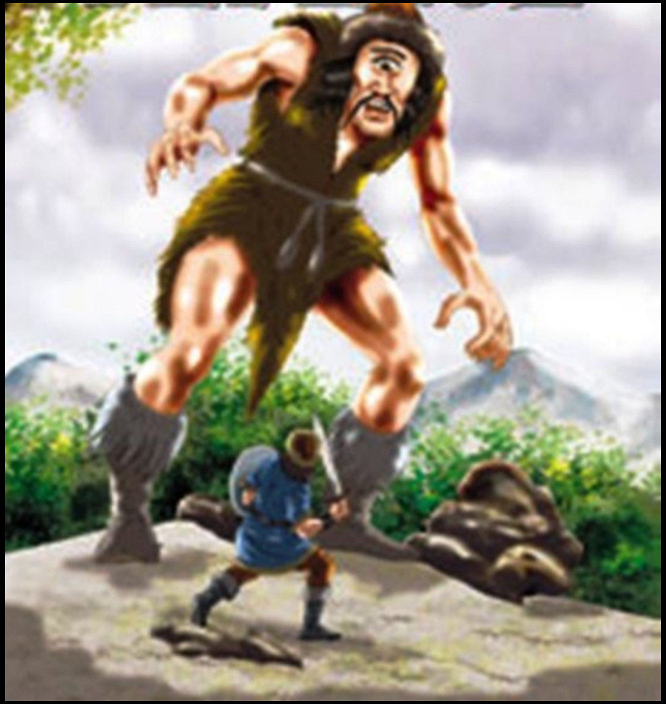
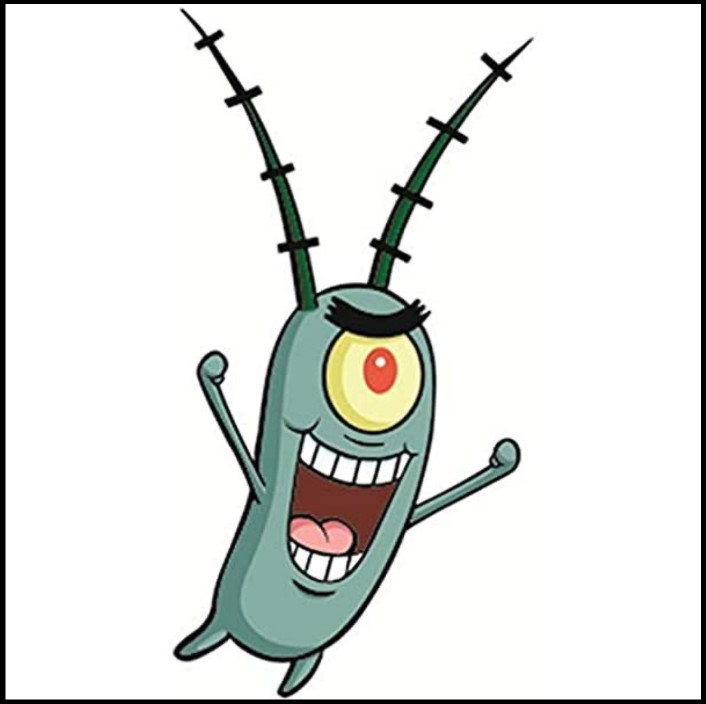
Learning multi-view stereo

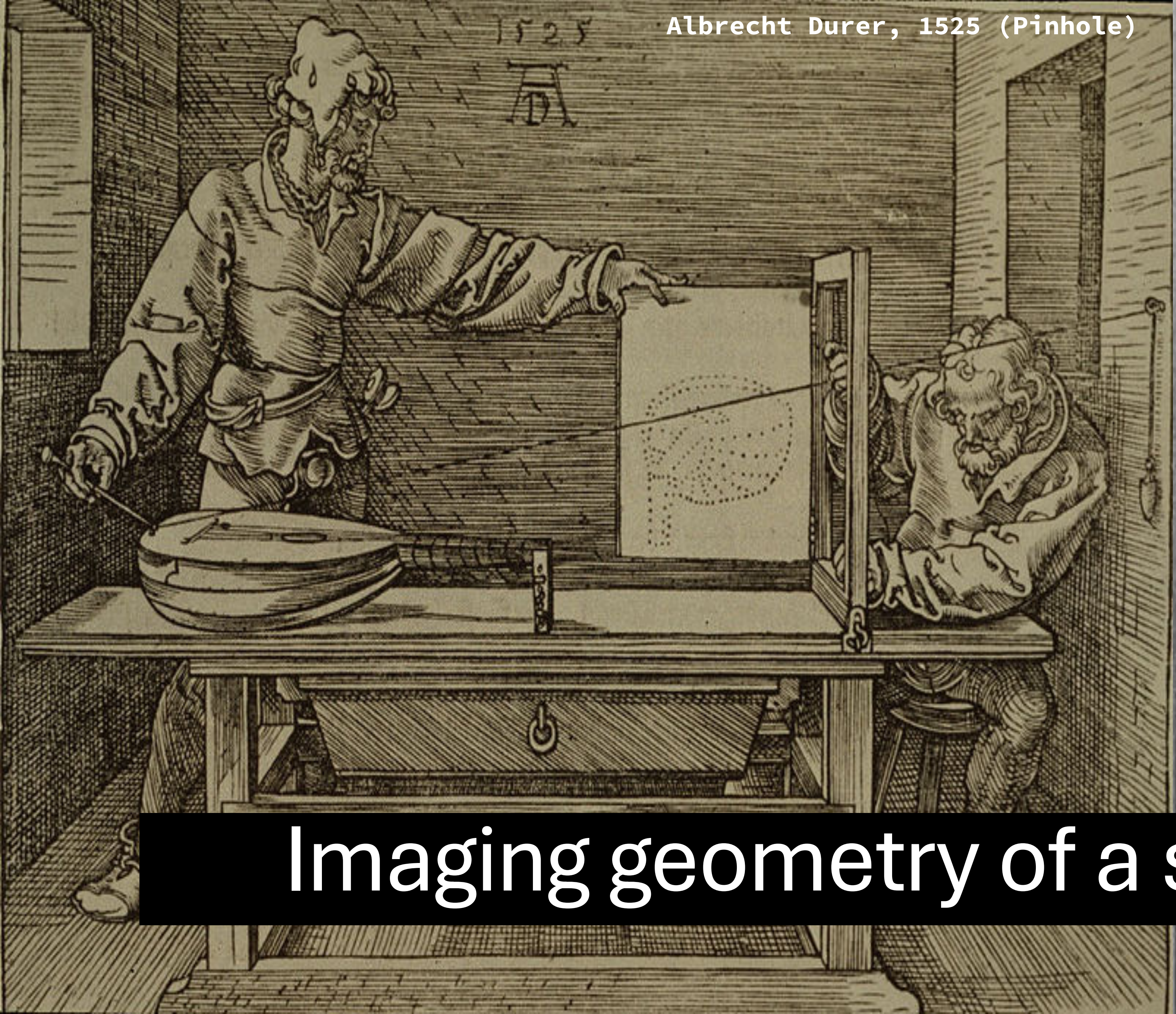
Nail Ibrahimli

What do these characters have in common?



Imaging geometry of a single eye



Albrecht Durer, 1525 (Pinhole)

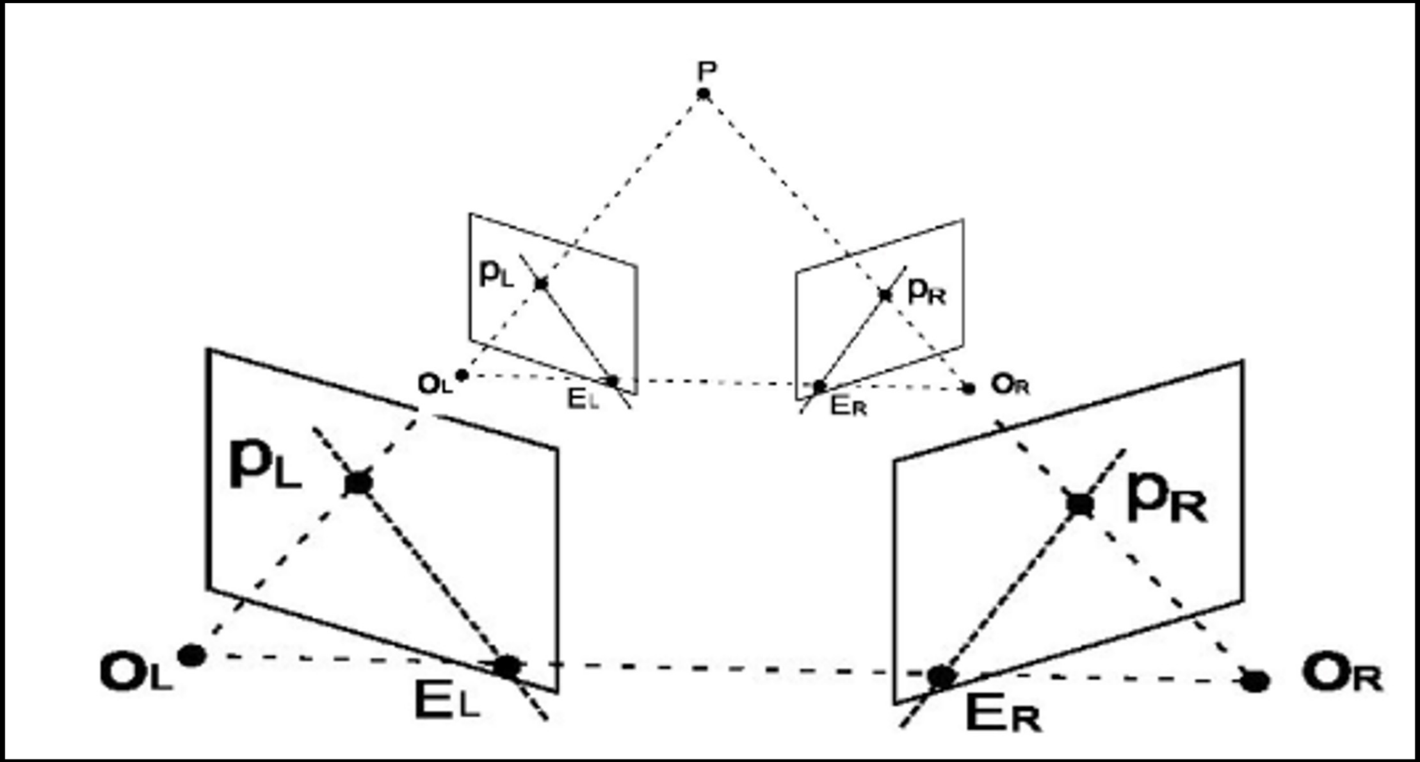
M. C. Escher, 1935 (Omnidirectional)

Limitations of single eye

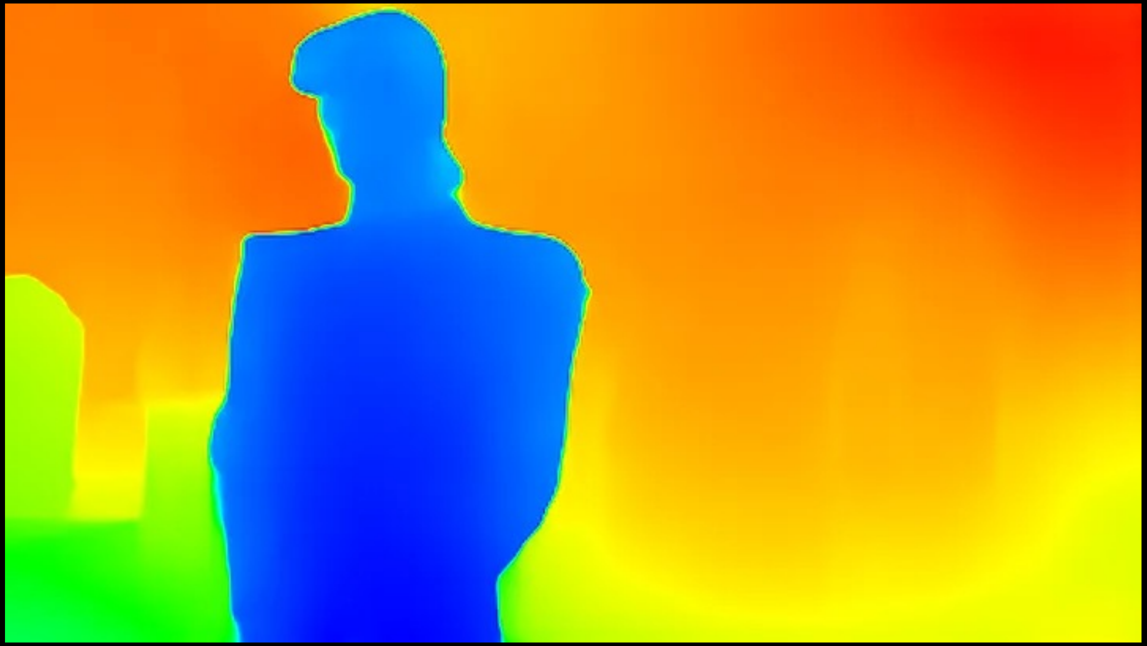
M.C. Escher

Limitations of single eye

M.C. Escher



Ask AI to recover geometry from a single image.

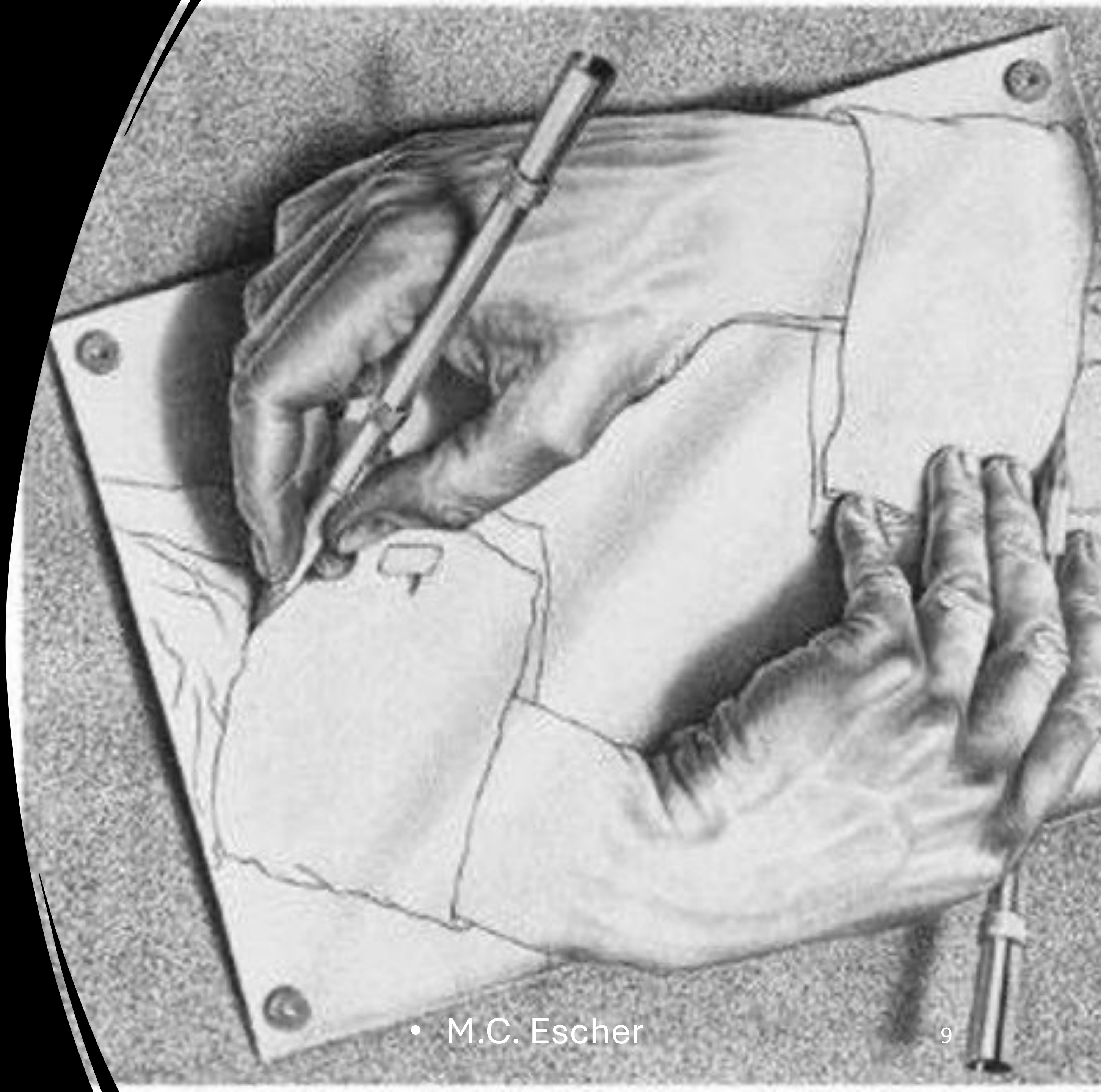




Good looking 2.5D != Good looking 3D

- SOTA 2019-2022 (MiDAS)
- Source (Patricio Gonzalez)

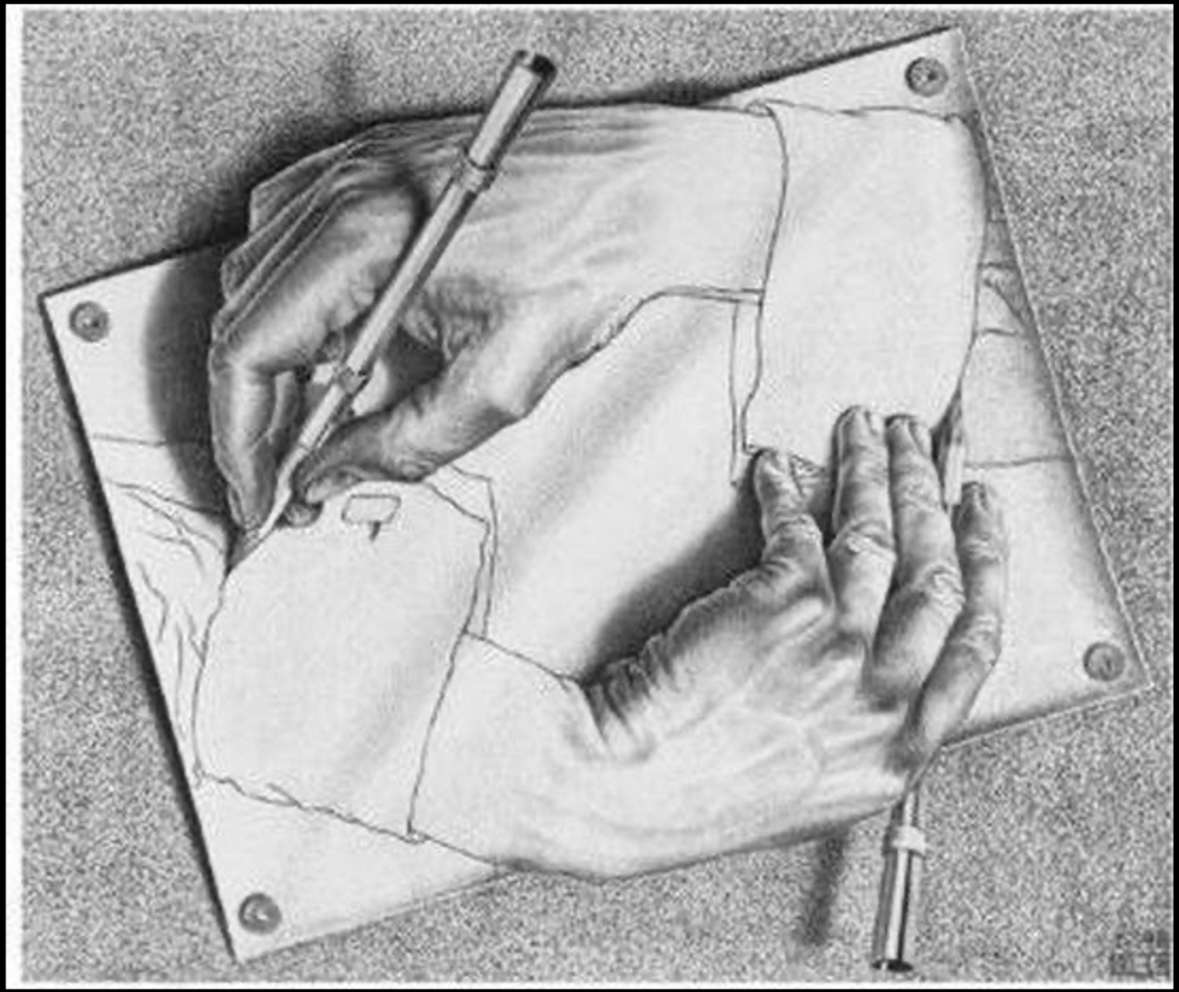
AI : Ugh, come on! How am I supposed to know what to do without you asking the exact same question?

Visual cues for 3D: Shading

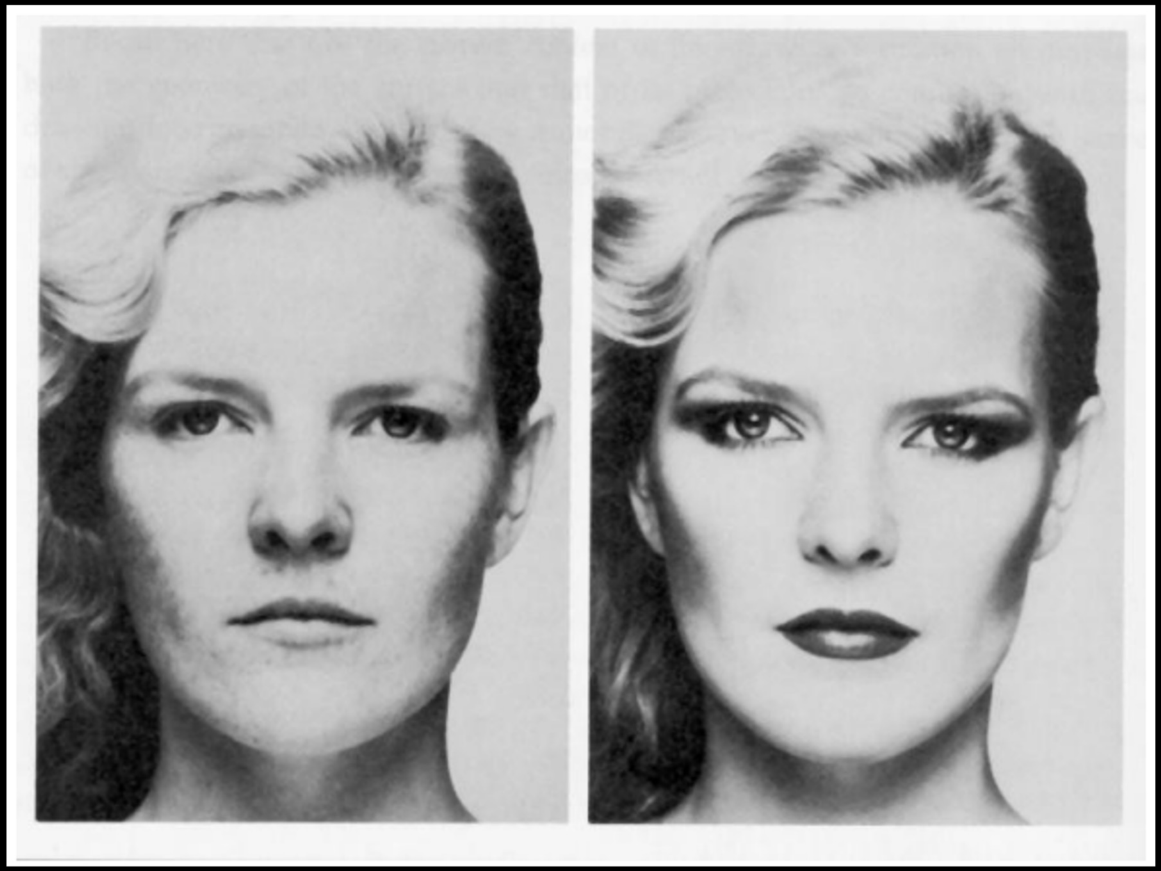


- M.C. Escher

Visual cues for 3D: Shading

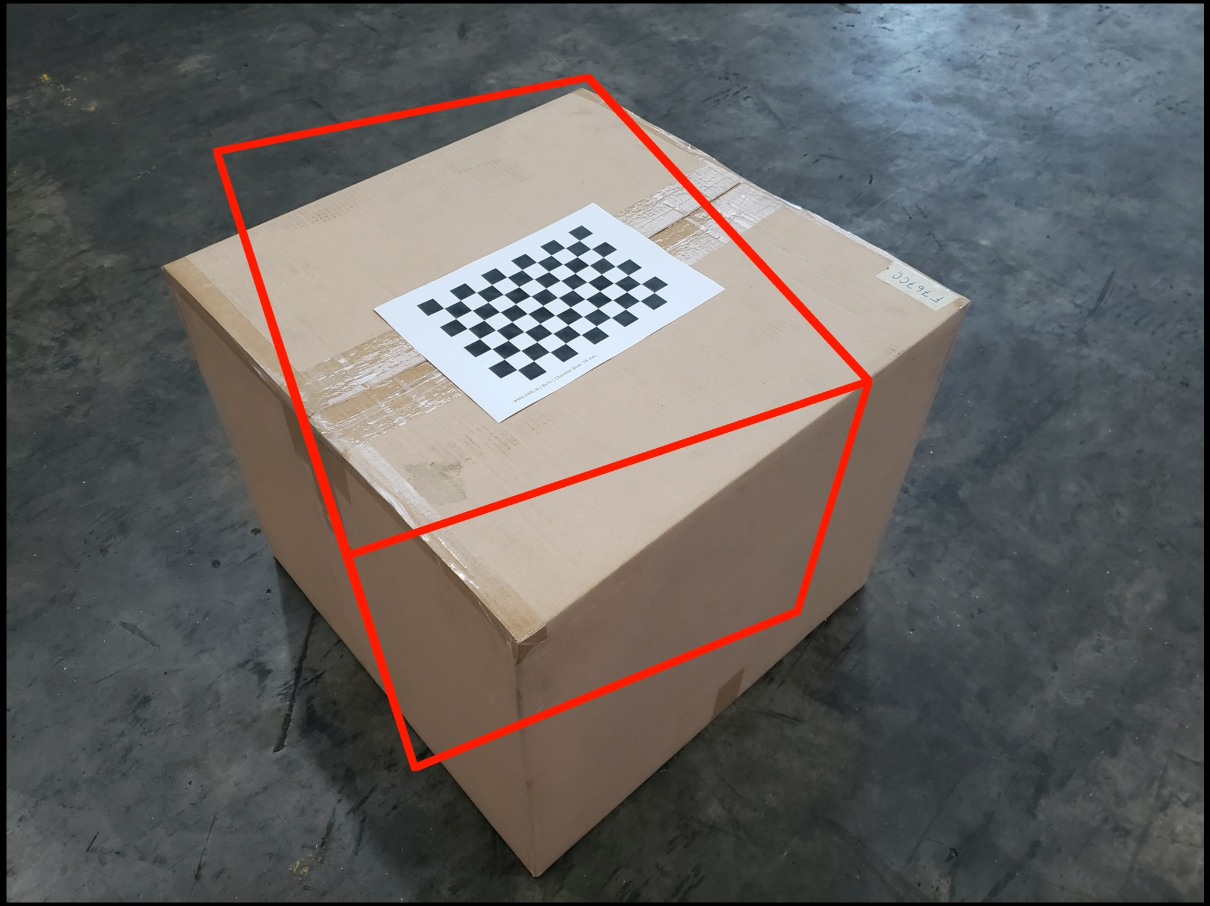


M.C. Escher



Merle Norman Cosmetics

Visual cues for 3D: Texture



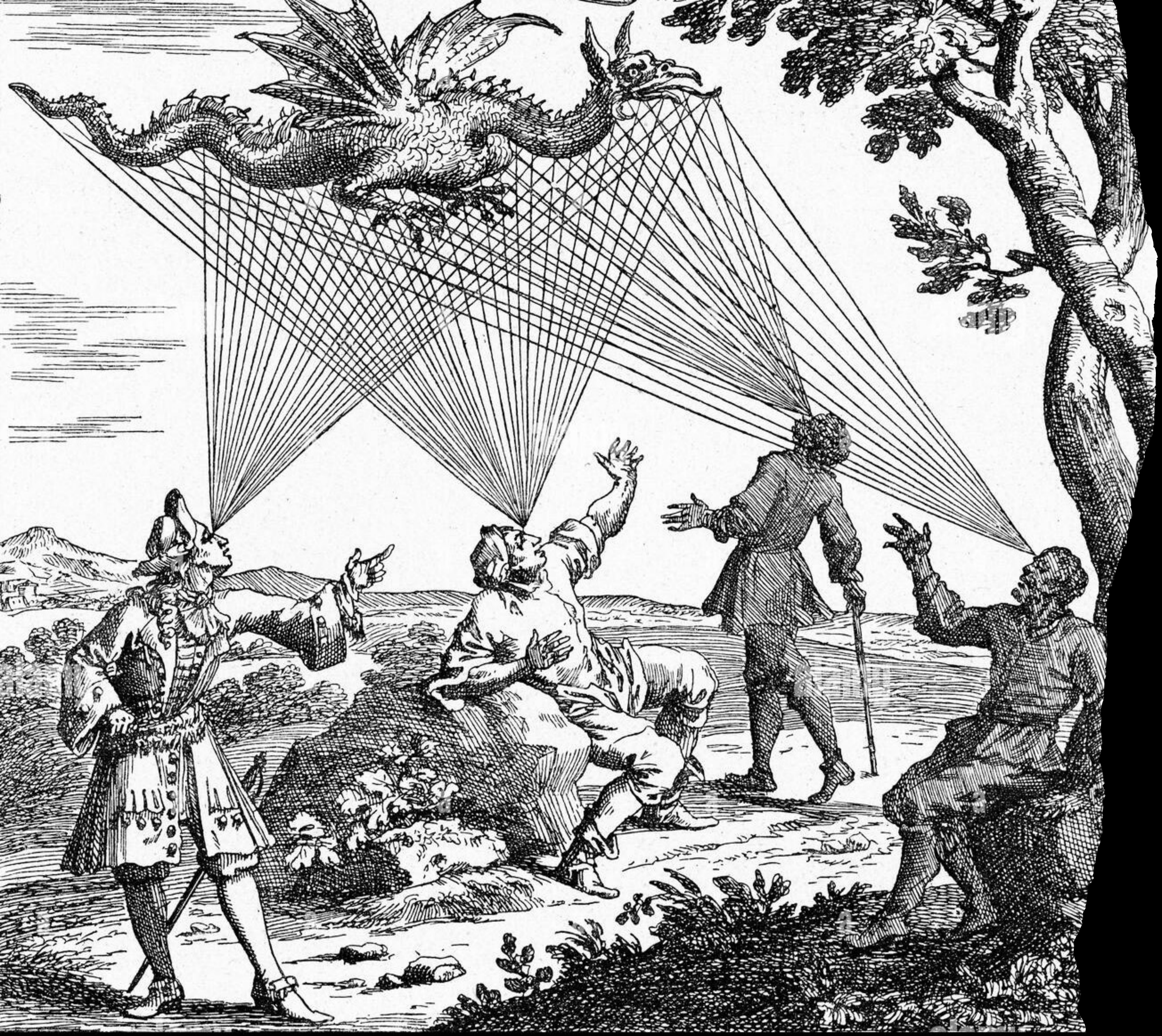
The Visual Cliff by William Vandivert

Visual cues for 3D: Focus, Motion

Image credit: James Hays

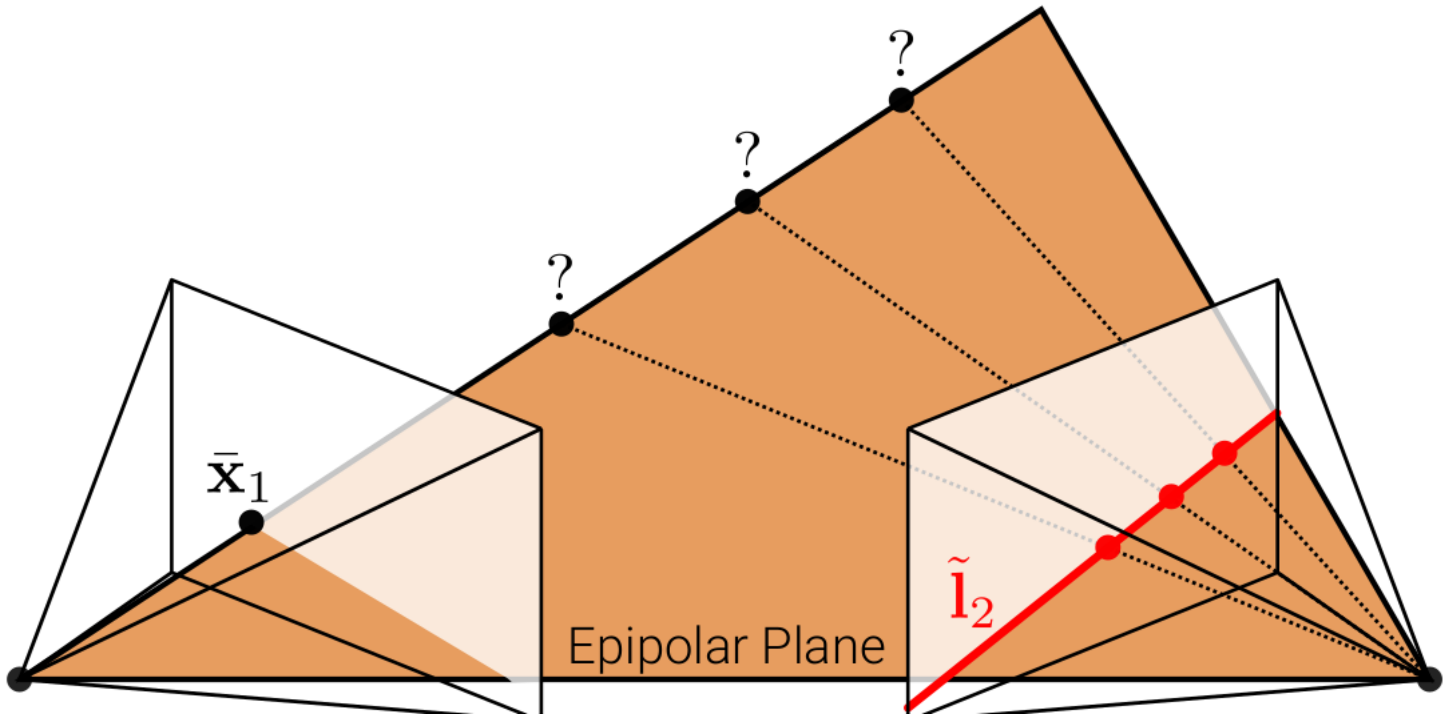
Two-view stereo



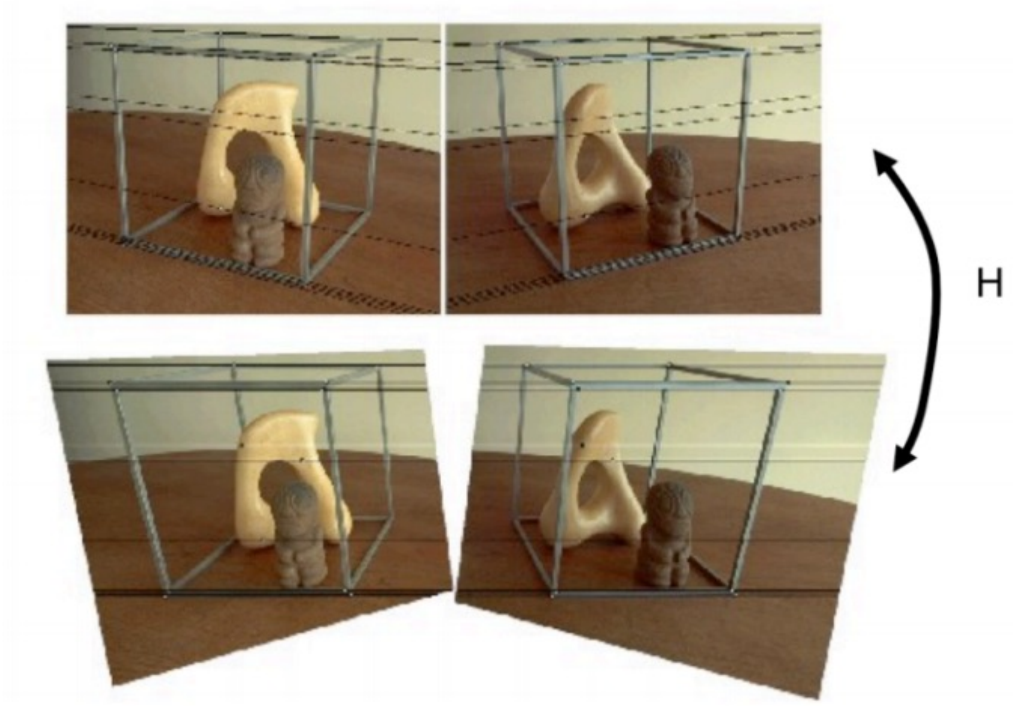
We need at least two observations to estimate the geometry.

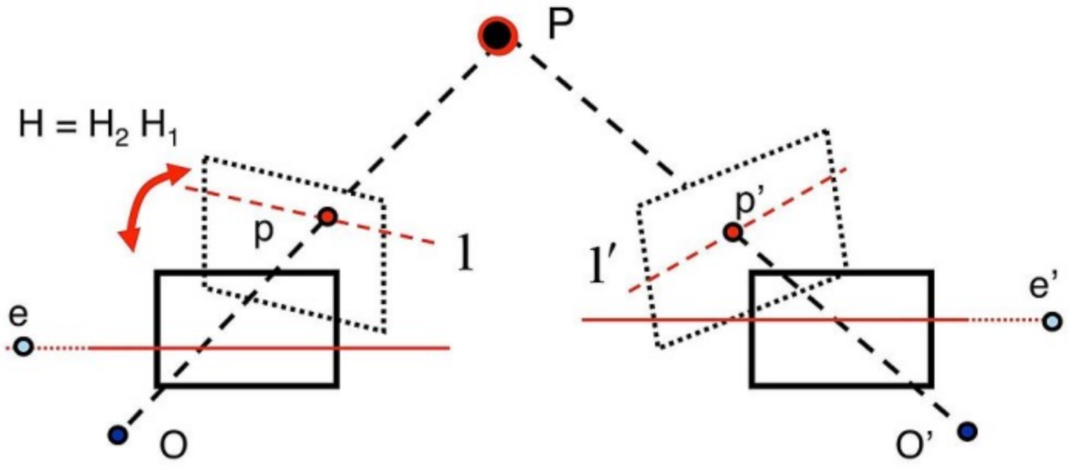
Johann Zahn, 1685

Two-view stereo



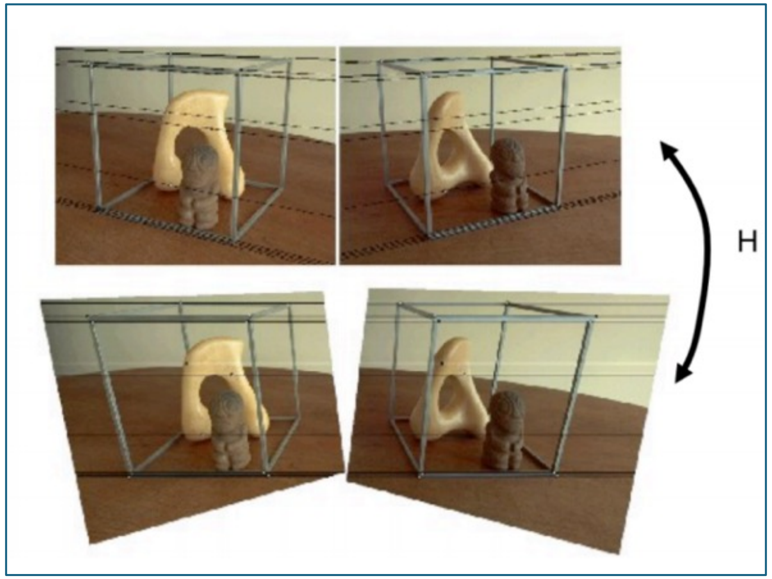
Stereo Rectification

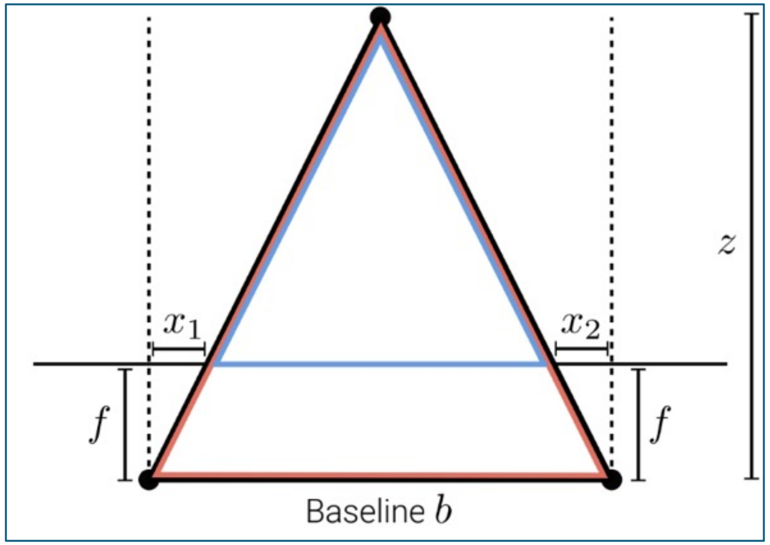


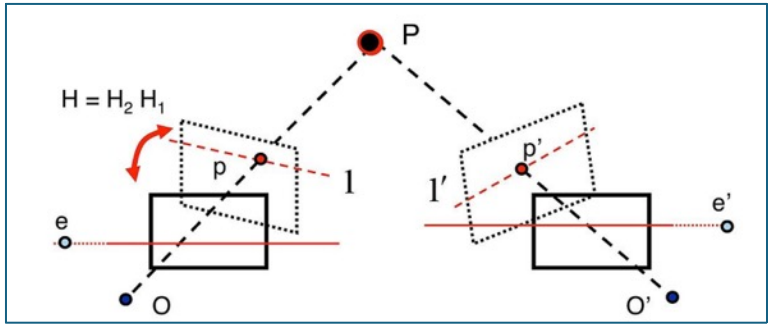


Two-view stereo

Slide credit: Fei-fei Li, Andreas Geiger

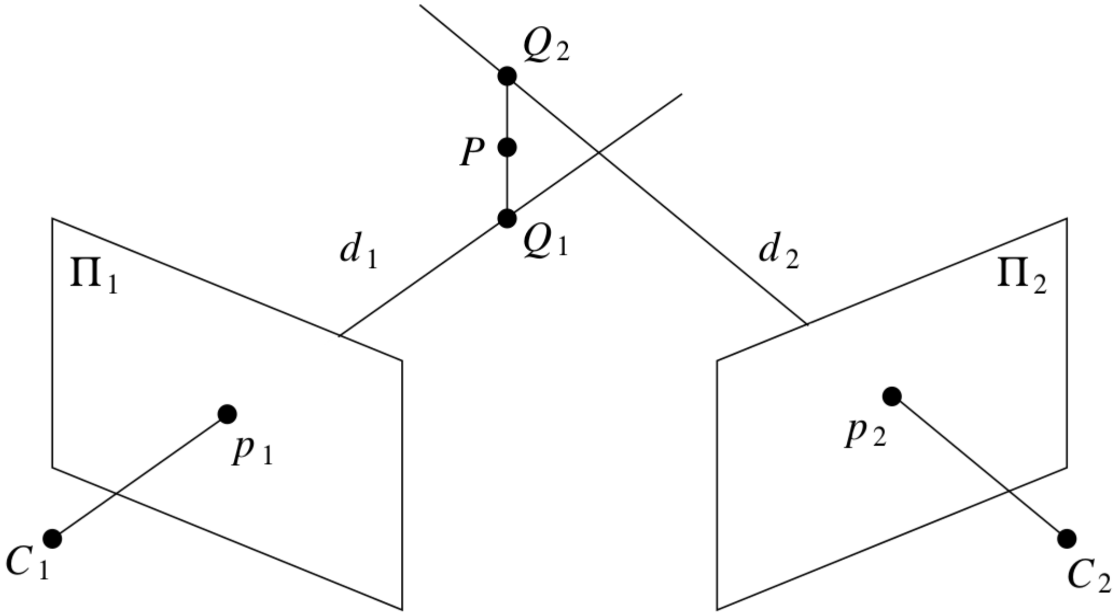


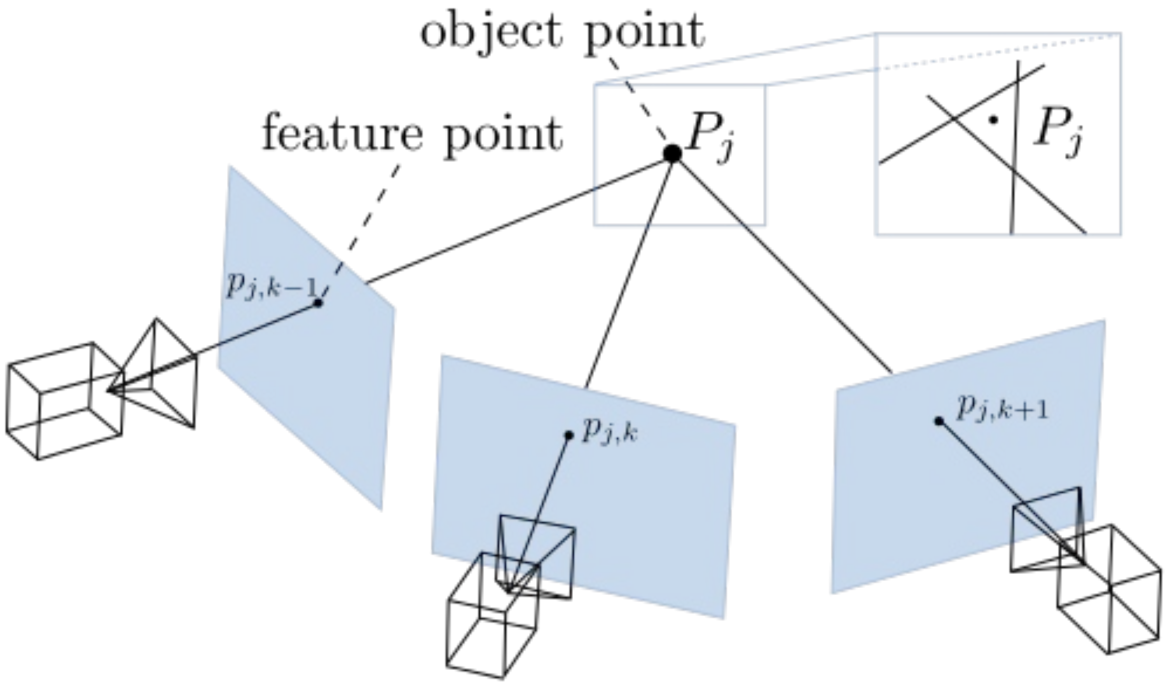




$$z = \frac{b \cdot f}{d}$$

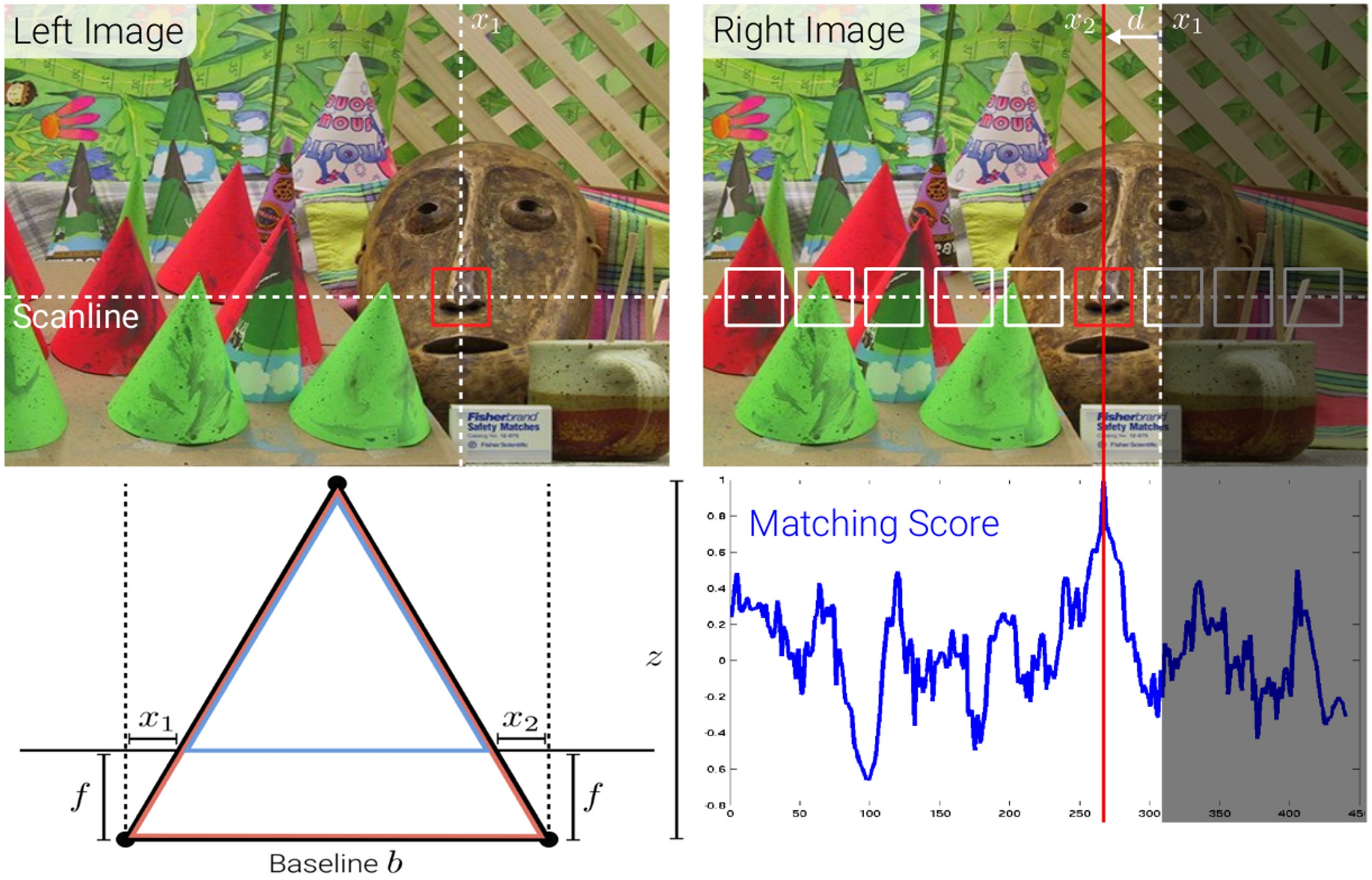
$$depth = \frac{baseline \cdot focal\ length}{disparity}$$





Other triangulation methods

Stereo matching



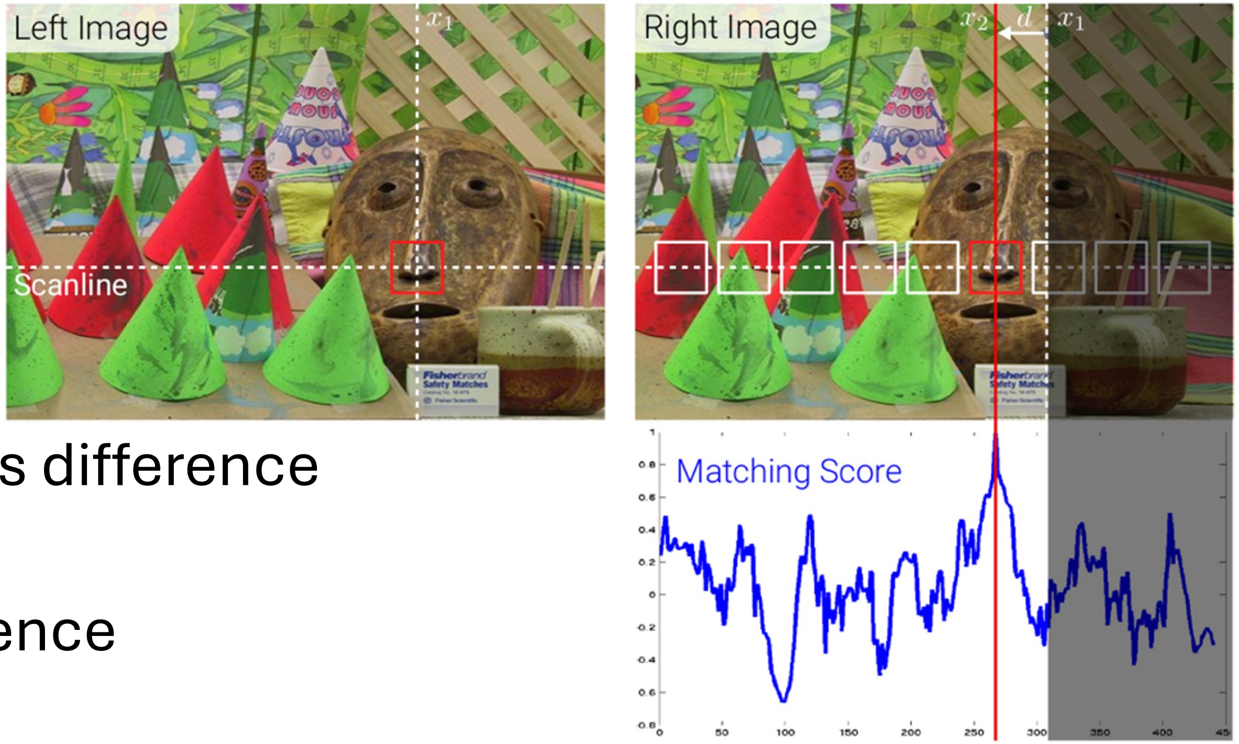
$$disparity = x_1 - x_2$$

$$\frac{baseline}{depth} = \frac{baseline - disparity}{depth - focal\ length}$$

$$depth = \frac{baseline \cdot focal\ length}{disparity}$$

Slide credit: Andreas Geiger

Block matching

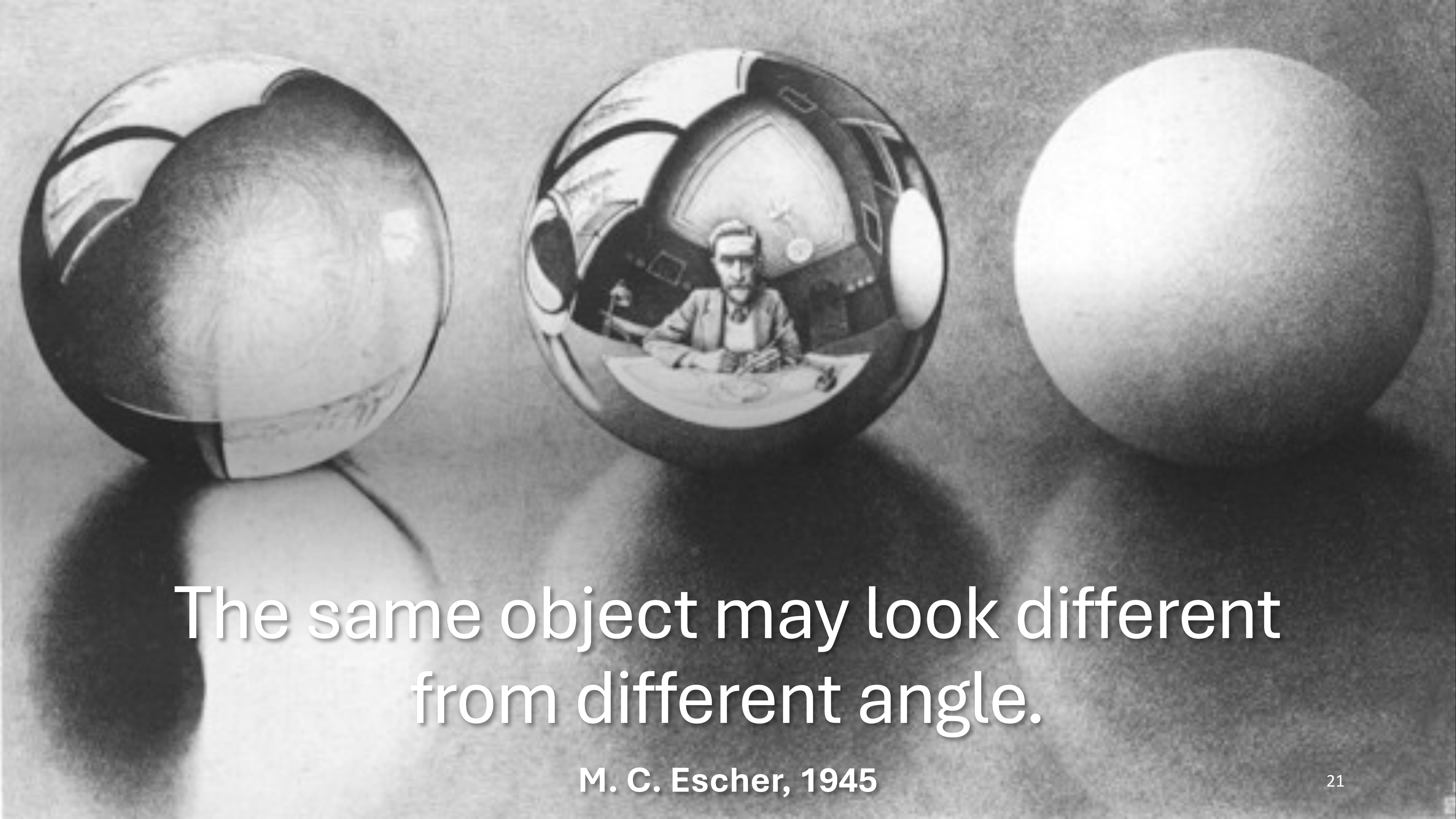


$SSD = \sum\sum\left(I_{left} - I_{right}\right)^2$ → Sum of squares difference

$\text{AD} = \sum\sum\left|\left(I_{left} - I_{right}\right)\right|$ → Absolute difference

$CC = \sum\sum\left(I_{left} \cdot I_{right}\right)$ → Cross correlation

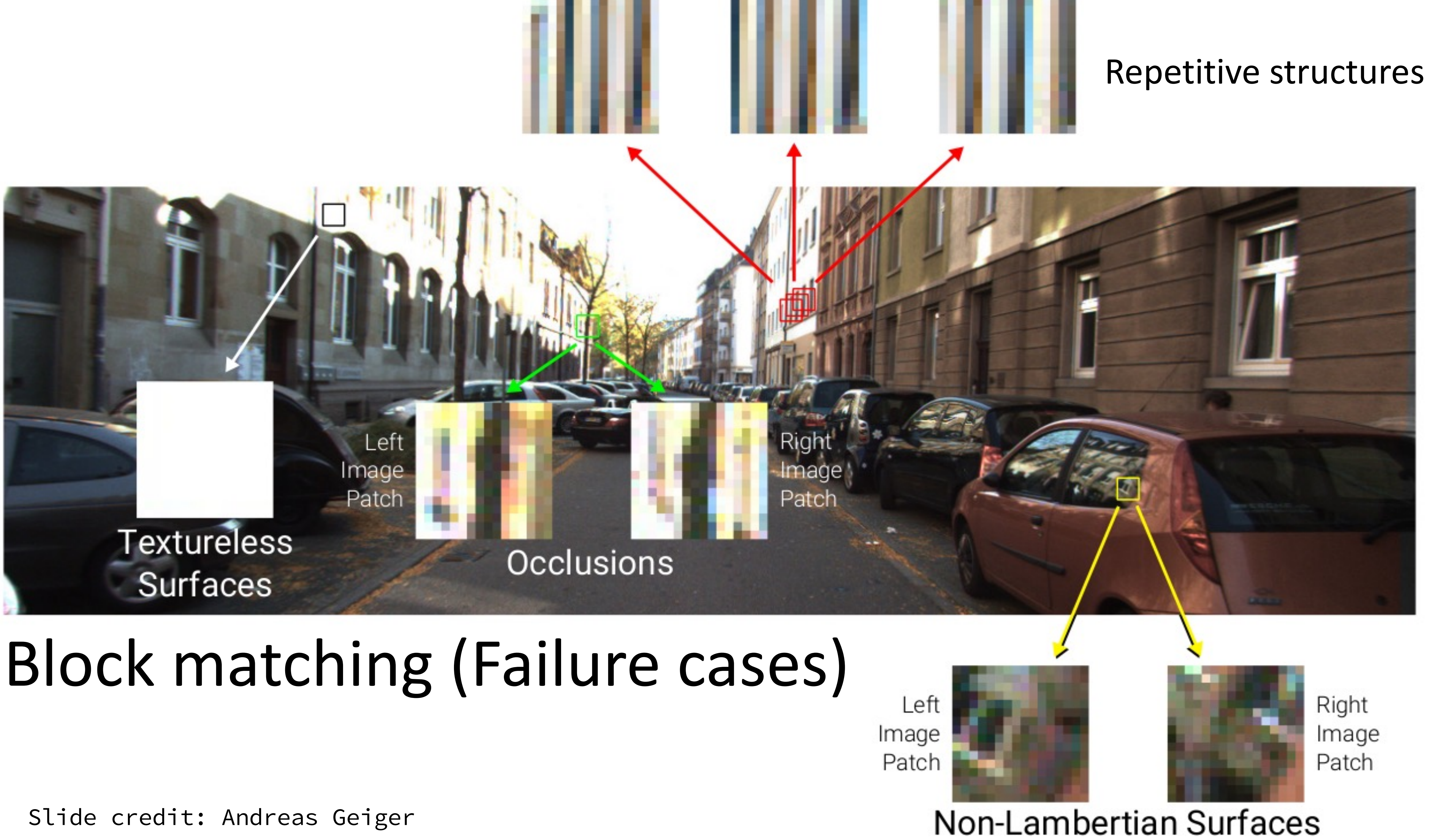
$$NCC = \frac{\sum\sum\left(I_{left} \cdot I_{right}\right)}{\sqrt{\sum\sum\left(I_{left} \cdot I_{left}\right)} \cdot \sqrt{\sum\sum\left(I_{right} \cdot I_{right}\right)}}$$ → Normalized cross correlation



The same object may look different from different angle.

M. C. Escher, 1945

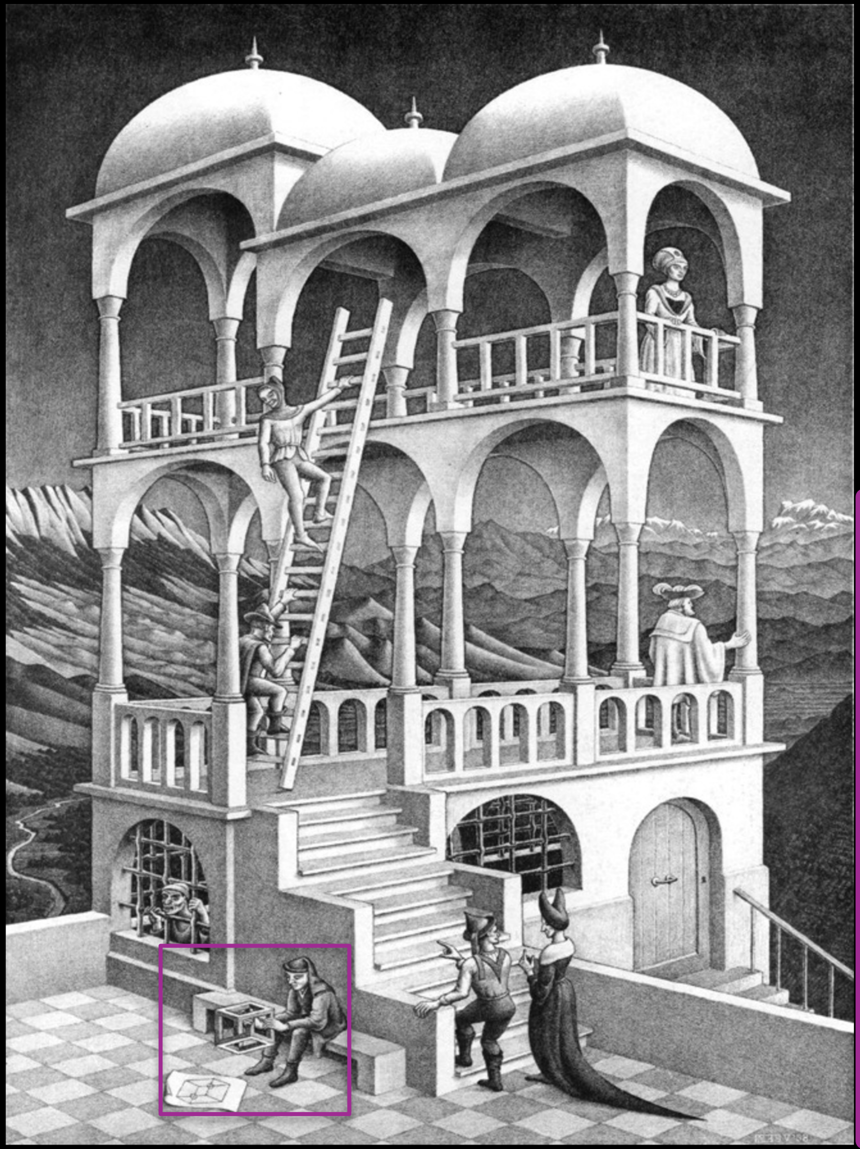
Block matching (Failure cases)



Slide credit: Andreas Geiger

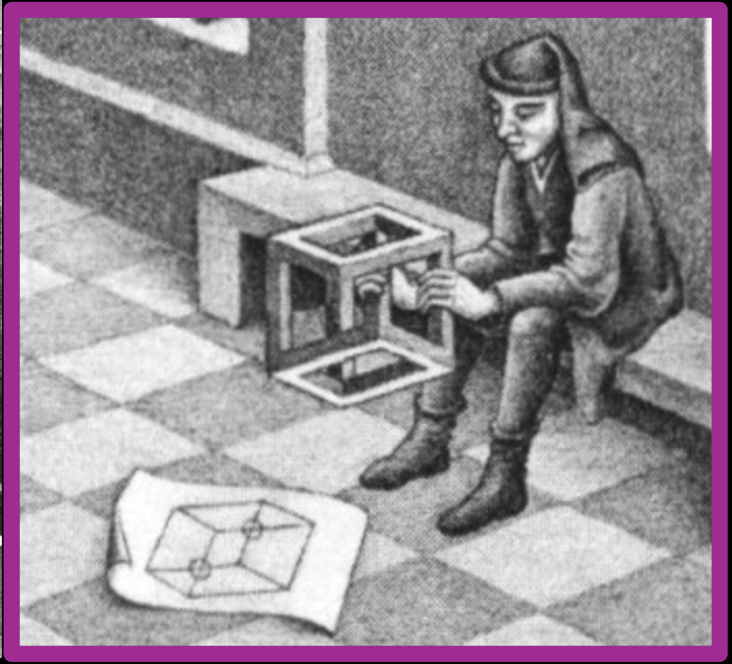
Other challenges:

- Repetitive structures
- Lighting variations
- Vignetting effects
- Motion blur
- Sensor noise
- Color inbalance
- White inbalance
- etc.

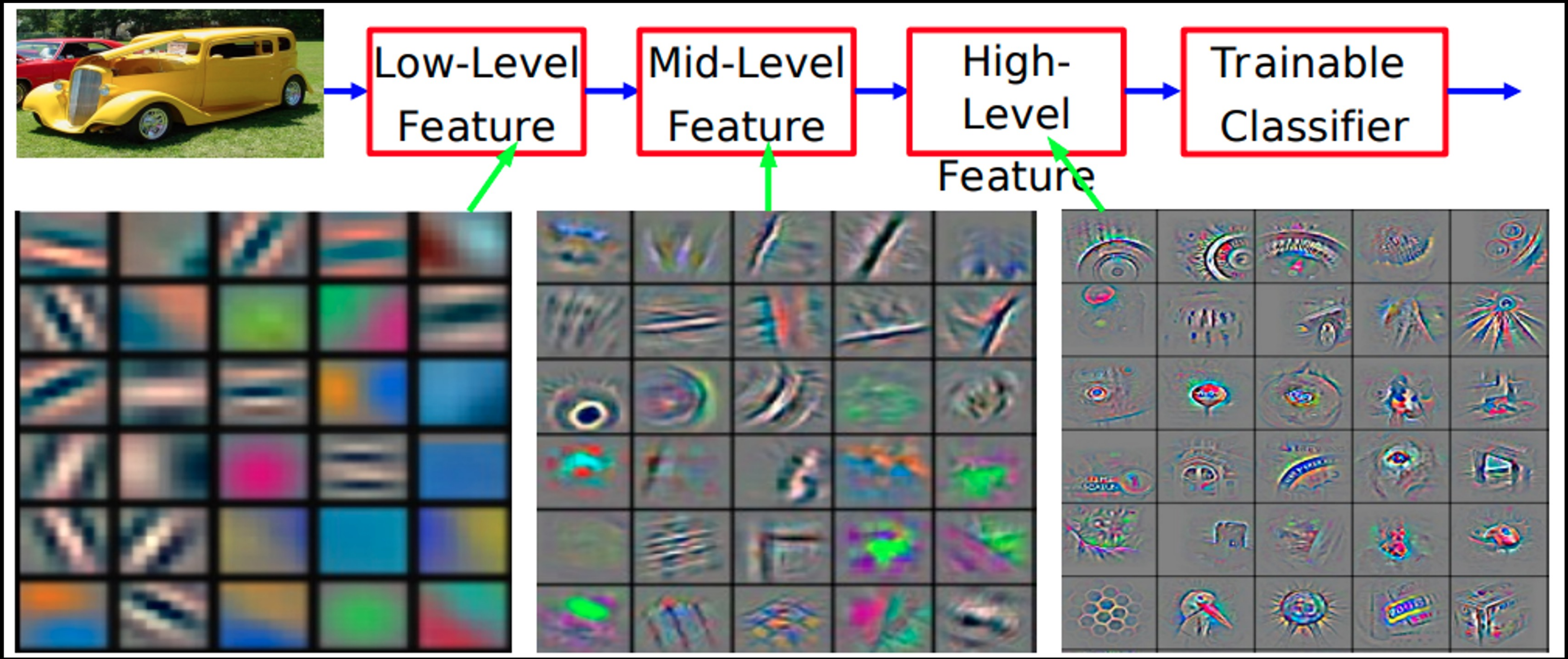


M. C. Escher, 1958

A PhD trying to use block matching

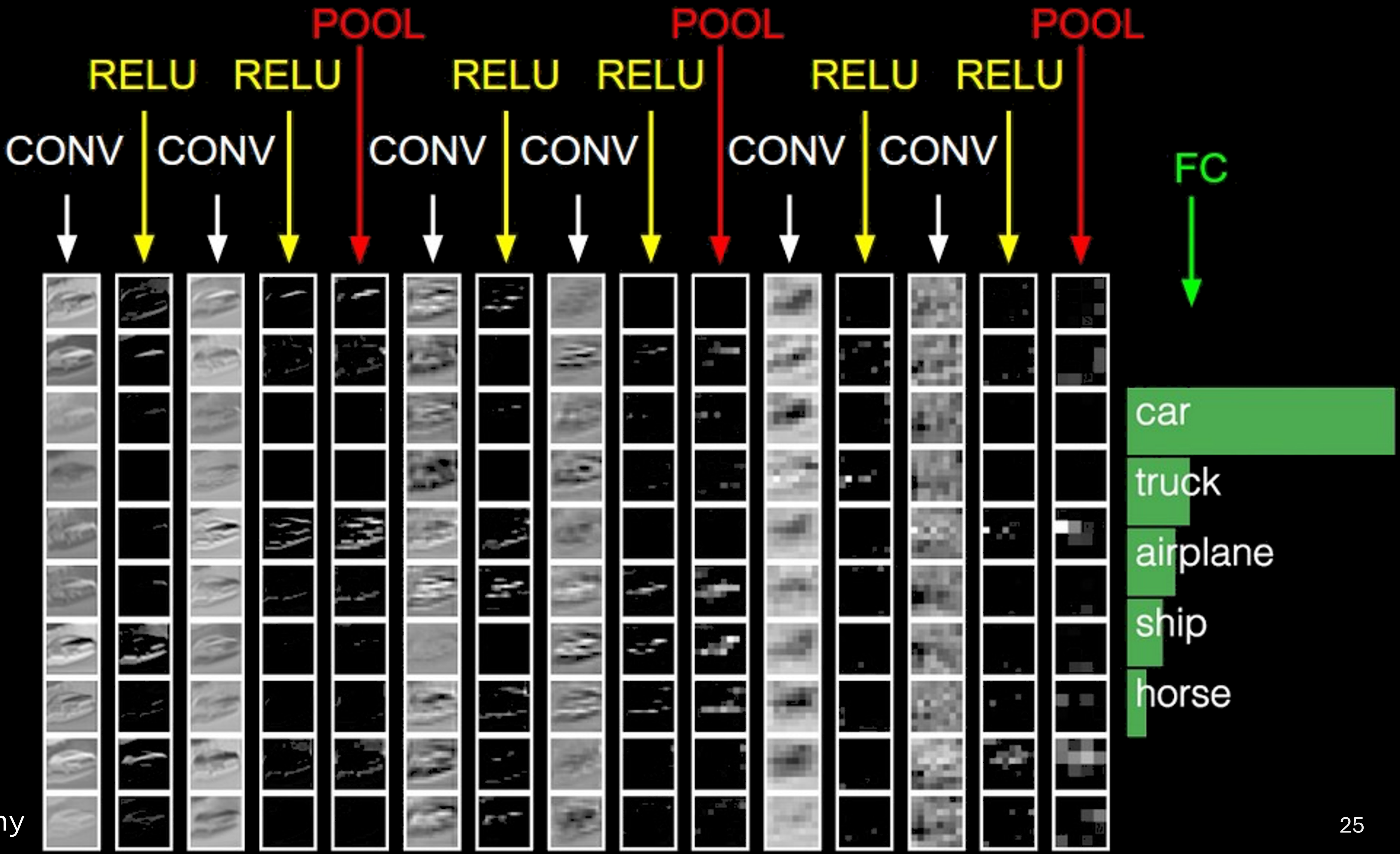


Convolutional features



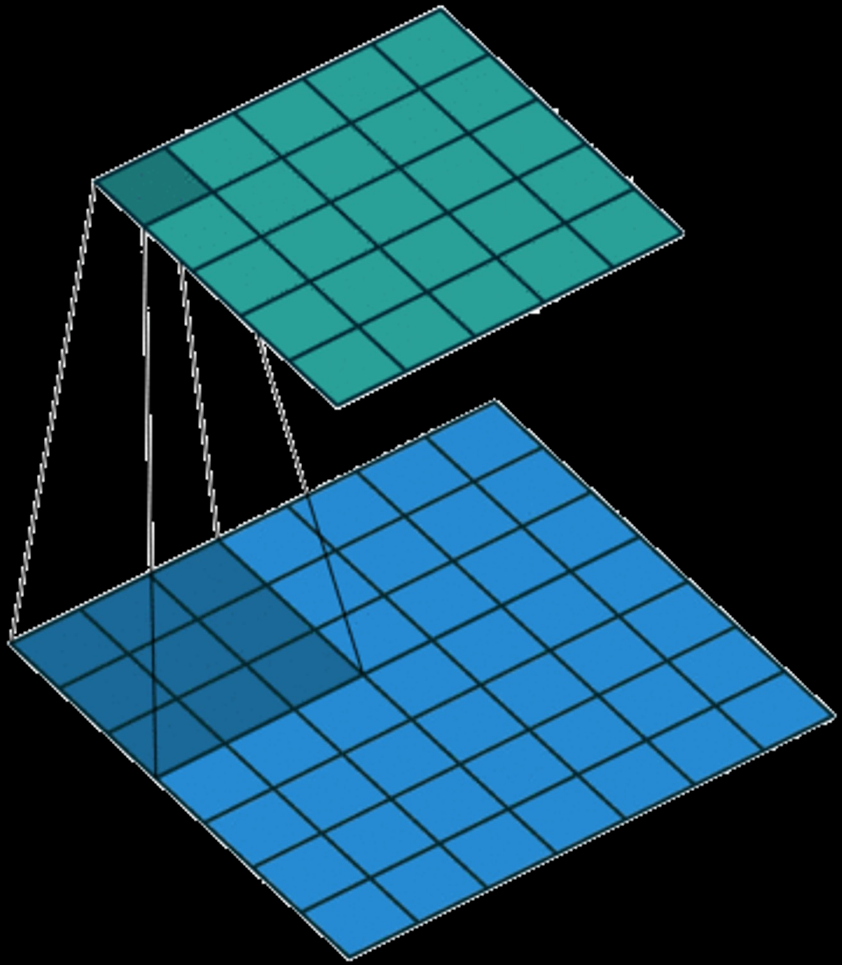
Slide credit: Yann Lecun
Image credit: Visualizing and Understanding Convolutional Networks (Zeiler & Fergus, 2013)

Convolutional network architecture



Slide credit: Andrej Karpathy

2D and 3D convolutions



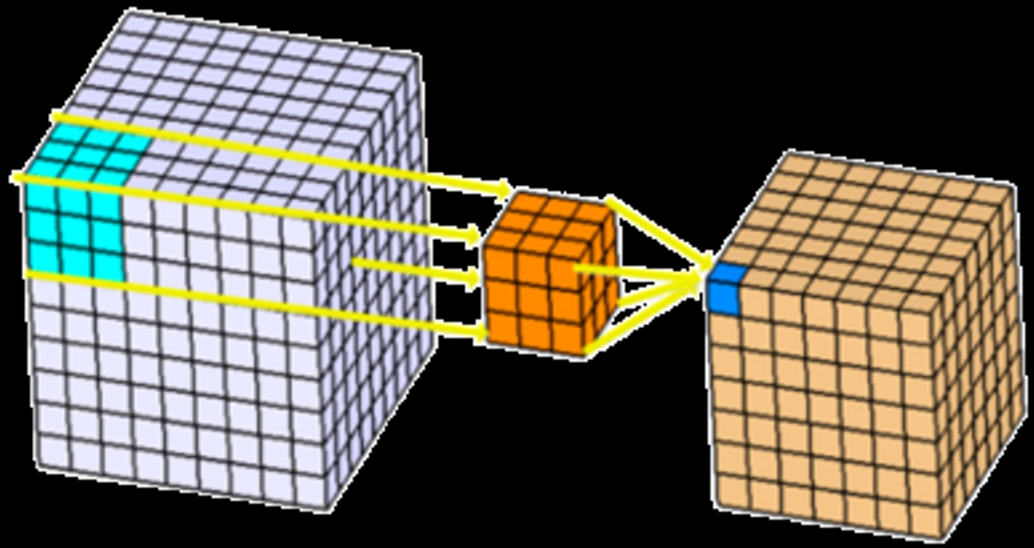
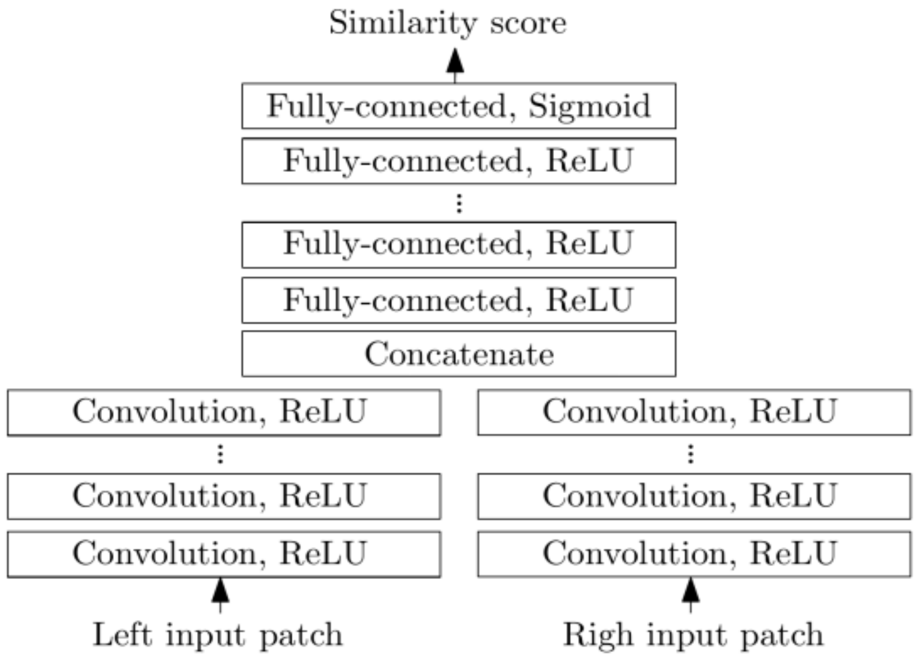


Image credit:
https://biplabbarman097.medium.com/3d-convolutions-and-its-applications-6dd2d0e9e63f

Block matching using deep learning

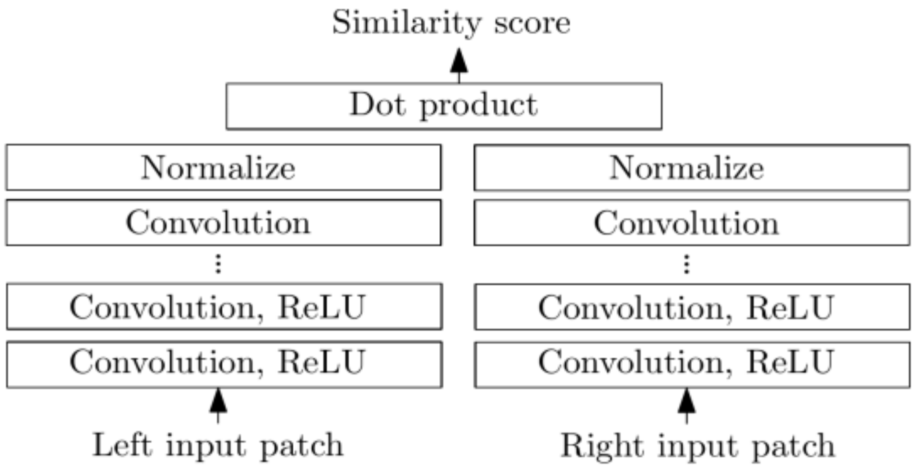
Learned Similarity:

- Learn features & sim. metric
- Potentially more expressive
- Slow (WxHxD MLP evaluations)



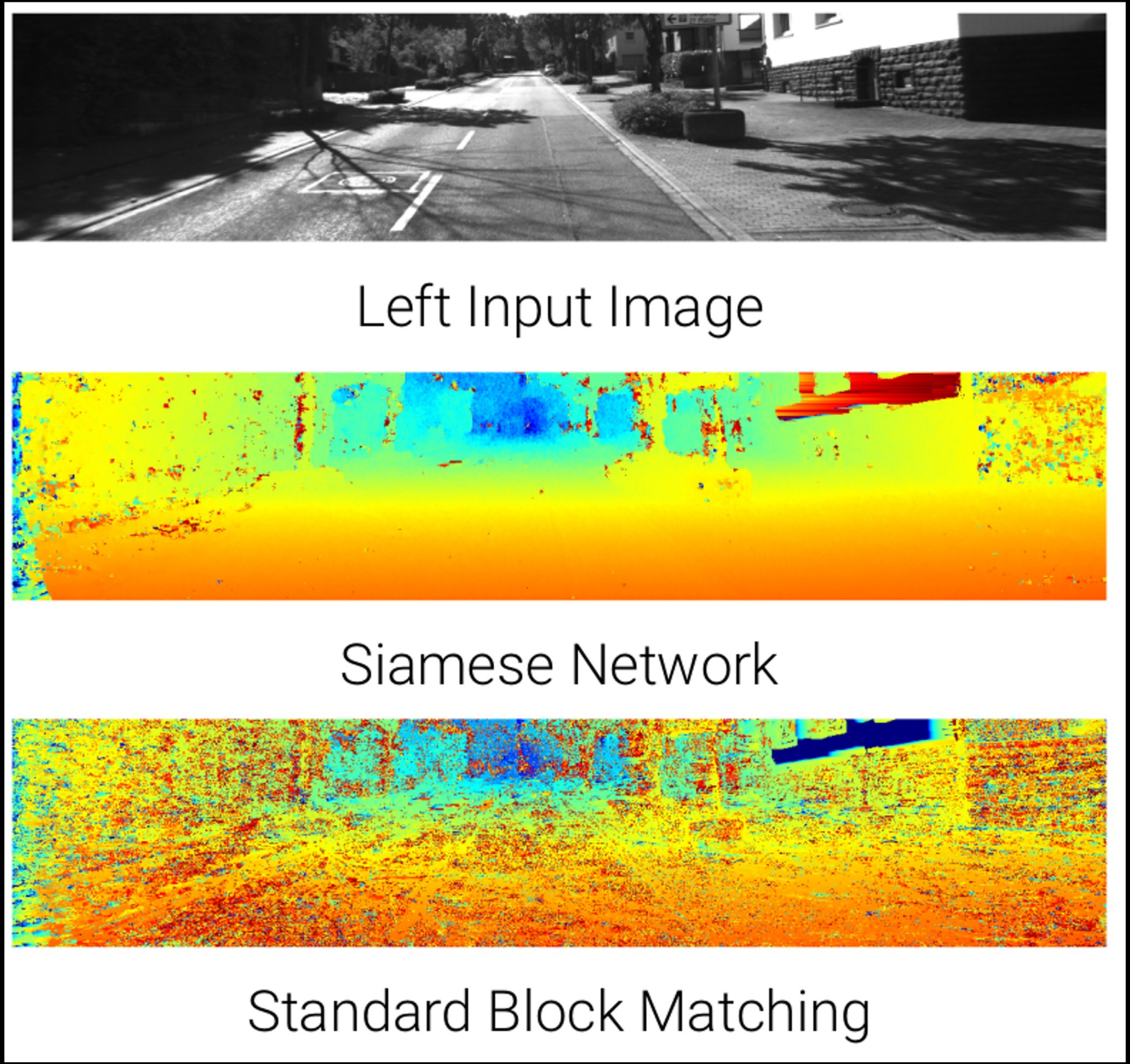
Cosine Similarity:

- Learn features & apply dot-product
- Features must do the heavy lifting
- Fast matching (no network eval.)



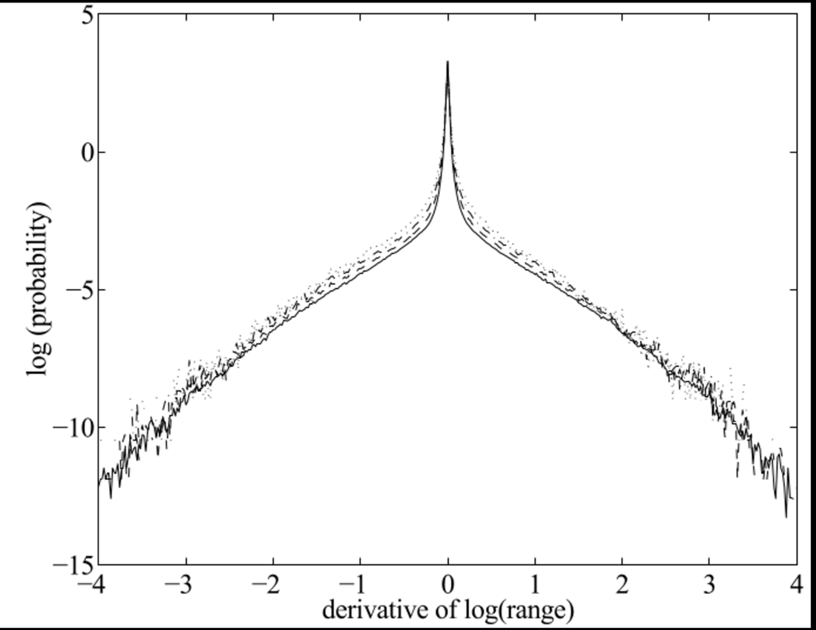
Zbontar and LeCun: Stereo Matching by Training a Convolutional Neural Network to Compare Image Patches. JMLR, 2016.

Block matching



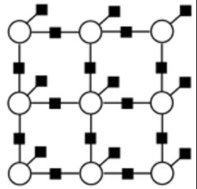
Zbontar and LeCun: Stereo Matching by Training a Convolutional Neural Network to Compare Image Patches. JMLR, 2016.

Block matching



Huang, Lee and Mumford: Statistics of Range Images. CVPR, 2000.

$$p(\mathbf{D}) \propto \exp\left\{ -\sum_i \psi_{data}(d_i) - \lambda \sum_{i\sim j} \psi_{smooth}(d_i, d_j) \right\}$$

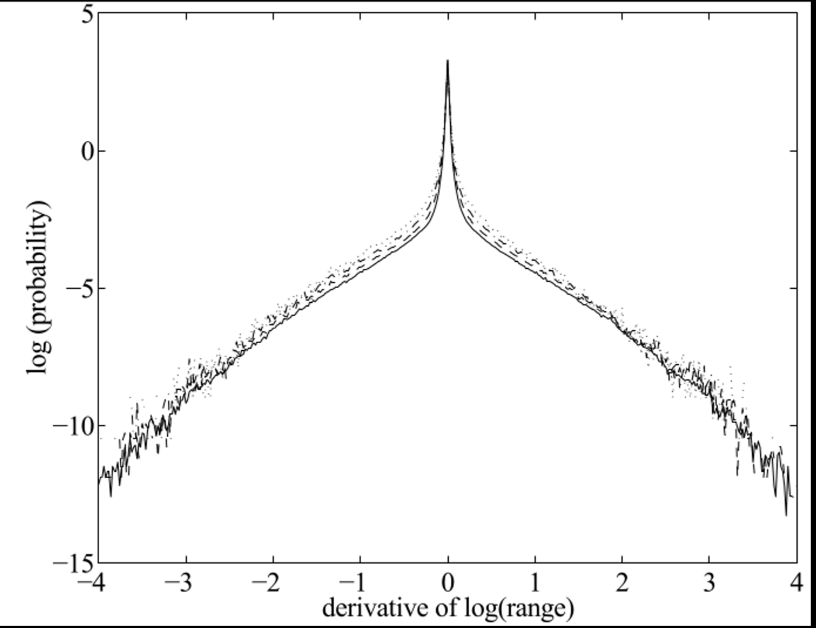


Y. Boykov, O. Veksler, and R. Zabih, “Fast approximate energy minimization via graph cuts”. PAMI(1999)

Zbontar and LeCun: Stereo Matching by Training a Convolutional Neural Network to Compare Image Patches. JMLR, 2016.

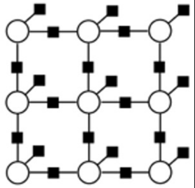
Semi-Global Matching Algorithm

Block matching

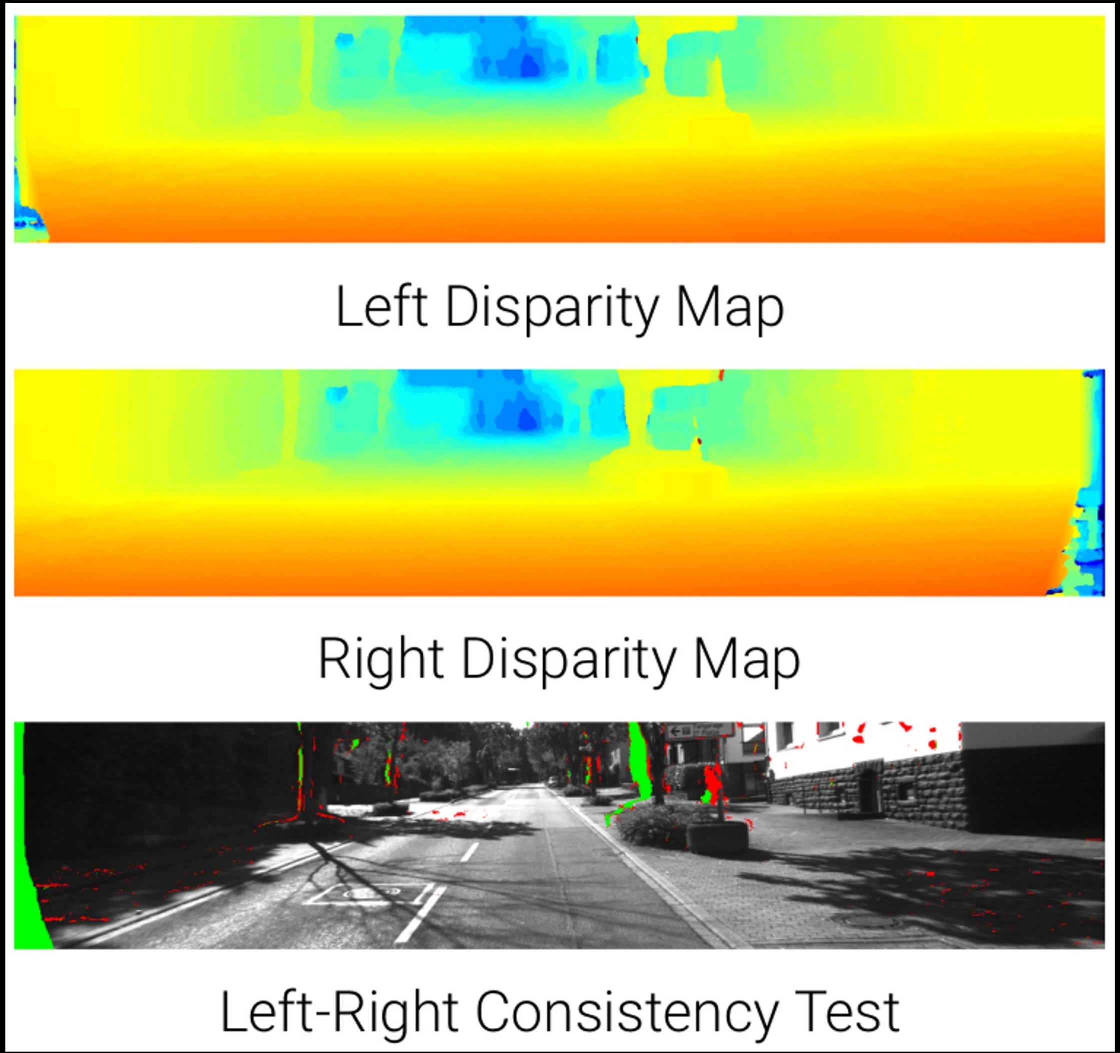


Huang, Lee and Mumford: Statistics of Range Images. CVPR, 2000.

$$p(\mathbf{D}) \propto \exp\left\{-\sum_i \psi_{data}(d_i) - \lambda \sum_{i\sim j} \psi_{smooth}(d_i, d_j)\right\}$$

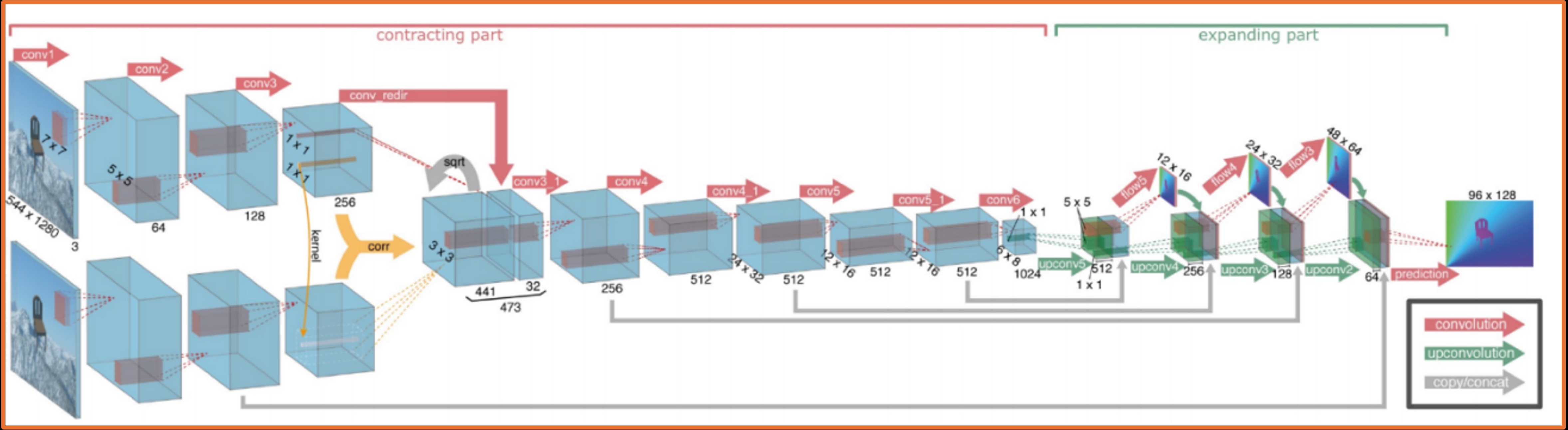


Y. Boykov, O. Veksler, and R. Zabih, "Fast approximate energy minimization via graph cuts". PAMI(1999)



Zbontar and LeCun: Stereo Matching by Training a Convolutional Neural Network to Compare Image Patches. JMLR, 2016.

DISPNET

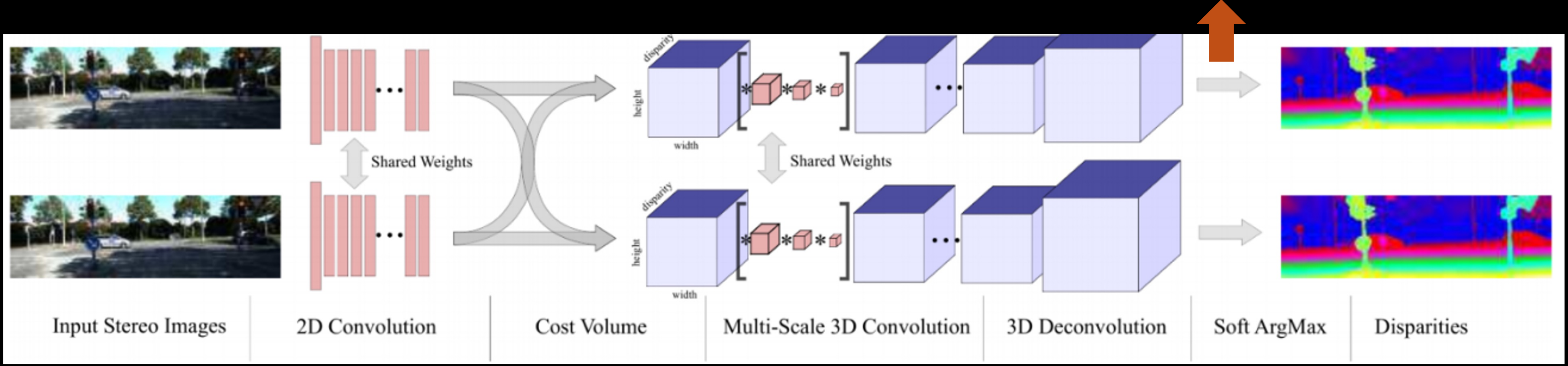


- DispNet was one of the first end-to-end trained deep neural network for stereo disparity
- It used a U-Net like architecture with skip-Connections to retain details
- It introduces correlation layer
- Multi-scale loss (disparity error in pixels), curriculum learning (easy-to-hard)

Mayer et al.: A Large Dataset to Train Convolutional Networks for Disparity, Optical Flow, and Scene Flow Estimation. CVPR, 2016.

GC-net

$$d^* = \mathbb{E}[d] = \sum_{d=0}^{D} \underbrace{\mathrm{softmax}(-c_\theta(d))}_{p(d)} \cdot d$$



- Key idea: calculate disparity cost volume and apply 3D convolutions on it
- Convert the learned matching cost c to disparity via the expectation(probability volume)
- Slightly better performance but large memory requirements (3D feature volume)

Kendall, Martirosyan, Dasgupta and Henry: End-to-End Learning of Geometry and Context for Deep Stereo Regression. ICCV, 2017.

Multi-view stereo

- MVS Goal: To find a 3D shape that explains the images.

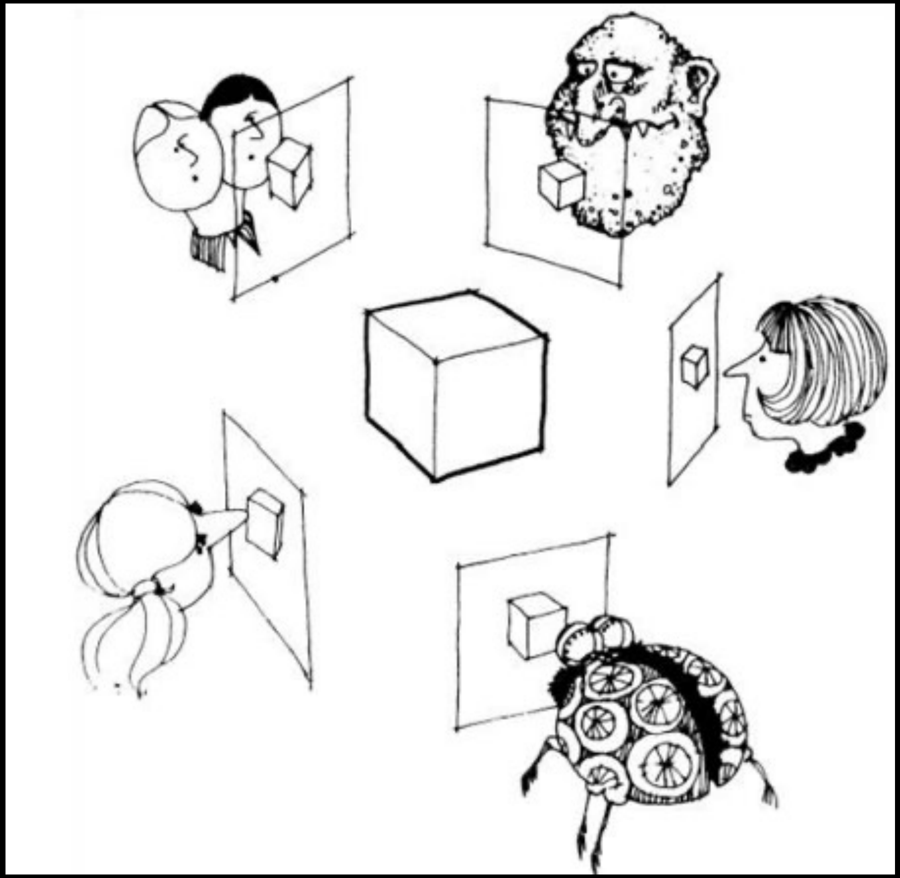


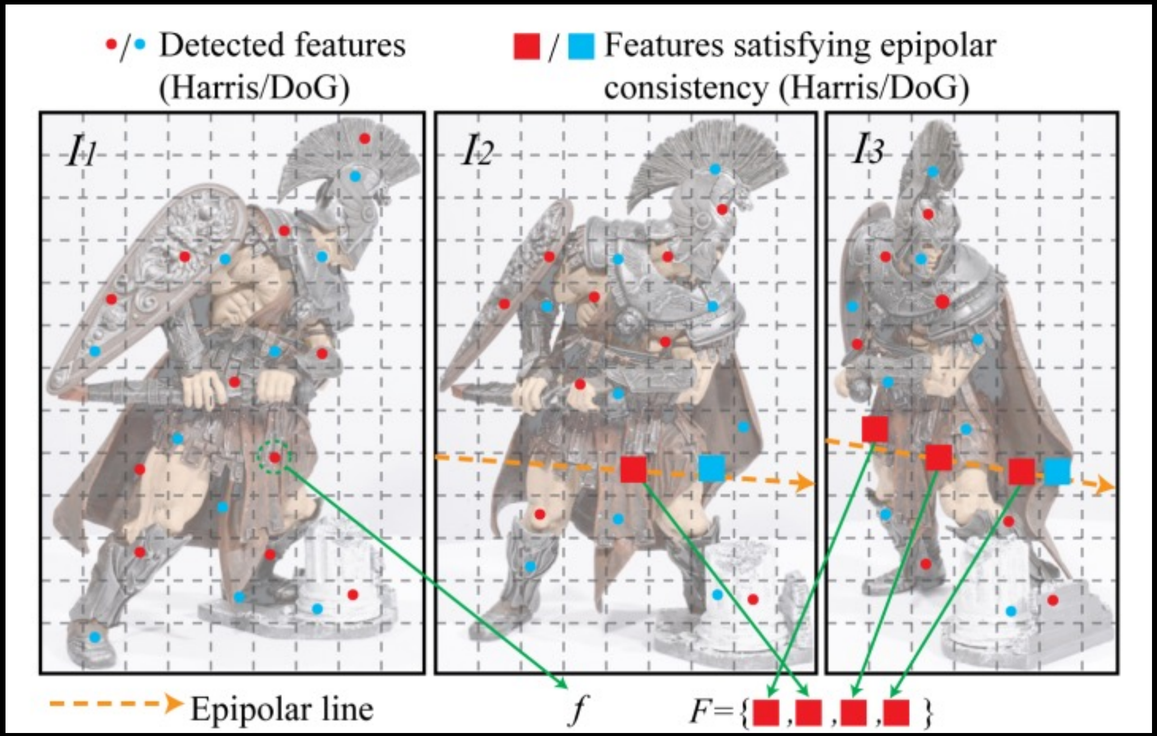
Image credit: Svetlana Lazebnik

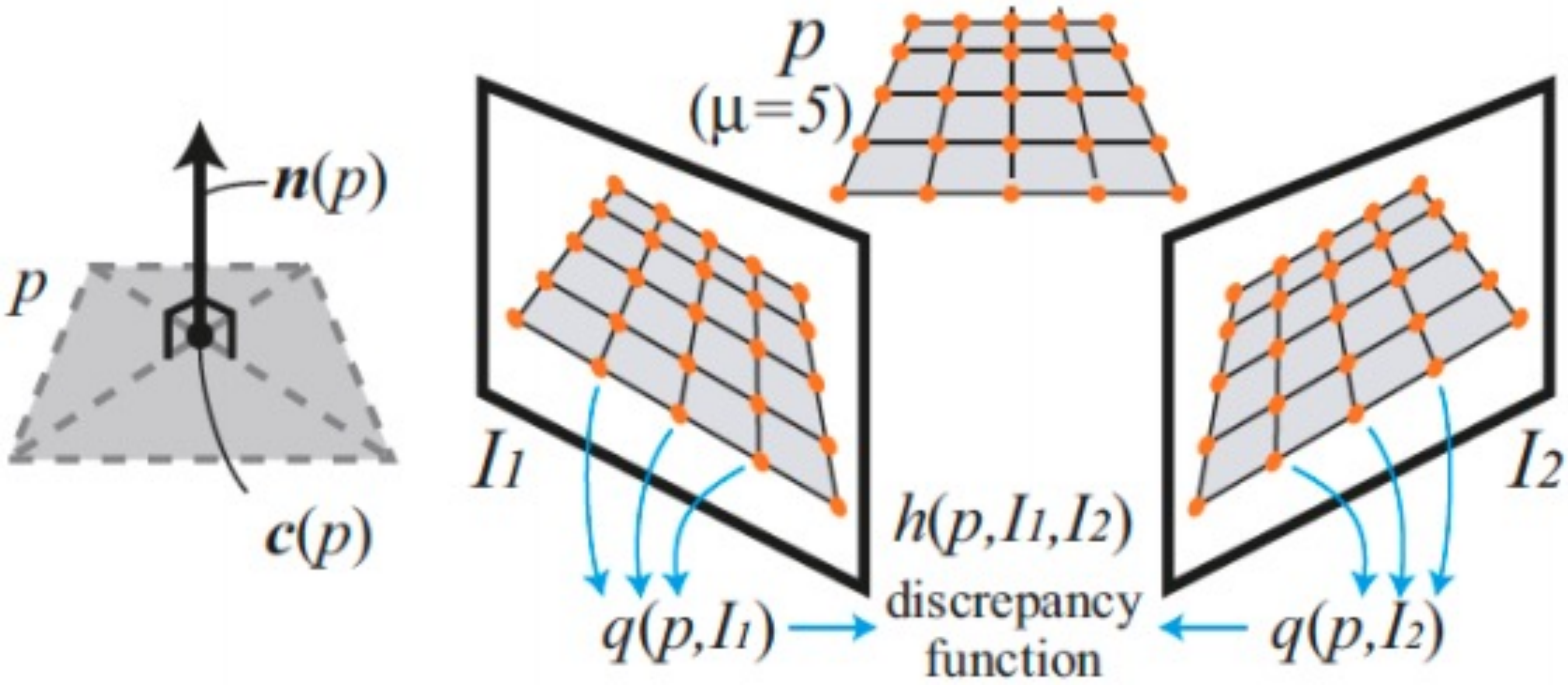
PMVS in 1 slide

Detect	• Detect keypoints
Triangulate	• Triangulate a sparse set of initial matches
Expand	• Iteratively expand matches to nearby locations
Filter	• Use visibility constraints to filter out false matches
Perform	• Perform surface reconstruction

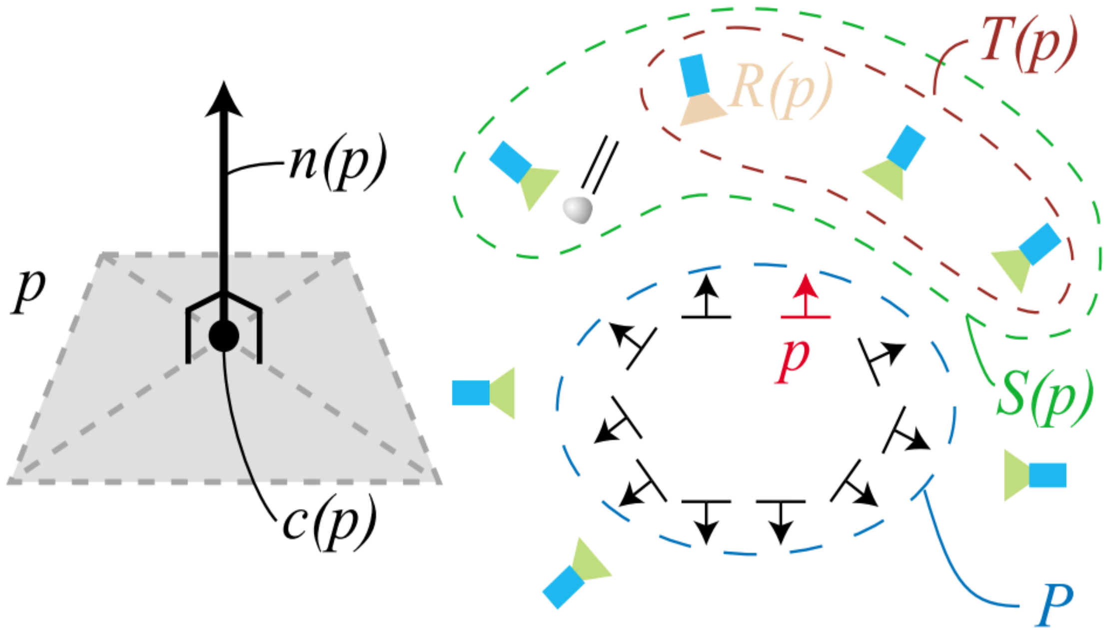
Feature Detection

1. Divide grid to cells (32x32)
2. Use Harris Detector and DoG to find corners
3. Try to find 4 good corners in each cell (uniform overage)





Patch Geometry



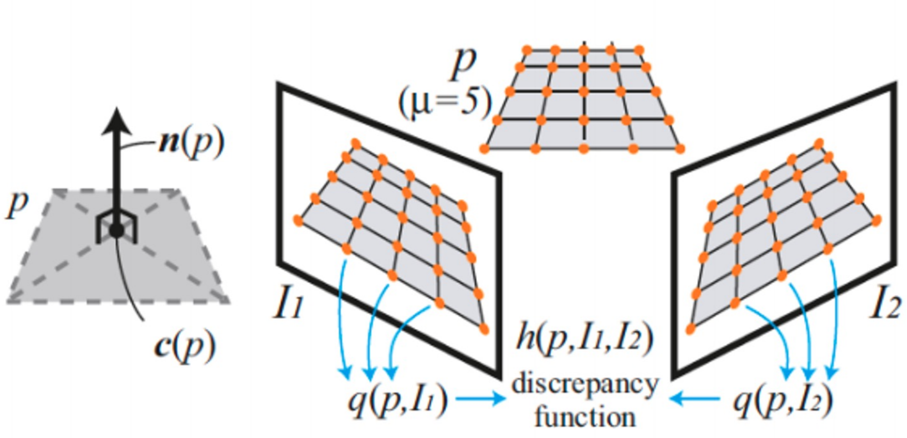
Patch Model Initialization

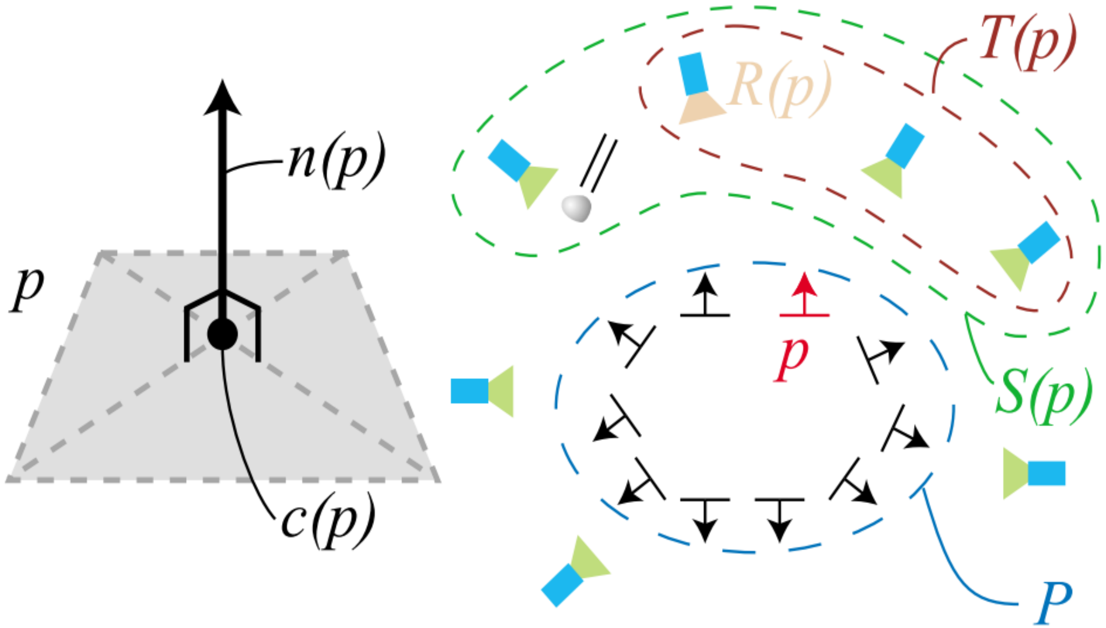
Patch Initialization

$c(p) \leftarrow$ Triangulation from from two patches

$n(p) \leftarrow c(p)O(I_i)/|c(p)O(I_i)|$ normal initialization

$R(p) \leftarrow I_i$ reference image of p





Patch Discrepancy

Patch Discrepancy

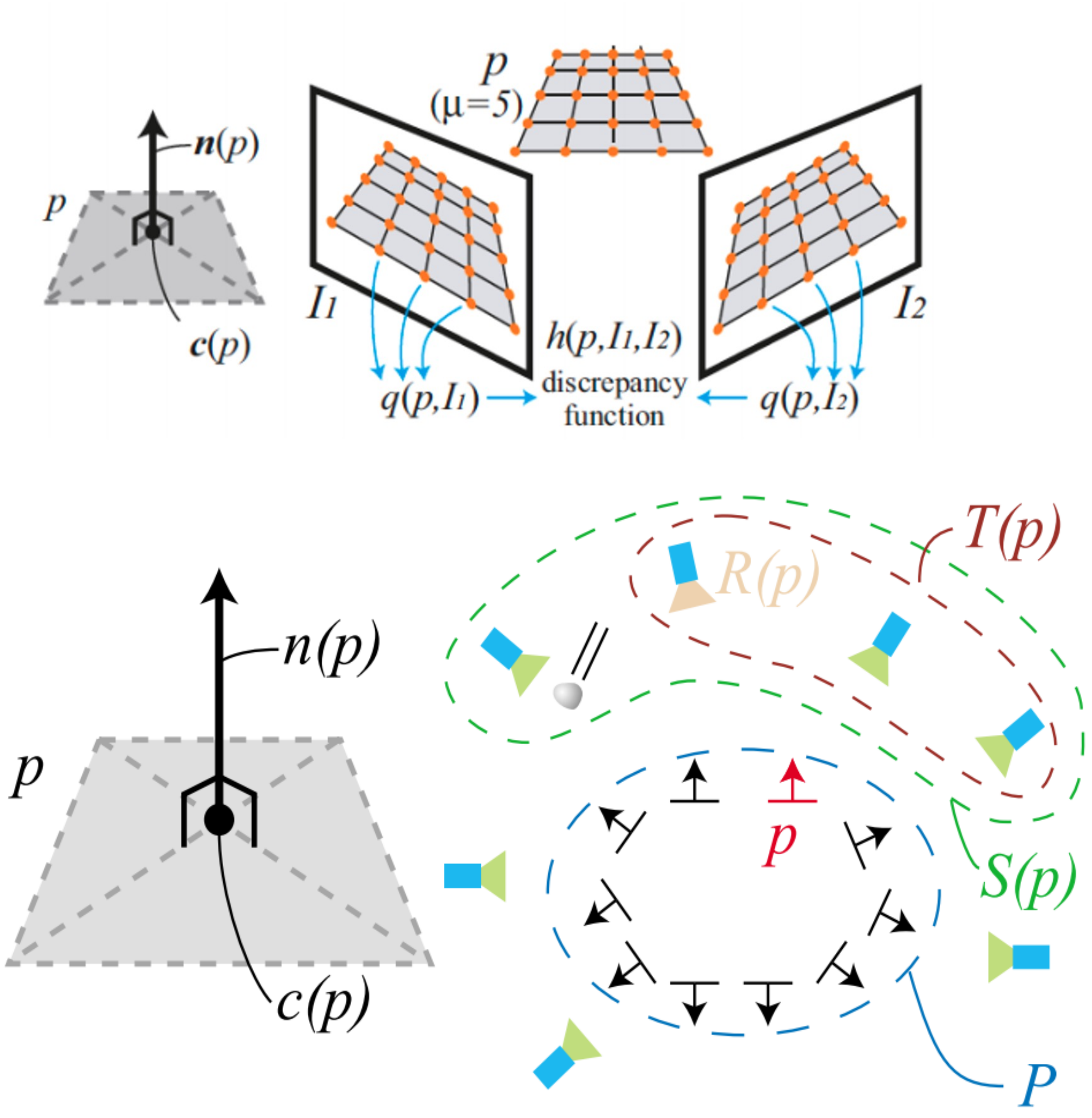
$$h(p,\ I,\ R(p)) = 1\ - NCC(p,\ I,\ R(p))$$

discrepancy function

$$g(p) = \frac{1}{|S(p) \setminus\ R(p)|}\sum_{I\in S(p)\ \setminus\ R(p)}\ h(p,I,R(p\ ))$$

Objective to minimize

$S(p) \leftarrow$ the set of images patch may seem



Patch True Discrepancy

Patch True Discrepancy

$$T(p) = \{I \mid I \in S(p), h\big(p, I, R(p)\big) \leq \tau\}$$

$$g^*(p) = \frac{1}{|T(p) \setminus R(p)|} \Sigma_{I \in T(p) \setminus R(p)} \; h(p, I, R(p\,))$$

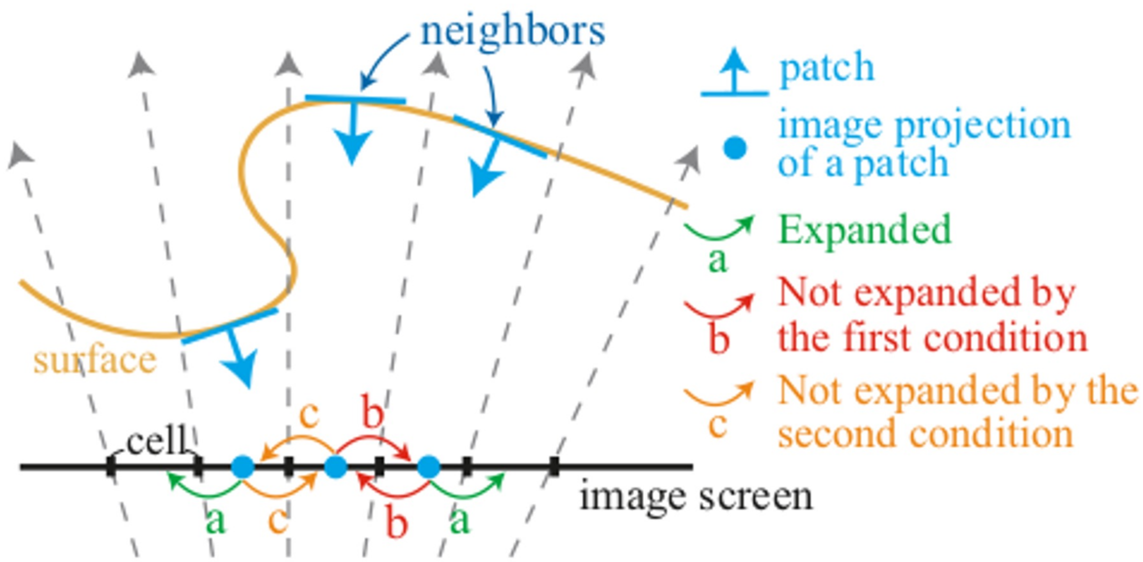
True objective to minimize

$$argmin_{n(c),c(p)} \; g^* \,(p)$$

$S(p) \leftarrow$ the set of images patch may seem

$\mathrm{T}(p) \leftarrow$ the set of images patch truly seem

$n(p), c(p) \leftarrow$ find normal and center of patch that minimizes objective



Expansion and Filtering

Expansion

1. Identify neighbouring cells for possible expansion
2. Test if there is already a patch very close to that region
3. Test for depth discontinuity

Filtering

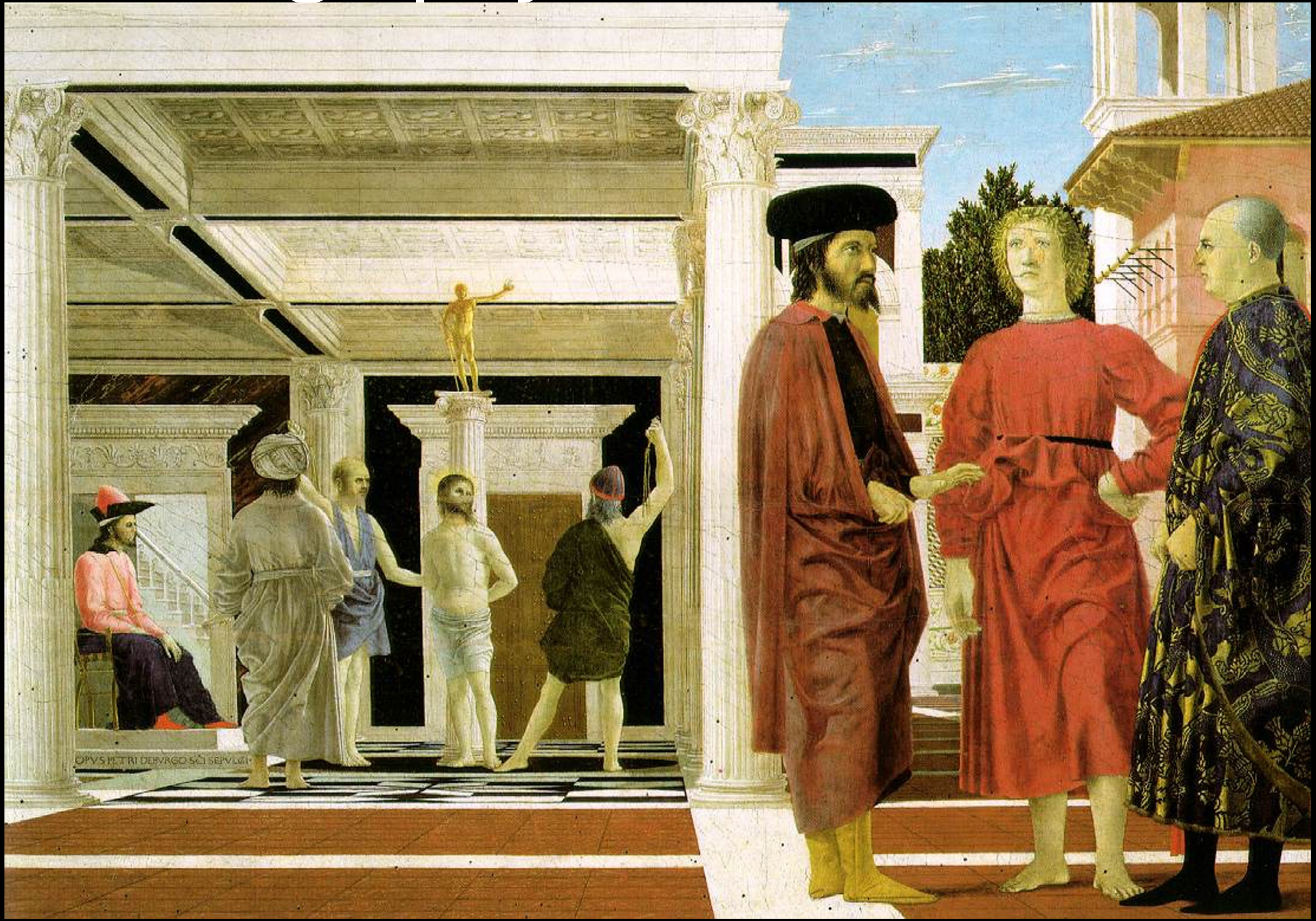
1. Photometric consistency filter
2. Geometric consistency filter
3. Occlusion check

VisualSFM+PMVS

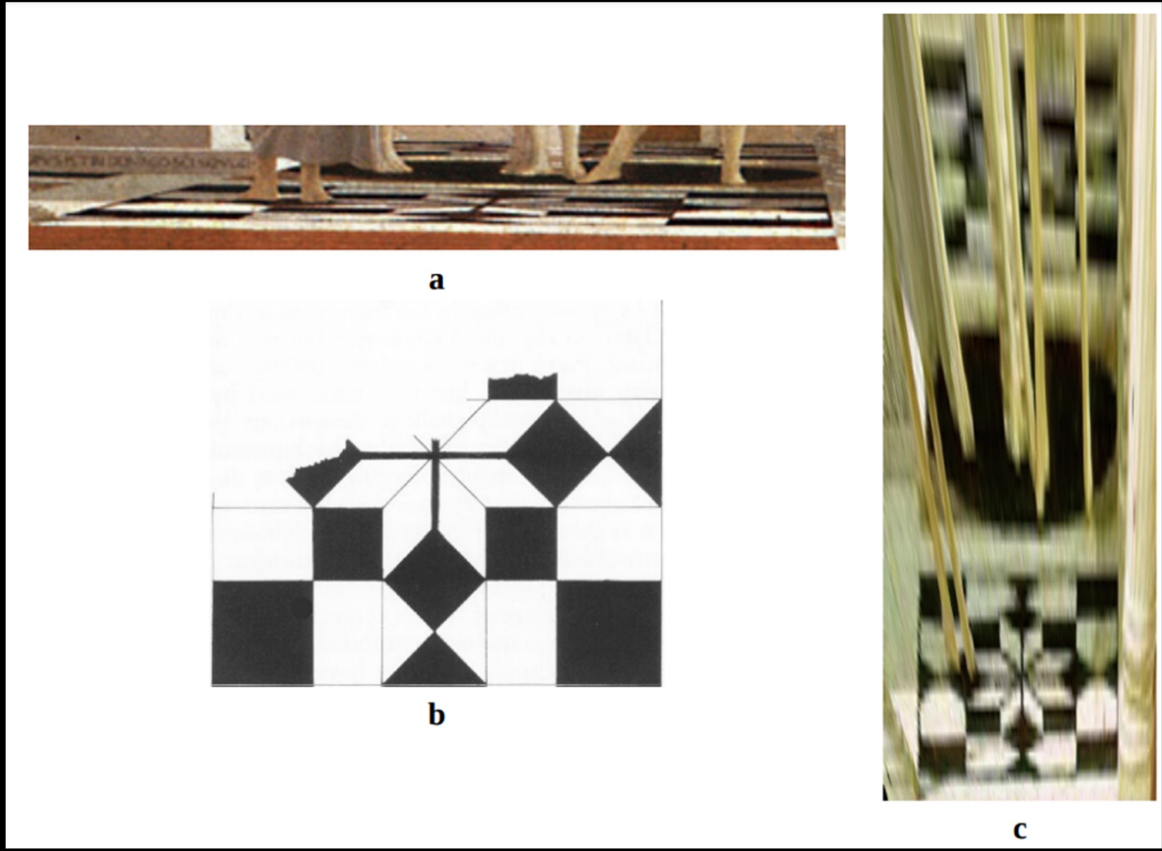
MVSNet – Differential Homography

Hans Holbein, The Ambassadors (1533)

MVSNet – Differential Homography



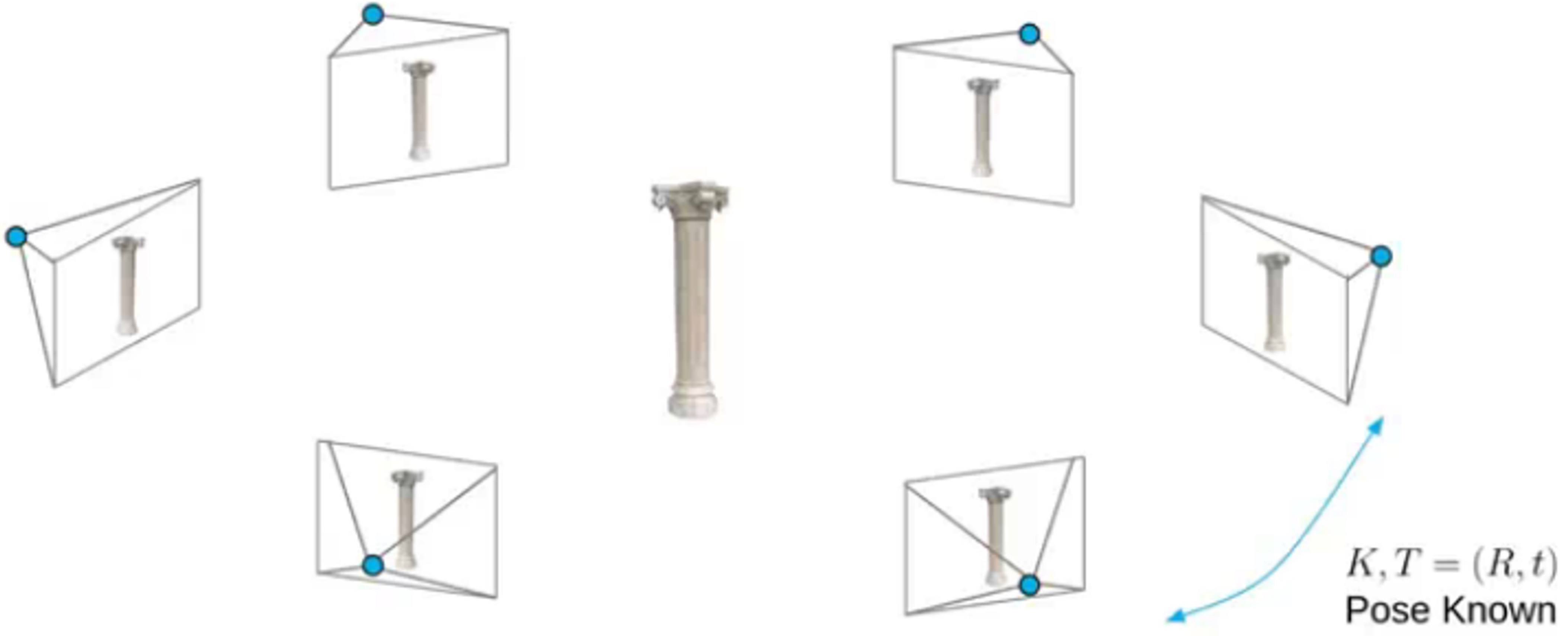
Piero della Francesca, Flaggelation (1468)



$$\mathbf{p}_{i,j} = \mathbf{K}_i \cdot (\mathbf{R}_{0,i} \cdot (\mathbf{K}_0^{-1} \cdot \mathbf{p} \cdot d_j) + \mathbf{t}_{0,i})$$

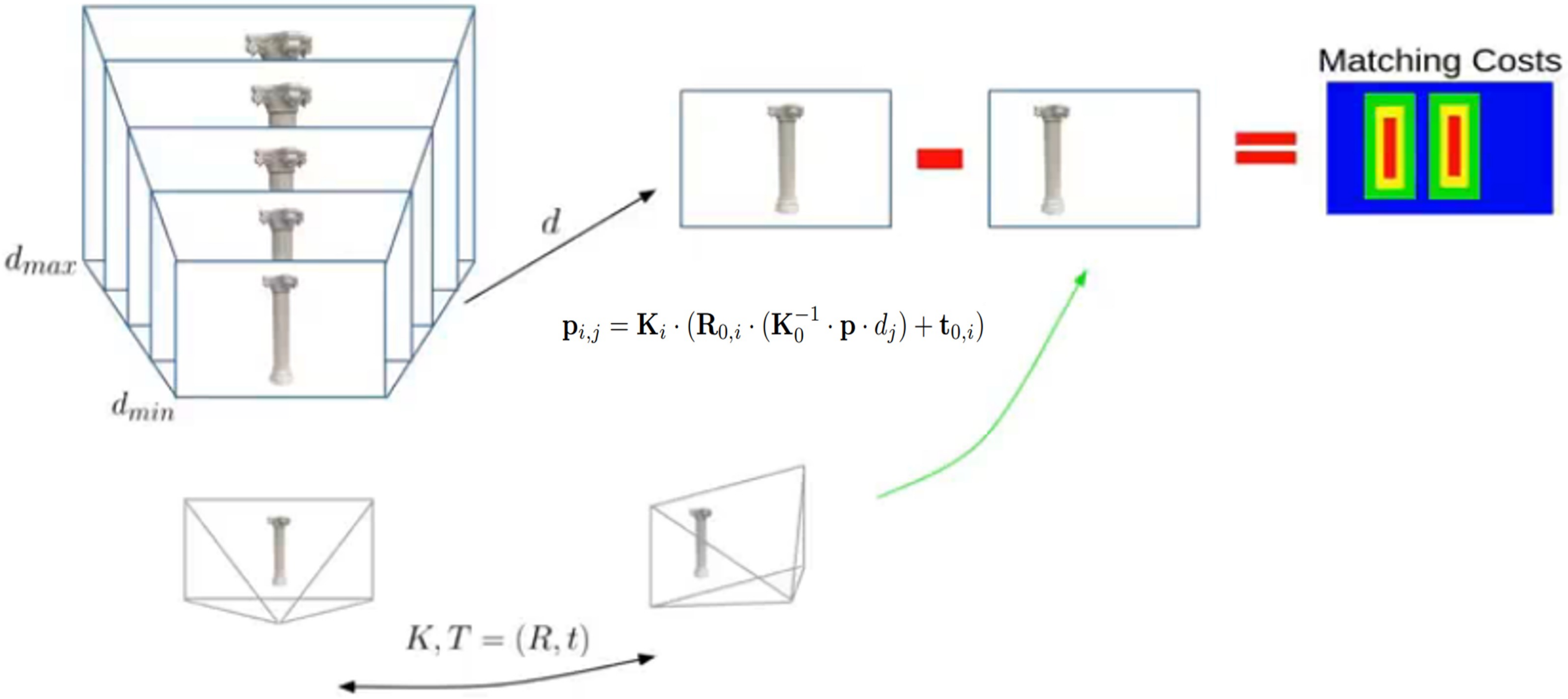
Criminisi et. al. (2002): Bringing Pictorial Space to Life

Multi-view stereo - plane sweep stereo



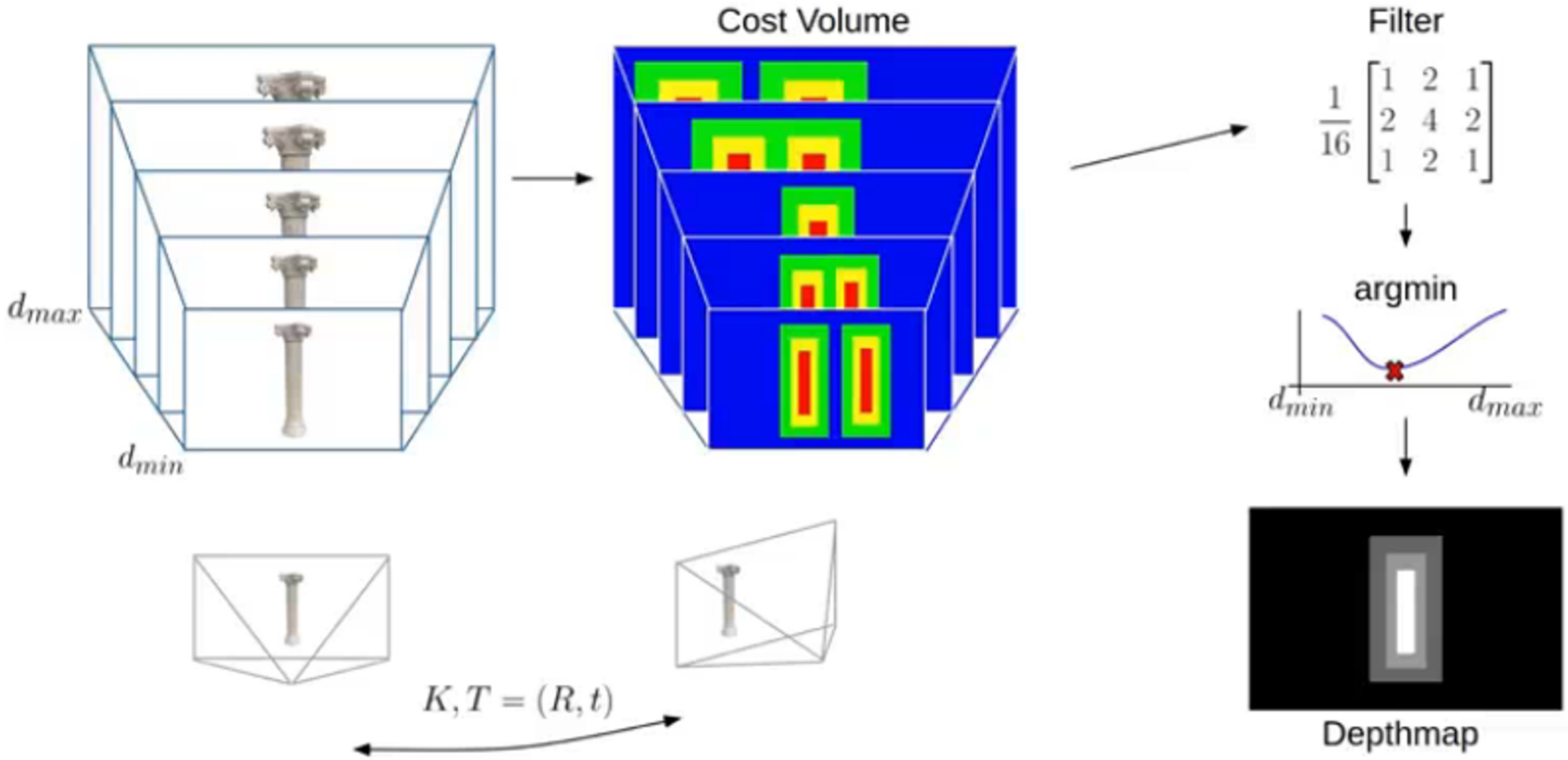
Slide credit: W. Nicholas Greene

Multi-view stereo - plane sweep stereo



Slide credit: W. Nicholas Greene

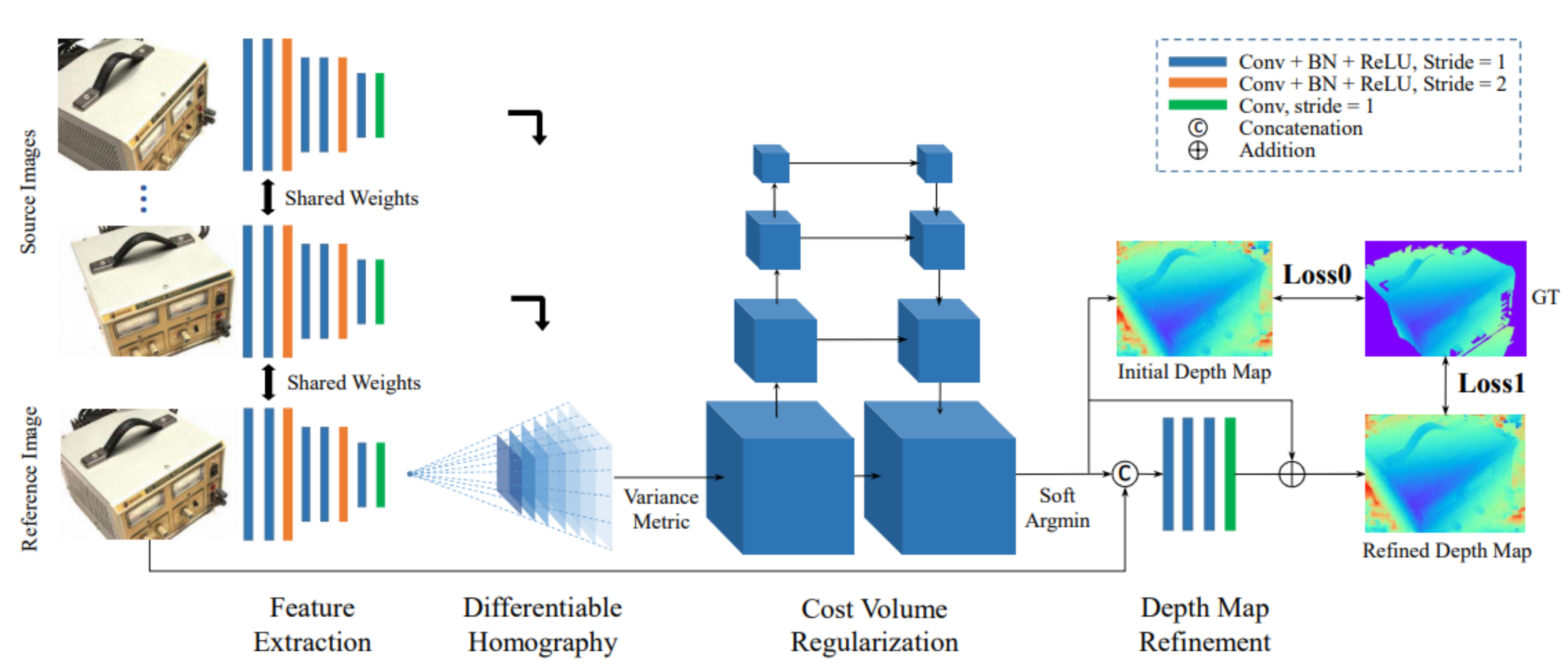
Multi-view stereo - plane sweep stereo

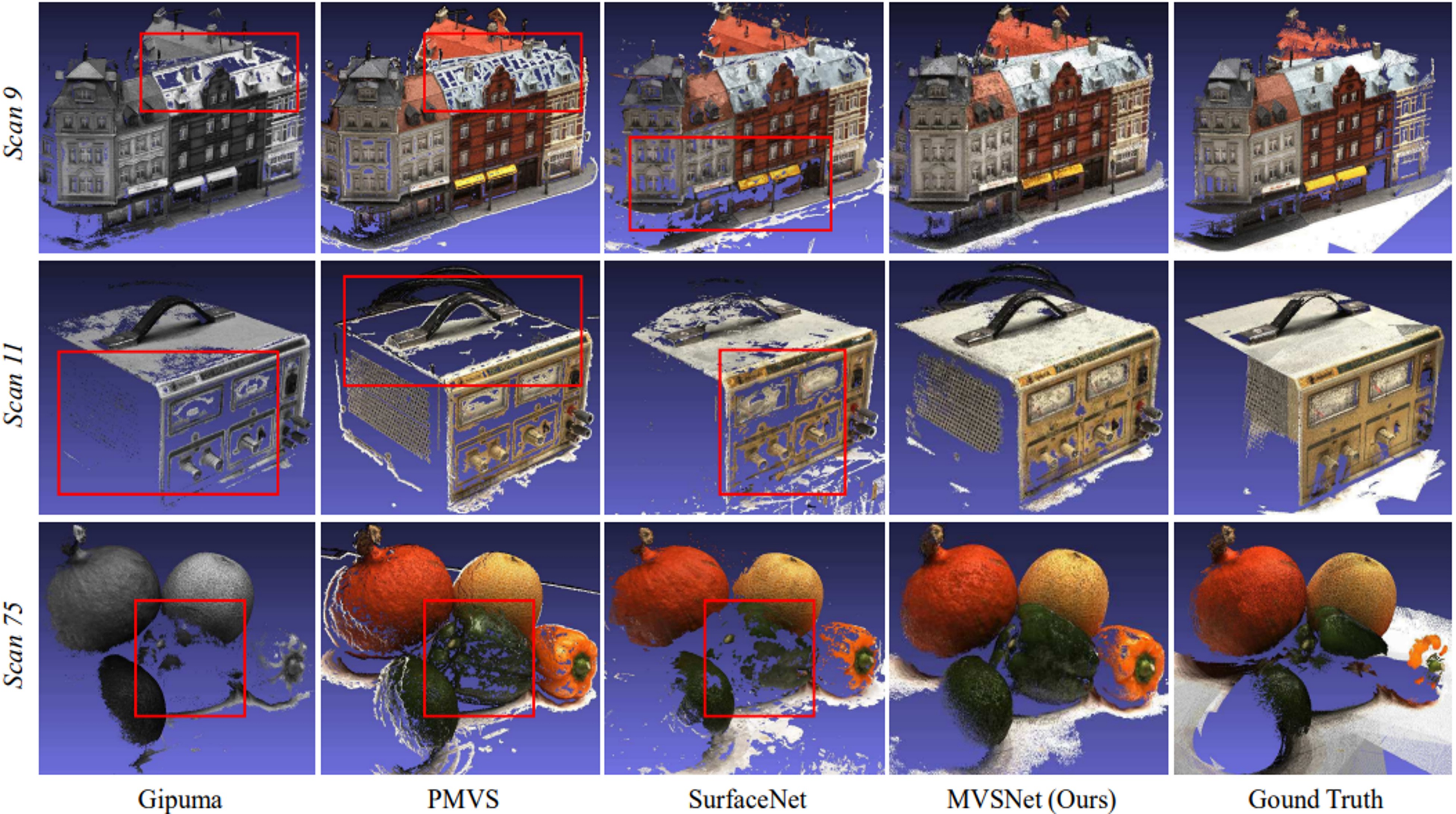


Slide credit: W. Nicholas Greene

MVSNET

Yao Yao et. al.: MVSNet: Depth Inference for Unstructured Multi-view Stereo. ECCV 2018



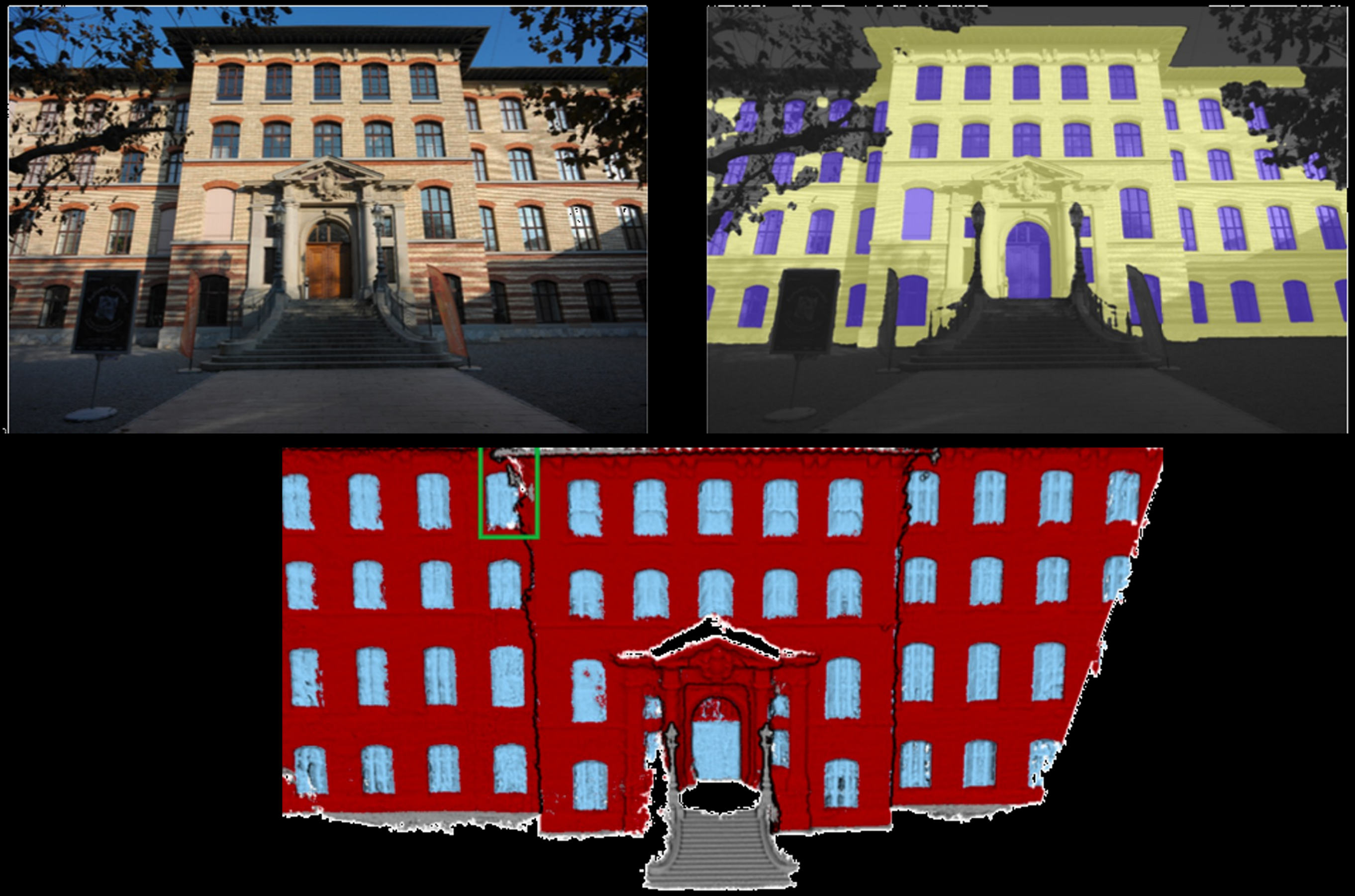

Scan 9
Scan 11
Scan 75
Gipuma
PMVS
SurfaceNet
MVSNet (Ours)
Gound Truth

DDLMVS

This video demonstrates visual comparisons with

COLMAP and PatchmatchNet

Semantic MVS



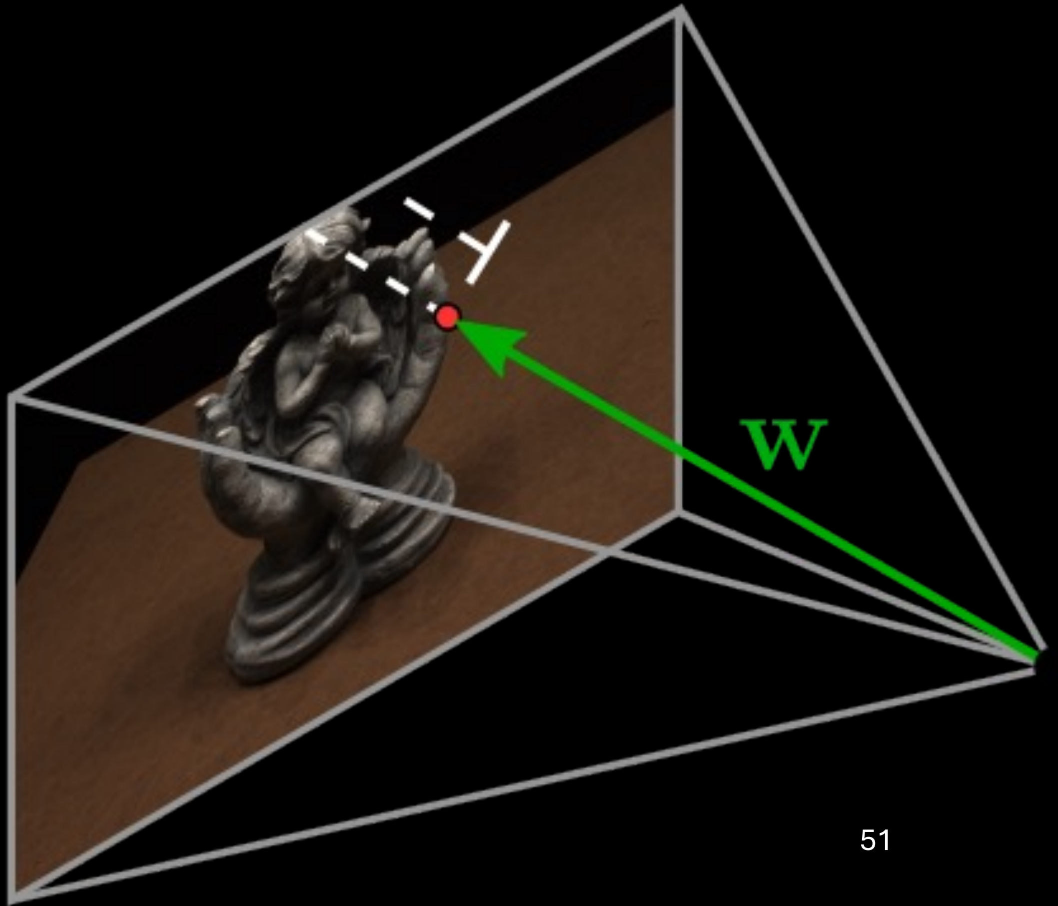
Input image

Semantic Reconstruction

Groundtruth

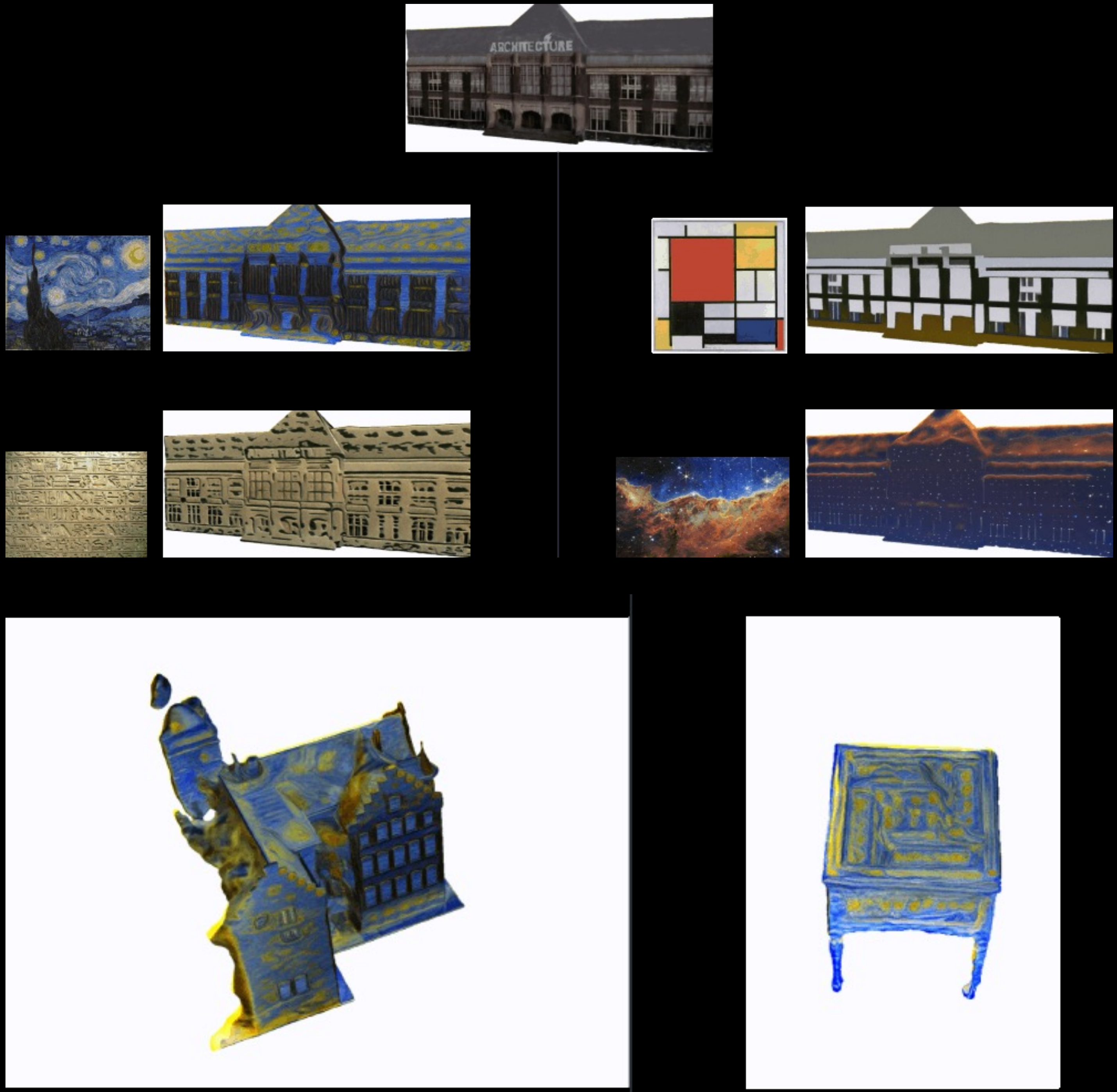
Source: Ioanna Panagiotidou, Semantic MVS (2023)

Differentiable Surface Rendering



Surface Reconstruction and Stylization

I. Collect masked calibrated images
II. Compute surface using rendering
III. Apply stylization to the surface

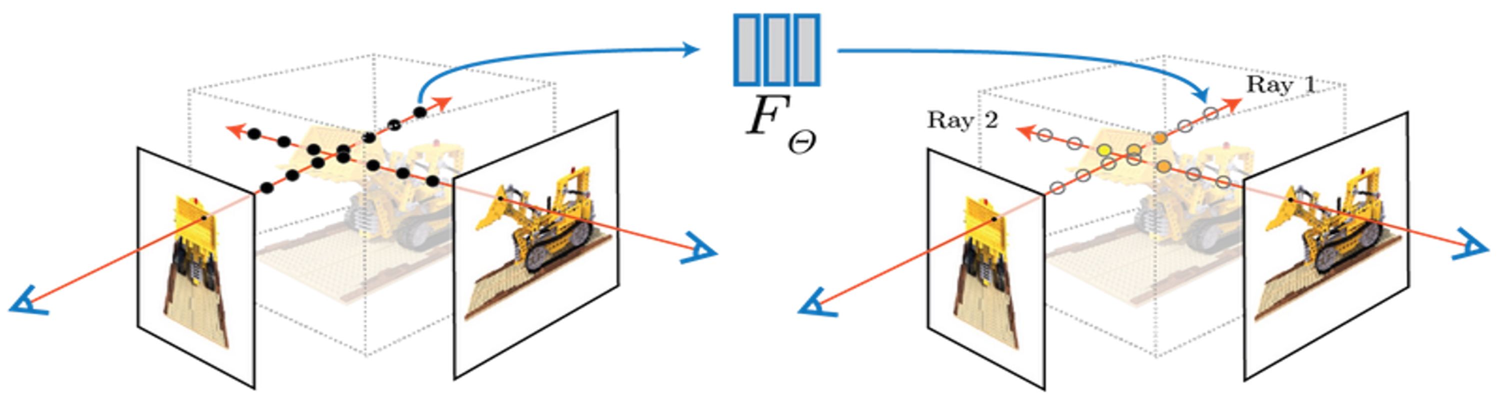


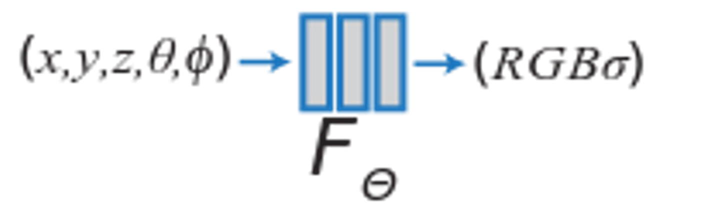
Source: Fabian Visser, StyleSDF (2023)

NeRF revolution

What is NeRF

- The word **Neural** obviously means that there's a Neural Network involved
- **Radiance** refers to the radiance of the scene that the Neural Network outputs. It is basically describing how much light is being emitted by a point in space in each direction, and
- The word **Field** means that the Neural Network models a continuous and non-discretized representation of the scene that it learns.





Assumptions:

- Camera poses are known
- Scene is static, objects do not move
- The scene appearance is constant
- Dense input capture

Architecture:

- 9 Layers MLP + ReLU
- 256 neurons in each layer
- 5D input (x,y,z) + view direction with PE
- 4D output representing RGB+density

NeRF Improvements

- Geometry → NeuS, VolSDF
- Speed → Plenoctrees, DVGO
- Memory-Time trade-off → TensorRF, Instant-NGP
- Sparse images → ReconFusion, DietNeRF
- Stylization → ARF, MuVieCAST
- Sparse pointcloud input → PointNeRF, Gaussian Splatting

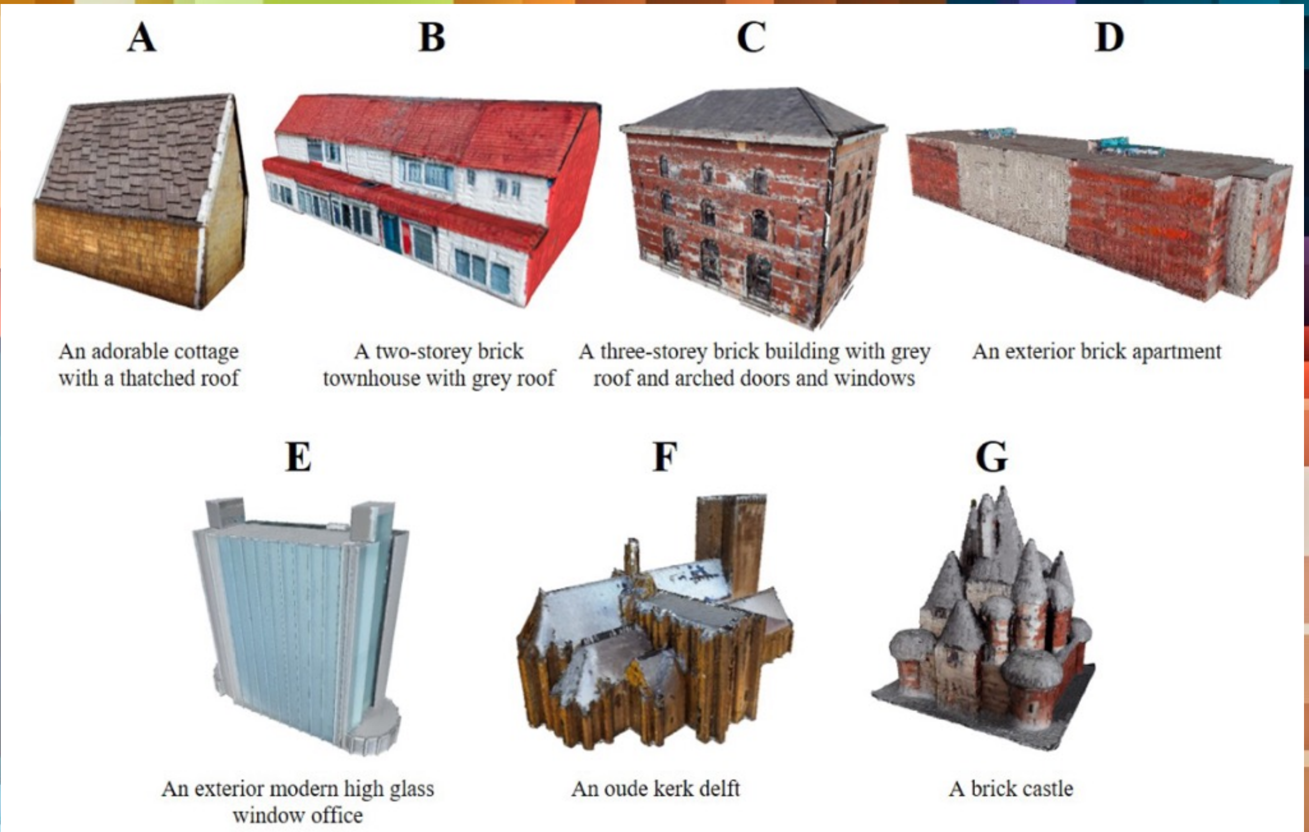
GenAI for 3D: Text-to-3D Generation (DreamFusion)

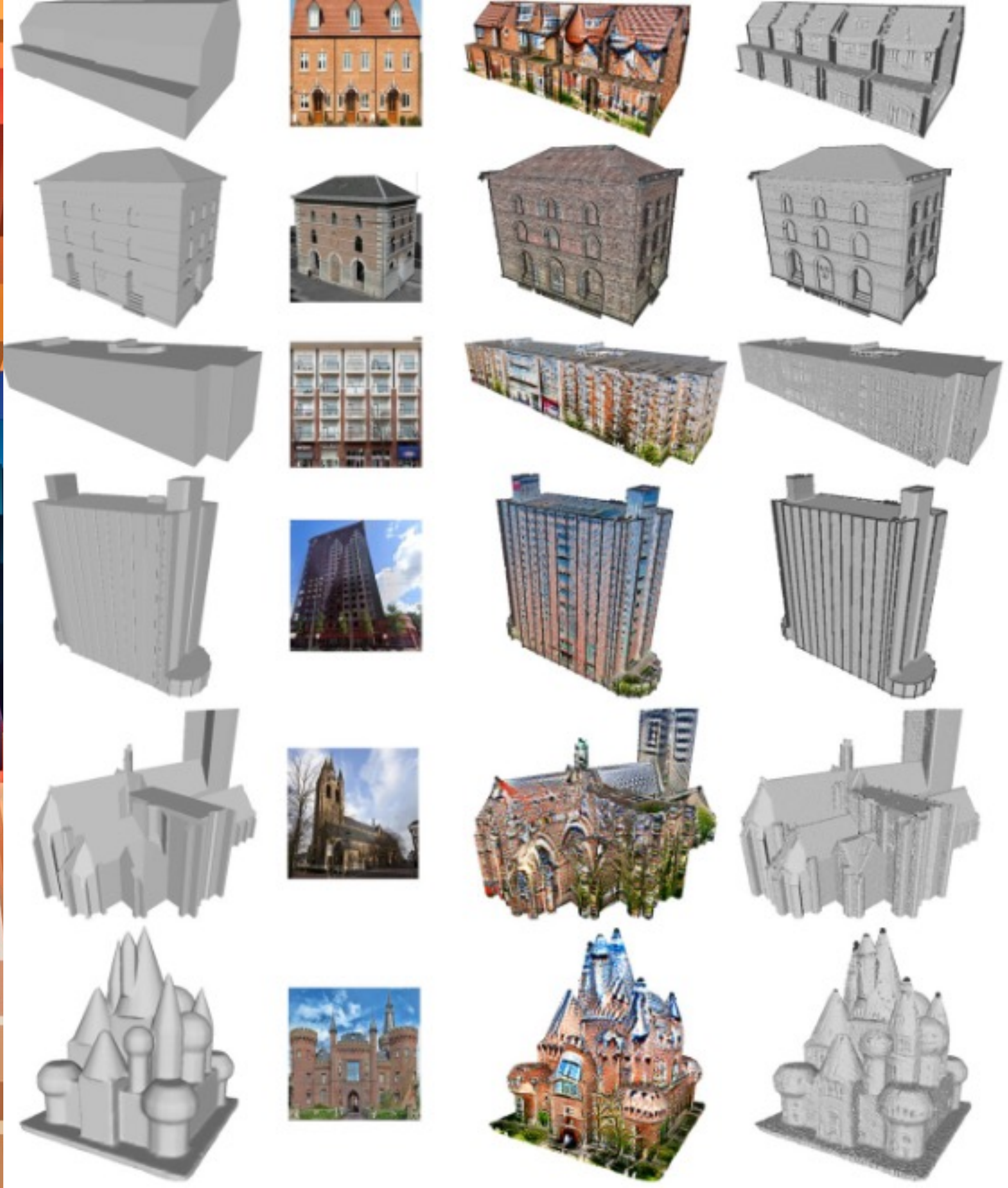
GenAI for 3D: Sparse Reconstruction (ReconFusion)

3 views 6 views 9 views

Source: Yingxin Feng, (2024)

GenAI for 3D: Texturing the Geometry





Thanks for listening.