Neural Radiance Fields for Game Engines and Web Applications

Converting Photos into 3D Assets for Real-time Rendering

Fraunhofer IIS
Fraunhofer-Institut für Integrierte Schaltungen IIS

Fraunhofer Institute for Integrated Circuits IIS

Founded in 1985, >1,100 employees, budget of €167.9 Million
Locations in **Erlangen**, Nuremberg, Fürth, Dresden, Ilmenau, Munich, Bamberg, Waischenfeld, Coburg, Würzburg, Deggendorf and Passau.

Fraunhofer IIS
Audio and Media Technologies

With more than **30 years of experience**, Fraunhofer IIS is the world's most respected authority in low bit-rate, high-quality audio coding, signal processing and image coding

Over **300 employees**

Fraunhofer IIS has licensed its audio codec software to more than **2,000 companies**

Fraunhofer audio technology is integrated in virtually **all Consumer Electronics devices, PCs and smartphones**

  intern

Fraunhofer IIS

Agenda

Part 1: Overview of view-interpolation and 3D-reconstruction

- Motivation
- Technology landscape overview

Part 2: Principles of Neural Radiance Fields (NERFs)

- Technical principles
- Relation to multi-view stereo
- Fundamental challenges

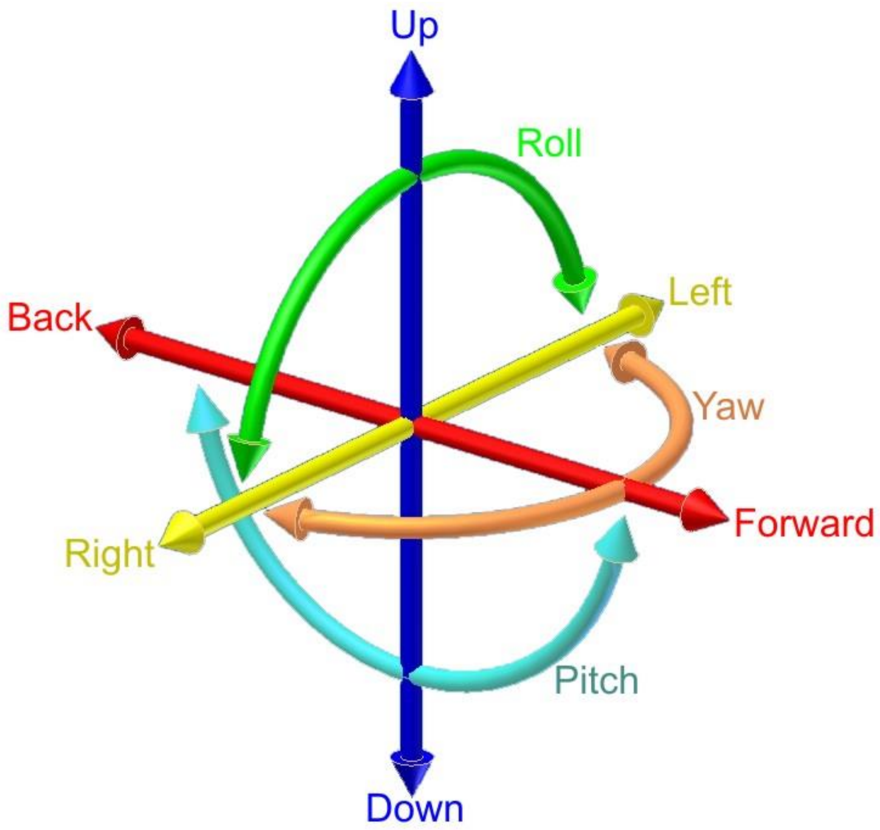
Part 3: Neural Radiance Fields for Real-time Rendering

- Solution Approaches
- Voxel Meshes
- Integration into Game Engines and Web Applications

Part 1: Overview of view-interpolation and 3D-reconstruction

Motivation

Observation of a scene or an object with six degrees of freedom (6-DoF)



6DoF

- Archival, Documentation
- Remote control, assistance
- Entertainment (Gaming, virtual reality, movie creation)
- Presentation (e-commerce, advertisement)
- Planning & simulation
- …

Not only for VR

Observation of a scene or an object with six degrees of freedom (6-DoF)

Fraunhofer IIS

Problem formulation

Interpolate views between capture cameras

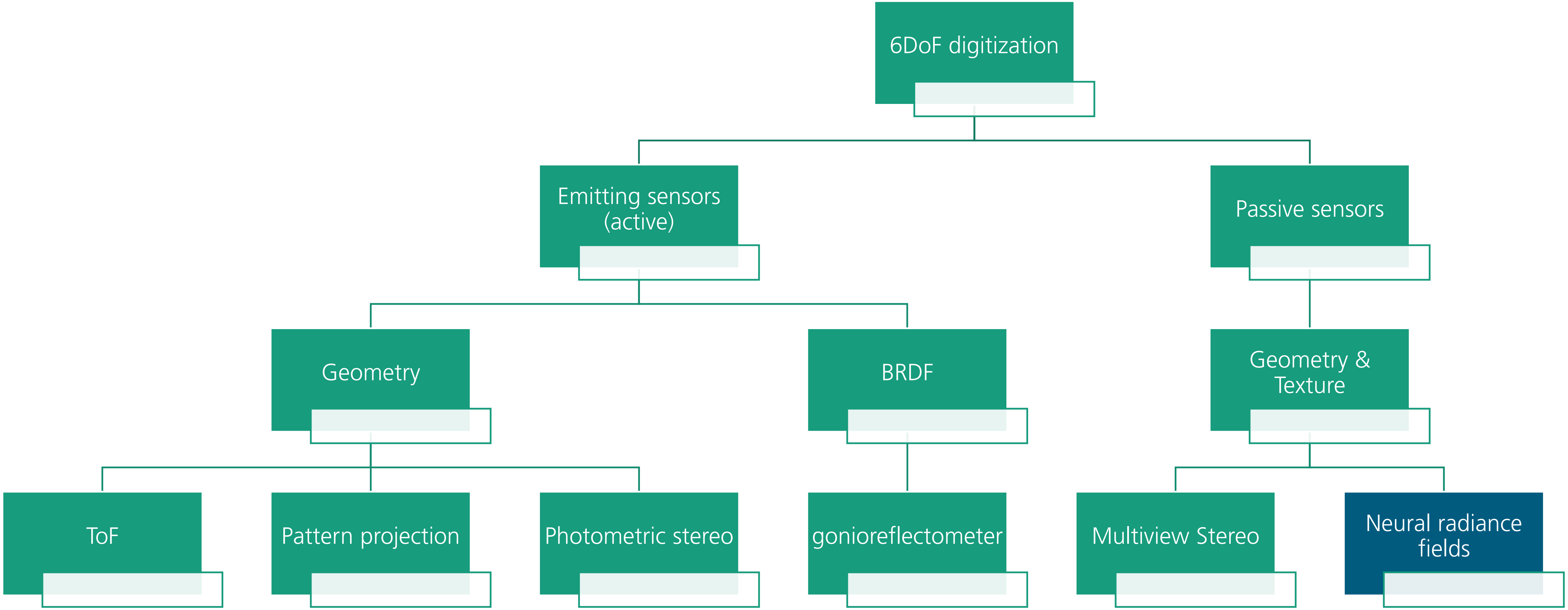
Solution aspects

View-dependent appearance
Solution aspects

Know the color of an object point from every possible viewing direction

Camera moves relative to constant car and light sources

Selected technology landscape



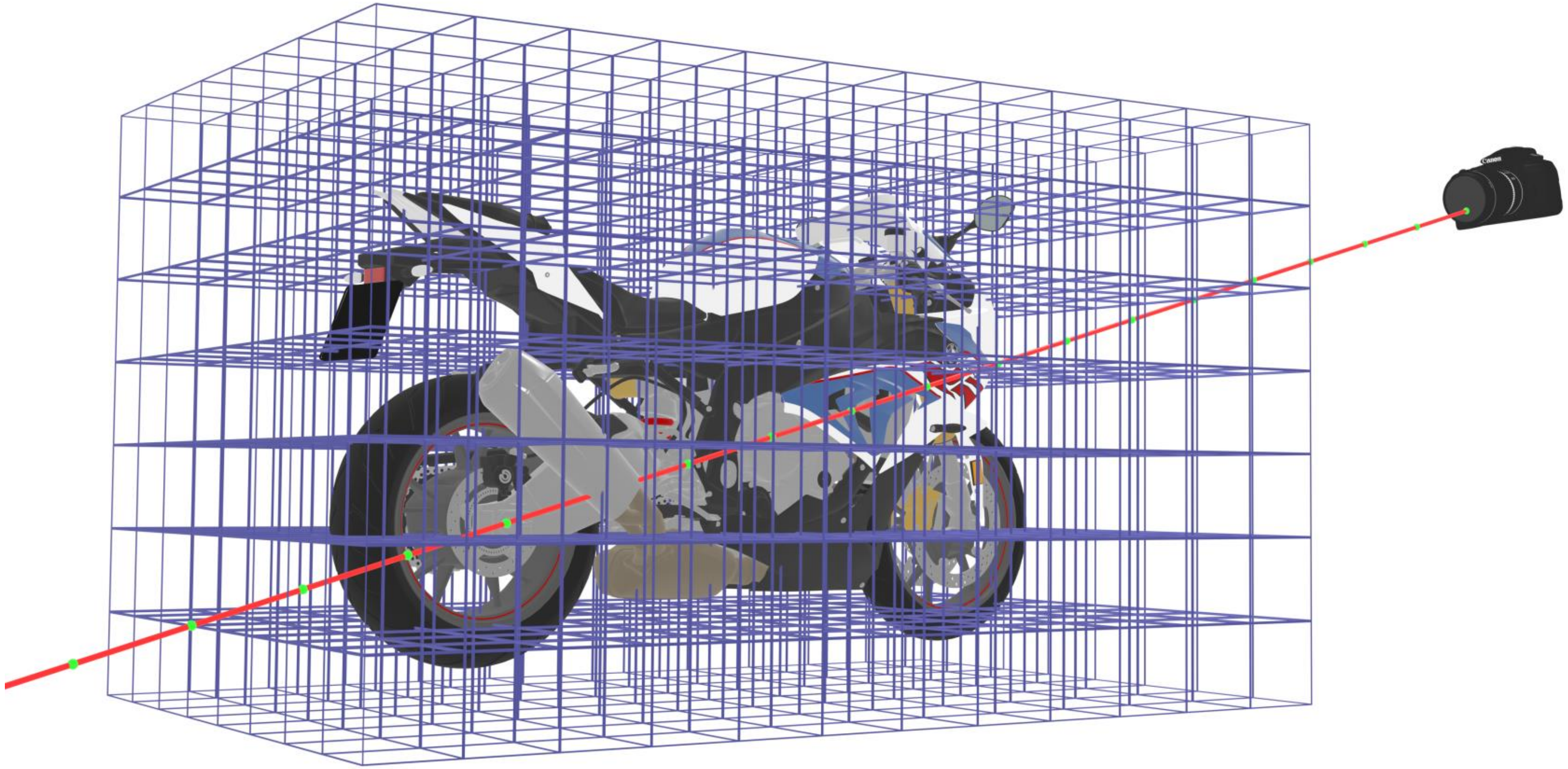
  intern

Fraunhofer IIS

Part 2: Principles of Neural Radiance Fields (NERFs)

Neural radiance fields

Technical principles



Approach

- Pixel based ray shooting
- Simulation of physical rendering
- Voxels contain
  - Density
  - Color
- Optimize voxel values to reproduce observed images

AI contribution

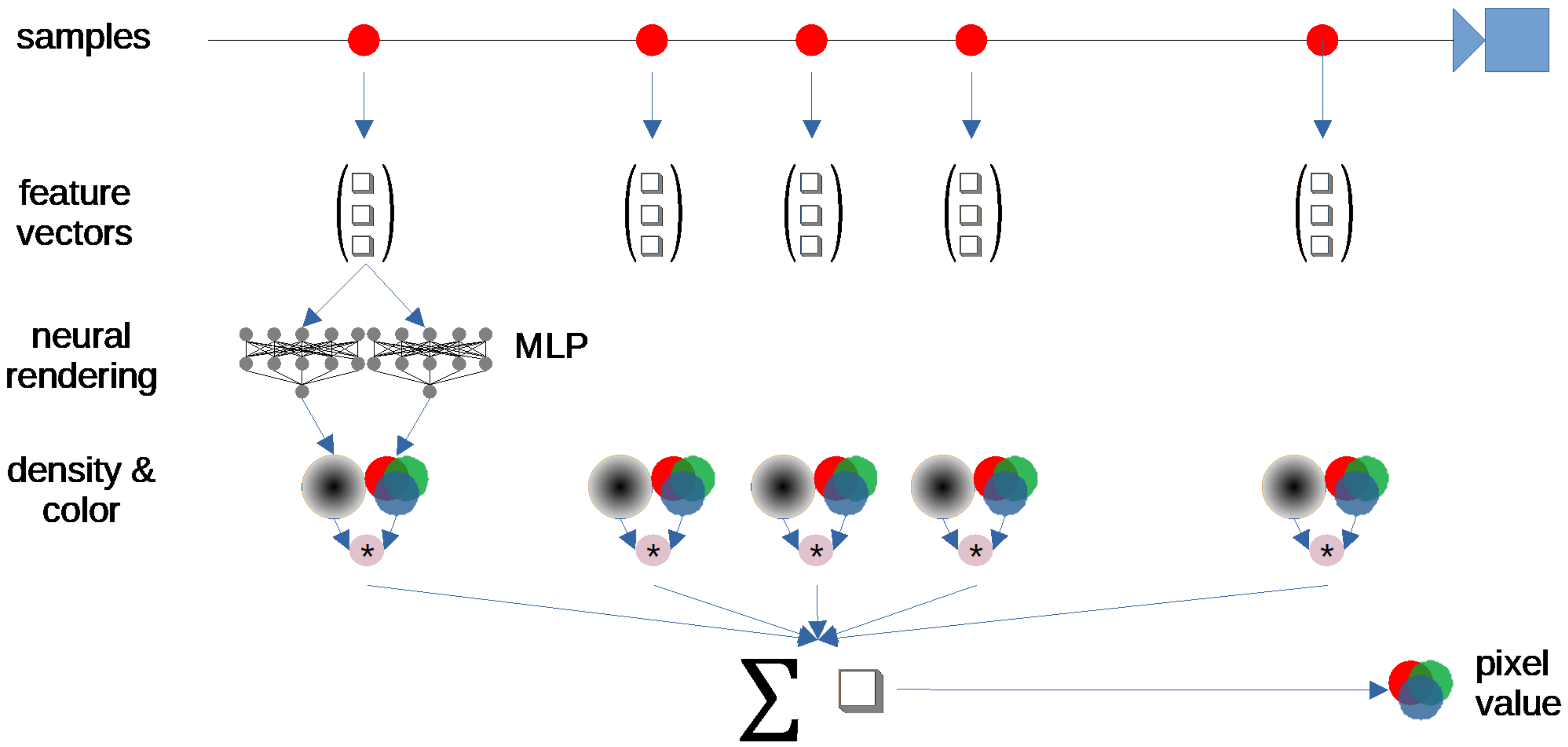
- Massive progress in large scale optimization using GPUs
- Neural representation and rendering

  intern

Fraunhofer IIS

Algorithm architecture

Schematic representation



Fraunhofer IIS

Relation and comparison with multi-view stereo

Triangulation based on correspondences

- Known correspondences and camera parameters permit to compute position in space
- Determination of reliable correspondences difficult
- Noisy and incomplete point cloud

Photogrammetry versus neural radiance fields

Photogrammetry

- Optimize for matching costs
  - Only indirectly optimize for visual appearance
- A single depth value per pixel
  - No support of transparent surfaces
- Generation of a Lambertian texture
  - No view dependent appearance
- Geometry driven

Neural radiance fields

- Optimize for visual appearance directly
  - Leads to better quality
- Voxels may be semi-transparent
- Voxel color depends on viewing direction
- Appearance driven
  - Geometry extraction more difficult

Fraunhofer IIS

Real-data (rendering only)

Comparison with traditional methods

Early results

Voxel based reconstruction

Photogrammetry

  intern Fraunhofer IIS

Comparison with traditional methods

Early results

Voxel based reconstruction

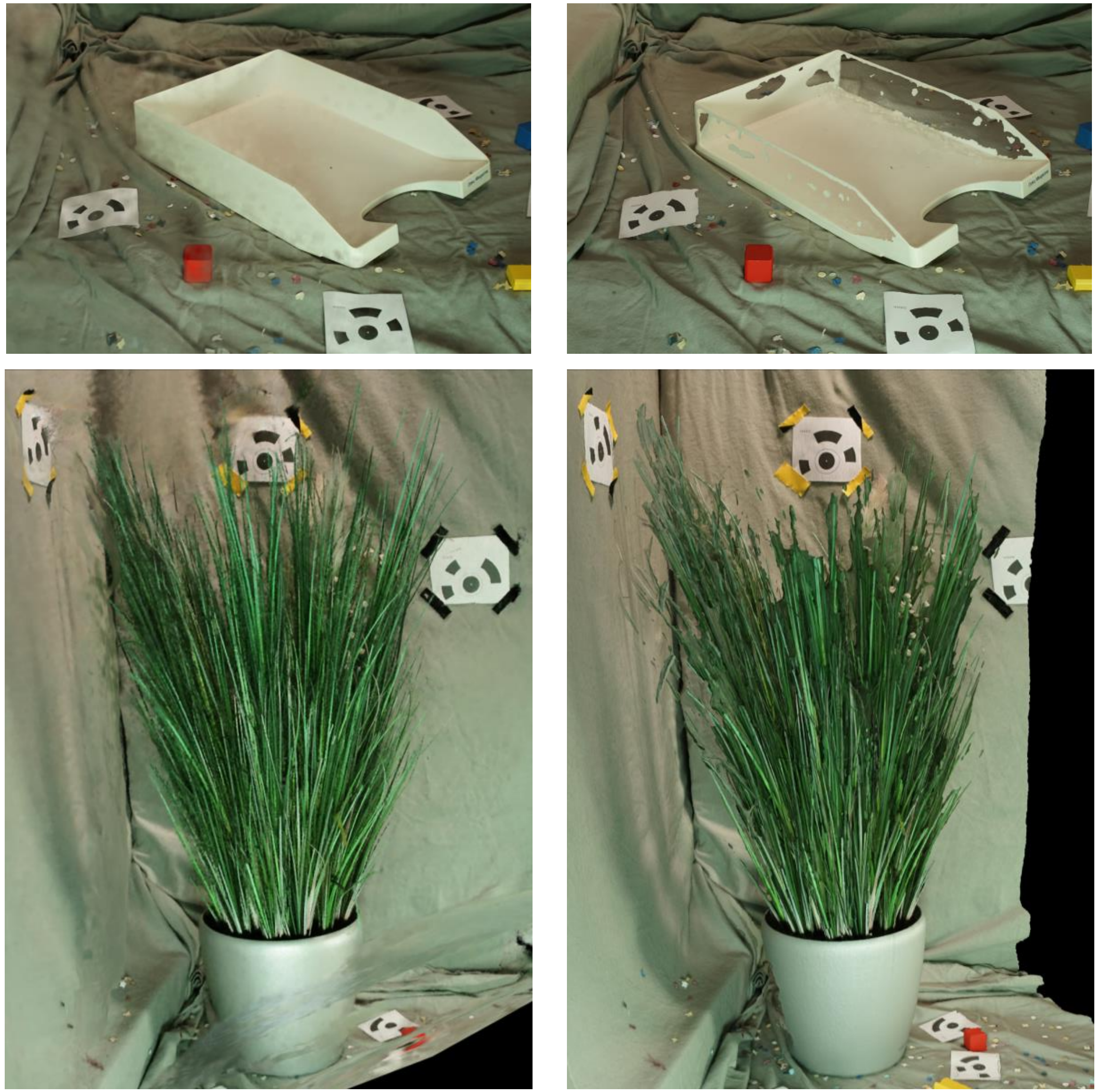
Photogrammetry

Summary

Neural Radiance Fields (NERFs)

- Can deliver superior reconstruction quality

- But are not perfect neither



Fundamental challenges

Memory consumption and compute time

Photogrammetry
- Depth-map based
- Complexity O(n^2)

Neural radiance fields
- Voxel grid
- Complexity O(n^3)

Break in Workflow

- Optimization of appearance instead of consistent mesh
- Imposing mesh constraint may reduce overall quality

Complex Optimizers

- Convergence not always granted

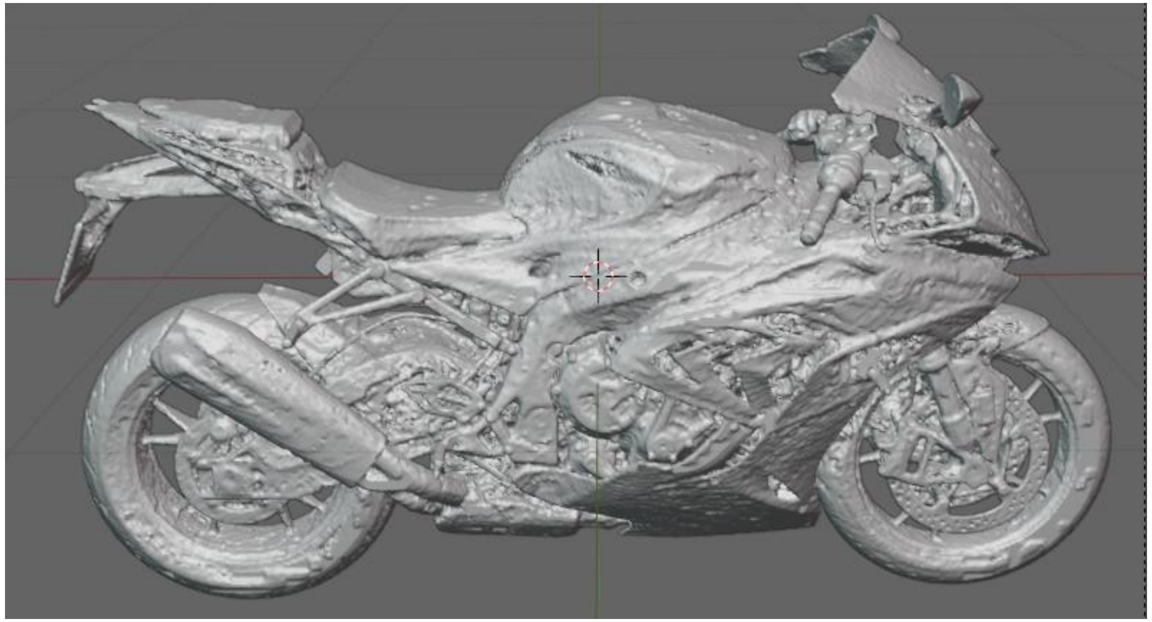
Need of

- Good parameter heuristics
- Clean an consistent input data
- Clever reconstruction methods

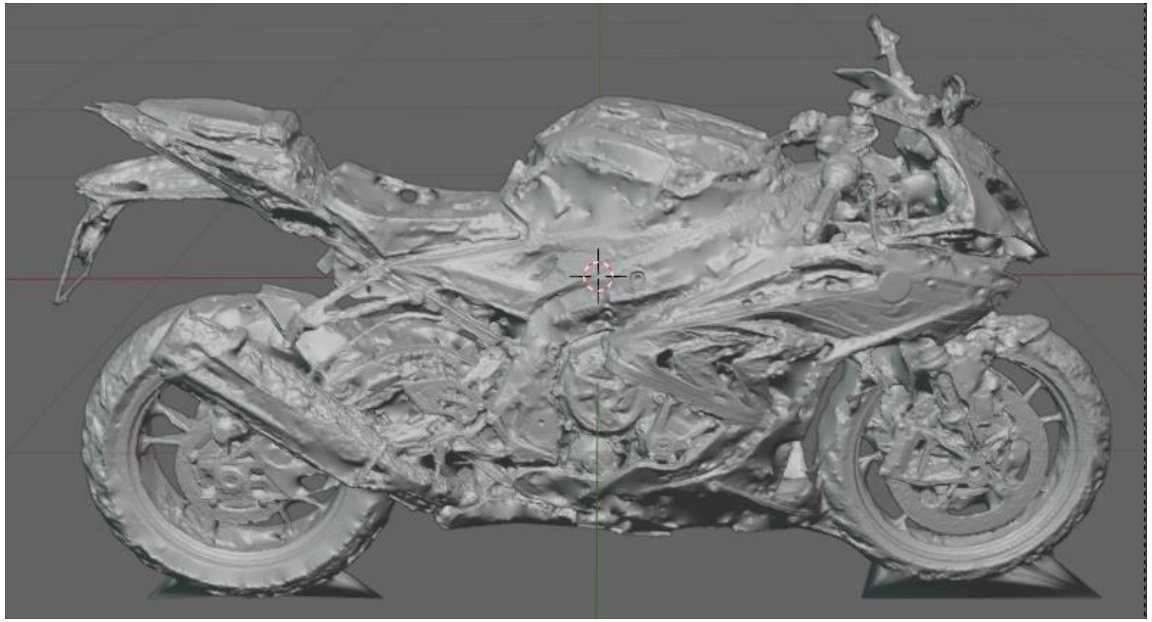
  intern

Fraunhofer IIS

On the geometry quality



Voxel based reconstruction



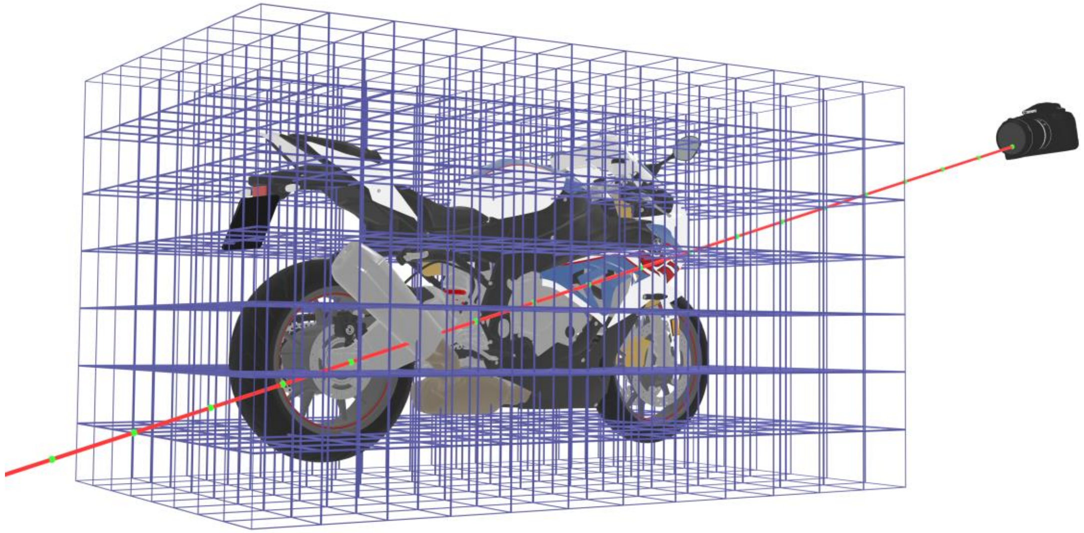
Photogrammetry

Part 3: Neural Radiance Fields for Real-time Rendering

Fraunhofer IIS

The fundamental problem

Rendering Neural Radiance Fields (NERFs)



Voxel rendering is slow

- For each ray, query all intersected voxels

Tricks for speed-up

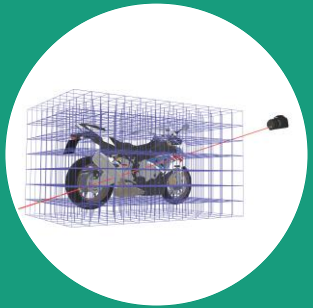
- Skip empty space
- Early ray termination

Limitations

- NERFs reconstruction voxel clouds
- Lack of discrete geometry
- Requires to render multiple voxels per pixels

Solution space

Real-time rendering



Optimize voxel rendering

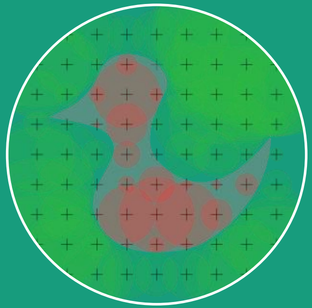
- New rendering libraries
- New rendering hardware

Use of novel rendering primitives

- [Gaussian] splatting
- New rendering pipeline (i.e. due to tiling)

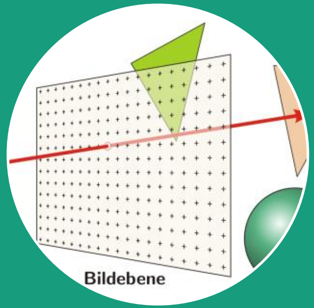
Use of voxel meshes

- Generalized meshes



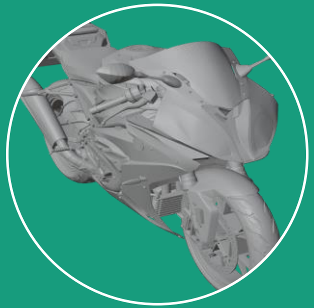
Integrate surface constraints into NERFs

- i.e. signed distance fields
- Harder to converge
- No support of transparencies



Direct mesh optimization (inverse rendering)

- Much harder to converge



Convert voxels into meshes

- Loss of quality due to break in rendering principle

Definition

Voxel meshes

Definition

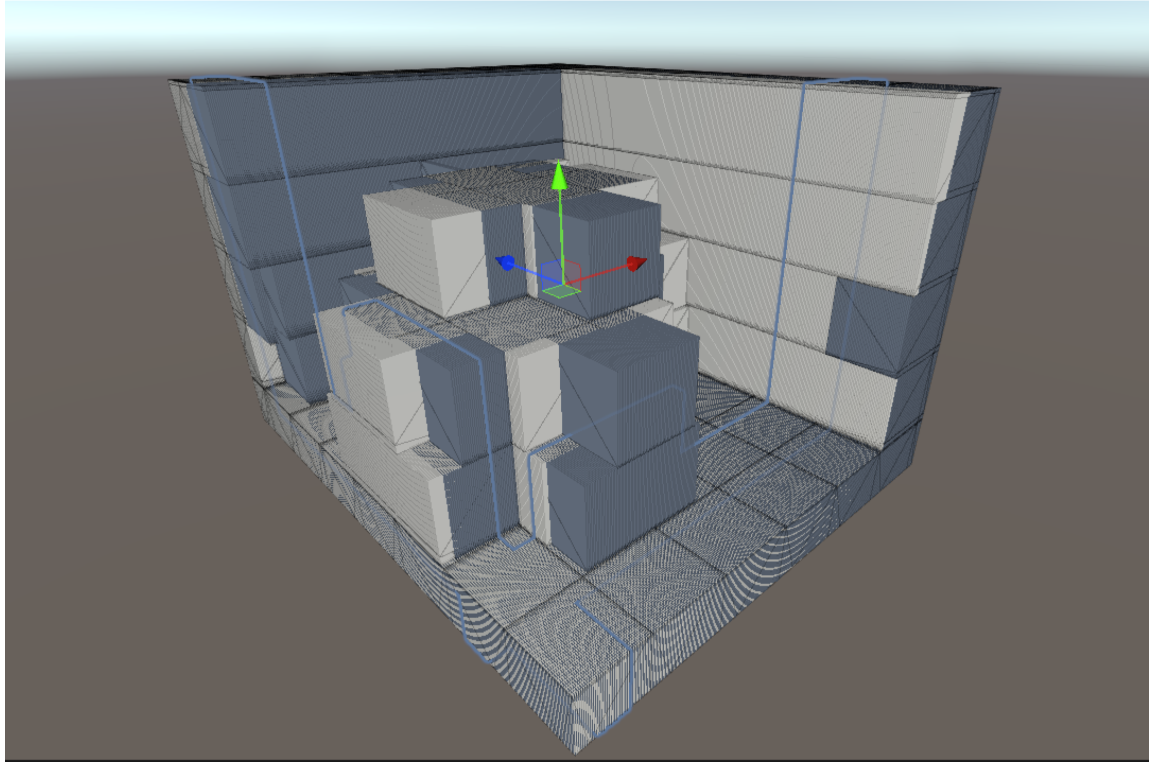
- Mesh with alpha and neural texture
- A neural renderer translates the neural colors into RGB values

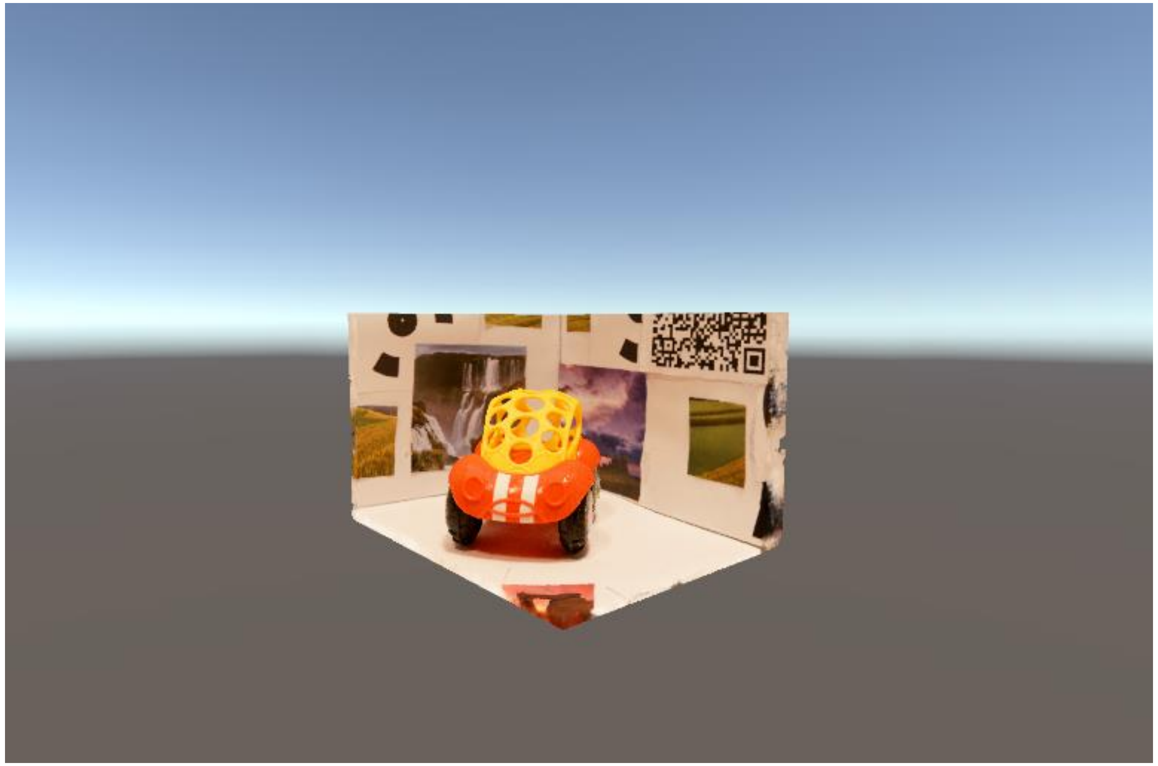
Represent continuous voxel regions

- Only coarse representation of the geometry
- Details are handled by alpha and neural texture

Example

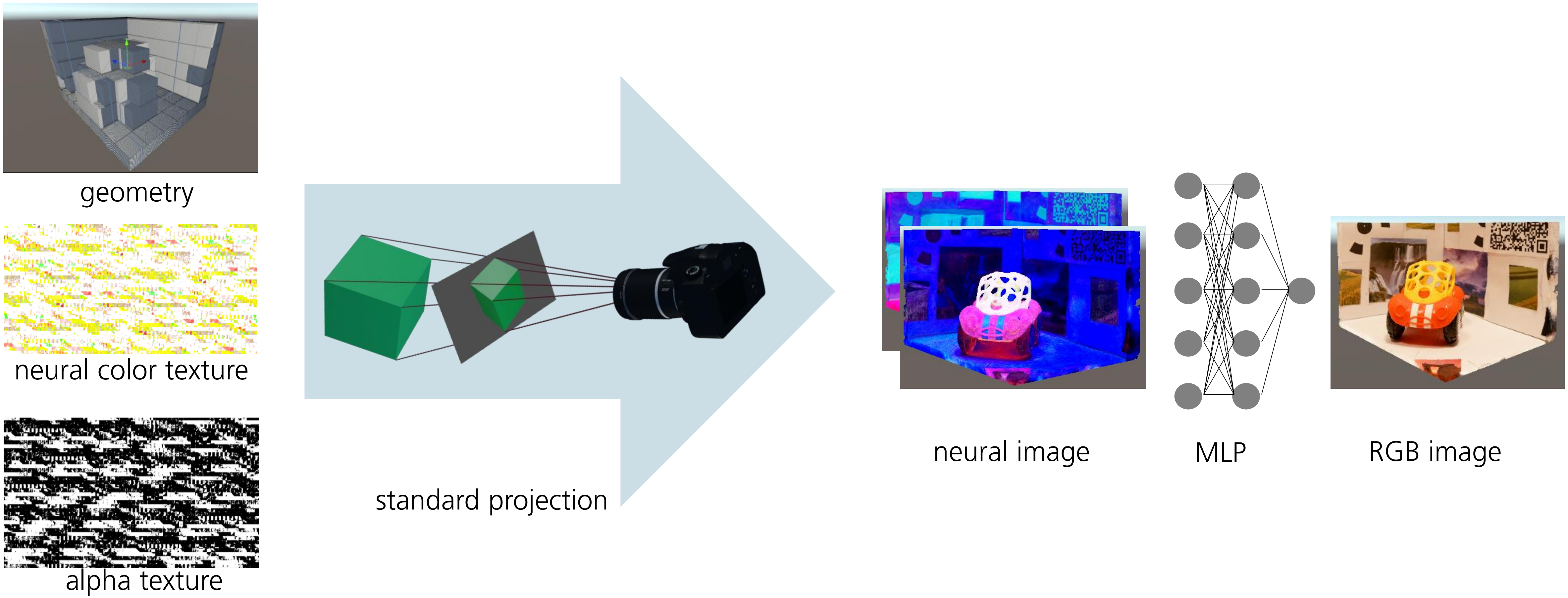
Voxel mesh rendering





Voxel mesh rendering

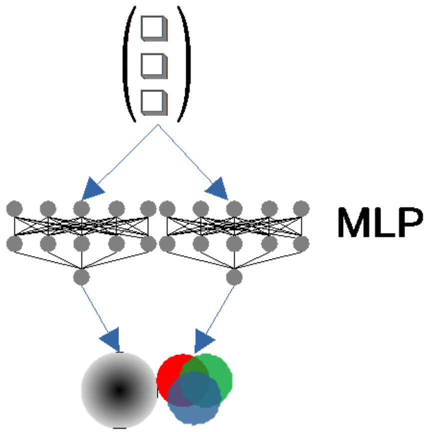
Algorithm principle



Real-time rendering in Unity

UniTorch Plugin

- Integration of libTorch into Unity
- Highly optimized execution of MLP rendering
- Support of more powerful neural networks
- Interaction with different rendering passes to support different algorithm scenarios

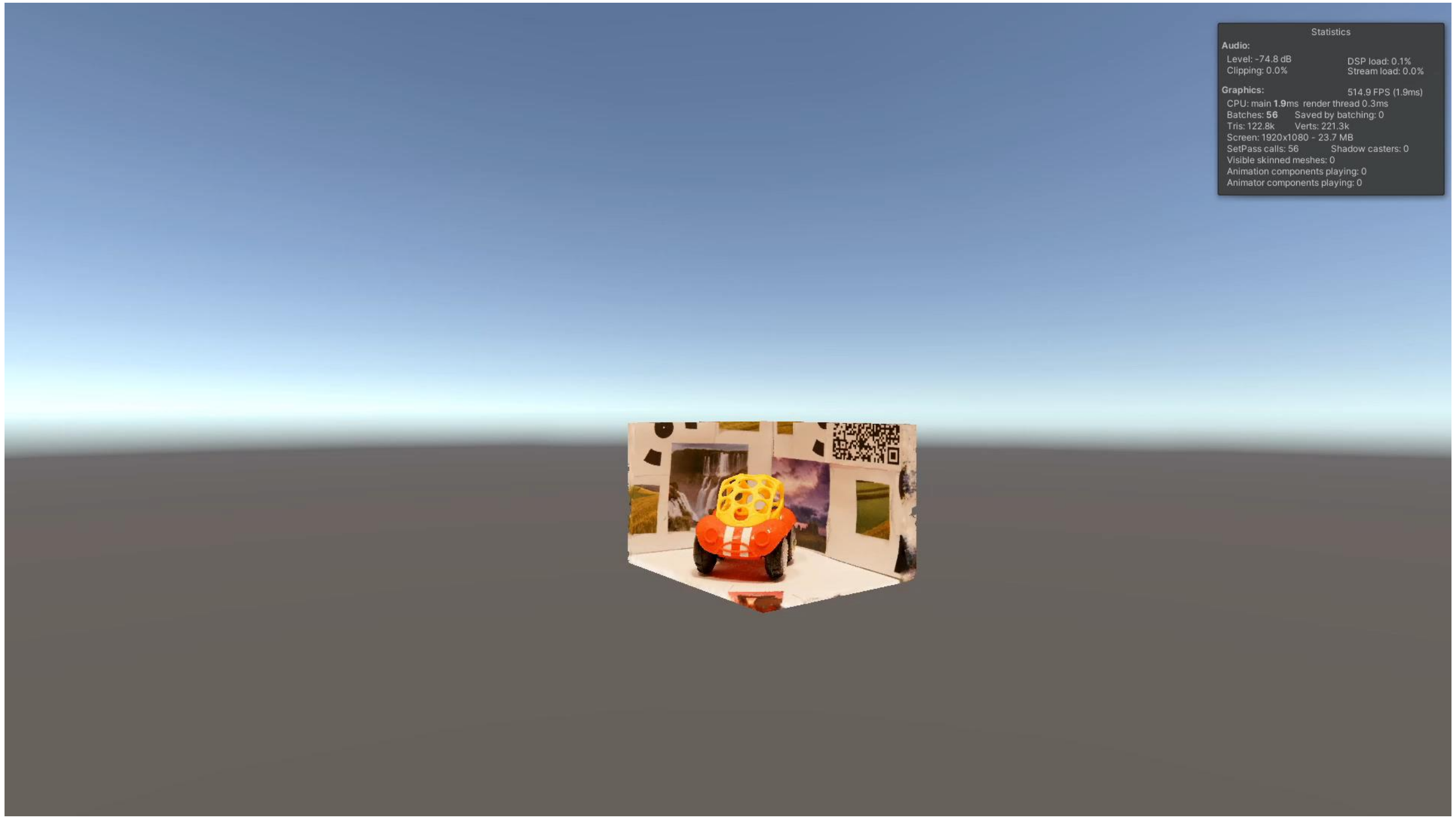


Use of shader language

- Mostly platform independent
- Efficient integration into pixel rendering pipeline
  - Fragment/pixel shader
- Easier support of different Unity rendering pipelines
  - Standard Building Pipeline
  - Universal Rendering Pipeline
- Extension for Web applications possible

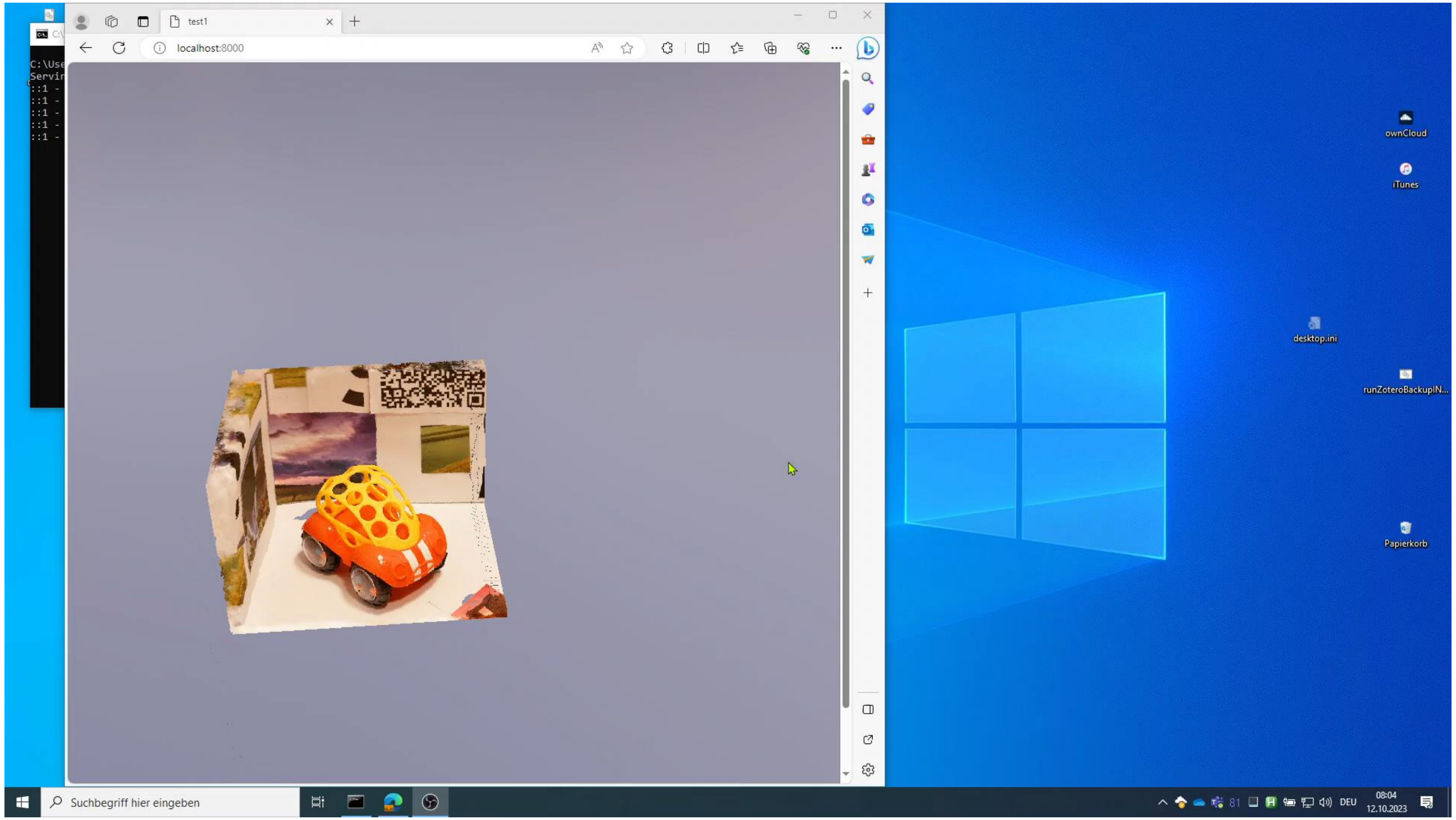
Unity plugin for voxel mesh rendering

Result video for shader-based implementation



Web browser support

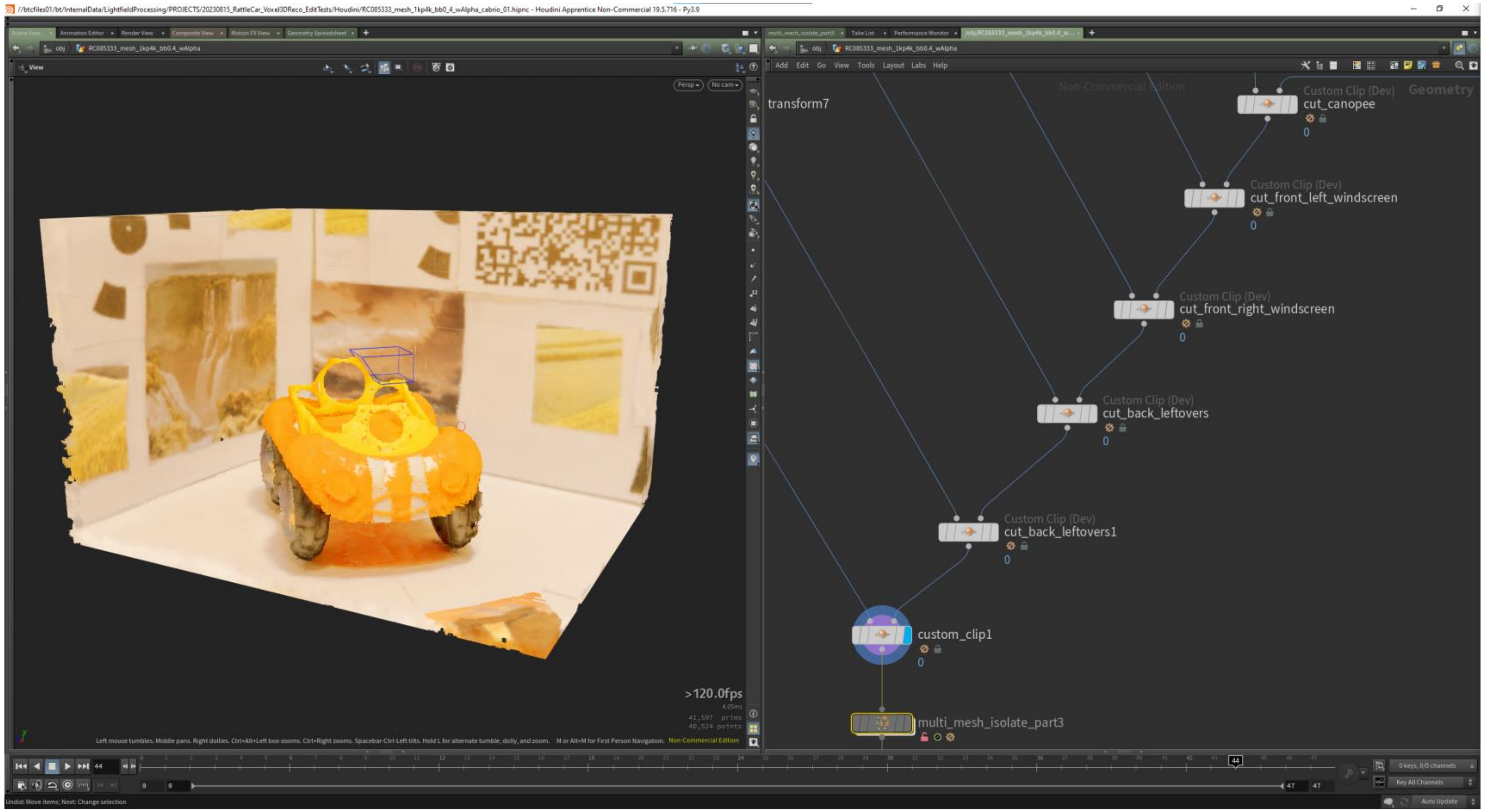
WebGPU implementation



  intern

Voxel mesh editing

Use of standard 3D editing tools



Voxel mesh editing
Rendering result

The end

Contact

Joachim.Keinert@iis.fraunhofer.de

XR Interaction Member (XR-INTERACTION.com)

- Research alliance for XR-technologies and applications
- Creation of publicly funded projects for basic technologies and applications
- Open for new complementary partners