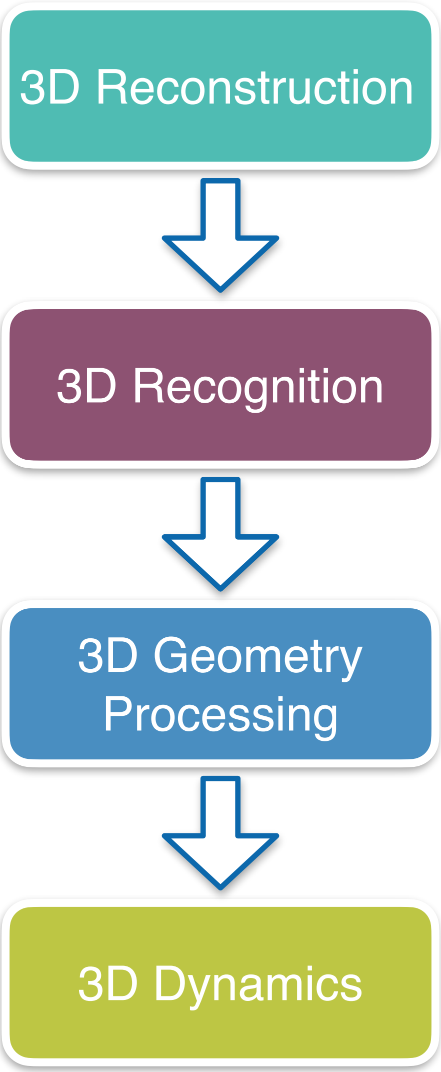
Teaching Plan

Starting from this lecture:

- Application-based lecture organization
- Go over important 3D learning techniques
- Introduce key technique points but not all the details of a DL pipeline

Syllabus

- High-level organization



3D Reconstruction

3D Recognition

3D Geometry Processing

3D Dynamics

L6: Learning-based Multi-View Stereo

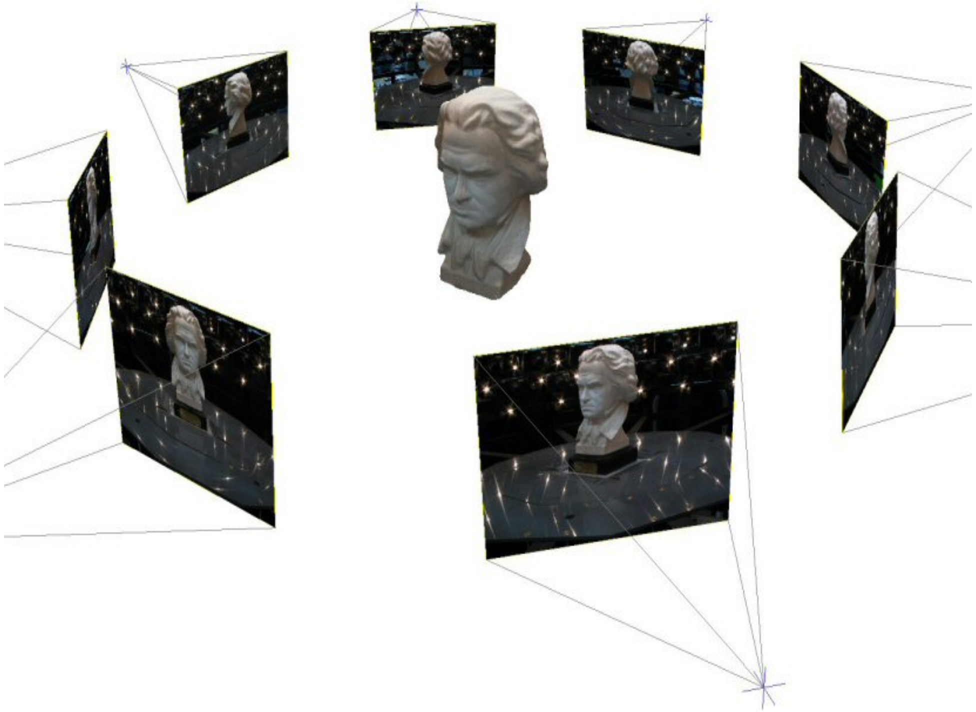
Hao Su

Agenda

- Photometric Consistency
- A First Pipeline: Deep Volumetric Stereo
- Key Techniques
    - Adaptive Space Sampling
    - Depth-Normal Consistency
- Appearance Information Capturing

Multi-View Stereo (MVS)

Reconstruct the dense 3D shape from a set of **images** and **camera parameters**



1. Goldlucke et al. “A Super-resolution Framework for High-Accuracy Multiview Reconstruction”

Applications of MVS


AR/VR


Autonomous Driving


Inverse Engineering


Robot Manipulation


Remote Sensing

Image source:
1. https://wisdomeweb.com/whats-a-lidar-sensor-and-why-it-on-the-iphone-12-pro/
2. https://cloudblogs.microsoft.com/industry-blog/wp-content/uploads/industry/2019/06/
3. https://www.tecnamachines.com/images/
4. https://scienceinfo.net/data-images/thumbs/
5. https://www.altizure.com/

Photometric Consistency

Triangulation

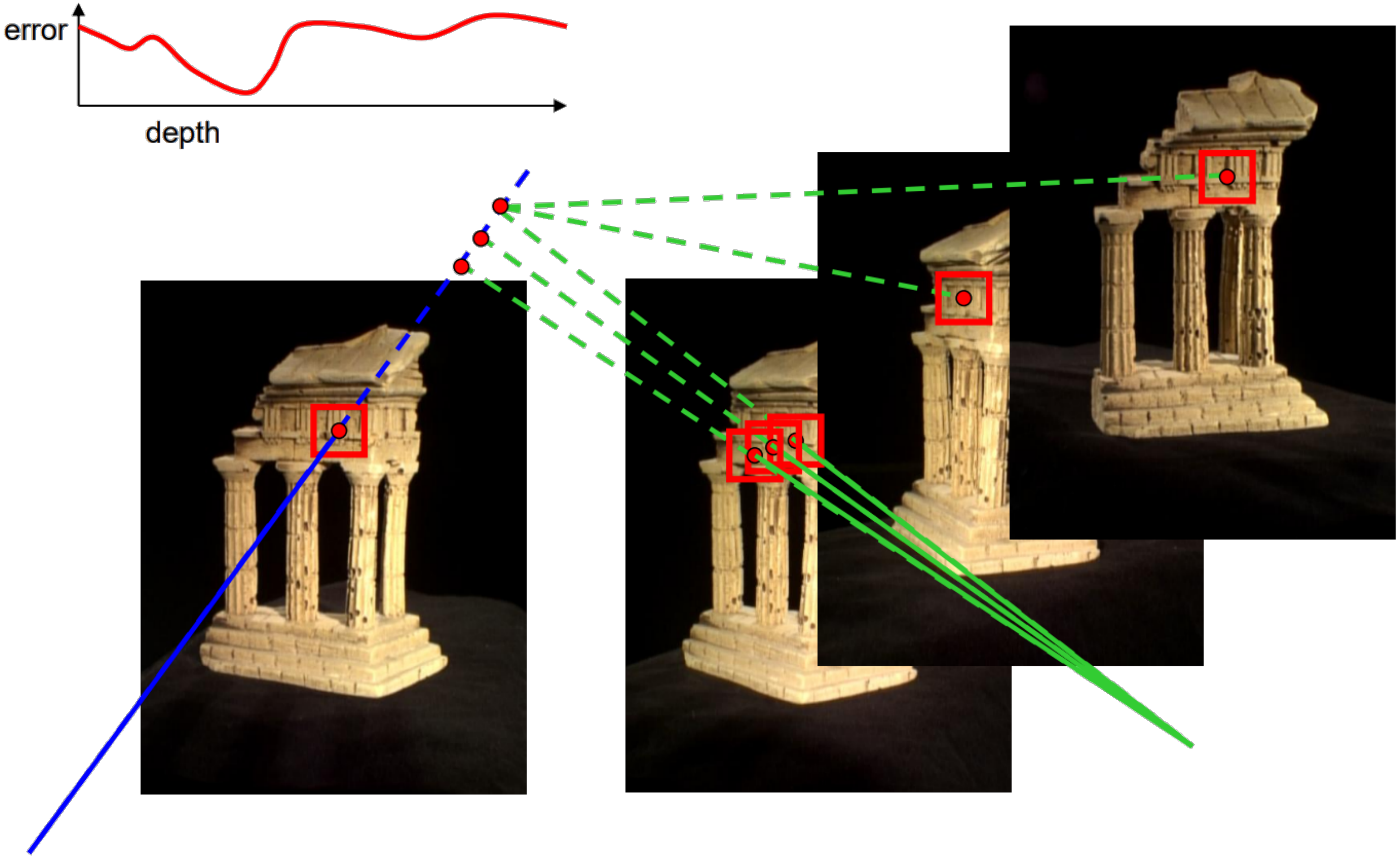
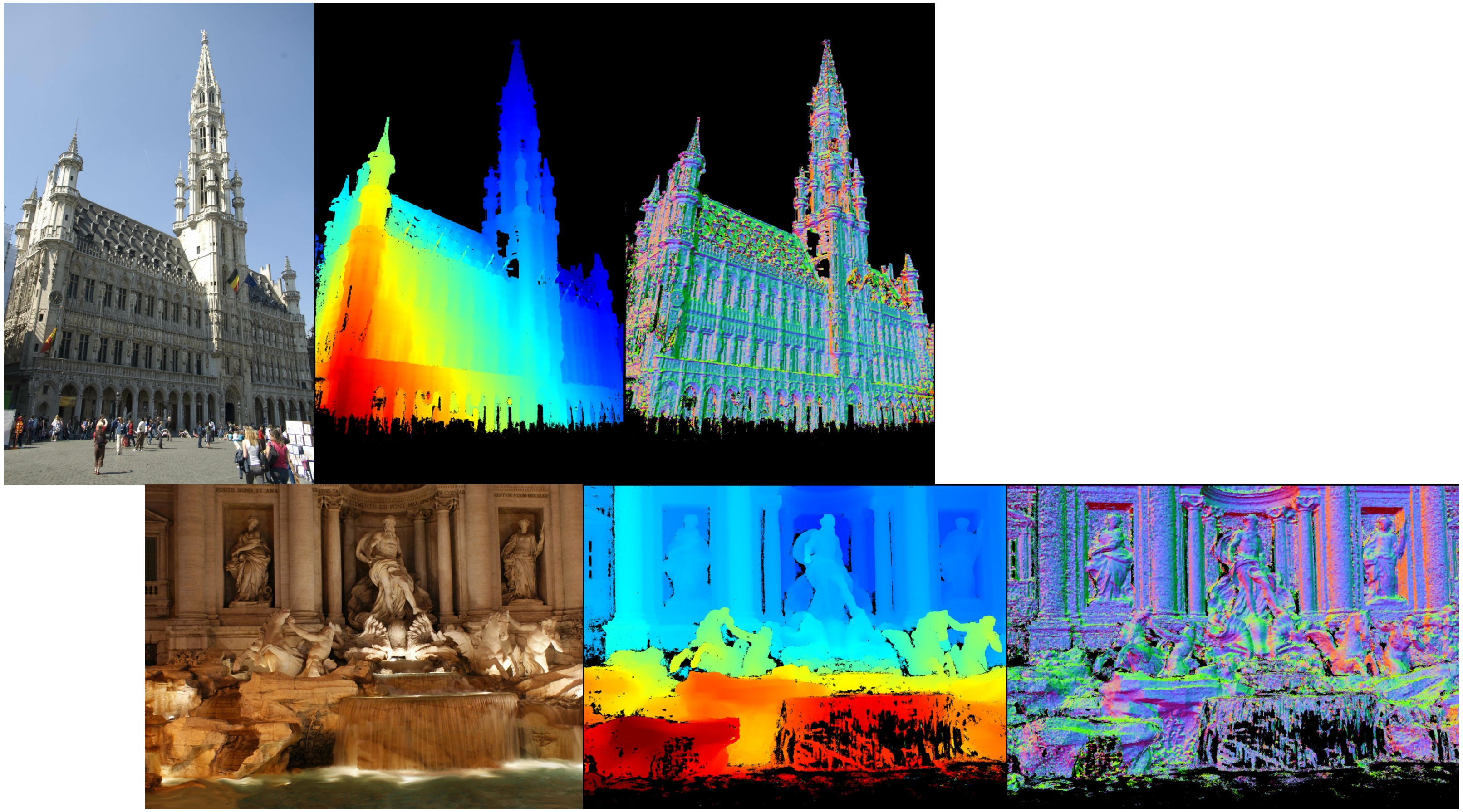


Image source: UW CSE455

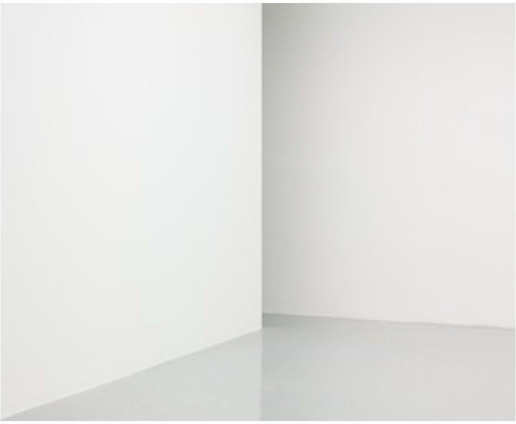
Stereo from Community Photo Collections



https://colmap.github.io/

Schönberger, Johannes L., Enliang Zheng, Jan-Michael Frahm, and Marc Pollefeys. "Pixelwise view selection for unstructured multi-view stereo." In European Conference on Computer Vision, pp. 501-518. Springer, Cham, 2016.

Limitation of Classical MVS



Textureless Area

Reflection
/Transparency

Repetitive
patterns

Yi Ma, An Introduction to Holistic 3D Reconstruction, ICCV 2019

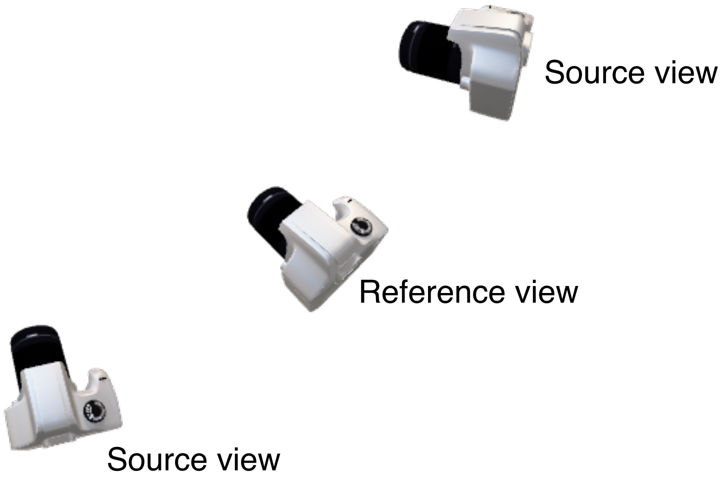
Learning-based MVS

Learned feature → more robust matching

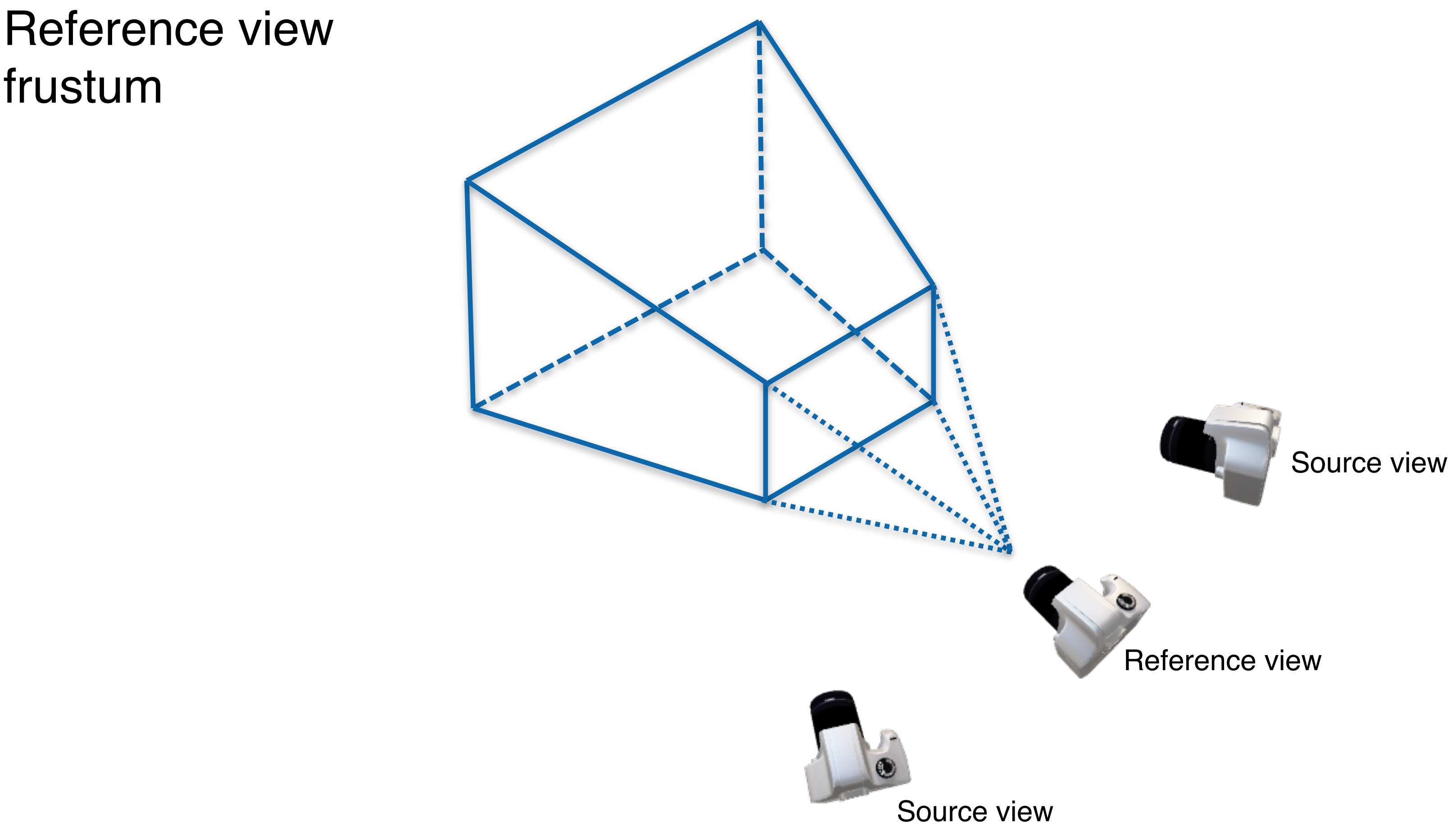
Shape prior → more complete reconstruction

A First Pipeline:
Deep Volumetric Stereo

Volumetric Stereo

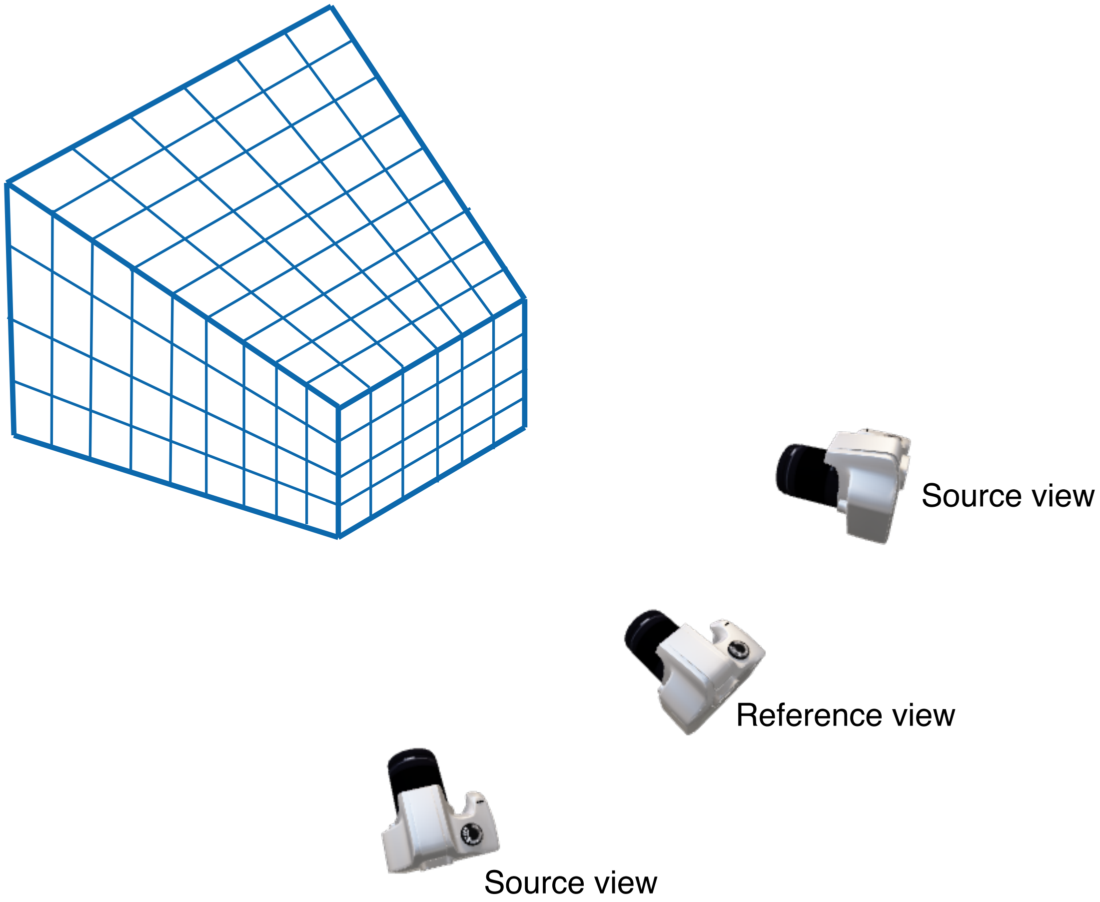


Volumetric Stereo



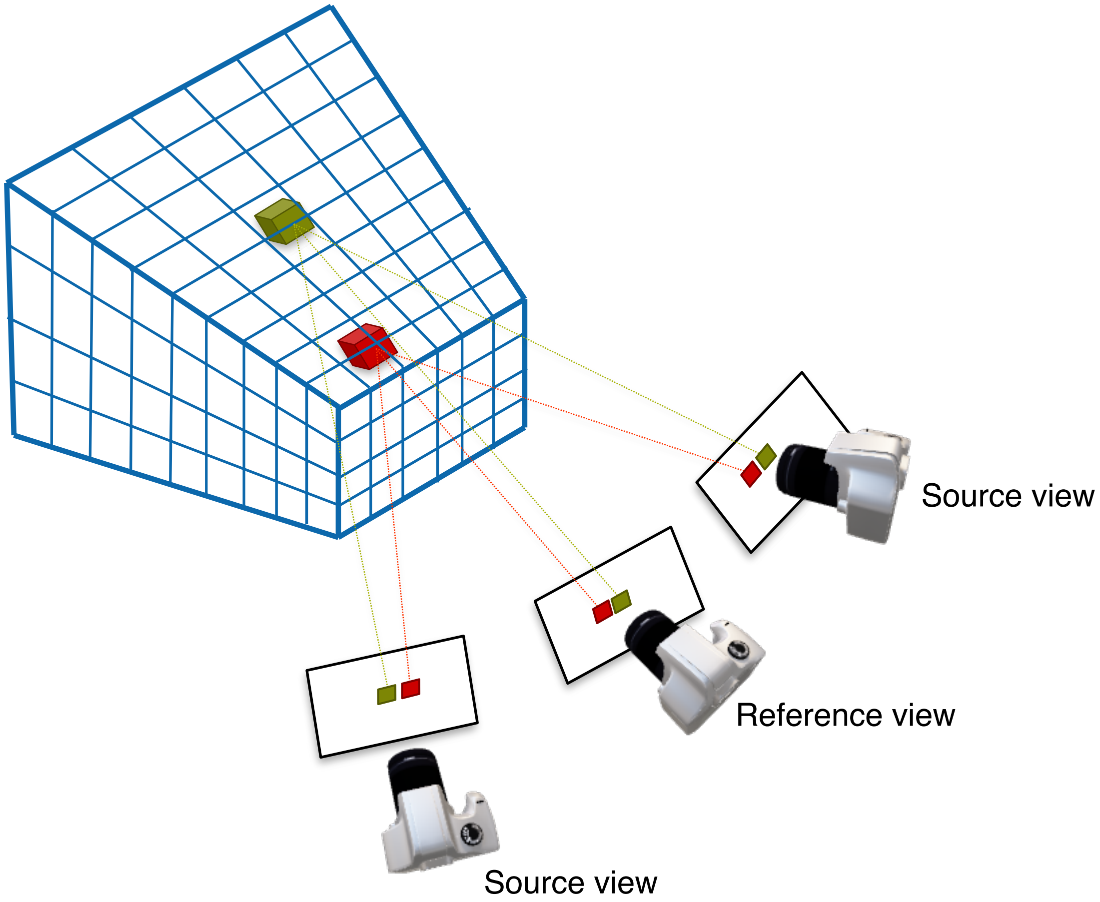
Volumetric Stereo

Reference view
frustum voxelization



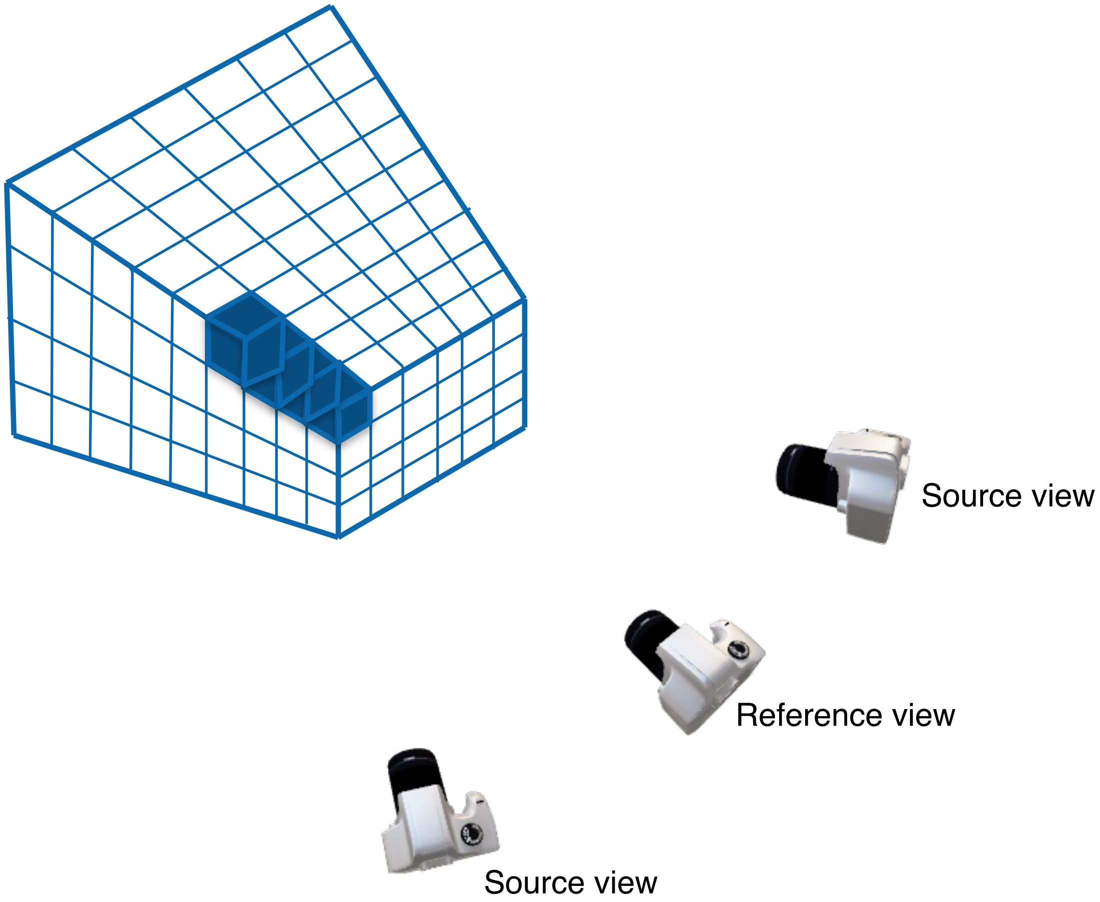
Volumetric Stereo

Image feature warping



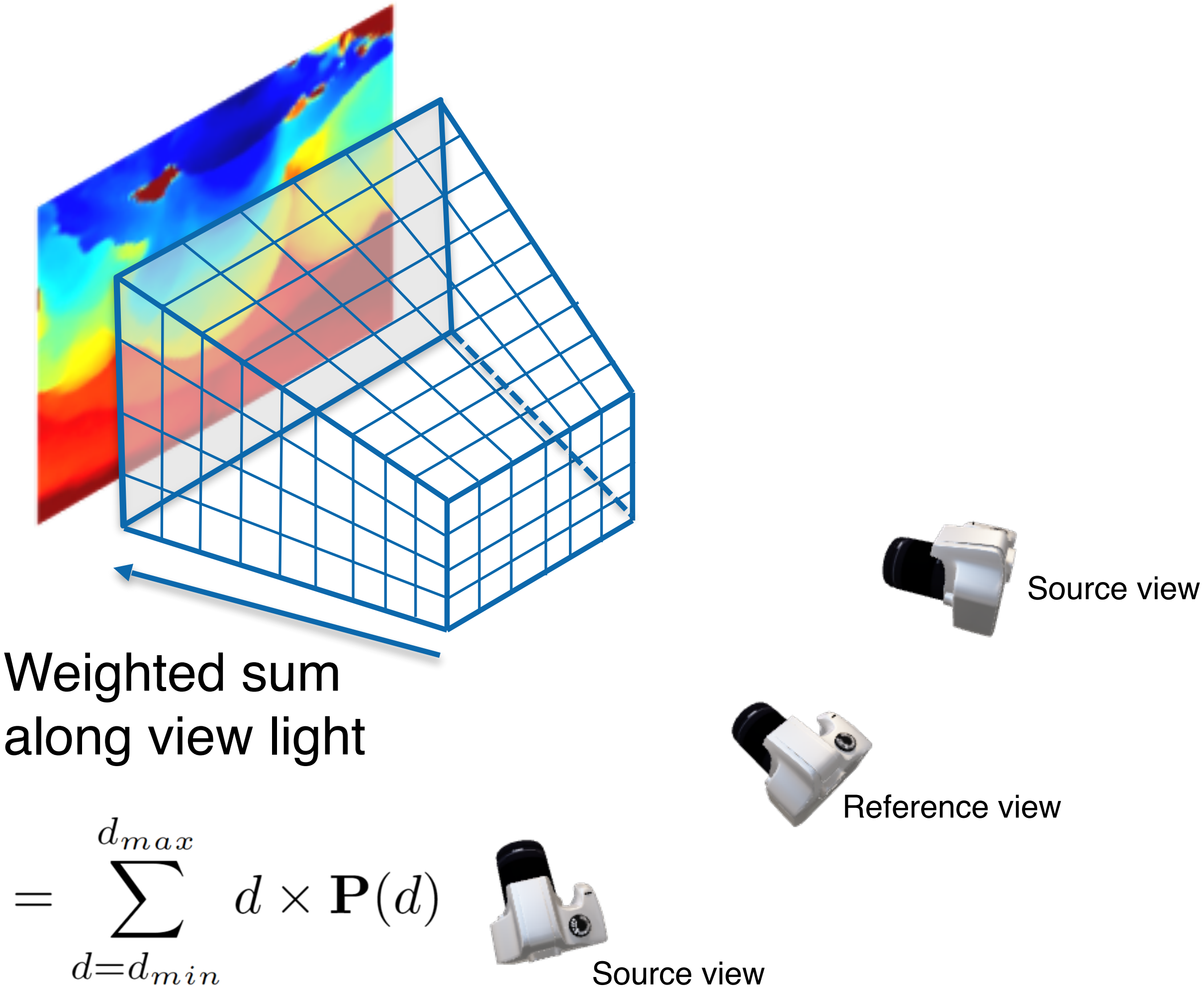
Volumetric Stereo

3D CNNs



Volumetric Stereo

Reference view
depth prediction



Weighted sum
along view light

$$\mathbf{D} = \sum_{d=d_{min}}^{d_{max}} d \times \mathbf{P}(d)$$

Reference-View Depth Loss

$$Loss = \sum_{p \in \mathbf{P}_{\text{valid}}} \left\| d\left(p\right) - \hat{d}\left(p\right) \right\|_1$$

Valid pixels

GT depth

Depth prediction

Issues

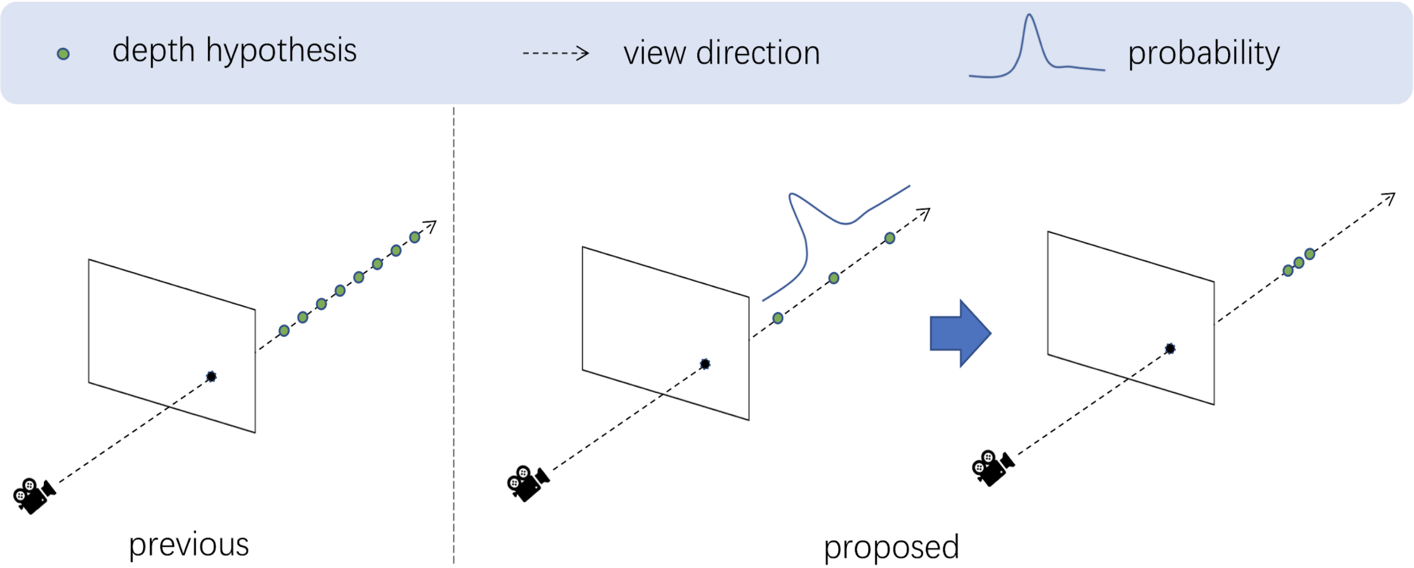
- Quality
- Speed
- Flying points when there is abrupt depth change
- Lacking appearance information

Perspectives for Improvement

- Adaptively sample the space near the surface
- Stronger loss function

Adaptive Space Sampling

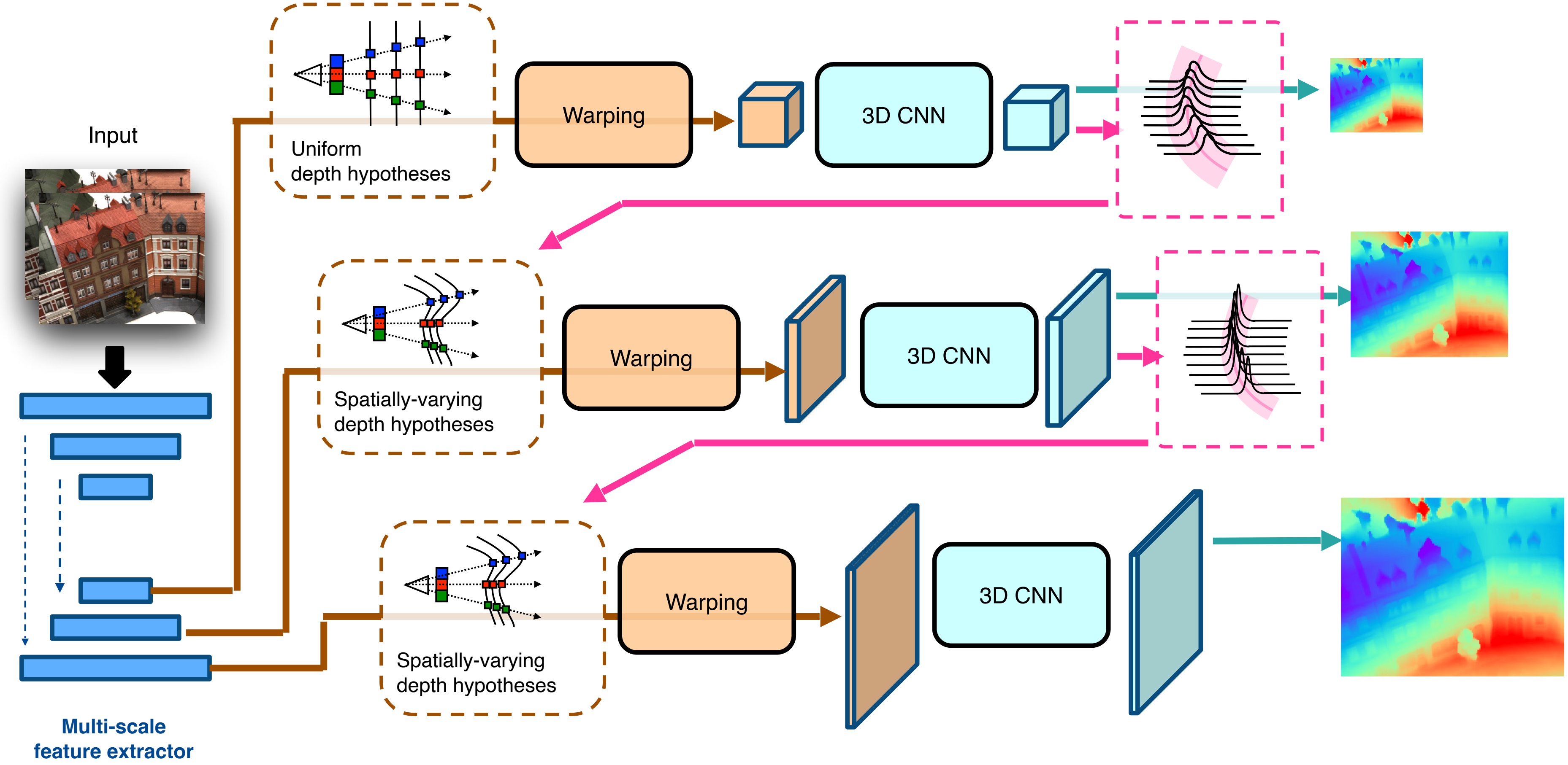
Coarse-to-fine Sampling



- Analyze per-pixel confidence intervals
- Narrow down the sampling range based on uncertainty

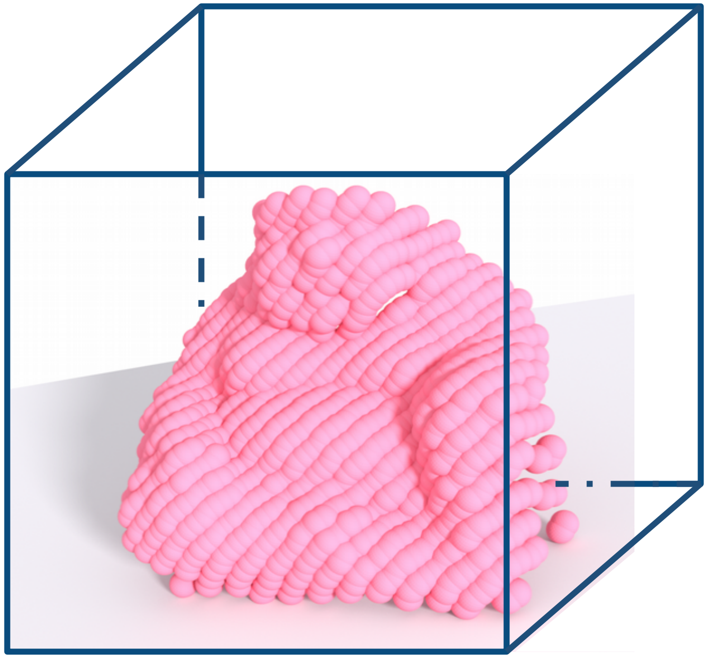
Cheng, Shuo, Zexiang Xu, Shilin Zhu, Zhuwen Li, Li Erran Li, Ravi Ramamoorthi, and Hao Su. "Deep stereo using adaptive thin volume representation with uncertainty awareness." In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 2524-2534. 2020.

Cascaded Depth Prediction



Point-based Multi-View Stereo Network

Point cloud representation
- Suitable for sparse occupancy
- Memory-efficient

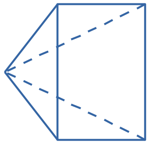


Chen, Rui, Songfang Han, Jing Xu, and Hao Su. "Point-based multi-view stereo network." In Proceedings of the IEEE International Conference on Computer Vision, pp. 1538-1547. 2019.

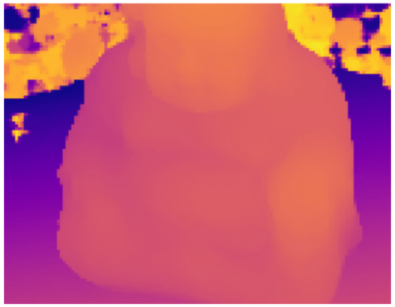
Read by yourself

Initial Point Cloud

Estimate **low-resolution** depth map with existing methods

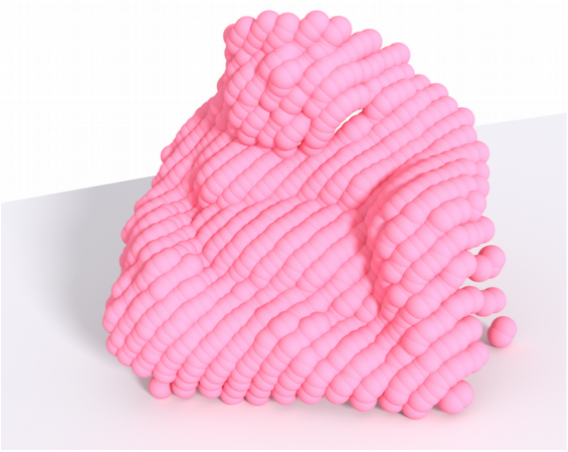


Reference camera



Coarse Depth map

Unprojection



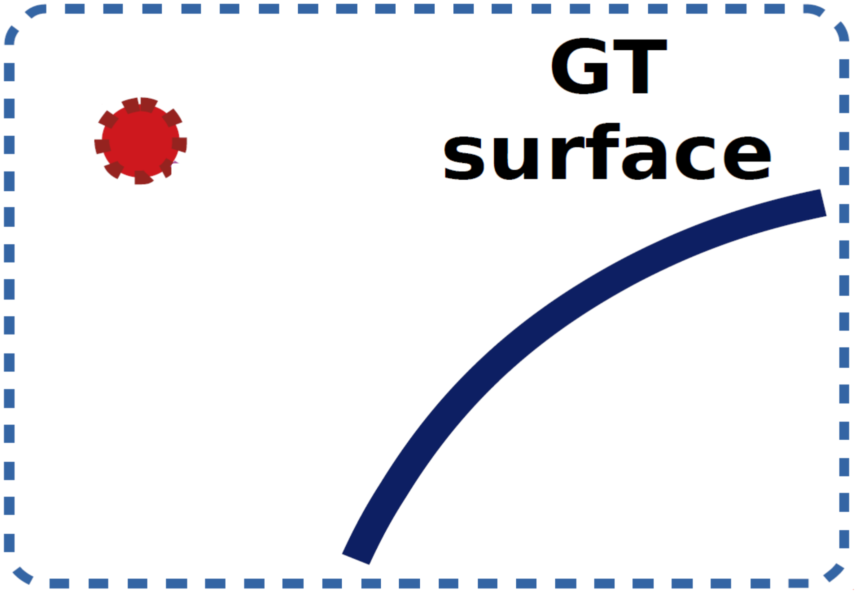
Initial point cloud

Read by yourself

Point Flow

Goal:

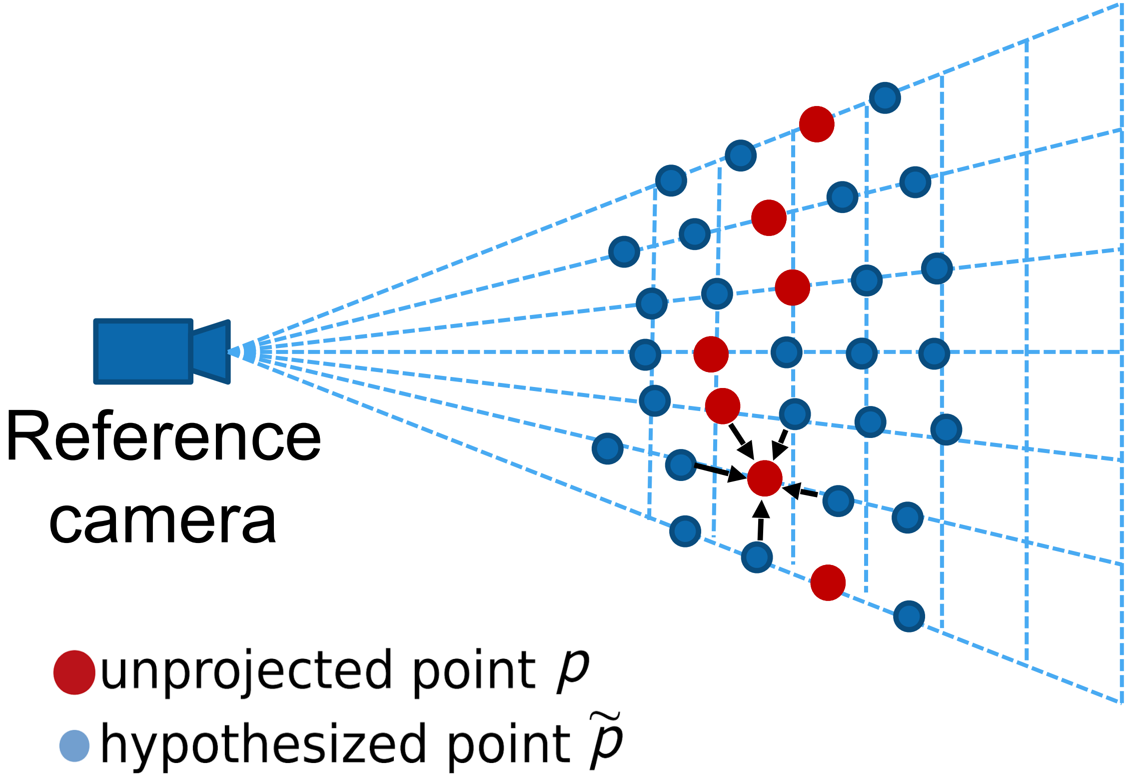
Refine the input depth map by moving the unprojected points along camera direction



Read by yourself

Flow Prediction

Flow prediction as expected offset



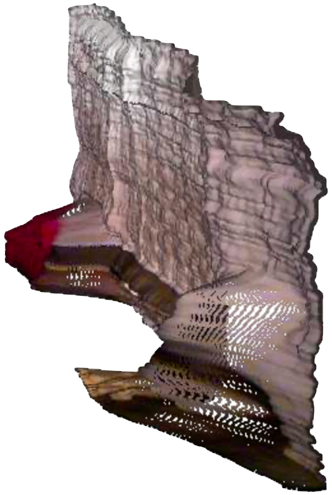
expected offset

$$\Delta d_p = \mathbf{E}(ks) = \sum_{k=-m}^{m} ks \times \text{Prob}(\tilde{\mathbf{p}}_k)$$

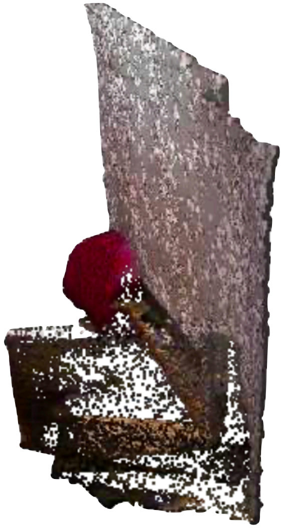
Read by yourself

Depth-Normal Consistency Loss

Depth Supervision Alone Does Not Give Smooth Surface



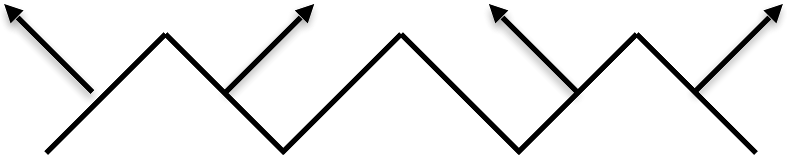
Prediction



Ground truth

How to Improve Surface Smoothness?

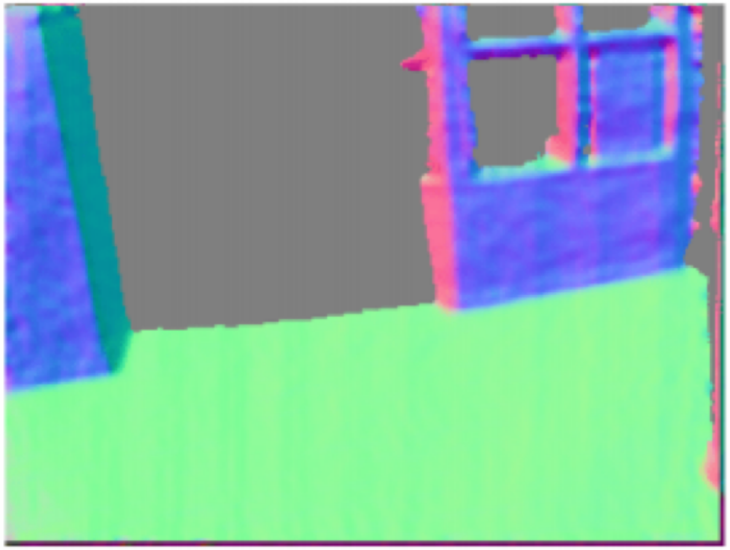
- **Key observation**: Surface smoothness is reflected by surface normal.



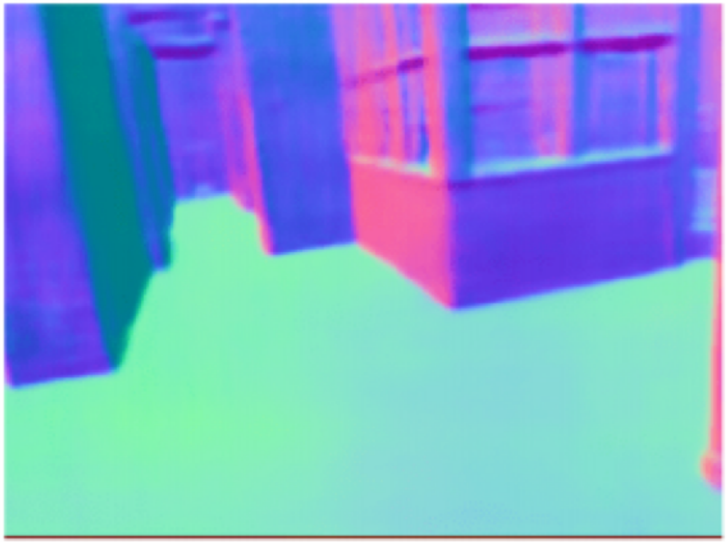
Rough surface

Plane surface

Observation: Normal Prediction is Easier



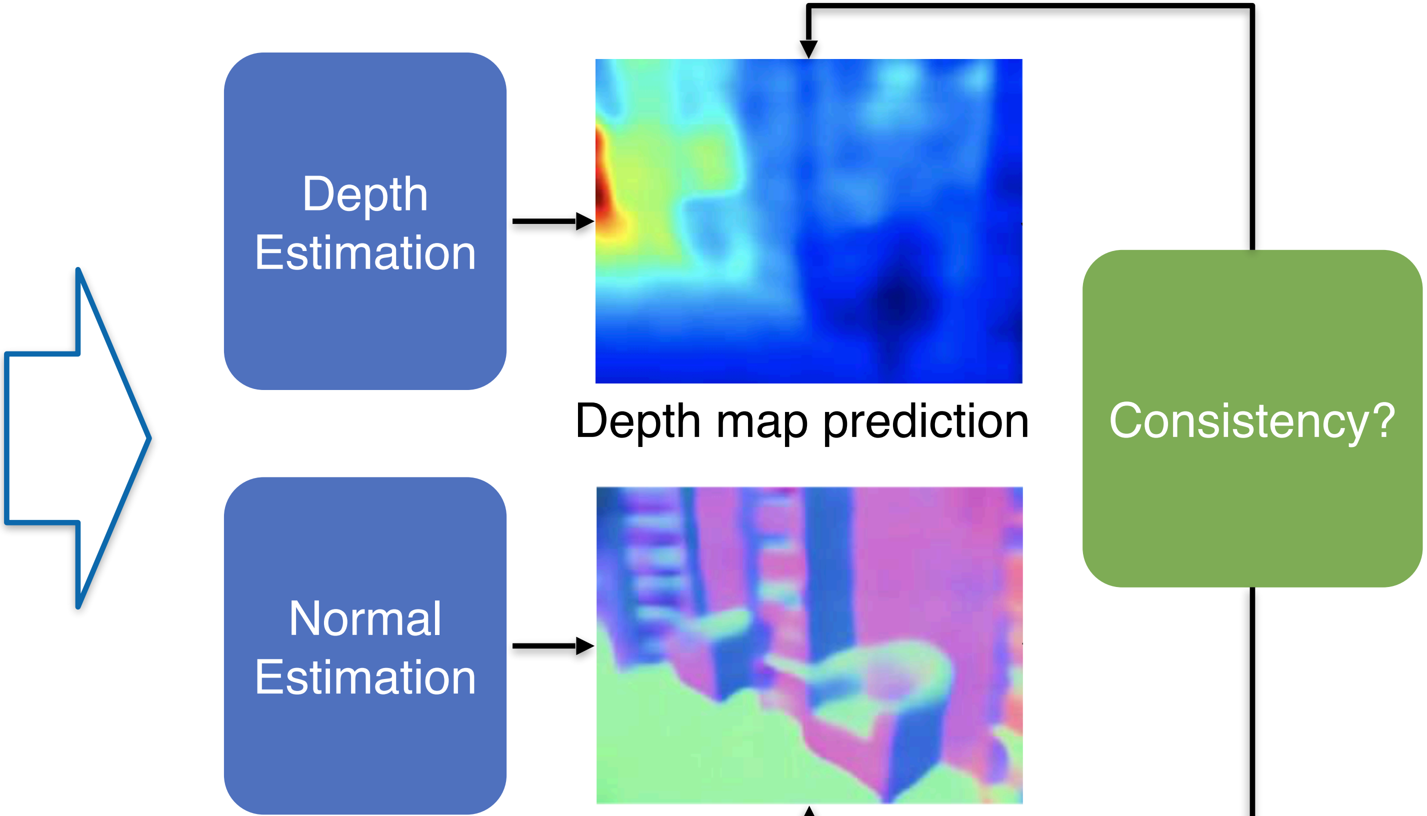
GT Normal



Predicted Normal

Depth Normal Consistency

- Estimate normal along with depth map.
- Regularize depth by normals.



Depth Estimation

Normal Estimation

Depth map prediction

Consistency?

Normal map prediction

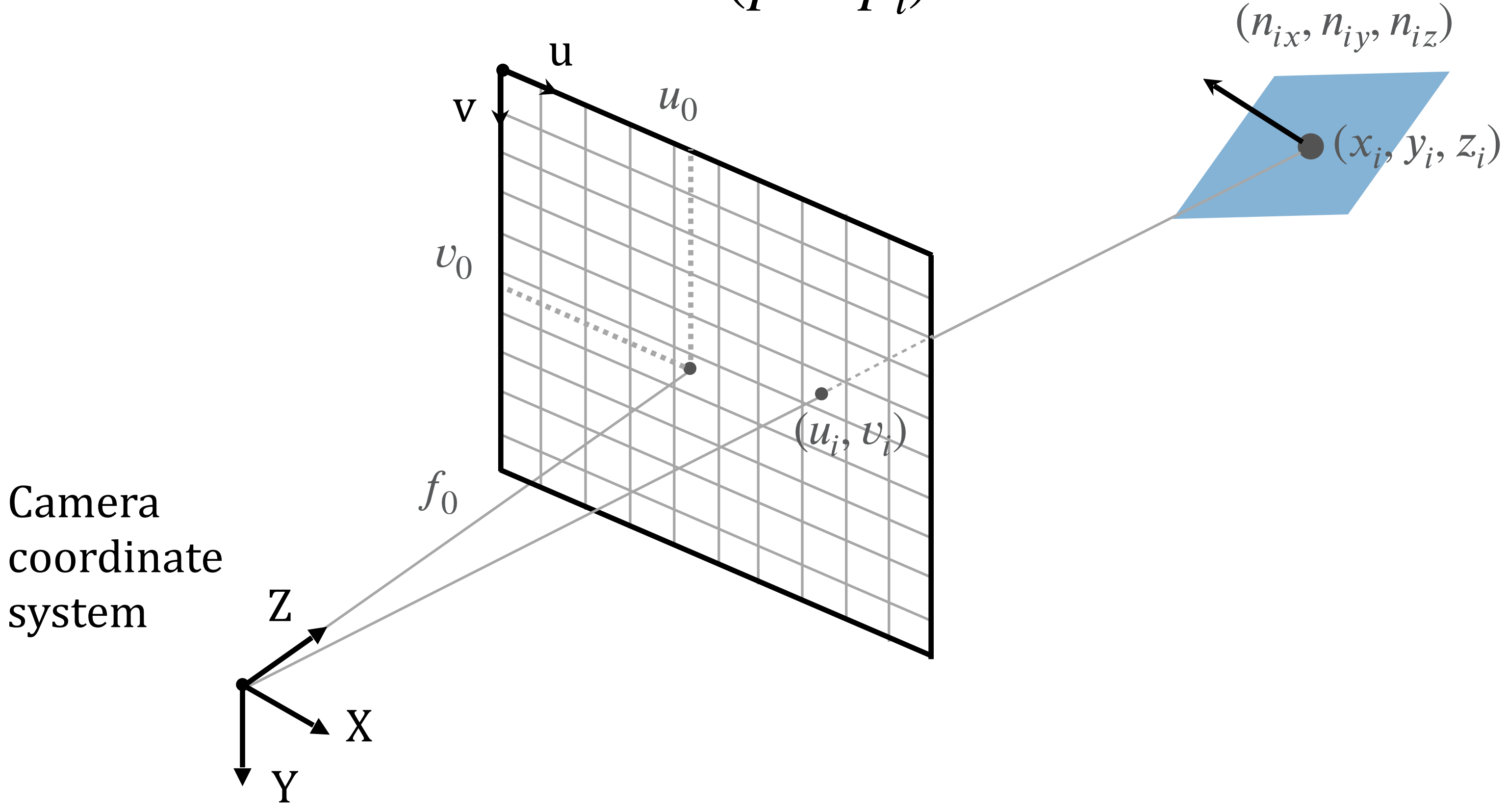
- Practice 1: Normal estimation as an auxiliary loss
    - Already quite effective

- Practice 2: Use normal estimation to correct depth

Refine Depth from Normal

- **Key assumption**: pixels within a local neighborhood lie on the same tangent plane.

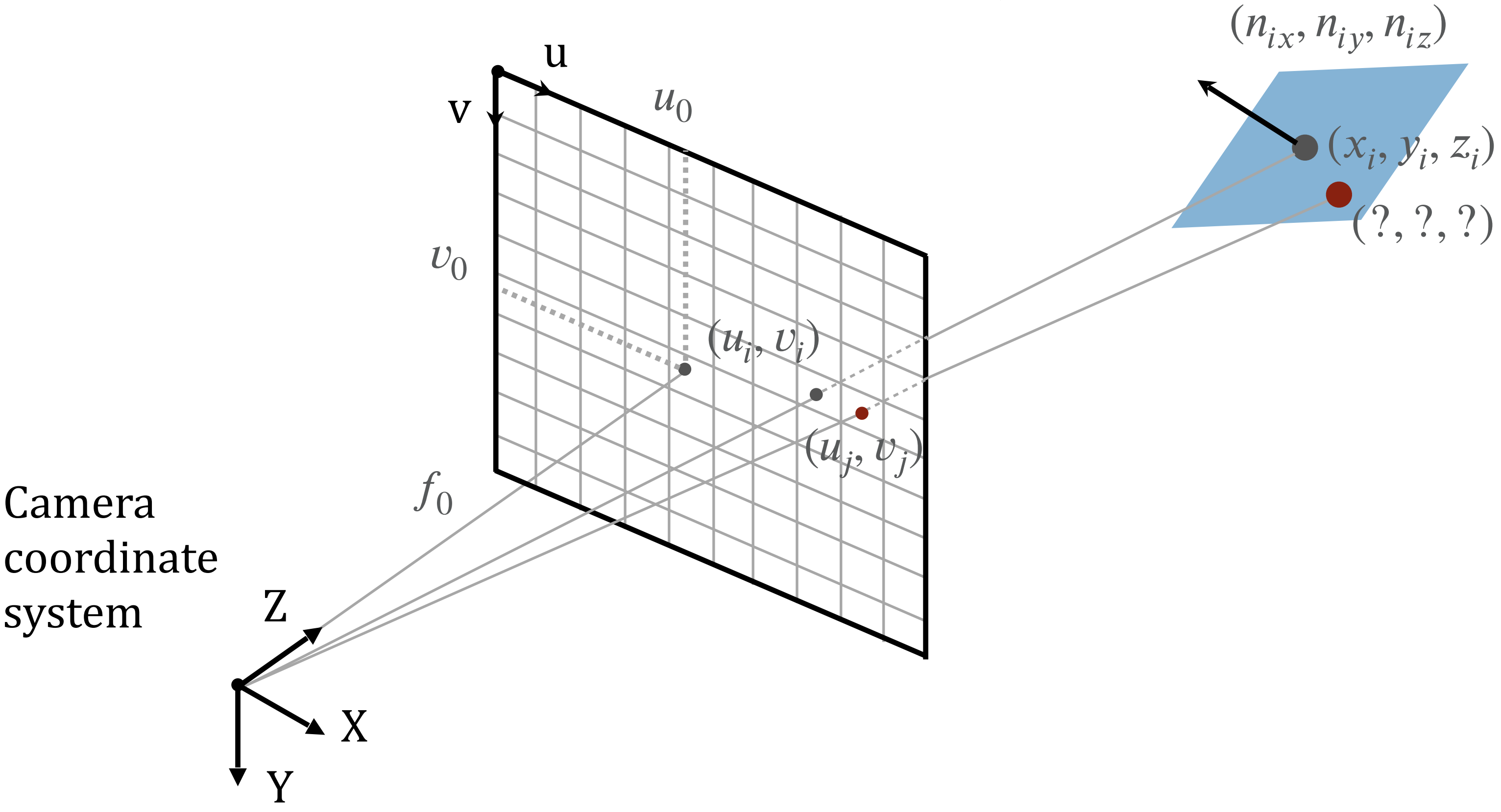
$$\vec{n}^T(p - p_i) = 0$$



Refine Depth from Normal

- Derive neighbor pixel depth from current pixel normal.

$$z'_{i\to j} = \frac{n_{ix}x_j + n_{iy}y_j + n_{iz}z_j}{(u_i - u_0)n_{ix}/f_0 + (v_i - v_0)n_{iy}/f_0 + n_{iz}}$$



Summary

- Deep volumetric stereo can lead to more robust matching and more complete reconstruction
- But volume-based methods are NOT computationally efficient, since the 3D target scene is sparse
- Adaptive sampling can improve computation efficiency and reconstruction quality
- Normal prediction is easier than depth, and can help improve depth accuracy and smoothness

Appearance Information Capturing

- Photometric-consistency gives geometry

- **Can we also get the appearance information?**

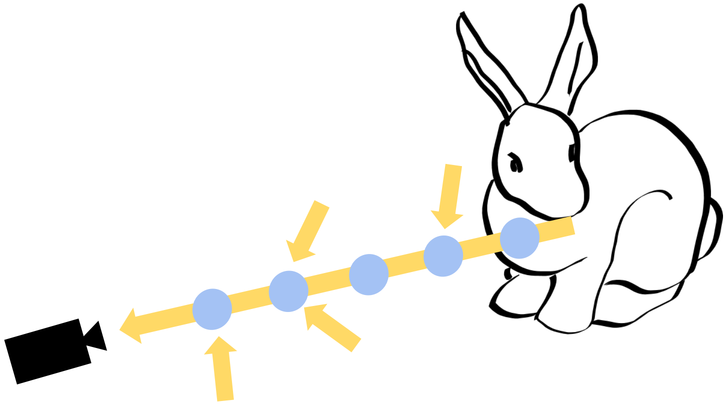
General Idea

- The appearance of the surface will be observed at views along the camera ray

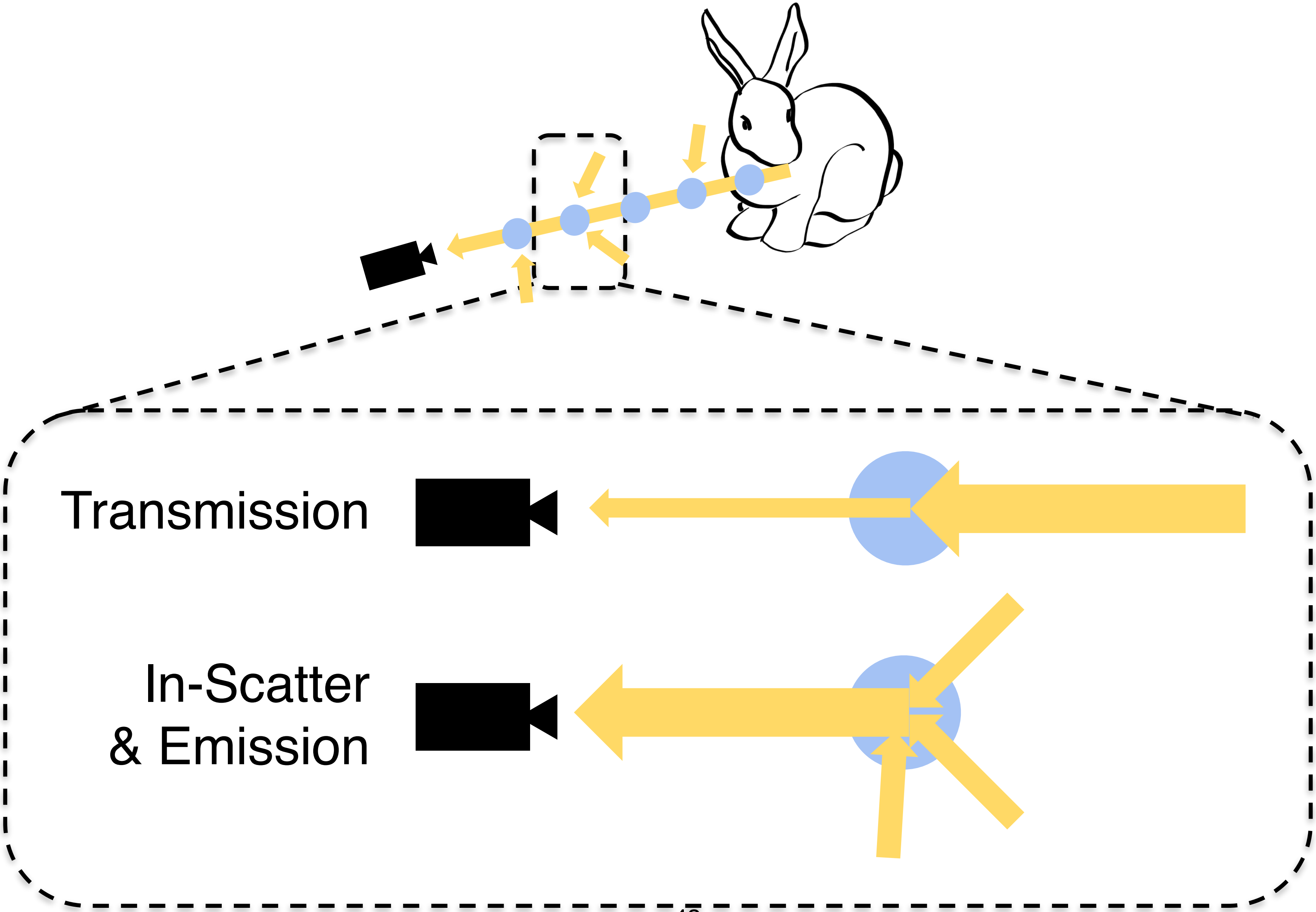
- If we have a **light transport model** from the surface along the ray to the pixels, we will know the pixel color

- By comparing the pixel color from the light transport model and from the ground-truth image, we can build a loss

Ray Marching

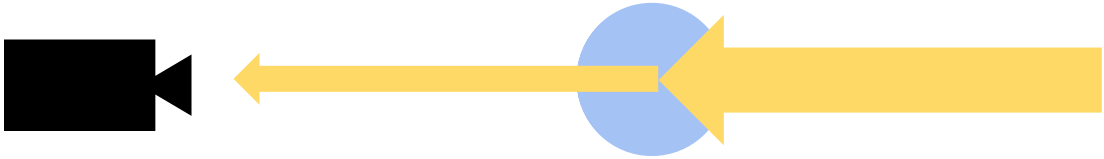


Ray Marching



Ray Marching

Transmission

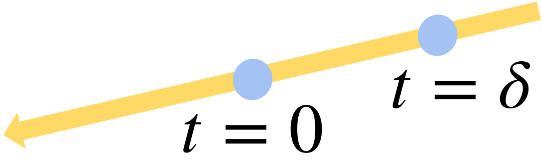


Attenuation coefficient σ (Transparency)

In-Scatter
& Emission

Emission Radiance c

Emission Radiance Passing a Ray Segment

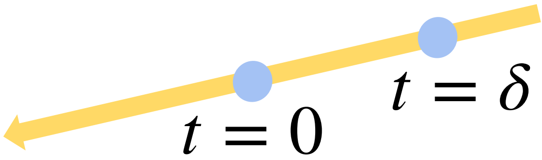


Beer-Lambert's Law: $\alpha(t) = 1 - \exp(-\sigma t)$

opacity

attenuation coefficient

Emission Radiance Passing a Ray Segment



Beer-Lambert's Law: $\alpha(t) = 1 - \exp(-\sigma t)$

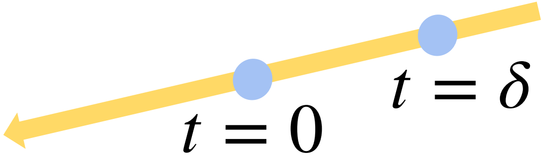
opacity

attenuation coefficient

emission radiance

Light emitted along a segment $= \int_0^\delta (1 - \alpha(t))c(t)\mathrm{d}t$

Emission Radiance Passing a Ray Segment

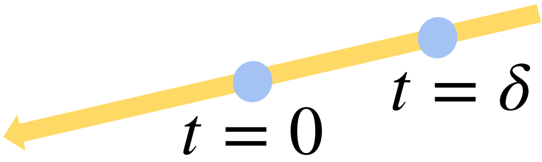


Beer-Lambert's Law: $\alpha(t) = 1 - \exp(-\sigma t)$

Light emitted along a segment $= \displaystyle\int_0^\delta (1 - \alpha(t))c(t)\mathrm{d}t \overset{c(t)=c}{\approx} c\int_0^\delta \exp(-\sigma t)\mathrm{d}t$

$$= \frac{c}{\sigma}(1 - \exp(-\delta\sigma))$$

Emission Radiance Passing a Ray Segment

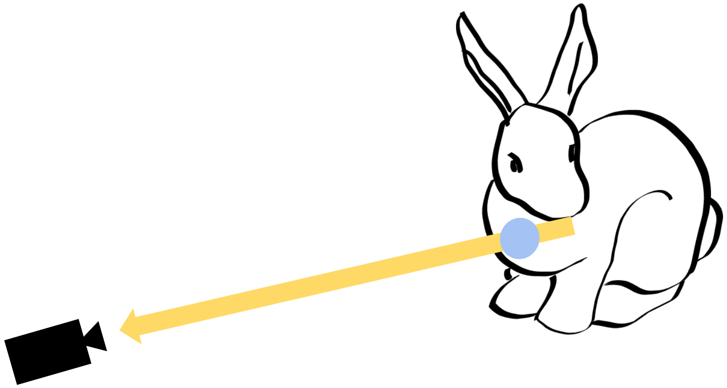


Beer-Lambert's Law: $\alpha(t) = 1 - \exp(-\sigma t)$

Light emitted along a segment $= \int_0^\delta (1 - \alpha(t))c(t)\mathrm{d}t \overset{c(t)=c}{\approx} c\int_0^\delta \exp(-\sigma t)\mathrm{d}t$

$$= \frac{c}{\sigma}(1 - \exp(-\delta\sigma)) = \alpha(\delta)\left(\frac{c}{\sigma}\right)$$

Discretized Radiance Integration (Ray Marching)



A single point

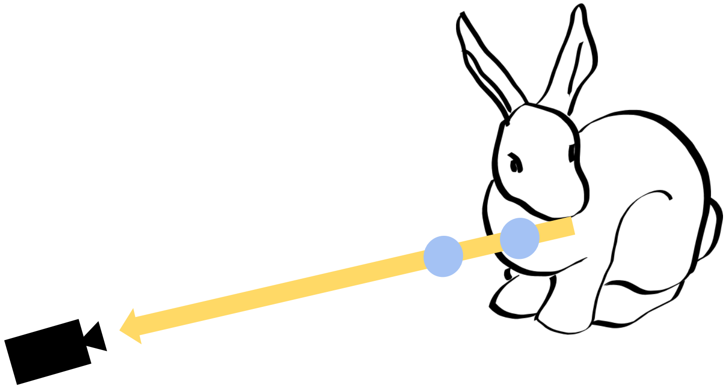
$$I_1 = \alpha_1 \left(\frac{c_1}{\sigma_1} \right)$$

I_1: light intensity after point 1

c_1: predicted emission radiance at point 1

α_1: opacity of point 1

Discretized Radiance Integration (Ray Marching)



2 points

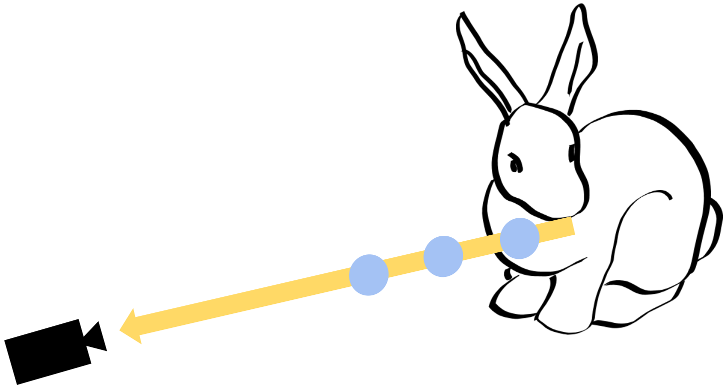
Point 1 acts like the previous case

$$I_1 = \alpha_1 \left(\frac{c_1}{\sigma_1} \right)$$

Point 2 additionally transmits I_2

$$I_2 = \alpha_2 \left(\frac{c_2}{\sigma_2} \right) + (1 - \alpha_2) I_1$$

Discretized Radiance Integration (Ray Marching)



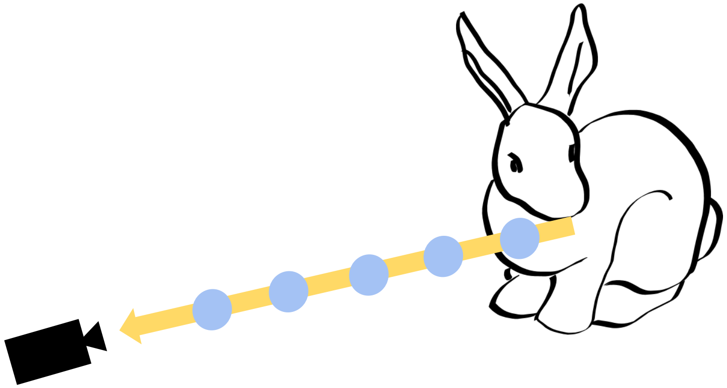
3 points

$$I_1 = \alpha_1 \left(\frac{c_1}{\sigma_1} \right)$$

$$I_2 = \alpha_2 \left(\frac{c_2}{\sigma_2} \right) + (1 - \alpha_2) I_1$$

$$I_3 = \alpha_3 \left(\frac{c_3}{\sigma_3} \right) + (1 - \alpha_3) I_2 + (1 - \alpha_3)(1 - \alpha_2) I_1$$

Discretized Radiance Integration (Ray Marching)



In general

n: the number of points

$$T_i = \prod_{j=i+1}^{n} (1 - \alpha_j) = \exp(-\sum_{j=i+1}^{n} \sigma_j \delta_j)$$

$$I = \sum_i T_i \alpha_i \left(\frac{c_i}{\sigma_i} \right)$$ = final radiance of the ray

Discretized Radiance Integration (Ray Marching)

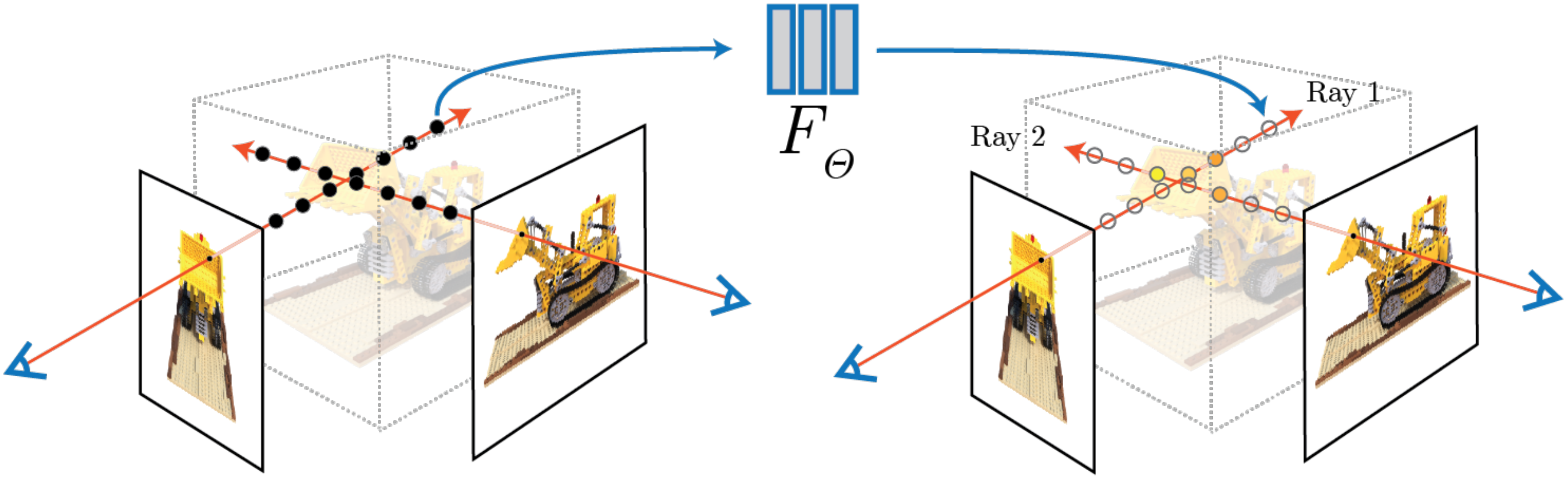
$$(\sigma_i, \frac{c_i}{\sigma_i}) = F_\Theta(x, y, z, \theta, \phi)$$

Note: It is quite common that σ_i and c_i are both close to zero, so we predict c_i/σ_i directly.

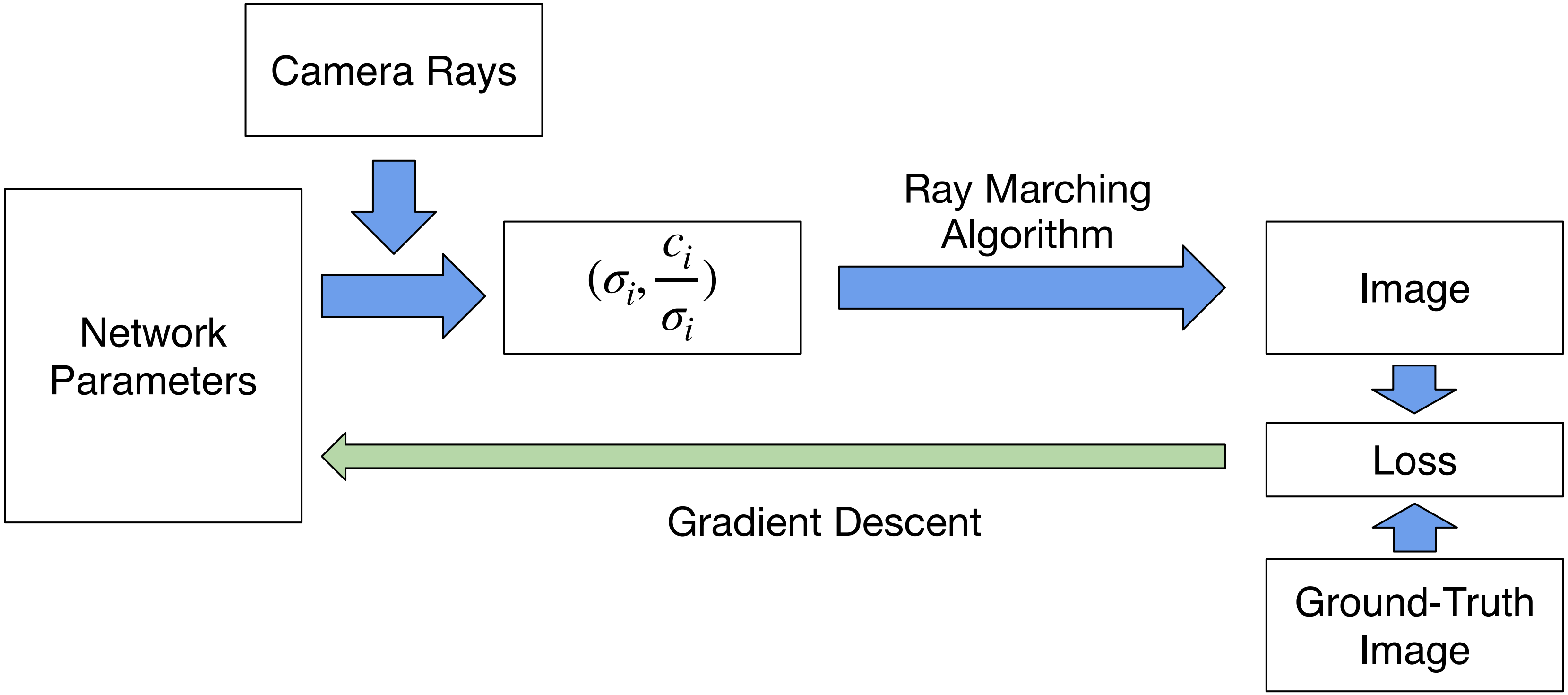
Pixel Loss

$$I = \sum_i T_i \alpha_i \left(\frac{c_i}{\sigma_i} \right)$$

Comparing I with ground-truth pixel value, we get a loss (e.g., L1, L2)



Train Pipeline (as in NeRF)



Camera Rays

Network Parameters

$(\sigma_i, \frac{c_i}{\sigma_i})$

Ray Marching Algorithm

Image

Loss

Ground-Truth Image

Gradient Descent

- Optimize on a single scene
  - store the scene in weights of the network
- Require ground-truth camera parameters

Result

- Novel view synthesis following light transport model (F_{Θ} optimized from ~100 views)

Summary

- We have described a volumetric rendering-based loss function for 3D estimation
- The approach takes an implicit neural function representation, allowing for infinite resolution
- This is an example of ***physics-based deep learning*** pipeline
- Knowing the domain knowledge is helpful for building network architecture!