GPU-accelerated Attention Map Generation for Dynamic 3D Scenes

Thies Pfeiffer*
CITEC, Faculty of Technology, Bielefeld University, Germany

Cem Memili†
Faculty of Technology, Bielefeld University, Germany

Figure 1: While a user is inspecting a 3D model in a CAVE, our algorithm generates attention-map textures on-the-fly using the GPU. These textures can be either visualized in real-time or exported to be used in high-end off-line renderings of the inspected 3D models. The picture shows the model "BMW 3 Series Coupe" by mikepan (Creative Commons Attribution, Share Alike 3.0 at http://www.blendswap.com).

Abstract

Measuring visual attention has become an important tool during product development. Attention maps are important qualitative visualizations to communicate results within the team and to stakeholders. We have developed a GPU-accelerated approach which allows for real-time generation of attention maps for 3D models that can, e.g., be used for on-the-fly visualizations of visual attention distributions and for the generation of heat-map textures for offline high-quality renderings. The presented approach is unique in that it works with monocular and binocular data, respects the depth of focus, can handle moving objects and is ready to be used for selective rendering.

Index Terms: I.3.3 [Computer Graphics]: Picture/Image Generation—Display algorithms;I.3.7 [Computer Graphics]: Three-Dimensional Graphics and Realism—Color, shading, shadowing, and texture;I.3.7 [Computer Graphics]: Three-Dimensional Graphics and Realism—Virtual reality

1 Introduction

The analysis of visual attention has important application areas in the field of virtual and augmented reality. It can be used to evaluate the visual quality of renderings or to evaluate human behavior in virtual reality simulations. Real-time information about visual attention can be used for human-computer interaction, e.g., for object selection or selective rendering.

2 Related Work

Stellmach et al. [4] present several approaches for visualizing prerecorded attention data. Their *Projected Attentional Maps* are 2D heat-maps overlaid over the 3D scene that ignore the 3D structure of the scene. Their *Object-based Attentional Maps* are too coarse grained as they reduce the level of attention to one value per object. More details are provided by their *Surface-based Attentional Maps*, as they assign colors based on the level of attention per triangle. However, this approach heavily depends on the granularity of the mesh, has high memory demands and severely affects the visualization performance, as the authors themselves attest. In addition to that, all these approaches only consider the first object that has been hit by the line of sight, not the objects in a cone-shaped volume of attention, which would be more realistic. They also do not consider the depth of a fixation or visibility issues.

Pfeiffer [3] presents 3D Attention Volumes to overcome some of these issues, but the created visualizations are still not realistic as they do not consider visibility issues. The volume-based approach is also very resource-intensive in terms of memory and processing time. Duchowski et al. [1] present a GPU-based algorithm to create attention maps in real-time, but for 2D video footage only. Maurus et al. [2] contribute to the state-of-the-art for attention mapping on 3D scenes by using Shadow Mapping to solve the visibility problem. Except that, their approach works similar to the Projected Attentional Maps. A drawback is that a Shadow Map has to be stored in a texture buffer for every distinct viewing direction of the viewers, which in a mobile setting could be as many as sampled fixations. As a consequence, the authors suggest to use textures of a low resolution, but this results in degradation of the visual quality.

3 Our Approach

We extend the state-of-the-art in several ways. First, we consider real 3D distributions around the 3D point of regard including depth of fixation and respecting visibility. Second, the aggregated attention data is stored as textures per object, which allows us to correctly deal with moving objects, selectively aggregate visual attention on a per-object level and to persist the created textures for creating high-quality renderings.

The central idea of the approach is to represent the attention distributions as textures on a per-object level and to generate these textures directly on the GPU. We call these textures Attention Textures. The algorithm takes a set of viewing events (position + gaze-

*e-mail: Thies.Pfeiffer@Uni-Bielefeld.de
†e-mail: cmemili@techfak.uni-bielefeld.de

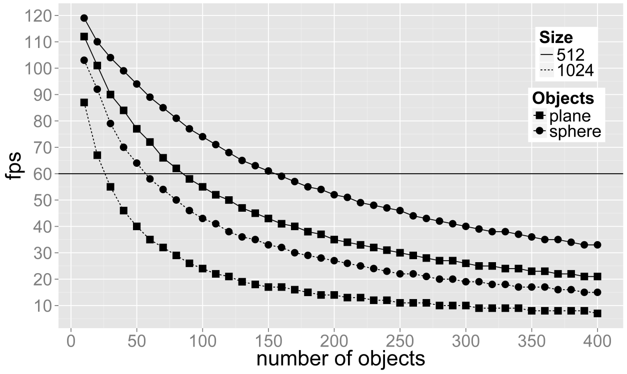

Figure 2: The graph shows the relation between frames per seconds and number of objects of different type within view for which Attention Textures of different resolutions are created.

direction or 3D point of regard) as input. The Attention Textures are then generated according to the algorithm outlined in the following, with the advances compared to previous approaches *emphasized*:

1. Loop over viewing events in the consideration set
   (a) Determine Shadow Map for current viewing event *up to the extension of the fixation in depth* to consider partial occlusions
   (b) Determine intersections of target objects with the *3D attention distribution around the point of regard*; use *Geometry Shader for backface culling*; *update Attention Texture of each object* accordingly; store *per-texel maximum in a single Max-Attention Texture*
2. *Determine global maximum value over all Attention Textures using the Max-Attention Texture*
3. Render scene from reviewer's perspective; *normalize and apply desired color ramp on-the-fly in a shader*

By deferring the normalization of the Attention Textures to the shader used for visualization, no update to normalize all texels is required unless they are to be persisted. By using the Max-Attention Texture, no additional iteration over all Attention Textures to identify the absolute maximum required for normalization is needed. By storing the maximum per texel and not per Attention Texture, the approach scales in O(1) regarding the required additional memory independently of the number of objects. The step 1(a) also *supports binocular eye-tracking data using two separate Shadow Maps* to handle cases correctly in which an object can only be seen with one eye.

More details can be taken from the algorithm and an example program, which will be made available on http://gpu-heatmap.multimodal-interaction.org/ on the date of publication.

4 Performance Analysis

We tested our algorithm on a dual Intel Xeon Processor E5-2643 (3.3 GHz, 8 GT/s) system with a 4 GB NVIDIA Quadro K5000 (173 GB/s, 256-bit); the newer K6000 has a 60% increase in bandwidth, which is the most delimiting factor except for memory size. As a worst-case scenario, we used a scene with increasing numbers of planes within focus of the viewer, so that all texels of each Attention Texture will be touched during the update. A scene with an increasing number of spheres defines our standard case, as our approach culls the texels that are facing away from the viewer and thus only half of the texels need to be updated.

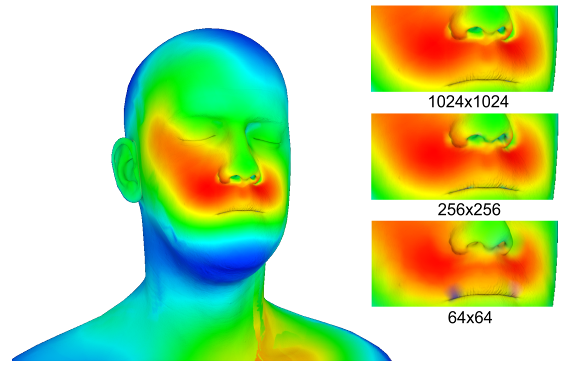

Figure 3: The visual quality depends on the resolution of the sampled Attention Texture (see magnified nose/mouth area to the right). For many objects a resolution of 256x256 might already be sufficient.

On the test system 80 planes or 155 spheres with a resolution of 512x512 can be updated at 60 Hz (Figure 2) or, alternatively, 25 planes or 55 spheres with a resolution of 1024x1024. Updates are only required when a new sample from the eye-tracking system arrives. We use the SMI Eyetracking Glasses with a sample rate of 60 Hz. The rate of required updates can be lowered further if they are only executed per fixation. As relevant fixation lengths start at 90 ms, no more than 10 fixations per second per user can be expected. In this case, the test system would be able to handle 280 planes of resolution 1024x1024 per update.

In common scenarios there are rarely 25 objects of interest within the focus of attention (about 2° of visual angle) at the same time. It may happen for very small objects, but then much smaller texture sizes can be used to sample the visual attention. Also, as attention spreads continuously, interpolation of the Attention Textures can help maintaining visual quality while being able to reduce texture resolutions (Figure 3).

5 Conclusion

The presented approach allows us to sample and visualize visual attention on 3D objects at interactive frame-rates. The approach respects visibility issues, supports binocular perspectives, depth of focus and moving objects. The results can also be made persistent and used for high-quality renderings (see Figure 1).

Besides applications in the areas of ergonomics, human factors and usability research, the approach can be used in human-computer interaction, e.g. for speech, gaze and gesture integration and for selective rendering.

References

[1] A. T. Duchowski, M. M. Price, M. Meyer, and P. Orero. Aggregate gaze visualization with real-time heatmaps. In *Proceedings of the Symposium on Eye Tracking Research and Applications*, ETRA '12, pages 13–20, New York, NY, USA, 2012. ACM.

[2] M. Maurus, J. H. Hammer, and J. Beyerer. Realistic heatmap visualization for interactive analysis of 3d gaze data. In *Proceedings of the Symposium on Eye Tracking Research and Applications*, ETRA '14, pages 295–298, New York, NY, USA, 2014. ACM.

[3] T. Pfeiffer. Measuring and visualizing attention in space with 3d attention volumes. In *Proceedings of the Symposium on Eye Tracking Research and Applications*, ETRA '12, pages 29–36, New York, NY, USA, 2012. ACM.

[4] S. Stellmach, L. Nacke, and R. Dachselt. 3d attentional maps: Aggregated gaze visualizations in three-dimensional virtual environments. In *Proceedings of the International Conference on Advanced Visual Interfaces*, AVI '10, pages 345–348, New York, NY, USA, 2010. ACM.