Lecture 20: Multi-View Stereo (MVS)

Instructor: Roni Sengupta

ULA: Andrea Dunn Beltran, William Li,
Liujie Zheng

Course Website:
Scan Me!

Geometry: How do we represent shape of an object?

2.5D representation:

1) Depth & Normal map

Explicit representation:

2) Mesh
3) Voxels
4) Point Cloud

Implicit representation:

5) Surface Representation (SDF)

Geometry: How do we represent shape of an object?

2.5D representation:

1) Depth & Normal map

Explicit representation:

2) Mesh

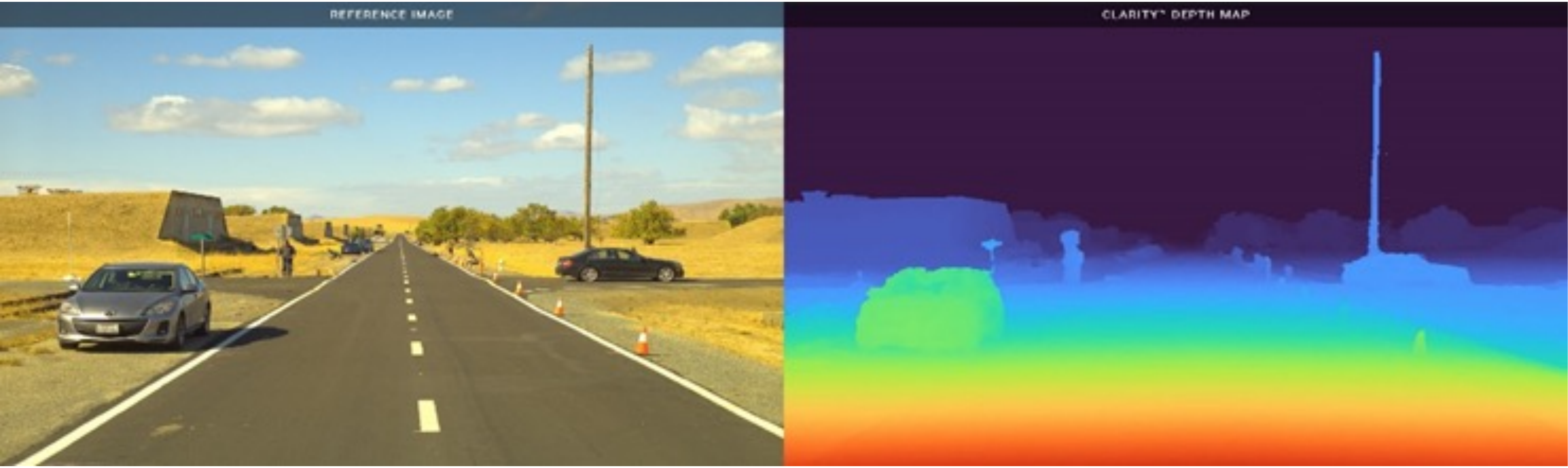
3) Voxels

4) Point Cloud

Implicit representation:

5) Surface Representation (SDF)

Depth Map

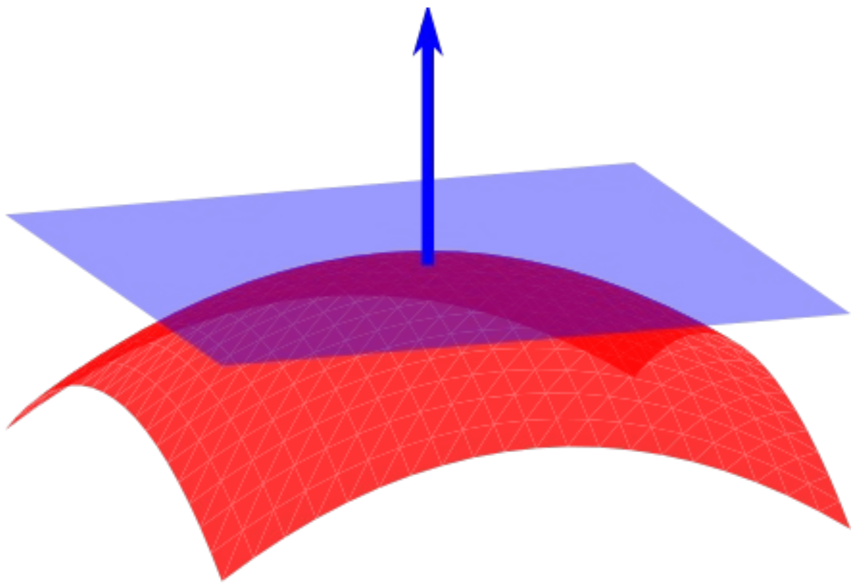


Depth Map D(u,v): Distance of any pixel (u,v) from the camera (usually image plane)

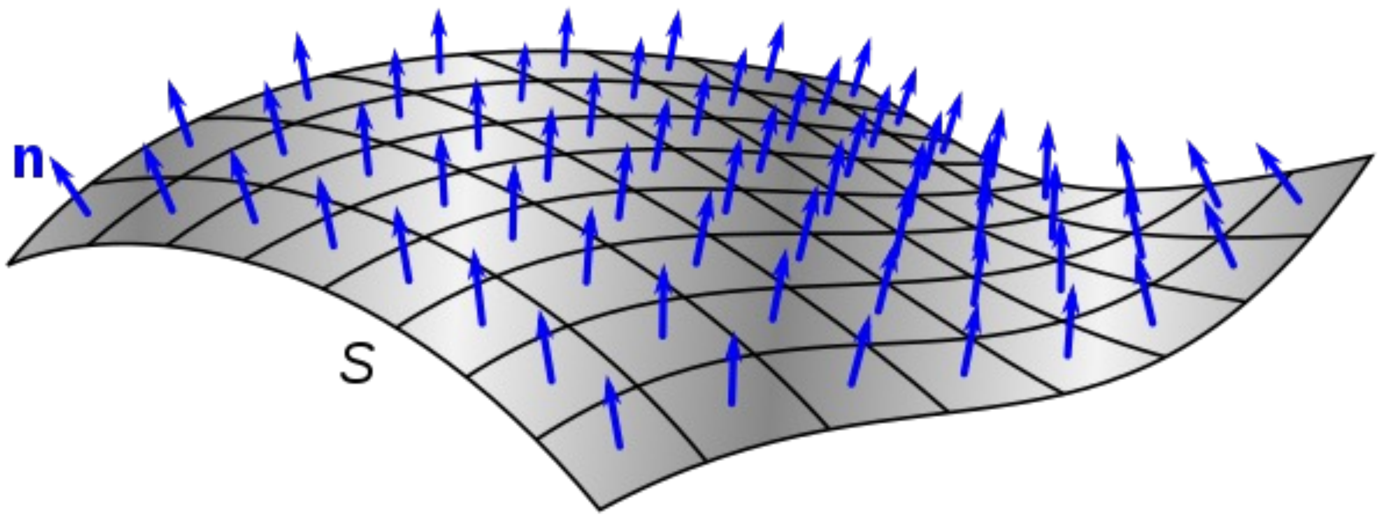
Red-> nearer; blue-> further

For an image HxWx3, a depth map is HxWx1 (scalar value for every pixel)

Surface Normal



Surface Normal (in blue) of a point P is a vector perpendicular to the tanget plane at P.



Surface normal (in blue) of a surface

Surface normal indicate orientation of the surface.

Normal Map

Normal Map N(u,v): [Nx,Ny,Nz] is a unit vector indicating the orientation of the surface.

Pink-> towards left; blue-> towards right

For an image HxWx3, a normal map is HxWx3.

Credits: ibug imperial college london

Relationship between Depth & Normal Map

$$\tilde{N} = [\frac{\partial D}{\partial x}, \frac{\partial D}{\partial y}, -1]$$

$$N = \frac{\tilde{N}}{\|\tilde{N}\|_2}$$

Normalizing to unit vector.

- Differentiation of depth map leads to normal map
- Integration of normal map leads to depth map

Further reading: Normal Integration: A Survey

Geometry: How do we represent shape of an object?

2.5D representation:

1) Depth & Normal map

Explicit representation:

2) Mesh

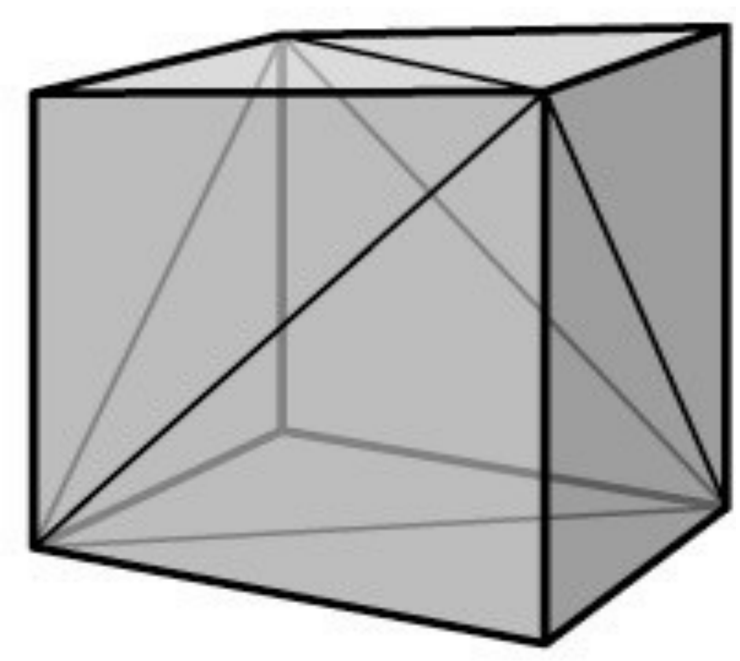
3) Voxels

4) Point Cloud

Implicit representation:

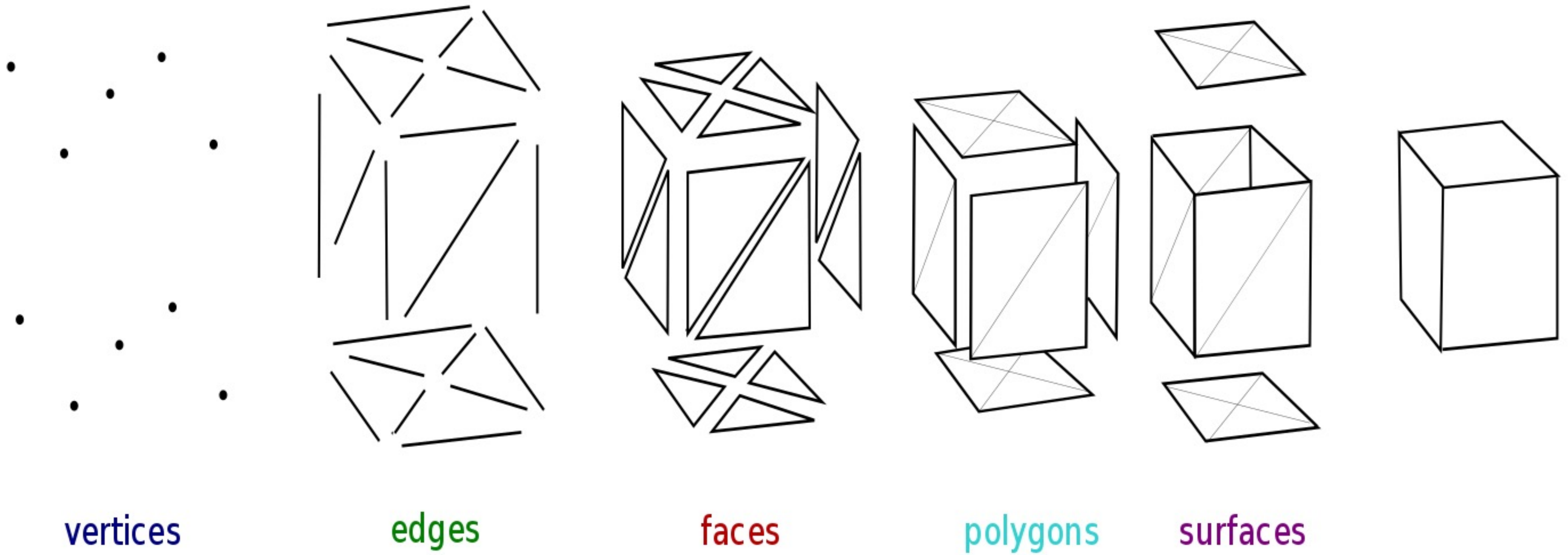
5) Surface Representation (SDF)

A Small Triangle Mesh



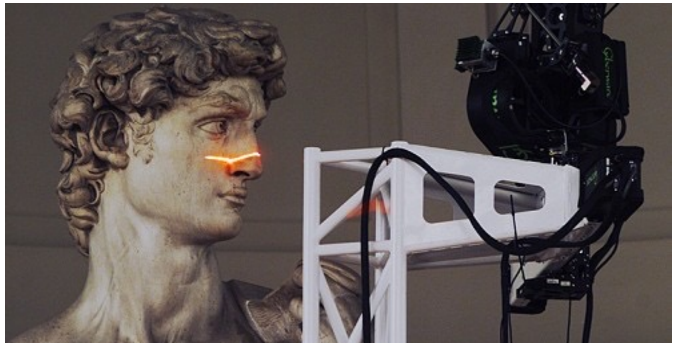
8 vertices, 12 triangles

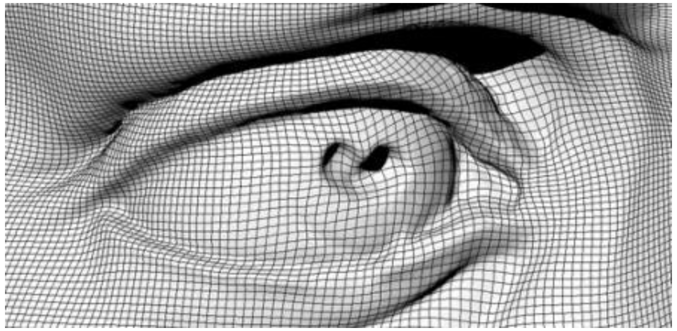
Mesh



A Large Triangle Mesh

David
Digital Michelangelo Project
28,184,526 vertices
56,230,343 triangles





Marc Levoy at Stanford (https://accademia.stanford.edu/mich/)

Geometry: How do we represent shape of an object?

2.5D representation:

1) Depth & Normal map

Explicit representation:

2) Mesh

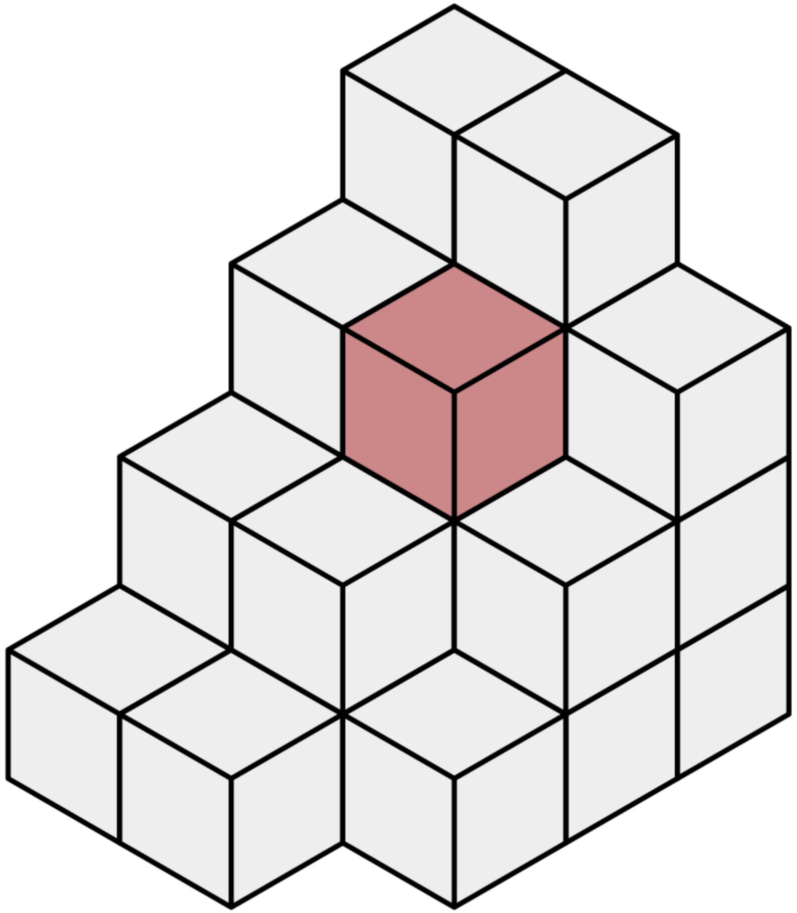
3) Voxels

4) Point Cloud

Implicit representation:

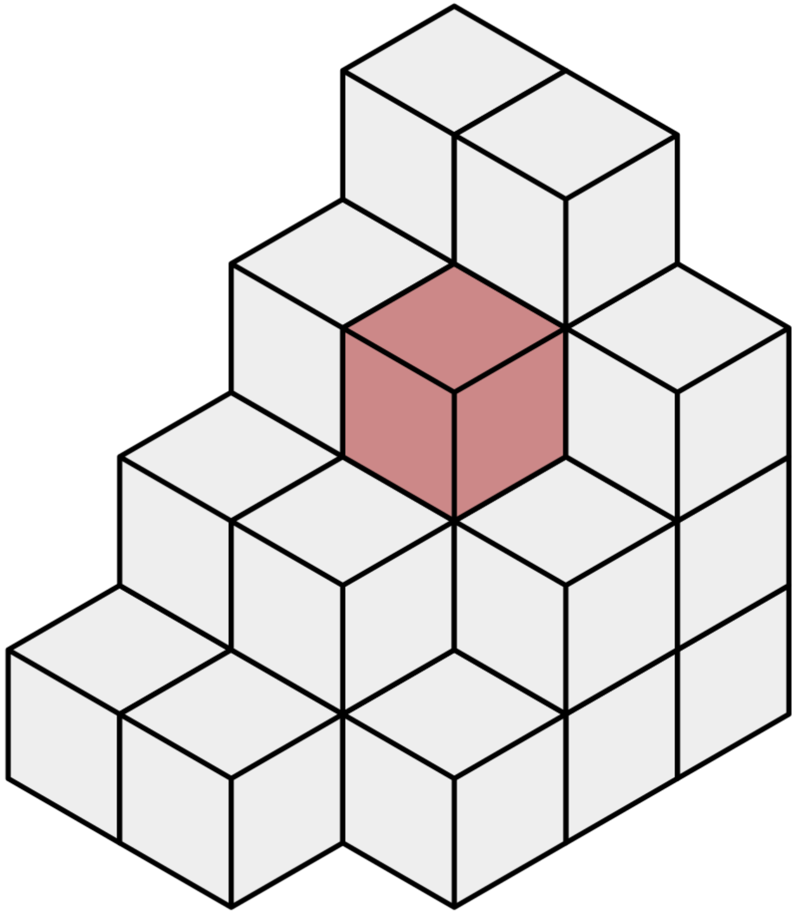
5) Surface Representation (SDF) – implicit

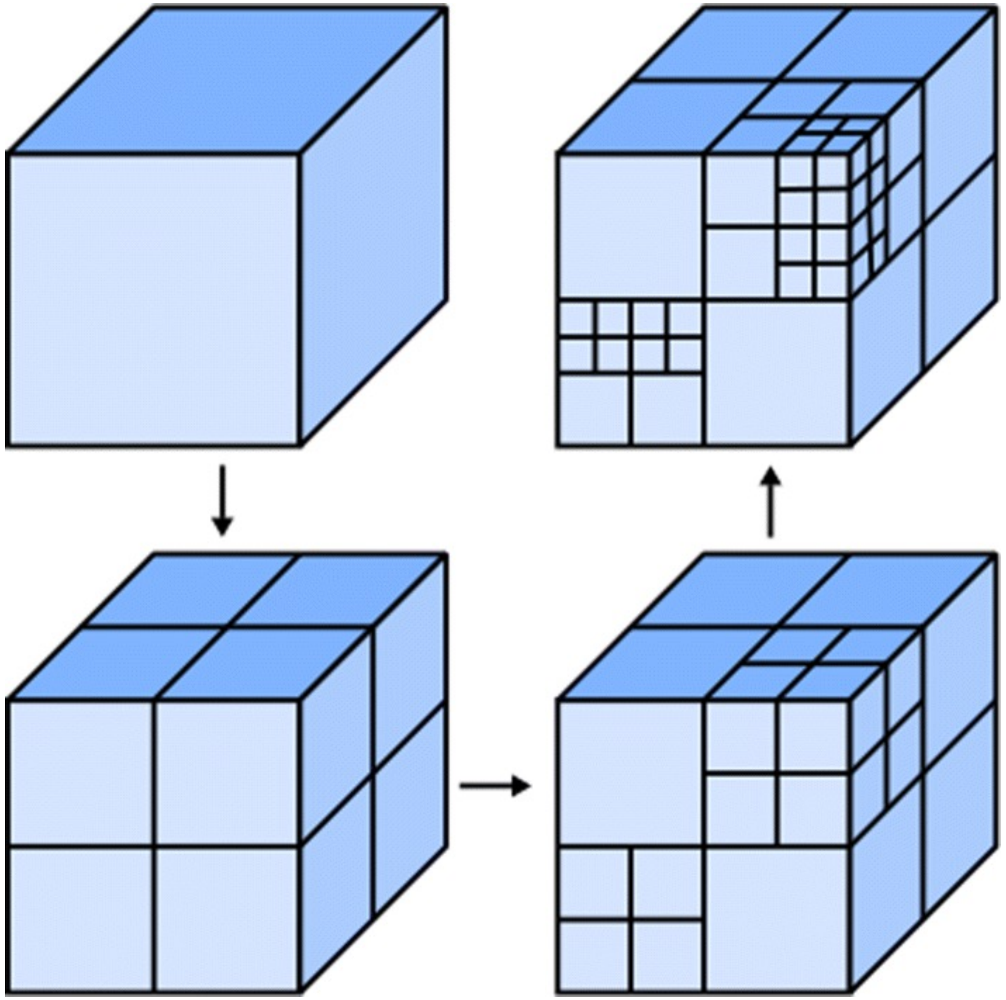
Voxel Representation



It's like playing with Lego!

Voxel Representation





Voxel with octree

Geometry: How do we represent shape of an object?

2.5D representation:

1) Depth & Normal map

Explicit representation:

2) Mesh

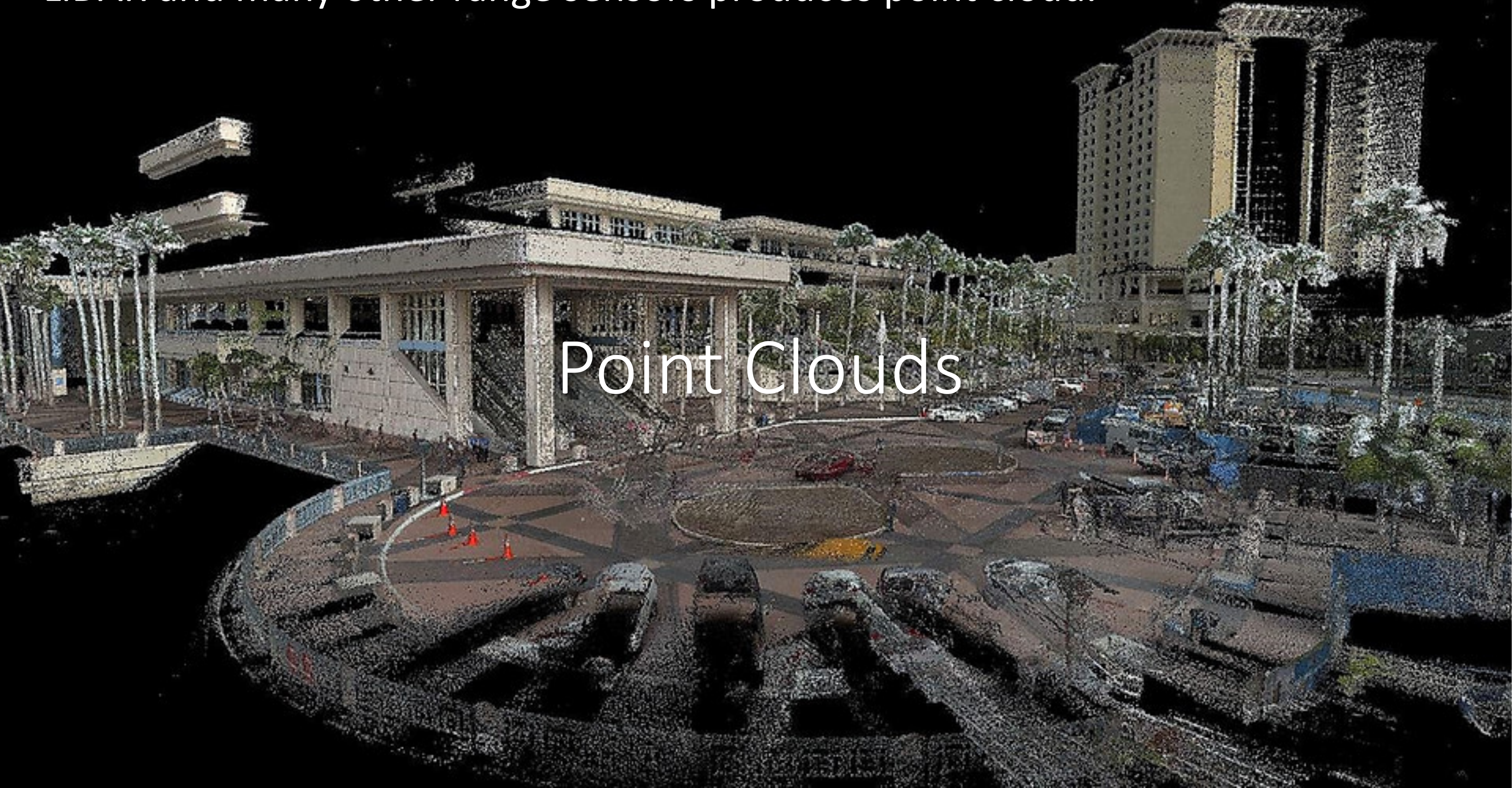
3) Voxels

4) Point Cloud

Implicit representation:

5) Surface Representation (SDF) – implicit

LiDAR and many other range sensors produces point cloud.



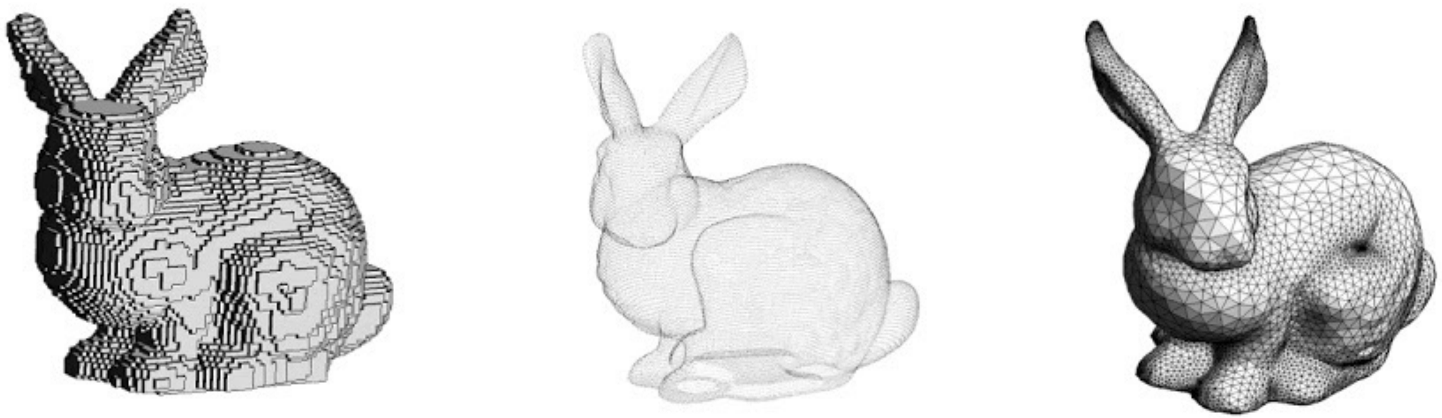
Point Clouds

Sparse model of central Rome using 21K photos produced by COLMAP's SfM pipeline.

Dense models of several landmarks produced by COLMAP's MVS pipeline.

Started at UNC!! – Jan Michael Frahm's group

3D Representations (Explicit)



	Voxel	Point cloud	Polygon mesh
Memory efficiency	Poor	Not good	Good
Textures	Not good	No	Yes
For neural networks	Easy	Not easy	Not easy

We adopt **polygon mesh** for its high potential

Images are from
http://cse.iitkgp.ac.in/~pb/research/3dpoly/3dpoly.html
http://waldyrious.net/learning-holography/pb-cgh-formulas.xhtm
http://www.cs.mun.ca/~omeruvia/philosophy/images/BunnyWire.gif

Geometry: How do we represent shape of an object?

2.5D representation:

1) Depth & Normal map

Explicit representation:

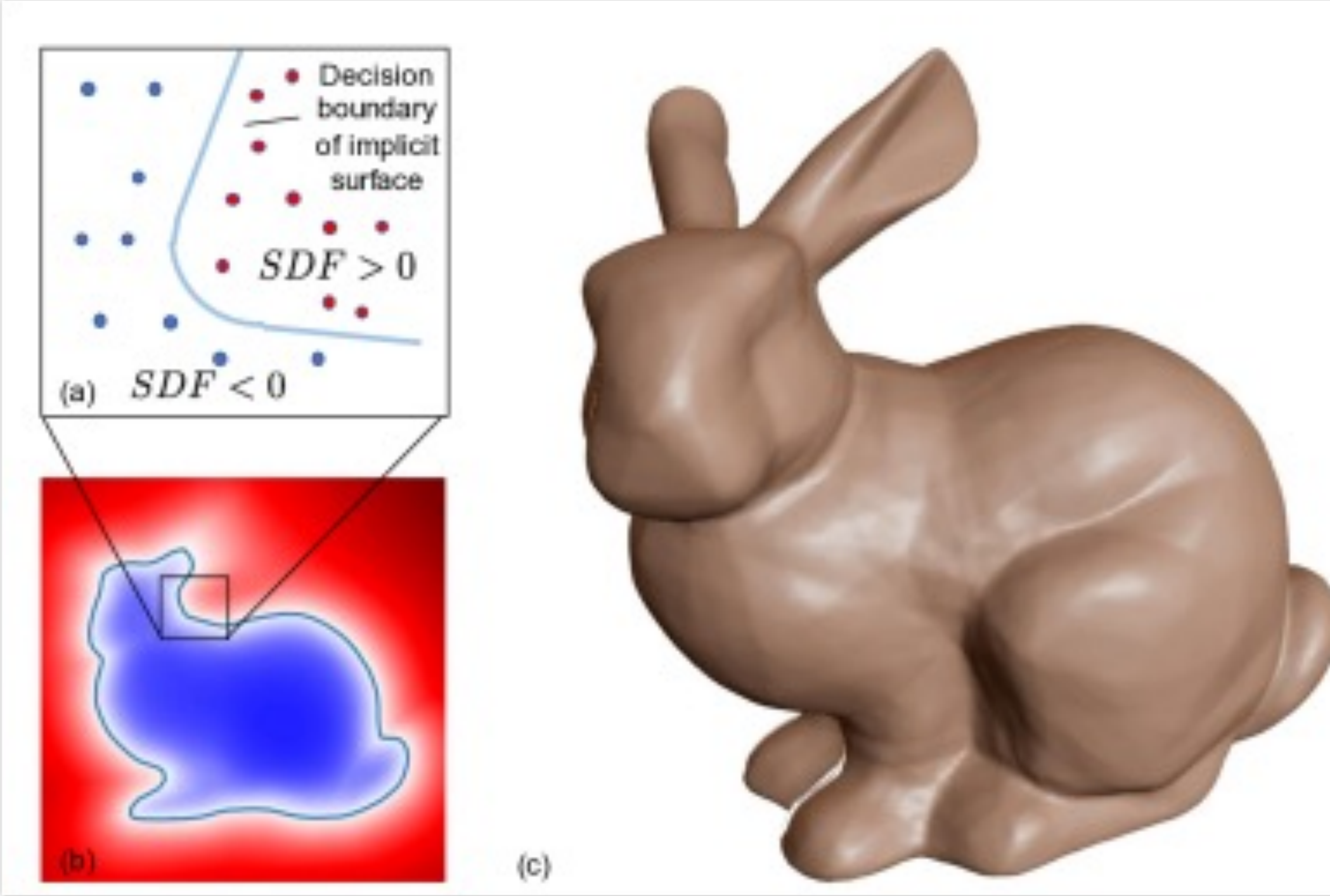
2) Mesh
3) Voxels
4) Point Cloud

Implicit representation:

5) Surface Representation (SDF)

Surface Representation: Signed Distance Function (SDF) - implicit representation via level set

SDF(X) = 0, when X is on the surface.
SDF(X) > 0, when X is outside the surface
SDF(X) < 0, when X is inside the surface

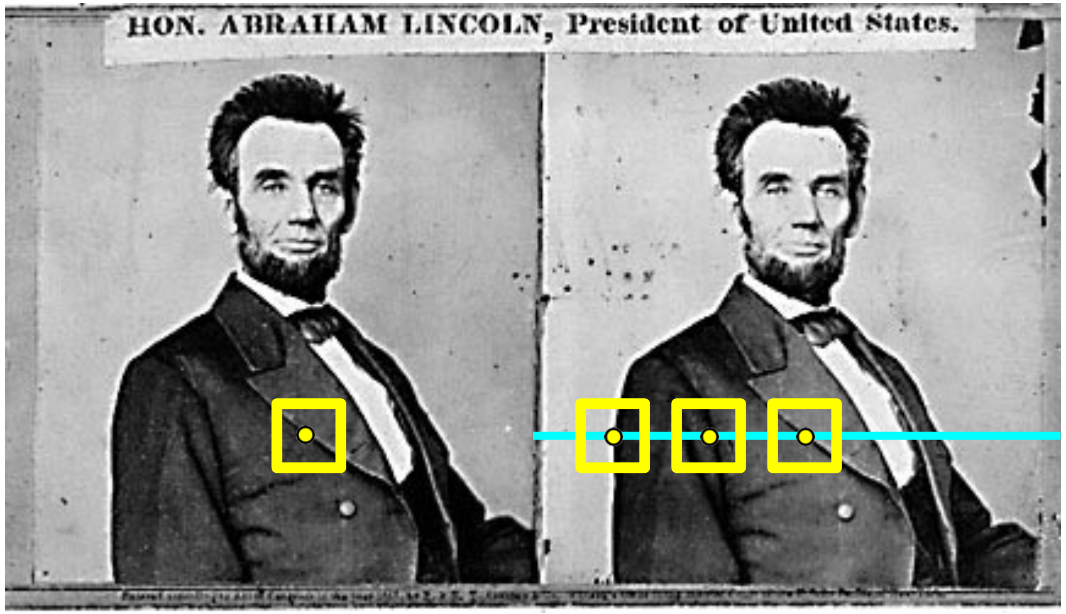


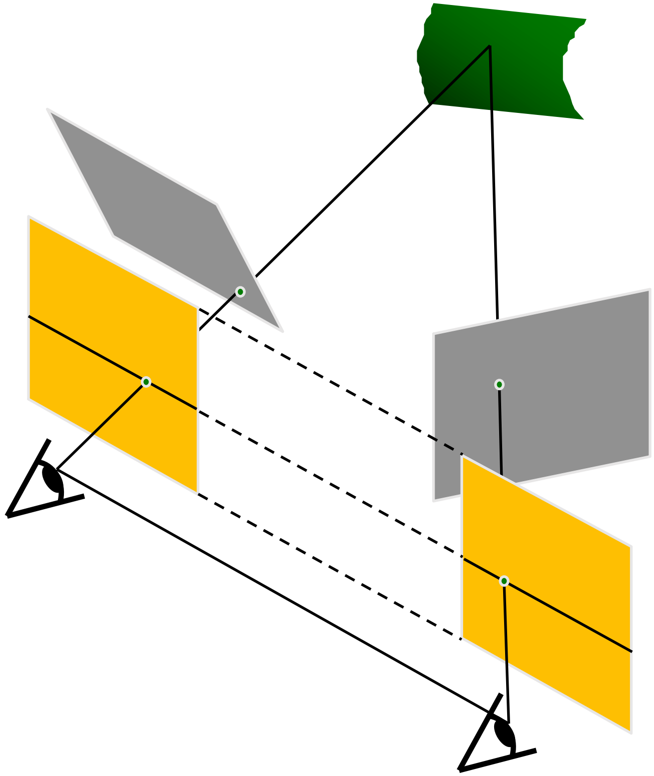
Note: SDF is an implicit representation! Suitable for neural networks but hard to import inside existing graphics software.

Deep SDF: Use a neural network (co-ordinate based MLP) to represent the SDF function.

Neural RGB-D Scene Reconstruction, Azinovic et. al.

Recap
(Stereo)





1. Rectify images
   (make epipolar lines horizontal)
2. For each pixel
   a. Find epipolar line
   b. Scan line for best match
   c. Compute depth from disparity

$$Z = \frac{bf}{d}$$

How to do Stereo Matching

- Greedy: for every pixel in left scanline -> choose best match in right scanline.

- What properties get violated in greedy approach?
  - Uniqueness: match should be unique
  - Smoothness: disparity should vary slowly
  - Occlusion: handle pixels when occluded in left or right image
  - Ordering constraint: Ordered set of points should have same match.

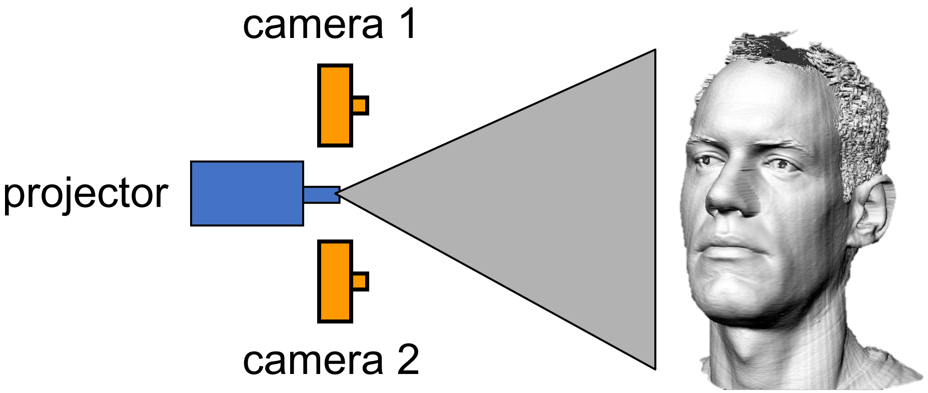
- Non-greedy: choose best match for all pixels in the left scanline. How?
  - Dynamic Programming
  - Graph Cut approach
  - Deep Learning

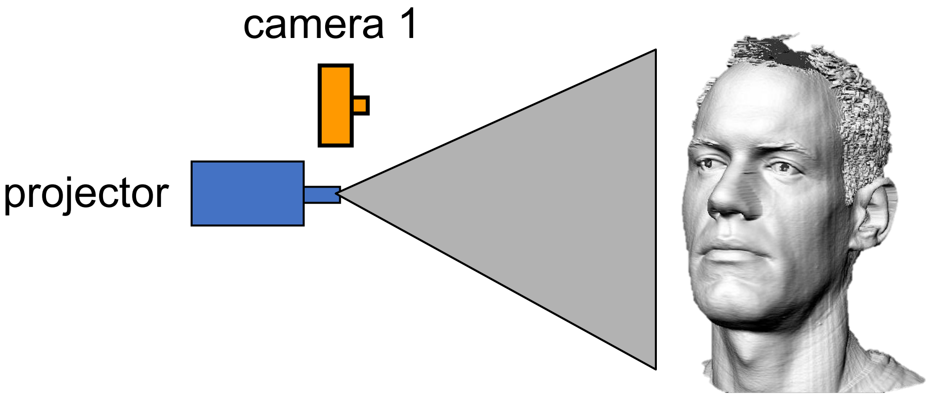
Why Study Stereo?

- Passive Stereo:
  - Self-driving car
  - Any autonomous robots
  - 3D movies
- Active Stereo: Make correspondence easier by projecting patterns (structured lights)
  - Apple TrueDepth
  - Kinect
  - Laser scanning for 3D reconstruction

Active stereo with structured light

Li Zhang's one-shot stereo





- Project "structured" light patterns onto the object
  - simplifies the correspondence problem
  - basis for active depth sensors, such as Kinect and iPhone X (using IR)

Today’s class

- Motivation
- Simple Approach to MVS
- Advanced Approach to MVS
  - Plane Sweep Stereo
  - Space Curving Stereo
- Converting depth to mesh
- MVS in deep learning era (more later)

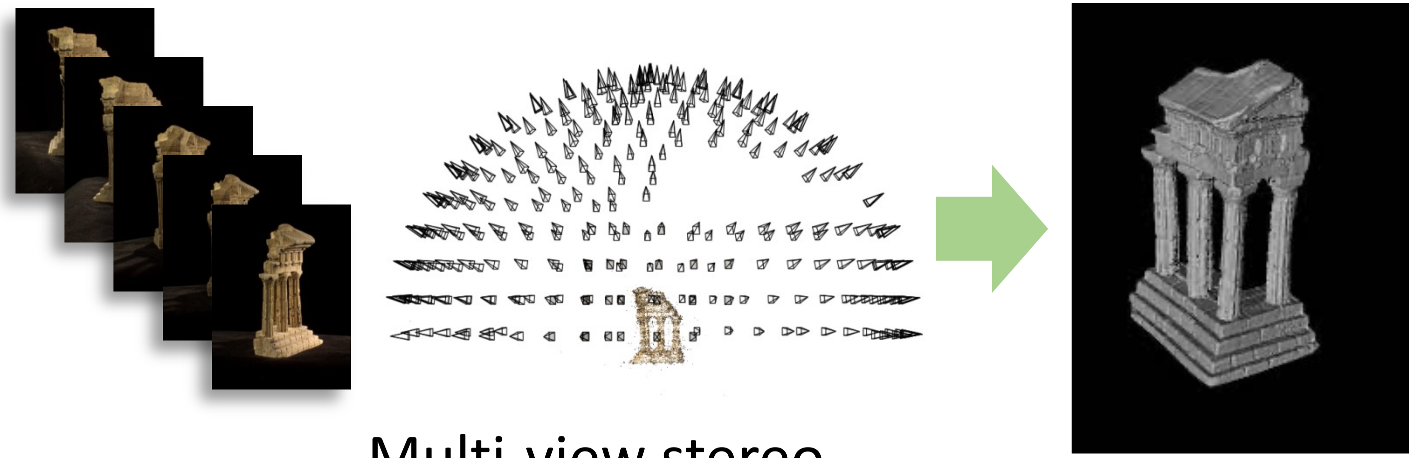
Today’s class

- Motivation
- Simple Approach to MVS
- Advanced Approach to MVS
    - Plane Sweep Stereo
    - Space Curving Stereo
- Converting depth to mesh
- MVS in deep learning era (more later)

Multi-view Stereo

Problem formulation: given several images of the same object or scene, compute a representation of its 3D shape

Binocular Stereo



Multi-view stereo

Multi-view Stereo

Point Grey's Bumblebee XB3

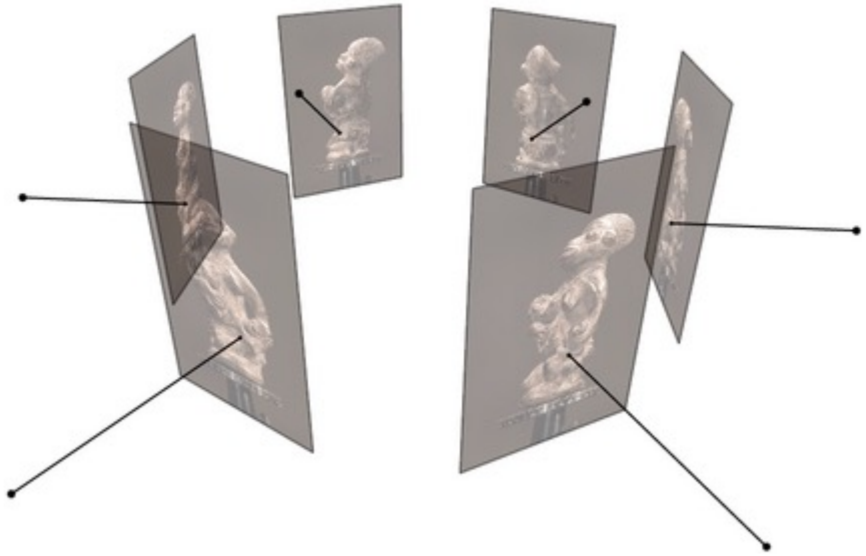
Point Grey's ProFusion 25

CMU's Panoptic Studio

Multi-view Stereo

Input: calibrated images from several viewpoints (known intrinsics and extrinsics / projection matrices)

Output: 3D object model



Figures by Carlos Hernandez


Data SIO, NOAA, U.S. Navy, NGA, GEBCO
Google earth

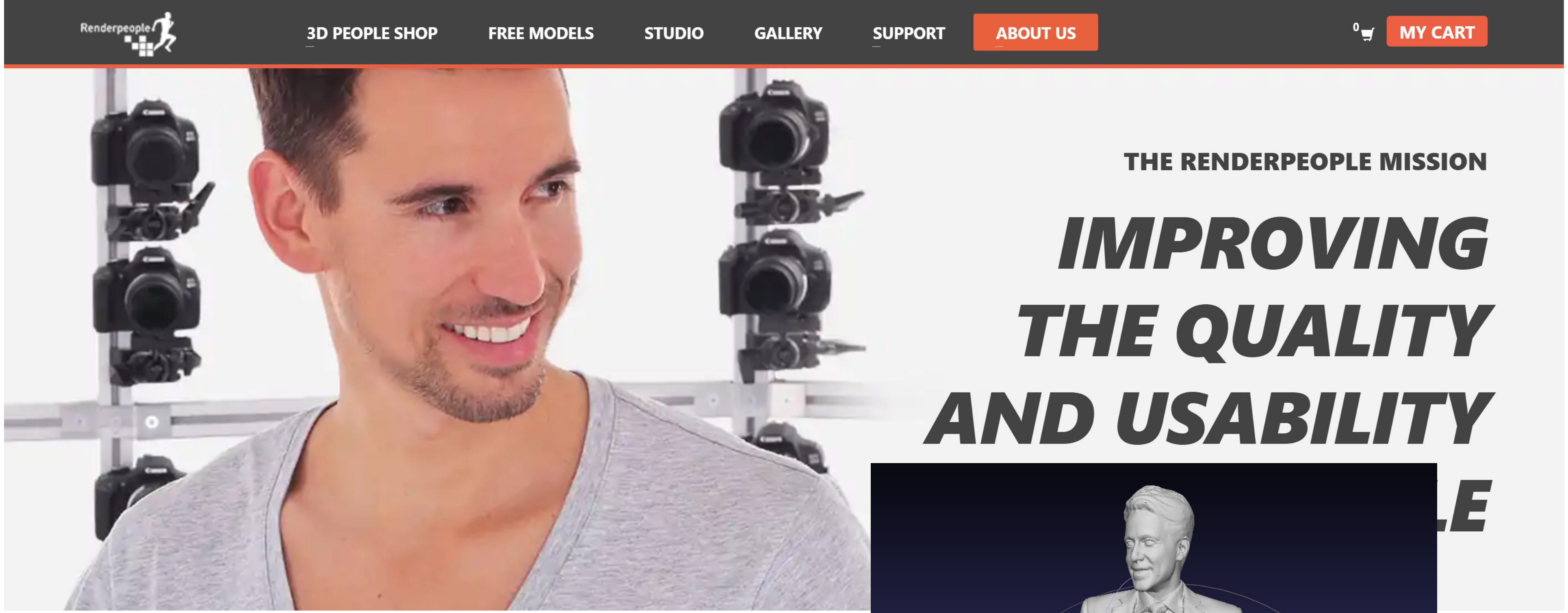
Whistle in the Form of Female Figure *600 AD - 900 AD*

Details Los Angeles County Museum of Art

Los Angeles County Museum of Art | Sculpture | Mexico

Share | Compare | Saved | Discover

Google



https://renderpeople.com/about-us/

Virtual Reality Video

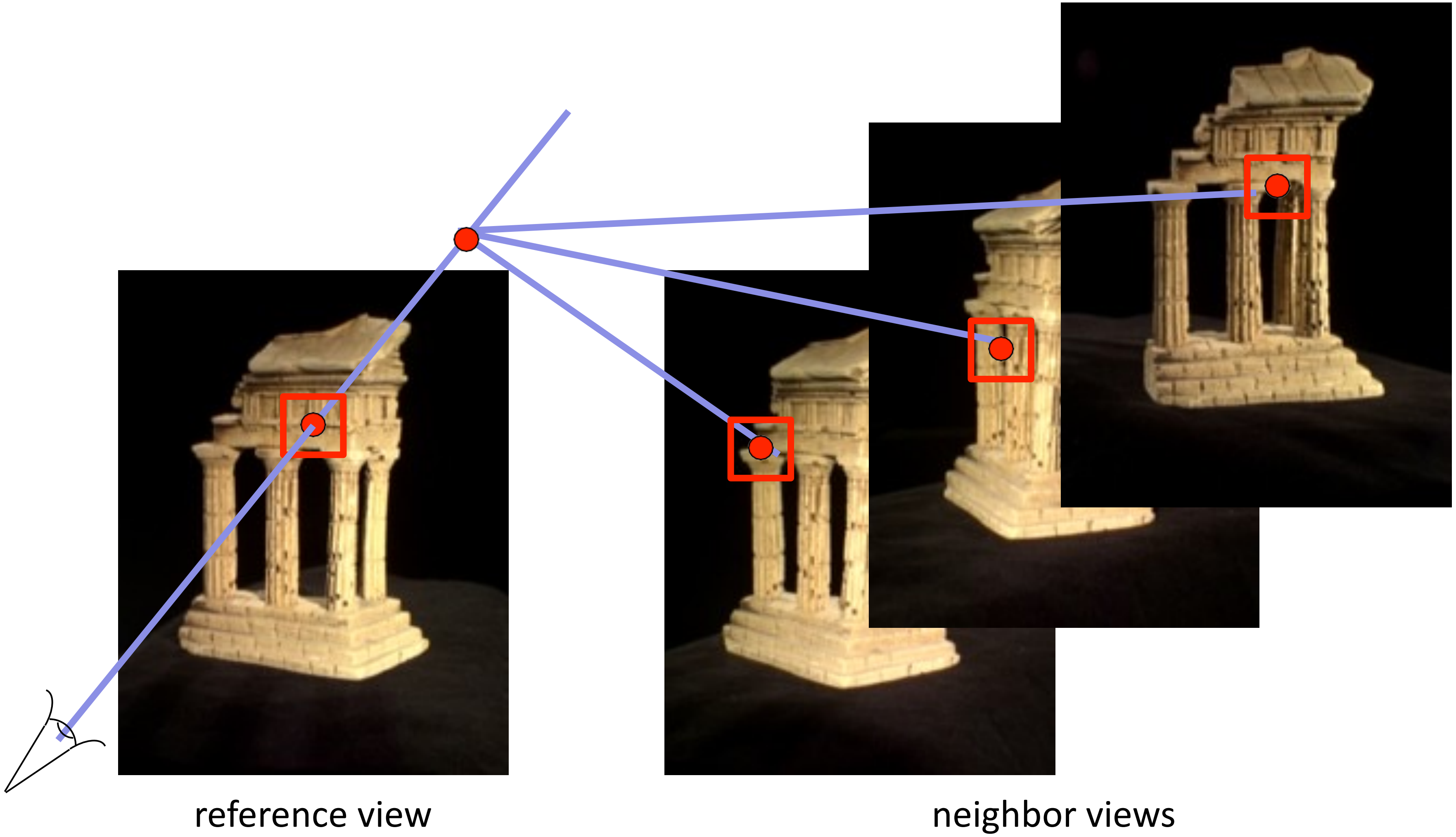
Anderson, et al. *Jump: Virtual Reality Video*. SIGGRAPH Asia 2016.

Broxton, et al. *Immersive Light Field Video with a Layered Mesh Representation*. SIGGRAPH 2020.

Today's class

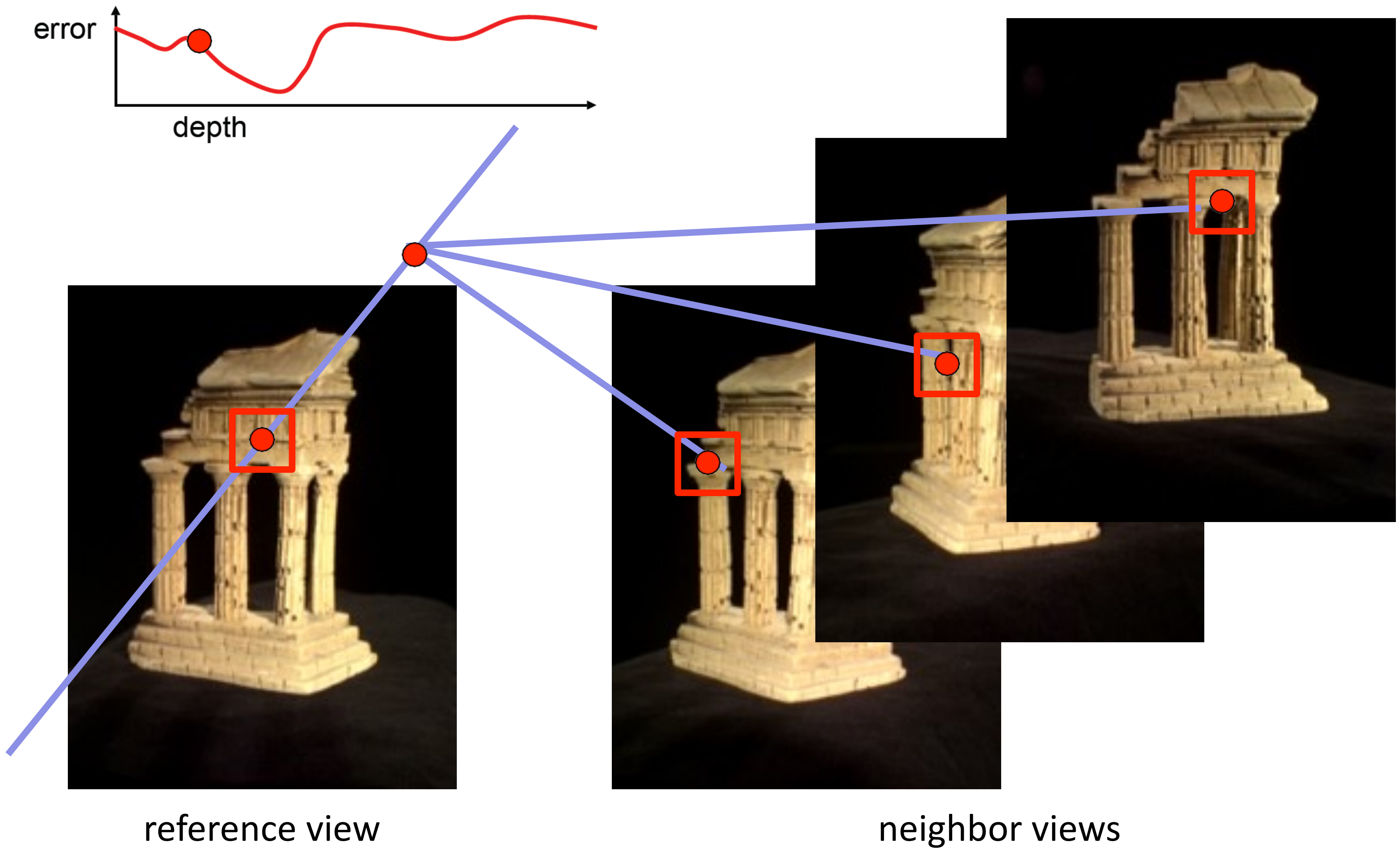
- Motivation
- Simple Approach to MVS
- Shape representations
- Advanced Approach to MVS
  - Plane Sweep Stereo
  - Space Curving Stereo
- Converting depth to mesh
- MVS in deep learning era (more later)

Multi-view stereo: Basic idea



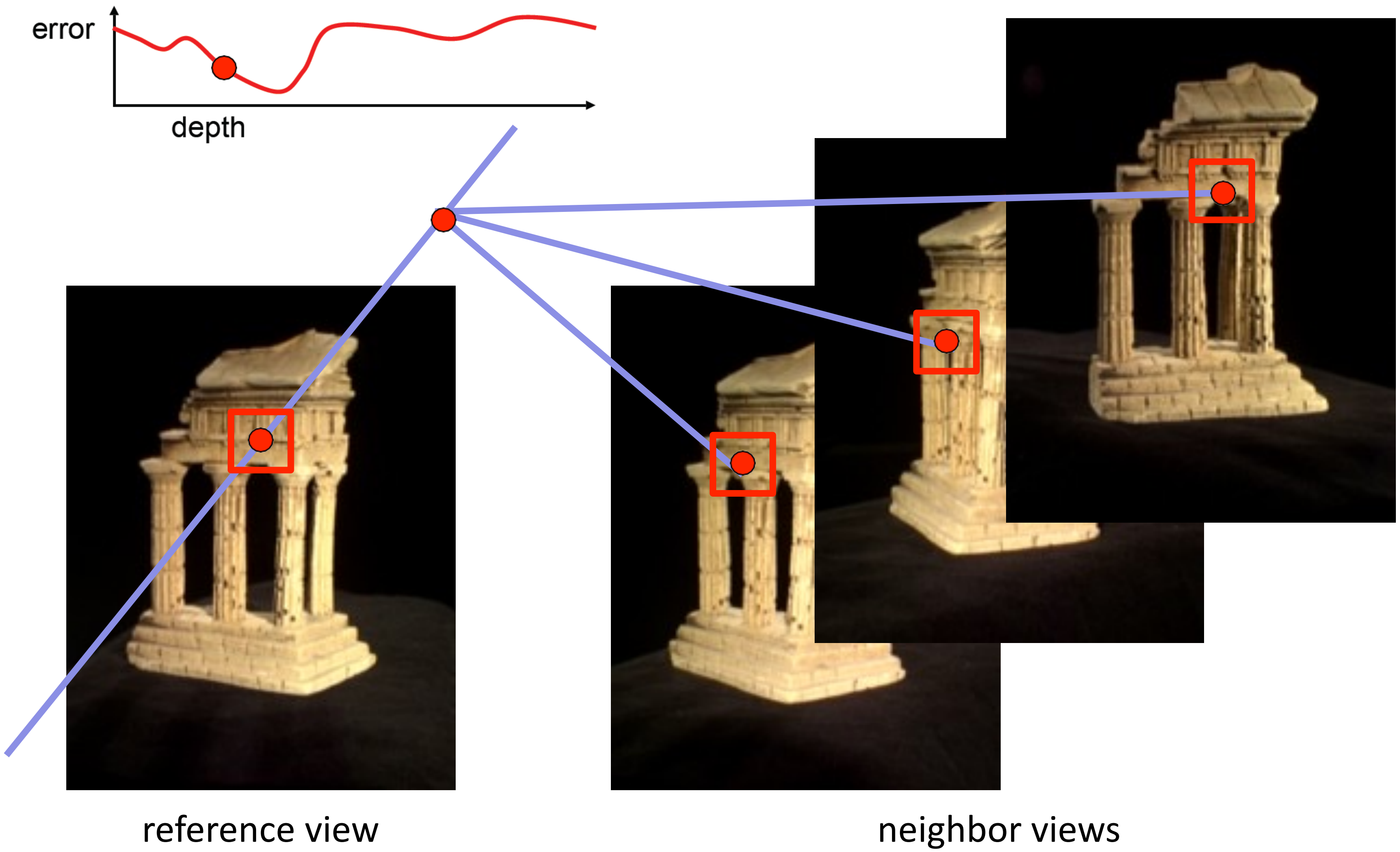
Source: Y. Furukawa

Multi-view stereo: Basic idea



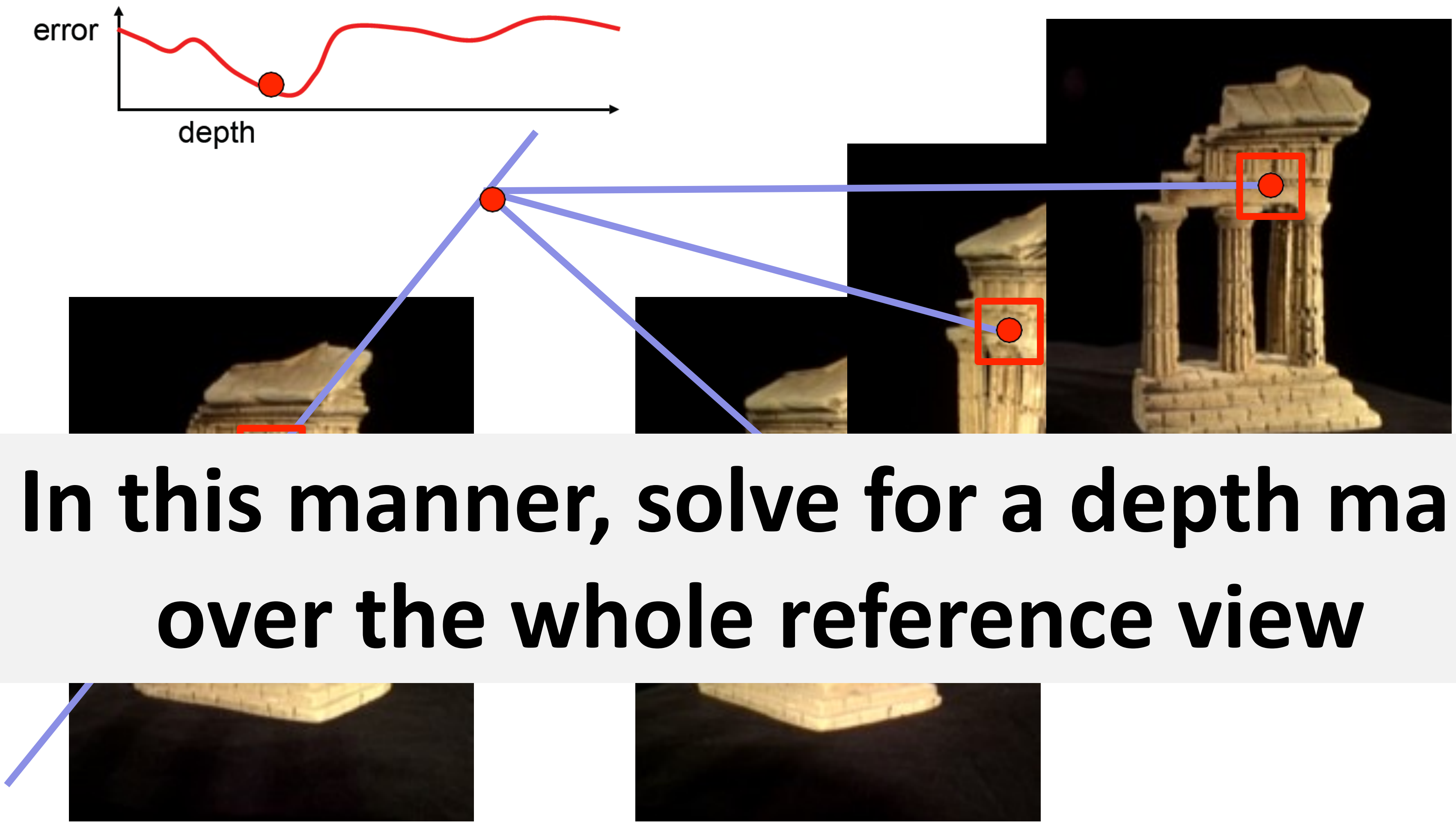
Source: Y. Furukawa

Multi-view stereo: Basic idea



Source: Y. Furukawa

Multi-view stereo: Basic idea



In this manner, solve for a depth map over the whole reference view

reference view

neighbor views

Source: Y. Furukawa

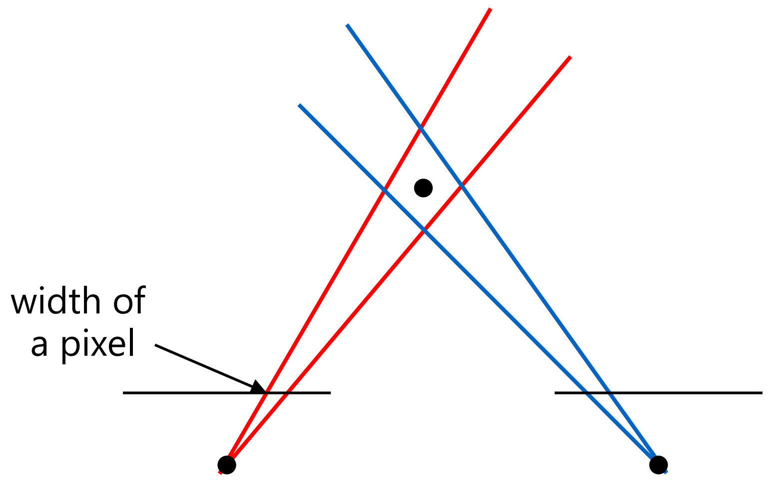
Multi-view stereo: advantages

- Can match windows using more than 1 neighbor, giving a **stronger match signal**

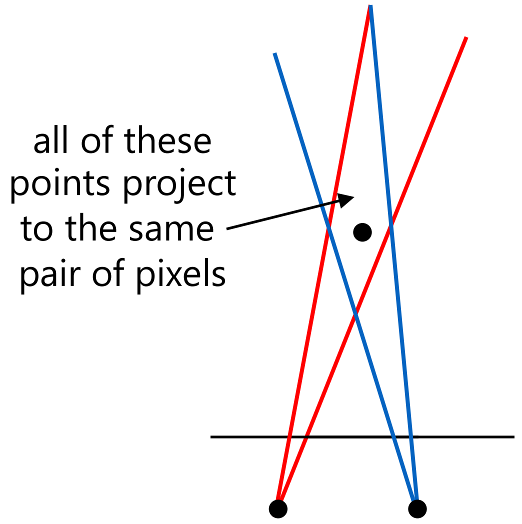
- If you have lots of potential neighbors, can **choose the best subset** of neighbors to match per reference image

- Can reconstruct a depth map for each reference frame, and the merge into a **complete 3D model**

Choosing the stereo baseline



Large Baseline

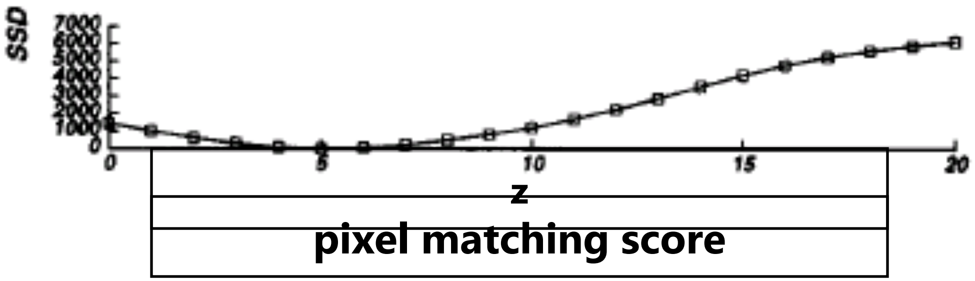


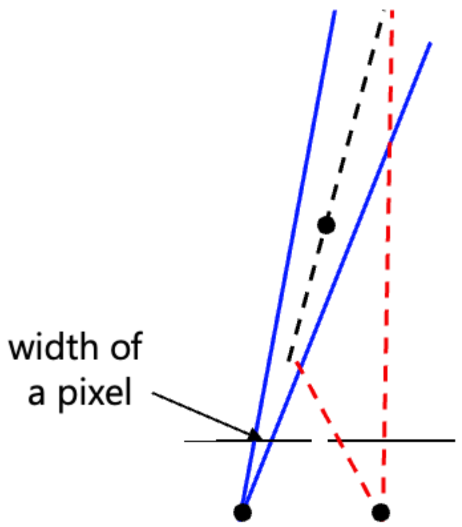
Small Baseline

What's the optimal baseline?

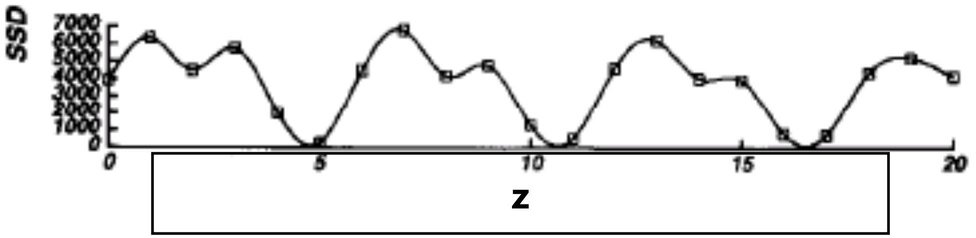
- Too small: large depth error
- Too large: difficult search problem

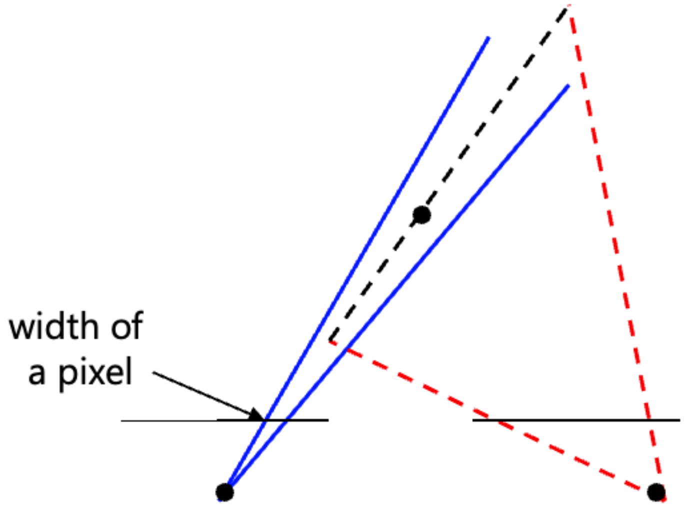
Multiple-baseline stereo





- For short baselines, estimated depth will be less precise due to narrow triangulation



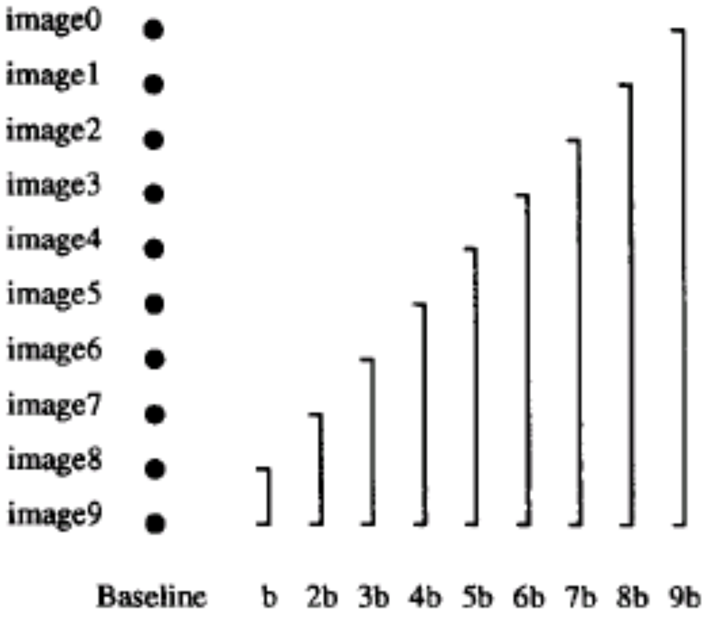


- For larger baselines, must search larger area in second image

M. Okutomi and T.Kanade, "A Multiple-Baseline Stereo System," IEEE Trans. on Pattern Analysis and Machine Intelligence, 15(4):353-363 (1993).

The Effect of Baseline on Depth Estimation

Figure 2: An example scene. The grid pattern in the background has ambiguity of matching.



I_1 I_2 I_{10}

5b

8b

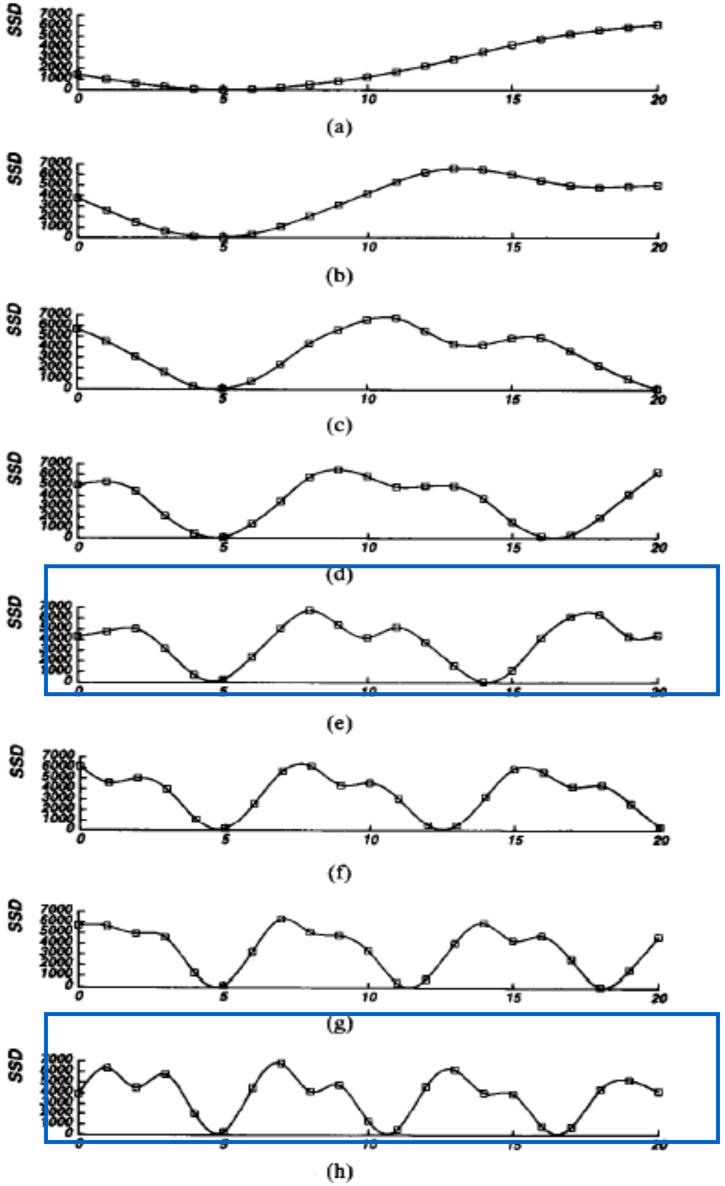


Fig. 5. SSD values versus inverse distance: (a) $B = b$; (b) $B = 2b$; (c) $B = 3b$; (d) $B = 4b$; (e) $B = 5b$; (f) $B = 6b$; (g) $B = 7b$; (h) $B = 8b$. The horizontal axis is normalized such that $8bF = 1$.

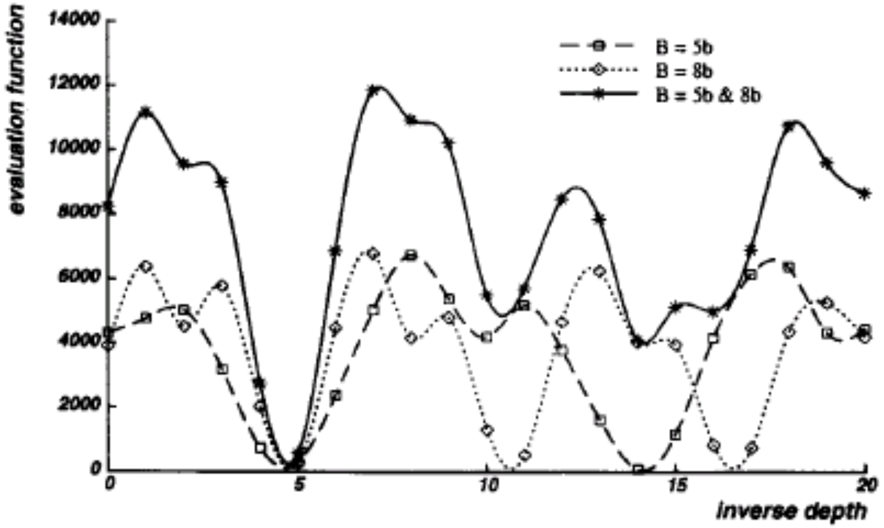


Fig. 6. Combining two stereo pairs with different baselines.

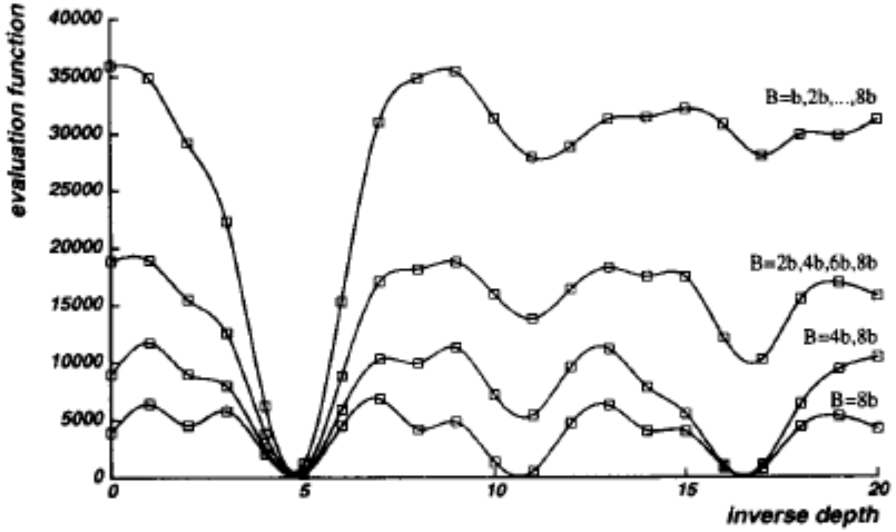
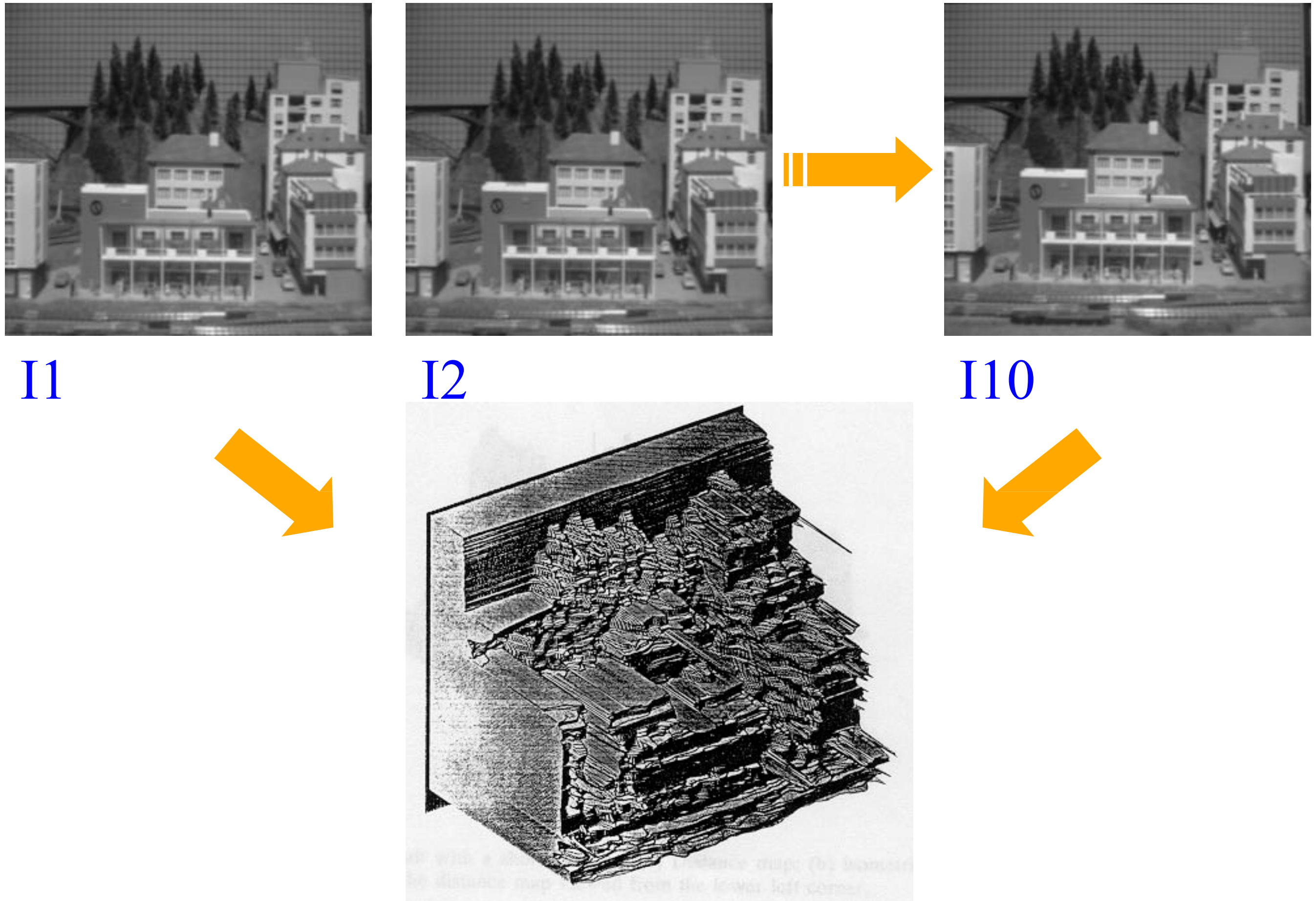


Fig. 7. Combining multiple baseline stereo pairs.

Multiple-baseline stereo results



M. Okutomi and T. Kanade, *A Multiple-Baseline Stereo System*, IEEE Trans. on Pattern Analysis and Machine Intelligence, 15(4):353-363 (1993).

Takeo Kanade

8 languages

Article Talk

Read Edit View history

From Wikipedia, the free encyclopedia

Takeo Kanade (金出 武雄, *Kanade Takeo*, born October 24, 1945 in Hyōgo) is a Japanese computer scientist and one of the world's foremost researchers in computer vision. He is U.A. and Helen Whitaker Professor at Carnegie Mellon University. He has approximately 300 peer-reviewed academic publications and holds around 20 patents.[1]

Honors and achievements [edit]

- In 1997, he was elected to the US National Academy of Engineering for contributions to computer vision and robotics.[2]
- In 1997, he was elected to the American Academy of Arts and Sciences
- In 1999 he was inducted as a Fellow of the Association for Computing Machinery.
- In 2008 Kanade received the Bower Award and Prize for Achievement in Science from The Franklin Institute in Philadelphia, Pennsylvania.[3]
- A special event called TK60: Celebrating Takeo Kanade's vision was held to commemorate his 60th birthday.[4] This event was attended by prominent computer vision researchers.
- Elected member of American Association of Artificial Intelligence, Robotics Society of Japan, and Institute of Electronics and Communication Engineers of Japan
- Marr Prize, 1990 for the paper Shape from Interreflections which he co-authored with Shree K. Nayar and Katsushi Ikeuchi[5]
- Longuet-Higgins Prize for lasting contribution in computer vision at
  - CVPR 2006 for the paper "Neural Network-Based Face Detection"[6] coauthored with H. Rowley and S. Baluja[7]
  - CVPR 2008[8] for the paper "Probabilistic modeling of local appearance and spatial relationships for object recognition"[9] coauthored with H Schneiderman
- The other awards he has received include the C&C Award, the Joseph Engelberger Award, FIT Funai Accomplishment Award, the Allen Newell Research Excellence Award, and the JARA Award.
- He has served for many government, industrial, and university advisory boards, including the Aeronautics and Space Engineering Board (ASEB) of the National Research Council, NASA's Advanced Technology Advisory Committee, PITAC Panel for Transforming Healthcare Panel, and the Advisory Board of Canadian Institute for Advanced Research.[10]
- In 2016 Kanade received the Kyoto Prize in Information Sciences.[11]

Takeo Kanade

Dr Takeo Kanade at the 2016 Kyoto Prize Presentation Ceremony

Born	October 24, 1945 (age 77) Hyōgo, Japan
Nationality	Japanese
Alma mater	Kyoto University
Known for	Lucas–Kanade method Tomasi-Kanade method Face Detection Virtualized Reality
Awards	Kyoto Prize (2016) Bowers Award (2008) NAE Member (1997)
Scientific career	
Fields	Computer vision Robotics

Multibaseline Stereo

Basic Approach

- Choose a reference view
- Use your favorite stereo algorithm BUT
  - replace two-view SSD with **SSSD** over all baselines
  - **SSSD**: the SSD values are computed first for each pair of stereo images, and then add all together from multiple stereo pairs.

Limitations

- Only gives a depth map (not an "object model")
- Won't work for widely distributed views.

Popular matching scores

- SSD (Sum of Squared Differences) $$\sum_{x,y} |W_1(x,y) - W_2(x,y)|^2$$
- SAD (Sum of Absolute Differences) $$\sum_{x,y} |W_1(x,y) - W_2(x,y)|$$
- ZNCC (Zero-mean Normalized Cross Correlation)

$$\frac{\sum_{x,y}(W_1(x,y) - \overline{W_1})(W_2(x,y) - \overline{W_2})}{\sigma_{W_1}\sigma_{W_2}}$$

  - where $$\overline{W_i} = \frac{1}{n}\sum_{x,y} W_i \qquad \sigma_{W_i} = \sqrt{\frac{1}{n}\sum_{x,y}(W_i - \overline{W_i})^2}$$

  - what advantages might NCC have?

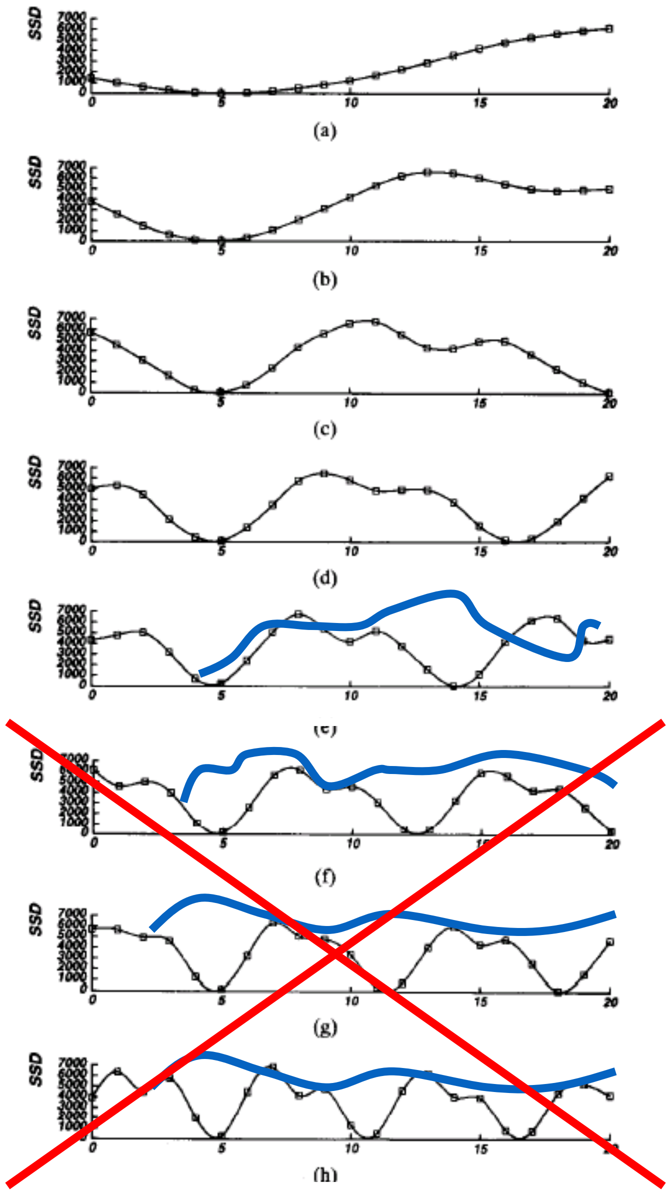


Fig. 5. SSD values versus inverse distance: (a) $B = b$; (b) $B = 2b$; (c) $B = 3b$; (d) $B = 4b$; (e) $B = 5b$; (f) $B = 6b$; (g) $B = 7b$; (h) $B = 8b$. The horizontal axis is normalized such that $8bF = 1$.

Problem: *visibility*

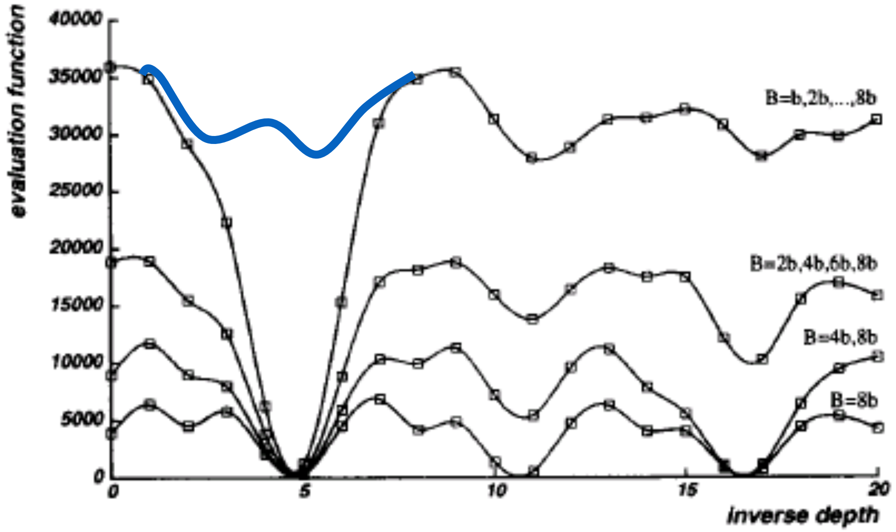


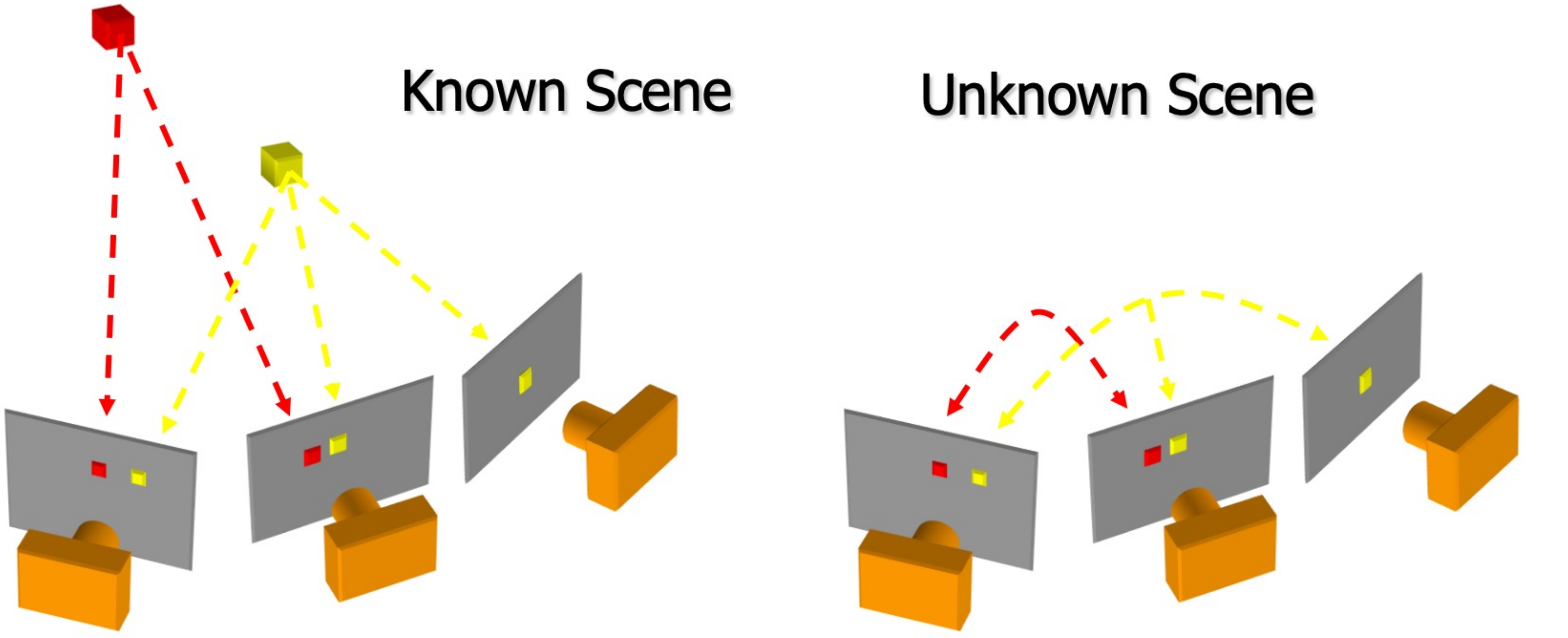
Fig. 7. Combining multiple baseline stereo pairs.

Some Solutions

- Match only nearby photos [Narayanan 98]
- Use NCC instead of SSD,
  Ignore NCC values > threshold
  [Hernandez & Schmitt 03]

Visibility

Which points are visible in which images?

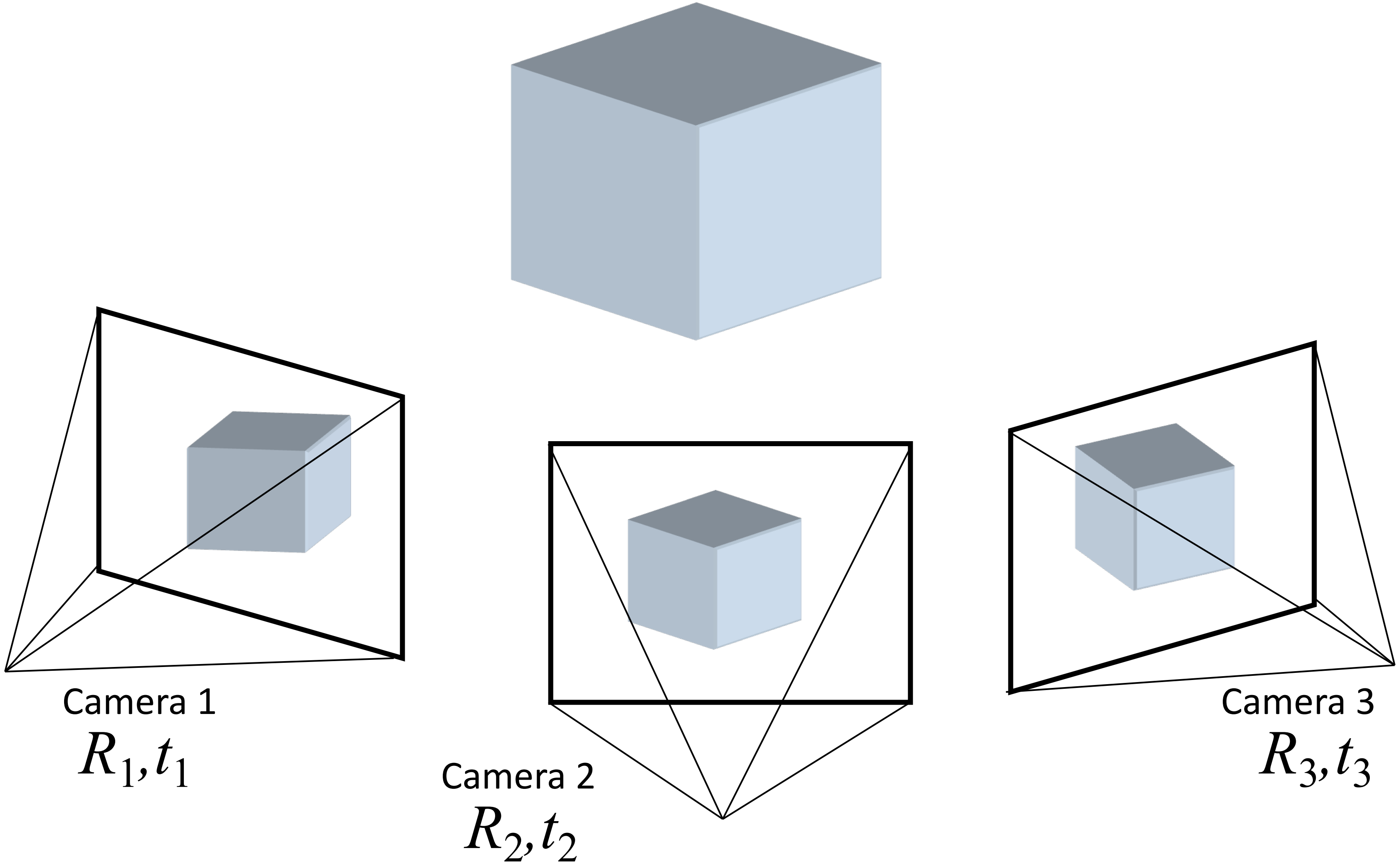


Forward Visibility *Inverse Visibility*

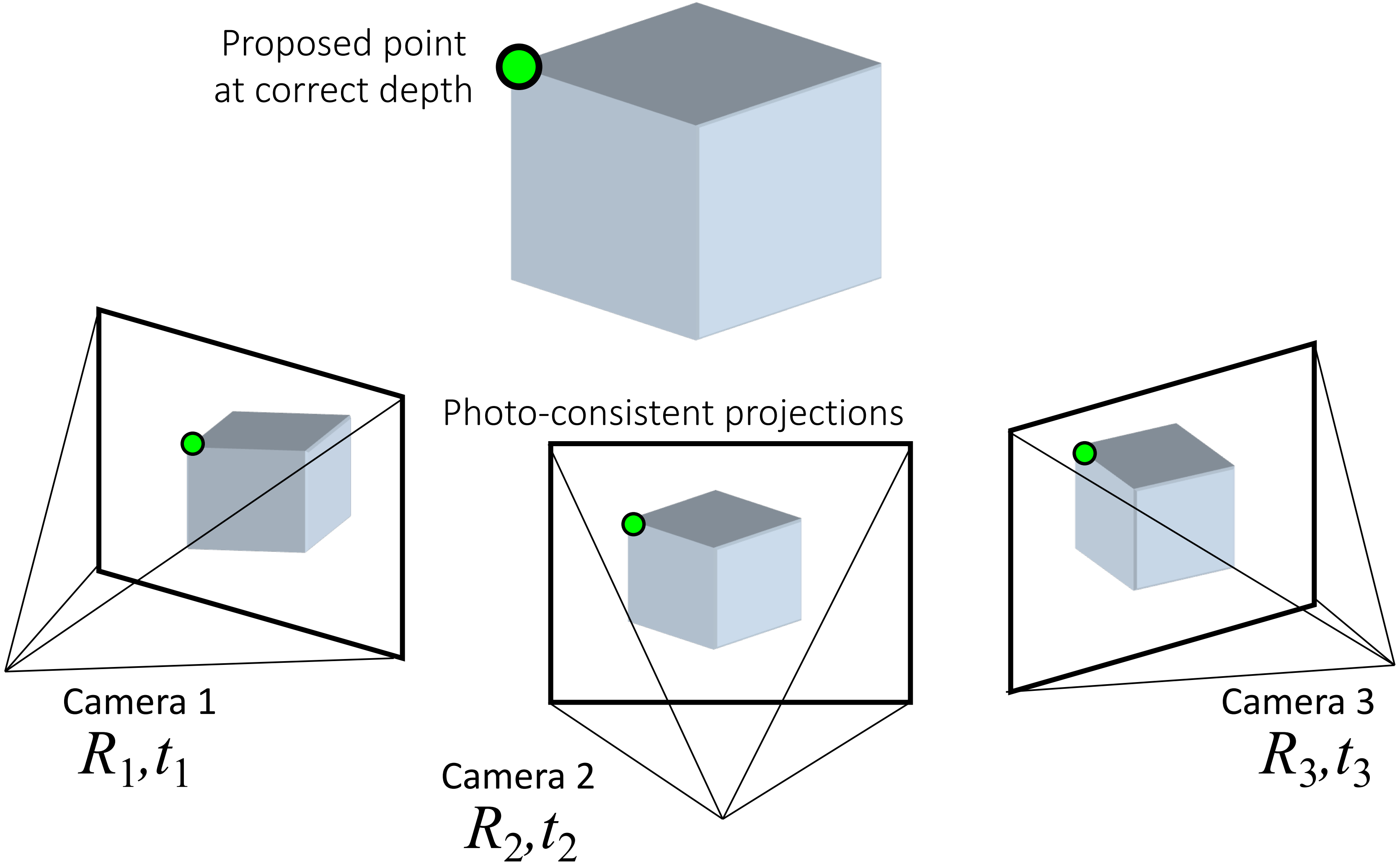
Today's class

- Motivation
- Simple Approach to MVS
- Advanced Approach to MVS
  - **Plane Sweep Stereo**
  - Space Curving Stereo
- Converting depth to mesh
- MVS in deep learning era (more later)

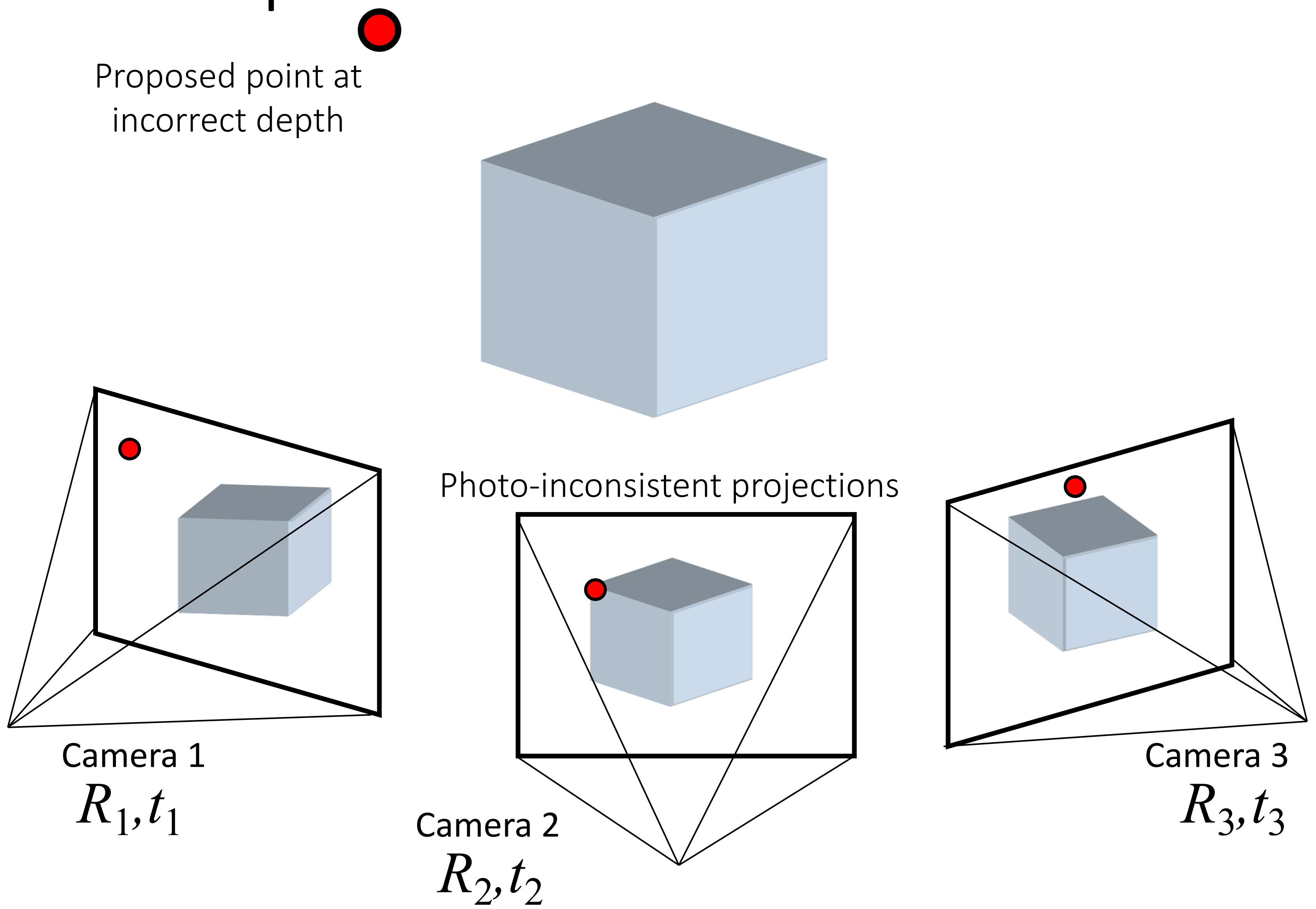
Plane-Sweep Stereo



Plane-Sweep Stereo

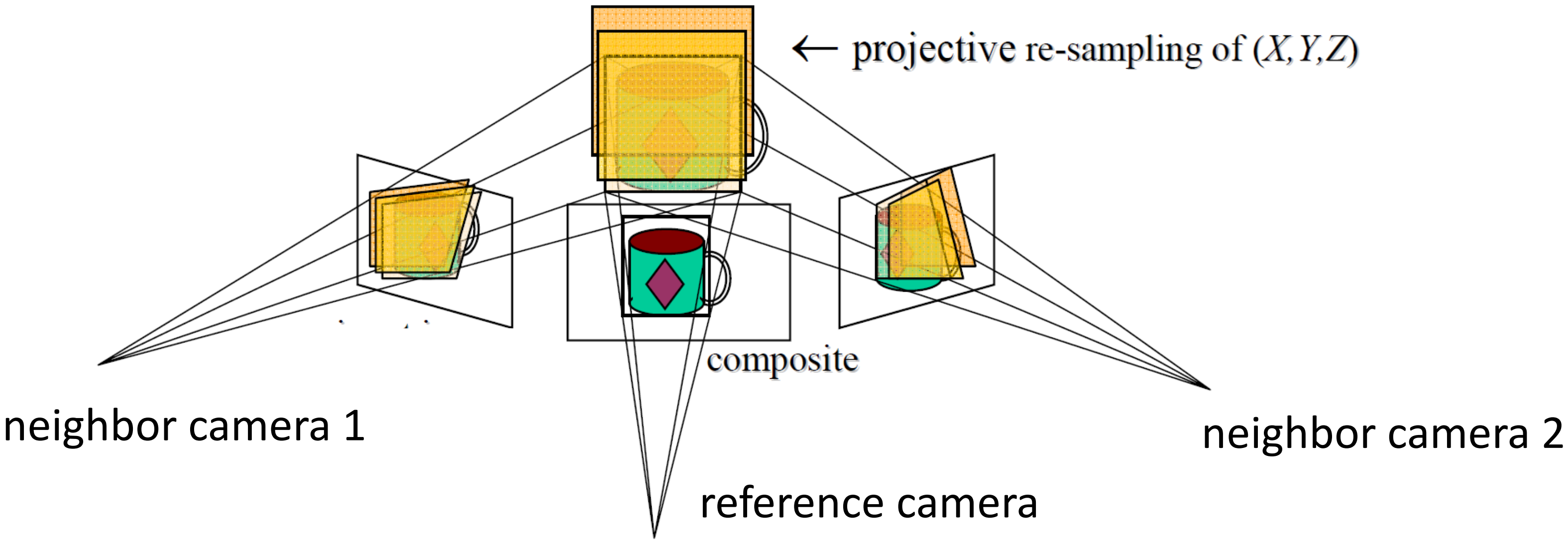


Plane-Sweep Stereo



Plane-Sweep Stereo

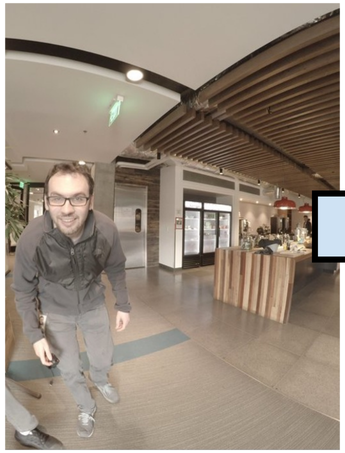
- Sweep family of planes parallel to the reference camera image plane
- Reproject neighbors onto each plane and compare reprojections



Example


Left neighbor


Reference image

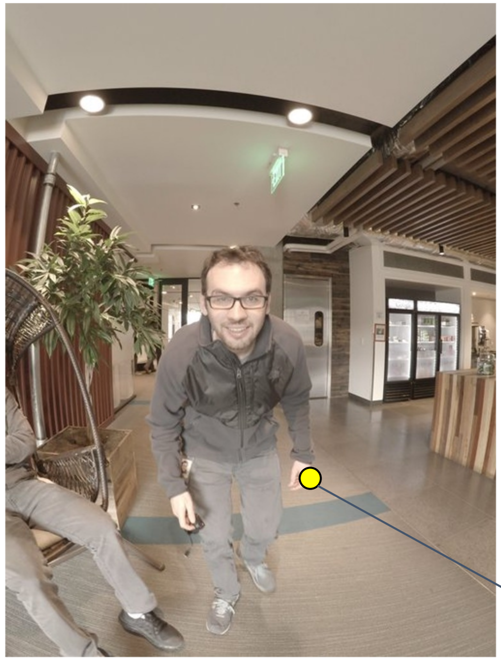

Right neighbor


Planar image reprojections swept over depth (averaged)

For a particular depth sweep, some regions in the average image appears sharp, i.e. photo-consistent.

Starts with near-depth and sweeps till far-depth

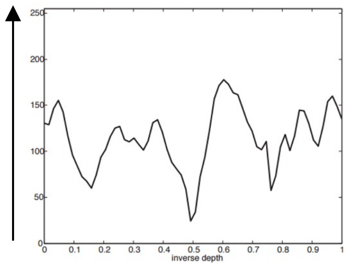
Cost Volumes -> Depth Maps



Reference image

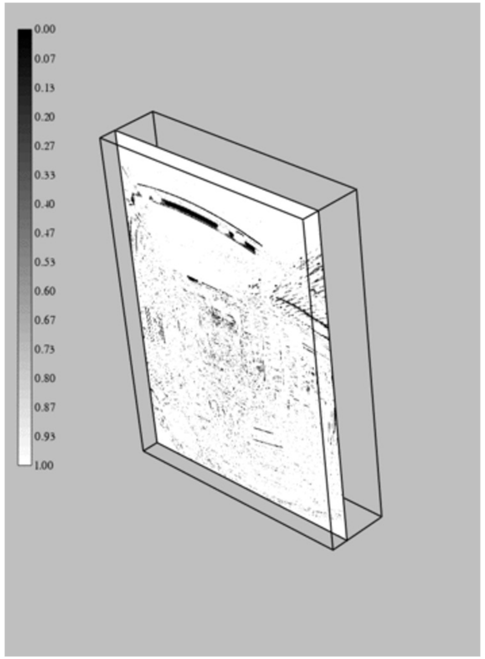
Plane sweep

cost



depth

Single pixel's cost profile

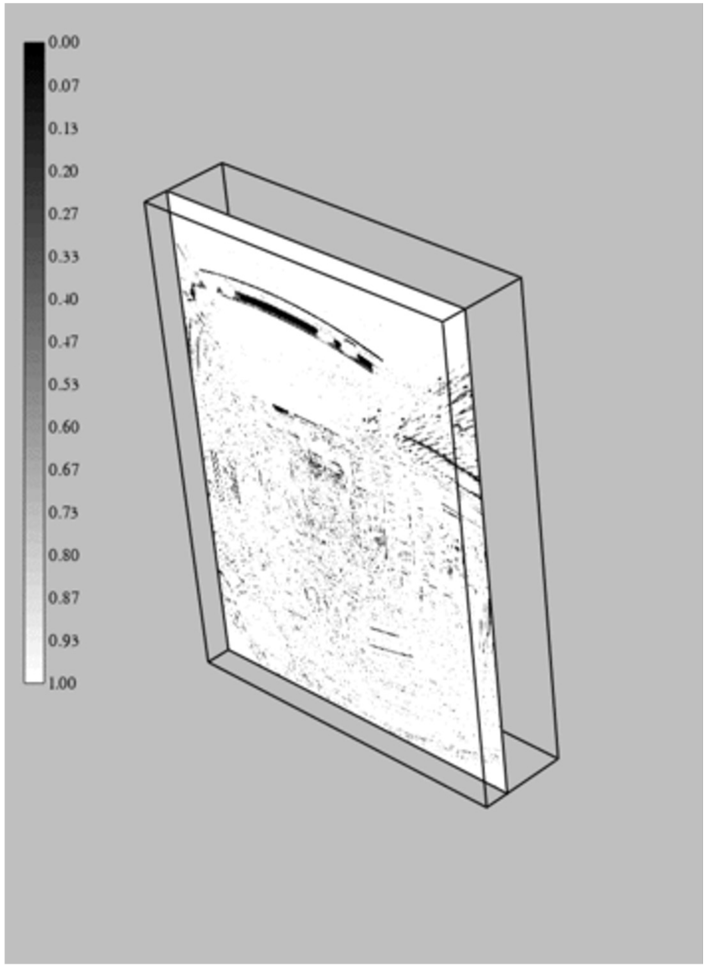


Full cost volume

Depth map solver

(Belief propagation, graph cuts, etc.)

Cost Volume



Cost Volume is a 3D tensor WxHxD for D depth proposal

C(u,v,d) indicates photo-consistency between (u,v) pixel in the reference image to that of it's corresponding pixels in neighboring images assuming the depth proposal at d.

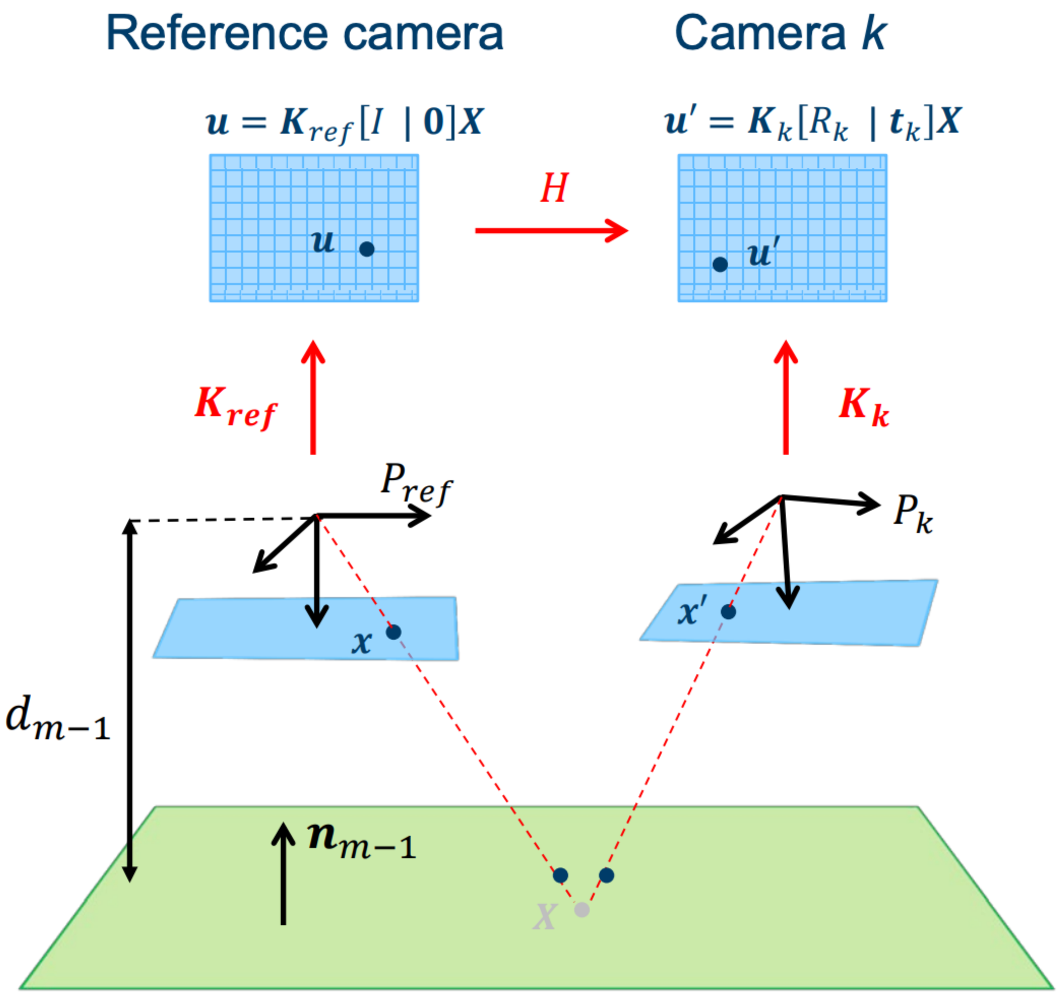
There are various measures of photo consistency, a simple one is variance.

How do we find the corresponding pixels in neighboring images?

- Backproject every pixel using camera matrix (Computationally expensive)
- **Homography transformation over the whole image.**

Plane-Sweep Stereo

- For each depth plane proposal d_m, *m*ap each target image I_k to the reference image I_{ref} using homography (H_{km}^{-1}). Let the warped image be W_{km}.
  - H_{km} can be calculated from the camera parameters and depth of the plane d_m.
- For each pixel (u,v) in the reference image compute similarity scores between W_{km} and I_{ref}.
  - If you use Zero Mean Normalized Cross Correlation, you have ZNCC(I_{ref}(u,v), W_{km}(u,v))



Robert Collins, A Space-Sweep Approach to True Multi-Image Matching, CVPR 1996.
D. Gallup, J.-M. Frahm, P. Mordohai, Q. Yang and M. Pollefeys, Real-Time Plane-Sweeping Stereo with Multiple Sweeping Directions, CVPR 2007

Plane-Sweep Stereo

- The family of depth planes in the coordinate frame of the reference view

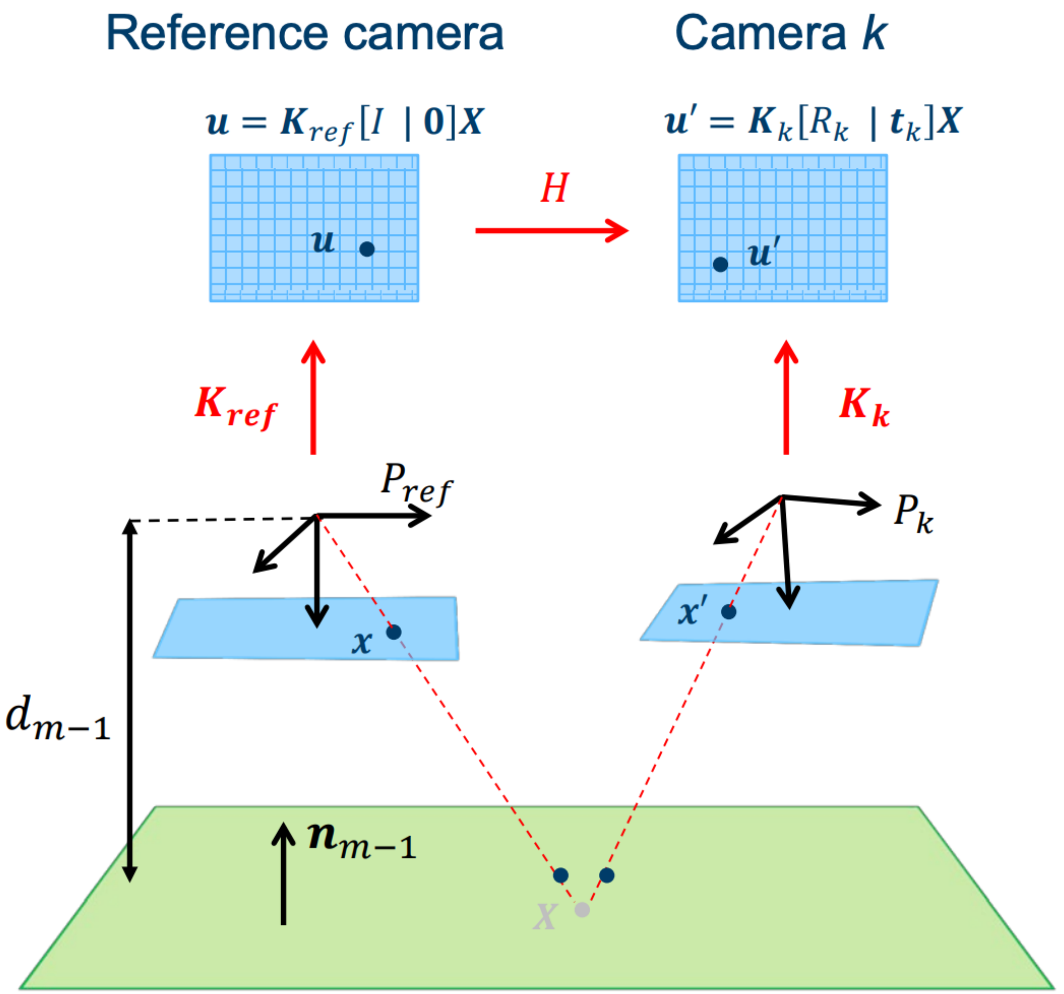
$$\Pi_m = \begin{bmatrix} \mathbf{n}_m^T & -d_m \end{bmatrix}$$

- The mapping from the reference camera P_{ref} onto the plane Π_m and back to camera P_k is described by the homography induced by the plane Π_m

$$H_{\Pi_m, P_k} = K_k \left(R_k - \mathbf{t}_k \mathbf{n}_m^T / d_m \right) K_{ref}^{-1}$$

Try the proof in HW!

- The mapping from P_k to P_{ref} induced by Π_m is the inverse homography H_{Π_m, P_k}^{-1}



Slight abuse of notation. In equation (x,y) are image co-ordinates, in figure u is image co-ordinate.

Robert Collins, A Space-Sweep Approach to True Multi-Image Matching, CVPR 1996.
D. Gallup, J.-M. Frahm, P. Mordohai, Q. Yang and M. Pollefeys, Real-Time Plane-Sweeping Stereo with Multiple Sweeping Directions, CVPR 2007

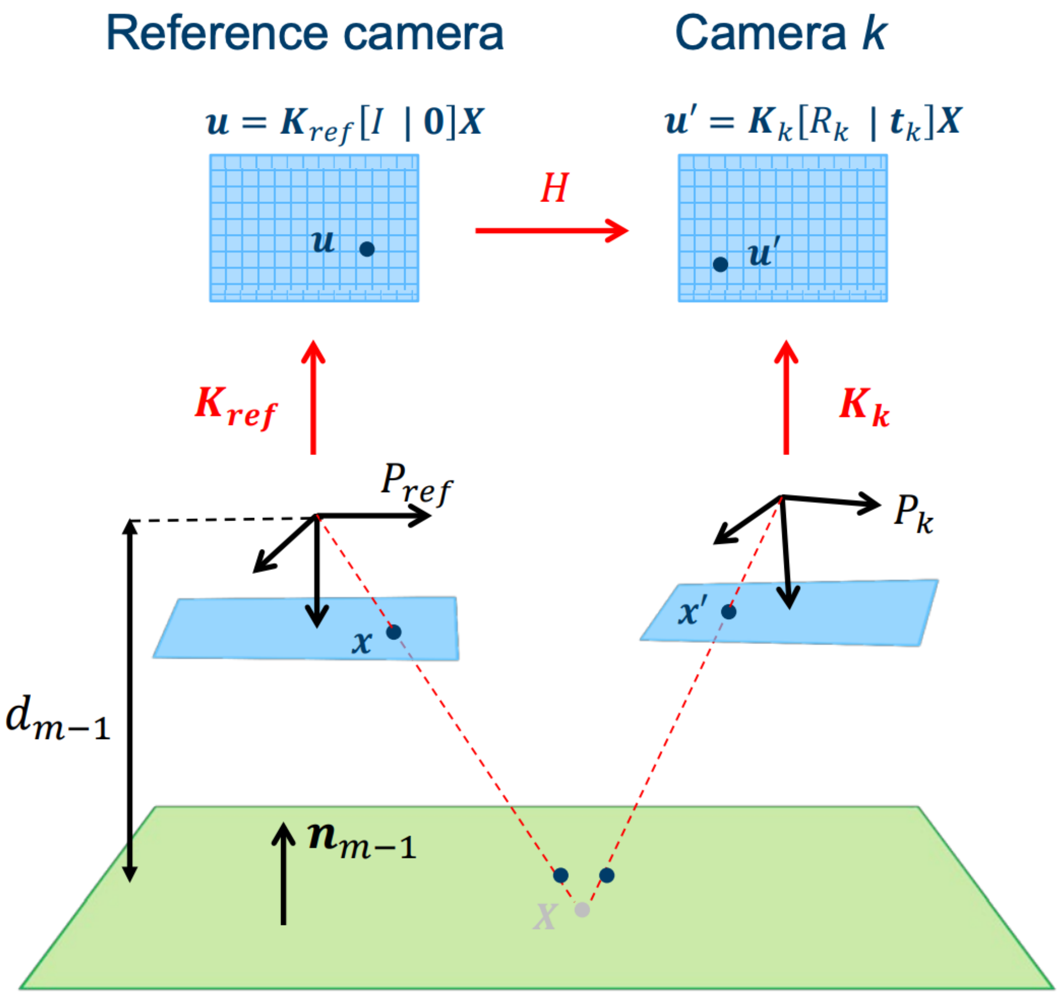
Plane-Sweep Stereo

- For each pixel (u,v) in the reference image compute similarity scores between W_{km} and I_{ref}, as ZNCC(I_{ref}(u,v), W_{km}(u,v))

- Create a cost volume C(u,v,m) = sum (ZNCC(I_{ref}(u,v), W_{km}(u,v)) over all k target images.

- Greedy: At each pixel choose the maximum of the cost volume as the correct depth.

- Non-greedy: Use advanced techniques like belief propagation, graph cut, or 3D convolution.



Robert Collins, A Space-Sweep Approach to True Multi-Image Matching, CVPR 1996.
D. Gallup, J.-M. Frahm, P. Mordohai, Q. Yang and M. Pollefeys, Real-Time Plane-Sweeping Stereo with Multiple Sweeping Directions, CVPR 2007

Plane-Sweep Stereo

Left neighbor

Reference image

Right neighbor

Left neighbor projected into reference image

Average images on each plane

Right neighbor projected into reference image

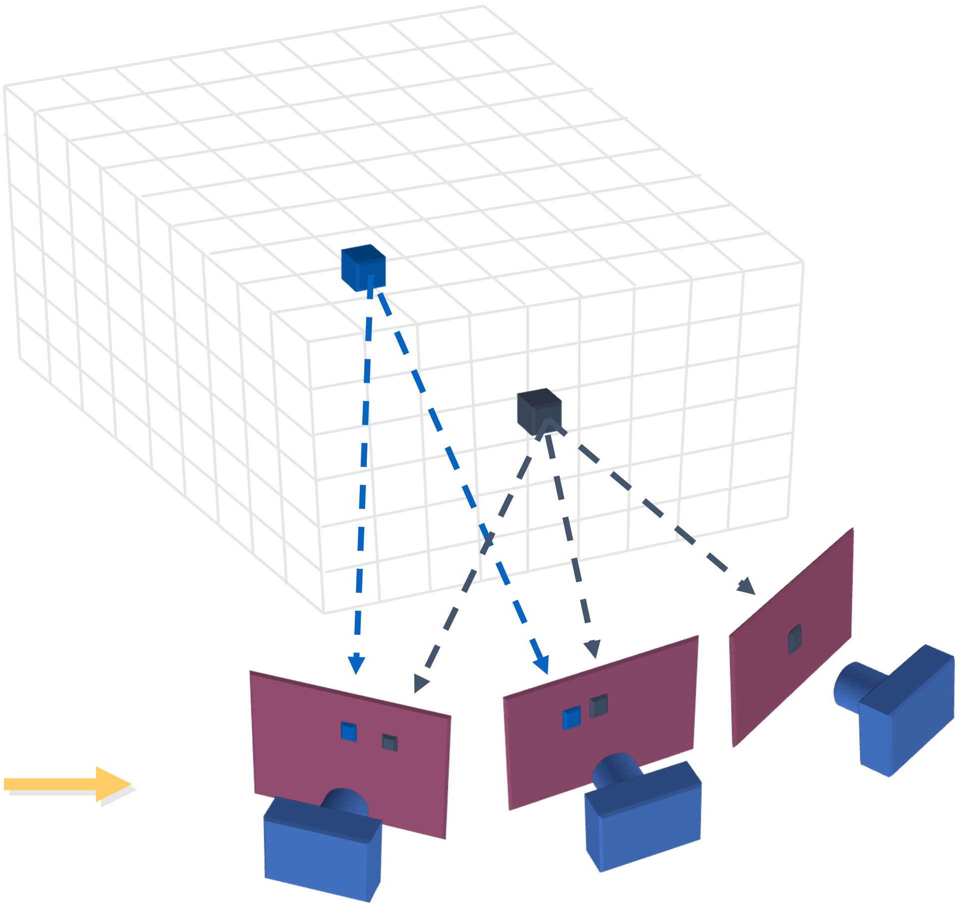
Today's class

- Motivation
- Simple Approach to MVS
- Shape representations
- Advanced Approach to MVS
  - Plane Sweep Stereo
  - **Space Curving Stereo**
- Converting depth to mesh
- MVS in deep learning era (more later)

Volumetric stereo

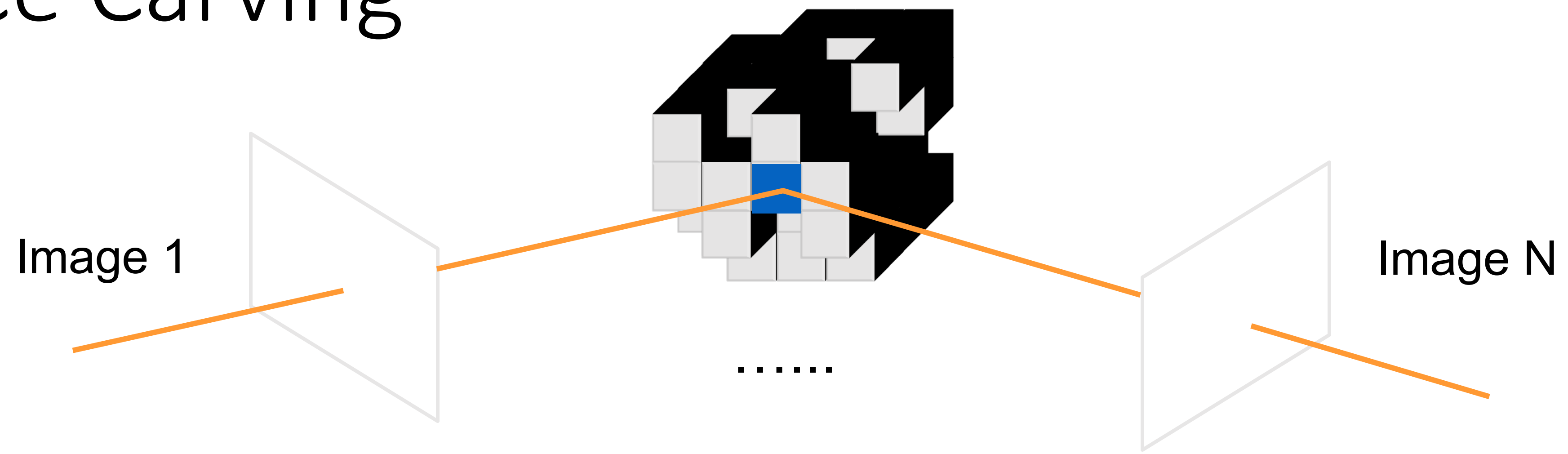
Discretized Scene Volume

Input Images (Calibrated)



Goal: **Assign RGB values to voxels in V**
photo-consistent **with images**

Space Carving



- Space Carving Algorithm
  - Initialize to a volume V containing the true scene
  - Choose a voxel on the outside of the volume
  - Project to visible input images
  - Carve if not photo-consistent
  - Repeat until convergence

K. N. Kutulakos and S. M. Seitz, [A Theory of Shape by Space Carving](), *ICCV* 1999

Space Carving Results

Input Image (1 of 45)

Reconstruction

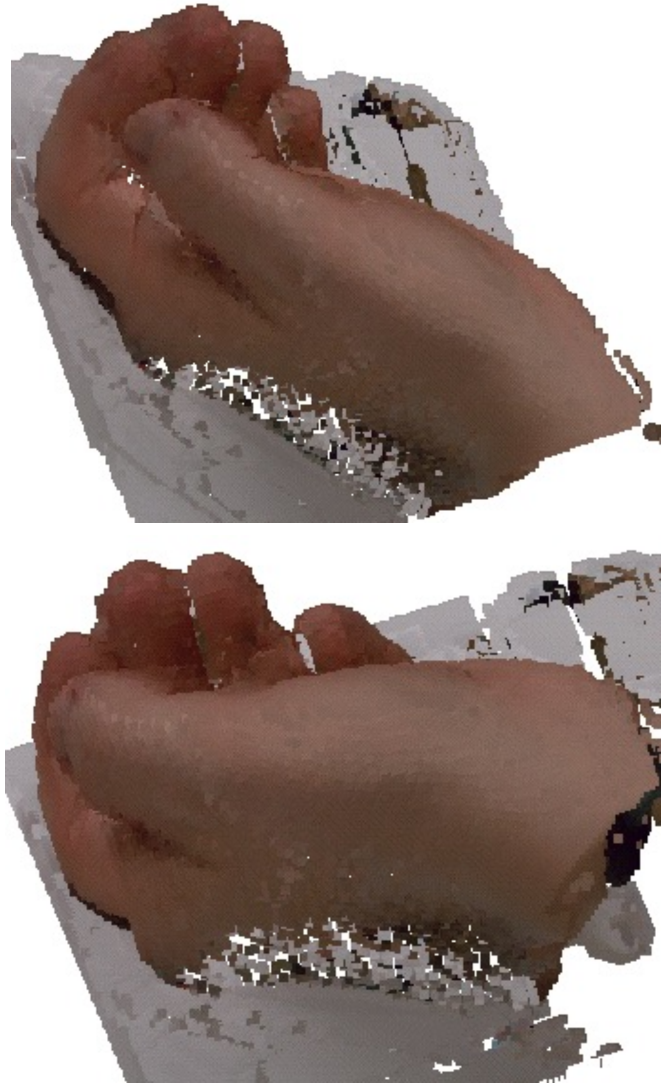
Reconstruction

Reconstruction

Source: S. Seitz

Space Carving Results

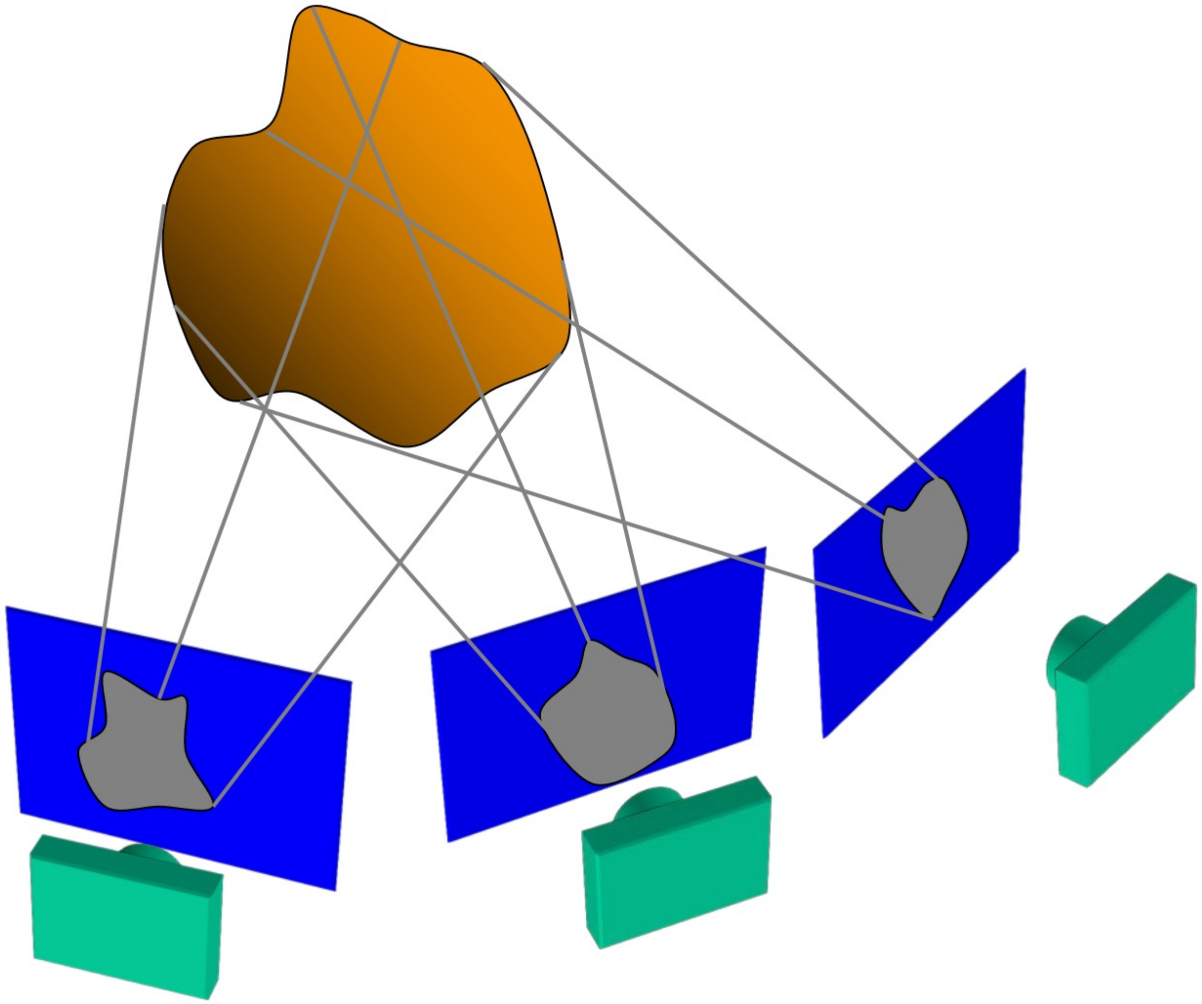
Input Image
(1 of 100)



Reconstruction

Source: S. Seitz

How do you initialize the voxel? Visual Hull Extraction



1. Segment out object from background
2. Backproject each silhouette
3. Intersect backprojected volumes

Summary of approached to MVS

- Plane Sweep Depth maps
  - Robust and adaptable multiple-view stereo matching
  - Real-time applications
  - Fusion of point clouds from different reference views
  - Sampling of scene depends on the reference views
- Volumetric Stereo
  - View-independent representation
  - Need silhouette extraction
  - Accuracy depends on the density of the grid
  - High computational and memory costs

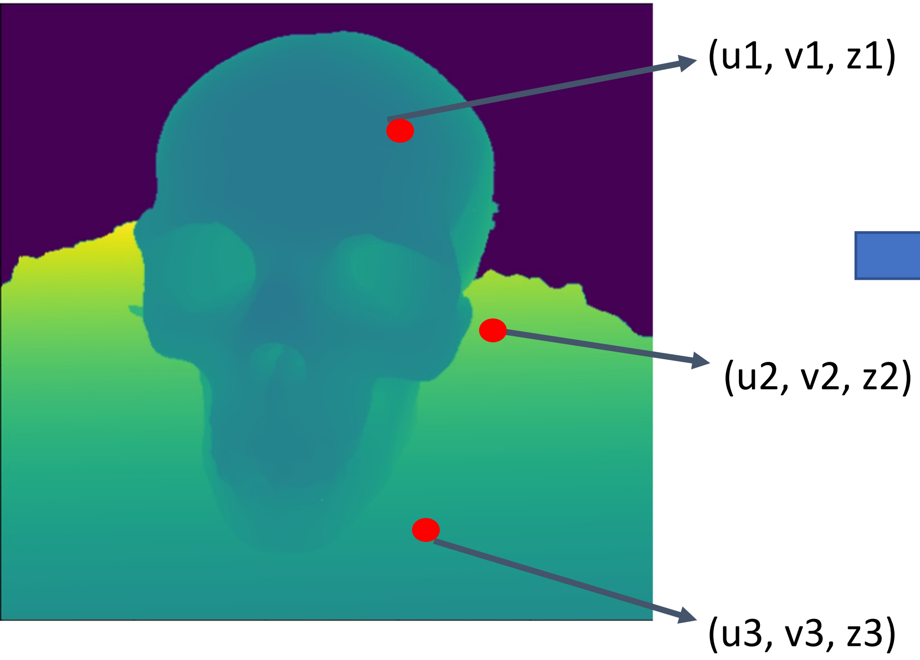
Today’s class

- Motivation
- Simple Approach to MVS
- Shape representations
- Advanced Approach to MVS
  - Plane Sweep Stereo
  - Space Curving Stereo
- Converting depth to mesh
- MVS in deep learning era (more later)

Depth Map to Point Cloud

Credits: Pierre-Nicolas Perrin

(u,v) are in image coordinate.
z is in camera coordinate



Convert (u,v) in image coordinate to (x,y) in camera coordinate.

$$x = \frac{z(u - c_x)}{f_x}$$

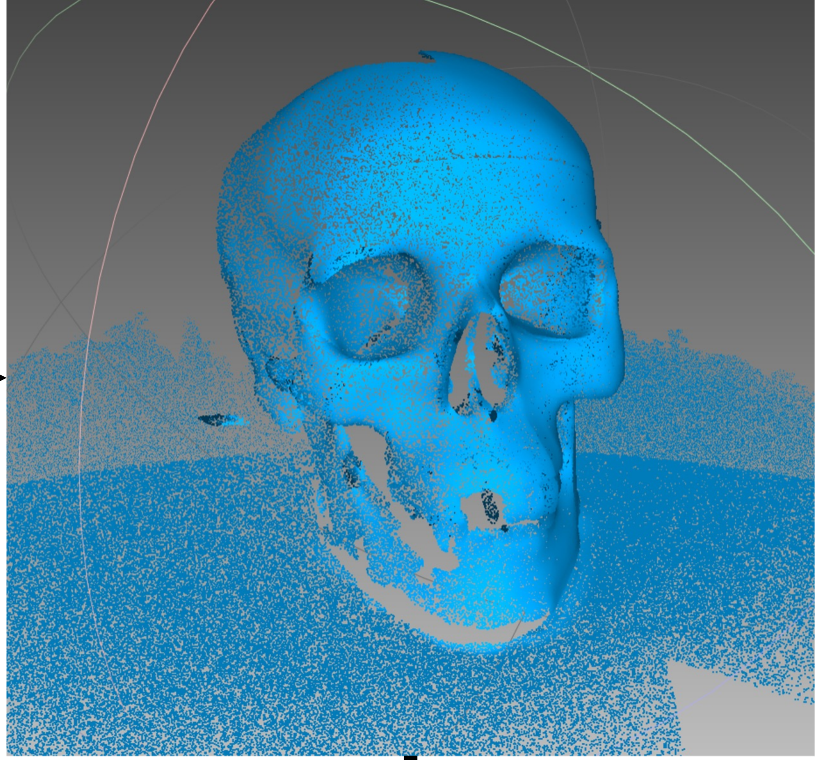
$$y = \frac{z(v - c_y)}{f_y}$$

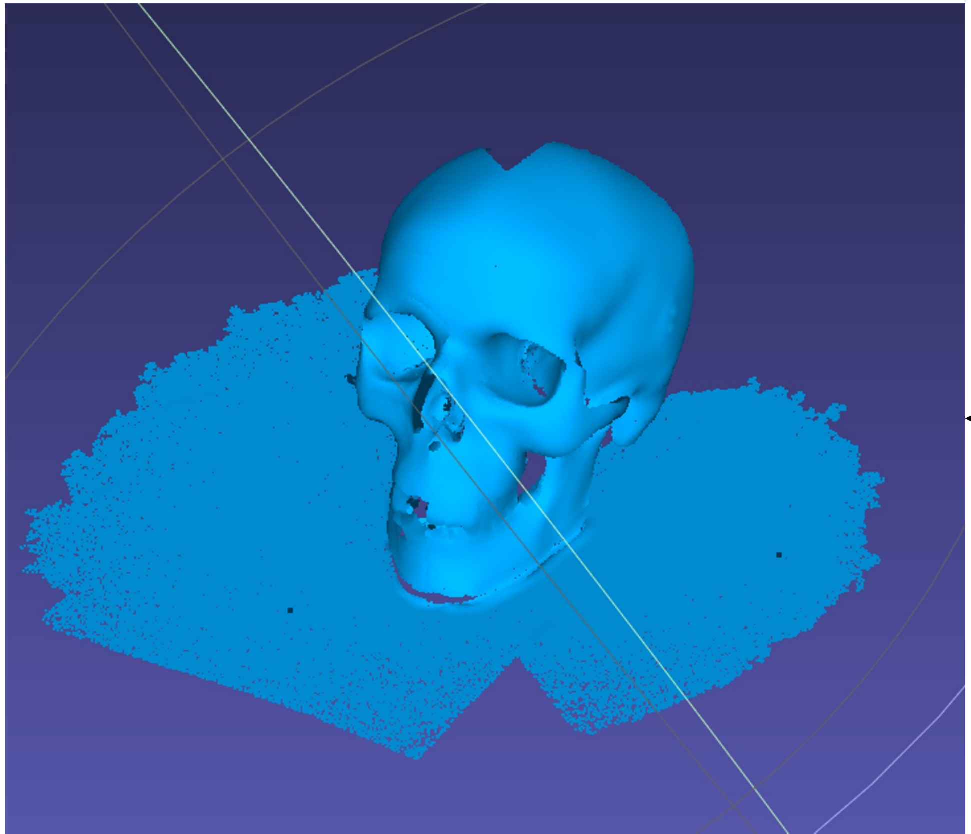
Convert from camera to world coordinate system

$$\begin{bmatrix} x_w \\ y_w \\ z_w \end{bmatrix} = \begin{bmatrix} x_c \\ y_c \\ z_c \end{bmatrix} [\mathbf{R}|\mathbf{t}]^{-1}$$

All cameras have different coordinate system.
Different Rotation and Translation.

Credits: Pierre-Nicolas Perrin





How do we obtain mesh from point cloud?

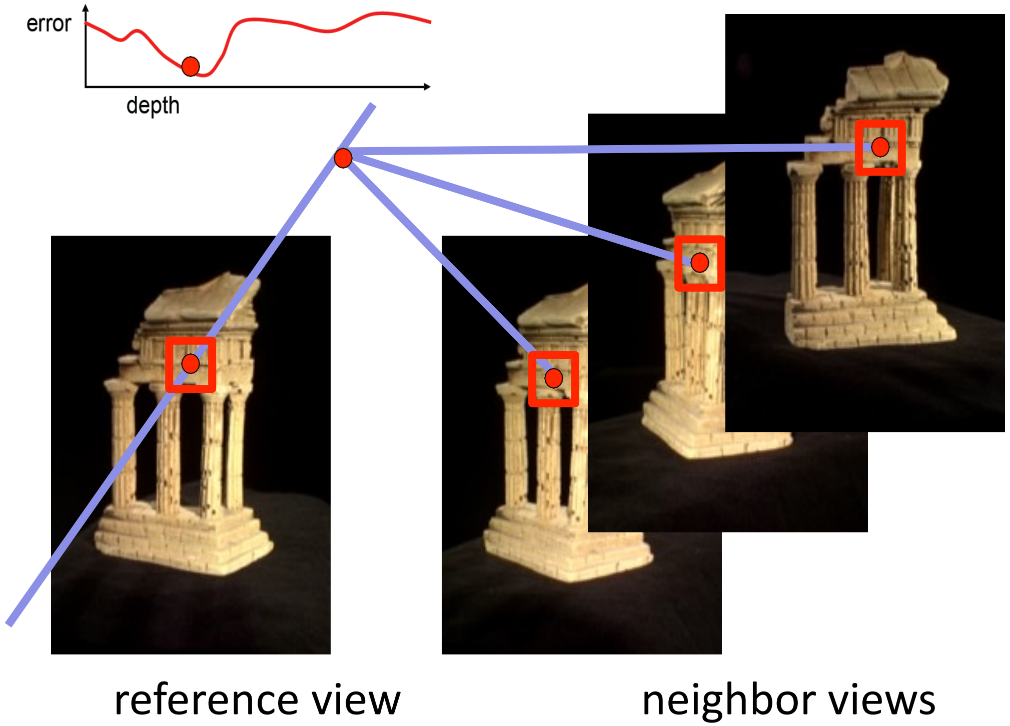
Screened Poisson Surface Reconstruction

Credits: Pierre-Nicolas Perrin

Today's class

- Motivation
- Simple Approach to MVS
- Shape representations
- Advanced Approach to MVS
  - Plane Sweep Stereo
  - Space Curving Stereo
- Converting depth to mesh
- MVS in deep learning era (more later)

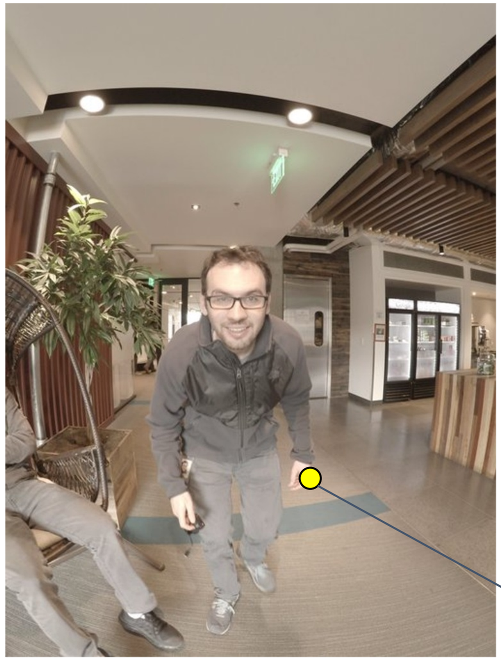
Multi-view stereo: in Deep Learning Era



Classical MVS	Deep MVS
Photo consistency (error metric) is applied on raw image intensities – not so robust w.r.t. illumination, highlights etc.	Photo consistency (error metric) is applied on Deep Features- very robust
Tries to minimize photo consistency error after reprojection	Uses reprojection error (self-supervision) + synthetic/real data with GT.
Cost Volume is used in both.	

Will learn more about this towards the end of the course in details.

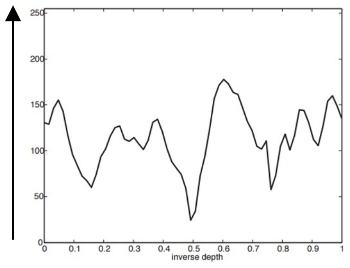
Plane Sweep Stereo in Deep Learning era



Reference image

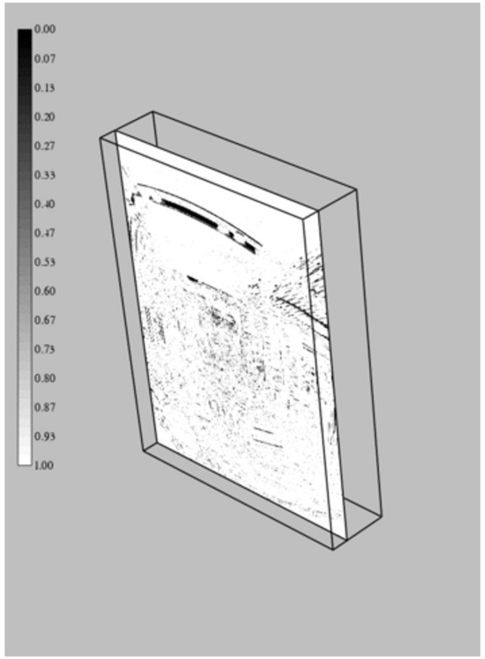
Plane sweep

cost



depth

Single pixel's cost profile



Full cost volume

Depth map solver

(Belief propagation, graph cuts, etc.)

Instead of raw-pixels use deep features

Use 3D convolution to predict depth map from cost volume.

Another approach: NeRF

- Represent scenes as functions from (x, y, z) to RGB and alpha (transparency), use volume rendering to render images



NeRF: Representing Scenes as Neural Radiance Fields for View Synthesis, ECCV 2020

Will learn more about this towards the end of the course in details.

https://www.matthewtancik.com/nerf

Slide Credits

- CS5670, Introduction to Computer Vision, **Cornell Tech, by Noah Snavely.**
- CS 194-26/294-26: Intro to Computer Vision and Computational Photography, **UC Berkeley, by Angjoo Kanazawa.**
- CS 16-385: Computer Vision, **CMU, by Matthew O'Toole.**
- **CSE 486: Computer Vision, by Robert Collins, Penn State.**
- **CS 543 Computer Vision, by Stevlana Lazebnik, UIUC.**