

Evaluation of Virtual Reality based Mesh Saliency Maps

Abschlussvortrag zur Masterarbeit

Vortragender: Richard Metzler

Aufgabensteller: Prof. Dr. Dieter Kranzlmüller

Betreuer: Markus Wiedemann

Datum des Vortrags: 18.09.2017



Goal of this work

- Evaluation of differences between automatically computed and through user selections determined important parts of 3D objects
- Gathering data and discussing differences

Background

- Automated complexity reduction of 3D objects for real-time rendering
- Maintain perceived quality and detail as much as possible

Introduction

Terms

- Mesh Saliency
 - **Prominence** of surface points of objects in 3D space
 - Measure of **perceived importance**, based on fundamentals of human cognition¹
- Immersive Virtual Reality environments
 - Virtual scene is **perceived** as real, actual stimuli are overshadowed
- Spatial indexing structures
 - Allow efficient operations on 3D data
 - Quad- and Octrees, R-trees...



<http://www.digitaleng.news/de/polygonal-mesh-library-for-postprocessing-3d-scan-data/>
- visited 07-06-2017

¹ Itti, Laurent, Christof Koch, and Ernst Niebur. "A model of saliency-based visual attention for rapid scene analysis." IEEE Transactions on pattern analysis and machine intelligence 20.11 (1998): 1254-1259.



- Comparison of vertices to their **surrounding** to determine their perceived importance
 - Critical value: **curvature**
 - Scalable via **radius** determining the surrounding
- Relevant for automatic **complexity reduction** (number of vertices and polygons) in the context of real-time rendering

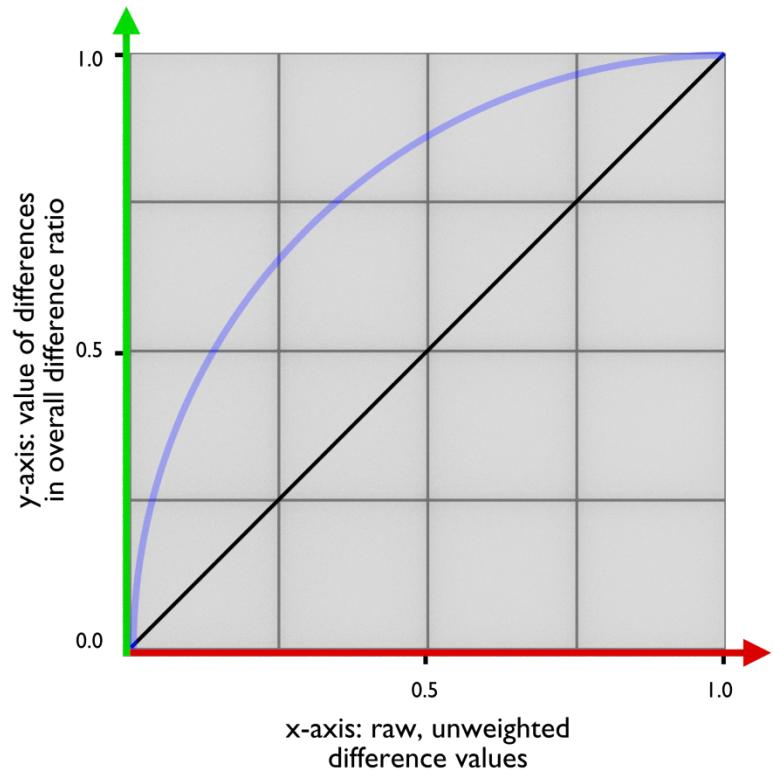


Chang Ha Lee, Amitabh Varshney, and David W Jacobs. Mesh saliency. In *ACM transactions on graphics (TOG)*, volume 24, pages 659–666. ACM, 2005.

- Implement **Selection Application**
 - Loading and indexing of 3D data
 - Selection functionality
- Conduct **User study** to collect data
 - Users select **interesting** regions via selection application
 - Selections logged and used as “**user saliency**” values
- **Difference evaluation**
 - Use *mesh saliency* to compute **mesh saliency** values
 - Gather data from user study
 - Compare mesh saliency and user saliency values
 - Conceptualize **measure of difference**

¹ <http://assimp.sourceforge.net/>

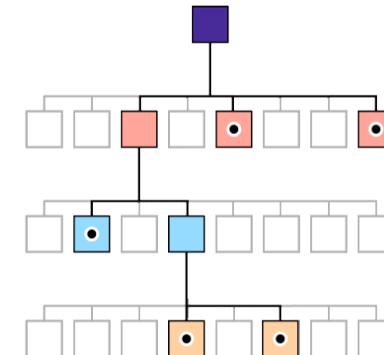
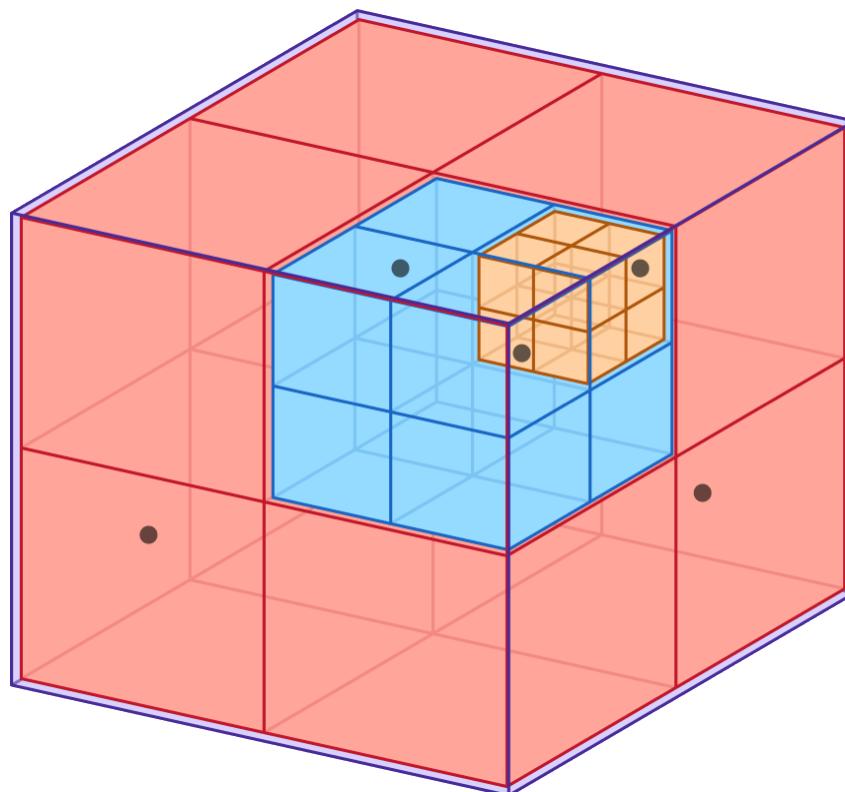
- $S_M(v_i)$ = calculated mesh saliency value for vertex v_i
- $S_U(v_i)$ = average user saliency value for vertex v_i
- $|S_M(v_i) - S_U(v_i)|$ = (unweighted) difference for vertex v_i
- Sum over differences = **measure of difference**
- All values normalized to range [0.0, 1.0], in relation to maximum values
- Some differences more interesting
 - Mesh saliency values for surrounding vertices of v_j high ->
 - $|S_M(v_j) - S_U(v_j)|$ gets weighted
- Goal: First, quick and easy measure of differences



- Spatial **index structure** as basis
 - Octree, implemented in C++
 - Load and render objects
 - Render selection target
- Selection functionality
 - **Select & deselect** vertices
 - Visual feedback on **where operations take place**
 - Visual feedback on **what is** currently **selected**
 - **Log** user selections & provide export function
- **Adjust** application for usage at V2C
 - **Distributed rendering** via graphics cluster
 - Ensure **synchronized** rendering

<http://assimp.sourceforge.net/>

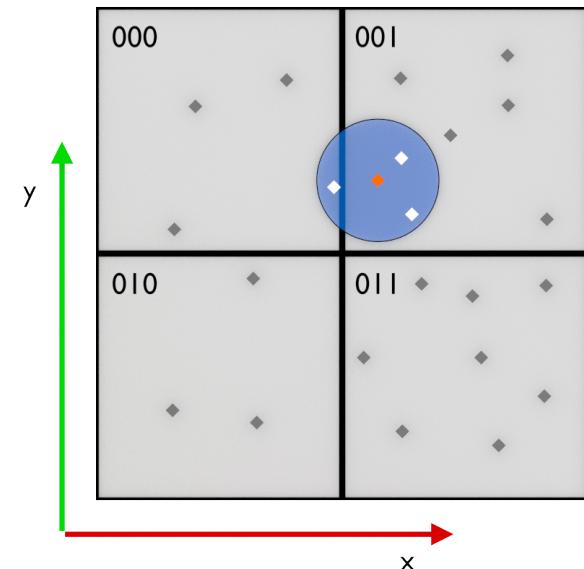
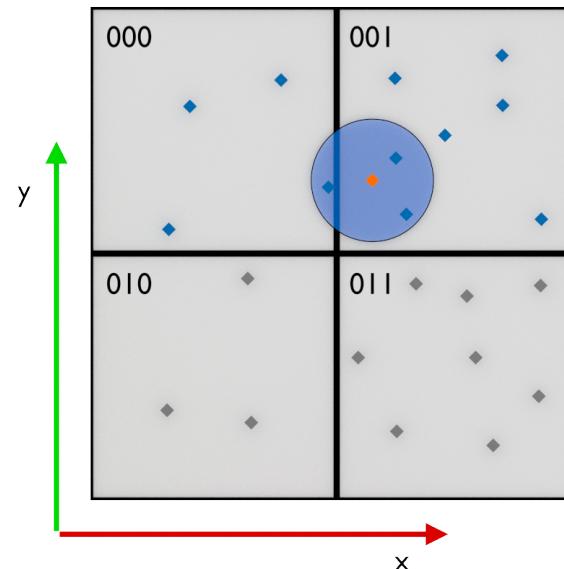
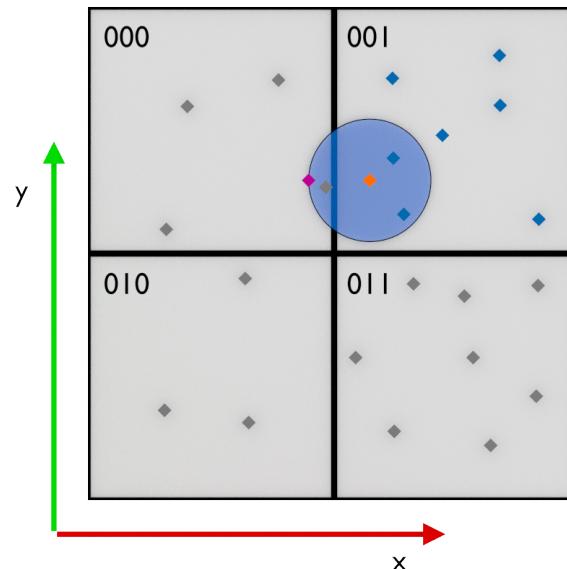
- **Load** objects via ASSIMP¹
- Spatial **indexing** of loaded 3D data
- Provide **real-time interaction**



<https://developer.apple.com/documentation/gameplaykit/gkoctree> - visited 08-12-17

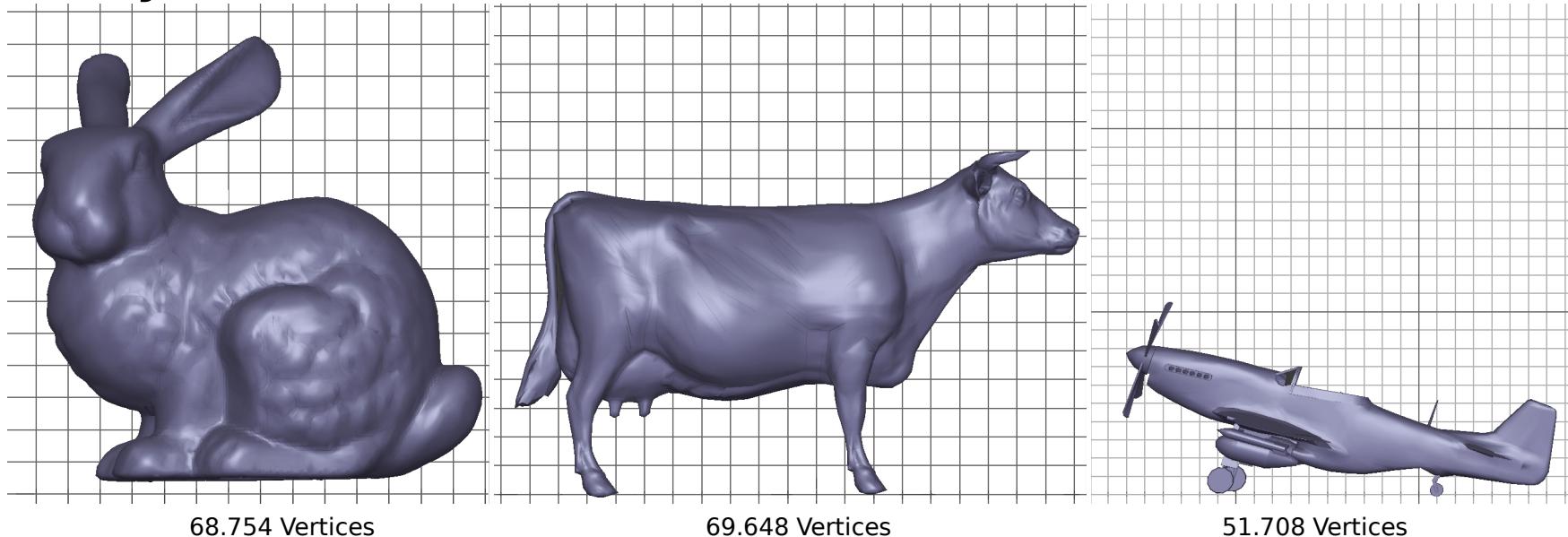
Vertex selection

- Add vertices to selection & remove them
- Next-neighbor queries and possible follow-up queries
 - Determine **target subtree** of original query
 - Determine **set of vertices to check**
 - Determine **crossed borders** of target subtree
 - Update set of vertices to check accordingly

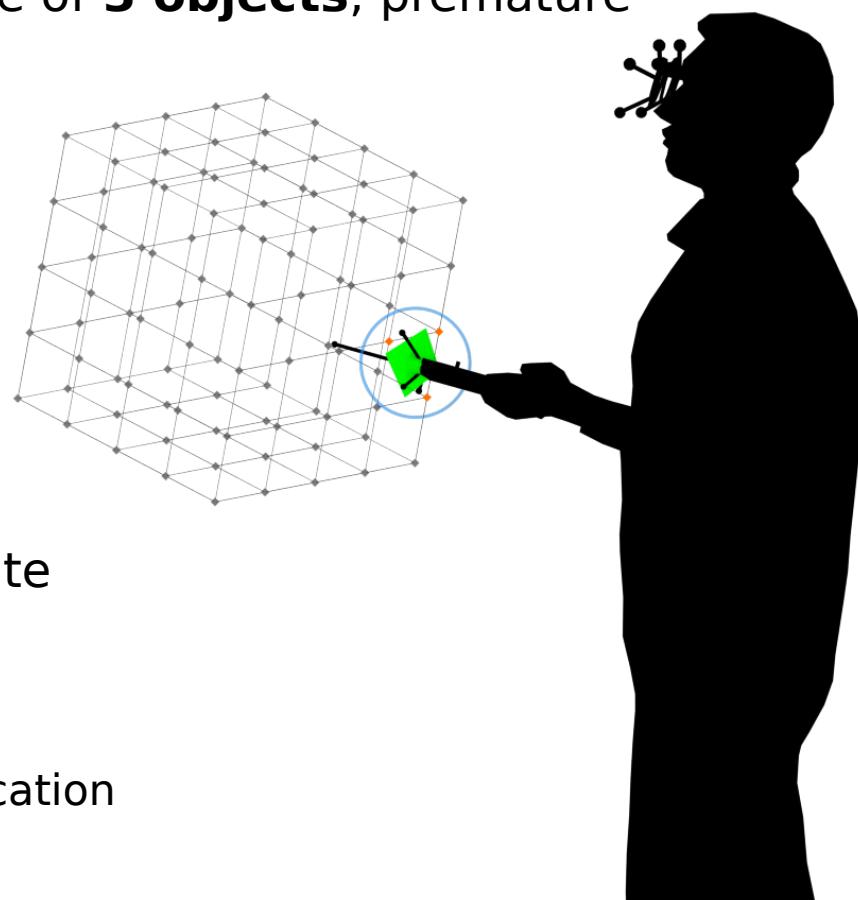


- Users select **parts of objects** considered “**interesting**”
- Selection-radius around input device: **95%** of smallest possible subtree
- “interesting” as interpreted **subjectively**. Suggested hints:
 - “*Parts which are **geometrically interesting** or important*”
 - “*Parts which intuitively attract **attention***”
 - “*Parts and regions which are **characteristic** for the object*”

Objects

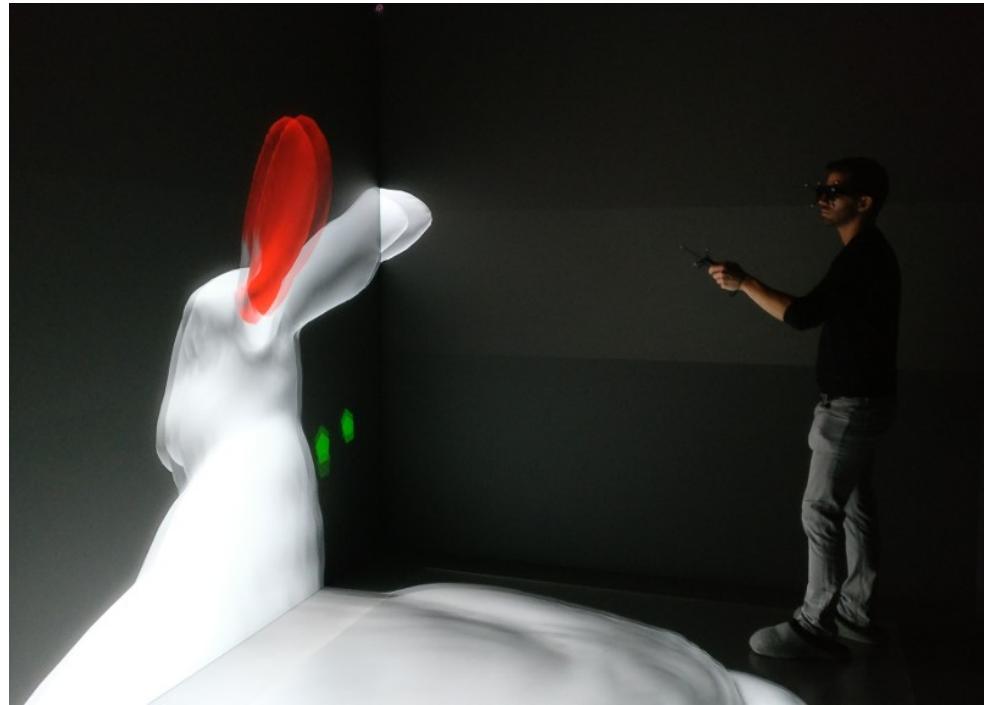


- **Take time** to **get familiar** with the application
- **5 Minutes** maximum for each one of **3 objects**, premature termination by user possible
- Consider **symmetry** in selection
- Select as **precise** as desired
- No “correct” or “incorrect” selection
- Clue at beginning of the last minute
- Questionnaire
 - Demographic information
 - Interests and feedback on the application





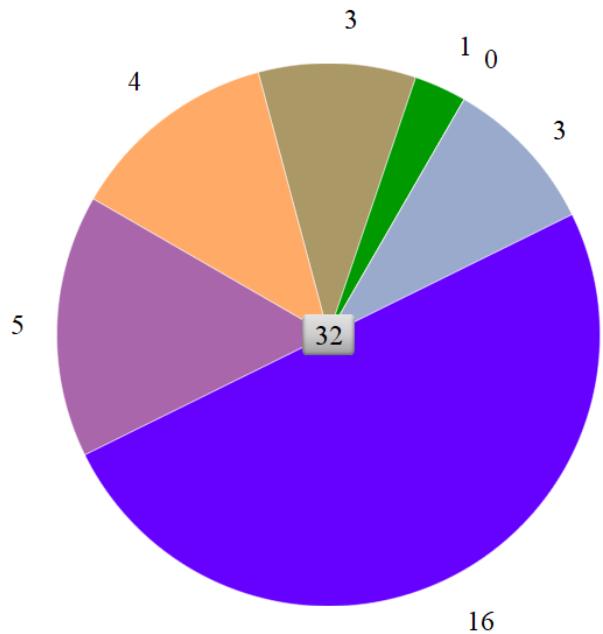
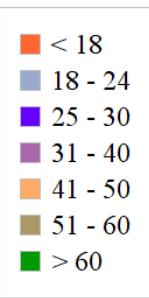
- 3 **User-selections** per user
 - Vertex-ID & Timestamp logged
 - Saved to logfile
- Get “**User saliency**” values
- Basis for evaluating differences between computed mesh saliency maps and average user selections



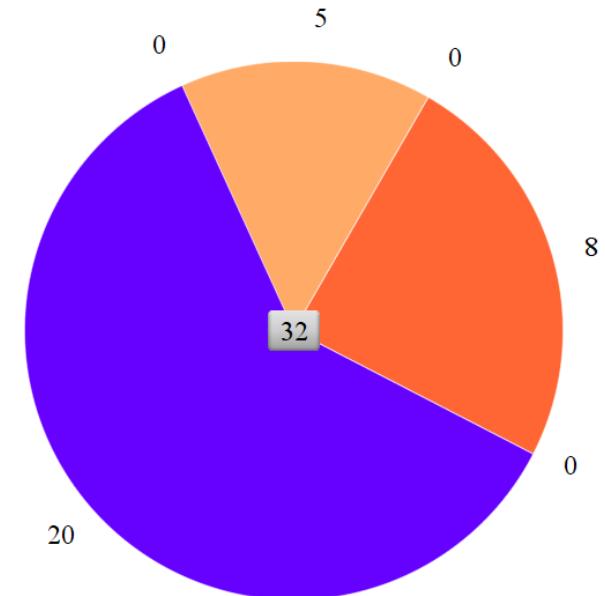
Results

Demography

- **32 participants total**, 50/50 male/ female
- **14 users with prior experience** with VR; **18 users with none**
- No strong impairments of vision
- 29 right-, 3 left-handed



Age groups



Working situation

Results

Feedback on application



Average value scores:
Likert-Scale

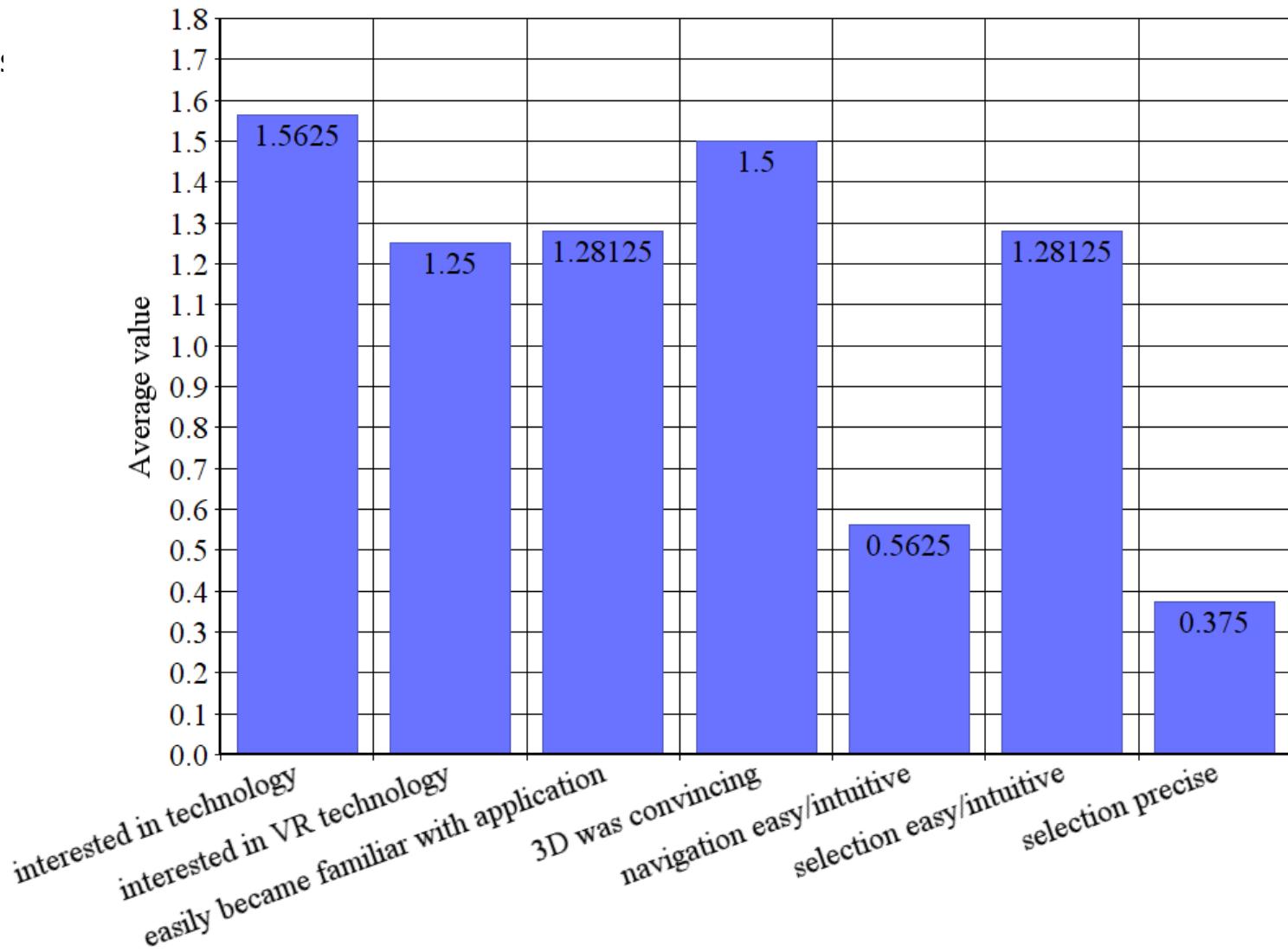
2 = "strongly agree"

1 = "agree"

0 = "no statement"

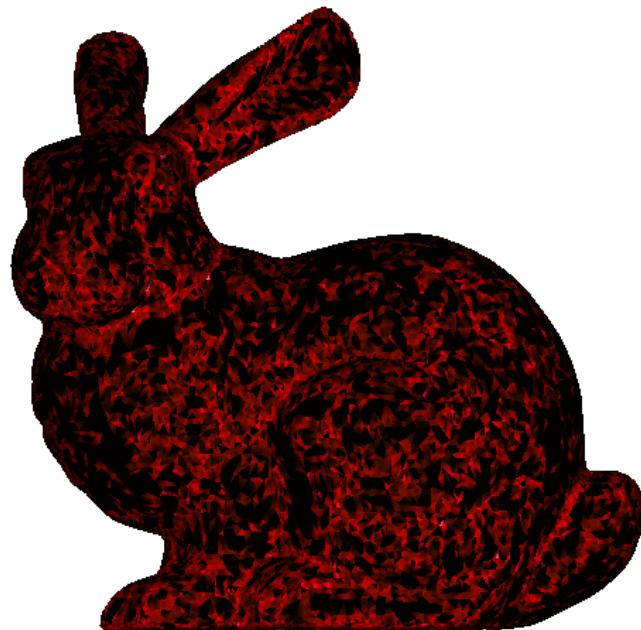
-1 = "do not agree"

-2 = "strongly disagree"



Results

Model “bunny”



Mesh Saliency



Average User Saliency

Scale



0.0

0.5

1.0

Results

Model “bunny”



Difference overall (unweighted):

0.23435

Vertices total:

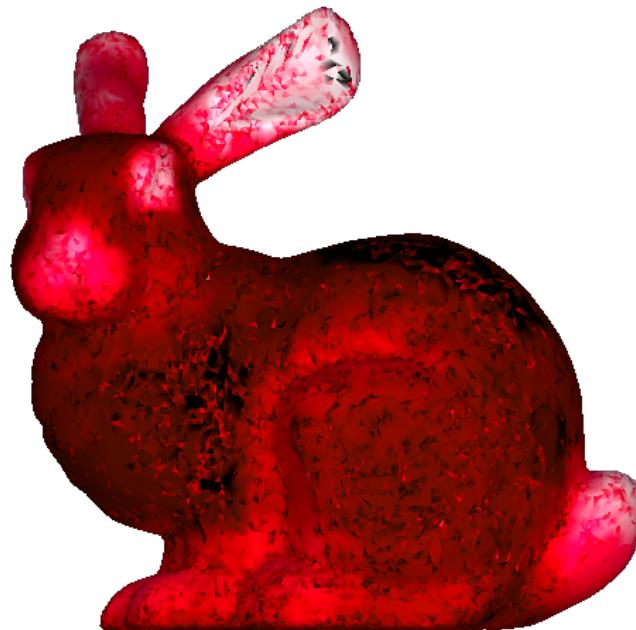
68.754

Difference overall (weighted):

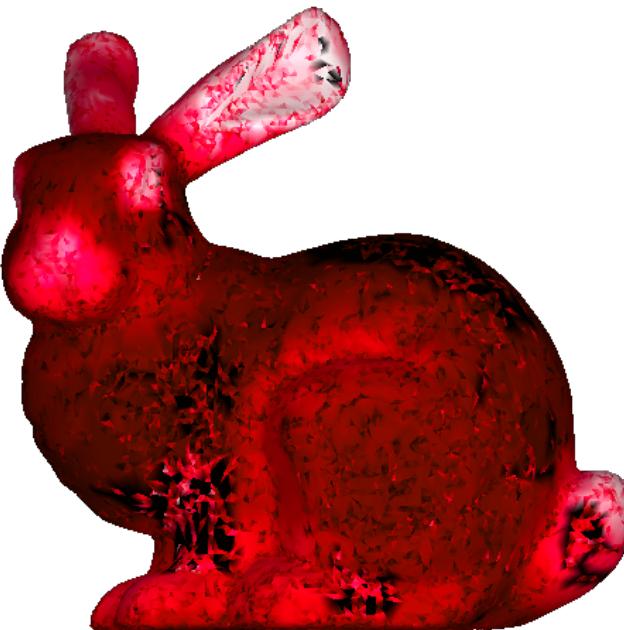
0.23455

Vertices weighted:

2667



Differences unweighted



Differences weighted

Scale



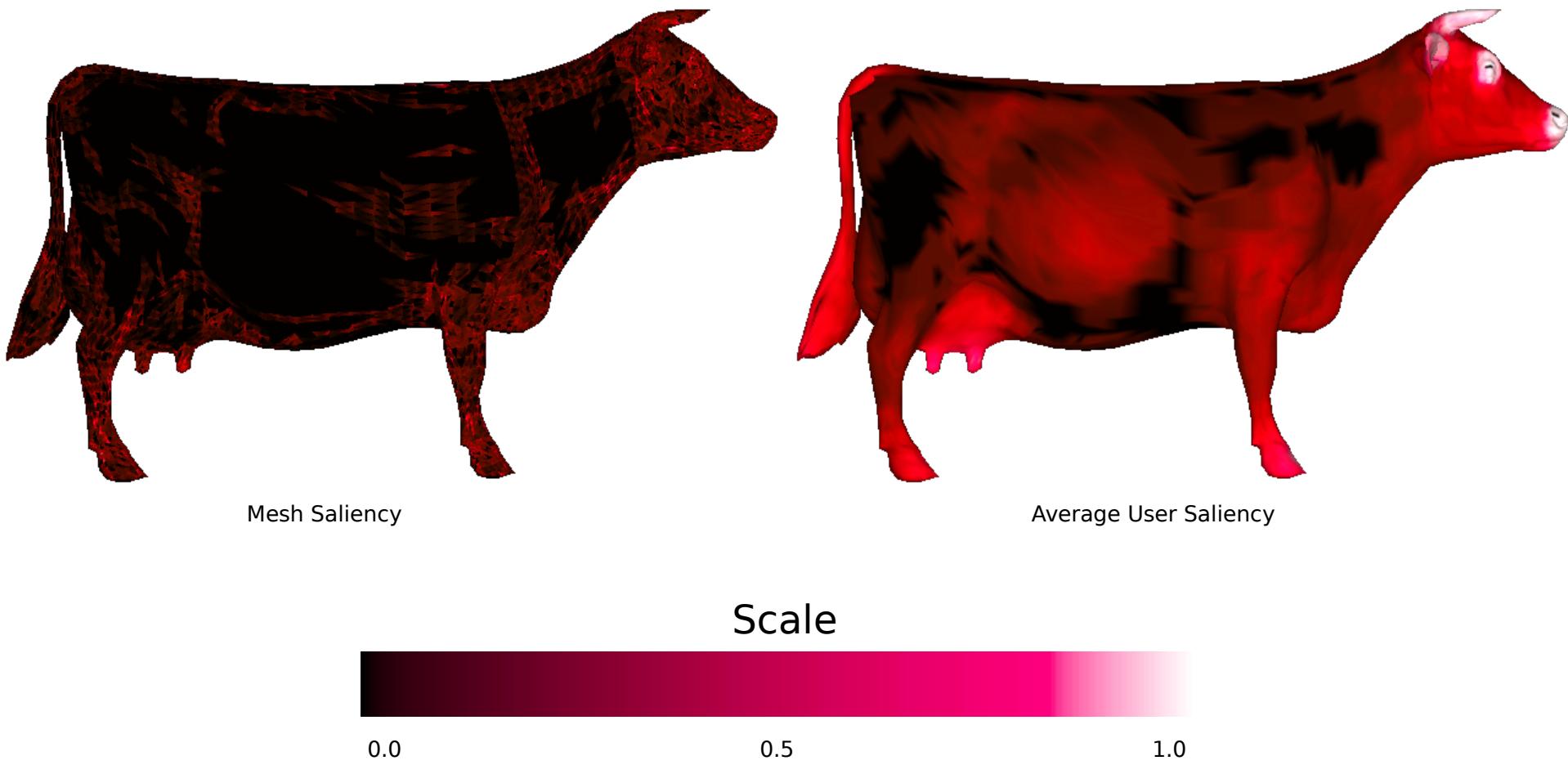
0.0

0.5

1.0

Results

Model “cow”



Results

Model “cow”

Difference overall (unweighted): **0.27431**
Difference overall (weighted): **0.26904**

Vertices total: **69.648**
Vertices weighted: **3200**



Differences unweighted



Differences weighted

Scale



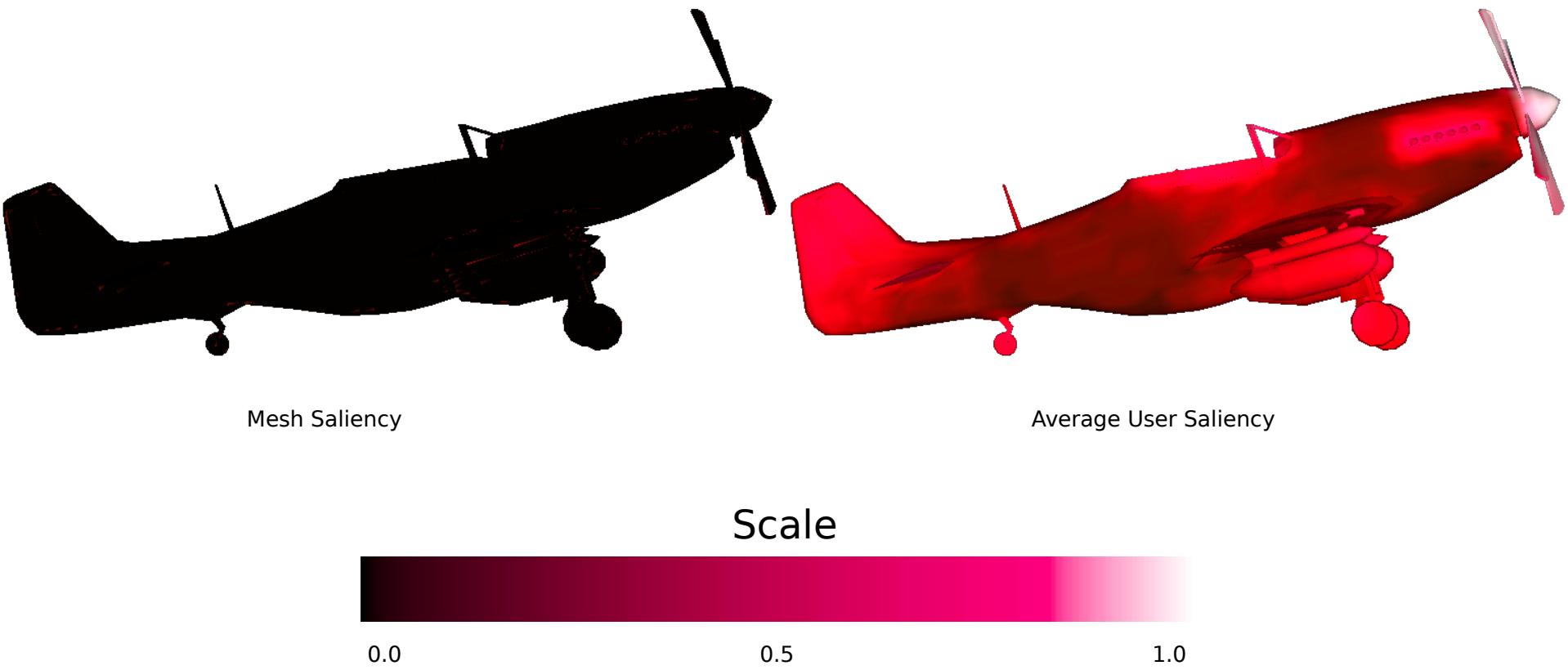
0.0

0.5

1.0

Results

Model “P51”



Difference overall (unweighted): **0.38002** Vertices total: **51.708**
Difference overall (weighted): **0.38002** Vertices weighted: **0**



Differences unweighted

Differences weighted

Scale



0.0

0.5

1.0

Measure of difference

- Overall differences between ~0.23 and ~0.38
- Low difference, as expected **but**
 - **Many very low values** “water” overall result
 - Very basic measure of difference

Observations

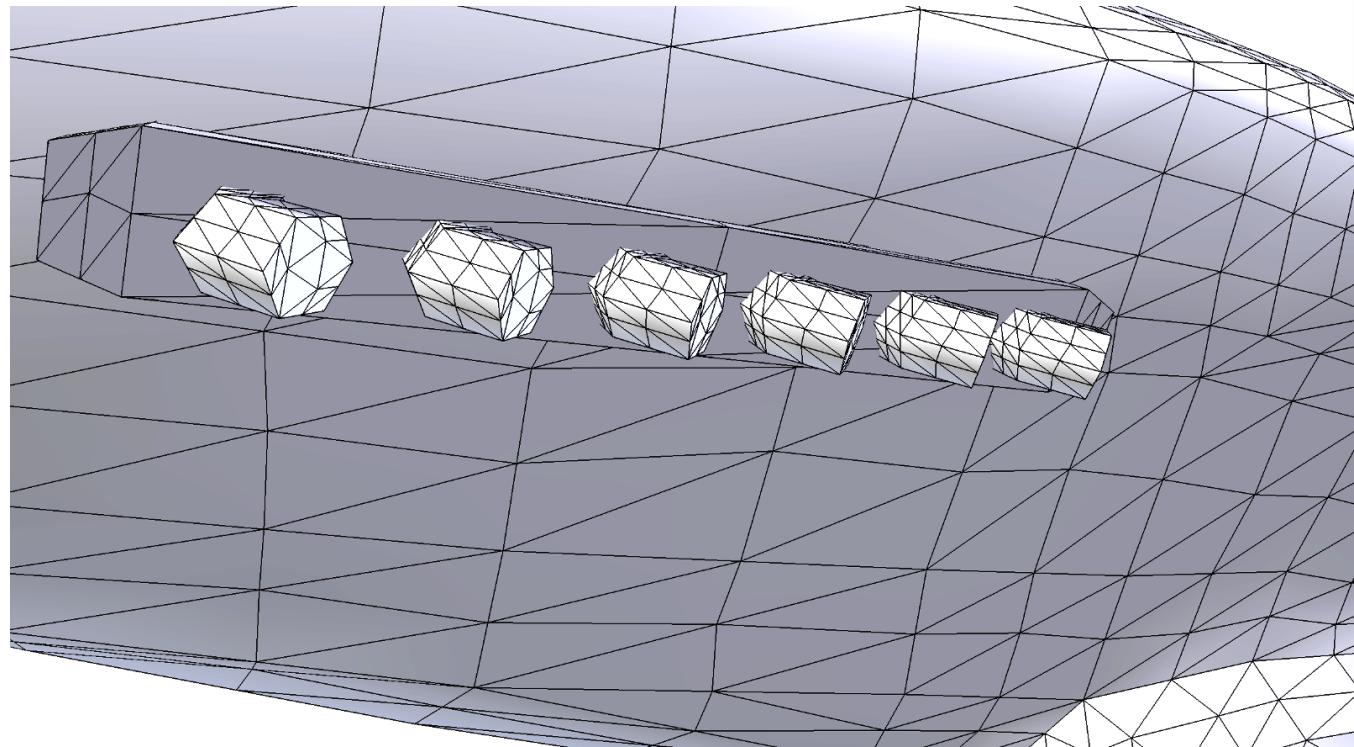
- Seemingly **zero-values** in the middle of highly “*popular*” regions
- Possible faults in implementation
- Stronger **variation** in user saliency values
- More clearly **distinct patches** in user saliency values
- Clearer values for **both** mesh and user saliency values for non-mechanical objects





Mesh saliency with mechanical objects

- **Normalizing** saliency values to the highest one
 - Very **few**, extremely **high** values
 - **Almost all** values are very **low**
- Weakness of difference measure



- Further develop **measure of difference**
 - No normalizing of values
 - Multiple figures expressing differences
- Other values for variables
 - Selection radius
 - Object size
 - Time
- More extensive **user study**
 - More users
 - Other kinds of immersive VR
- Use **additional data** collected
 - Determine influences of VR (feedback)
 - Use logged **time**

References

- Chang Ha Lee, Amitabh Varshney, and David W Jacobs. Mesh saliency. In *ACM transactions on graphics (TOG)*, volume 24, pages 659–666. ACM, 2005.
- De Ladurantaye, Vincent, Jacques Vanden-Abeele, and Jean Rouat. Models of information processing in the visual cortex. INTECH Open Access Publisher, 2012.
- Pausch, Randy, Dennis Proffitt, and George Williams. "Quantifying immersion in virtual reality." Proceedings of the 24th annual conference on Computer graphics and interactive techniques. ACM Press/Addison-Wesley Publishing Co., 1997.
- Rodrigo Barni Munaretti. Perceptual guidance in mesh processing and rendering using mesh saliency. 2007.
- Paul Atchley and Arthur F Kramer. Attentional control within 3-d space.
- Gabriel Taubin. Estimating the tensor of curvature of a surface from a polyhedral approximation. In *Computer Vision, 1995. Proceedings., Fifth International Conference on*, pages 902–907. IEEE, 1995.
- <http://assimp.sourceforge.net/>
- <https://developer.apple.com/documentation/gameplaykit/gkoctree> - visited 08-12-17
- http://http.developer.nvidia.com/GPUGems2/gpugems2_chapter37.html - visited 07-06-17