EUVIP 2022

# DEEP 3D MODEL OPTIMIZATION FOR IMMERSIVE AND INTERACTIVE APPLICATIONS

Elena Camuffo, Federica Battisti, Francesco Pham, Simone Milani

Department of Information Engineering - University of Padova (IT)

Università degli Studi di Padova | Dipartimento di Ingegneria dell'Informazione

# Outline

- **Introduction**: 3D Model Optimization in VR/AR applications.
- **Problem Assessment**:
  - Related Work & Problem Statement
  - **Problem Analysis**: tradeoff Quality-Fluidity.
- **Development:**
  - Analysis of **Parameters and Metrics**
  - **Data Analysis** for Feature Selection
  - Deep Learning based Metrics Estimation **Pipeline**.
- **Experimental Results**
- **Conclusions**

# Introduction

- Growing **diffusion** of AR and VR systems has posed new and challenging problems.
- Systems require immersivity through:
  - **Real-time rendering** of 3D objects;
  - **High fidelity resolution** of environments;
  - **Fluid interaction** with the synthetic world.
- Adapt the **LOD** of 3D objects depending on the user's **proximity** and **interaction** with 3D objects' virtual environment.

# Related Work

- Similar techniques employed in **video transmission**:
  - Quality maximization strategy under bandwidth constraints.
  - HTTP Adaptive Streaming (HAS) packet dropping to tune the transmission stream minimizing the quality decrement.
- Optimization strategies rely on:
  - Linear programming solvers.
  - **Deep learning** solutions.
- More challenging application to 3D models in AR/VR:
  - 3D models are heterogeneous
  - They require different rendering capabilities.

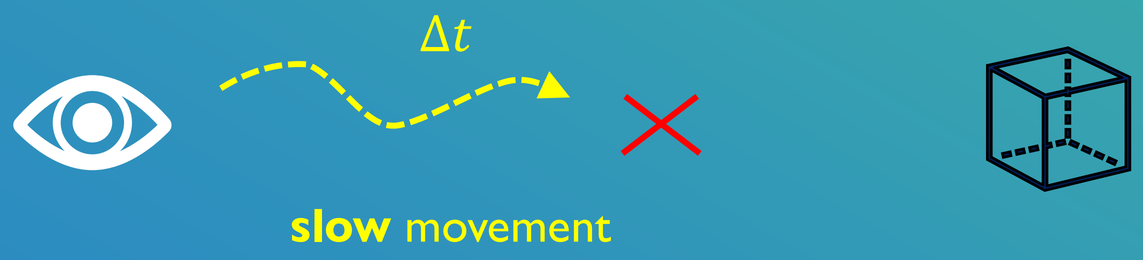
| **video** | **3D models** |
|---|---|
| TX channel | perspective view |
| Rate ↑ | Quality ↑ |
| Resolution ↓ | LOD ↓ |

# Problem Statement



- The LOD $L$ of a 3D mesh model in AR/VR applications strongly affects the system efficiency and the quality perceived by the end user.

**Fluidity**




- **Quality-based optimization** is modeled as a dual problem of quality maximization given complexity, constraints on LOD level:

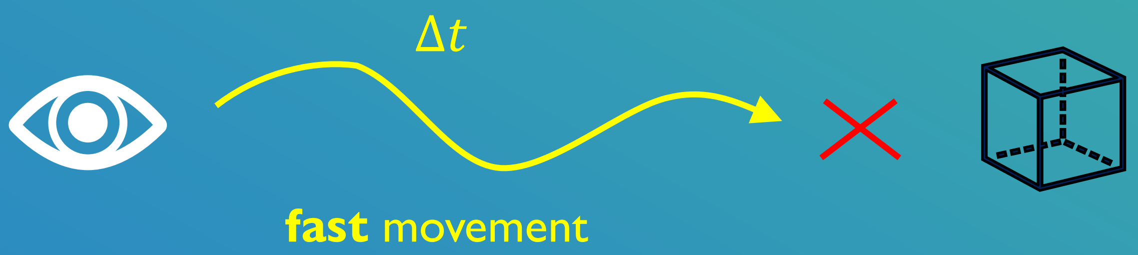
$$\min L \qquad s.t. \quad Q(L, \boldsymbol{p}, \boldsymbol{o}) > Q_0$$
$$FPS(L) > FPS_0$$

# Problem Statement



- The LOD $L$ of a 3D mesh model in AR/VR applications strongly affects the system efficiency and the quality perceived by the end user.

**Fluidity**




- **Quality-based optimization** is modeled as a dual problem of quality maximization given complexity, constraints on LOD level:

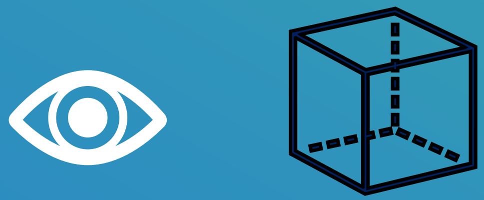
$$\min L \quad s.t. \quad Q(L, \boldsymbol{p}, \boldsymbol{o}) > Q_0$$
$$FPS(L) > FPS_0$$

# Problem Statement



- The LOD $L$ of a 3D mesh model in AR/VR applications strongly affects the system efficiency and the quality perceived by the end user.

**Fluidity** **Quality**



**close** 3D object

- **Quality-based optimization** is modeled as a dual problem of quality maximization given complexity, constraints on LOD level:

$$\min L \qquad s.t. \quad Q(L, \mathbf{p}, \mathbf{o}) > Q_0$$
$$FPS(L) > FPS_0$$

# Problem Statement



- The LOD $L$ of a 3D mesh model in AR/VR applications strongly affects the system efficiency and the quality perceived by the end user.

**Fluidity** **Quality**

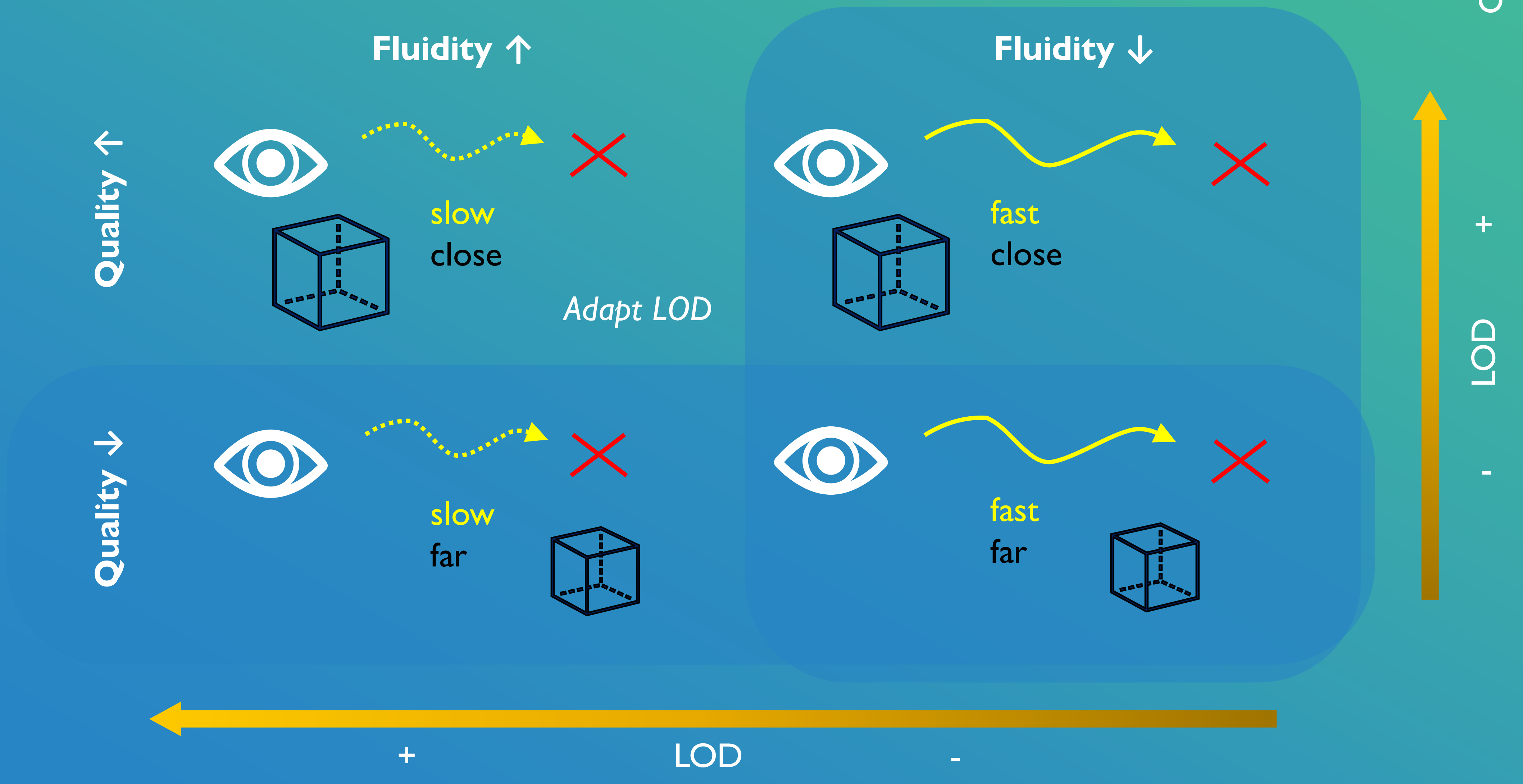
**far** 3D object

- **Quality-based optimization** is modeled as a dual problem of quality maximization given complexity, constraints on LOD level:

$$\min L \quad s.t. \quad Q(L, \boldsymbol{p}, \boldsymbol{o}) > Q_0$$
$$FPS(L) > FPS_0$$

# Problem Analysis





# Parameters & Metrics

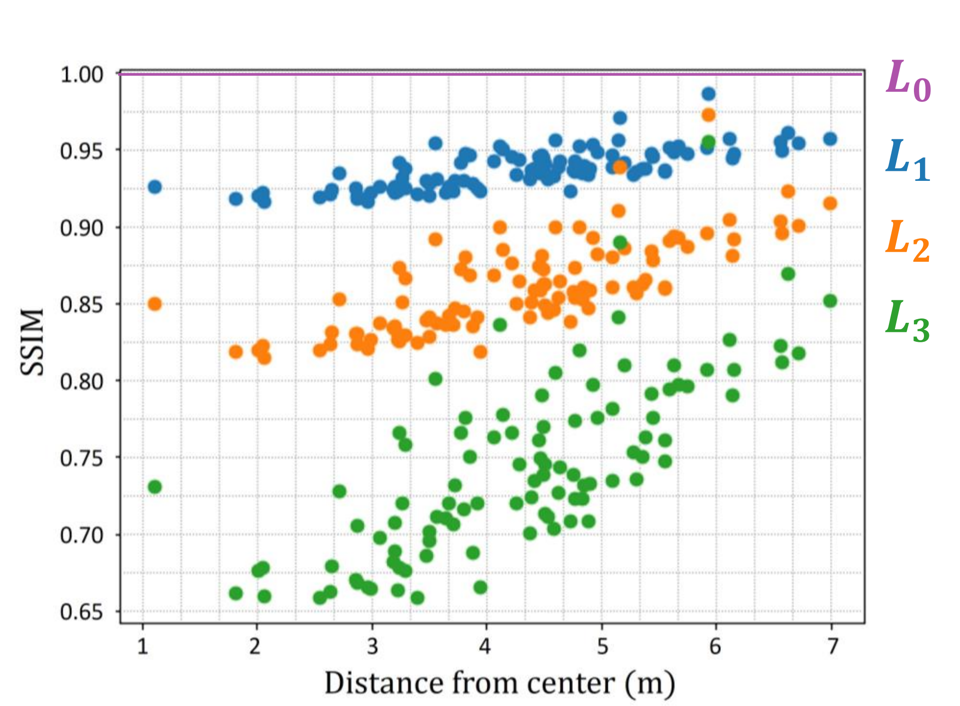


INTRA-VIEW SSIM

- Suppose the viewer is **static.**
- Measure quality varying **LOD**.

$$\mathrm{SSIM}(L_0, L_t), \quad t = 0,1,2,3$$

- Keeping LOD fixed:
  - Greater distance ↑, greater SSIM ↑
- Keeping distance fixed:
  - Greater LOD $L$ ↑, greater SSIM ↑





$$L_0 > L_1 > L_2 > L_3$$

# Parameters & Metrics



### INTRA-VIEW SSIM

- Suppose the viewer is **static.**
- Measure quality varying **LOD**.

### INTER-VIEW SSIM

- Suppose the viewer is **moving.**
- Measure quality varying **position**.

# Parameters & Metrics



INTRA-VIEW SSIM

- Suppose the viewer is **static.**
- Measure quality varying **LOD**.

INTER-VIEW SSIM

- Suppose the viewer is **moving.**
- Measure quality varying **position**.

VERTEX COUNT

- Total number of vertices of a mesh.
- Measure complexity of the mesh.

# Parameters & Metrics



INTRA-VIEW SSIM

- Suppose the viewer is **static.**
- Measure quality varying **LOD**.

INTER-VIEW SSIM

- Suppose the viewer is **moving.**
- Measure quality varying **position**.

VERTEX COUNT

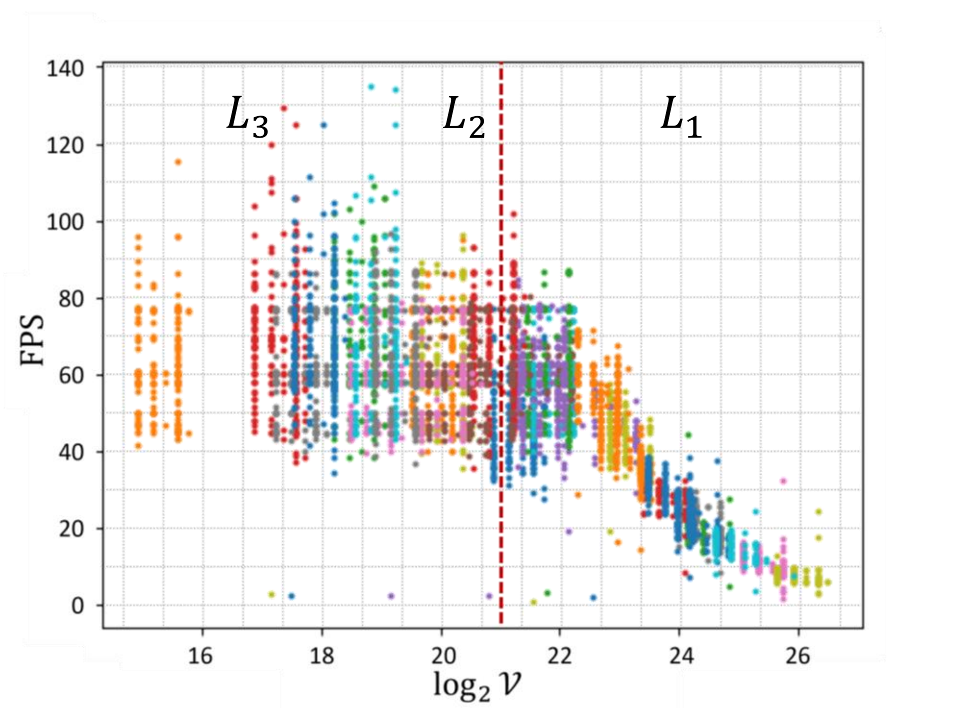
- Total number of vertices of a mesh.
- Measure complexity of the mesh.

FRAME VERTEX COUNT

- Total number of vertices **from a view**.
- Measure rendering power needed related to complexity.

# Parameters & Metrics






different colors = different models

- **FPS is dependent on the device**: some screens cap FPS to 60Hz.
- Measure fluidity with another metric: frame vertex count is directly related to FPS, without suffering capping.

FRAME VERTEX COUNT

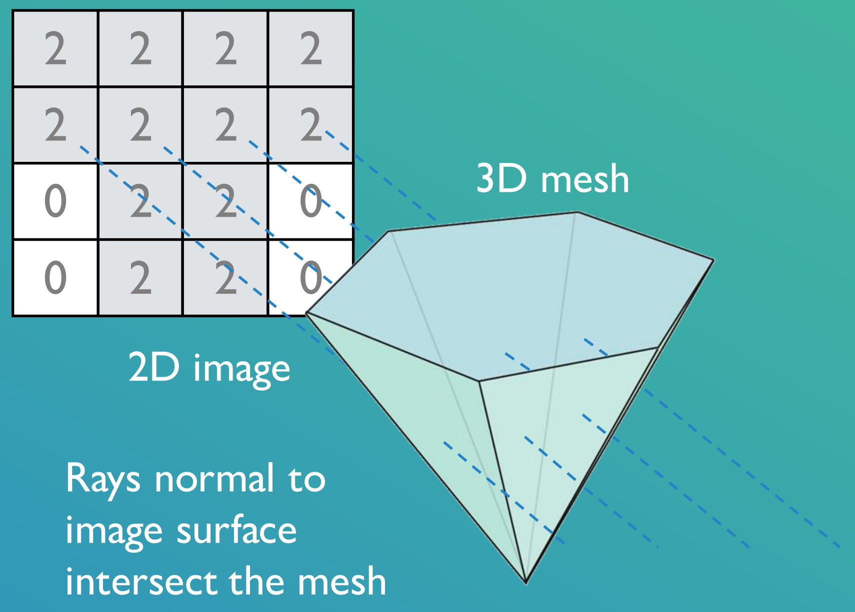
- Total number of vertices **from a view**.
- Measure rendering power needed related to complexity.

# Parameters & Metrics



ORTOGRAPHIC TRIANGLE COUNT PROJECTIONS

- Projection of the 3D object along x-y-z axes.
- Each pixel value corresponds to the number of faces intersected by the normal to the pixel center.
- Outline a **shape** of the object defining its **complexity**.



# Objectives

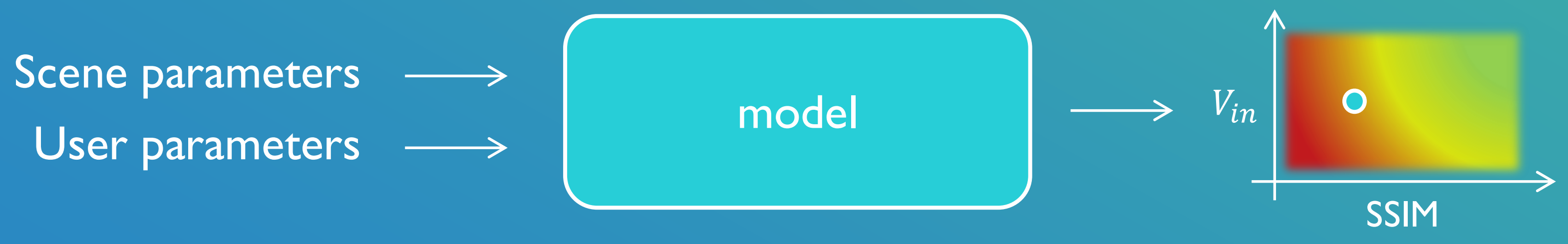
- **Optimize LOD** adaptively, maximizing the **quality.**

OBJECTIVES:

- Keep sufficient **quality:**
  - adapt the LOD to the **distance** of viewer from 3D object.
- Keep sufficient **fluidity:**
  - adapt the LOD depending on the **velocity** of movement.

DEEP NEURAL NETWORK:

- For 3D model at given LOD estimate:
  - **Inter-view SSIM:** actual quality.
  - **Frame vertex count:** actual 3D model complexity.




No original reference scene > you want to know if **quality is ok or not** > if not decrease LOD

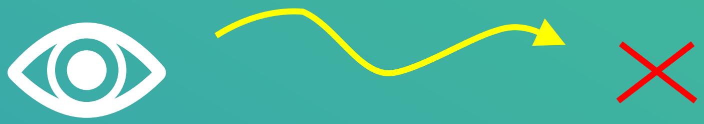
# Setup Analysis



CONDITIONS:

1. The viewer is **moving**.

LOD $L_\tau$



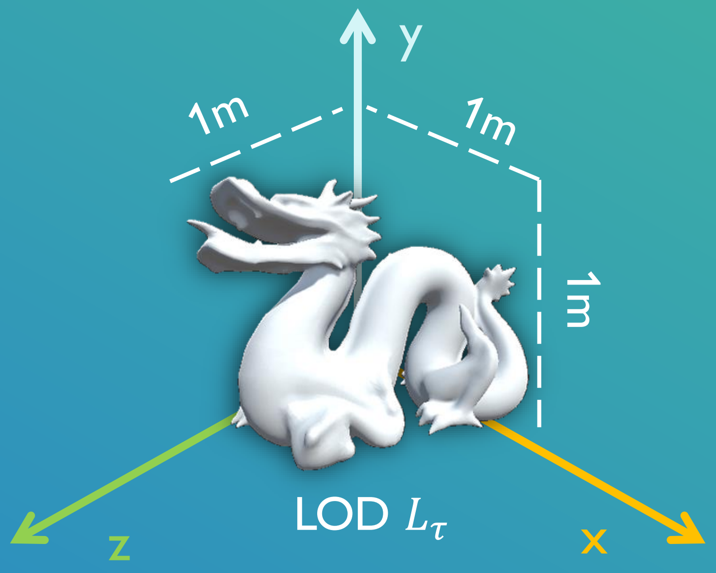
**Move Fast** → decrease LOD ↓
**Move Far** → decrease LOD ↓

# Setup Analysis



CONDITIONS:

1. The viewer is **moving**.
2. The 3D object is placed in the **axes' origin**.



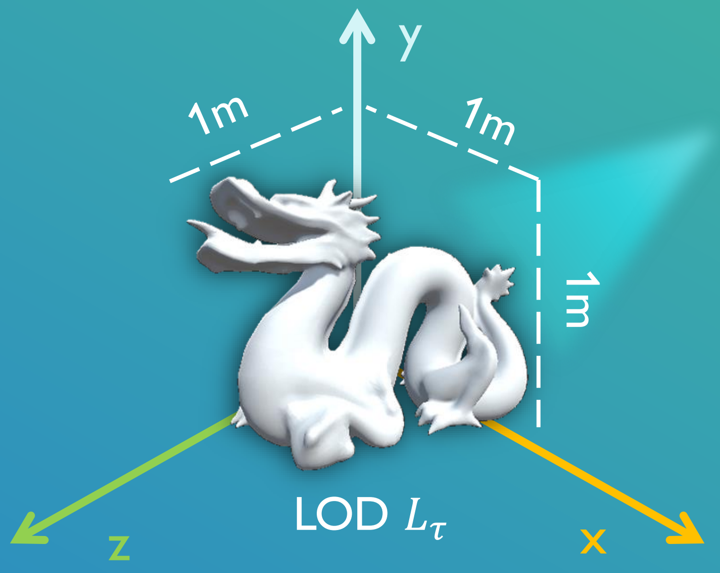

**Move Fast** → decrease LOD ↓
**Move Far** → decrease LOD ↓

# Setup Analysis



CONDITIONS:

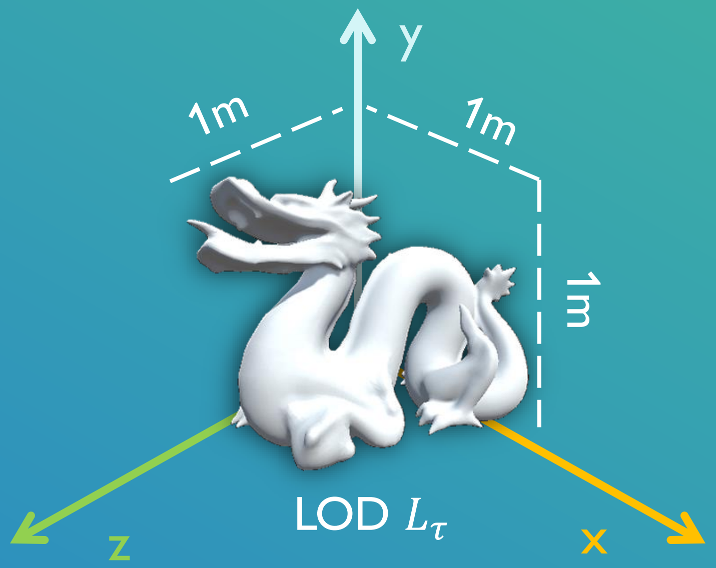
1. The viewer is **moving**.
2. The 3D object is placed in the **axes' origin**.
3. The viewer always looks at the 3D object (**orientation fixed**).




**Move Fast** → decrease LOD ↓
**Move Far** → decrease LOD ↓

# Dataset Analysis

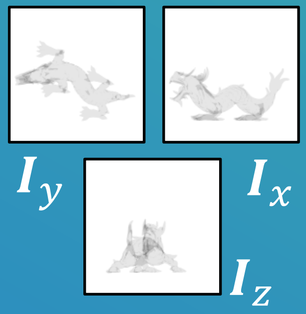


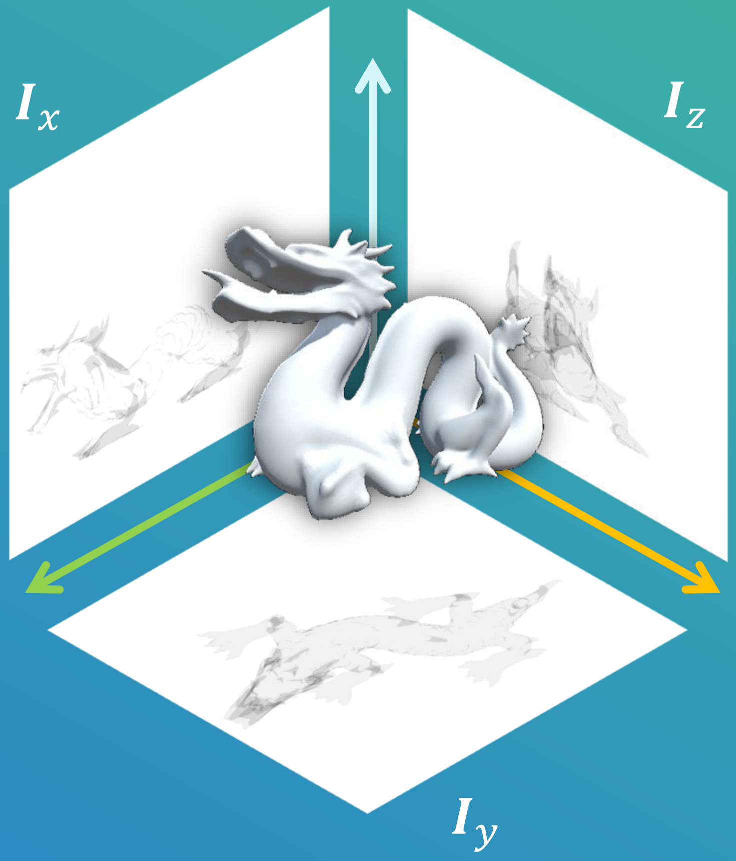


# Dataset Analysis



OTC-Projections

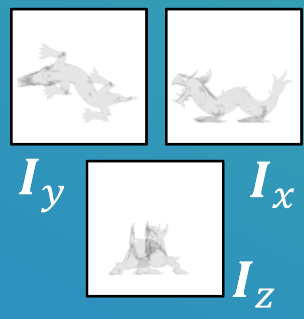


# Dataset Analysis



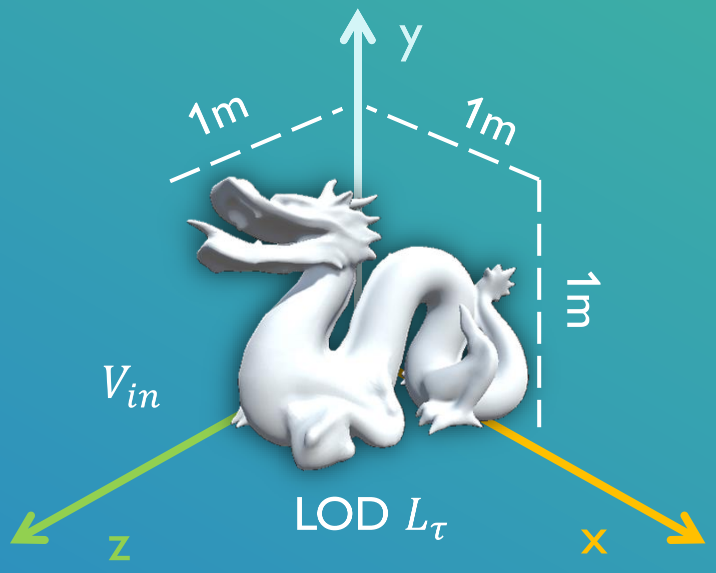
OTC-Projections




Reference Position
$p_0 = (x_0, y_0, z_0)$

Target Position
$p_t = (x_t, y_t, z_t)$

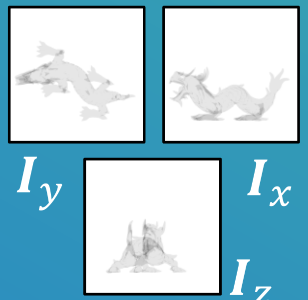
Total vertex
count $V_{in}$




# Dataset Analysis



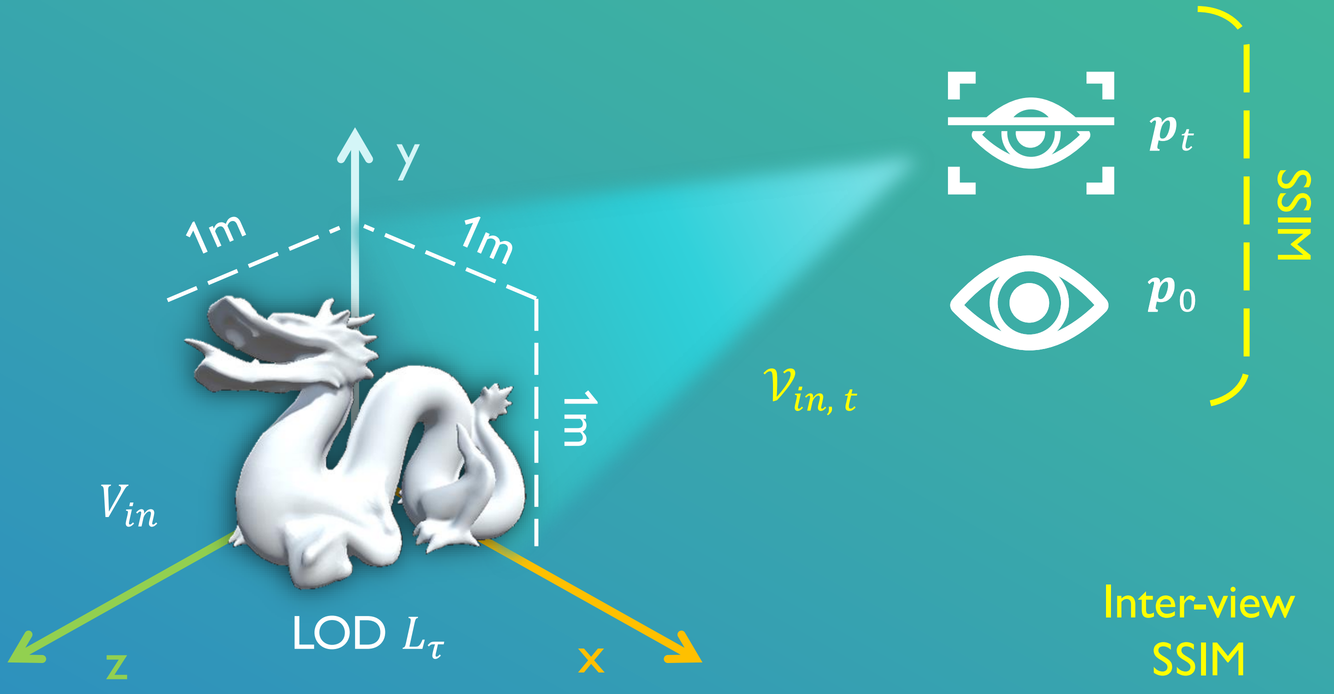
OTC-Projections




Reference Position
$p_0 = (x_0, y_0, z_0)$

Target Position
$p_t = (x_t, y_t, z_t)$

Total vertex count $V_{in}$

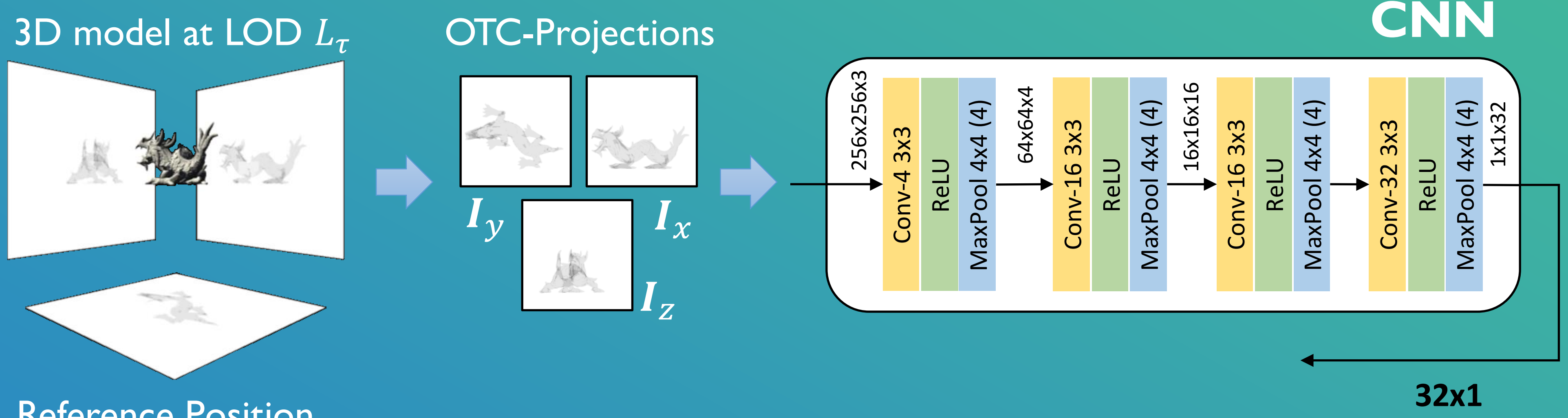



Inter-view SSIM

Frame vertex count $\mathcal{V}_{in,\,t}$

# Estimation Pipeline






Reference Position
$\boldsymbol{p}_0 = (x_0, y_0, z_0)$

Target Position
$\boldsymbol{p}_t = (x_t, y_t, z_t)$

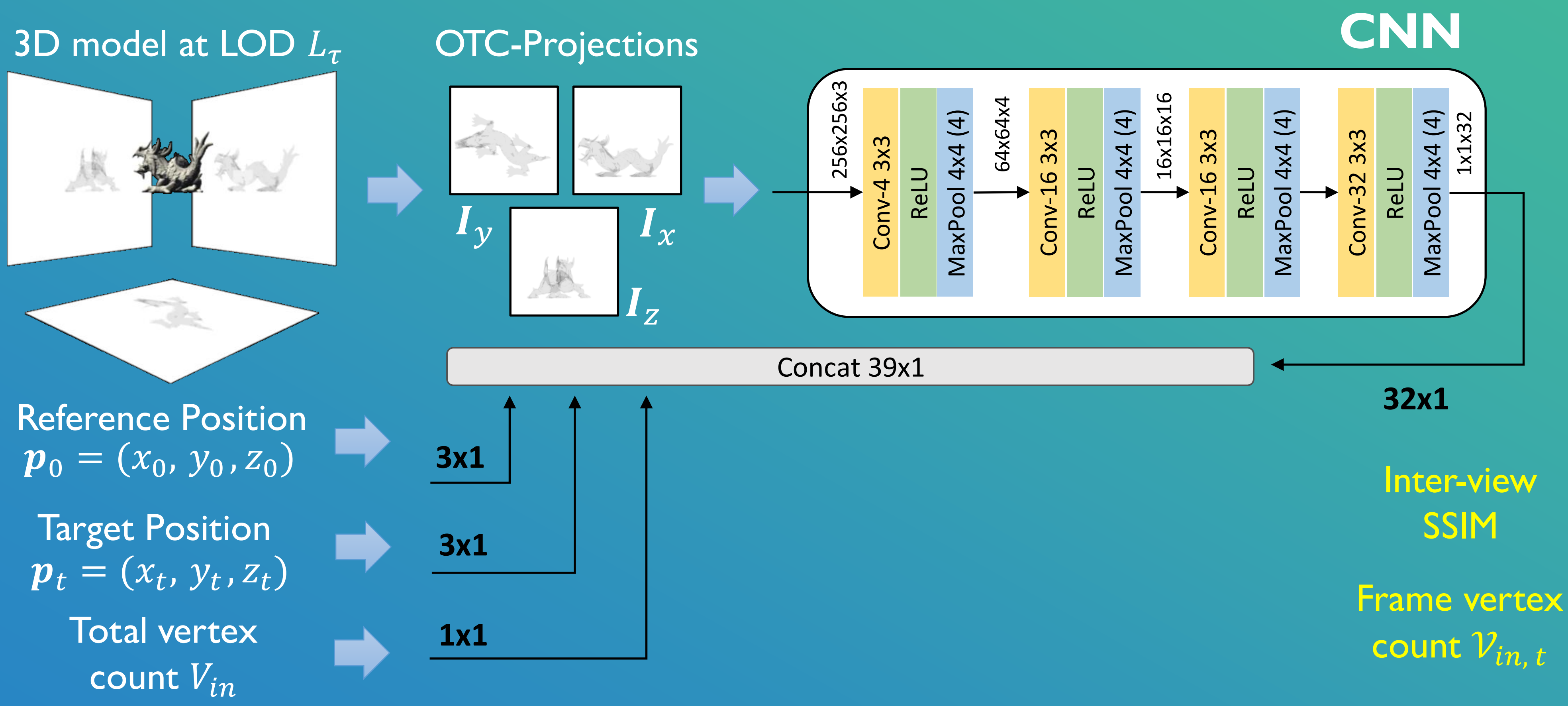
Total vertex
count $V_{in}$

Inter-view
SSIM

Frame vertex
count $\mathcal{V}_{in,\,t}$

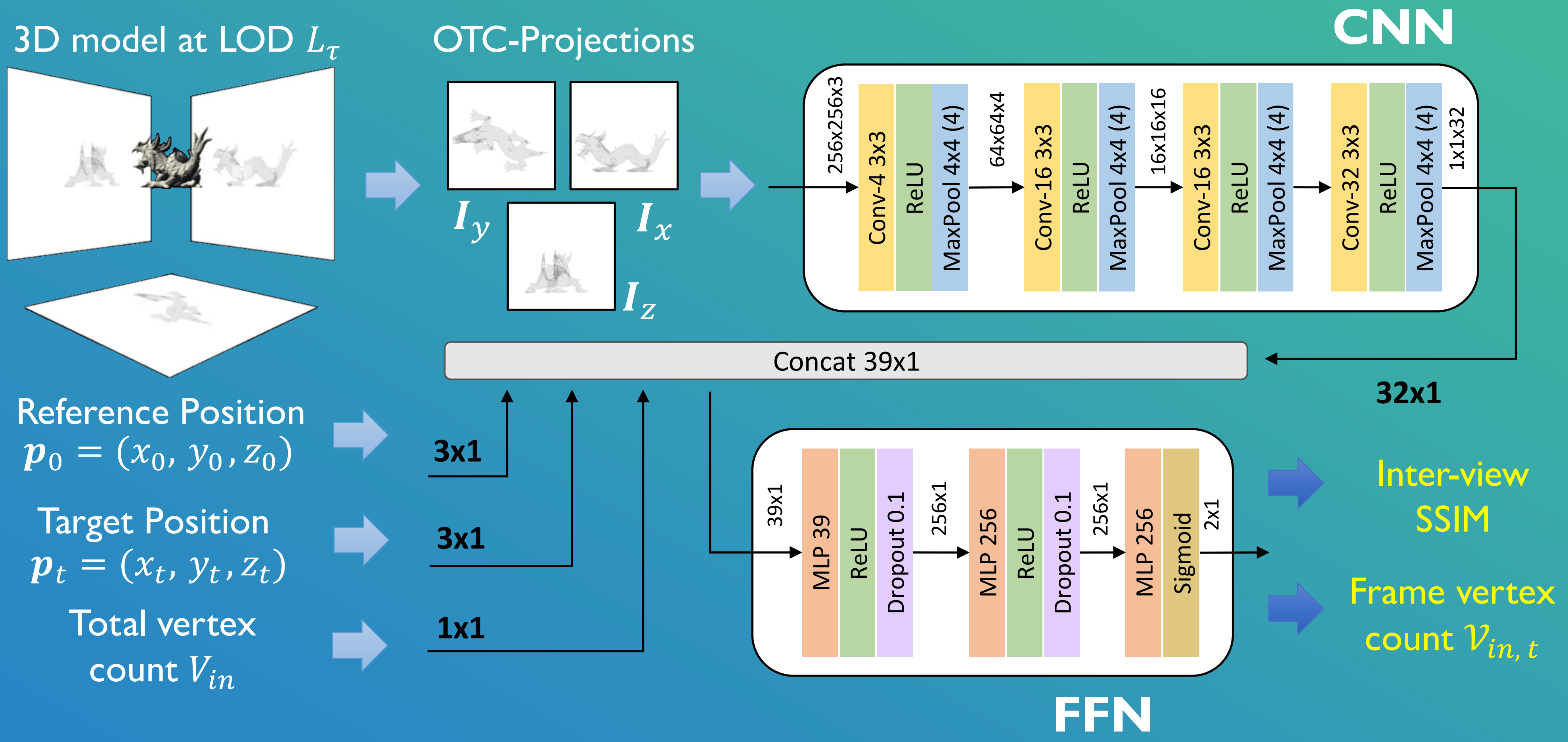
# Estimation Pipeline





# Estimation Pipeline





# Results



- Tested usability through **Unity3D DEMO simulation**.
- Training time ~2.6s per epoch, Inference time ~1ms per model.

**Varying LOD**

| | Inter-View SSIM | | | $\mathcal{V}_{in}$ | | |
|---|---|---|---|---|---|---|
| LOD | True | Pred | Δ↓ | True | Pred | Δ↓ |
| $L_0$ | 0.9001 | 0.8660 | **0.0341** | 0.7222 | 0.7270 | **0.0048** |
| $L_1$ | 0.9029 | 0.8704 | **0.0325** | 0.6969 | 0.6894 | **0.0075** |
| $L_2$ | 0.9000 | 0.8675 | **0.0325** | 0.6701 | 0.6602 | **0.0099** |
| $L_3$ | 0.9004 | 0.8711 | **0.0293** | 0.6450 | 0.6340 | **0.0110** |
| | 0.9008 | 0.8687 | **0.0321** | 0.6850 | 0.6766 | **0.0084** |

**Fixed LOD**

| | | Inter-View SSIM | | | $\mathcal{V}_{in}$ | | |
|---|---|---|---|---|---|---|---|
| $p_0$ | $p_t$ | True | Pred | Δ↓ | True | Pred | Δ↓ |
| f | f | 0.9124 | 0.8742 | **0.0382** | 0.7857 | 0.7073 | **0.0784** |
| c | f | 0.8476 | 0.7612 | **0.0864** | 0.6876 | 0.7046 | **0.0170** |
| f | c | 0.8569 | 0.7568 | **0.1001** | 0.6706 | 0.6977 | **0.0271** |
| c | c | 0.8583 | 0.8159 | **0.0424** | 0.6818 | 0.6995 | **0.0177** |

- Predictions respect the **behavior** of True values:
  - LOD ↓, $\mathcal{V}_{in}$ ↓.
  - **Inter-View SSIM** varies with $p_0$, $p_t$: greater if viewer moves from far to far (f > f) or from close to close (c > c).

# Conclusions

- Deep learning based approach **optimizing the visualization of 3D objects** in an interactive scenario, **adaptively** selecting the most suitable set of parameters.
- Future developments:
  - extend the **number of 3D models**, and their complexity.
  - considering attributes, e.g., **texture and normal maps**.
  - extend to **subjective tests**.


THANK YOU
Any questions?