NYU

Virtual and Augmented Reality

CS-GY 9223/CUSP-GX 6004

https://nyu-icl.github.io/courses/2022fall-vr-ar

Prof. Qi Sun
qisun@nyu.edu | www.qisun.me

Final Project Presentation

2.5min in total (no more) =

cover page +

~.5min on motivation/problem +

~1min on system showcasing +

~1min on tech details

Panoramic Imaging and Cinematic VR

Jaunt
VR


Jaunt VR
JAUNT

Lytro

Lytro

Google

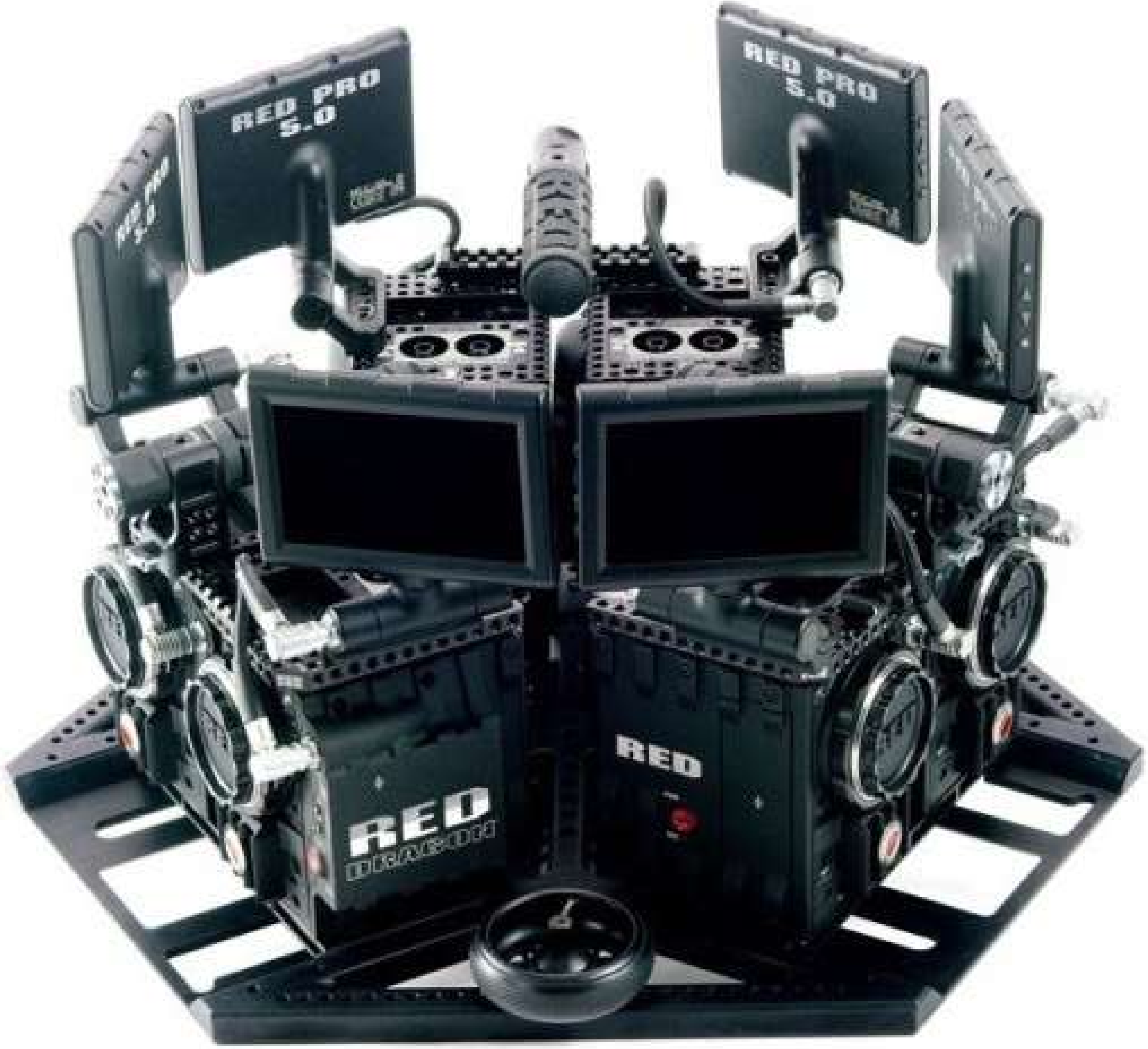
Nokia

Facebook

see Brian Cabral’s SCIEN talk @ talks.stanford.edu


facebook
SURROUND360
facebook
SURROUND360

Red

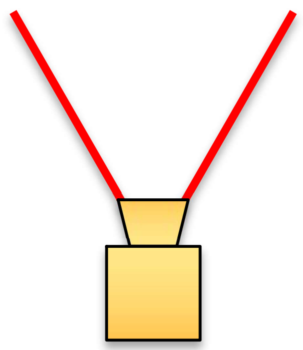


Samsung

Panorama v Stereo Movie v Stereo Panorama

Panorama

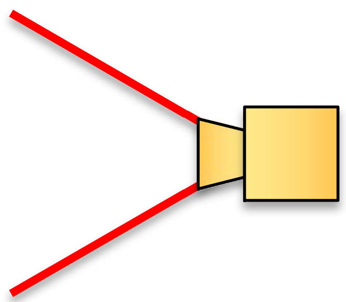
mono & head rotation



Panorama v Stereo Movie v Stereo Panorama

Panorama

mono & head rotation

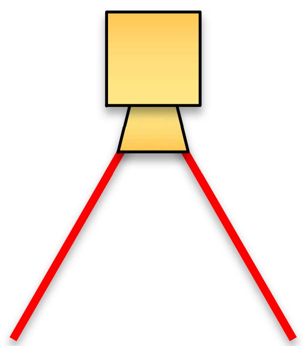


Panorama v Stereo Movie v Stereo

Panorama

Panorama

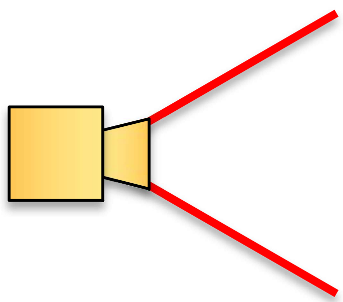
mono & head rotation



Panorama v Stereo Movie v Stereo Panorama

Panorama

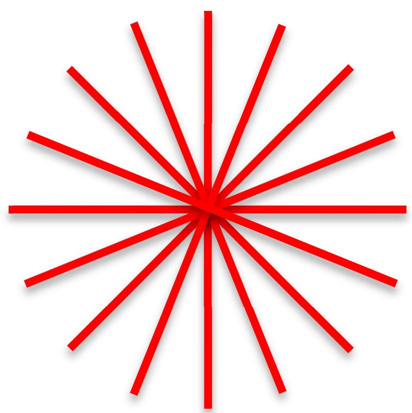
mono & head rotation



Panorama v Stereo Movie v Stereo Panorama

Panorama

mono & head rotation

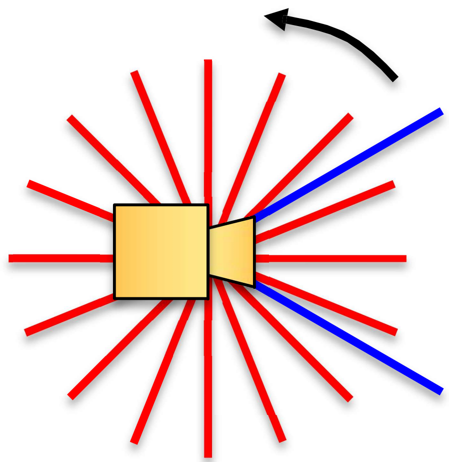


1 center of projection!

Panorama v Stereo Movie v Stereo Panorama

Panorama

mono & head rotation

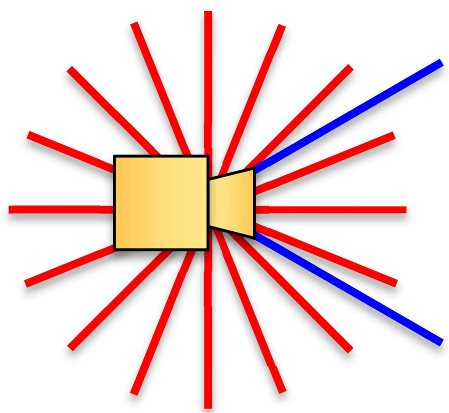


1 center of projection!

Panorama v Stereo Movie v Stereo Panorama

Panorama

mono & head rotation

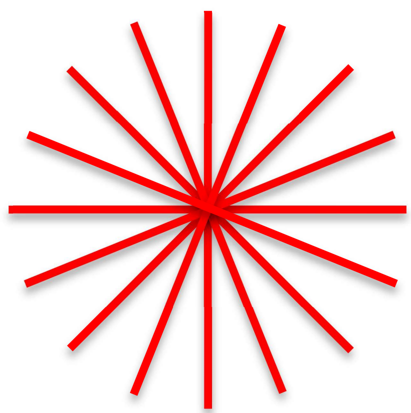


1 center of projection!

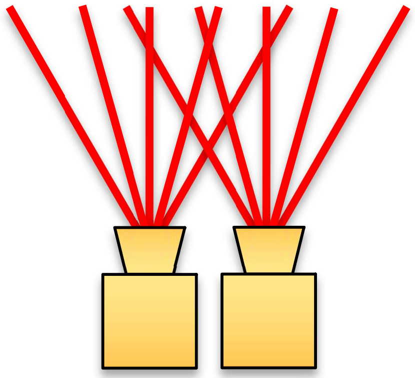
center of projection

Panorama v Stereo Movie v Stereo Panorama

Panorama	Stereo	Stereo Panorama
mono & head rotation	stereo & no head rotation	stereo & head rotation

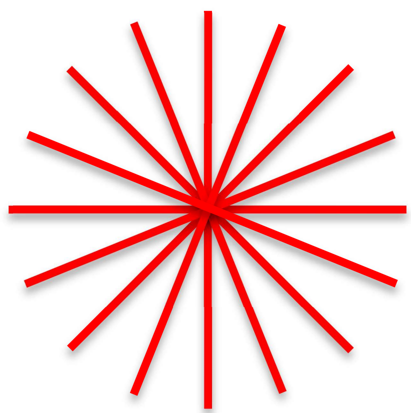


1 center of projection!

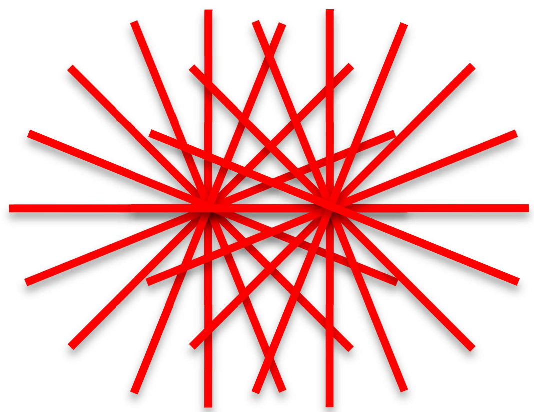


Panorama v Stereo Movie v Stereo Panorama

Panorama	Stereo	Stereo Panorama
mono & head rotation	stereo & no head rotation	stereo & head rotation



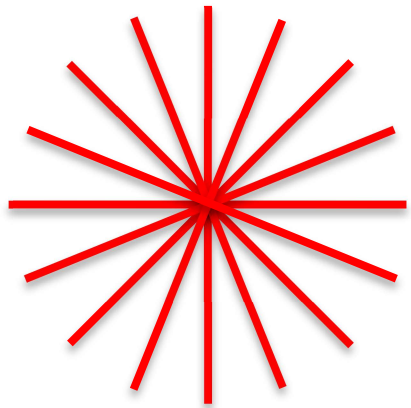
1 center of projection!

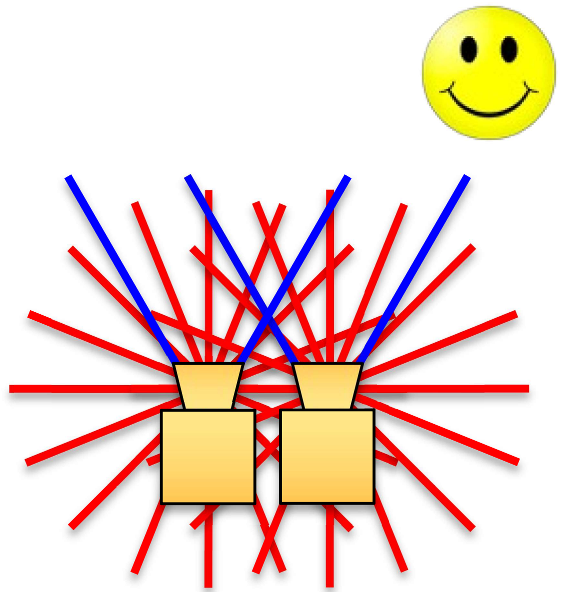


2 centers of projection!

Panorama v Stereo Movie v Stereo Panorama

Panorama	Stereo	Stereo Panorama
mono & head rotation	stereo & no head rotation	stereo & head rotation



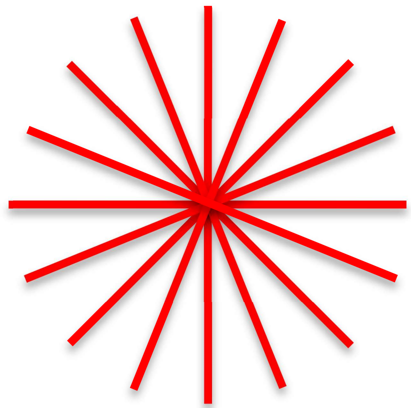


1 center of projection!

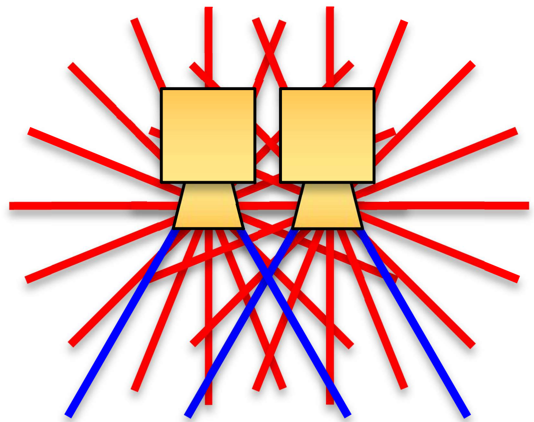
2 centers of projection!

Panorama v Stereo Movie v Stereo Panorama

Panorama	Stereo	Stereo Panorama
mono & head rotation	stereo & no head rotation	stereo & head rotation



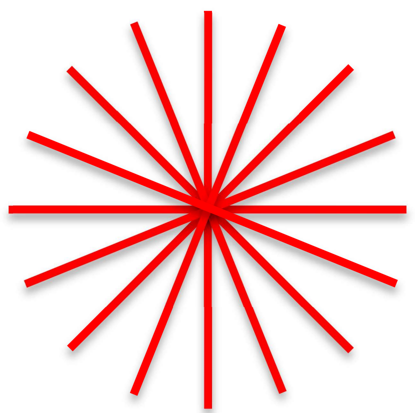
1 center of projection!



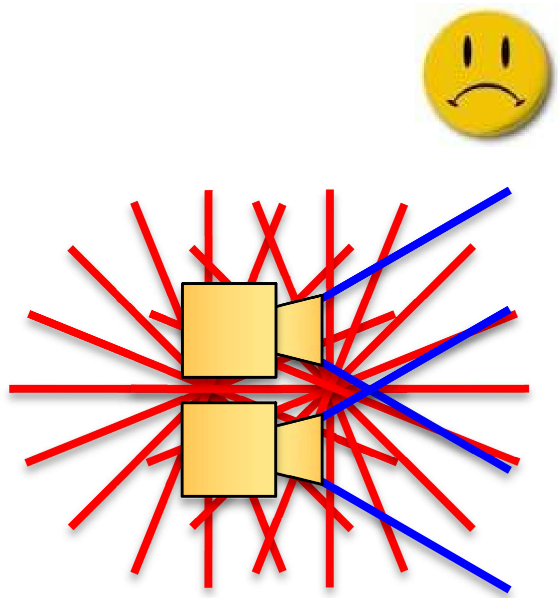
2 centers of projection!

Panorama v Stereo Movie v Stereo Panorama

Panorama	Stereo	Stereo Panorama
mono & head rotation	stereo & no head rotation	stereo & head rotation



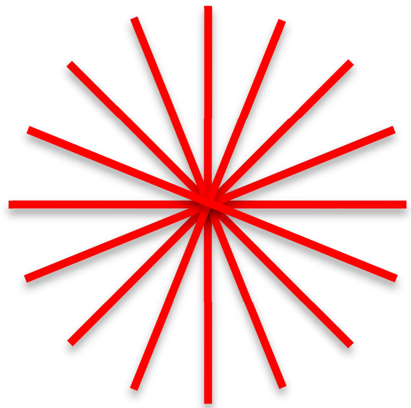
1 center of projection!



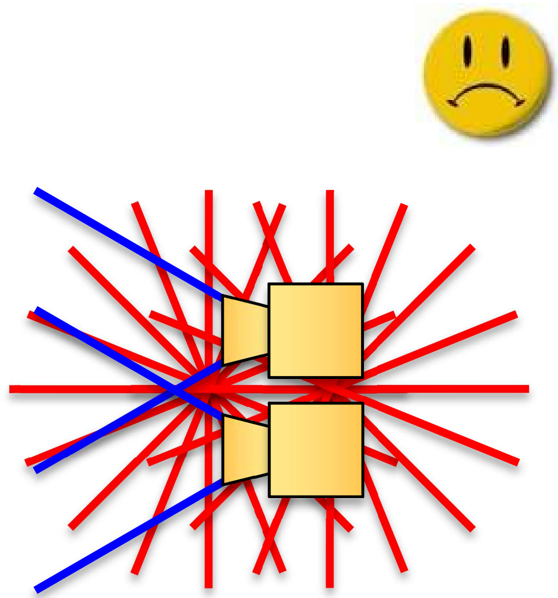
2 centers of projection!

Panorama v Stereo Movie v Stereo Panorama

Panorama	Stereo	Stereo Panorama
mono & head rotation	stereo & no head rotation	stereo & head rotation



1 center of projection!

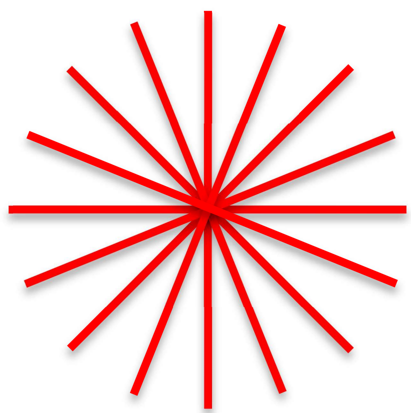


2 centers of projection!

Panorama v Stereo Movie v Stereo Panorama

Panorama

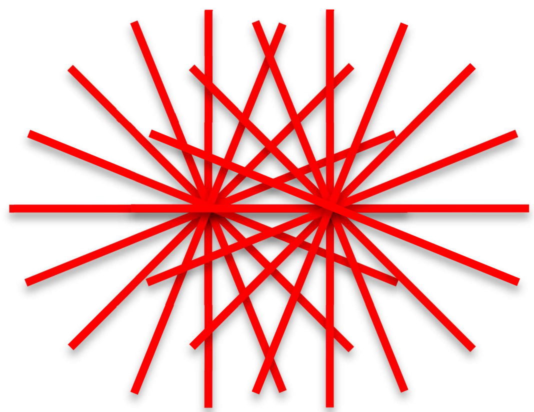
mono & head rotation



1 center of projection!

Stereo

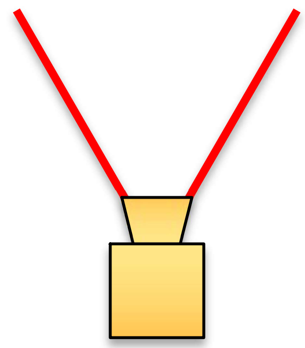
stereo & no head rotation



2 centers of projection!

Stereo Panorama

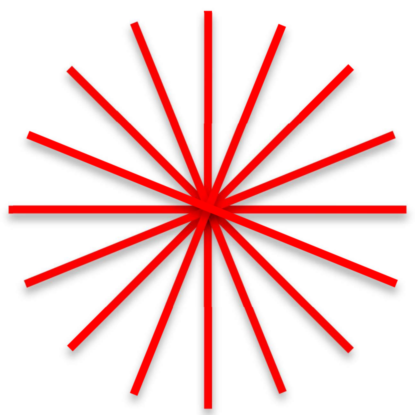
stereo & head rotation



Panorama v Stereo Movie v Stereo Panorama

Panorama

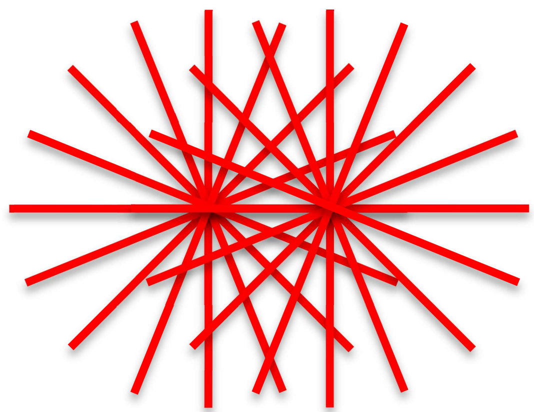
mono & head rotation



1 center of projection!

Stereo

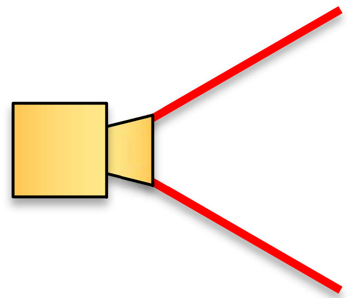
stereo & no head rotation



2 centers of projection!

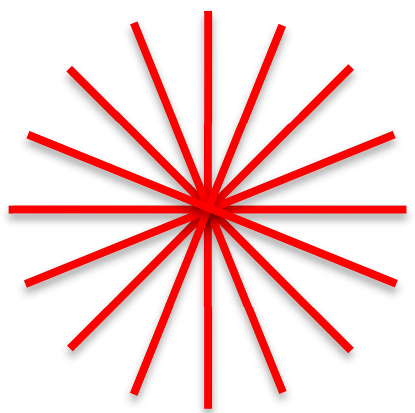
Stereo Panorama

stereo & head rotation

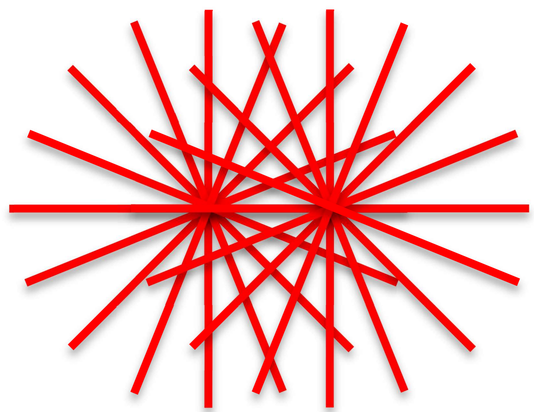


Panorama v Stereo Movie v Stereo Panorama

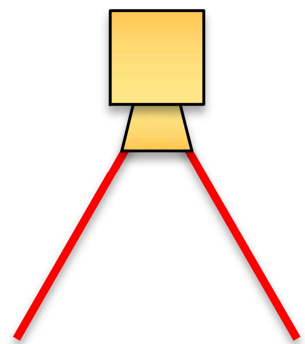
Panorama	Stereo	Stereo Panorama
mono & head rotation	stereo & no head rotation	stereo & head rotation



1 center of projection!



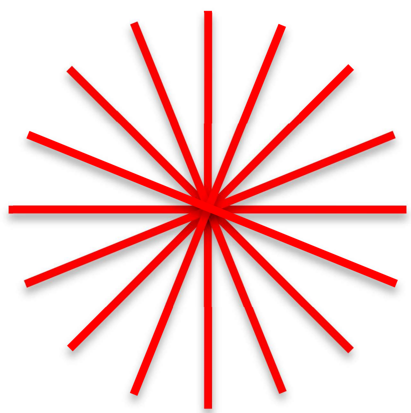
2 centers of projection!



Panorama v Stereo Movie v Stereo Panorama

Panorama

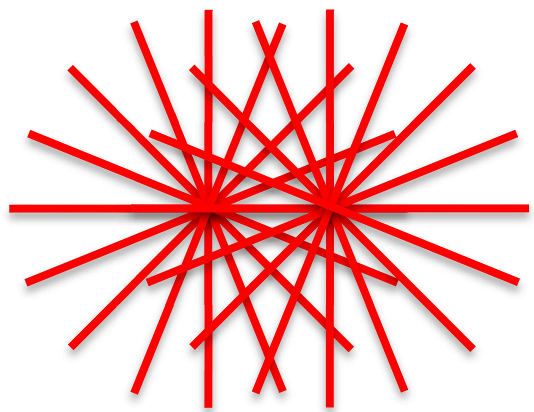
mono & head rotation



1 center of projection!

Stereo

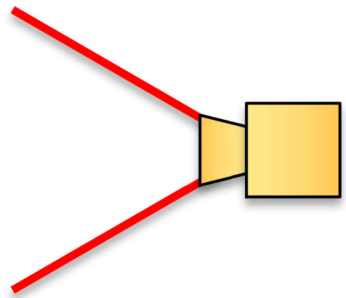
stereo & no head rotation



2 centers of projection!

Stereo Panorama

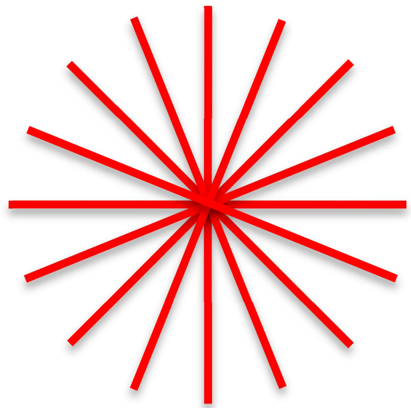
stereo & head rotation



Panorama v Stereo Movie v Stereo Panorama

Panorama

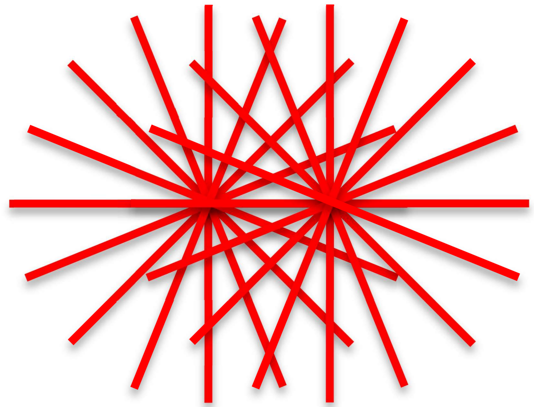
mono & head rotation



1 center of projection!

Stereo

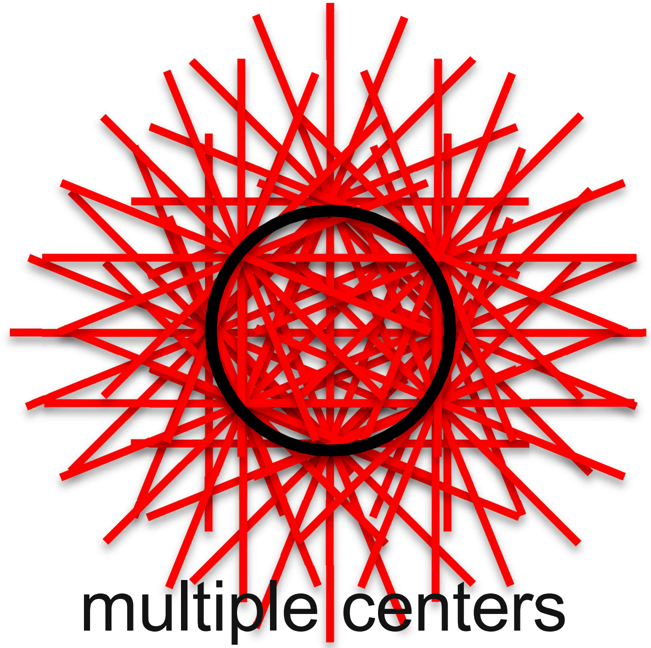
stereo & no head rotation



2 centers of projection!

Stereo Panorama

stereo & head rotation

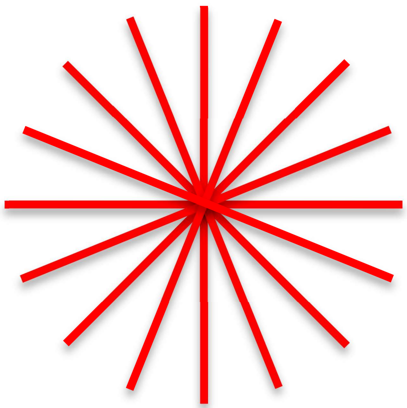


multiple centers of projection

Panorama v Stereo Movie v Stereo Panorama

Panorama

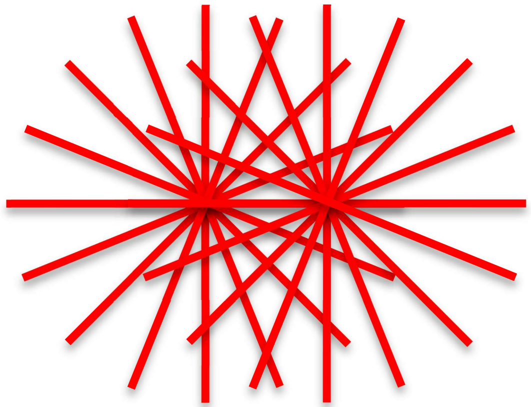
mono & head rotation



1 center of projection!

Stereo

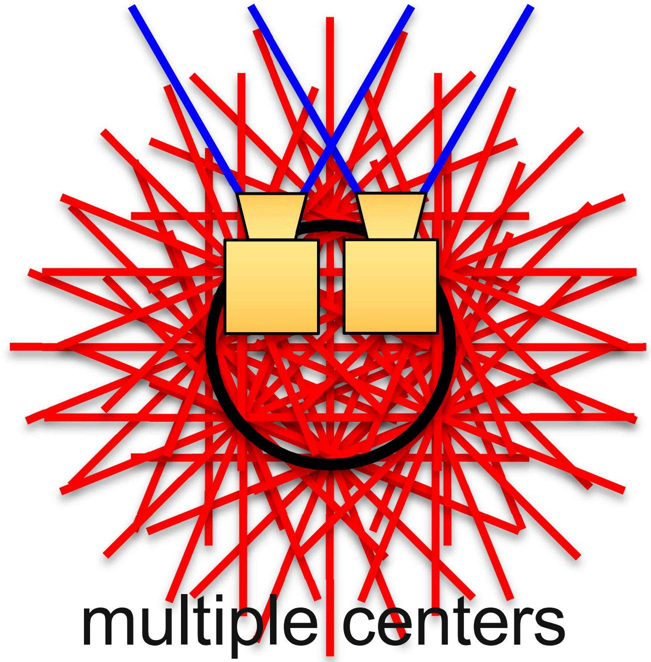
stereo & no head rotation



2 centers of projection!

Stereo Panorama

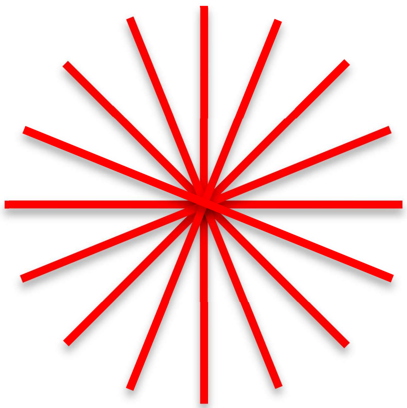
stereo & head rotation



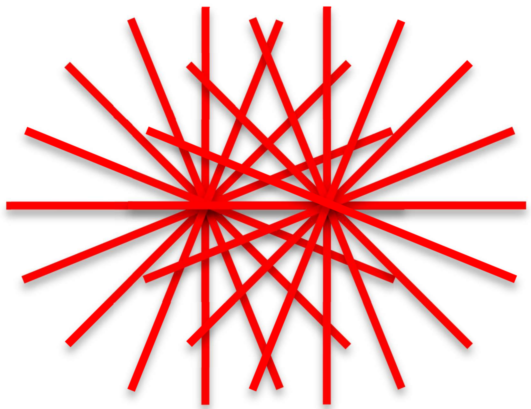
multiple centers of projection

Panorama v Stereo Movie v Stereo Panorama

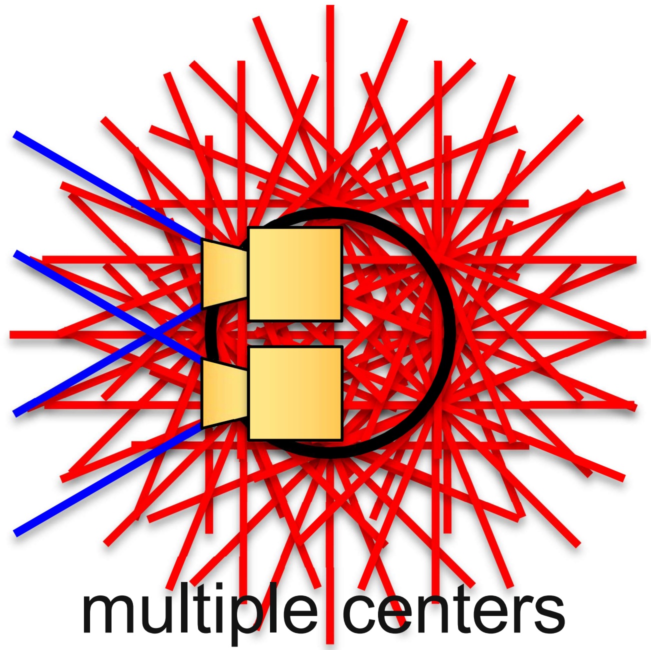
Panorama	Stereo	Stereo Panorama
mono & head rotation	stereo & no head rotation	stereo & head rotation



1 center of projection!



2 centers of projection!



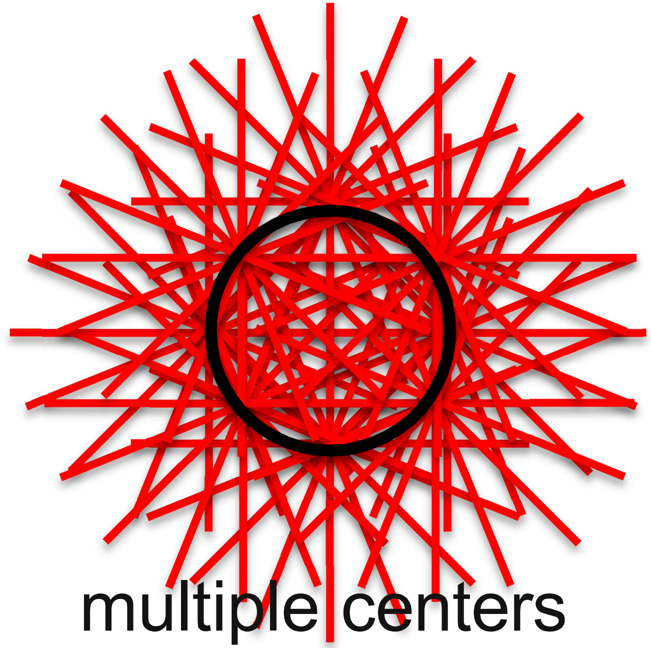
multiple centers of projection

Panorama v Stereo Movie v Stereo Panorama

Light Field!

Stereo Panorama

stereo & head rotation



multiple centers of projection

Panorama v Stereo Movie v Stereo Panorama

Panorama

mono & head rotation

Ricoh Theta

Stereo

stereo & no head rotation

Stereo Panorama

stereo & head rotation

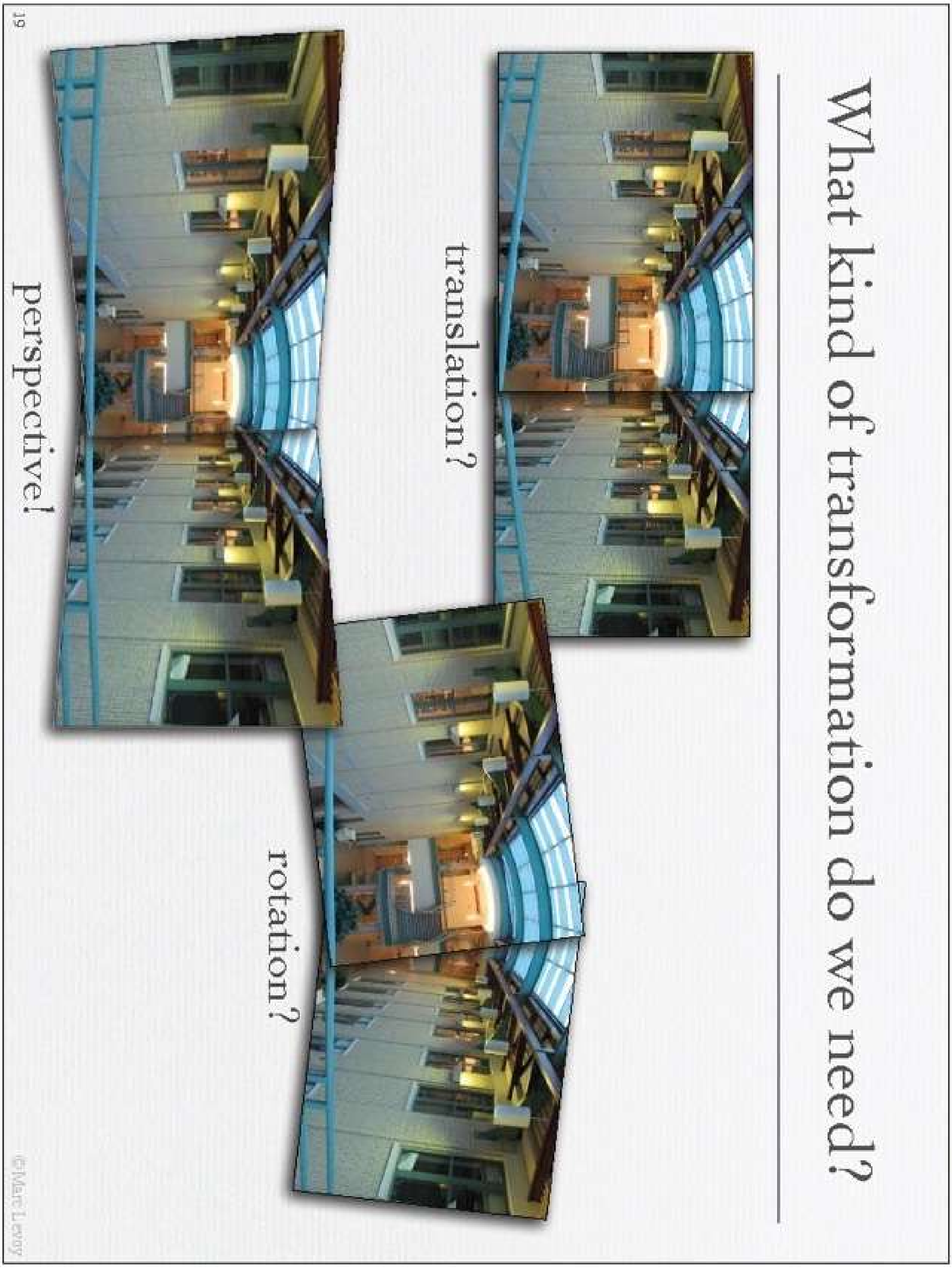
horizontal-only parallax

Panoramas

Slides from Marc Levoy

Panoramas

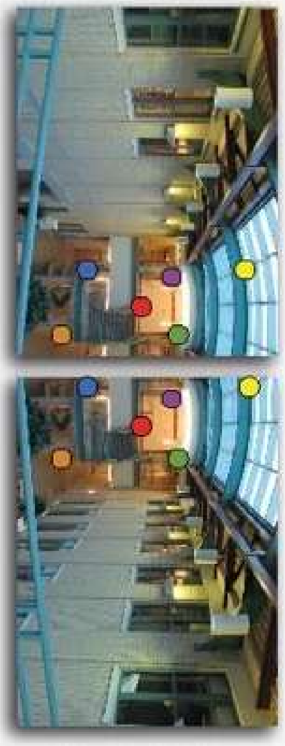
Slides from Marc Levoy



Panoramas

Slides from Marc Levoy

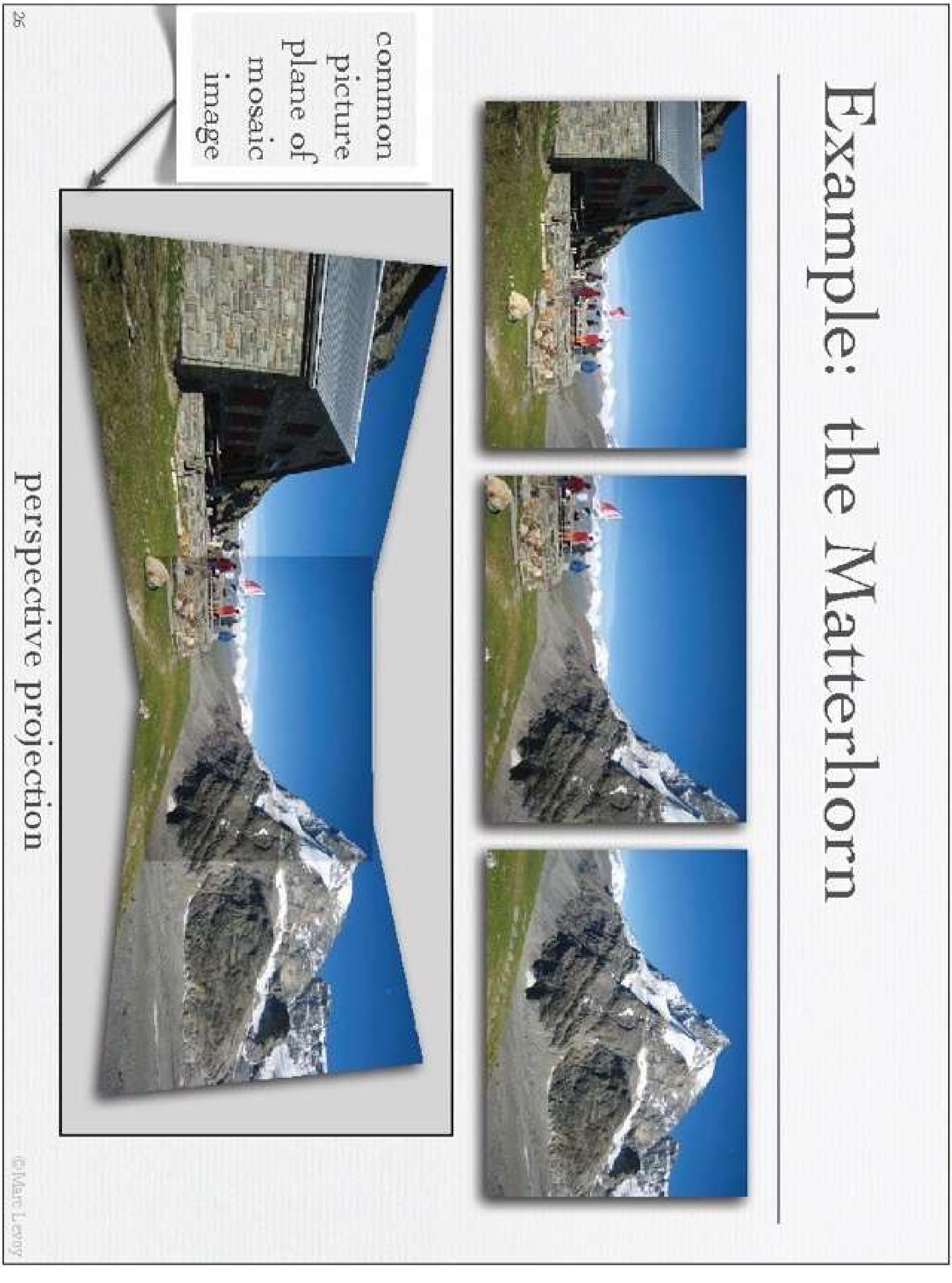
Stitching images together to make a mosaic



- step 1: find corresponding features in a pair of image
- step 2: compute perspective from 2^{nd} to 1^{st} image
- step 3: warp 2^{nd} image so it overlays 1^{st} image
- step 4: blend images where they overlap one another
- repeat for 3^{rd} image and mosaic of first two, etc.

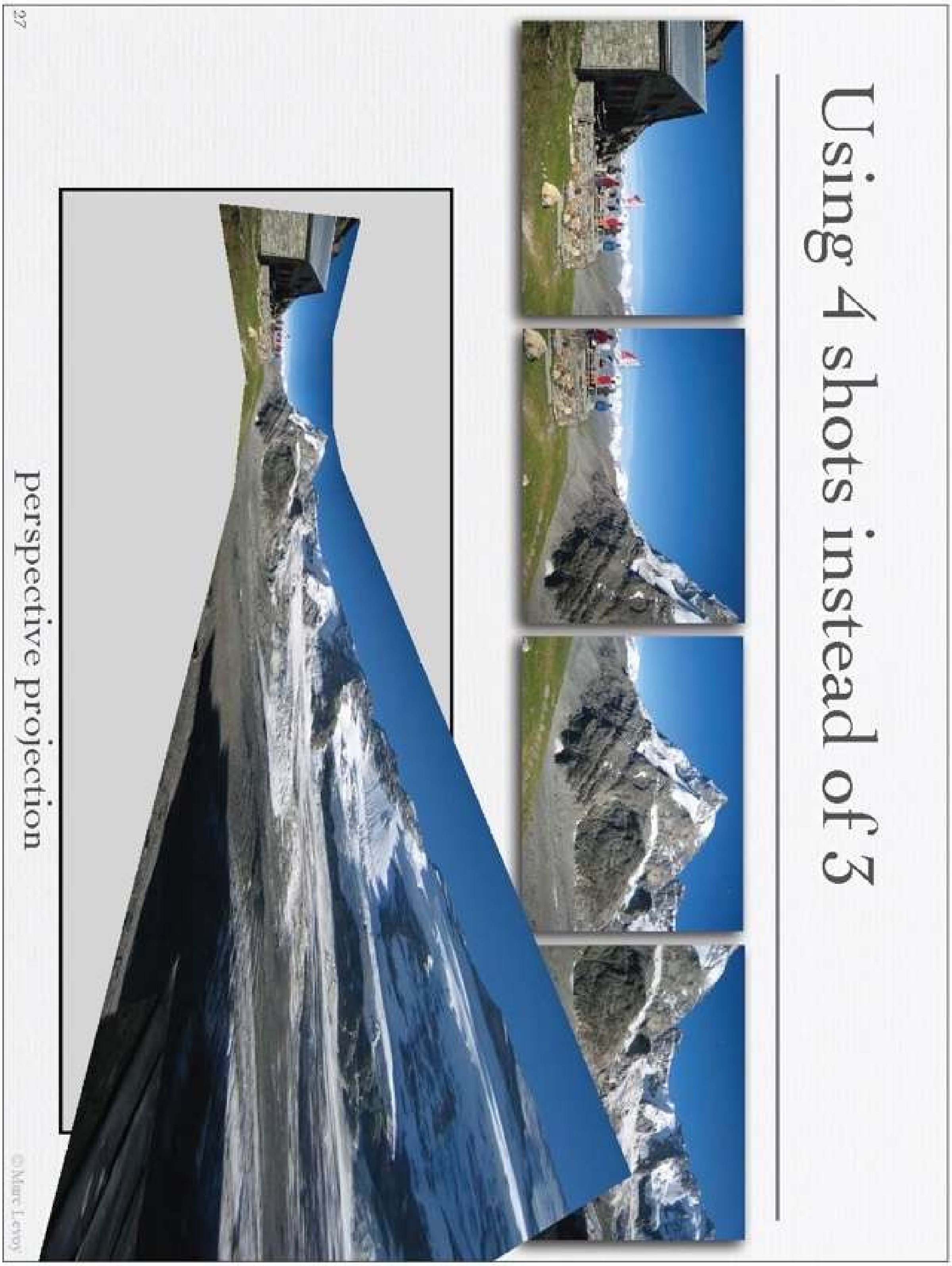
Panoramas

Slides from Marc Levoy



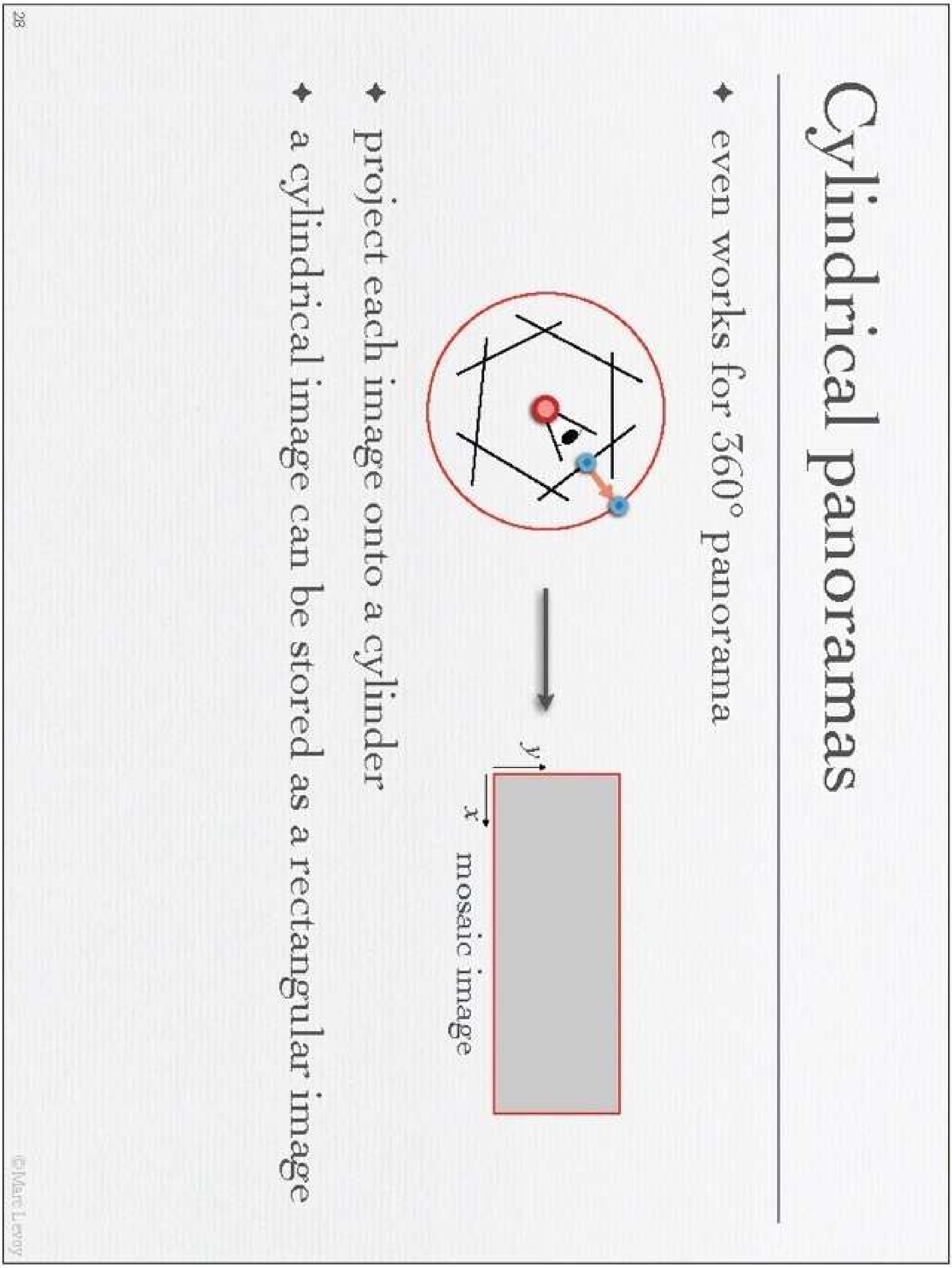
Panoramas

Slides from Marc Levoy



Panoramas

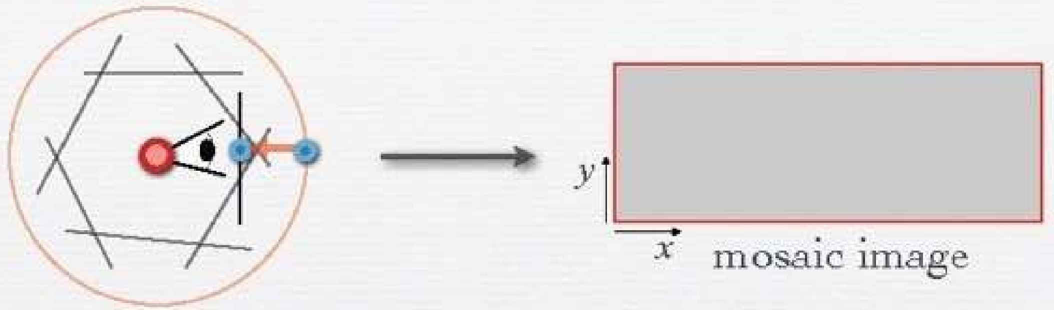
Slides from Marc Levoy



Cylindrical panoramas

(FLASH DEMO)
http://graphics.stanford.edu/courses/cs178/applets/projection.html

- even works for 360° panorama



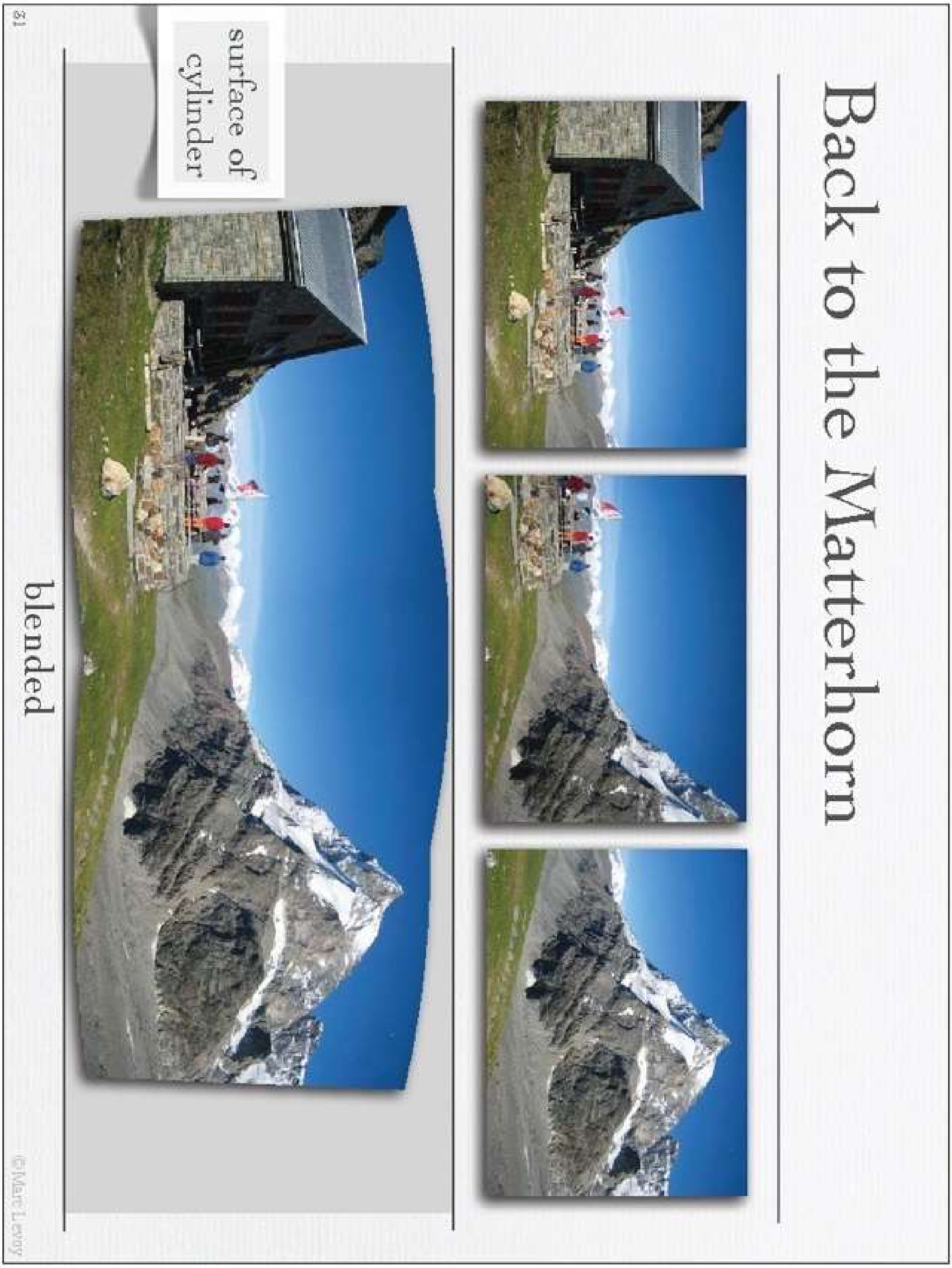
- project each image onto a cylinder
- a cylindrical image can be stored as a rectangular image
- to view without distortion, reproject part of the cylinder onto a picture plane representing the display screen
  - if your FOV is narrow, this view won't be too distorted

Panoramas

Slides from Marc Levoy

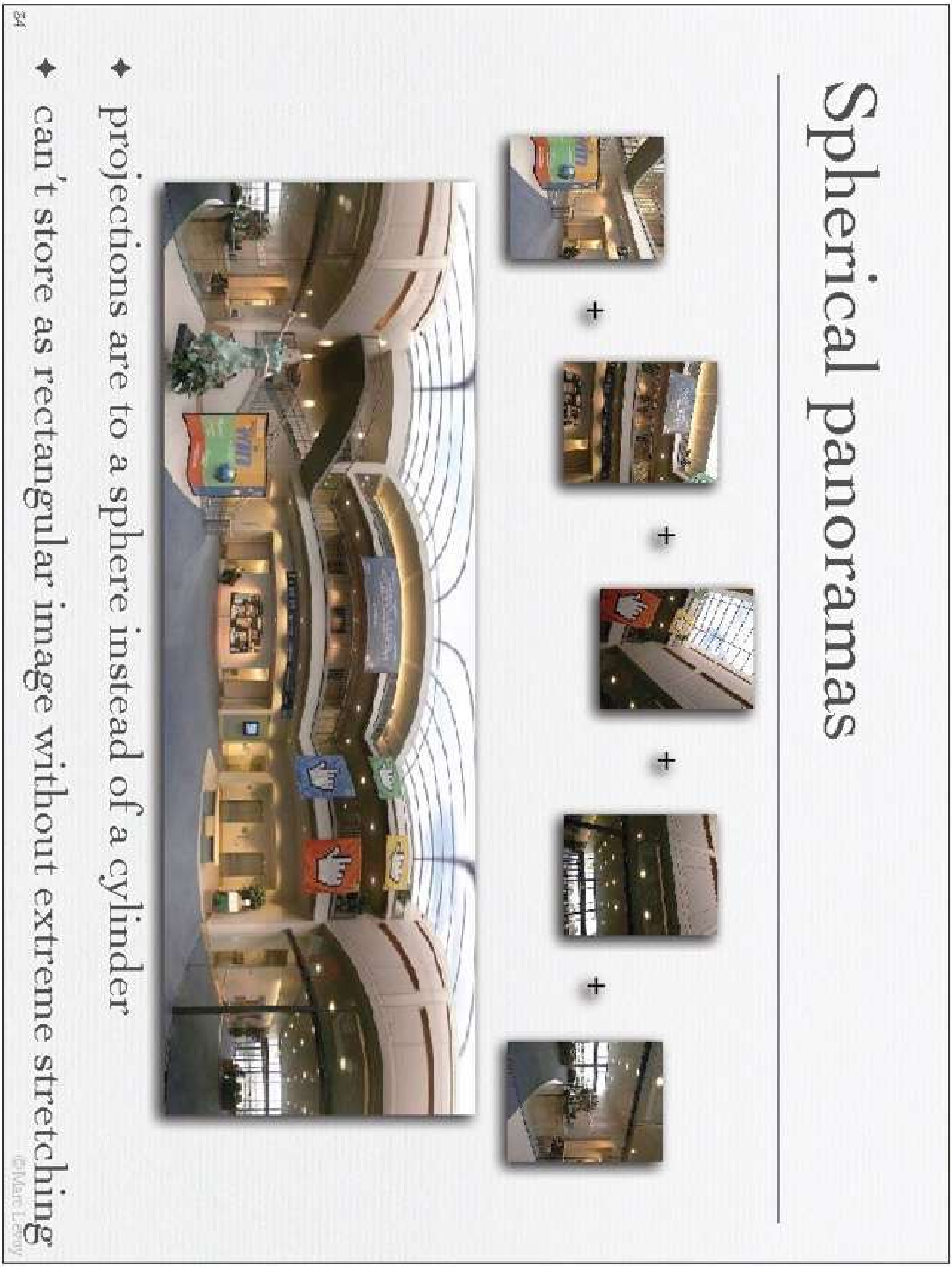
Panoramas

Slides from Marc Levoy

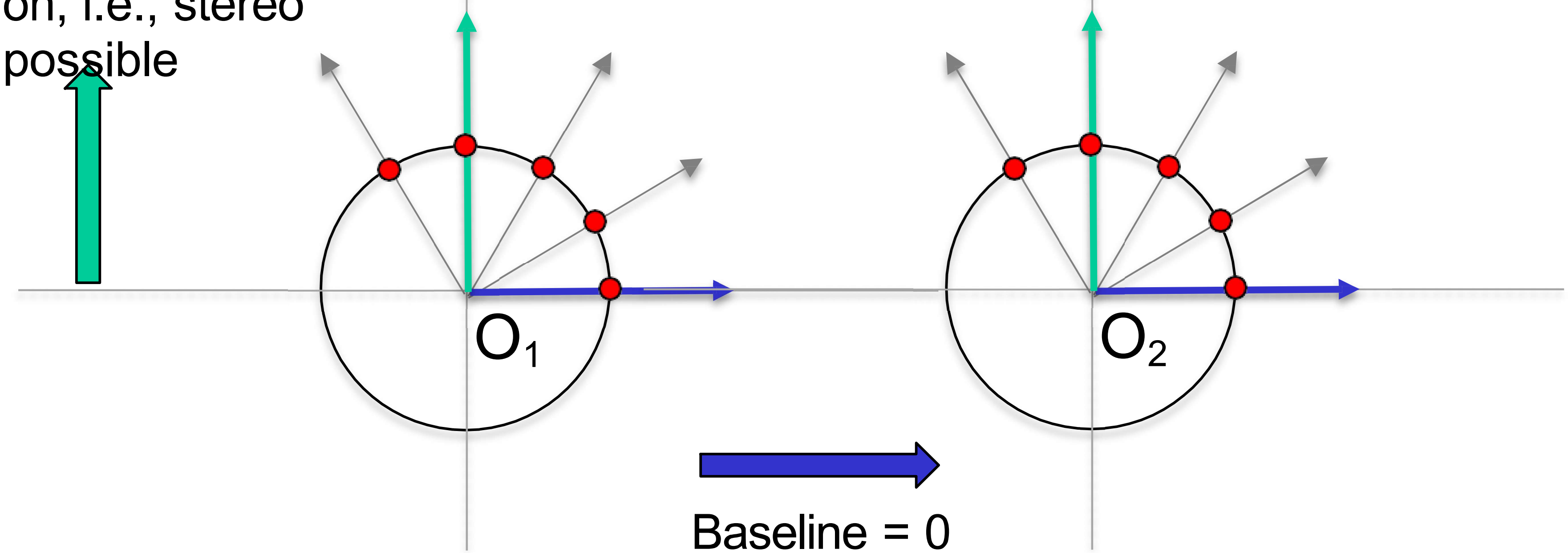


Panoramas

Slides from Marc Levoy



A Pair of Mono Panoramas

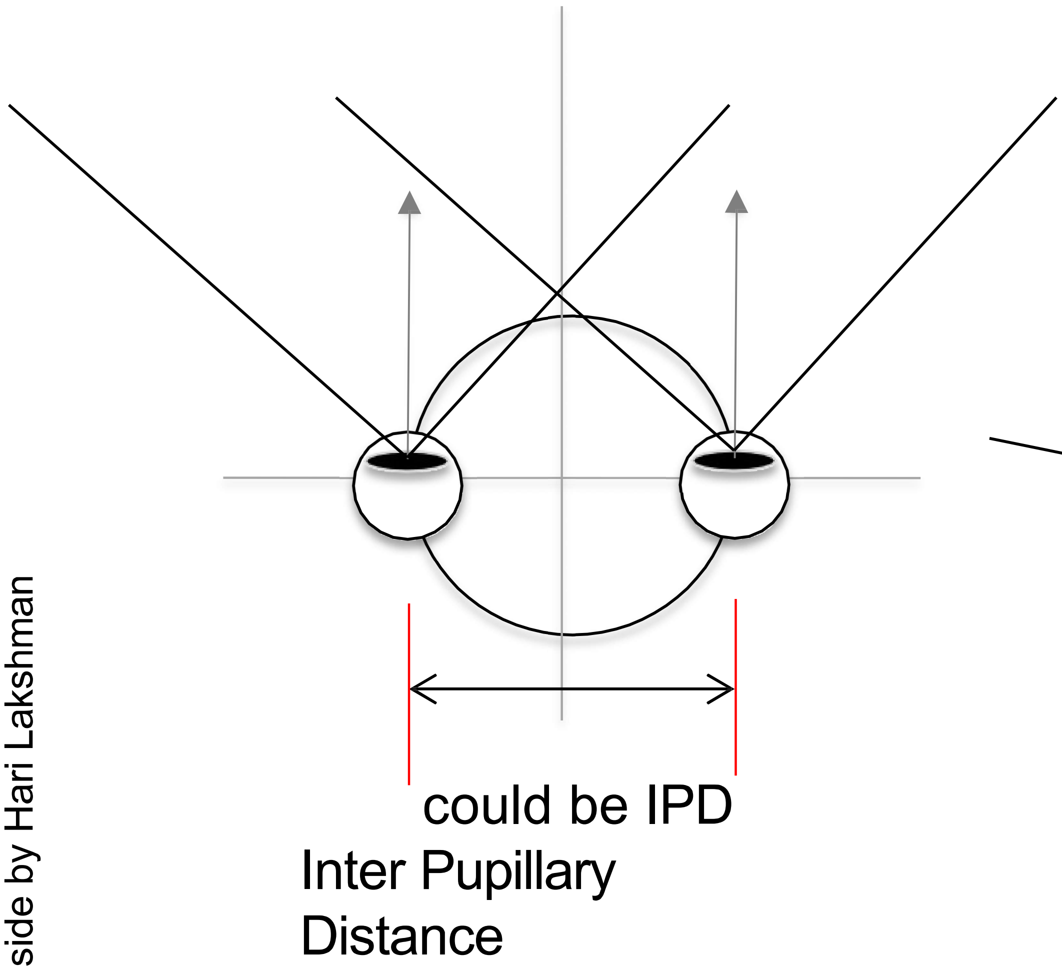


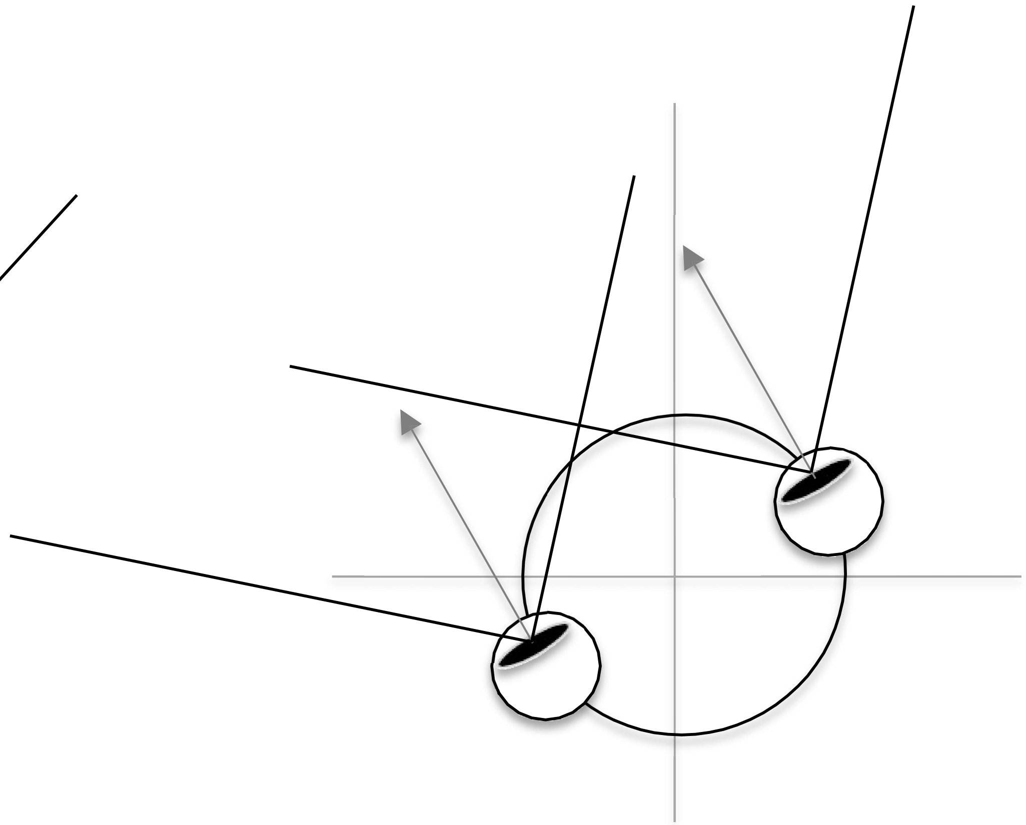
Baseline = $|O_1 - O_2|$ in this viewing direction, i.e., stereo possible

Baseline = 0 in this viewing direction, i.e., no stereo

side by Hari Lakshman

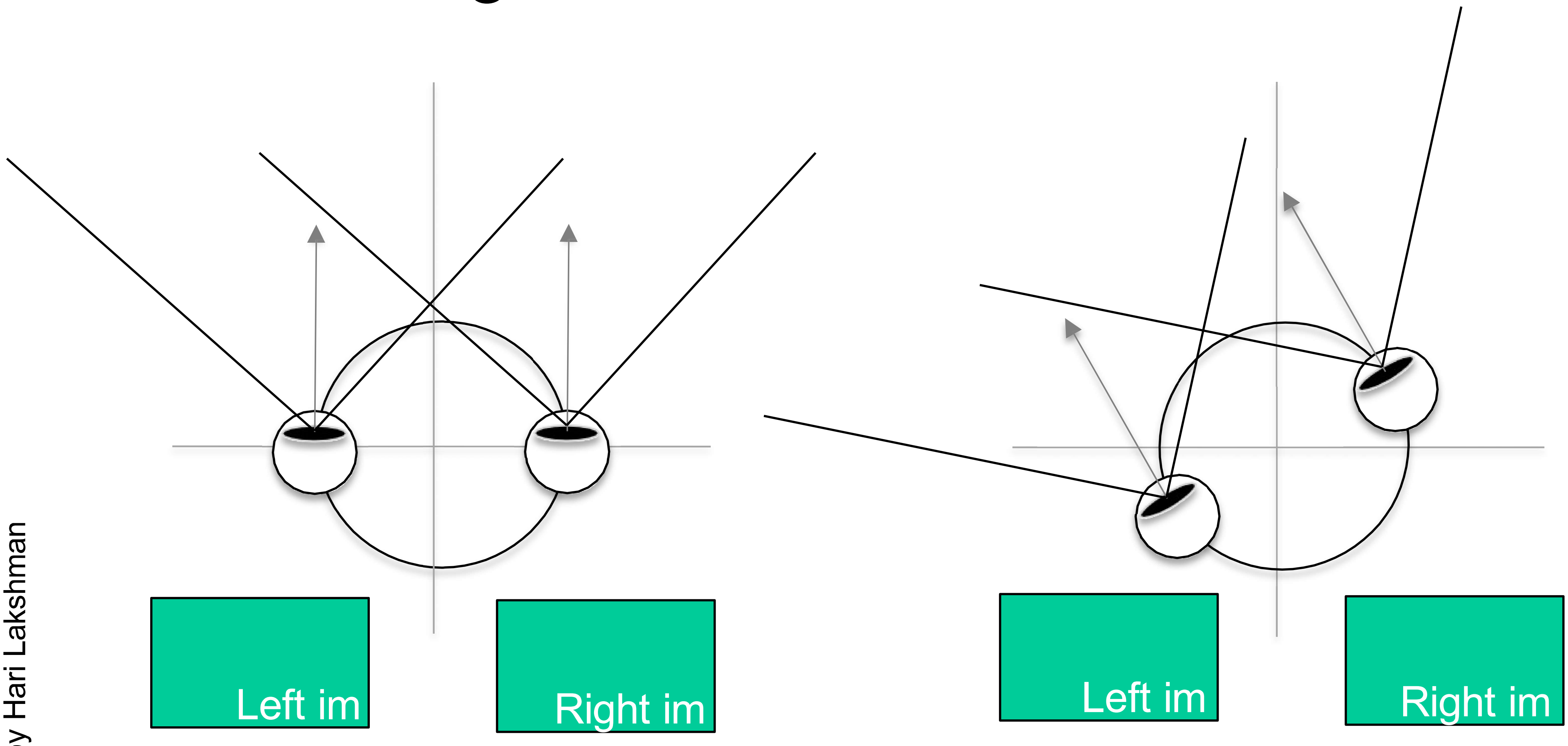
Head Rotation





side by Hari Lakshman

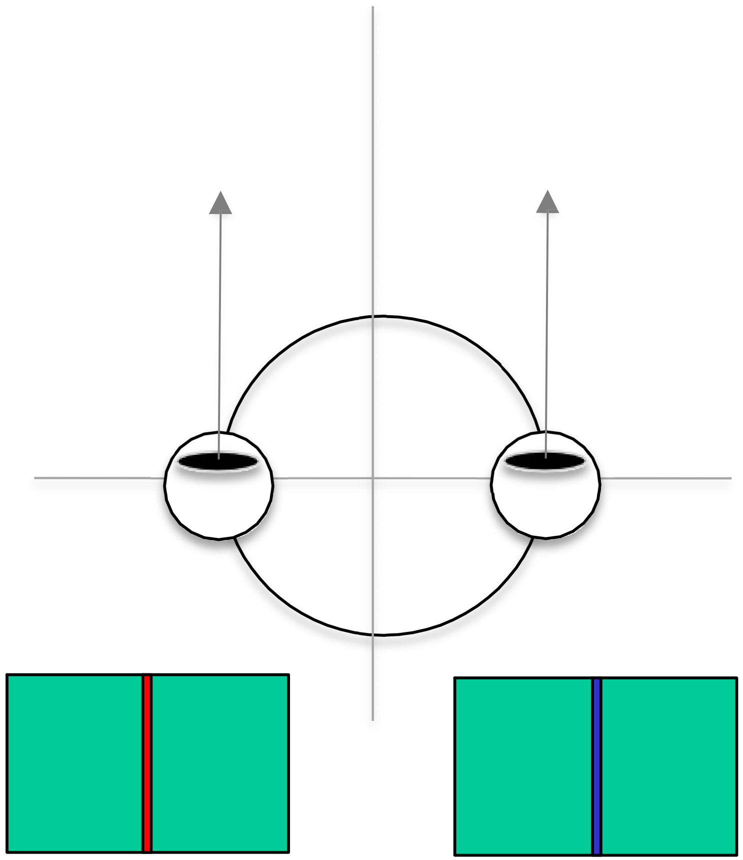
Image Pair for Each Direction



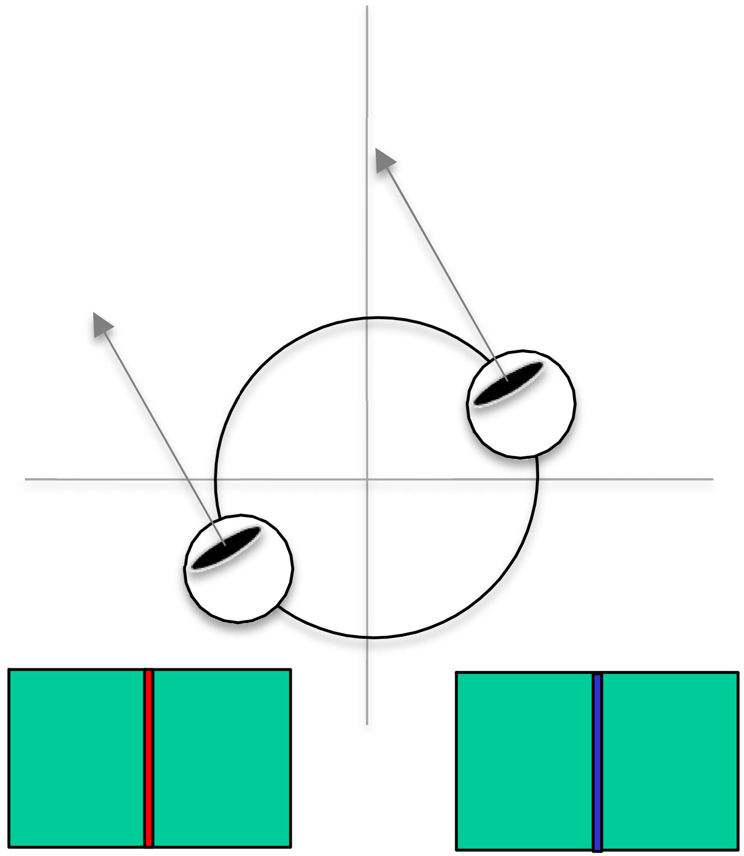
Store image pair for each direction Problem: Too much data

side by Hari Lakshman

Approximation: Store only Middle Ray



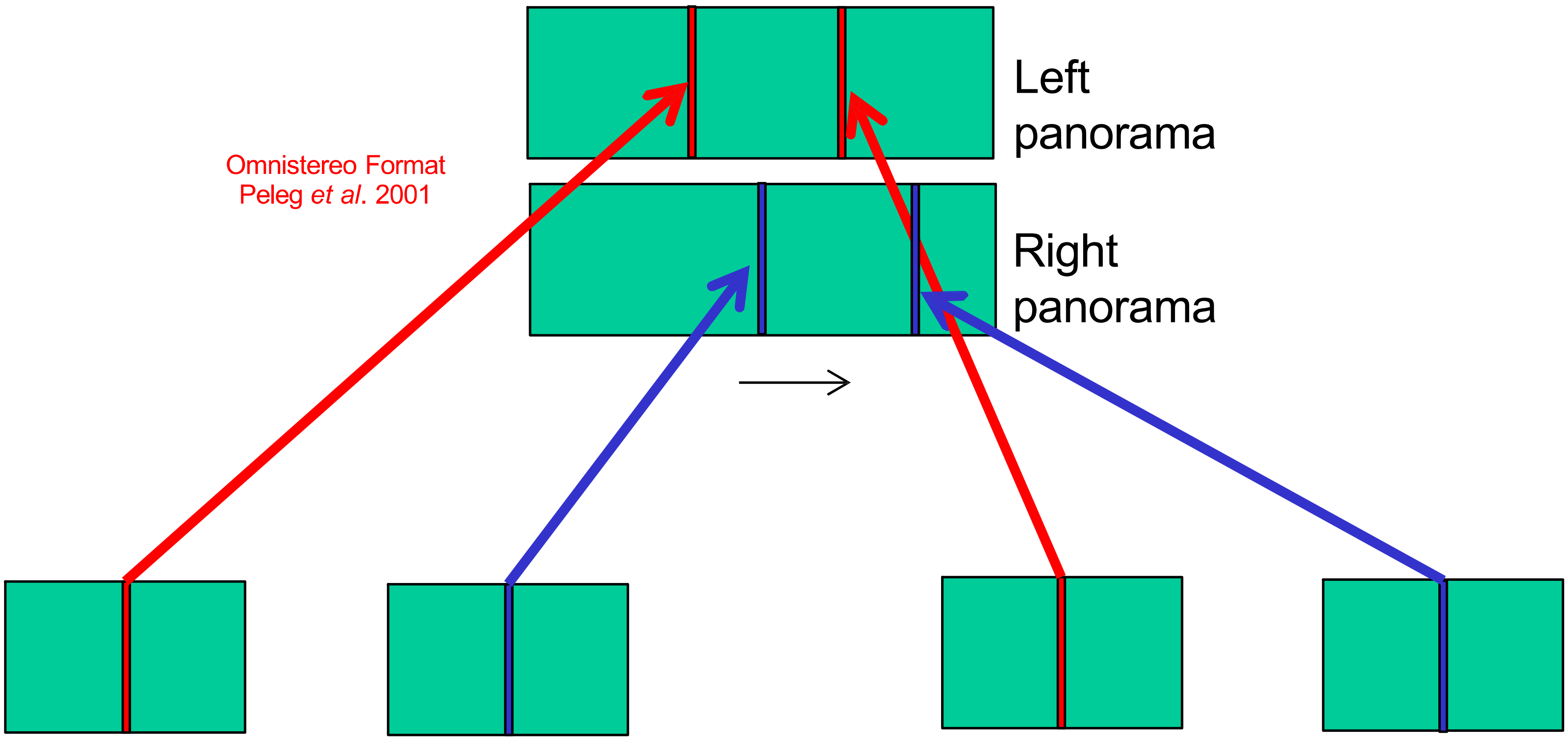
Omnistereo Format
Peleg *et al*. 2001



Approximation: store only middle ray for L and R eyes for each direction

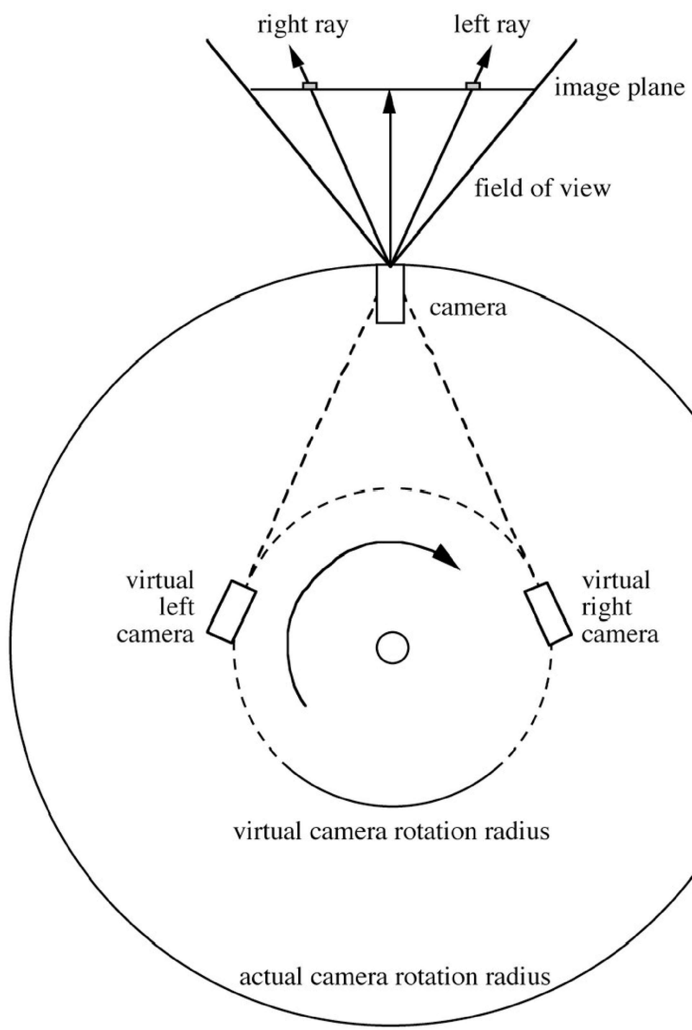
side by Hari Lakshman

Omnistereo Panoramas



side by Hari Lakshman

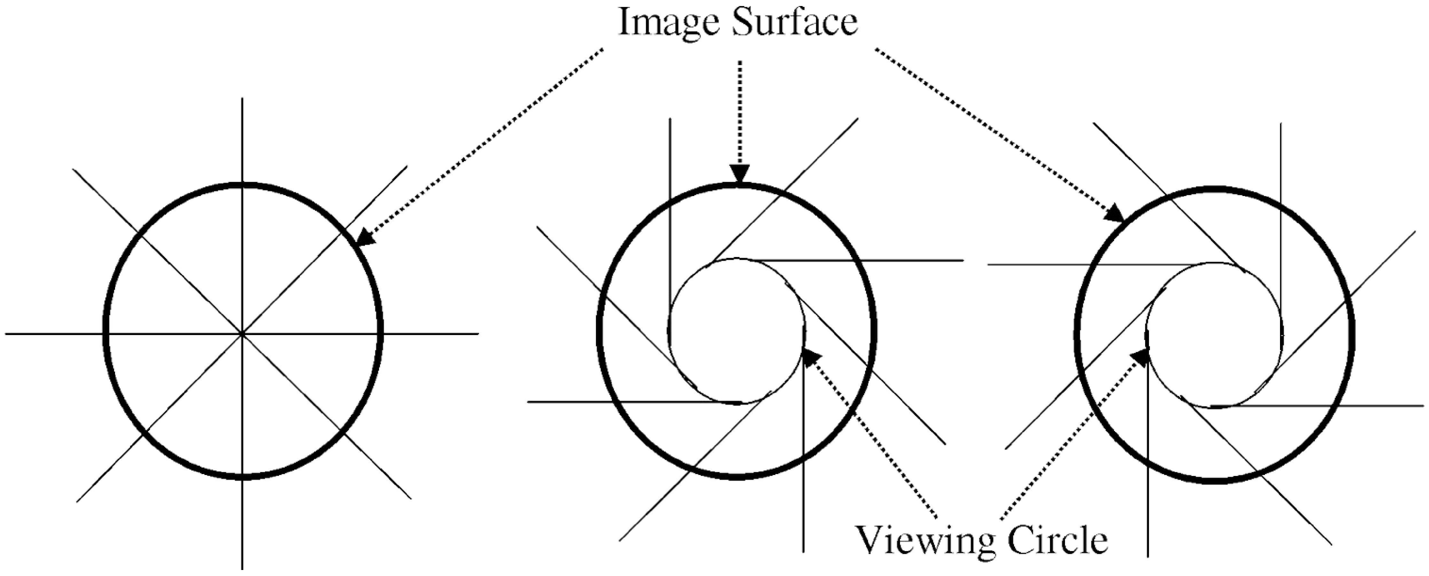
Capture using Single Camera



Peleg et al. 2001

side by Hari Lakshman

Comparison: Mono and Stereo Panoramas



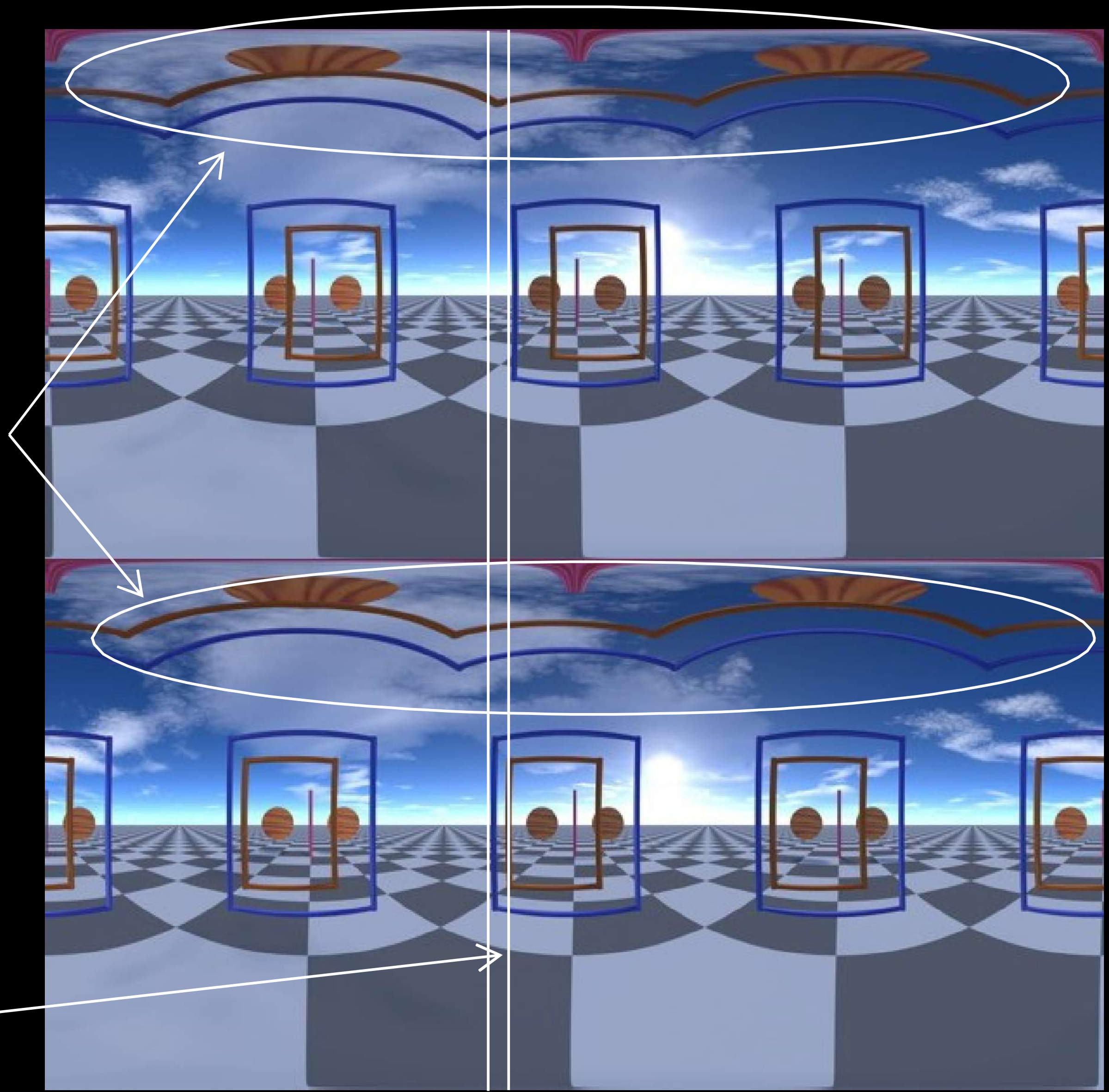
Central,
a.k.a. Mono

Omnistereo,
Multiperspective

Peleg et al. 2001

side by Hari Lakshman

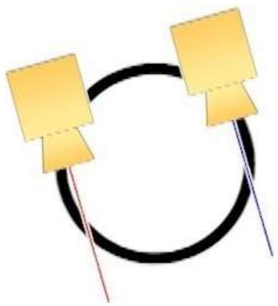
Omnistereo example



Multiperspective Projection

Jingyi Yu

Omnidirectional Stereo



Left Eye

Right Eye

widely used by YouTube VR, Google Daydream, Facebook, ...

Existing VR Cameras

Recorded Videos ~ 17 Gb/sec

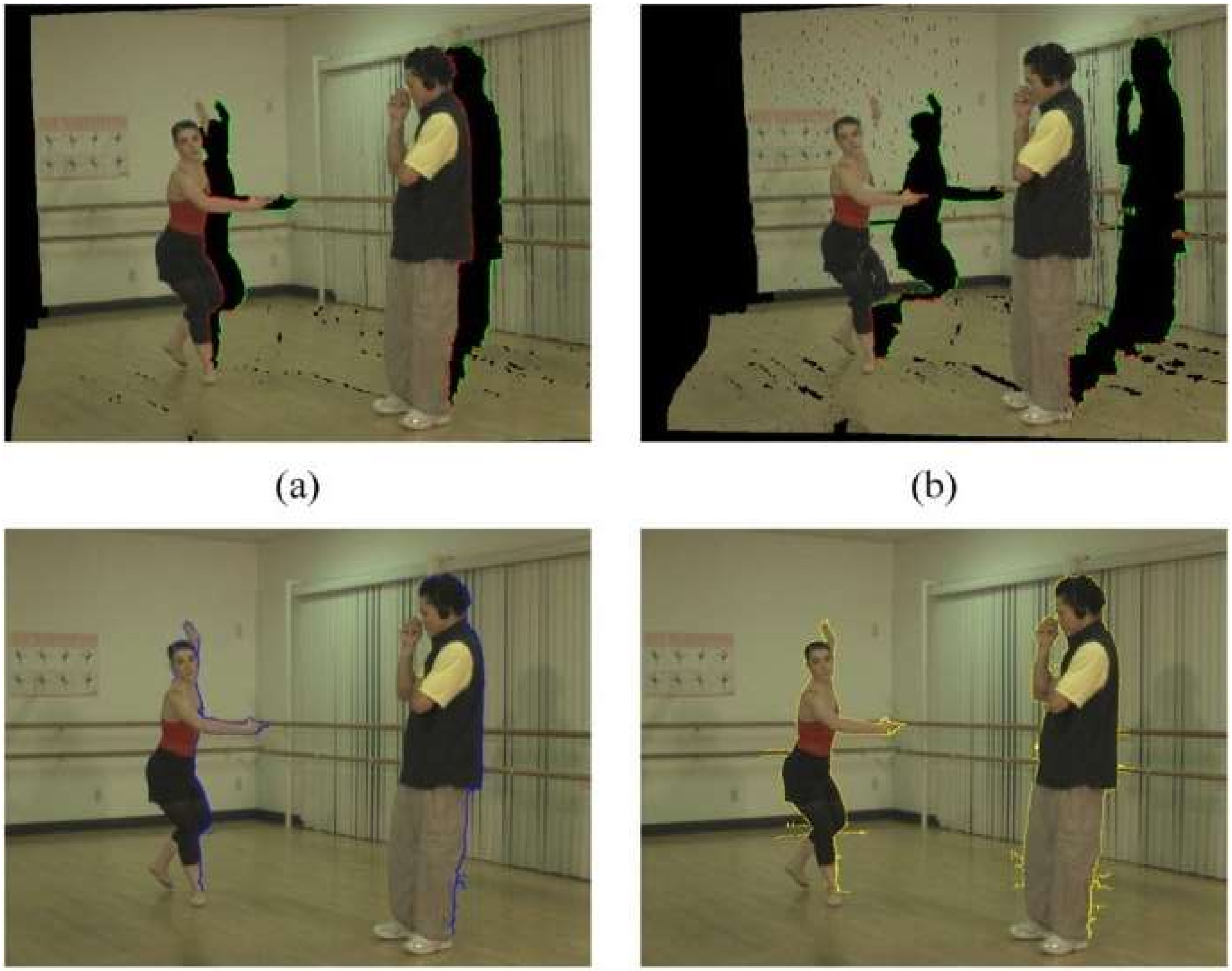
Facebook’s Surround 360

RAW Data: 17 Gb/sec

Compute time: days to weeks on conventional computer,
minutes to hours on data center

Facebook’s Surround 360
RAW Data: 17 Gb/sec
Compute time: days to weeks on conventional computer,
minutes to hours on data center

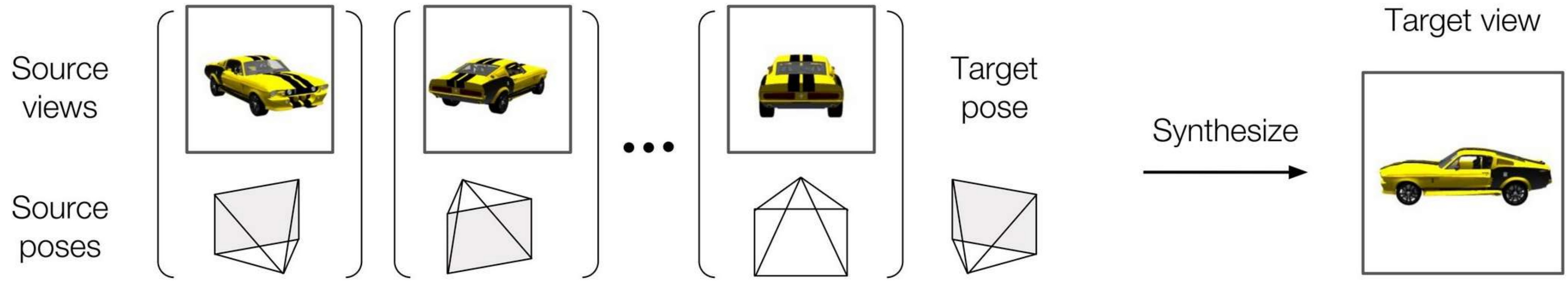
Biggest Problem of Panorama – 6DoF Viewing



(a) (b)

Problem: Novel View Synthesis

- The process of using images of a scene and their camera poses to synthesize new images of the scene at arbitrary camera poses.



- Subset of *Neural Rendering*, “deep image or video generation approaches that enable explicit or implicit control of scene properties”. (Ayush Tewari et. al.)
  - E.g. illumination, camera parameters, pose, geometry, appearance, and semantic structure

Image: “Multi-view to Novel View: Synthesizing Novel Views with Self-Learned Confidence” (Sun et al., 2018)

Applications in VR

Related Work

- Neural Volumes (NV), 2019
  - Deep 3D convolutional network architecture
  - Predicts a fixed-size discretized voxel grid
  - Limitation: discrete voxel grids do not scale well and lose fine detail at high resolutions
  - Limitation: requires a bounded volume and knowledge of the background
- Scene Representation Networks (SRN), 2019
  - Uses a recurrent neural network to model a rendering function
  - Limited to simple shapes with low geometric complexity
- Local Light Field Fusion (LLFF), 2019
  - 3D convolutional network architecture
  - Predicts multiplane images and fuses them to create new views
  - Fast to train (<10 minutes) at the cost of large storage requirements (~GB for each scene)

NeRF: Key Insights

- Represent static scenes in a **continuous** space.
- Encode a continuous radiance field **within the parameters** of a fully-connected neural network.
- Regress directly from viewing location and direction to color and transparency

Problem Setting

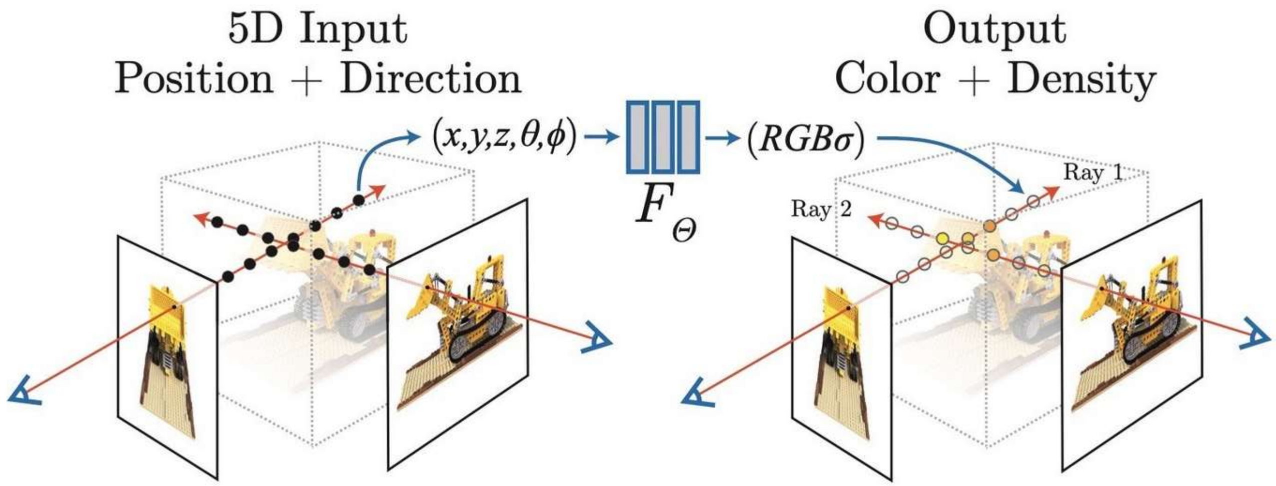
- Given a dataset containing RGB images of a static scene, their corresponding camera poses, and intrinsic parameters,
- Predict the color and volume density for every viewing location and direction

Inputs:

- x, y, z Target position of the camera
- θ, ϕ Target orientation of the camera

Outputs:

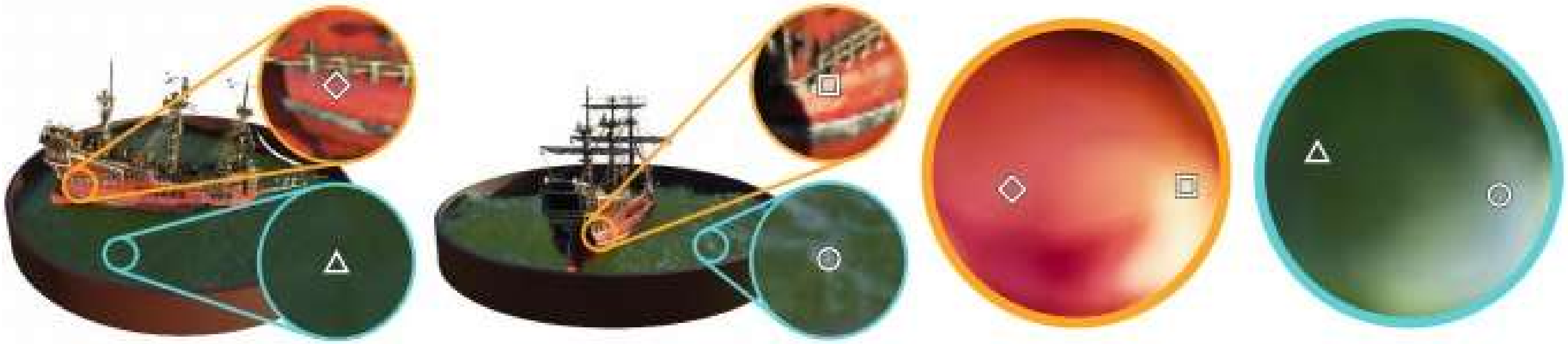
- c = (R, G, B) Color
- σ Volume density



NeRF: Inspired by Volume Rendering

Definition of radiance field

- Radiance field is a 5-dimensional function which maps a 3D location <u>**x,y,z**</u> and a direction in 3D sphere <u>**d**</u> to a color **(r,g,b):**



(a) View 1 (b) View 2 (c) Radiance Distributions

[8] Mildenhall20

NeRF: Volume Rendering

- Generating a view from NeRF requires rendering all rays that pass through each pixel of the desired virtual camera

$$C(\mathbf{r}) = \int_{t_n}^{t_f} T(t)\sigma(\mathbf{r}(t))\mathbf{c}(\mathbf{r}(t), \mathbf{d})dt\,, \text{ where } T(t) = \exp\left(-\int_{t_n}^{t} \sigma(\mathbf{r}(s))ds\right)$$

Expected color of a camera ray

Predicted Volume Density

Predicted Color

Probability that nothing has blocked the ray up to this point

- Numerically estimated using quadrature and stratified sampling

$$\hat{C}(\mathbf{r}) = \sum_{i=1}^{N} \underline{T_i(1 - \exp(-\sigma_i\delta_i))}\mathbf{c}_i\,, \text{ where } T_i = \exp\left(-\sum_{j=1}^{i-1} \sigma_j\delta_j\right)$$

The relative contribution of this segment

Predicted Color

Probability that nothing has blocked the ray up to this point

- Differentiable: allows optimization using gradient descent

NeRF: Hierarchical Sampling

- Problem: It is inefficient to integrate over empty and occluded spaces in a scene
- Solution: Allocate samples proportionally to their expected effect on the final rendering.

- Evaluate a “coarse” network on a set of N_c locations along a ray to produce a PDF along the ray

- Evaluate a “fine” network on N_c and a second a set of N_f locations sampled from the PDF

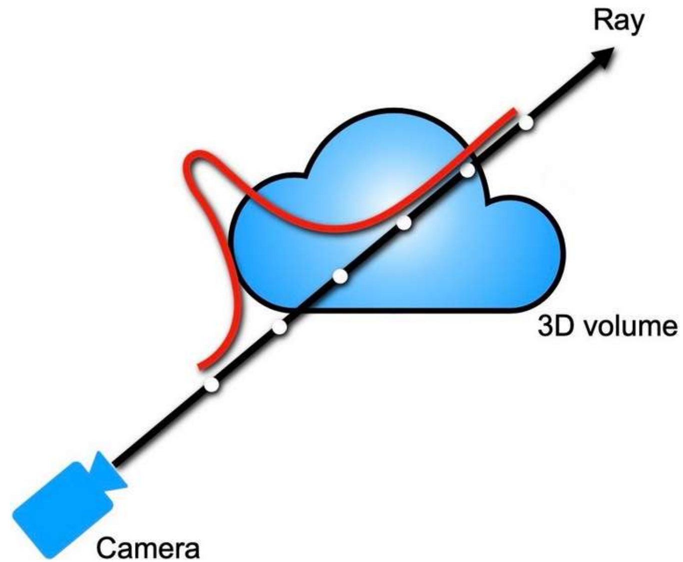


Image: Jonathan Barron

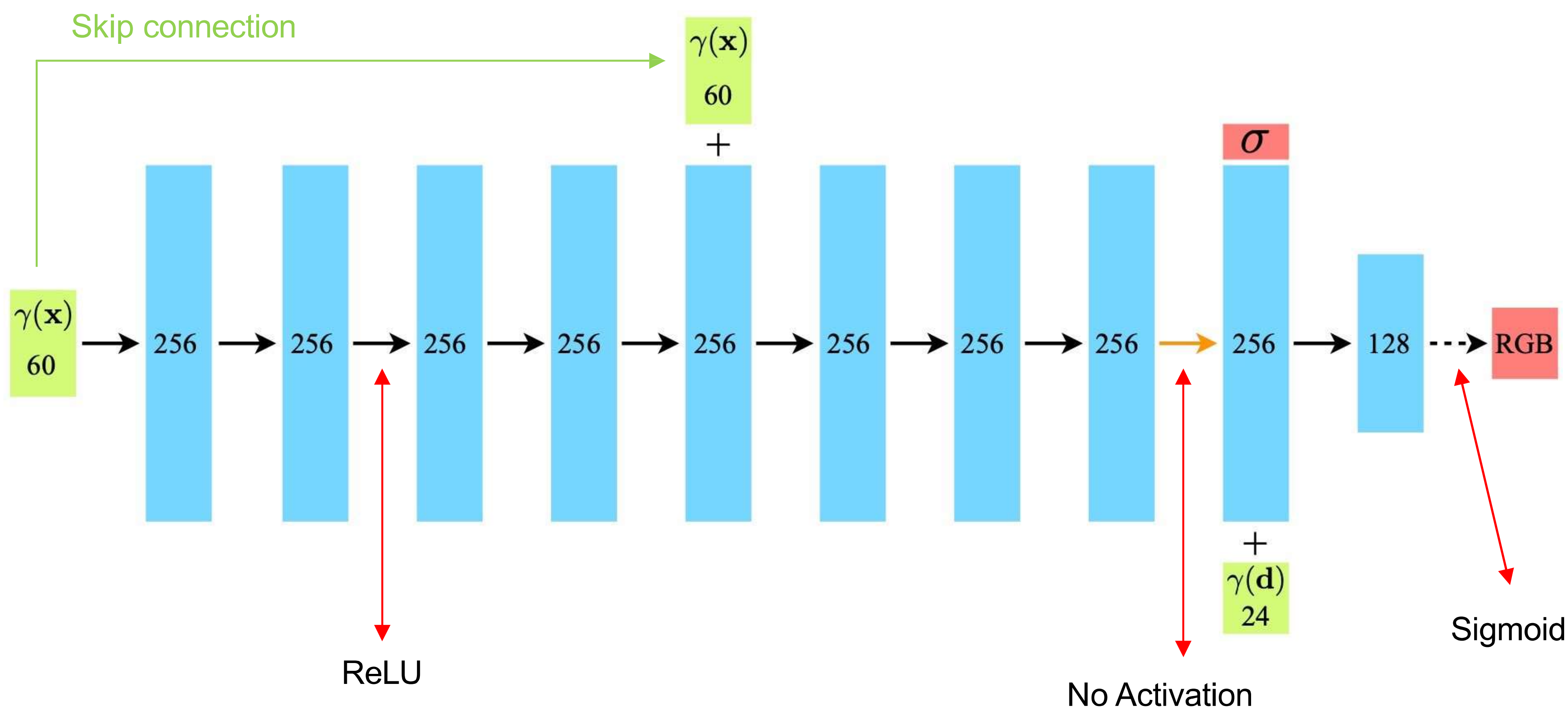
NeRF: Positional Encoding

- Map individual components of position and direction vectors to a higher dimensional space

$$\gamma(p) = \left(\sin\left(2^0 \pi p\right), \cos\left(2^0 \pi p\right), \cdots, \sin\left(2^{L-1} \pi p\right), \cos\left(2^{L-1} \pi p\right) \right)$$

- Similar concept as positional encoding in Transformer Architectures
- Empirically improves the preservation of high-frequency geometry and texture
- Surprising result, explored further in a follow-up work by the same authors
  - "Fourier Features Let Networks Learn High Frequency Functions in Low Dimensional Domains" (Tancik, Srinivasan, Mildenhall, et al., 2020)

Network Architecture



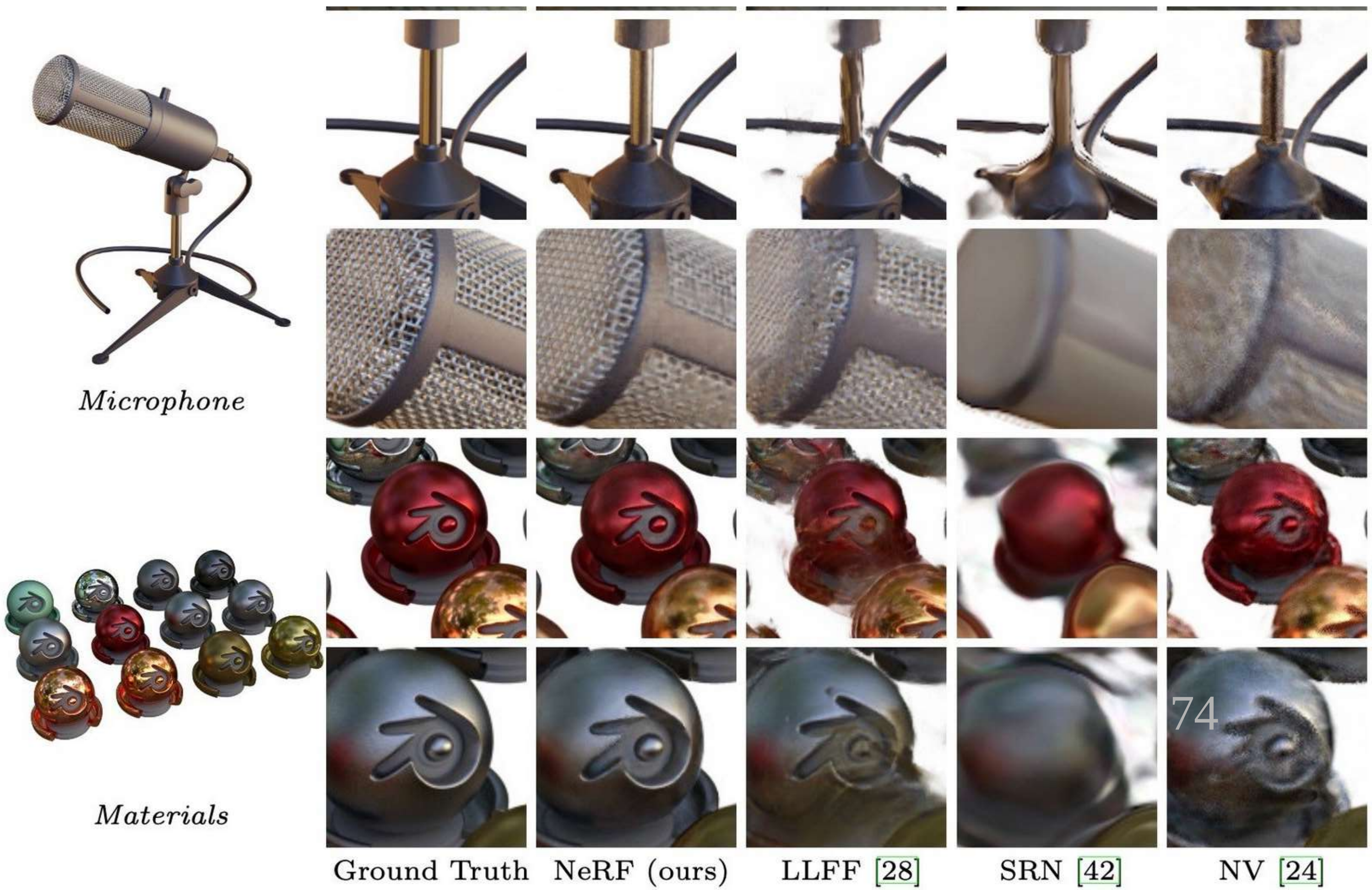
Training Summary

- Sample a batch of camera rays from the dataset (bs=4096)
- Use hierarchical sampling to query coarse and fine points
- Use the volume rendering equation to calculate the color of the ray
- Compute the squared error between rendered and true pixel colors

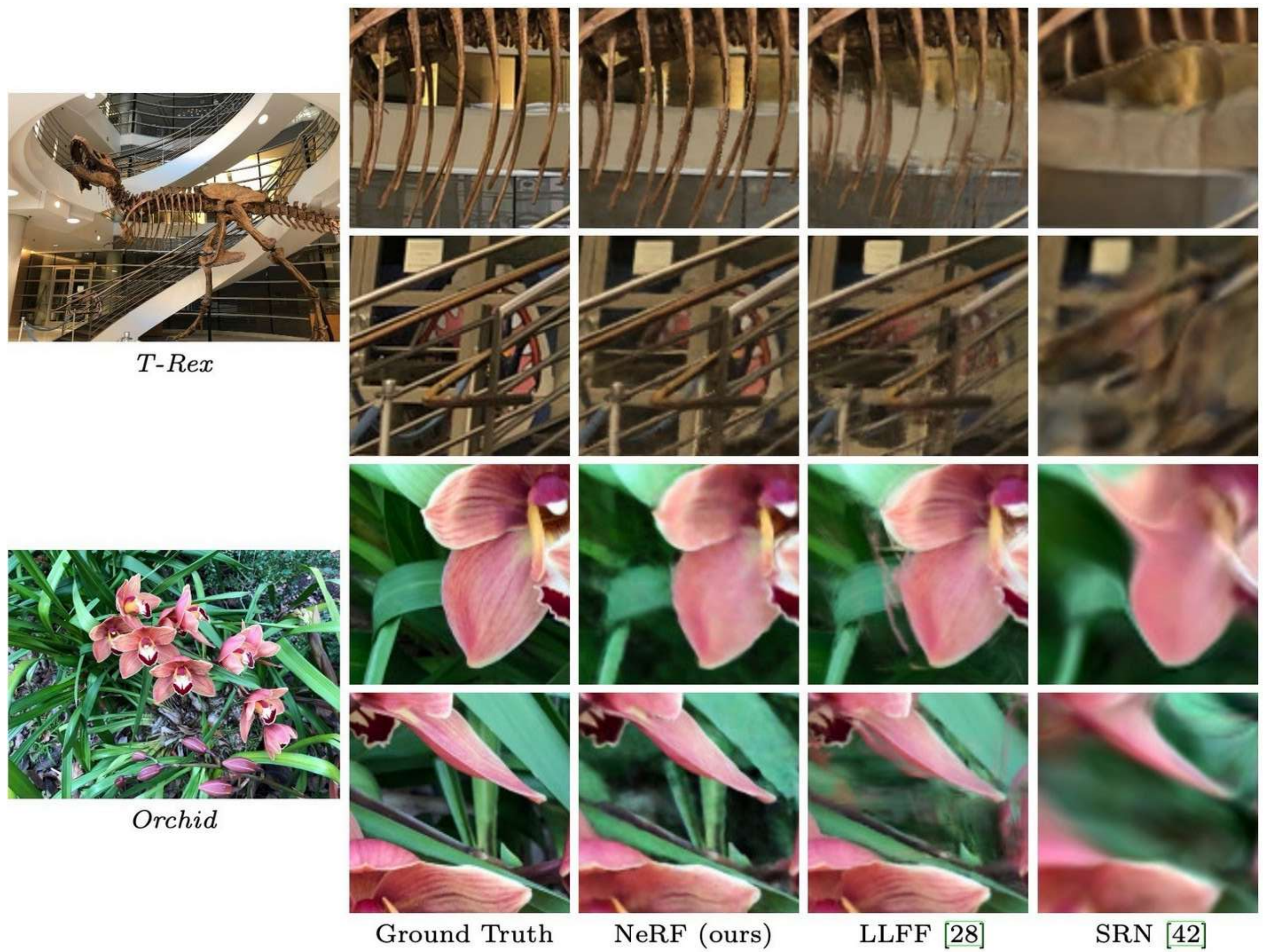
$$\mathcal{L} = \sum_{\mathbf{r} \in \mathcal{R}} \left[\left\| \hat{C}_c(\mathbf{r}) - C(\mathbf{r}) \right\|_2^2 + \left\| \hat{C}_f(\mathbf{r}) - C(\mathbf{r}) \right\|_2^2 \right]$$

- Optimize network parameters using Adam

Qualitative Results: Realistic Synthetic Objects



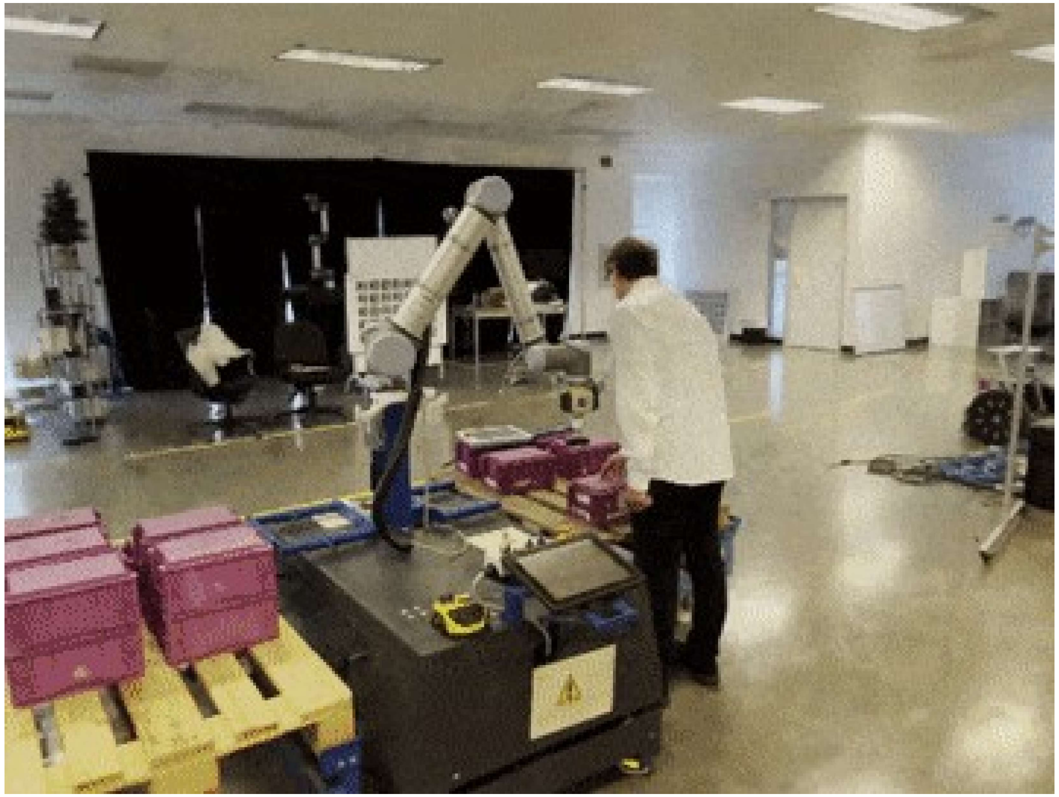
Qualitative Results: Real-world scenes



Animated Results

Video: https://www.matthewtancik.com/nerf

Instant Neural Graphics Primitives



https://github.com/NVlabs/instant-ngp

Limitations

- NeRF only works with static scenes
- A trained NeRF model does not generalize to more than one scene
- Computationally expensive: 1-2 days to train each individual scene on a modern GPU
- Inference is slow: each pixel in a synthesized image requires volume rendering

Summary

- Novel view synthesis generates images of scenes at previously unseen viewpoints.
- Prior works are limited to simple shapes and do not scale well to high-resolution images
- NeRF encodes a static scene within the parameters of a feedforward neural network.
- The authors show very impressive qualitative results and show state-of-the-art performance with quantitative metrics and different scene types.

D-NeRF: Neural Radiance Fields for Dynamic Scenes

NeRF

- Only applicable to rigid scenes
- 5D continuous function
- Requiring multiple views of a rigid scene

D-NeRF

+ Applicable for rigid and non-rigid scenes
+ 6D continuous function by considering time-component as an additional input
+ Requiring a single view per time instant for non-rigid scenes.

[12] Pumarola20

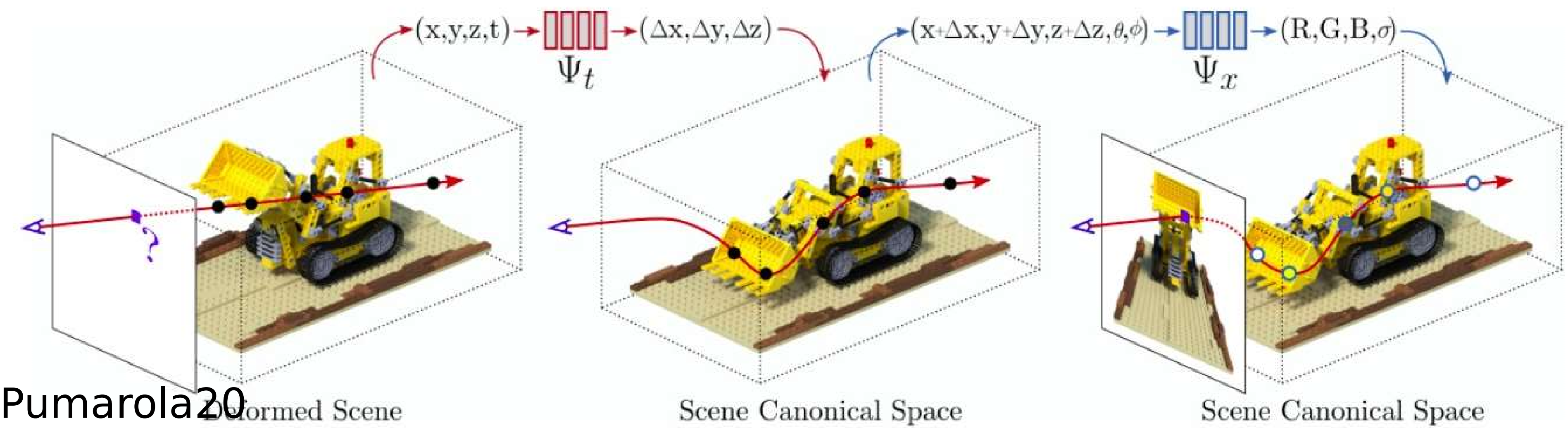
D-NeRF: Neural Radiance Fields for Dynamic Scenes

- **Deformation network** Ψ_t : to predict deformation field between the scene at time instant t and the scene in canonical space (t=0)

$$\Psi_t(\mathbf{x}, t) = \begin{cases} \Delta\mathbf{x}, & \text{if } t \neq 0 \\ 0, & \text{if } t = 0 \end{cases}$$

- **Canonical network** Ψ_x : to predict color and density in

$$\Psi_x(\mathbf{x}, \mathbf{d}) \mapsto (\mathbf{c}, \sigma)$$



[12] Pumarola20

D-NeRF: Neural Radiance Fields for Dynamic Scenes

Volumetric rendering is the same as NeRF in **canonical space**:

$$\text{6D radiance field} \xrightarrow{\textbf{Deformation network } \Psi_t} \text{5D radiance field } \textit{(canonical space)} \xrightarrow{\textbf{NeRF's rendering}} \text{Rendering image}$$

$$C(p,t) = \int_{h_n}^{h_f} \underbrace{\mathcal{T}(h,t)}_{\text{Opacity}} \underbrace{\sigma(\mathbf{p}(h,t))}_{\text{Volume density}} \underbrace{\mathbf{c}(\mathbf{p}(h,t),\mathbf{d})}_{\text{Predicted colors}} dh$$

$$\text{where} \quad \mathbf{p}(h,t) = \mathbf{x}(h) + \Psi_t(\mathbf{x}(h),t),$$

$$[\mathbf{c}(\mathbf{p}(h,t),\mathbf{d}), \sigma(\mathbf{p}(h,t))] = \Psi_x(\mathbf{p}(h,t),\mathbf{d}),$$

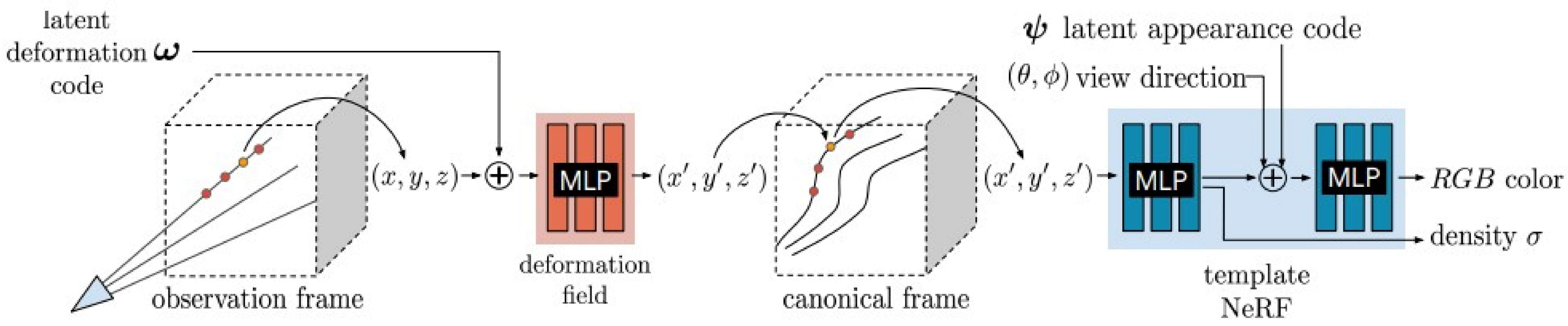
$$\text{and} \quad \mathcal{T}(h,t) = \exp\left(-\int_{h_n}^{h} \sigma(\mathbf{p}(s,t))ds\right).$$

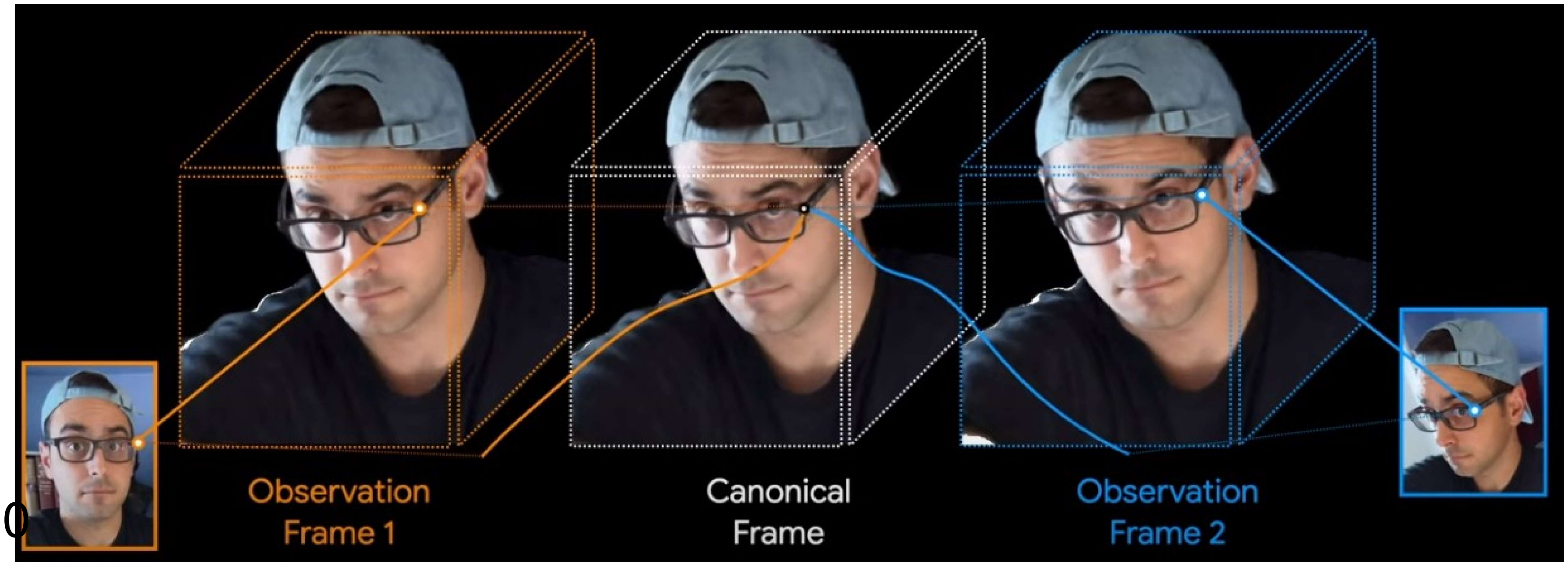
[12] Pumarola20

D-NeRF: Neural Radiance Fields for Dynamic Scenes

[12] Pumarola20

Deformable Neural Radiance Fields

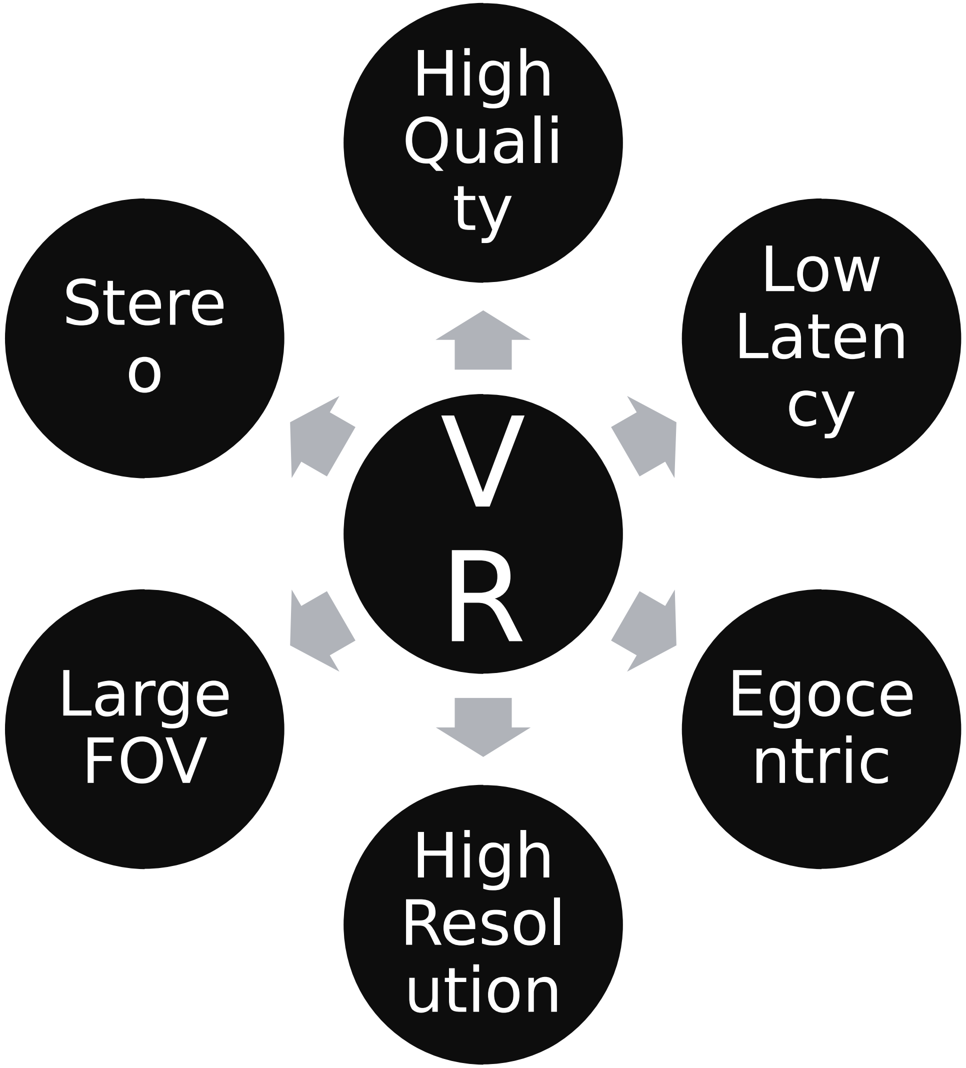




[13] Park20

Deformable Neural Radiance Fields

[13] Park20


V
R
High
Quali
ty
Low
Laten
cy
Egoce
ntric
High
Resol
ution
Large
FOV
Stere
o

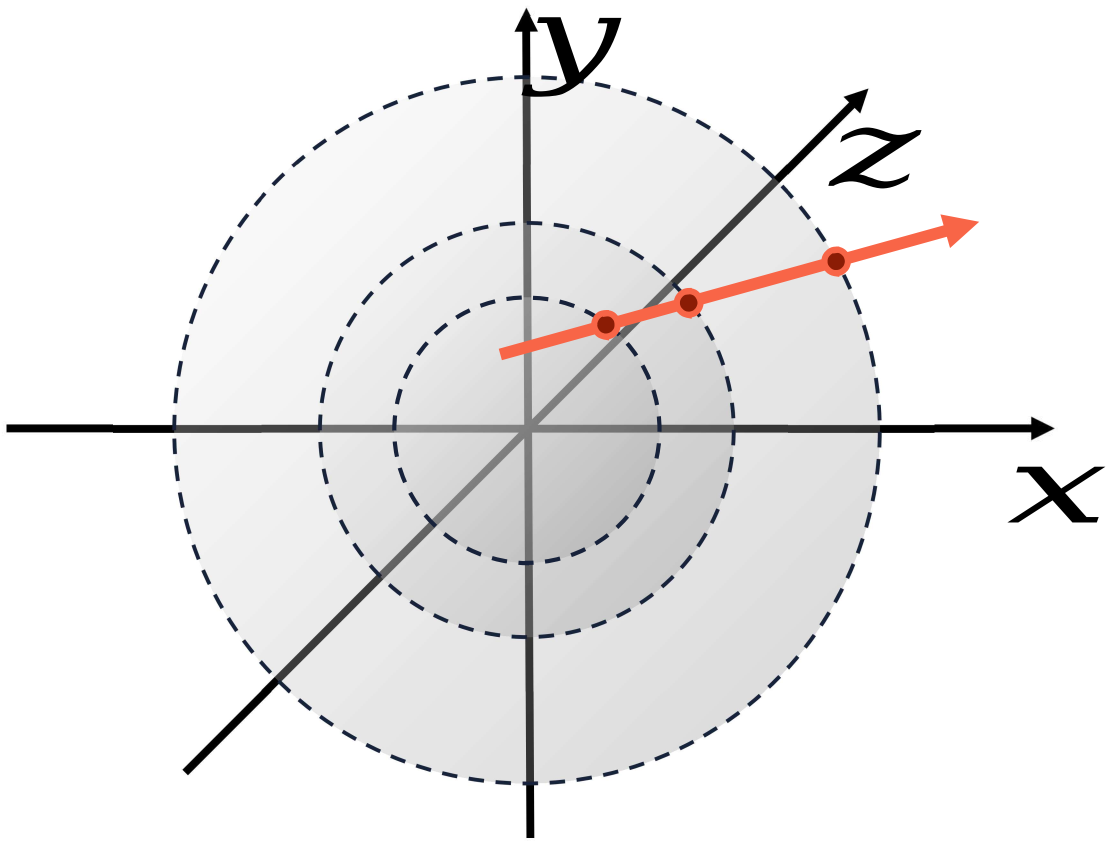
NeRF		VR
High Latency		High Resolution
Low Quality		Stereo
☹		Egocentric

Gaze-Contingent Neural Rendering for VR

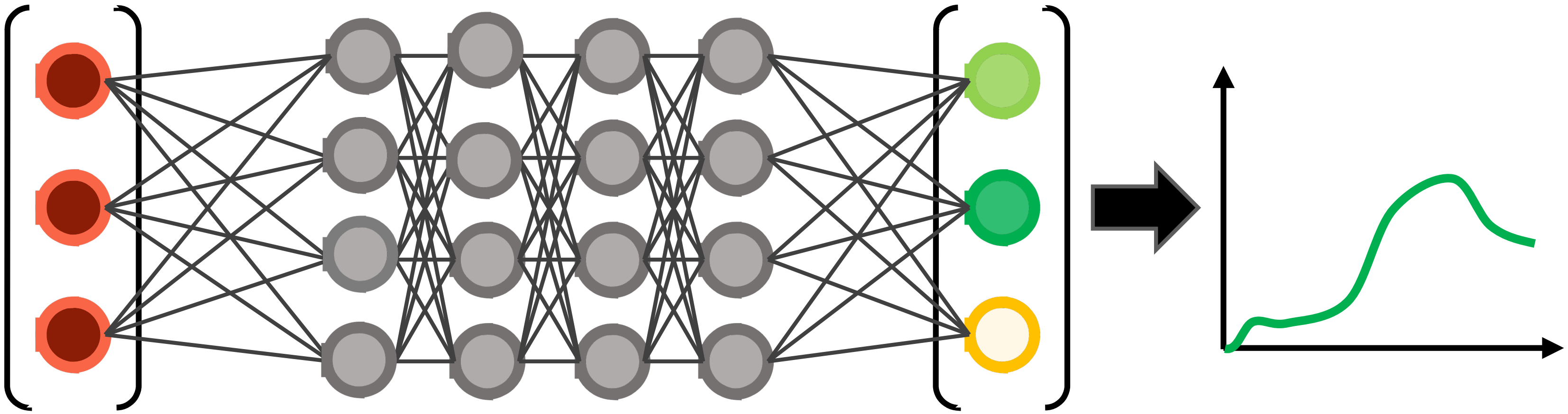
- Egocentric neural representation
- Adaptive perceptual (visual & stereo)
- Latency-Quality Joint Optimization


x90 Speed
50 FPS
NeRF
Ours

#1 Egocentric Neural Representation



#1 Egocentric Neural Representation

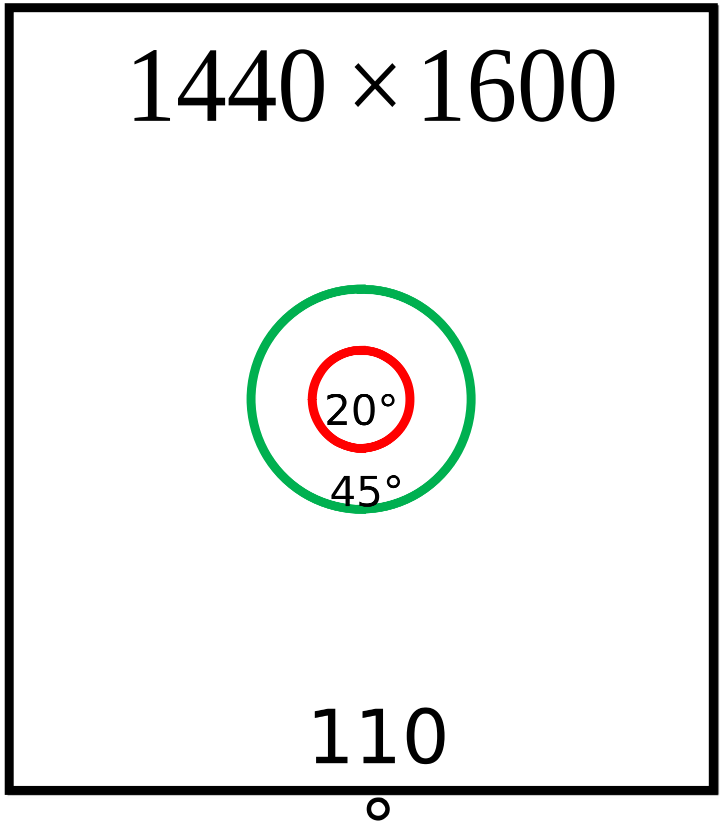


Higher quality

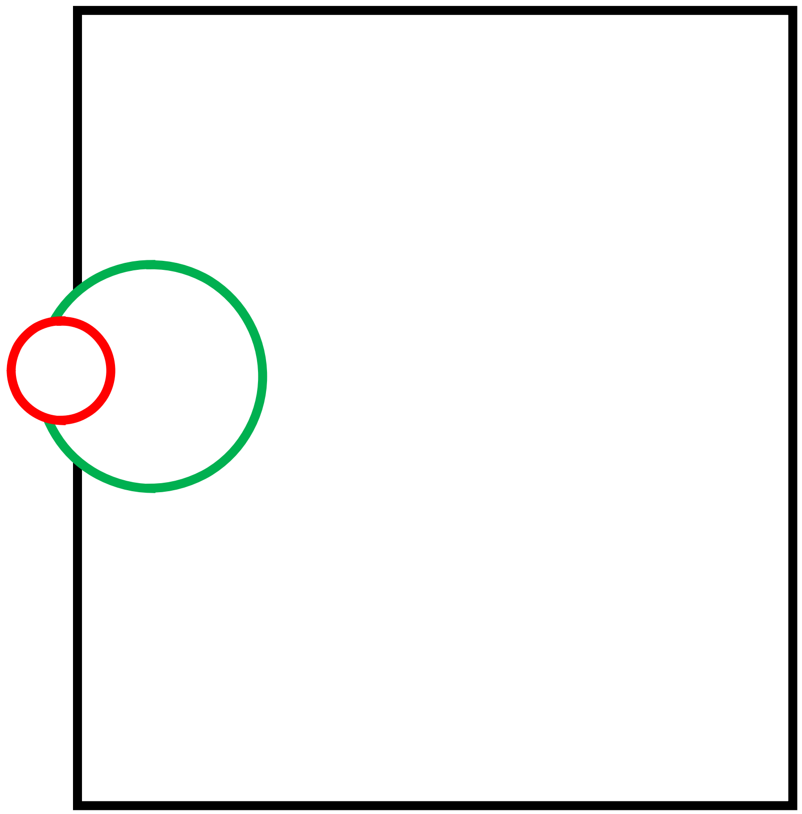
Faster

#2. Adaptive Visual Acuity

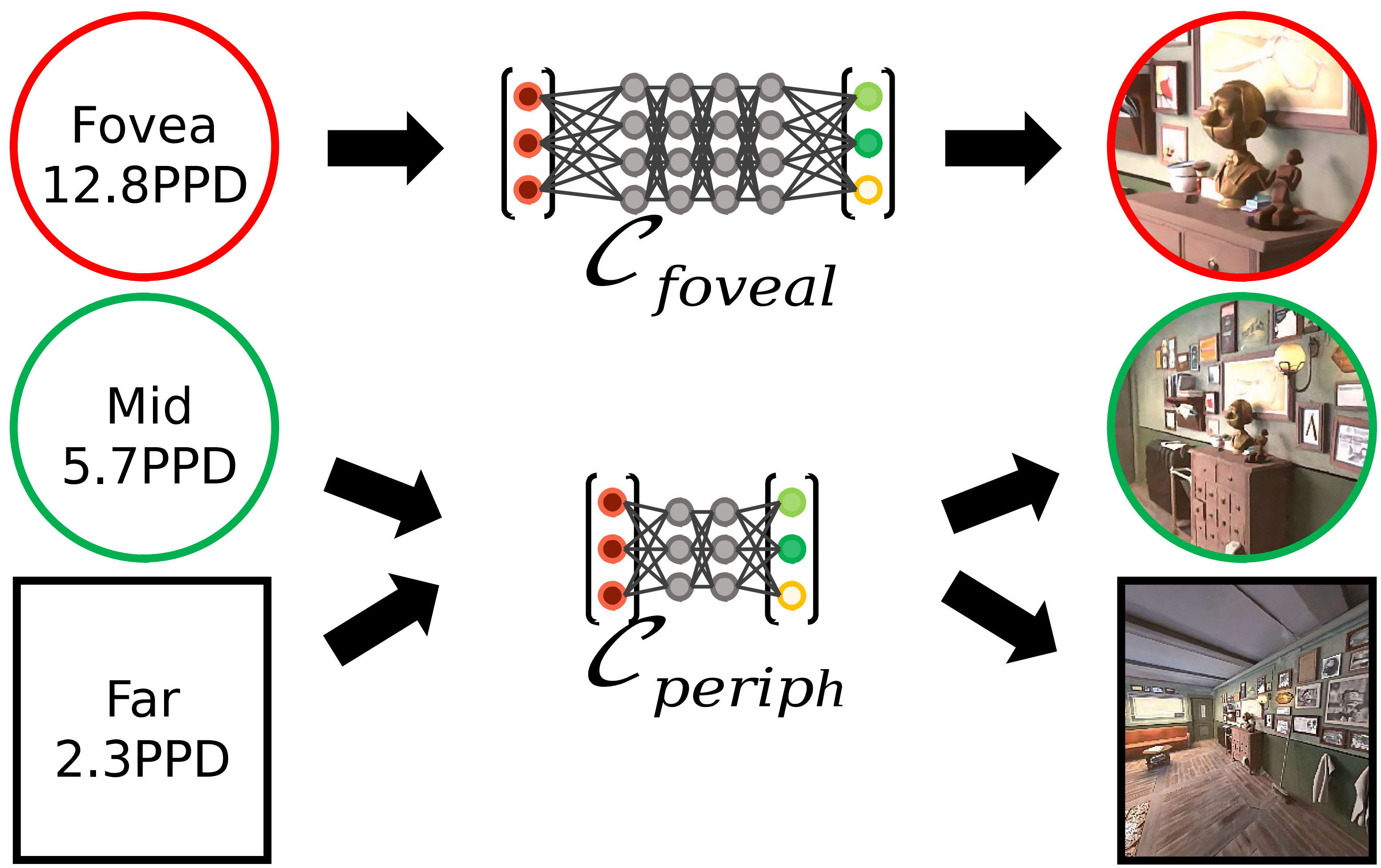
#2. Adaptive Visual Acuity



#2 Adaptive Visual Acuity



#2 Adaptive Visual Acuity



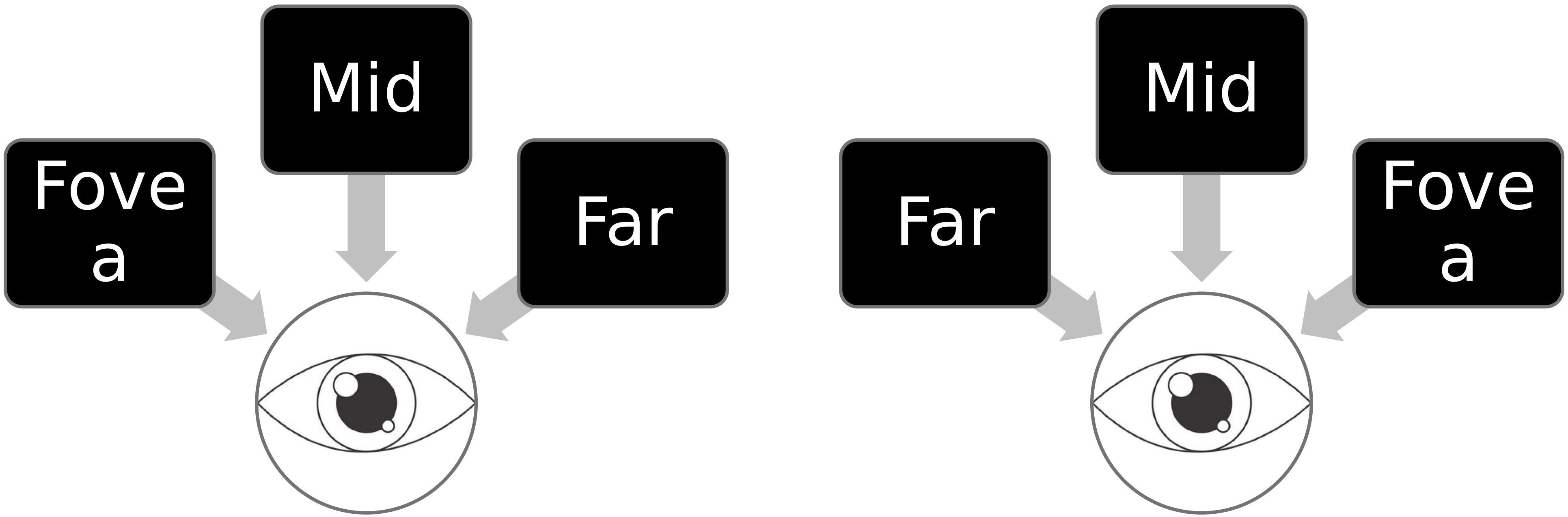
#2 Adaptive Visual Acuity

Save

~98%

computational cost

#3 Adaptive Stereoscopic Acuity



Naïve

#3 Adaptive Stereoscopic Acuity

25% faster

Adaptive

#4 Latency-Quality Joint Optimization

Hyper parameters:

: # of networks

: # of layers per network

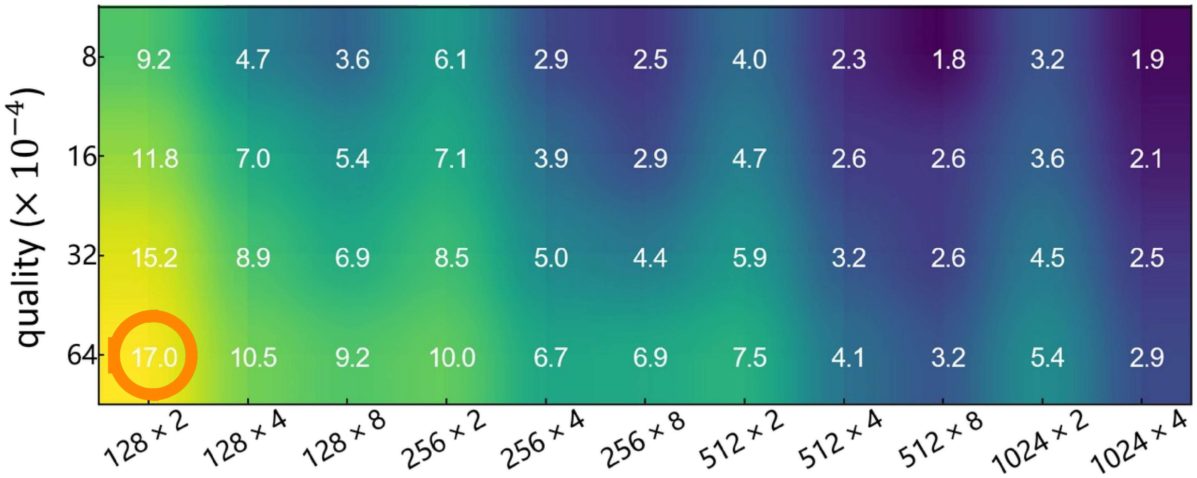
: # of channels per layer

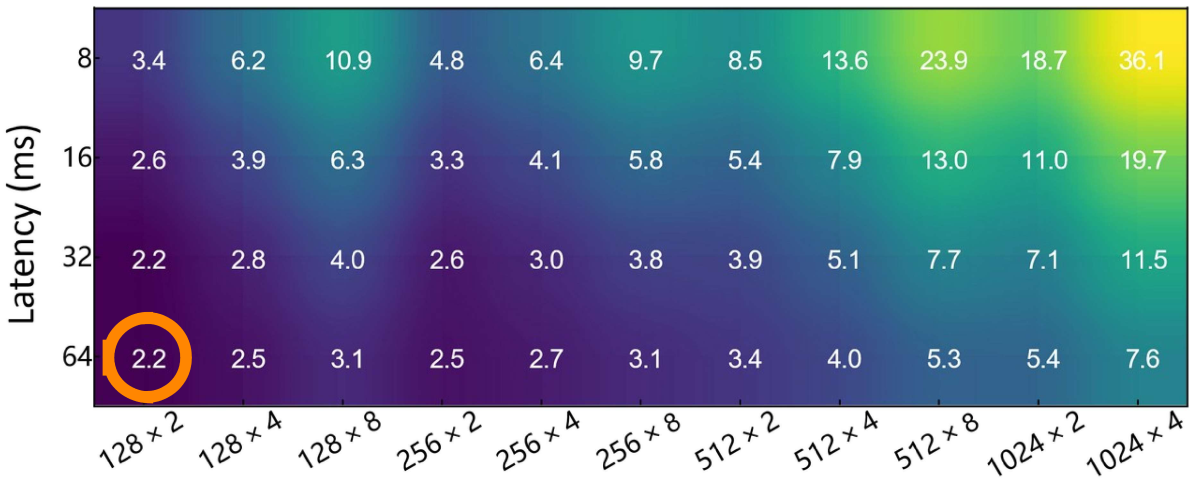
Spatial-temporal modeling:

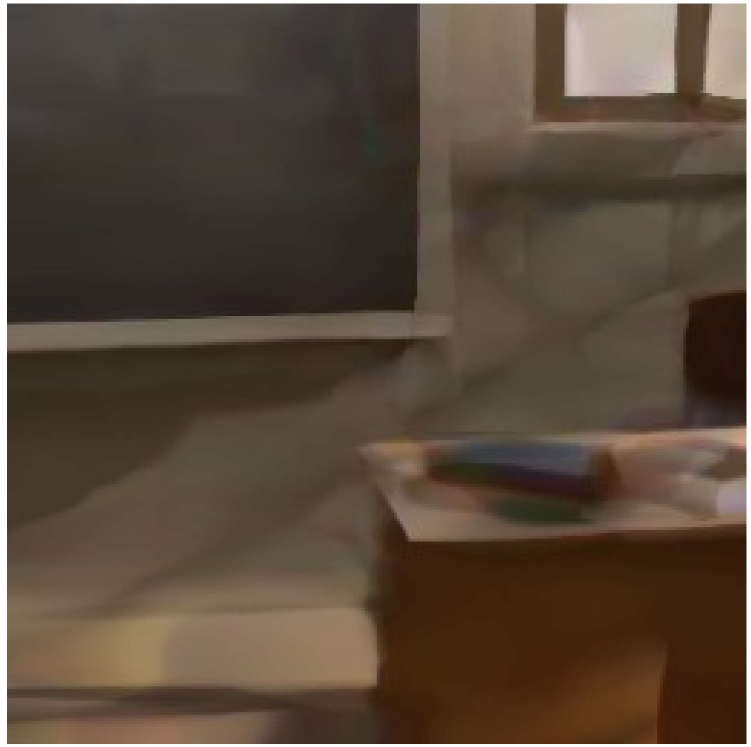
Objective function:

$$\underset{N, N_m, N_c}{argmin}\, E\left(N, N_m, N_c\right), s.t.\, l\left(N, \boldsymbol{r}\right) < L_0$$

#4 Latency-Quality Joint Optimization

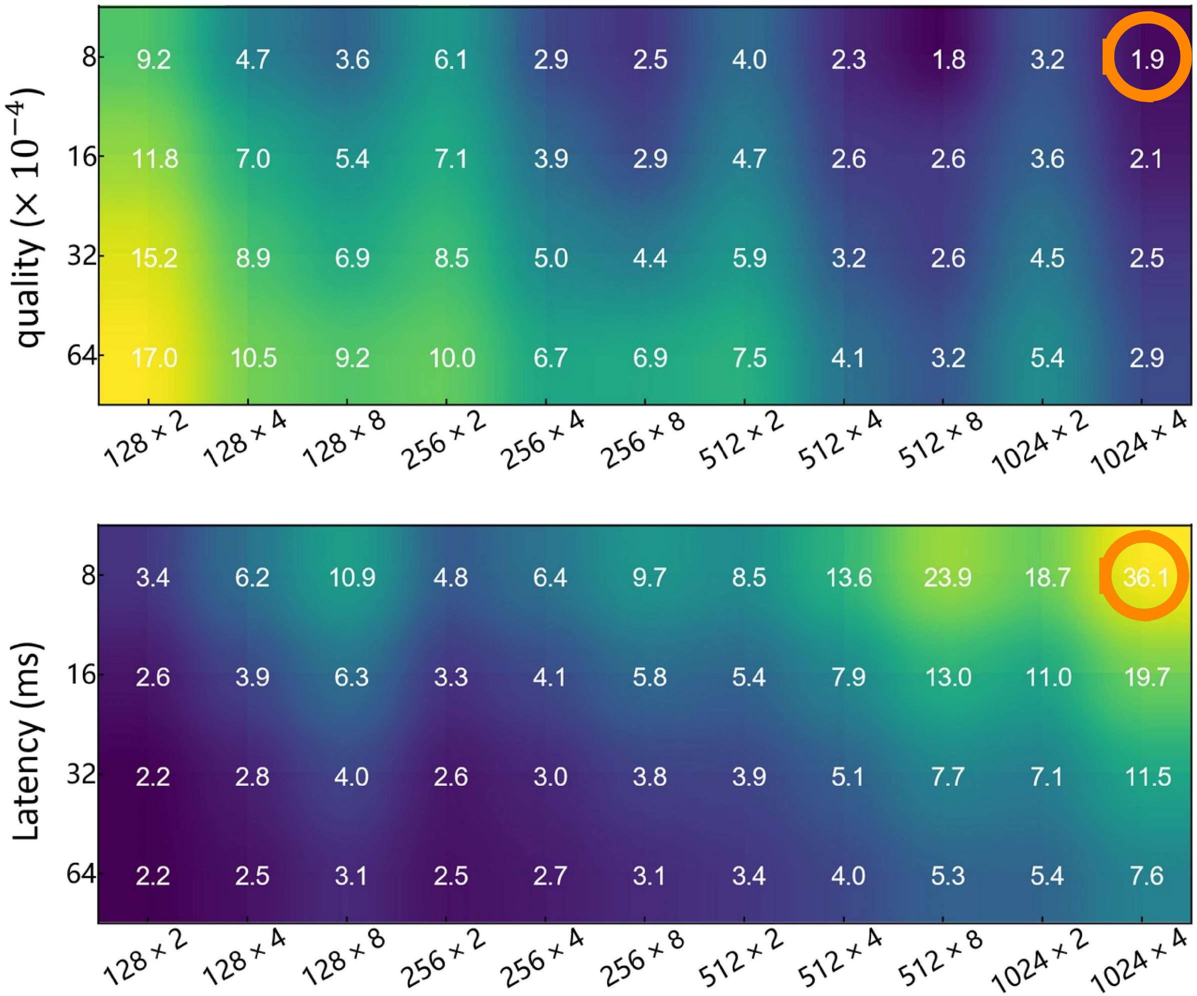


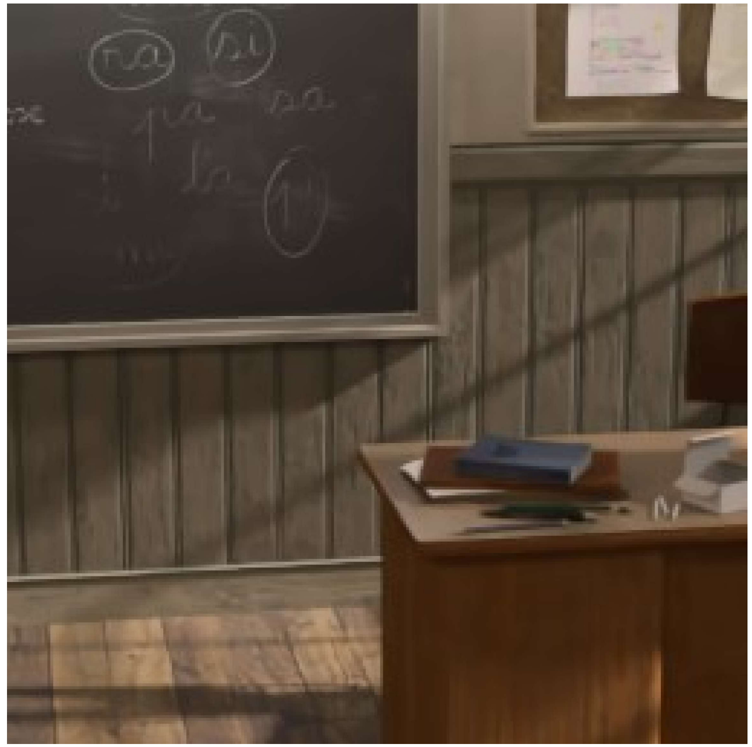




Best Performance

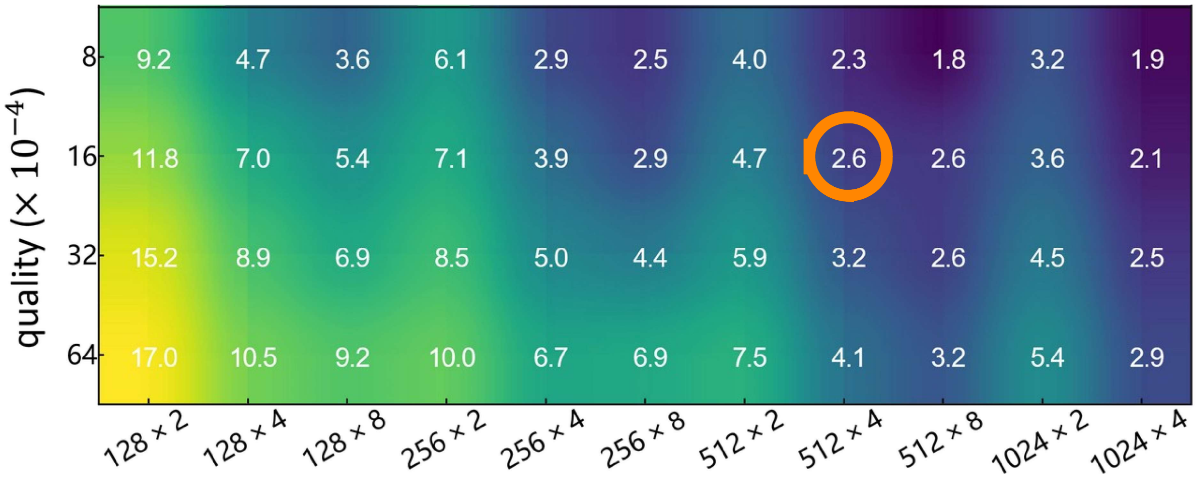
#4 Latency-Quality Joint Optimization

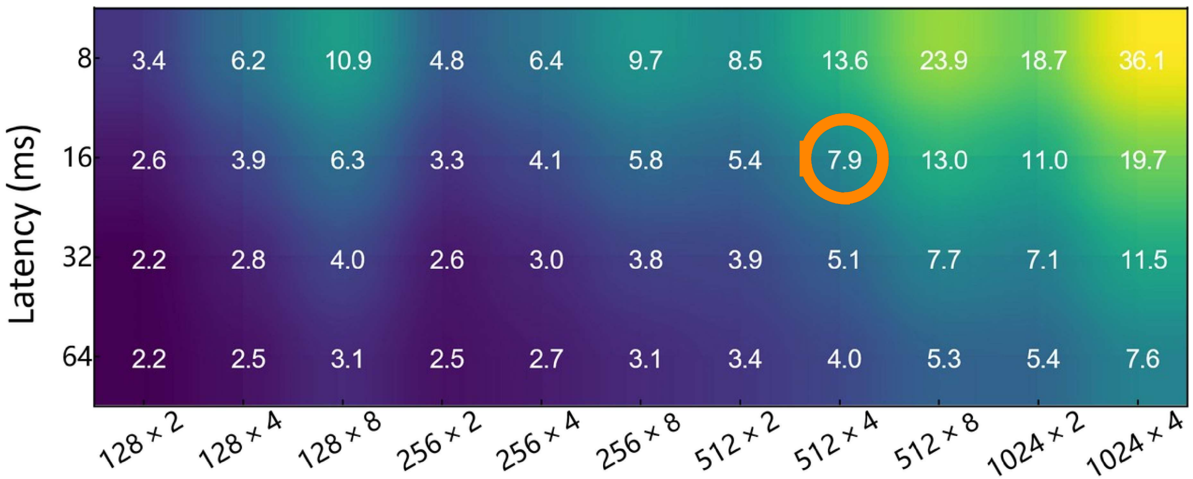


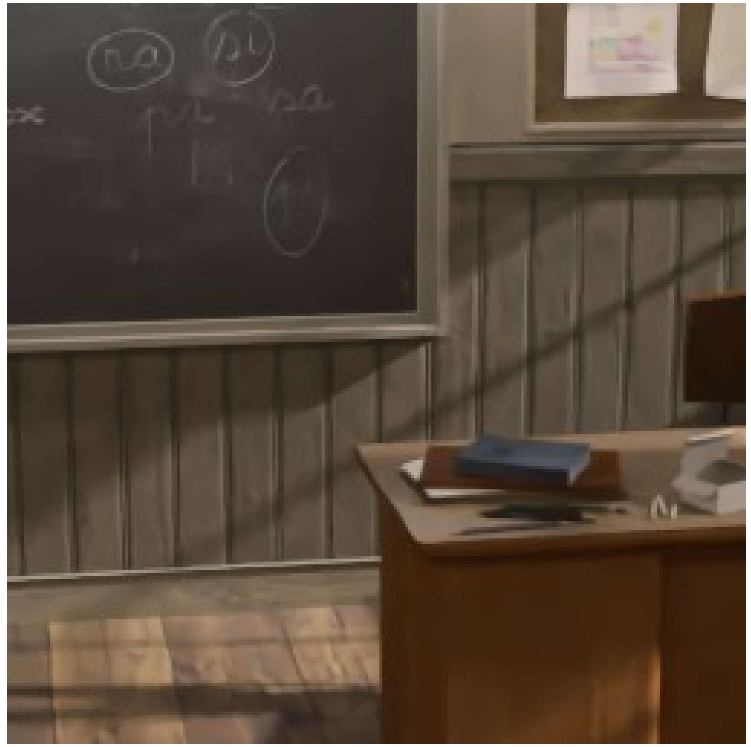


Best Quality

#4 Latency-Quality Joint Optimization

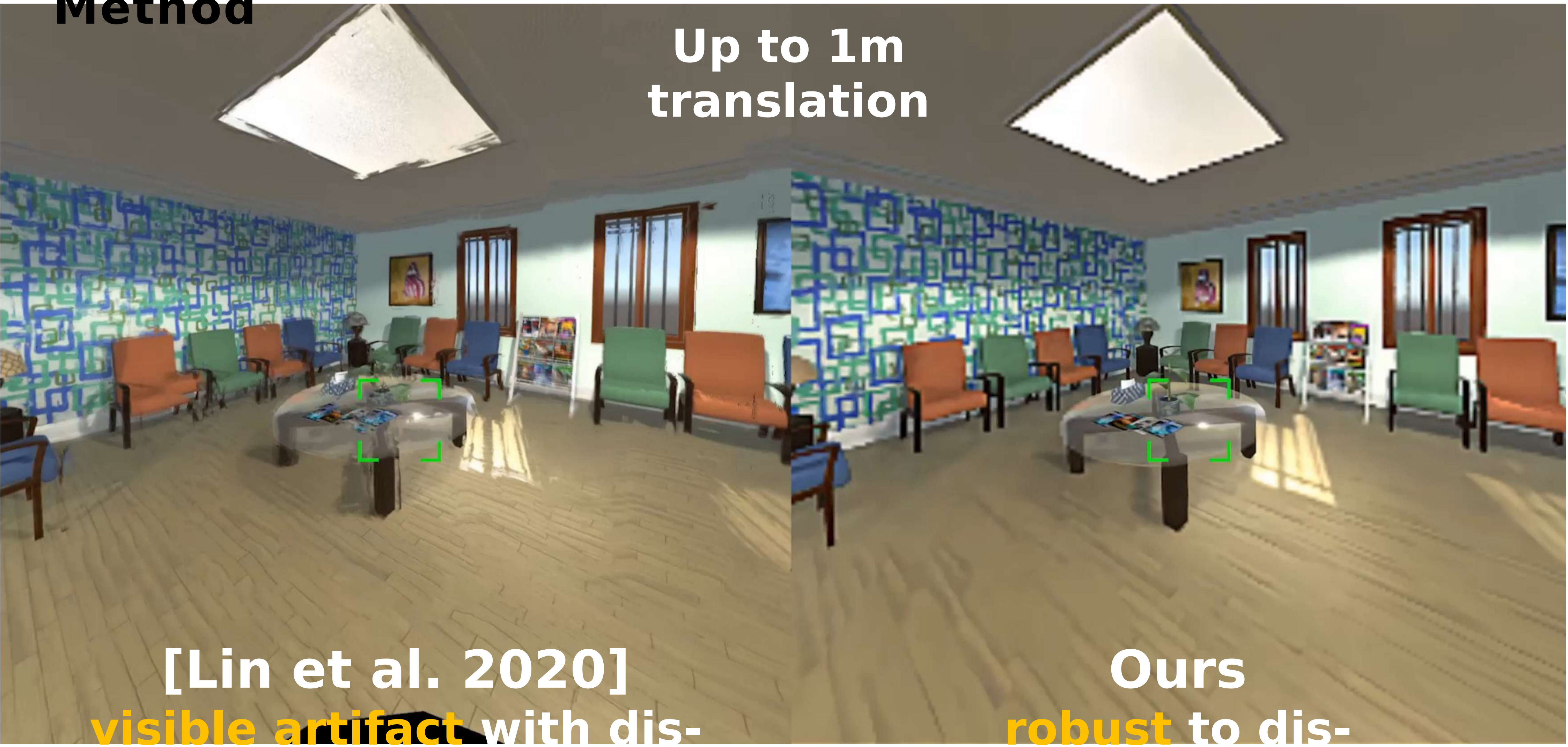






Ours

Comparison with Panorama-Based View Synthesis Method



Explicit vs Implicit Representation (2D)

Explicit:

$$\mathbf{f}(\alpha) = (r\cos(\alpha), r\sin(\alpha))^T$$

Domain: $[0,2\pi]$

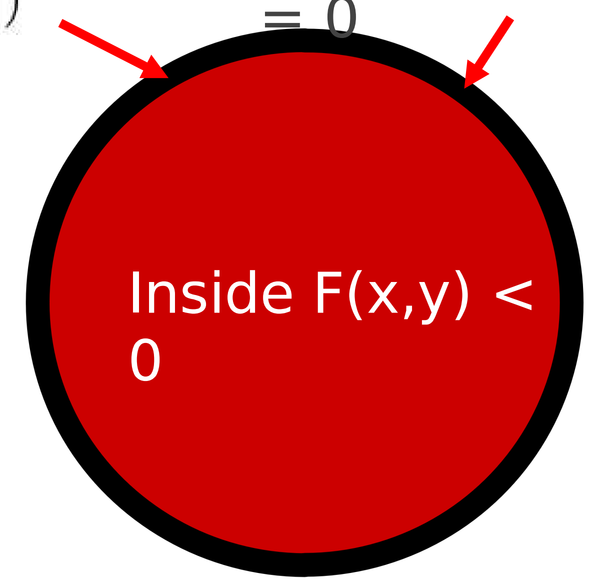
Implicit:

$$F(x,y) = \sqrt{x^2+y^2} - r$$

Domain: (x,y) $\in \mathbb{R}^2$

$\Longrightarrow$ Circle is implicitly defined by $\{(x,y) | F(x,y) = 0\}$

$\mathbf{f}([0,2\pi])$



$\mathbf{f}(\alpha)$ defines the interface

$F(x,y)$ defines the **Signed Distance Function** of the circle

Explicit vs Implicit Representation (3D)

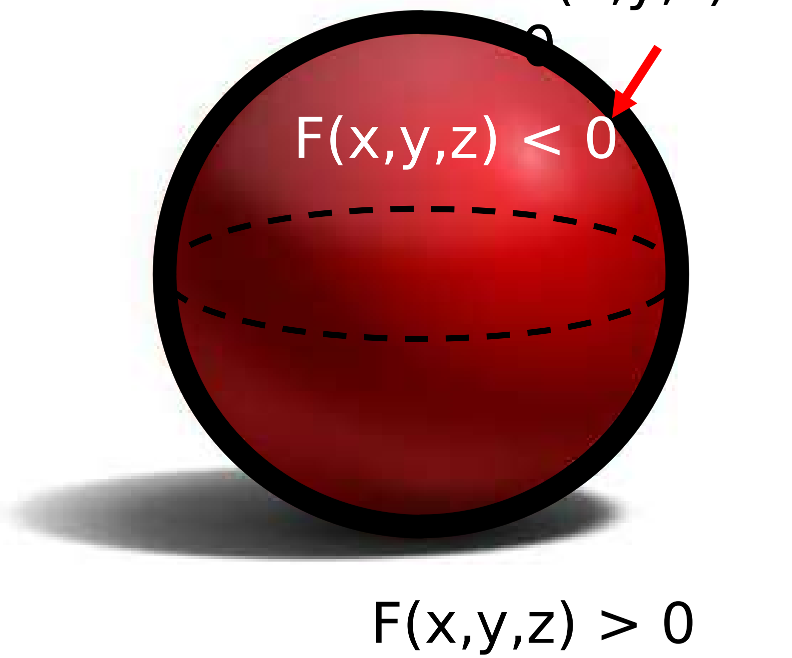
Explicit:
$$\mathbf{f}(\alpha, \beta) = (r\sin(\alpha)\cos(\beta), -r\cos(\beta), r\sin(\alpha)\sin(\beta))$$

Domain: $\alpha \in [0; 2\pi], \beta \in [0; \pi]$

Implicit:
$$F(x, y, z) = \sqrt{x^2 + y^2 + z^2} - r$$

Domain: (x,y,z) $\in \mathbb{R}^3$

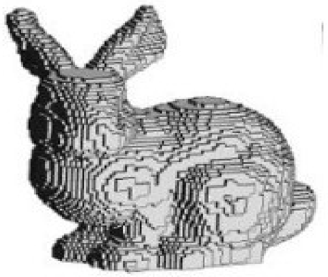
$\Longrightarrow$ Sphere is implicitly defined by $\{(x, y, z) | F(x, y, z) = 0\}$



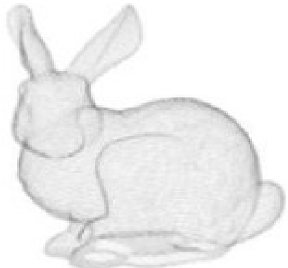
$\mathbf{f}(\alpha, \beta)$ defines the 3D surface
$F(x, y, z)$ defines the **Signed Distance Function** of the sphere

Representing 3D surfaces

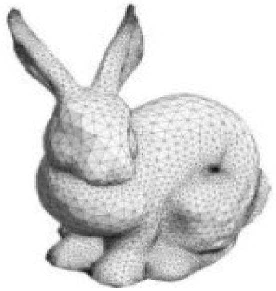
Explici
t:



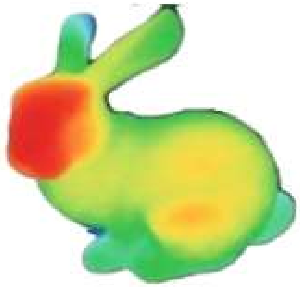
Voxels



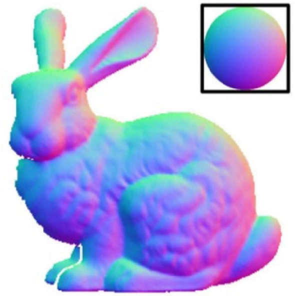
Point
clouds



Mes
h

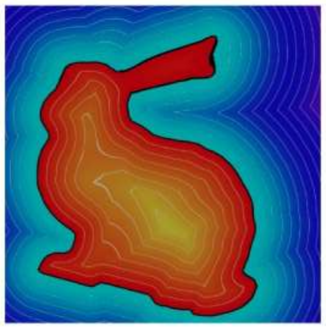


Dept

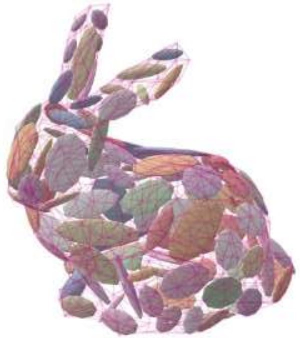


Surface
Normals

Implicit:



Signed distance
field



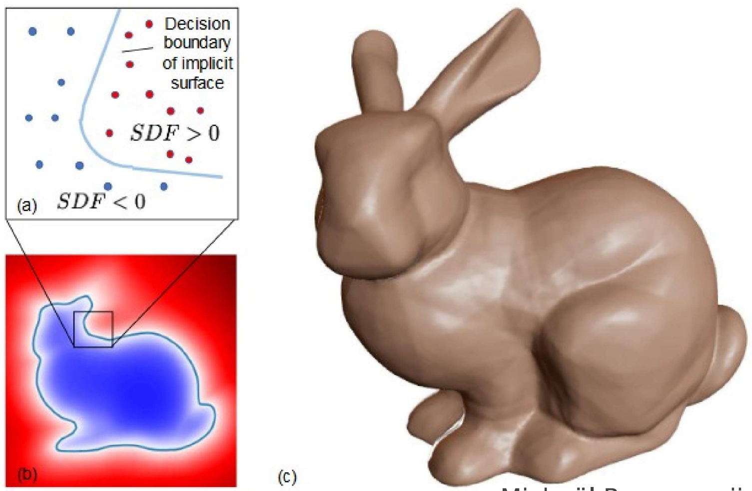
Mixture of primitives
(e.g gaussian mixtures)

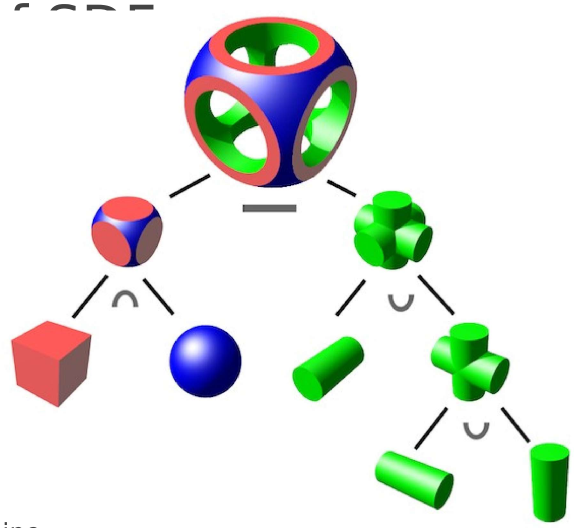
Signed Distance Field (SDF)

- Maps each 3D points ***p*** to it's signed distance to the object surface *S.* The sign is positive if the ***p*** is inside the object, and negative otherwis

$$SDF(p) = \mathrm{sign}(p) \cdot \min_{q \in \mathcal{S}} \|p - q\|$$

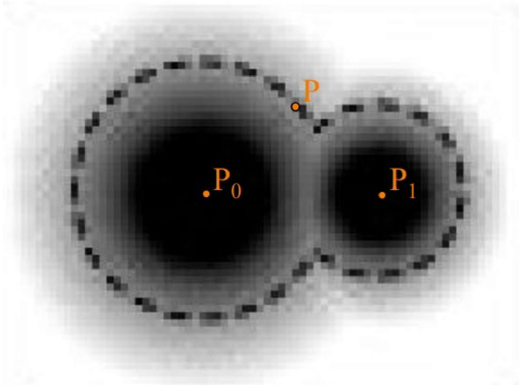
- Sign indicates whether the point ***p*** is inside (-) or outside (+) of the shape
- Shape's boundary as the zero-level-set of SDF
- d Geometry (n





Mixture of Gaussians

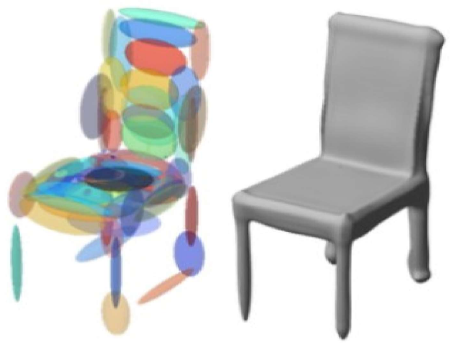
- Represents a shape as a mixture of local implicit functions (3D ga



$$F(\mathbf{x}, \boldsymbol{\Theta}) = \sum_{i \in [N]} f_i(\mathbf{x}, \theta_i)$$

$$f_i(\mathbf{x}, \theta_i) = c_i \exp\left(\sum_{d \in \{x,y,z\}} \frac{-(\mathbf{p}_{i,d} - \mathbf{x}_d)^2}{2\mathbf{r}_{i,d}^2} \right)$$

- Shape's boundary is defined as an iso-level of the **global** implicit function



[1] Genova19
[2] Genova20

Representing 3D surfaces with Implicit Functions

Pros:

- Compared to **point clouds**: **clearly defines the (iso-)surface**
- Compared to **meshes**: can continuously **adapt to arbitrary topology**
- Compared to **voxels**: can be represented with **few parameters** (e.g. mixture of simple implicit functions)
- They are **continuous** in 3D
- Can give analytic normals, can be applied with boolean operations, etc

Representing 3D surfaces with Implicit Functions

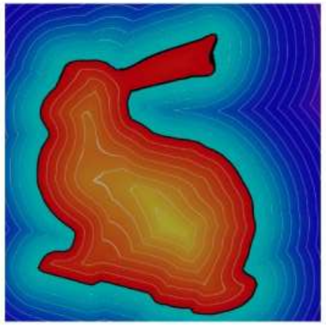
Pros:

- Compared to **point clouds**: **clearly defines the (iso-)surface**
- Compared to **meshes**: can continuously **adapt to arbitrary topology**
- Compared to **voxels**: can be represented with **few parameters** (e.g. mixture of simple implicit functions)
- They are **continuous** in 3D
- Can give analytic normals, can be applied with boolean operations, etc

Cons:

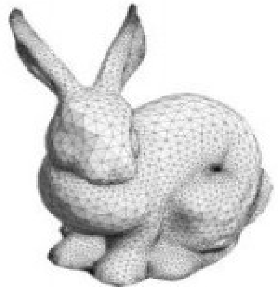
- SDF is well-defined for only watertight meshes (there is an interior and an exterior)

Converting Implicit Surfaces to meshes



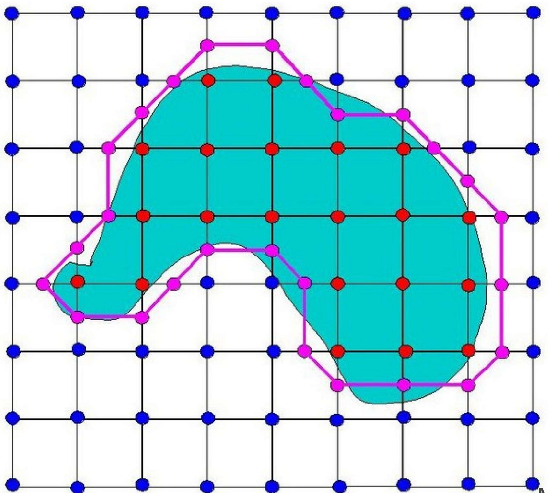
Implicit function

Extract (zero-level) iso-surface

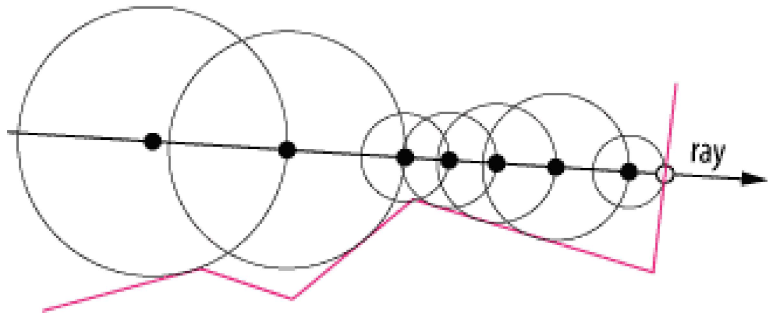


Mesh

Marching Cubes



Ray marching



Representing 3D surfaces with Implicit Functions

Pros:

- Compared to **point clouds**: **clearly defines the (iso-)surface**
- Compared to **meshes**: can continuously **adapt to arbitrary topology**
- Compared to **voxels**: can be represented with few parameters (e.g. mixture of simple implicit functions
- They are **continuous** in 3D
- Can give analytic normals, can be applied with boolean operations, etc

Cons:

- Implicit functions is well-defined for only watertight meshes (there is an interior and an exterior)
- Need extra steps to visualize
- Not all complex shapes can be efficiently / accurately represented with simple primitives

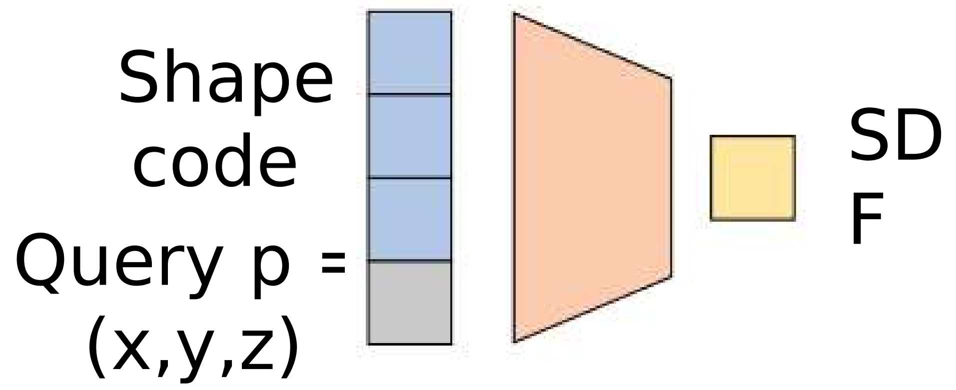
Representing 3D surfaces

DeepSDF: **Efficiently** representing complex shapes by learning their SDF

Idea: Learn a **continuous** representation of 3D implicit surfaces

Query p = (x,y,z), Shape latent code Z

$$F(p; \mathbf{Z}) = \mathrm{SDF}(p, \mathcal{M})$$

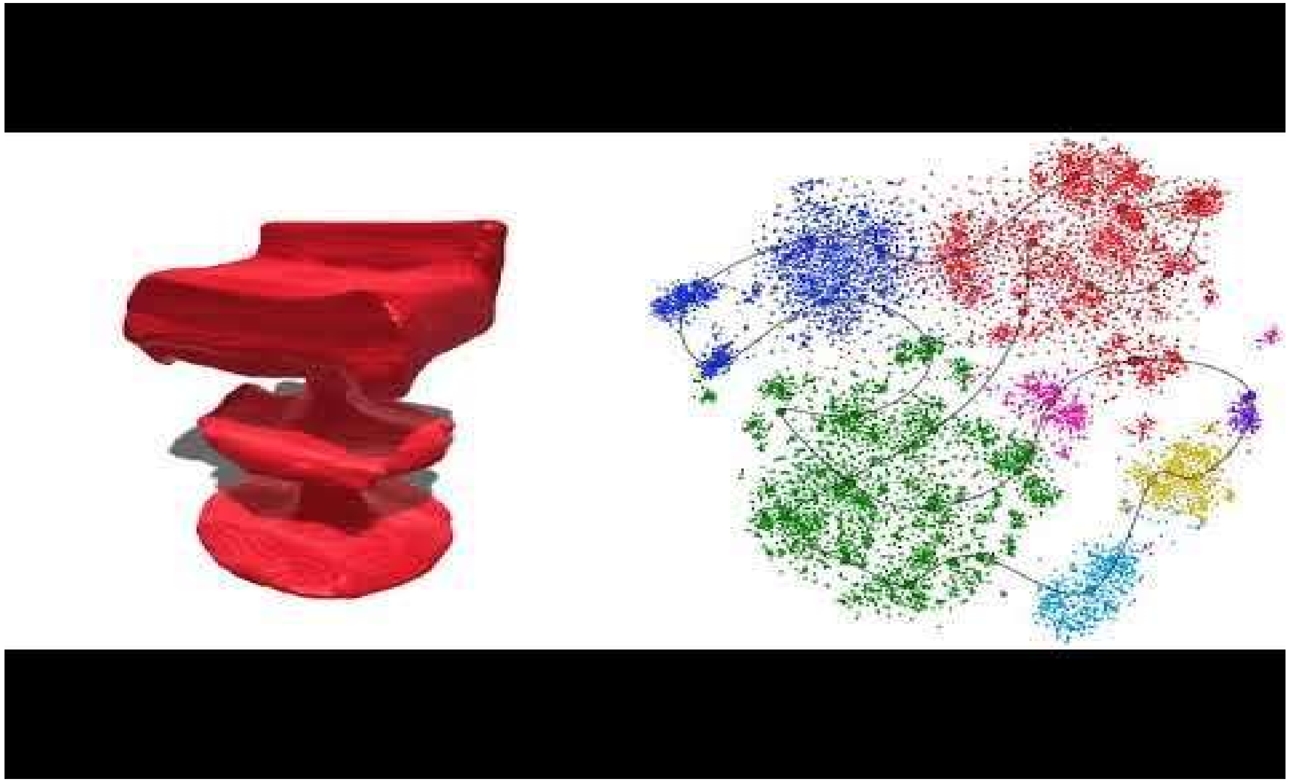


=> **Continuity** in 3D space AND shapes space

[3] Park19

Representing 3D surfaces

DeepSDF: Representing complex shapes by learning their SDF

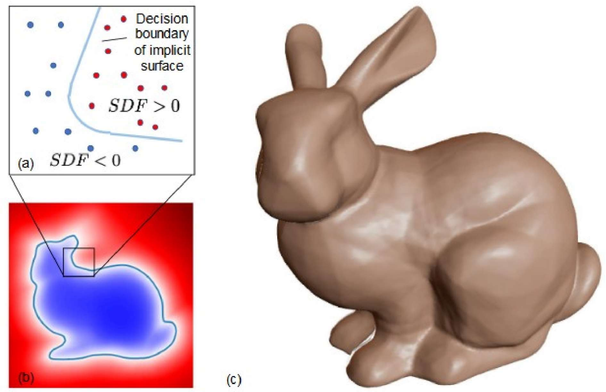


[3] Park19

Take home message on Implicit Functions

Representation of a continuous field

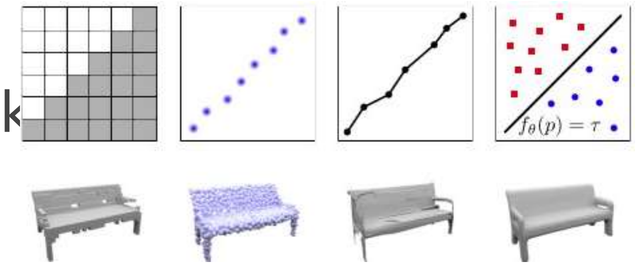
Learned implicit functions:
- Can represent complex shapes
- Are **continuous mappings** because they use
- Are applicable to N-D data: 2D images, 3D sha
  radiance fields



Visualization of implicit functions is done by extracting iso-surfaces:
1. Running inference for multiple queries in input space
2. Rendering the result by combining the queries

More works on Implicit Functions for 3D shape

- Occupancy Network



[4] Mescheder19

- PiFu and PiFuHD

[5] Saito19
[6] Saito20

References

[1] Genova et al., Learning Shape Templates with Structured Implicit Functions, ICCV 2019
[2] Genova et al., Local Deep Implicit Functions for 3D Shape, CVPR 2020
[3] Park et al., DeepSDF: Learning Continuous Signed Distance Functions for Shape Representation, CVPR 2019
[4] Mescheder et al., Occupancy Networks: Learning 3D Reconstruction in Function Space, CVPR 2019
[5] Saito, Huang, Natsume et al., PIFu: Pixel-Aligned Implicit Function for High-Resolution Clothed Human Digitization, ICCV 2019
[6] Saito et al., PIFuHD: Multi-Level Pixel-Aligned Implicit Function for High-Resolution 3D Human Digitization, CVPR 2020

Lecture on Implicit geometry
Lecture on Implicit surface
Lecture on Explicit & Implicit Surfaces

Courses and Seminars

Thomas Funkhouser's talk at 3DGV seminar

Princeton COS 426, Spring 2014 on Implicit Surfaces & Solid Representations