IEEE VR
2025
SAINT-MALO, FRANCE
March 8-12, 2025
THE 32nd IEEE CONFERENCE ON VIRTUAL REALITY AND 3D USER INTERFACES

A Practical Guide to **Radiance Fields for XR** Research and Applications

08.03.2025

Shohei Mori, University of Stuttgart, Germany

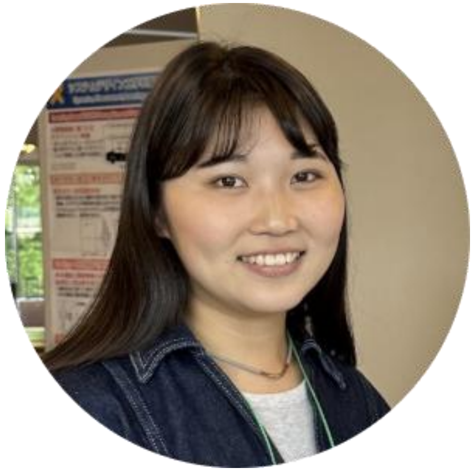
Ke Li, University of Hamburg, Germany

Mana Masuda, Keio University, Japan

Team

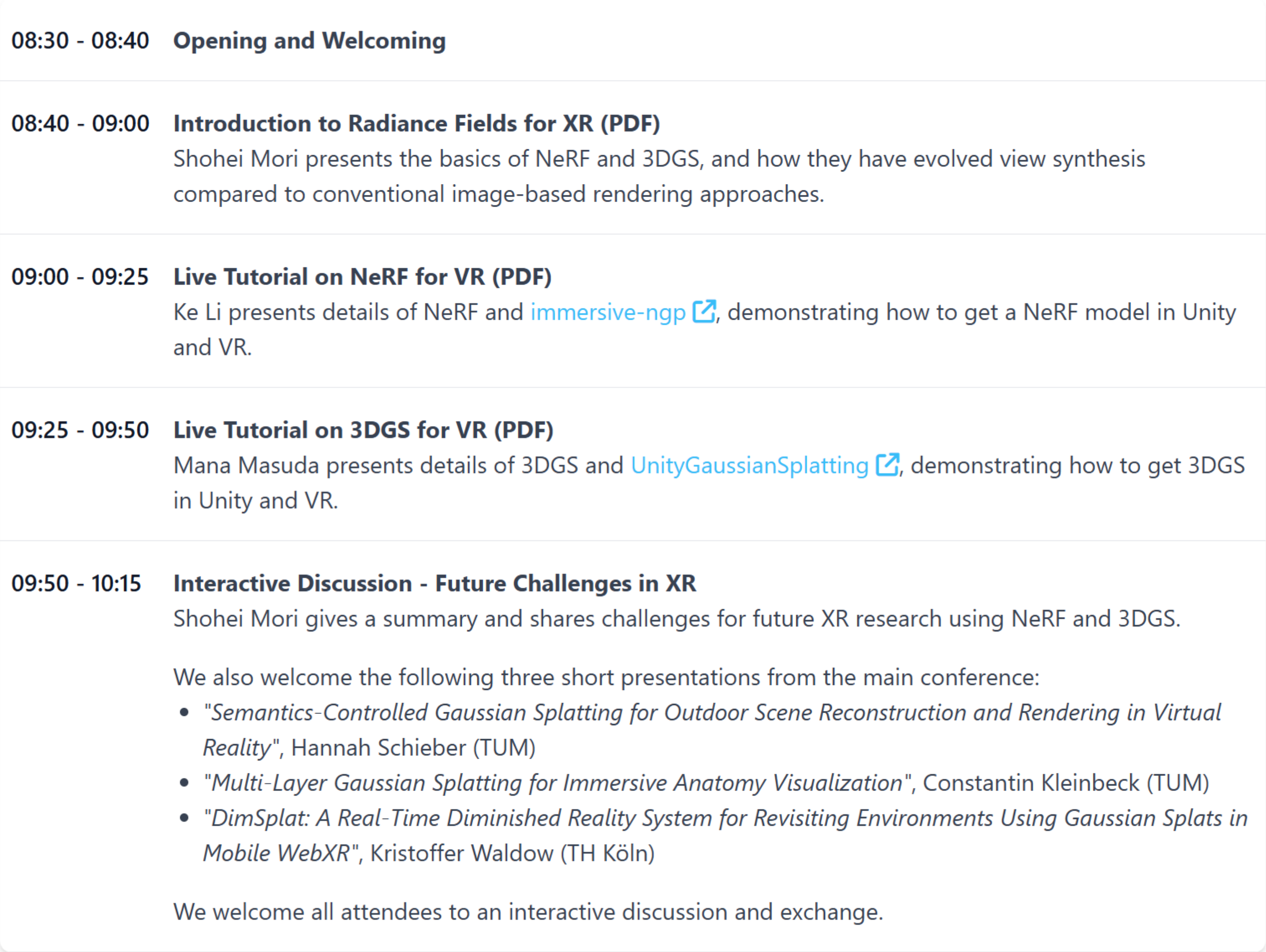
Shohei Mori
University of Stuttgart

Ke Li
University of Hamburg



Mana Masuda
Keio University

PROGRAM



08:30 - 08:40 **Opening and Welcoming**

08:40 - 09:00 **Introduction to Radiance Fields for XR (PDF)**
Shohei Mori presents the basics of NeRF and 3DGS, and how they have evolved view synthesis compared to conventional image-based rendering approaches.

09:00 - 09:25 **Live Tutorial on NeRF for VR (PDF)**
Ke Li presents details of NeRF and immersive-ngp, demonstrating how to get a NeRF model in Unity and VR.

09:25 - 09:50 **Live Tutorial on 3DGS for VR (PDF)**
Mana Masuda presents details of 3DGS and UnityGaussianSplatting, demonstrating how to get 3DGS in Unity and VR.

09:50 - 10:15 **Interactive Discussion - Future Challenges in XR**
Shohei Mori gives a summary and shares challenges for future XR research using NeRF and 3DGS.

We also welcome the following three short presentations from the main conference:

- "*Semantics-Controlled Gaussian Splatting for Outdoor Scene Reconstruction and Rendering in Virtual Reality*", Hannah Schieber (TUM)
- "*Multi-Layer Gaussian Splatting for Immersive Anatomy Visualization*", Constantin Kleinbeck (TUM)
- "*DimSplat: A Real-Time Diminished Reality System for Revisiting Environments Using Gaussian Splats in Mobile WebXR*", Kristoffer Waldow (TH Köln)

We welcome all attendees to an interactive discussion and exchange.

Radiance Fields at VR 2025

Papers & Keynote

Monday

- 09:45-10:45 | **Keynote** | The 3D Gaussian Splatting Adventure: Past, Present and Future
- 11:15-12:15 | GaussianHand: Real-Time 3D Gaussian Rendering for Hand Avatar
- 11:15-12:15 | SplatLoc: 3D Gaussian Splatting-based Visual Localization for Augmented Reality

Tuesday

- 08:30-09:30 | Look at the Sky: Sky-aware Efficient 3D Gaussian Splatting in the Wild
- 11:15-12:15 | **Semantics-Controlled Gaussian Splatting for Outdoor Scene Reconstruction and Rendering in Virtual Reality** by Hannah Schieber, *et al.* 🤝
- 11:15-12:15 | Frequency-aware Uncertainty Gaussian Splatting for Dynamic Scene Reconstruction
- 17:15-18:15 | SRBF-Gaussian: Streaming-Optimized 3D Gaussian Splatting
- 17:15-18:15 | MPGS: Multi-plane Gaussian Splatting for Compact Scenes Rendering

Wednesday

- 15:15-16:15 | 360o 3D Photos from a Single 360o Input Image
- 15:15-16:15 | Fov-GS: Foveated 3D Guassian Splatting for Dynamic Scene
- **15:15-16:15 | Multi-Layer Gaussian Splatting for Immersive Anatomy Visualization** by Constantin Kleinbeck, *et al.* 🤝

Radiance Fields at VR 2025

Posters & Demos

Monday

- Creating Virtual Environments with 3D Gaussian Splatting: A Comparative Study

Tuesday

- Human-in-the-Loop Gaussian Model Enhancement with Mobile Robotic Re-Capture
- **DimSplat: A Real-Time Diminished Reality System for Revisiting Environments Using Gaussian Splats in Mobile WebXR** by Kristoffer Waldow, *et al*. 🤝
- GaussianShopVR: Facilitating Immersive 3D Authoring using Gaussian Splatting in VR
- LIVE-GS: LLM Powers Interactive VR by Enhancing Gaussian Splatting

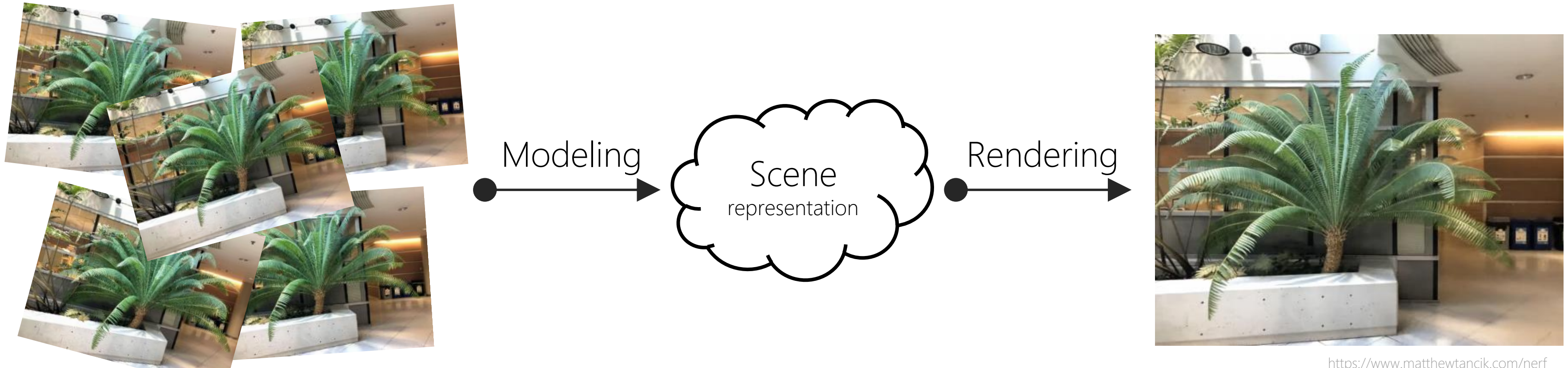
Wednesday

- Extending Gaussian Splatting to Audio: Optimizing Audio Points for Novel-view Acoustic Synthesis

Demo

- LLM-powered Gaussian Splatting in VR interactions

Pipeline



Modeling

Scene
representation

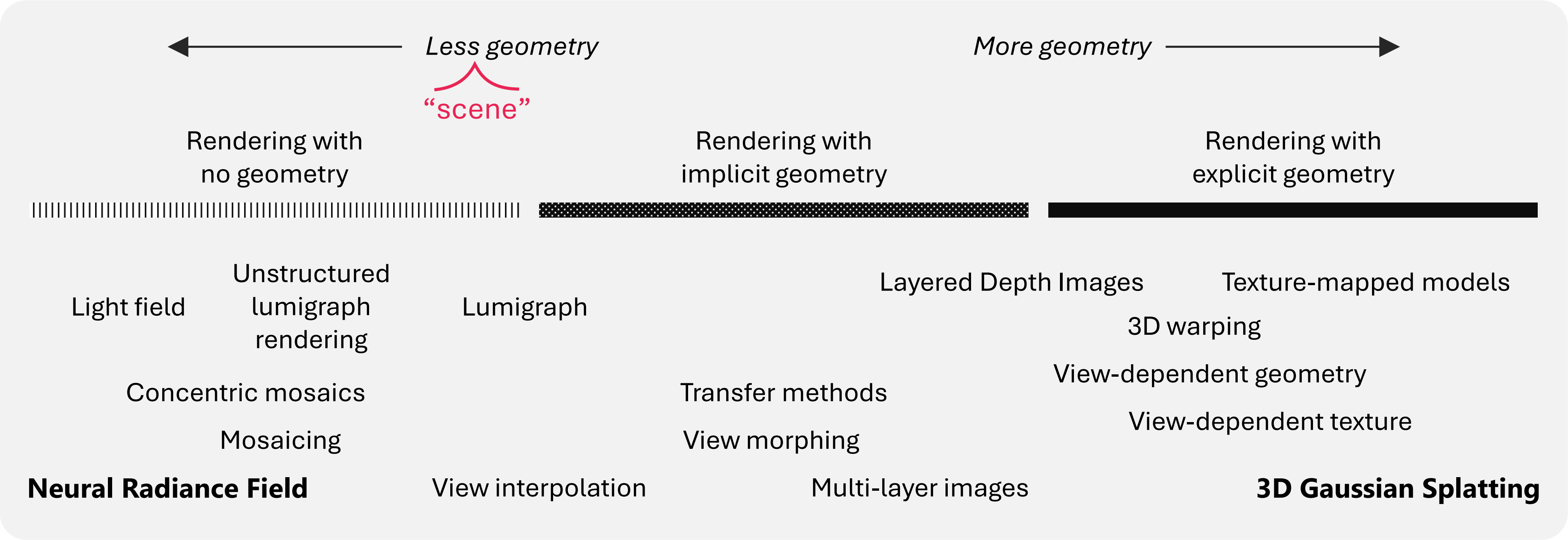
Rendering

https://bmild.github.io/llff/

https://www.matthewtancik.com/nerf

Image-Based & Neural Rendering Continuum

Modified from H.-Y. Shum, S.-C. Chan, and S. B. Kang, “Image-Based Rendering,” Springer (2007)



Less geometry

“scene”

More geometry

Rendering with no geometry

Rendering with implicit geometry

Rendering with explicit geometry

Light field

Unstructured lumigraph rendering

Lumigraph

Layered Depth Images

Texture-mapped models

3D warping

View-dependent geometry

Concentric mosaics

Transfer methods

View-dependent texture

Mosaicing

View morphing

Neural Radiance Field

View interpolation

Multi-layer images

3D Gaussian Splatting

Image-Based Rendering (IBR)

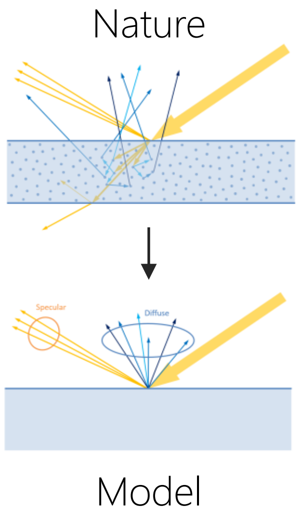
More photographs instead of precise scene models

Geometry (surface locations, visibility, deformation, etc.)
Surface properties (albedo, reflection, BRDF, etc.)

Recently, often referred to as **view synthesis**

Significantly reduced modeling effort, but still visually convincing

that comes with many known limitations…

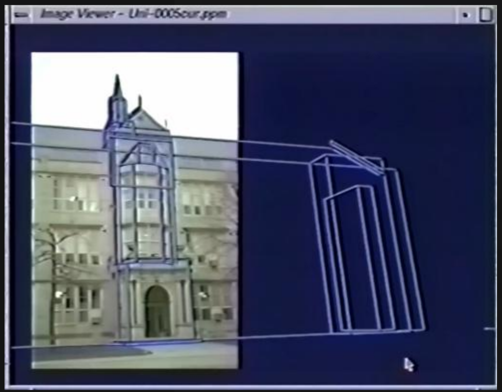


Photogrametry/scanning falls short in specular, hairs, transparency, complex sub-surface scattering, dynamics, and more…

View-Dependent Texture Mapping (VDTM)

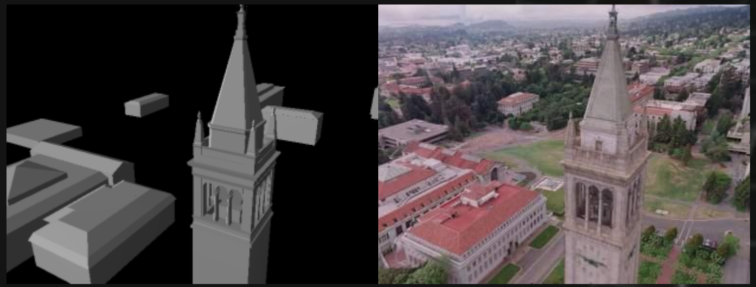
P. Debevec, Modeling and Rendering Architecture from Photographs, Ph.D. Thesis, UC Berkley, 1996.

- Each surface maintains a list of reference views, from which it is visible
- For rendering current view, reference views are projected onto the surface
- Pixels with overlaps are blended depending on proximity of current view to the reference views
- GPU-based projective texture mapping for further efficiency



GUI for manual modeling and camera/photograph registration

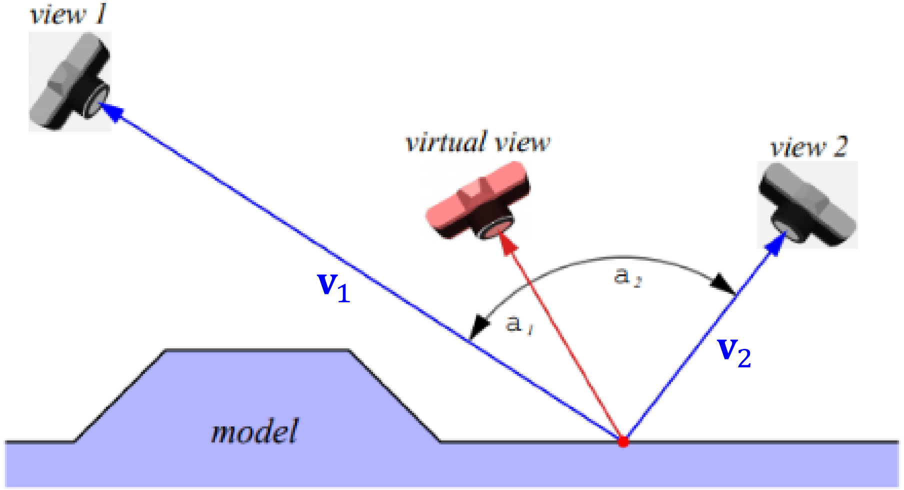
Camera/photograph registered in the scene

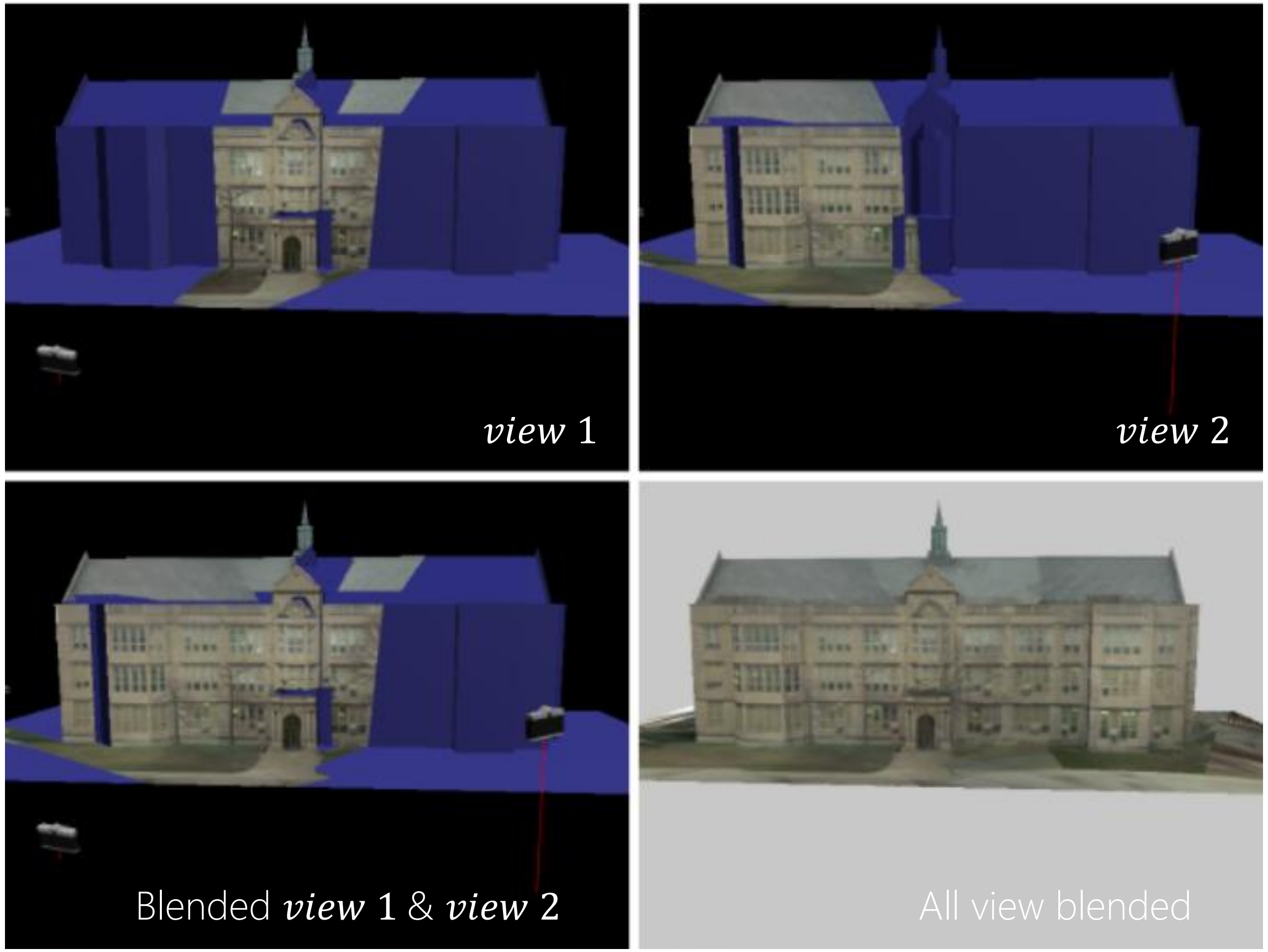


Blending Photographs

P. Debevec, Modeling and Rendering Architecture from Photographs, Ph.D. Thesis, UC Berkley, 1996.

- Calc. $\mathbf{v}_1$, $\mathbf{v}_2$, and $\mathbf{v}$
- Calc. a_1 and a_2
- Calc. weights (w_1 and w_2)
- Blend $view\ 1$ and $view\ 2$
  - $virtual\ view = w_1 * view\ 1 + w_2 * view\ 2$





Limitation in IBR

- Choices are
  - Poor 3D reconstruction + More images → More storage
  - Accurate 3D reconstruction + Fewer images → More computation

Not preferable to XR

- Select *k* „nearest“ views per vertex
  - View switching artifacts
  - Views selection during photographing and rendering with care!
- No defacto-standard solutions
- Very few open source projects (especially, in XR)

Differentiable Rendering Pipeline

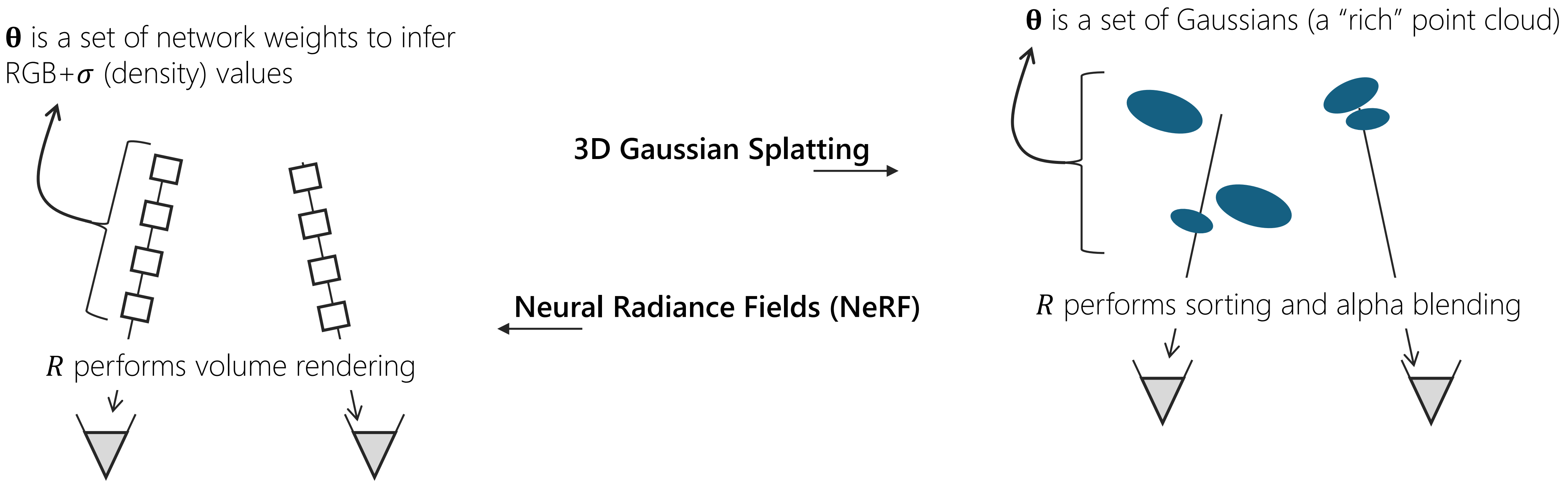
$R(\boldsymbol{\theta}) = \mathrm{I}'$

- A set of scene parameters $\boldsymbol{\theta}$ is given
  - $\boldsymbol{\theta}$ can be points, voxel, mesh, implicit (or network parameters), Gaussians, etc.
- A differentiable renderer R calculates an image I' from $\boldsymbol{\theta}$
- Calculate the loss between the rendered image I' and its ground truth I
  - Auto-grad backpropagates to reflect the loss to the scene parameters $\boldsymbol{\theta}$

R provides an "interpretation" to $\boldsymbol{\theta}$ so that $\boldsymbol{\theta}$ can represent a scene and optimization framework tries to find the best $\boldsymbol{\theta}$ through evaluations of generated image I' by comparing it with its ground truth I (i.e., $\mathrm{Loss}(\mathrm{I}', \mathrm{I})$, that is *analysis by synthesis*)

Differentiable Rendering Pipeline

$$R(\boldsymbol{\theta}) = \mathrm{I}'$$



$\boldsymbol{\theta}$ is a set of network weights to infer RGB+$\boldsymbol{\sigma}$ (density) values

R performs volume rendering

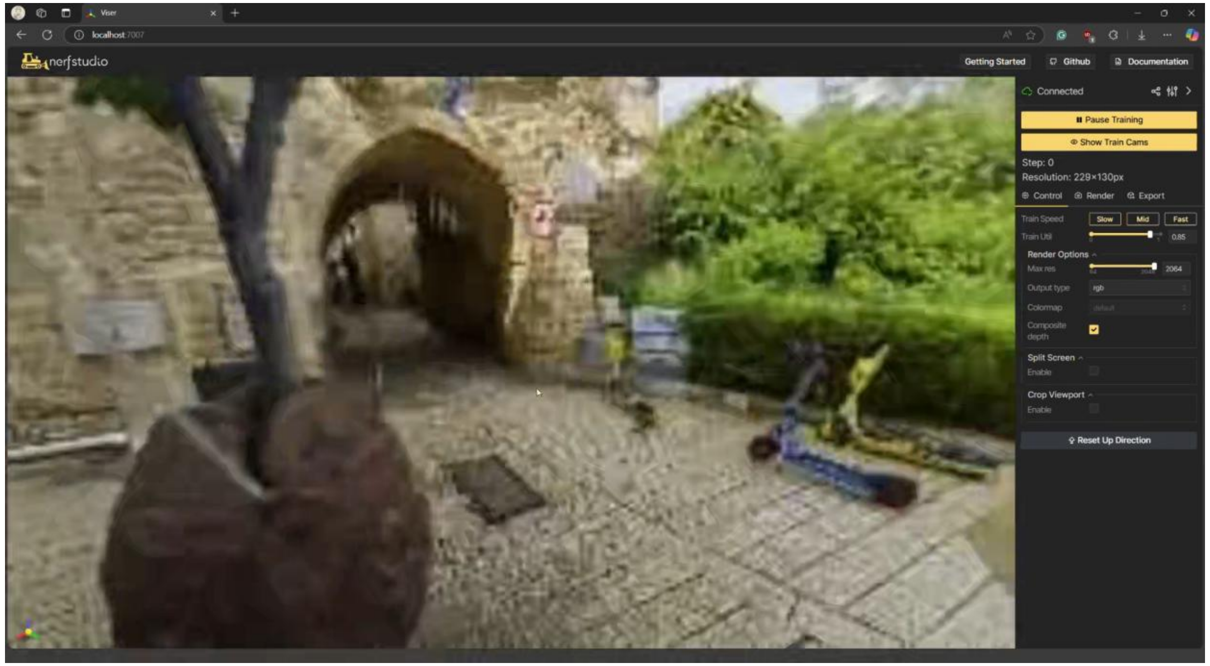
3D Gaussian Splatting →

← **Neural Radiance Fields (NeRF)**

$\boldsymbol{\theta}$ is a set of Gaussians (a "rich" point cloud)

R performs sorting and alpha blending

NeRF vs. 3DGS



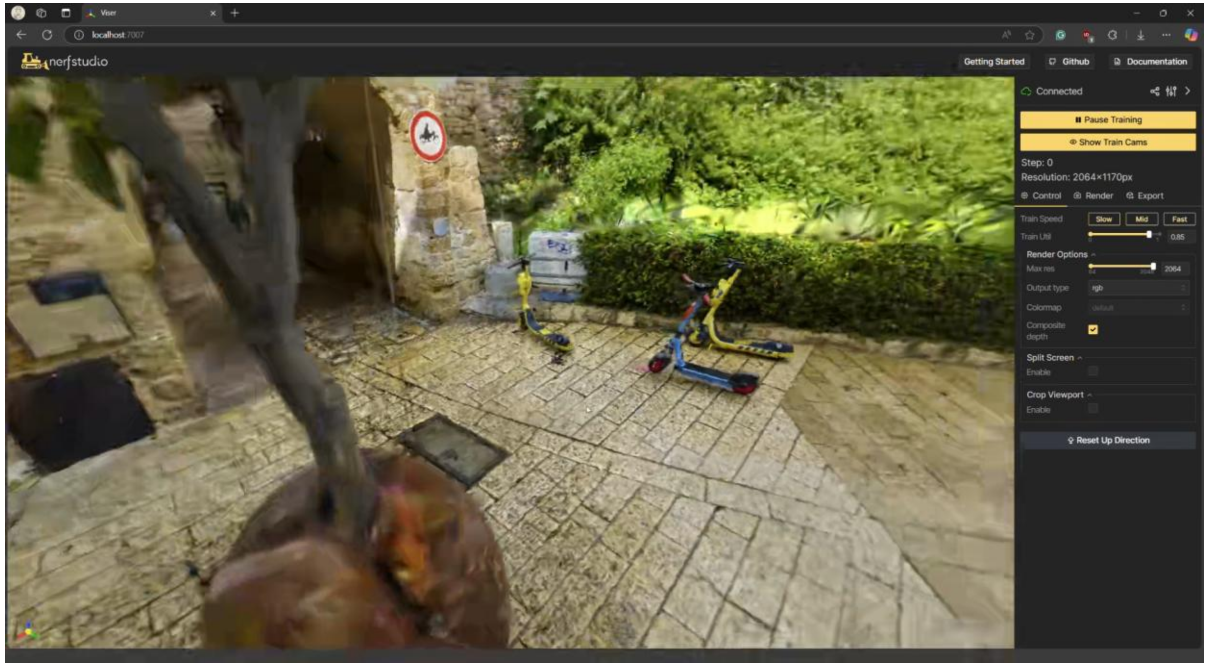
Step: 0
Resolution: 229×130px

Train Speed
Train Util 0.85

Render Options
Max res 2064
Output type rgb
Colormap default
Composite depth ☑

Split Screen
Enable □

Crop Viewport
Enable □



Step: 0
Resolution: 2064×1170px

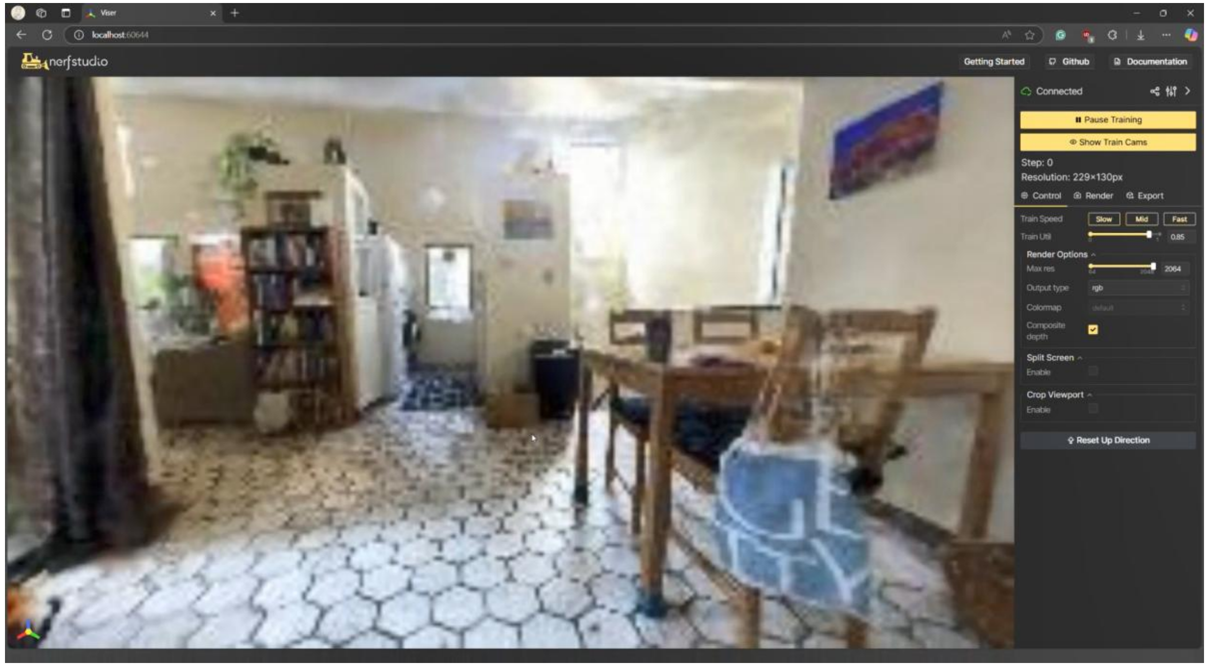
Train Speed
Train Util 0.85

Render Options
Max res 2064
Output type rgb
Colormap default
Composite depth ☑

Split Screen
Enable □

Crop Viewport
Enable □

NeRF vs. 3DGS



Connected

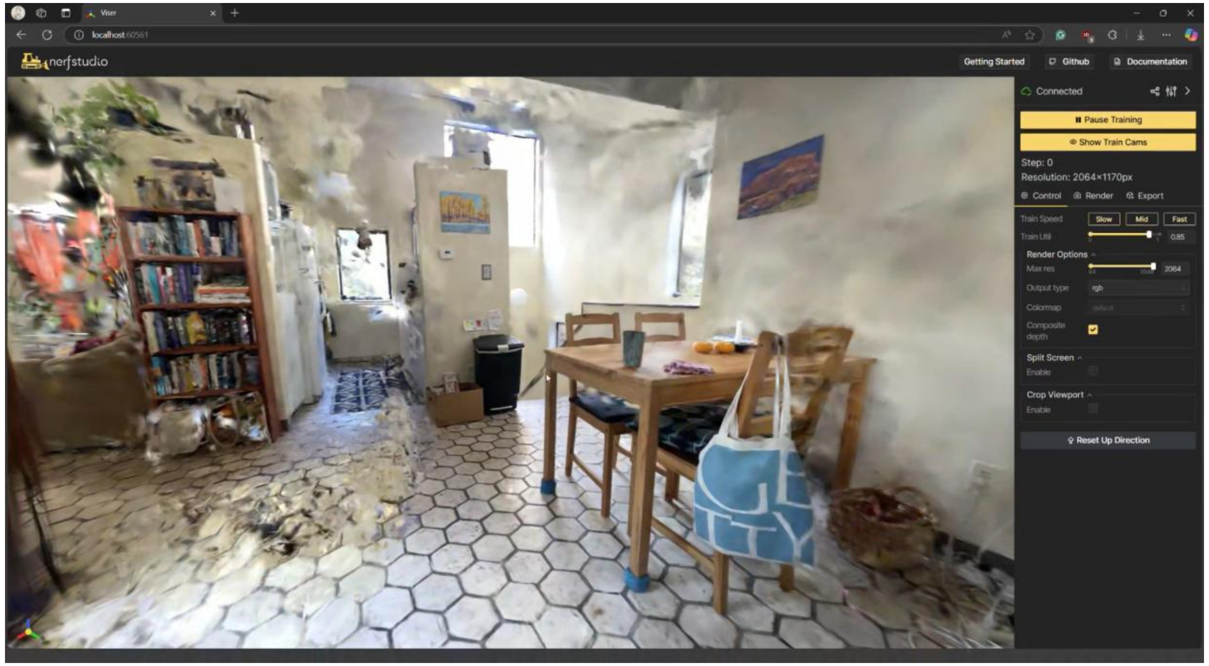
Step: 0
Resolution: 229×130px

Train Speed
Train Util 0.85

Render Options
Max res 2064
Output type rgb
Colormap default
Composite depth ☑

Split Screen
Enable □

Crop Viewport
Enable □



Connected

Step: 0
Resolution: 2064×1170px

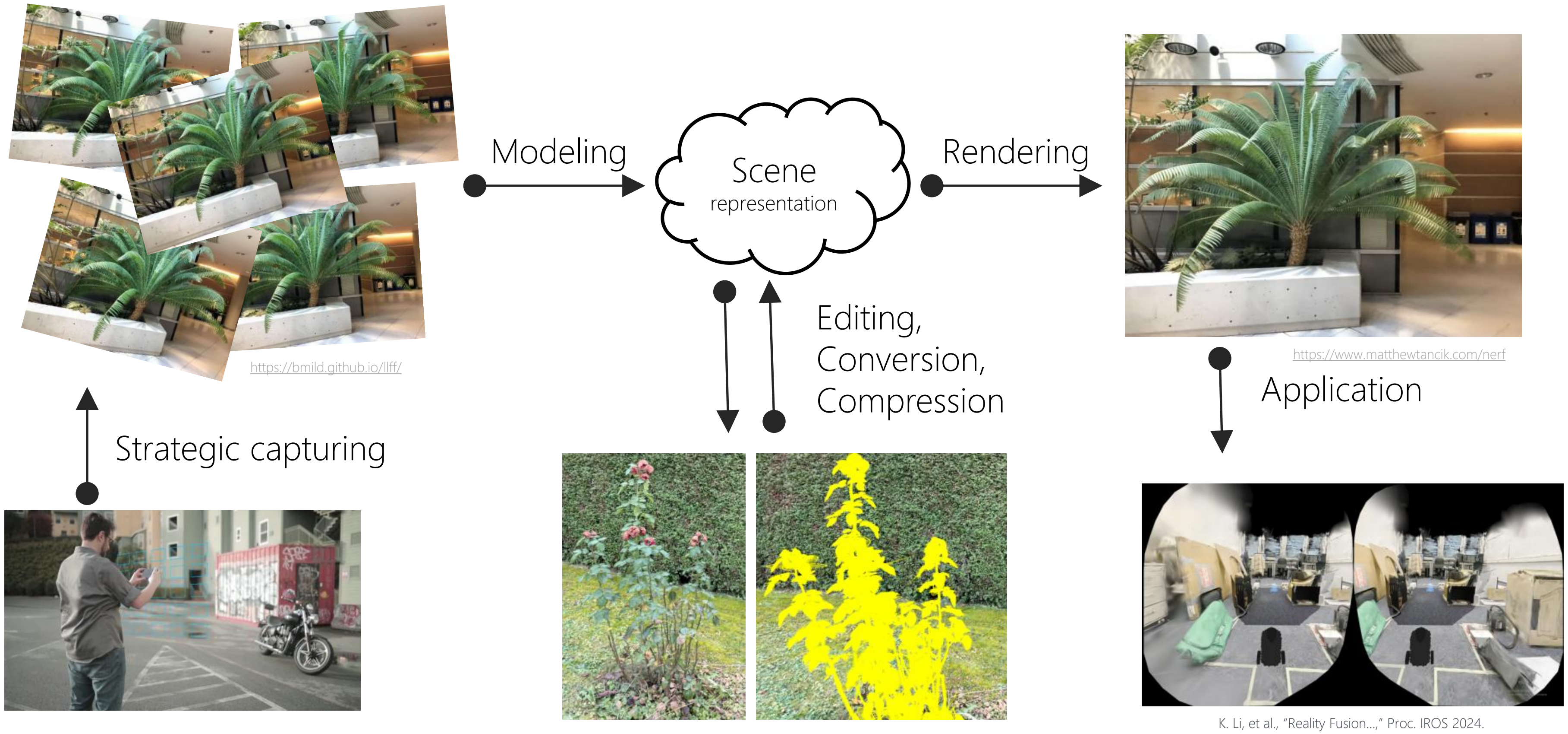
Train Speed
Train Util 0.85

Render Options
Max res 2064
Output type rgb
Colormap default
Composite depth ☑

Split Screen
Enable □

Crop Viewport
Enable □

Pipeline



Modeling

Scene representation

Rendering

Strategic capturing

Editing, Conversion, Compression

Application

https://bmild.github.io/llff/

https://www.matthewtancik.com/nerf

K. Li, et al., "Reality Fusion...," Proc. IROS 2024.

Capture Tools

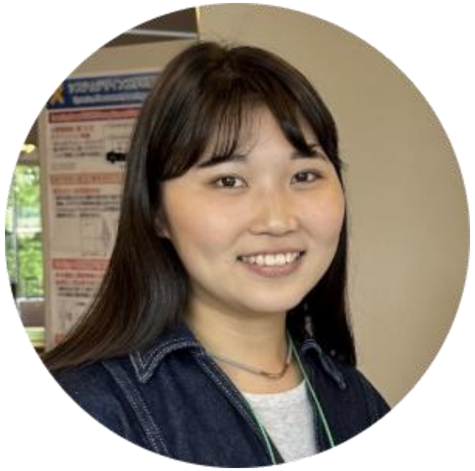
- Scaniverse
  - 3DGS (local processing)
  - iOS (with LiDAR) / Android
    - Real-time 3DGS on iOS
  - Free of charge
- Polycam
  - 3DGS (server processing, up to 100 images)
  - iOS (with LiDAR) / Android
- COLMAP / GLOMAP
  - Defacto standard for research
  - Python interface

Next-up

Ke Li

University of Hamburg

Neural Radiance Fields
Immersive-ngp
(Unity + VR + Instant-ngp)



Mana Masuda

Keio University

3D Gaussian Splatting
UnityGaussianSplatting
(Unity + VR + 3DGS)

Future Directions

Topics around Radiance Fields

- **Interactive Experiences**: Education, Training, Telepresence / Collaboration / Avatar
- **User-centric rendering**: Foveated Rendering / VST-HMD
- **Benchmarking**: Quality Assessment / Toolkit
- **Multi-modal**: Haptics / Audio
- **Content Editing**: Style Transfer / Style Edit / Relighting / Diminished Reality
- **Content Generation**: Text to 3D (content / scene) / Content Generation
- **Spatial Registration**: Tracking / SLAM / Bundle adjustment
- **Dynamic Capture**: 4D / Performance / Dynamics
- **Optimization Techniques**: Faster Rendering / Few-shot / Streaming / Compression