# TOWARDS MODELLING OF VISUAL SALIENCY IN POINT CLOUDS FOR IMMERSIVE APPLICATIONS


| | |
|---|---|
| Evangelos Alexiou | evangelos.alexiou@epfl.ch |
| Peisen Xu | pxu008@e.ntu.edu.sg |
| Touradj Ebrahimi | touradj.ebrahimi@epfl.ch |

# Introduction



- Modelling visual attention is important in several applications in computer graphics and signal processing
- Extensive experimentation with 2D imaging
- Several predictors of salient regions on 3D models (meshes and point clouds)
- Limited number of eye-tracking experiments to provide ground truth data for 3D models
  - Unnatural way of content consumption
  - No user engagement

**In this study:**

- Point cloud models under inspection
- Extend state-of-the-art by tracking visual attention in 6-DoF VR experience
- Task-dependent protocol to motivate exploration
- First step towards visual saliency in immersive experiences

# Apparatus



- **HTC Vive Pro** (**headset**)
  - Screen: 2440x1600 px per eye, 615 ppi
  - Field of view: 110º
  - Refresh rate: 90 Hz
- **Pupil Labs** (**eye-tracking**)
  - Binocular add-on cameras
  - Independent gaze tracking
  - Tracking frequency: 120 Hz
- **Unity** (**development platform**)
  - Design of the virtual scene
  - Capture head-related data from Vive Base Stations installed in the room
  - Connectivity with Pupil Labs SDK using network messages for eye-related data
  - Synchronization of both streams with the rendering frame rate
  - Recording data


# Models



- Static point cloud contents; different content types
- Different acquisition techniques and number of points → **voxelization**

objects

**d: 10-bit p: 814.474** | **d: 10-bit p: 1.181.016** | d: 12-bit p: 272.684 | **d: 10-bit p: 636.127** | d: 12-bit p: 1.009.132 | d: 12-bit p: 499.660

human figures

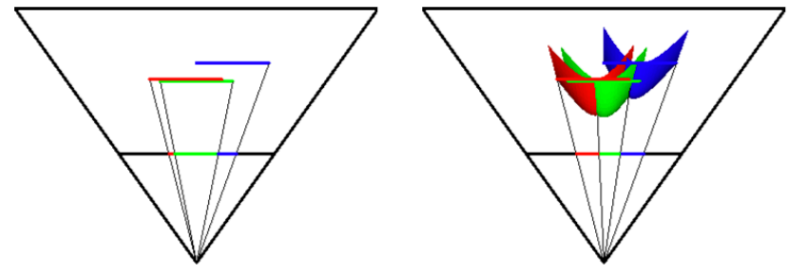
d: 10-bit p: 857.966 | d: 10-bit p: 805.285 | d: 10-bit p: 757.691 | d: 10-bit p: 1.089.091 | **d: 10-bit p: 1.553.937** | **d: 12-bit p: 811.019**

By default voxelized
**Manually voxelized**

# Rendering



- Pcx importer* to load point clouds in Unity
  - Convert a point cloud to a mesh-based object
- For a fine balance between complexity, fidelity and water-tightness, custom implementations:
  - Interpolation shader using paraboloids as primitive elements [1]
  - Adaptive size of primitives based on *k* nearest neighbors [2]



(a) without depth offset (b) with depth offset

**Interpolation shader [3]**

**Model representation using quad (left) against paraboloid (right) primitives**

* https://github.com/keijiro/Pcx

# VR scene



- Non-distracting square virtual room with mid-grey walls (extending ITU-T P.910 recommendation [3])
- Models scaled at a natural size and placed at the center of the room
  - Smaller objects placed on a stage
- Point light source with real-time lighting
- Manually generated shadows simulating first order light reflection
- Subjects could navigate either physically or by using the HTC Vive Controllers that allowed motion control (i.e., teleport and rotation)

**Object on a stage**

**Human figure**

# Test design



- **Task-dependent** inspection:
  - "Examine a set of models. After visualization, order them based on your preference, according to a criterion of your preference"
  - *No time* limitations
  - *No memorization* of the models required

**Training** → **Test** (Round 1 → Round 2)

Training:

- Dummy content
- Subjects got familiarized with equipment and navigation

Test:

- Split in two rounds, each with a different type of contents (i.e., objects, human figures)
- 5-minute mandatory break in between to avoid fatigue

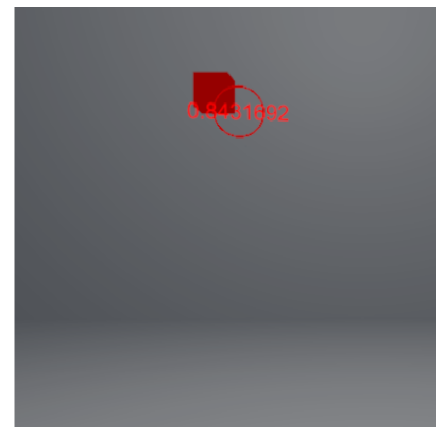
# Session steps



**For each session:**

1. External calibration › Pupil Labs SDK using 7 points in 2D calibration mode
2. Inspection of model › Visualization of a content
3. Internal error profiling › Average angular error at 9 regularly spaced markers, at the end of a session (account for HMD slippage)

**External calibration**



**Internal profiling**

# Data pre-processing



- **Recorded gaze samples:**
  - Left and right gaze positions along with corresponding quality values
  - Middle gaze position is computed as the average of left and right

*For each gaze sample:*

**Discard** blinks, low-quality and out-of-range gaze positions

If at least one is discarded, middle gaze cannot be used

→

**Estimate** average angular error of remaining gaze samples

Barycentric interpolation using angular errors of highest quality markers during internal profiling

→

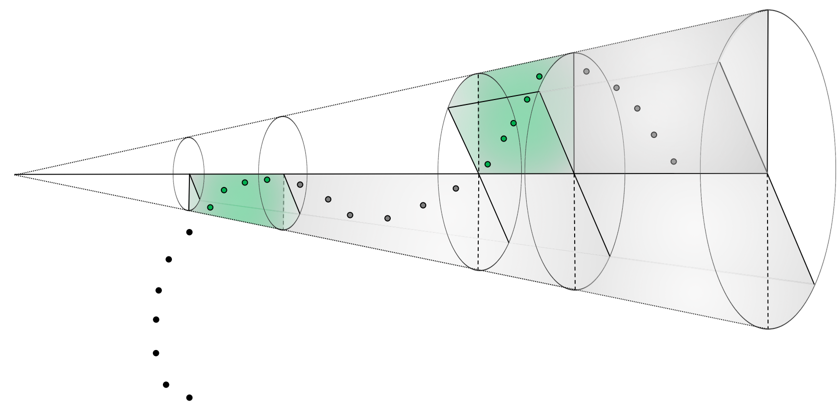
**Keep** gaze positions of lowest angular error

Ensure highest quality of data

# Heat map generation



1. **Fixation point** estimation:
   - Dispersion-based algorithm [4] with adjusted window of 150 ms and 1° of max dispersion
   - Consecutive samples of same gaze type
   - Angular error based on barycentric interpolation
2. **Gaze vector** definition:
   - Between average camera and gaze position in world coordinated over the fixation's duration
3. **Attention region** identification:
   - Cast a cone towards gaze vector
   - Identify frontal points by splitting the cone in sectors and by setting depth threshold
4. **Importance weight** assignment:
   - Gaussian weighting of point $x$, that belongs to a fixation $f$ as a function of duration $t$, distance between user and model $d$, angular error $\theta$, and point deviation from gaze vector $p$



$$w(x) = \frac{t}{\sqrt{2\pi\sigma_f^2}} e^{-\frac{p^2}{2\sigma_f^2}}, \quad \sigma_f = d \cdot \tan\theta$$

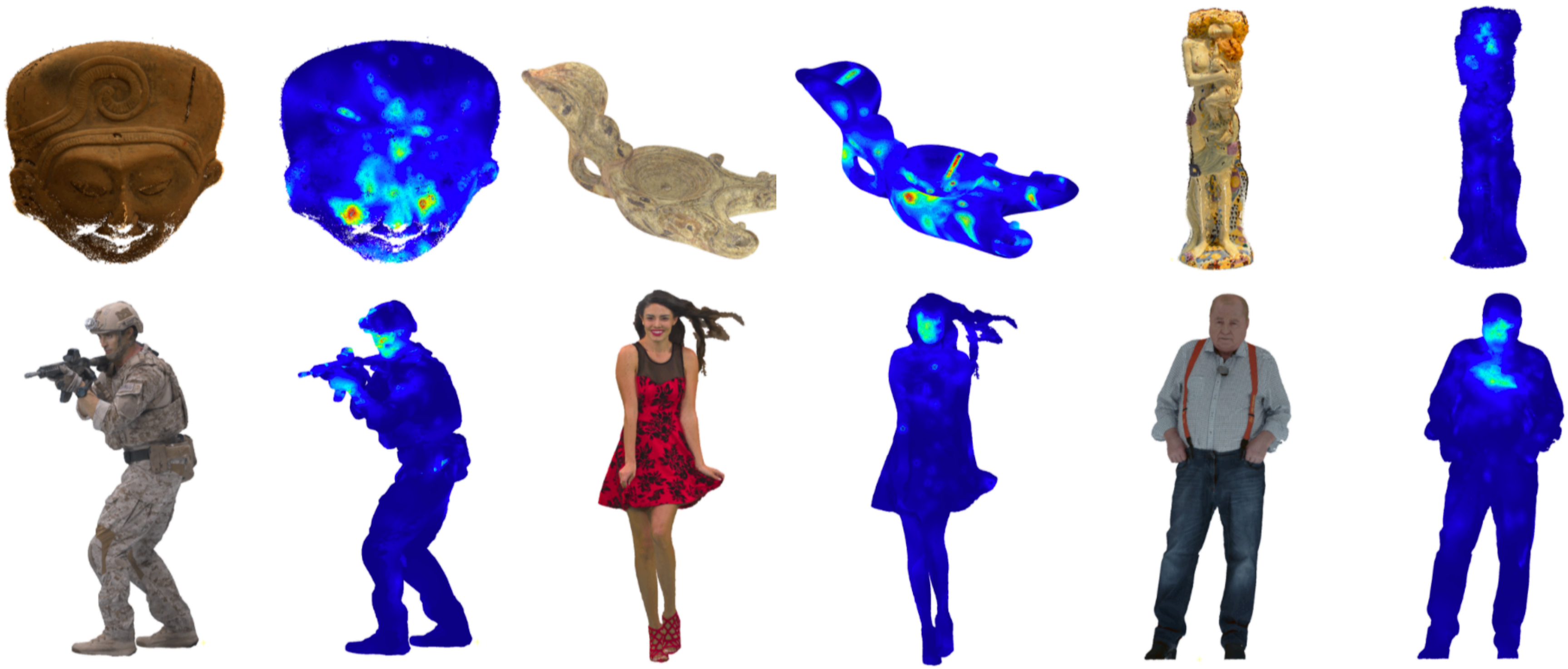
# Evaluation of a session (track)



- **Proposed metrics:**
  - Tracking accuracy (< 17.5%)
  - In-range fixations (> 75%)
- **Valid session:**
  - If both conditions are satisfied
- **Statistics of valid sessions:**
  - 73% of the sessions were used
    - *10% low-confidence gaze positions*
    - *92% in-range fixation points*
  - $44.1 \pm 7$ avg number of fixations per model
  - $259.1 \pm 30.5$ ms avg duration
  - $1.9^{\circ} \pm 0.84^{\circ}$ angular error
- **Visual attention maps:**
  - Fixation density maps
  - Fusion of importance weights from fixations on models from valid sessions

# Visual attention maps





- Low-level features (i.e., edges and contrast)
- High-level features (i.e., faces)
- Text and unexpected objects

# Users statistics



- The majority were naïve users of VR
- Average interaction time
  - **60.9 ± 10.7 sec** for objects
  - **56.4 ± 4.6 sec** for human figures
- More time at bigger and more complicated models
- Inspection from mid- and closed-range distances
- Visual quality: 3.7 out of 5*
- Quality of experience: 4.35 out of 5*
- Discomfort level: 1.15 out of 3**
- Criteria of preference:
  - **Realistic** and **smoothness** for objects
  - **Realistic** and **details** for human figures


* 5: *Excellent*, 4: *Good*, 3: *Fair*, 2: *Poor*, 1: *Bad*

** 1: *No*, 2: *Mild*, 3: *Strong*

# Conclusions



- **First attempt** for an eye-tracking experiment in a 6-DoF task-dependent scenario in VR
- We propose a methodology to exploit lowest-error gaze positions based on per-session profiling
- We propose a methodology to identify and weight fixations
- **Dataset** publicly available:
  - Head plus eye data
  - Scripts to prepare contents
  - Scripts to compute statistics

mmspg.epfl.ch/visual-attention-point-clouds/

# Discussion



Thank you!

Multimedia Signal Processing Group
EPFL

http://mmspg.epfl.ch/

Evangelos Alexiou
evangelos.alexiou@epfl.ch

# References



[1] M. Schütz and M. Wimmer, "High-quality point-based rendering using fast single-pass interpolation," *2015 Digital Heritage*, Granada, 2015, pp. 369-372.

[2] E. Alexiou and T. Ebrahimi, "Exploiting user interactivity in quality assessment of point cloud imaging," *2019 Eleventh International Conference on Quality of Multimedia Experience (QoMEX)*, Berlin, Germany, 2019, pp. 1-6.

[3] ITU-T P.910, "Subjective video quality assessment methods for multimedia applications," International Telecommunications Union, Apr 2008.