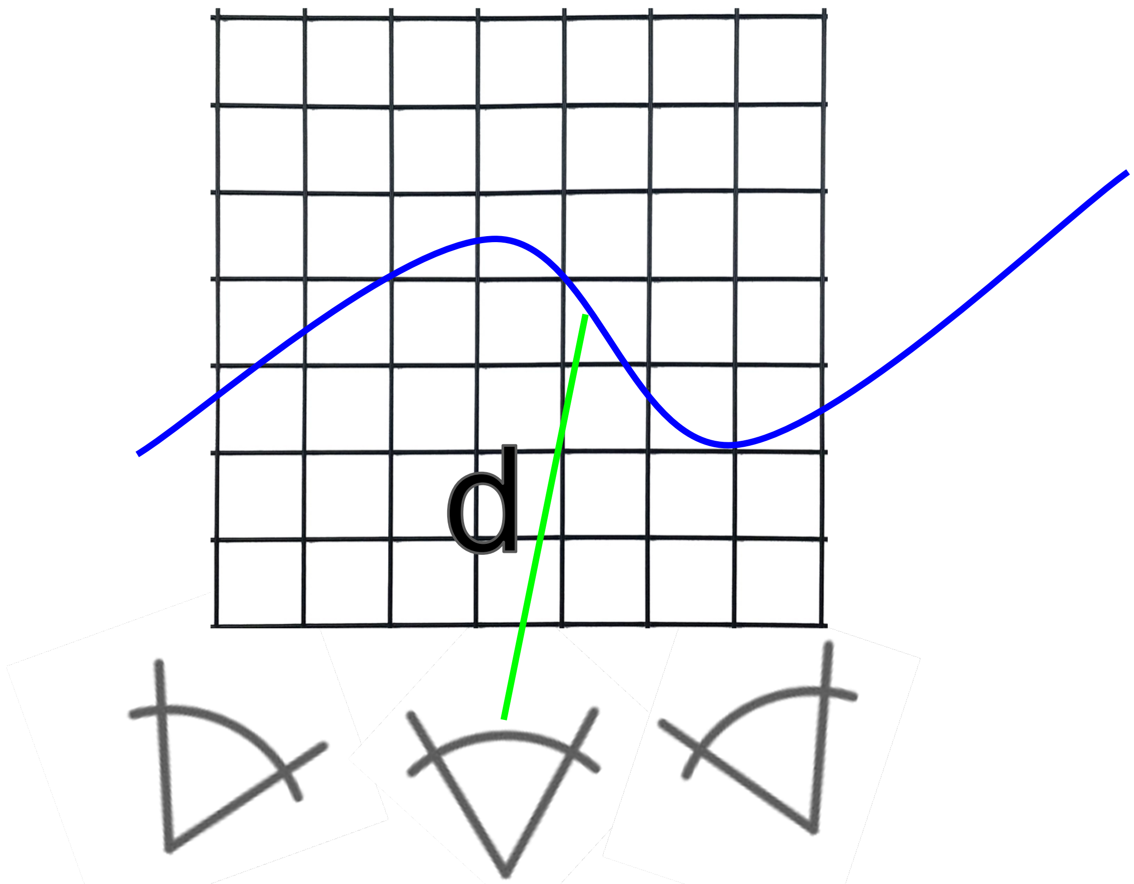
Known Camera Poses



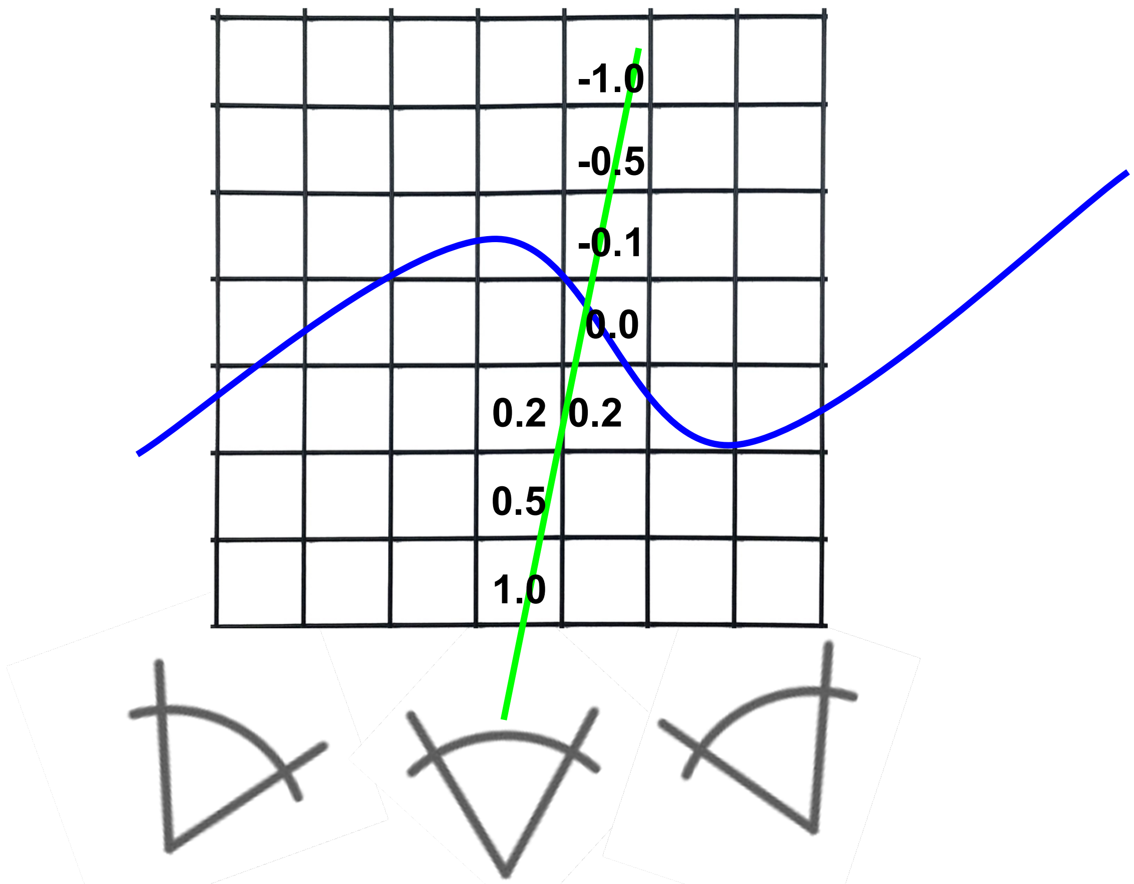
Fusing Multi-view Depth



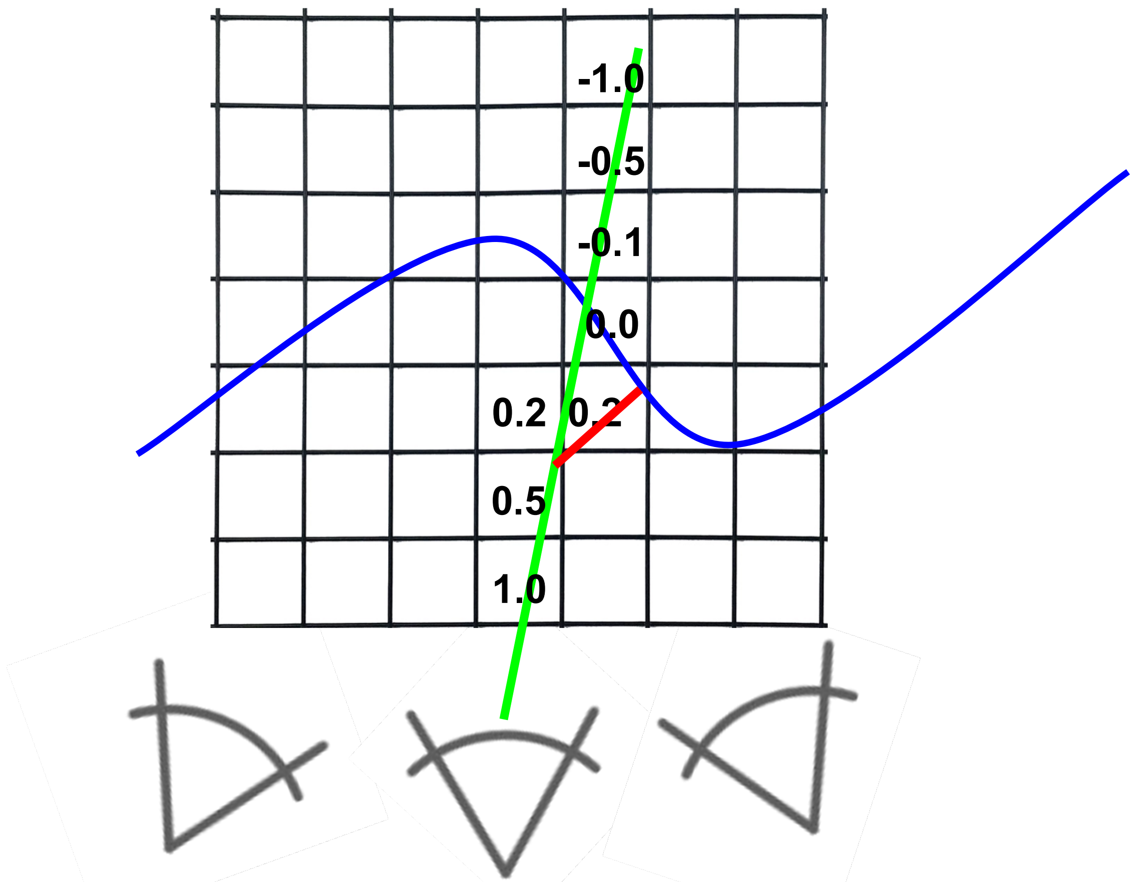
Fusing Multi-view Depth



Fusing Multi-view Depth

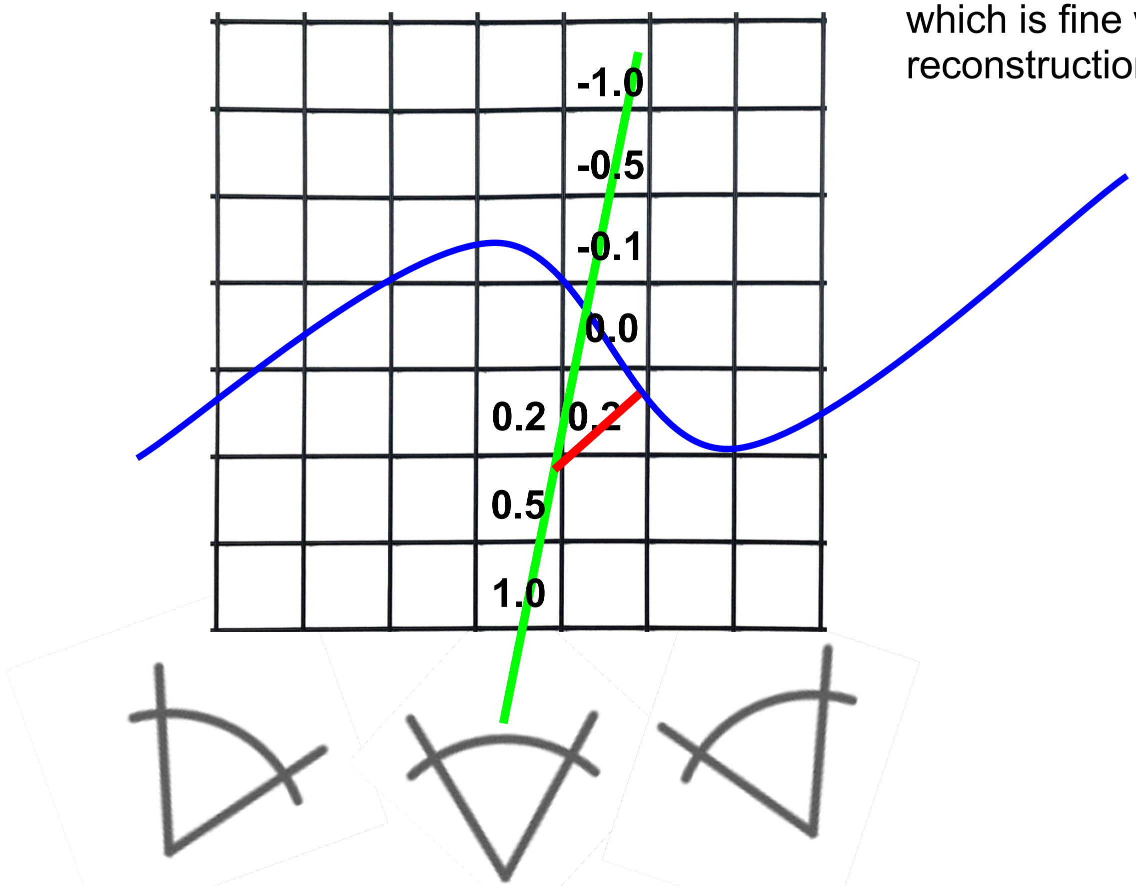


Projective vs True SDF

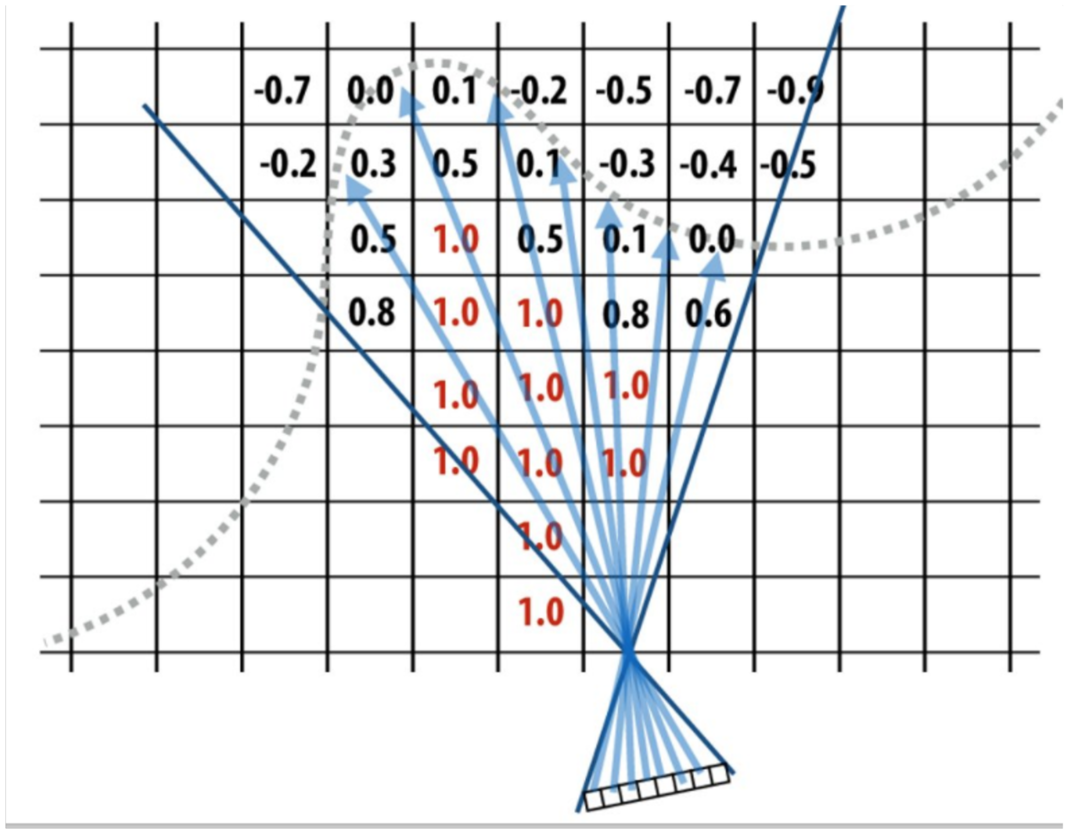


Projective vs True SDF

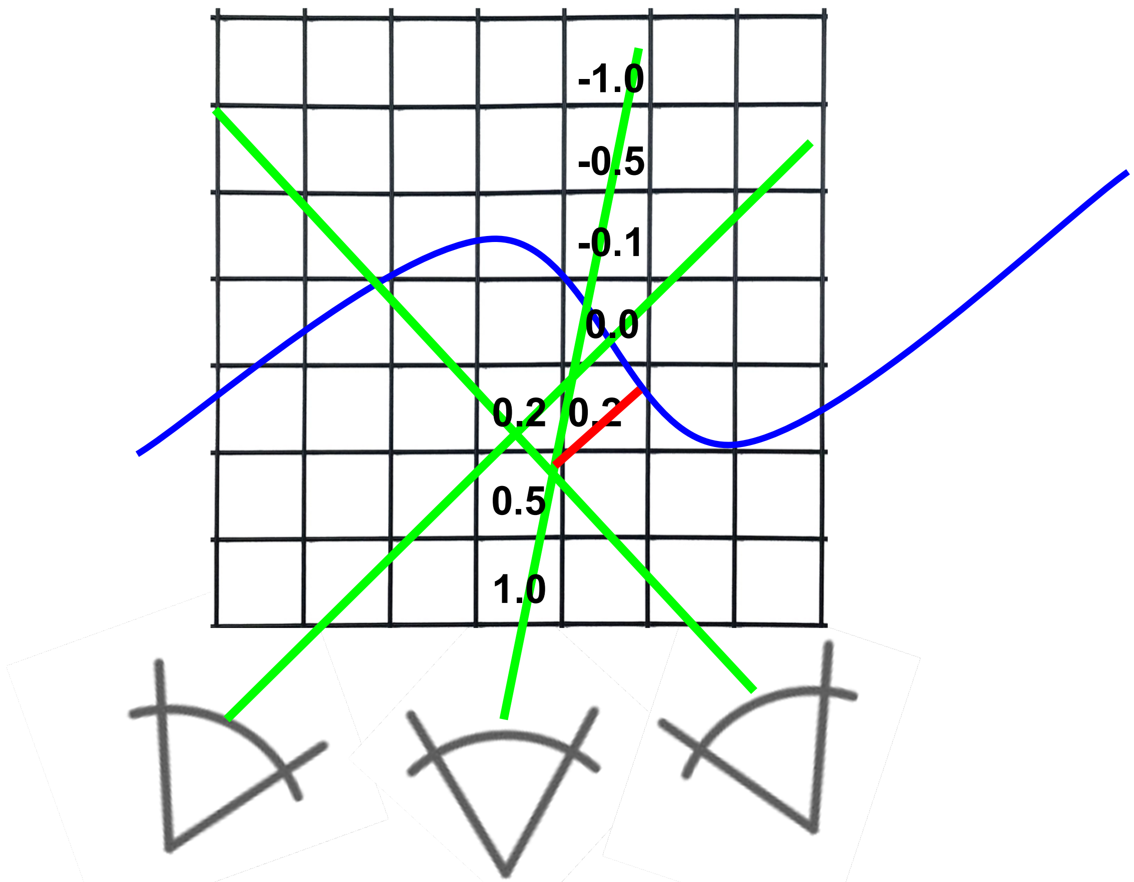
Only precise at zero-crossing,
which is fine when interested in surface reconstruction.



Projective vs True SDF



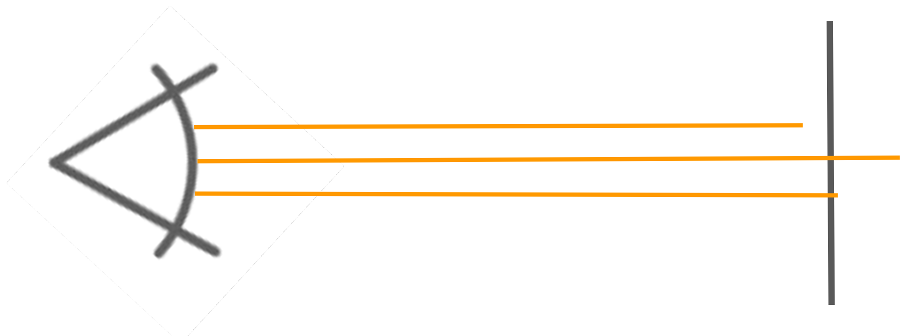
Projective vs True SDF -- Averaging!



Projective SDF, Why Averaging?

Measurement: r_1, r_2, r_3, …

Averaging is optimal in least squares sense

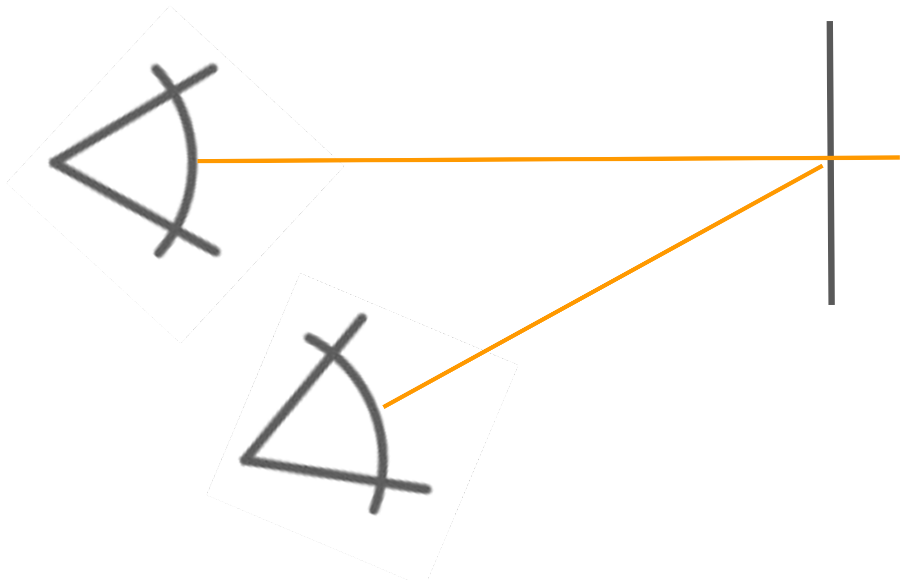


$$\operatorname{argmin} \frac{1}{n} \sum_{i} (\hat{r} - r_i)^2$$

Projective SDF, Why Averaging?

Measurement: r_1, r_2, r_3, …

Weighted Averaging: Weights based on view-angle, camera speed, etc



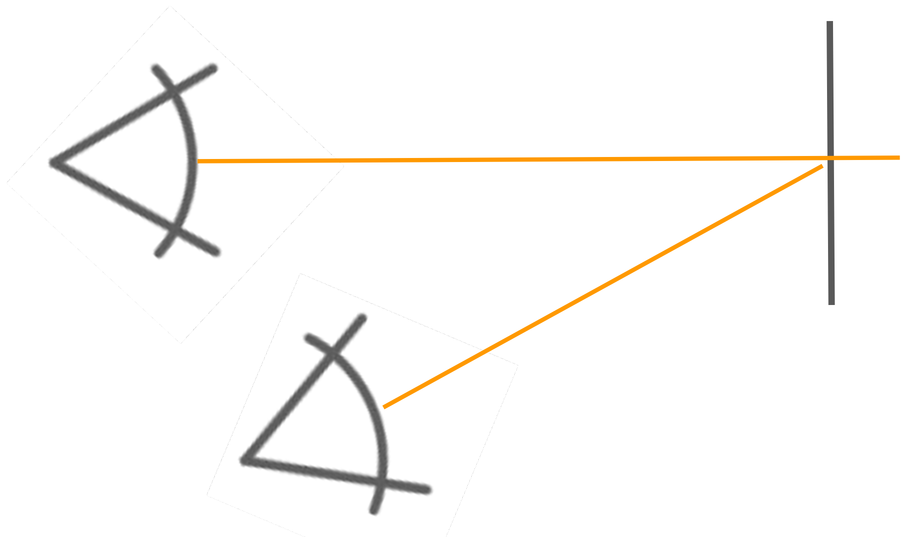
$$R = \frac{\Sigma w_i r_i}{\Sigma w_i}$$

Projective SDF, Why Averaging?

Measurement: r_1, r_2, r_3, …

Weighted Averaging: Weights based on view-angle, camera speed, etc

Provably Optimal in Least Squares sense



$$R = \frac{\Sigma w_i r_i}{\Sigma w_i}$$

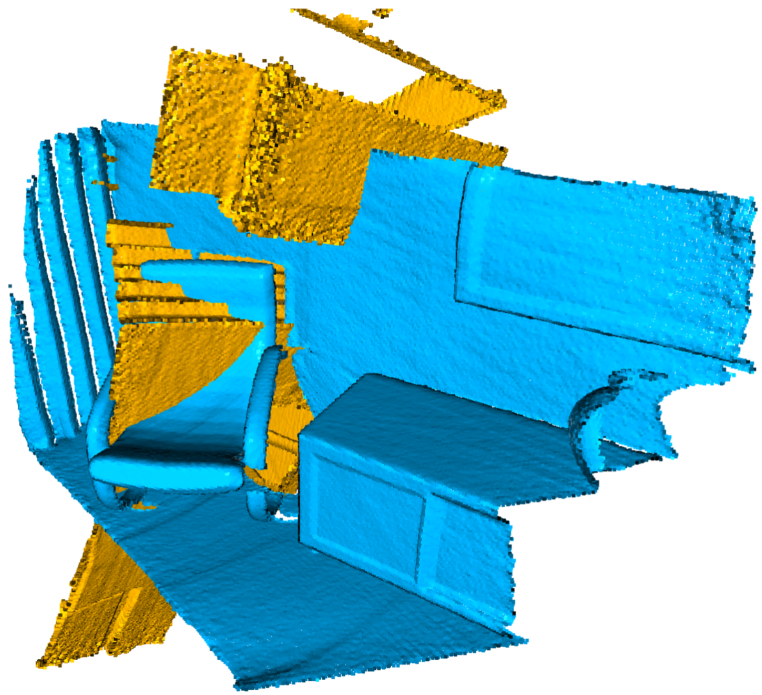
Curless, Brian, and Marc Levoy. "A volumetric method for building complex models from range images." *Proceedings of the 23rd annual conference on Computer graphics and interactive techniques*. 1996.

Dense 3D Reconstruction (KinectFusion 2011)

- Signed Distance Function (SDF)
- Raycasting
- SDF Fusion (Curless 1996)
- **Tracking (Iterative Closest Point)**

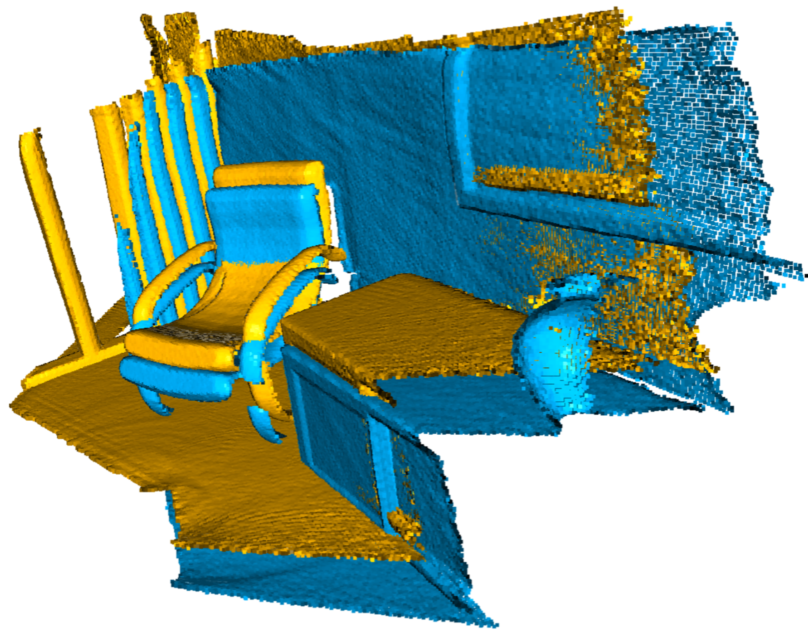
Camera Tracking

Registering two Point Clouds



Camera Tracking

Registering two Point Clouds

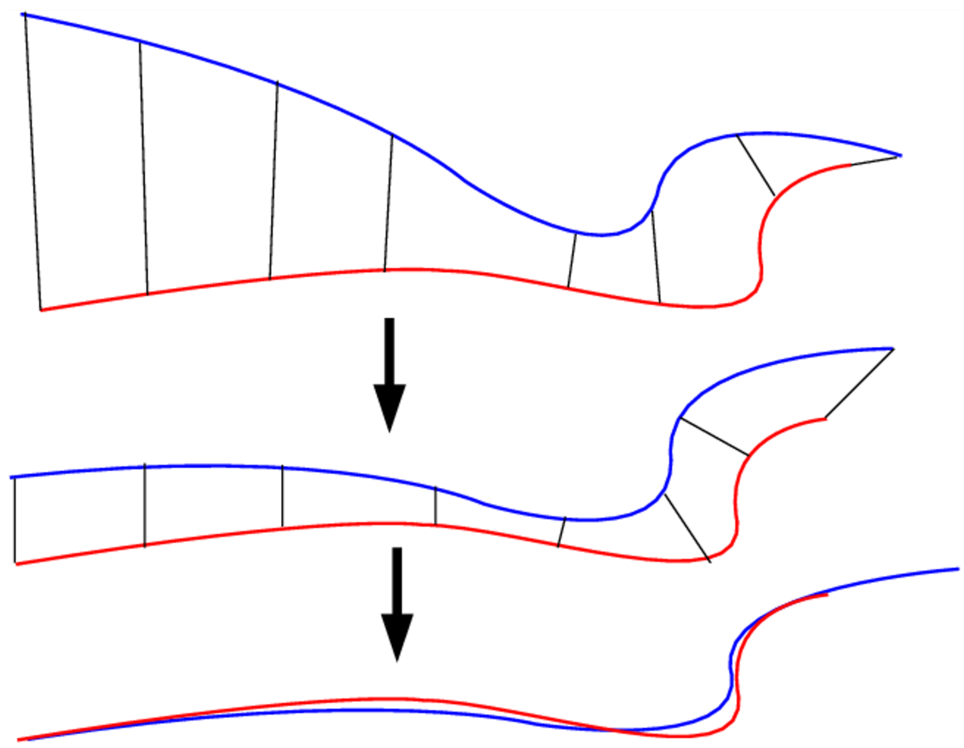


Camera Tracking

Registering two Point Clouds

Iterative Closest Point

[Besl and McKay 1992]

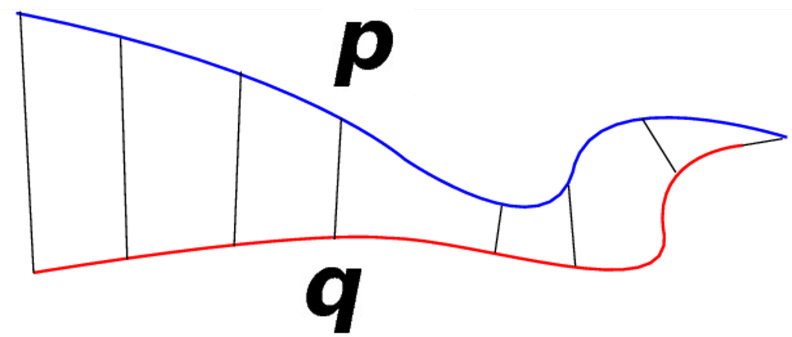


Besl, Paul J., and Neil D. McKay. "Method for registration of 3-D shapes." Sensor fusion IV: control paradigms and data structures. Vol. 1611. *International Society for Optics and Photonics*, 1992.

Camera Tracking

Step 1. Find Correspondence

Correspondence set $\mathcal{K} = \{(\boldsymbol{p}, \boldsymbol{q})\}$

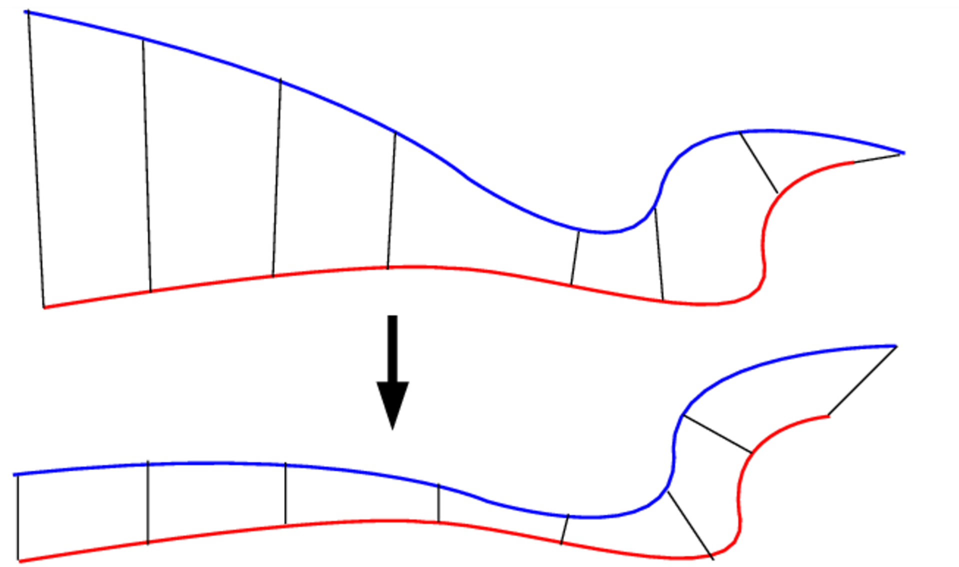


Transformation Matrix

$$\boldsymbol{T} = \begin{pmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{21} & a_{22} & a_{23} & a_{24} \\ a_{31} & a_{32} & a_{33} & a_{34} \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

Camera Tracking

Step 2: Minimize distance between correspondences



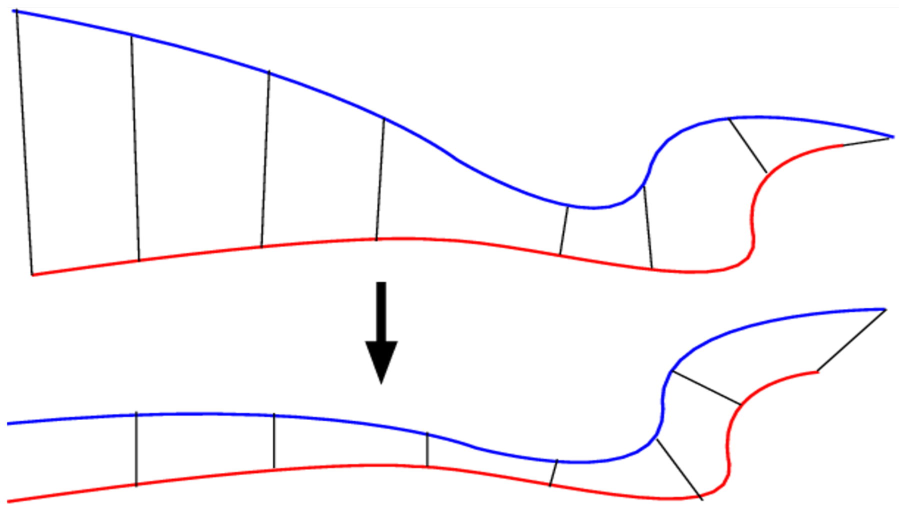
Camera Tracking

Step 2: Minimize distance between correspondences

by finding optimal Transformation using Small-angle approximation

https://www.comp.nus.edu.sg/~lowkl/publications/lowk_point-to-plane_icp_techrep.pdf

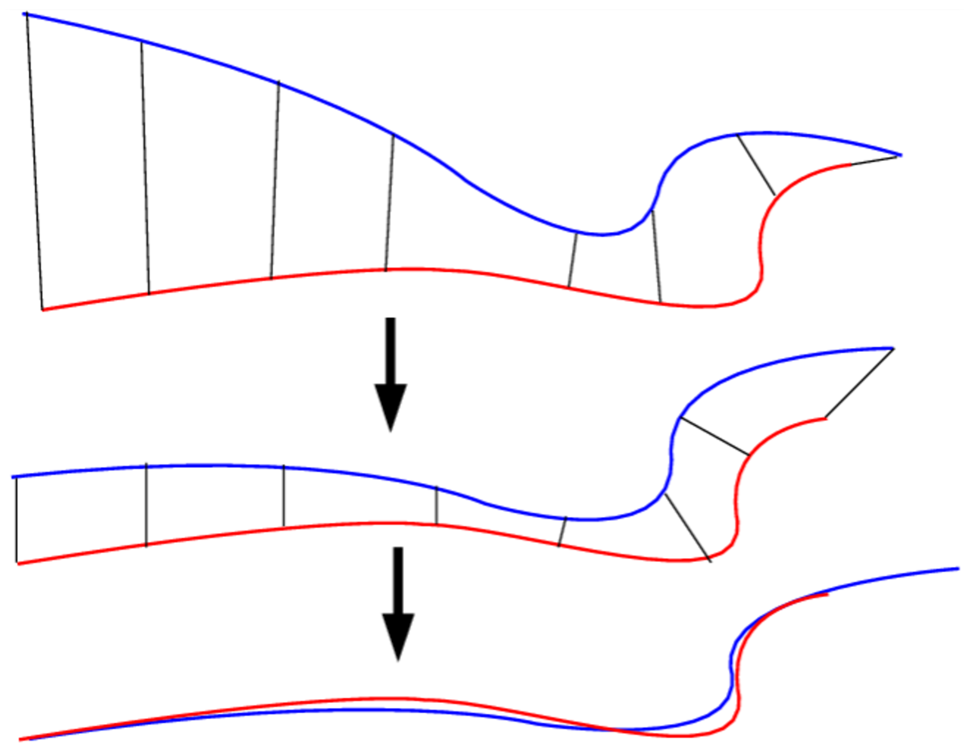
$$E(\mathbf{T}) = \sum_{(\mathbf{p},\mathbf{q})\in\mathcal{K}} \|\mathbf{p} - \mathbf{T}\mathbf{q}\|^2$$



Camera Tracking

Iterate Step 1 & 2
Until the energy is low enough!

$$E(\mathbf{T}) = \sum_{(\mathbf{p},\mathbf{q})\in\mathcal{K}} \|\mathbf{p} - \mathbf{T}\mathbf{q}\|^2$$

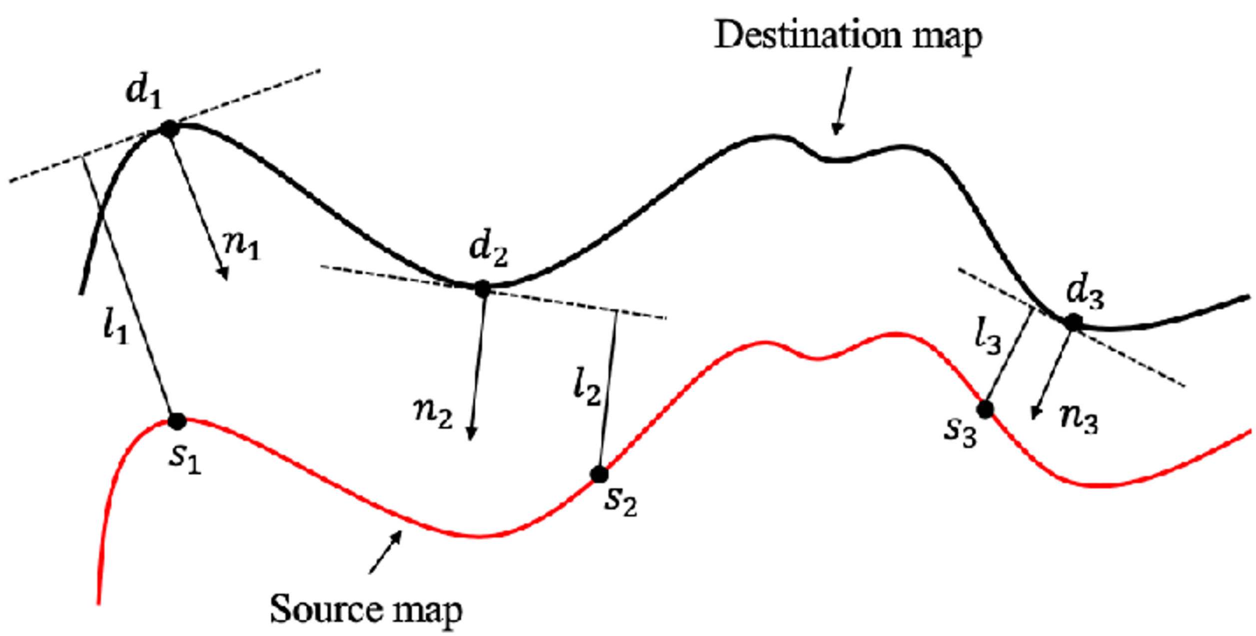


Camera Tracking

$$E(\mathbf{T}) = \sum_{(\mathbf{p},\mathbf{q})\in\mathcal{K}} \left((\mathbf{p} - \mathbf{T}\mathbf{q}) \cdot \mathbf{n_p}\right)^2$$

Often use **Point-to-Plane loss [Rusinkiewicz 2001]**

Converges Faster

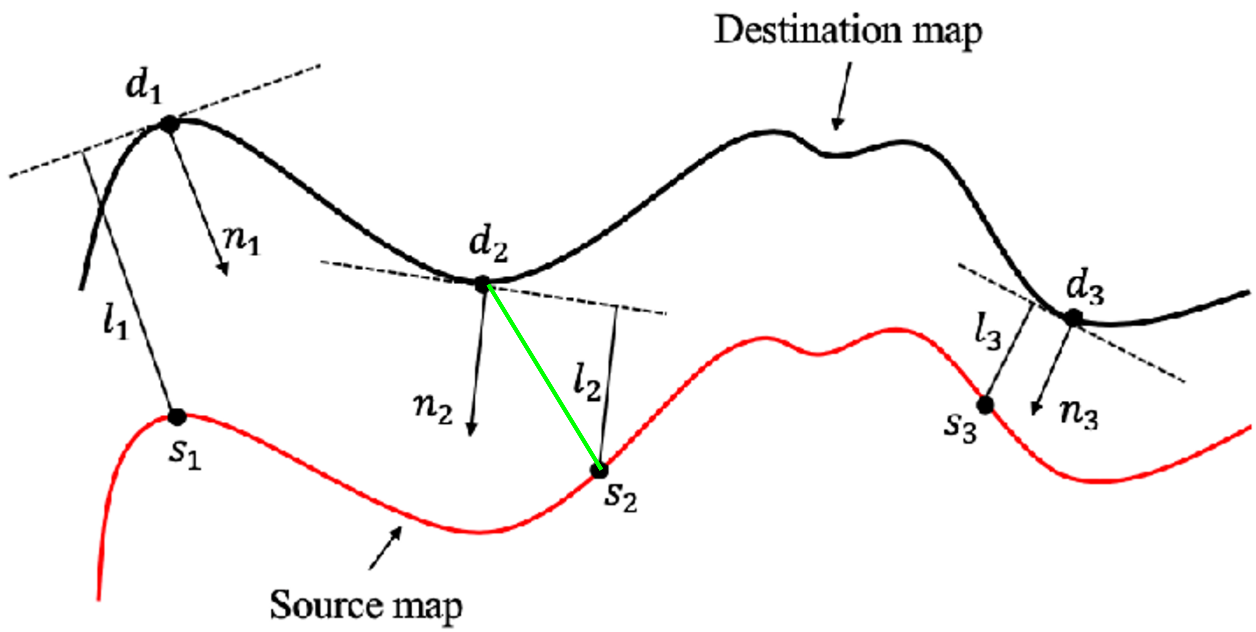


Camera Tracking

$$E(\mathbf{T}) = \sum_{(\mathbf{p},\mathbf{q})\in\mathcal{K}} \left((\mathbf{p} - \mathbf{T}\mathbf{q}) \cdot \mathbf{n}_{\mathbf{p}}\right)^2$$

Often use **Point-to-Plane loss [Rusinkiewicz 2001]**

Converges Faster

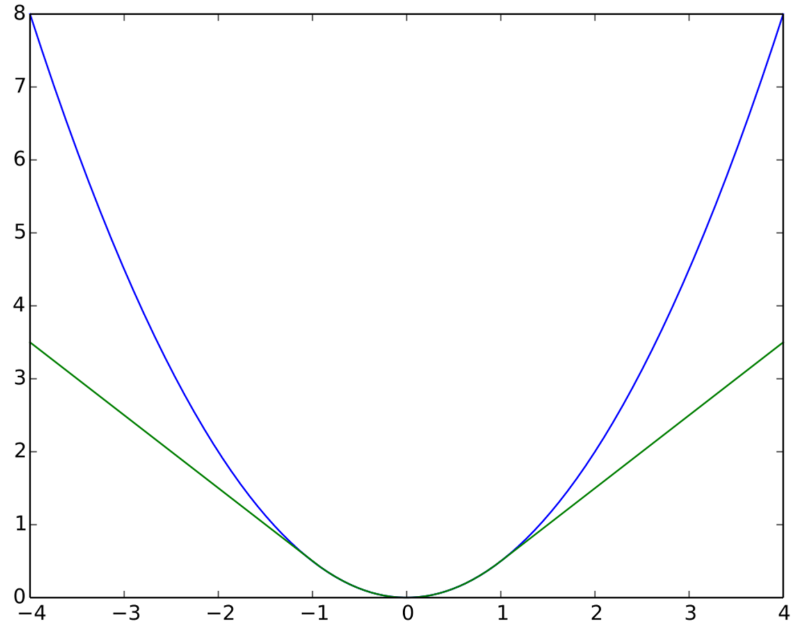


Rusinkiewicz, Szymon, and Marc Levoy. "Efficient variants of the ICP algorithm." *Proceedings Third International Conference on 3-D Digital Imaging and Modeling*. IEEE, 2001.

Camera Tracking

$$E(\mathbf{T}) = \sum_{(\mathbf{p},\mathbf{q})\in\mathcal{K}} \left((\mathbf{p} - \mathbf{T}\mathbf{q}) \cdot \mathbf{n}_{\mathbf{p}}\right)^2$$

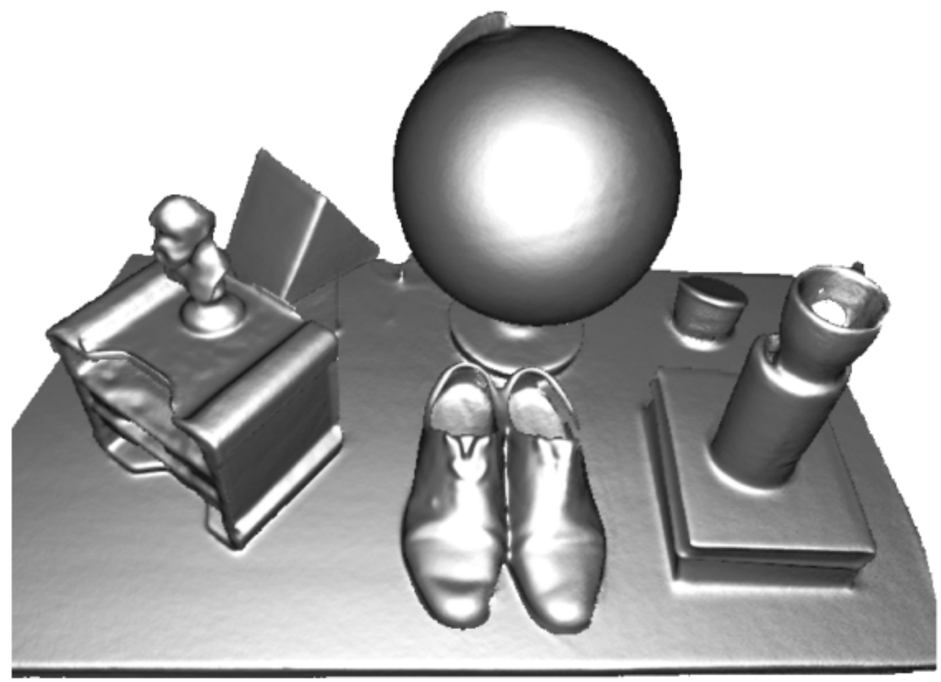
Robust Loss: [Huber 1964], reduce effects of outliers



Huber, Peter J. (1964). "Robust Estimation of a Location Parameter". *Annals of Statistics*. 53 (1): 73–101. doi:10.1214/aoms/1177703732. JSTOR 2238020.

Dense 3D Reconstruction (KinectFusion 2011)

- Signed Distance Function (SDF)
- Raycasting
- SDF Fusion (Curless 1996)
- Tracking (Iterative Closest Point)



Applications

1. Appearance & Lighting Reconstruction

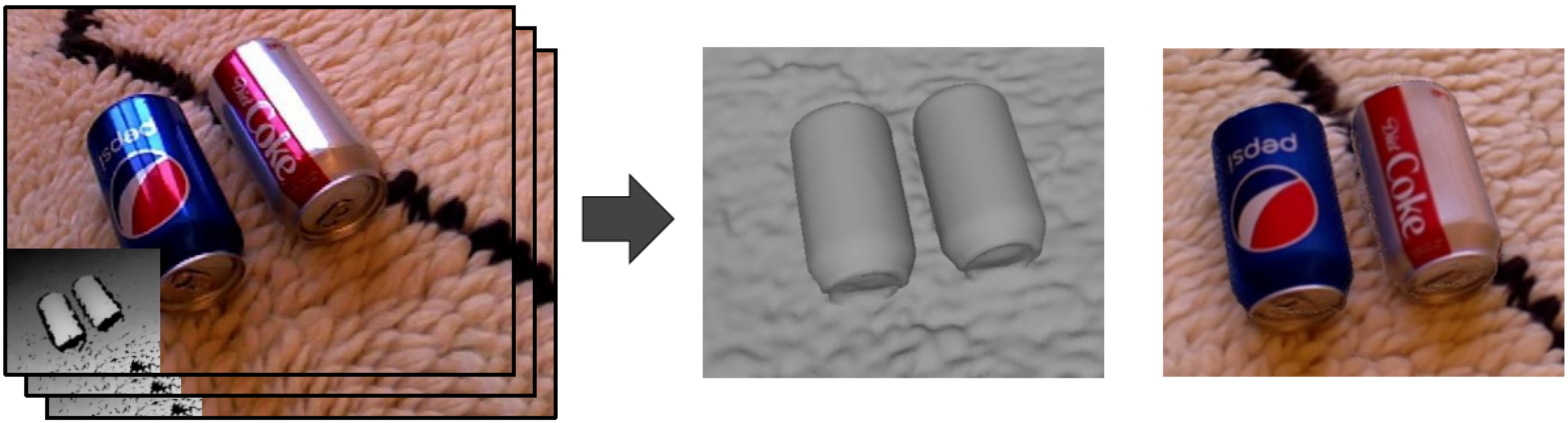
1. DeepSDF

Seeing the World in a Bag of Chips

Jeong Joon Park, Aleksander Holynski, Steve Seitz

University of Washington

Seeing the World in a Bag of Chips

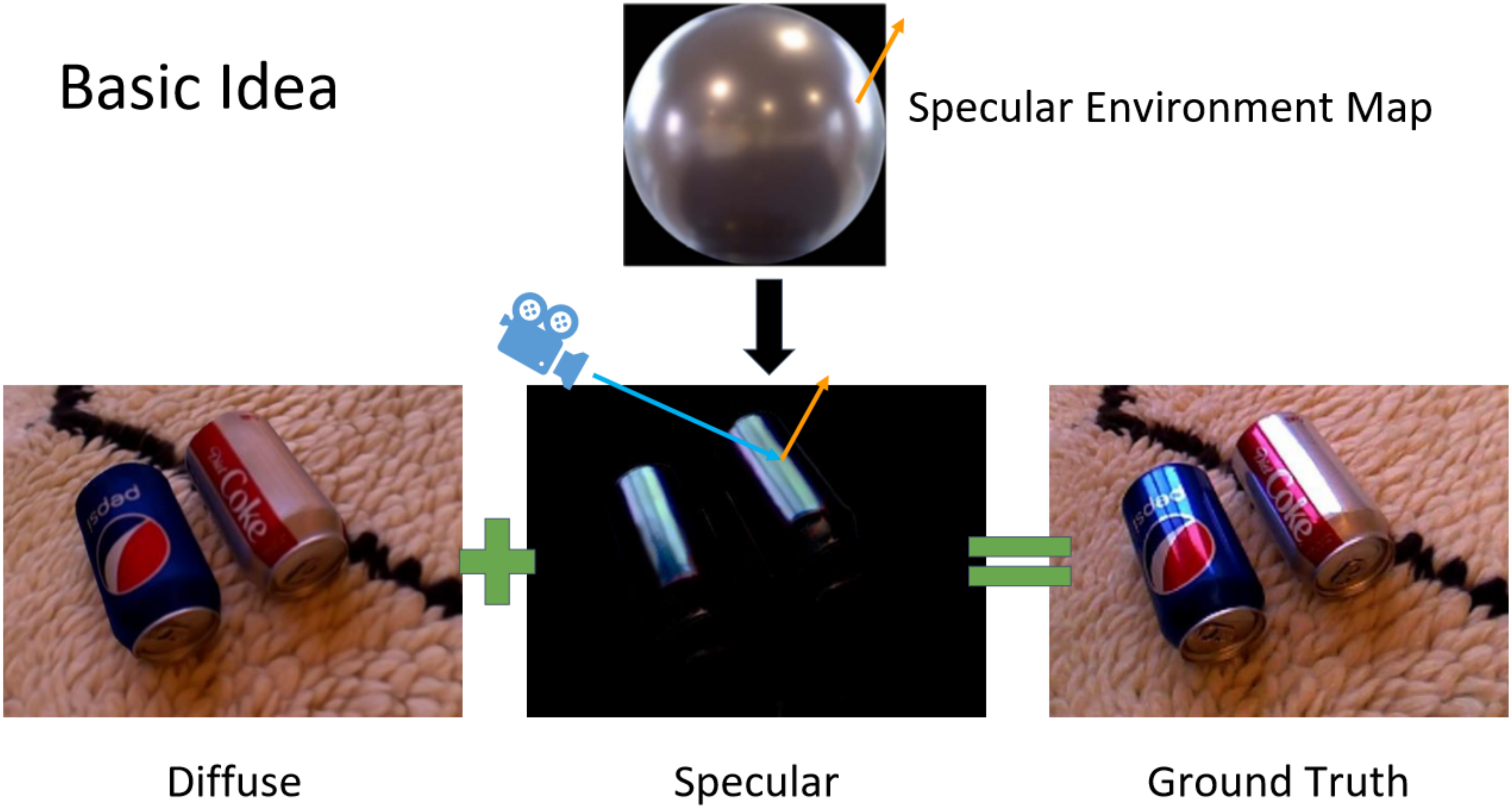


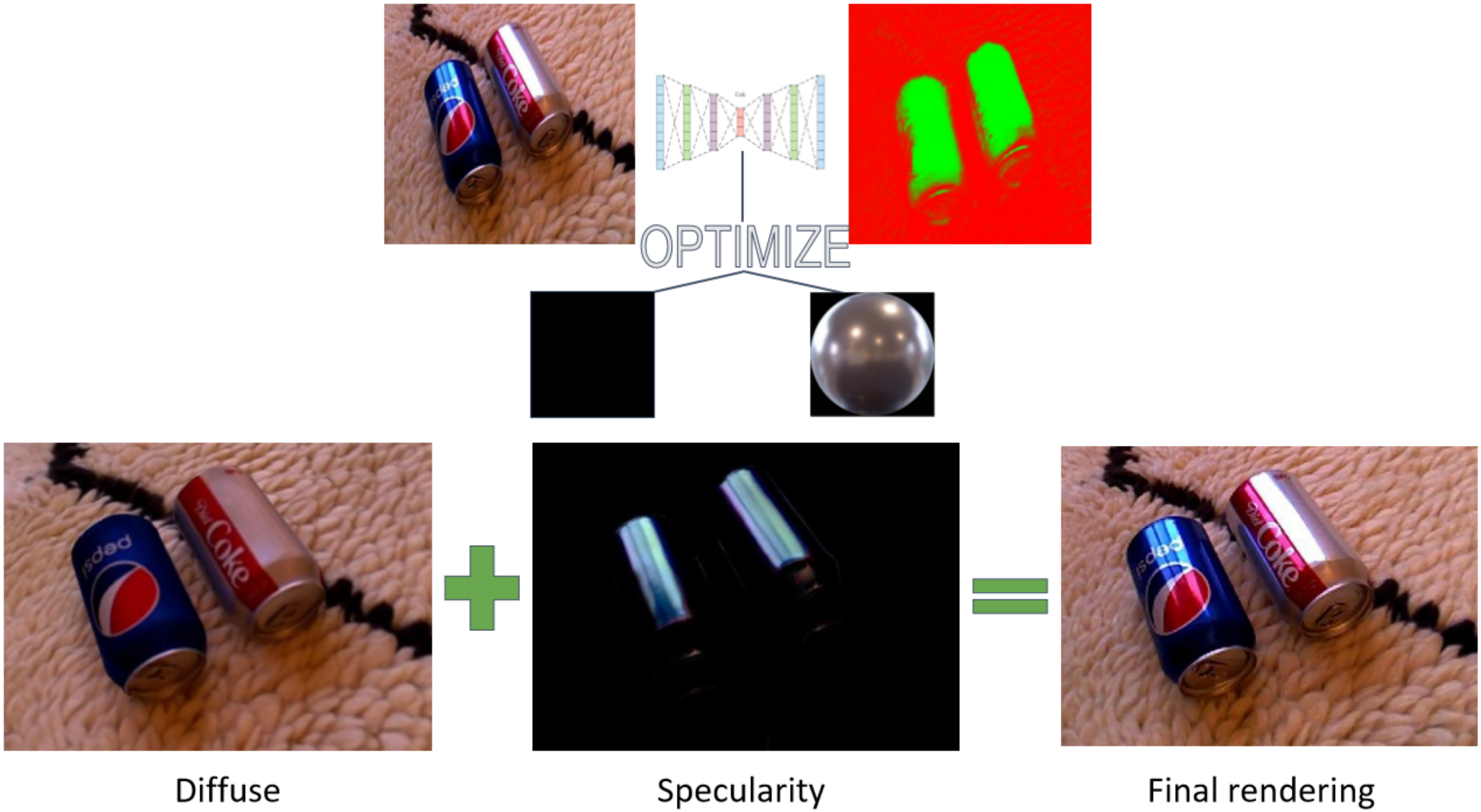
Input: RGBD Video

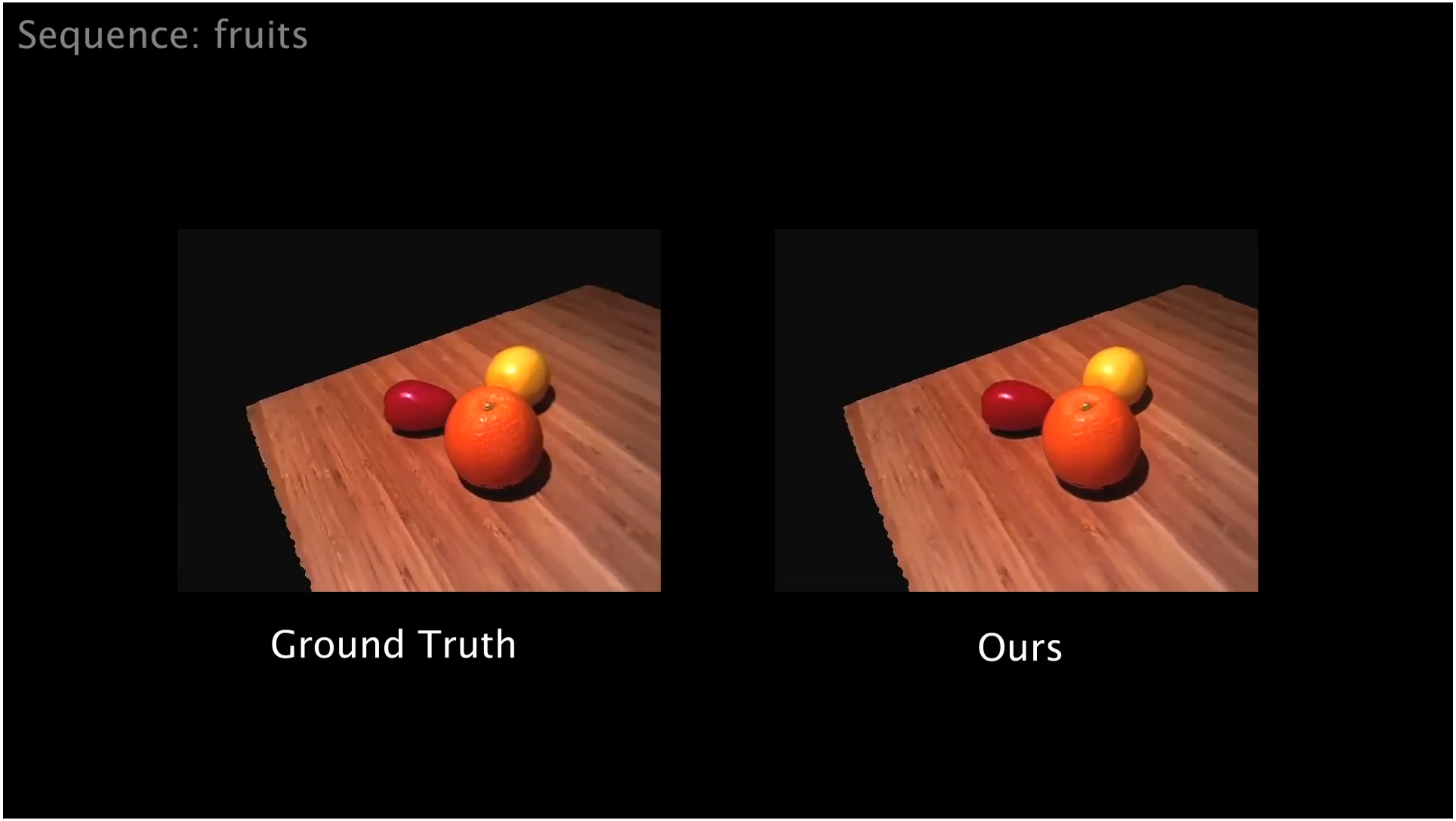
Geometry
(KinectFusion)

Diffuse Texture
(Park et al.)

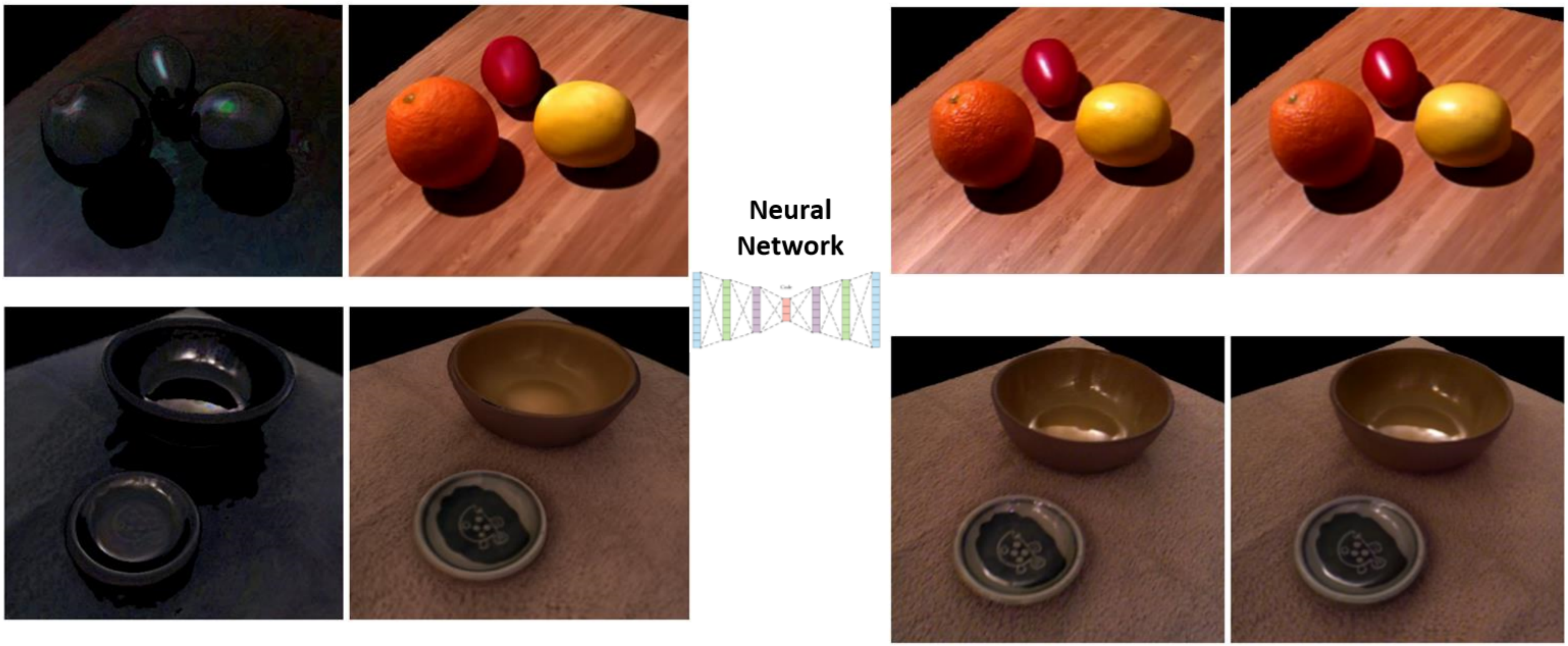
Basic Idea




OPTIMIZE
Diffuse
Specularity
Final rendering


Sequence: fruits
Ground Truth
Ours

Neural Rendering



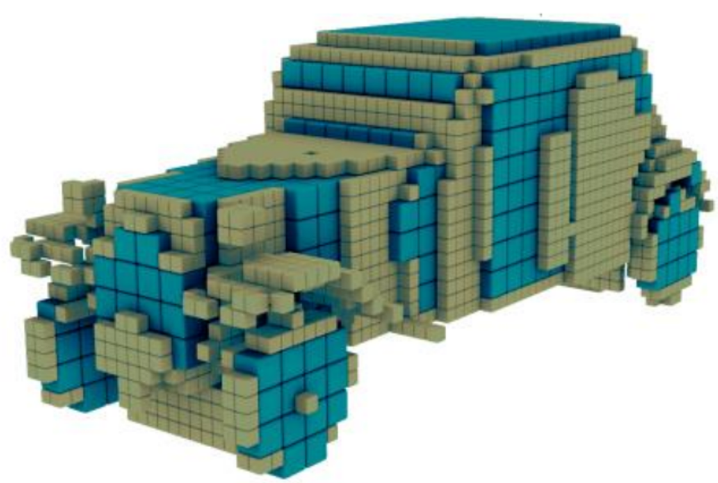
Specularity (from recovered env maps) Diffuse Texture Neural Rendering Ground Truth

DeepSDF: Learning Continuous SDFs for Shape Representation

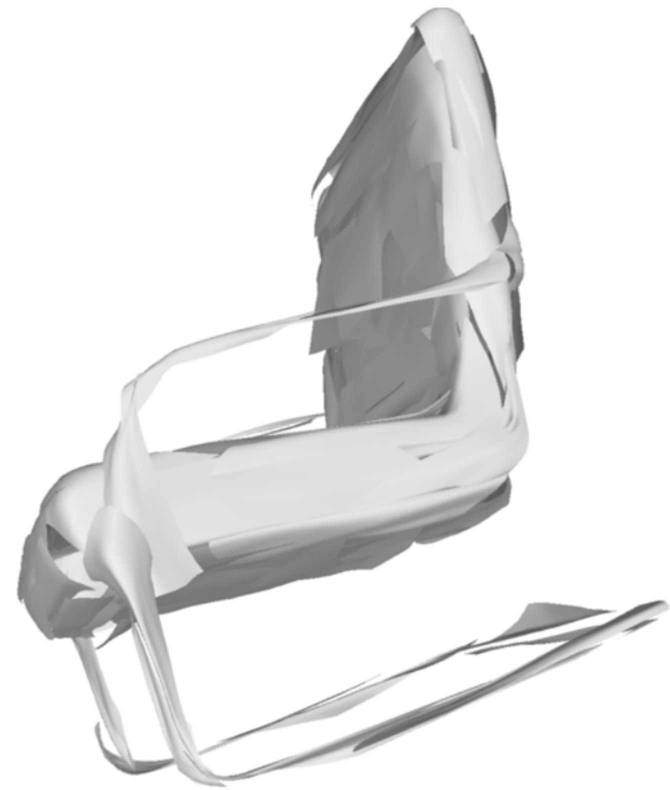
Jeong Joon Park, Peter Florence, Julian Straub,
Richard Newcombe, Steven Lovegrove

Representations for 3D Deep Learning

Wu et al. 2016

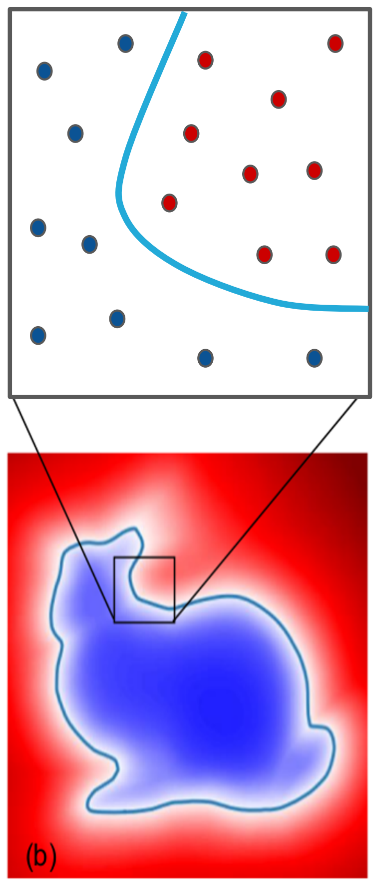


Tatarchenko et al. 2015

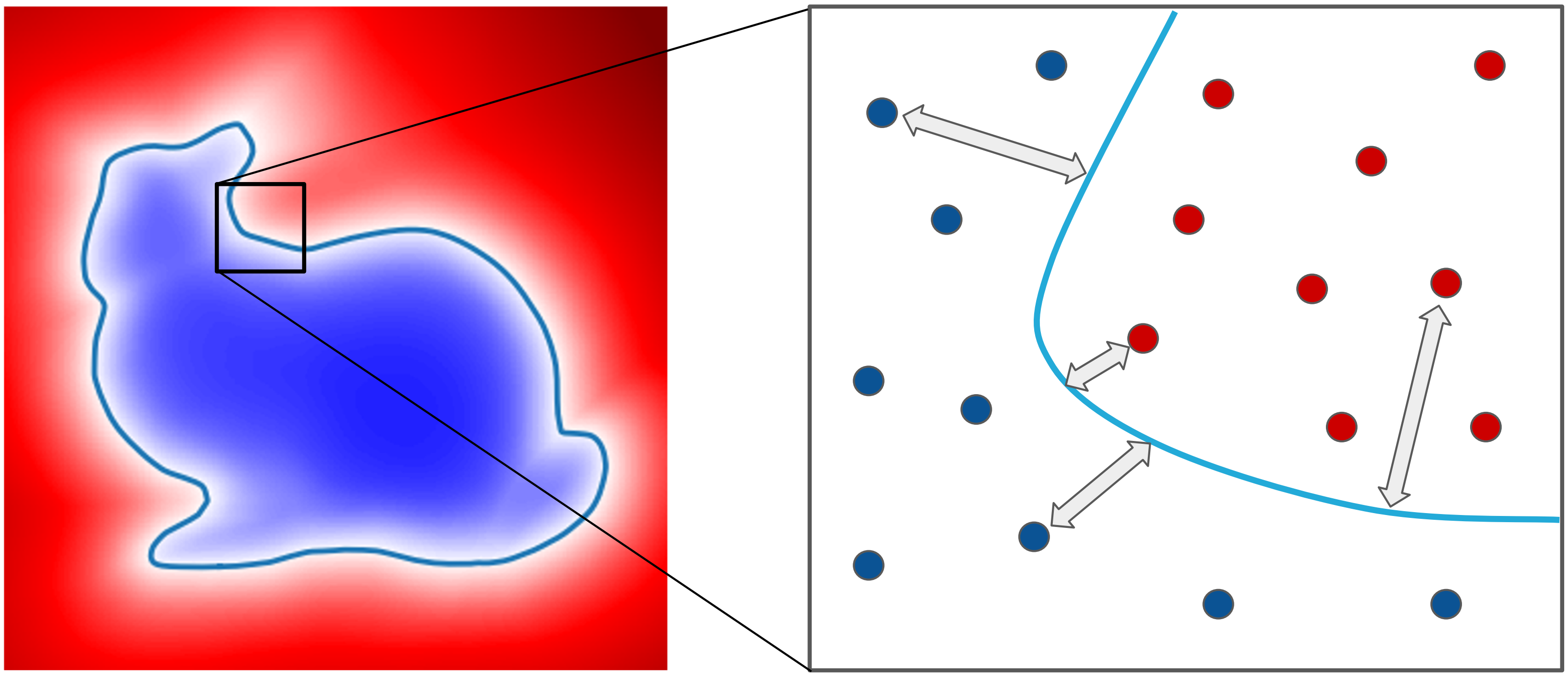


Groueix et al. 2018

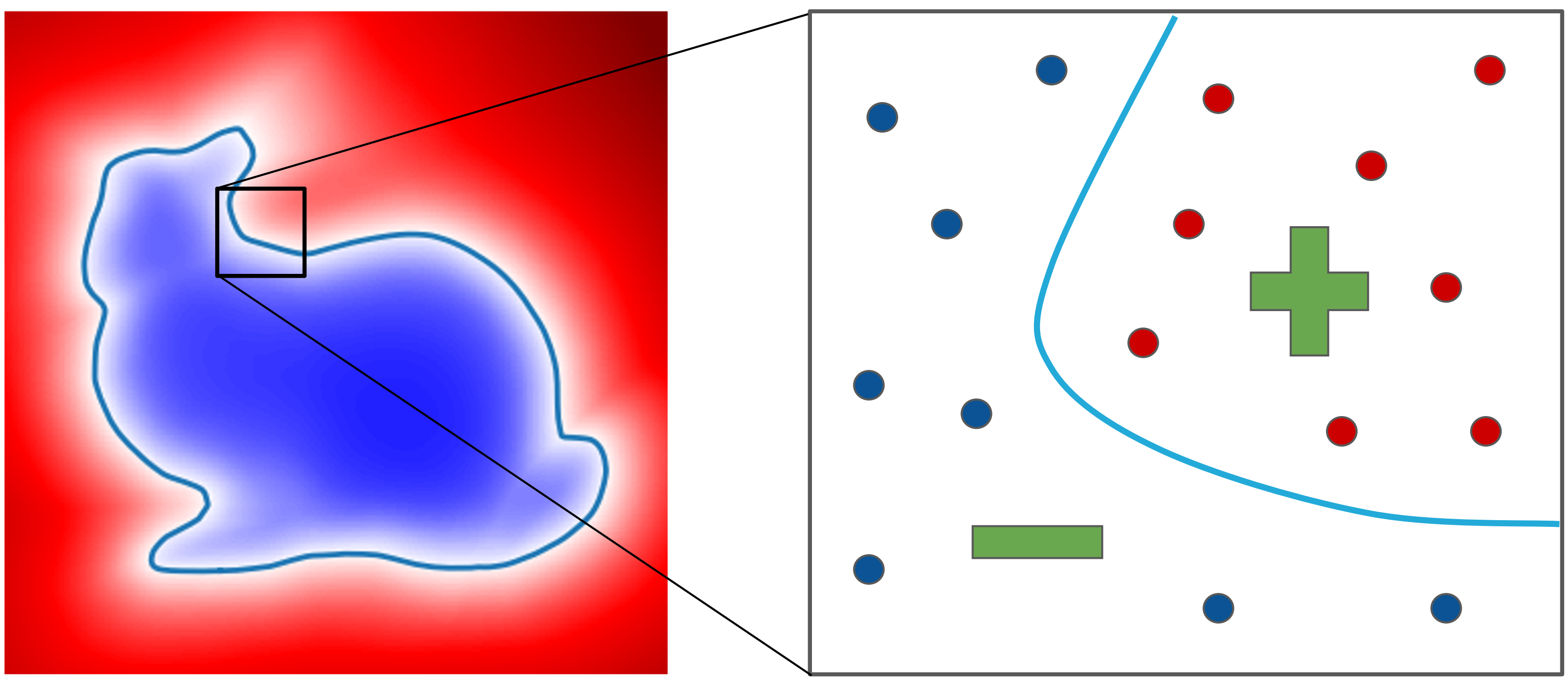
Key Idea: Directly regress SDF



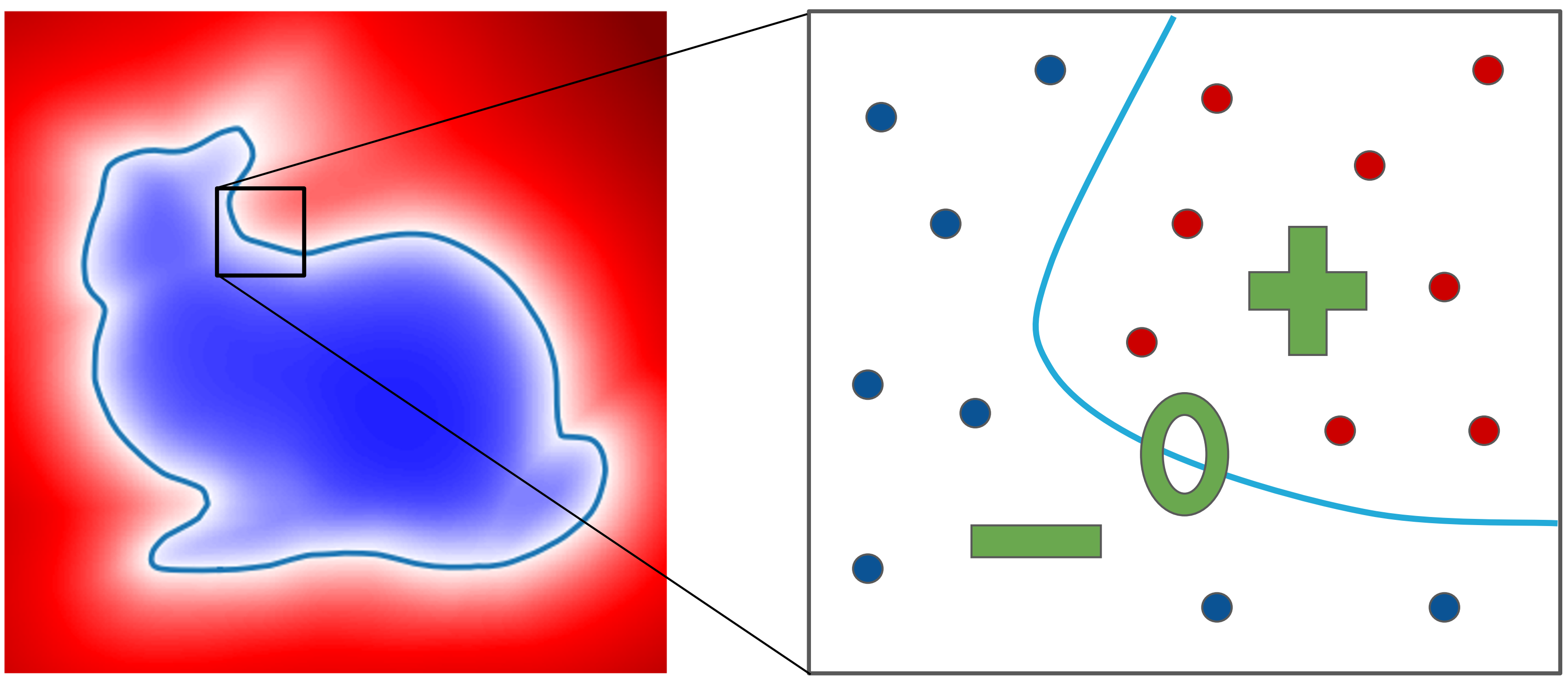
Signed Distance Function



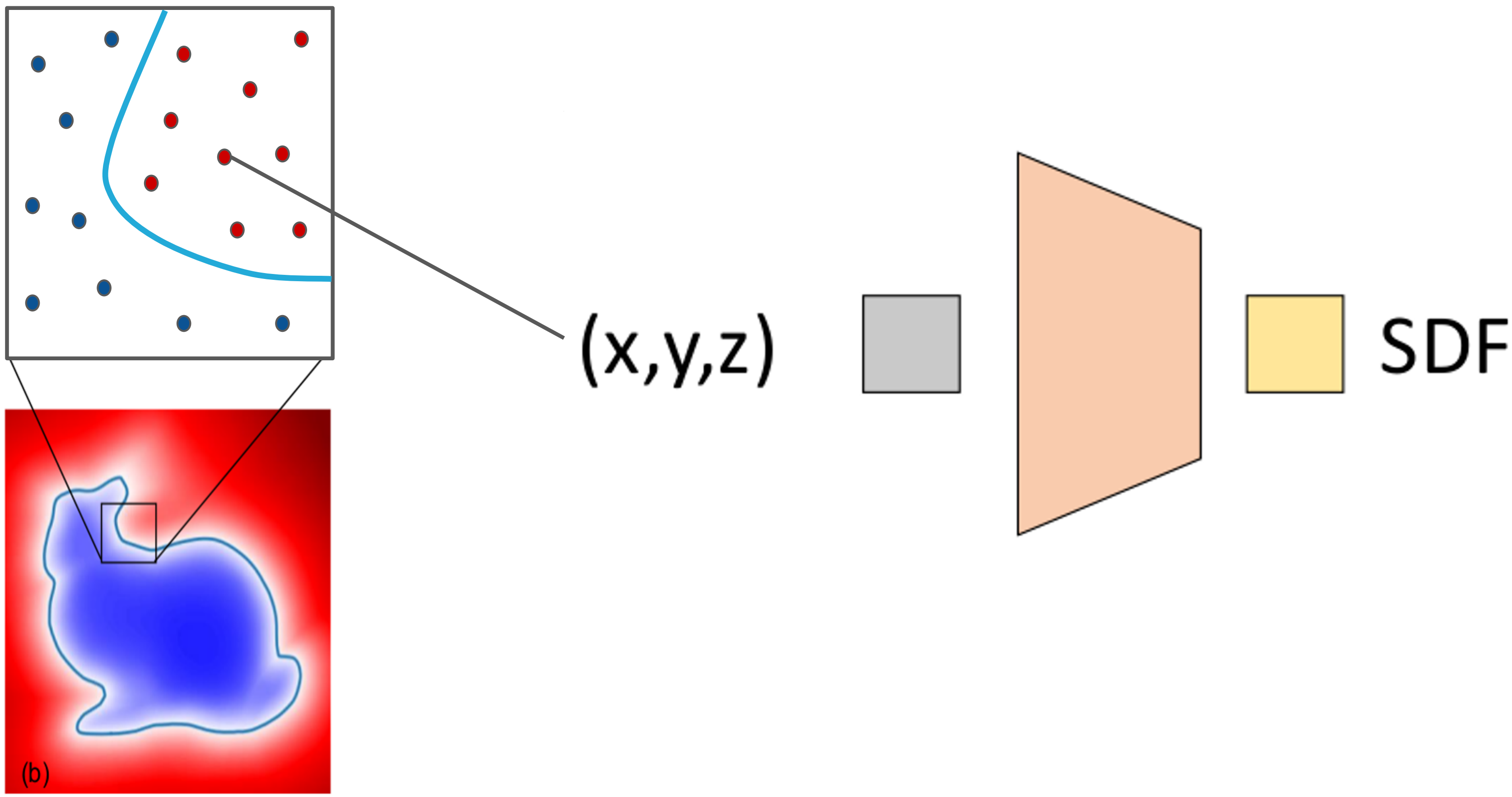
Signed Distance Function



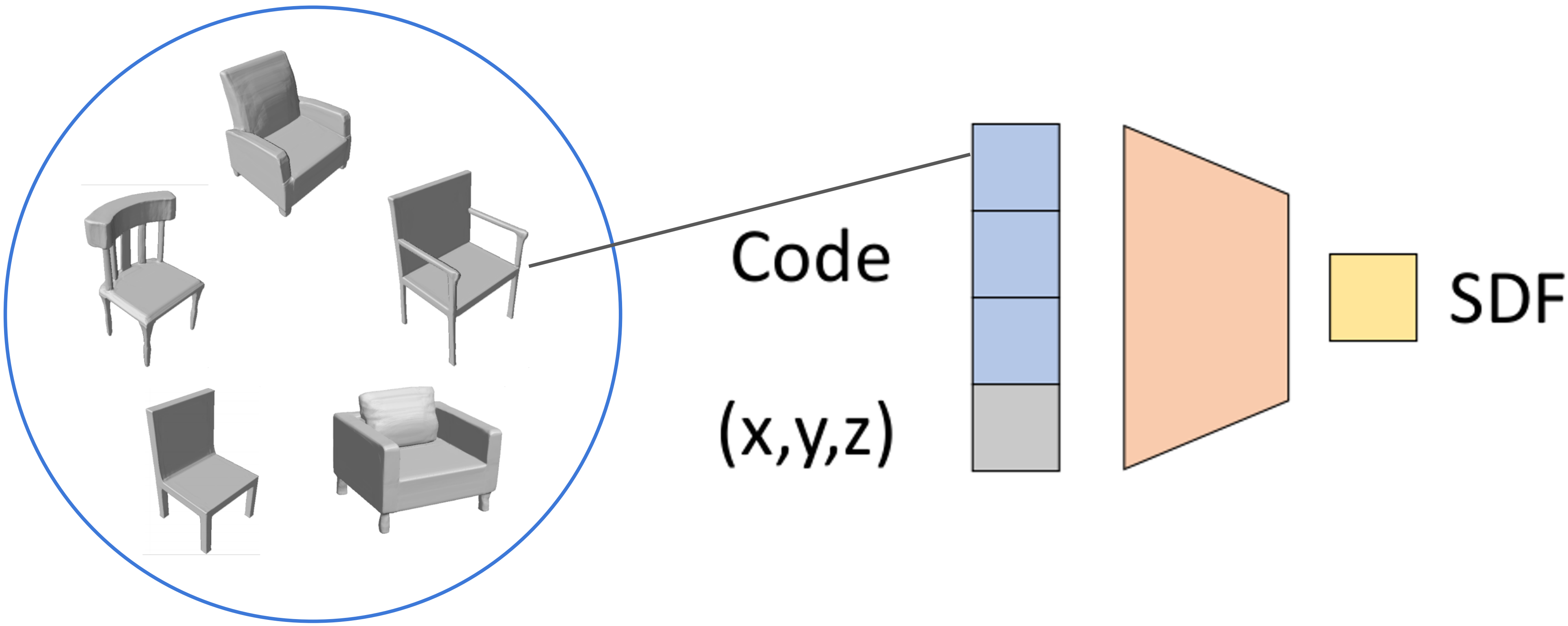
Signed Distance Function

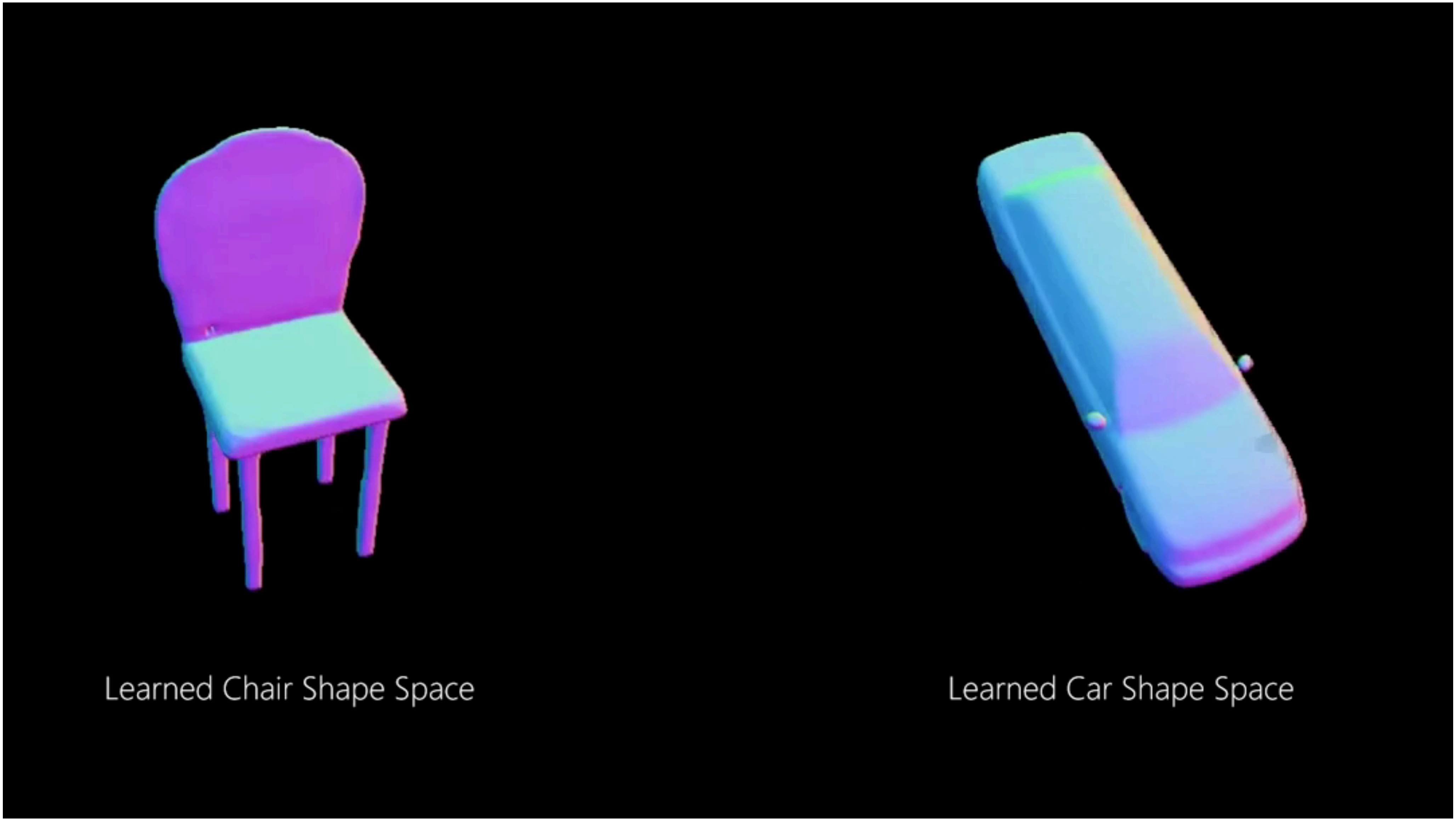


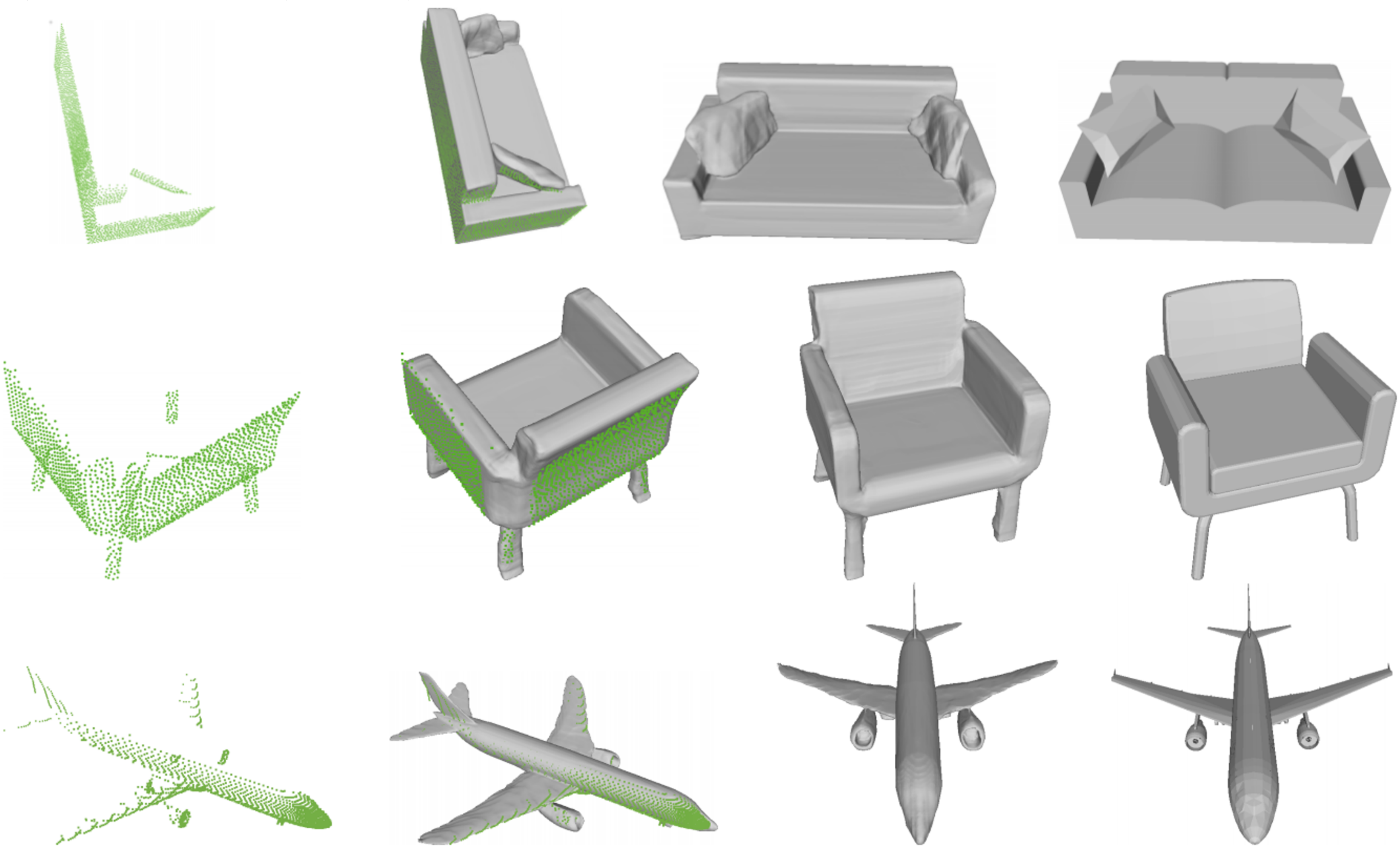
Direct Regression of SDF



Coding Multiple Shapes




Learned Chair Shape Space
Learned Car Shape Space



(a) Input Depth **(b)** Completion (ours) **(c)** Second View (ours) **(d)** Ground truth

References

1. KinectFusion: https://www.microsoft.com/en-us/research/wp-content/uploads/2016/02/ismar2011.pdf
2. Curless 1996: https://graphics.stanford.edu/papers/volrange/volrange.pdf
3. Seeing the World in a Bag of Chips: https://youtu.be/9t_Rx6n1HGA
4. DeepSDF: https://arxiv.org/abs/1901.05103
5. Tanner Schmidt: https://courses.cs.washington.edu/courses/cse571/16au/slides/10-sdf.pdf
6. CuriousInventor: https://www.youtube.com/watch?v=uq9SEJxZiUg
7. Carleton lecture note: http://www.cs.carleton.edu/cs_comps/0405/shape/marching_cubes.html
8. ICP: https://www.comp.nus.edu.sg/~lowkl/publications/lowk_point-to-plane_icp_techrep.pdf