Realistic Heatmap Visualization for Interactive Analysis of 3D Gaze Data

Michael Maurus*
Fraunhofer Institute of Optronics, System Technologies and Image Exploitation IOSB

Jan Hendrik Hammer†
Karlsruhe Institute of Technology
Department of Informatics
Institute for Anthropomatics
Vision and Fusion Laboratory

Jürgen Beyerer‡
Fraunhofer Institute of Optronics, System Technologies and Image Exploitation IOSB

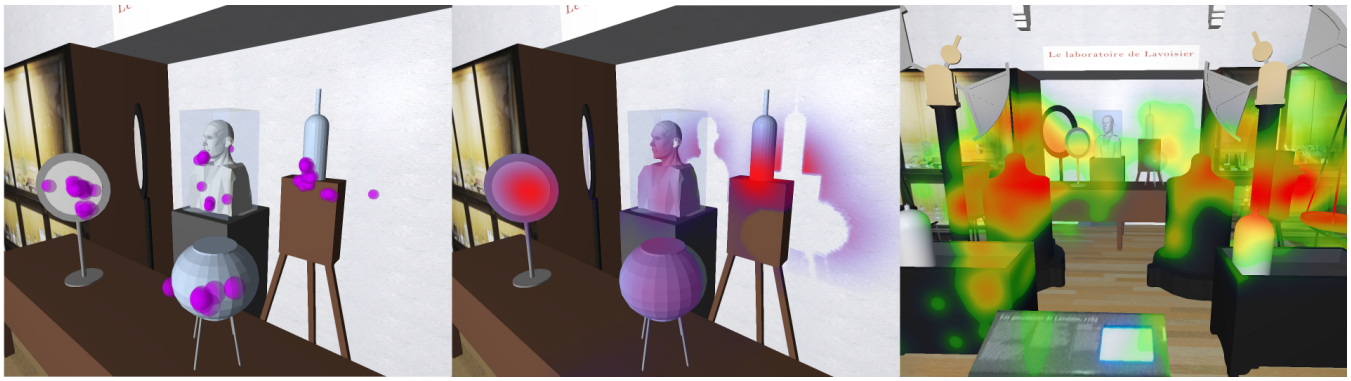

Figure 1: *About hundred gaze points (violet) on objects in the scene (left), the corresponding heatmap considering occlusions (middle) and a projection of the gaze for a scanpath containing over 13,000 gaze points into the 3D scene (right). The gaze data was recorded in the Lavoisier Laboratory, Musée des Arts et Métiers (Paris, France).*

Abstract

In this paper, a novel approach for real-time heatmap generation and visualization of 3D gaze data is presented. By projecting the gaze into the scene and considering occlusions from the observer's view, to our knowledge, for the first time a correct visualization of the actual scene perception in 3D environments is provided. Based on a graphics-centric approach utilizing the graphics pipeline, shaders and several optimization techniques, heatmap rendering is fast enough for an interactive online and offline gaze analysis of thousands of gaze samples.

CR Categories: 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;

Keywords: heatmaps, projection, shadow mapping, occlusion, gaze analysis, 3D environments

1 Introduction

Gaze analysis using static eye trackers is typically applied for 2D scenarios like images and videos on screen, e.g. in advertisement departments for analyzing the effectiveness of adverts in print media or television. But with modern, fast and accurate mobile eye tracking systems, even more complicated, three-dimensional scenarios like in museums [Hammer et al. 2013], shops [Harwood et al. 2013] or even in sports [Campbell et al. 2013] can be analyzed.

For the analysis of gaze data in mobile scenarios, there exist two possibilities: One is based on the correct projection of the point of regard into the image of the scene camera. The gaze point is then associated with objects detected in the frame based on methods for object recognition in 2D images as in [Toyama et al. 2012]. The other possibility is to use 3D information e.g. the pose of the mobile eye tracker and the viewing direction in the world coordinate system of the environment to estimate the 3D point of regard. This so called geometry-based gaze point estimation [Pfeiffer 2012b] requires a geometric representation of the scene, which can be created manually or automatically [Paletta et al. 2013].

To make an easy and fast manual analysis of gaze data possible, an efficient and informative visualization is needed. In Figure 1 on the left about 100 gaze points are visualized as violet spheres on the observed objects. But as can be seen, it is difficult to figure out which areas have been attended most due to gaze points occluding each other. For this purpose heatmaps are much more appropriate. Therefore, they are a typical visualization technique for gaze data by mapping scalar values to a color determined by a specific color mapping function. In Figure 1 on the middle a color mapping function using a red-blue color gradient can be seen. The right image of Figure 1 in comparison shows the widely used rainbow color mapping. In both cases, red associates regions of more visual attention. The person analyzing the data quickly gets the gist of which areas have been looked at most and which areas did not attract visual attention at all. For 3D gaze points the generation of heatmaps is much more complicated than for scanpaths only consisting of 2D gaze samples: First, the visual attention needs to be projected onto a three-dimensional scene and not just onto a 2D plane. Second, objects occlude parts of the scene viewed from the tracked person

*e-mail: michael.maurus@iosb.fraunhofer.de
†e-mail: jan.hammer@kit.edu
‡e-mail: juergen.beyerer@iosb.fraunhofer.de

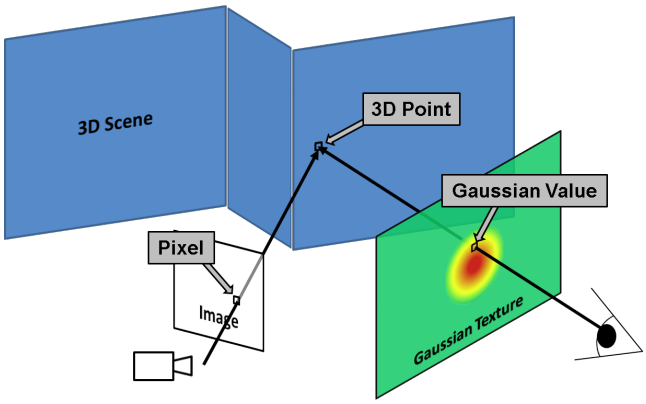

Figure 2: *Projection of a texture into the scene: For each pixel, the corresponding 3D point is projected onto the image plane of the eye to get the Gaussian value for this 3D point.*

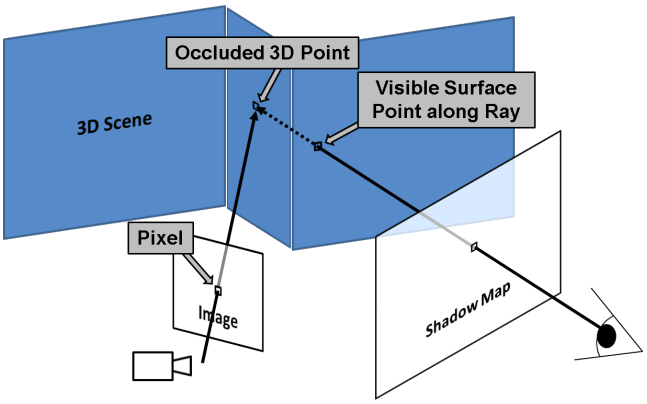

Figure 3: *Occlusion test for a point of the scene: The 3D point corresponding to a pixel is projected onto the image plane of the eye to determine the depth value in the shadow map. This depth value is compared to the distance from the nodal point of the eye to the actual 3D point. If the 3D point is farther away from the nodal point than the nearest surface, it is occluded.*

which must not be visualized as being viewed. To also facilitate an interactive steering of the online and offline gaze analysis, an efficient and fast implementation of the heatmap generation is needed, allowing the user to quickly change the visualization parameters like the viewed time window.

Section 2 provides a short overview of the latest approaches for visualizing gaze data in 2D and 3D environments using heatmaps. In Section 3 our novel approach for realistic 3D heatmap generation in real-time is presented and some implementation details for boosting the performance of the algorithm are given. Finally, a conclusion is given in Section 4.

2 Related Work

Stellmach et al. [2010] presented three approaches for visualizing gaze data in 3D environments: a projected, an object- and a triangle-based representation. The first one can be used for a spatial overview of the scene using e.g. a bird's eye view like in 3D modeling tools. The object-based representation gives an overview of viewed objects in the scene, typically computing the cumulative fixation time for each object and respectively coloring these. This can also be compared to an area-of-interest (AOI) analysis where the AOIs equal the objects of the scene. The triangle-based heatmap representation on the other side visualizes the gaze information on the surfaces of the scene. For the triangle-based heatmap generation, the vertices of the scene mesh are weighted according to a 3D Gaussian function centered around the gaze point. Therefore, the resulting heatmap depends on the resolution of the triangulation. For example, if a cube is viewed from one side where the gaze point lies in the middle of one face, as illustrated in Figure 3 in [Stellmach et al. 2010], and the cube is rendered using only two triangles per face, the four points of the face sample the 3D Gaussian function far away from the mean resulting in low weights. Using bi-linear interpolation for colorizing the triangles and a rainbow color mapping function, the whole side of the cube would be colored green because of the low weights at the vertices. This sub-sampling of the visual acuity results in an incorrect visualization of the scene perception. Another problem concerning the 3D Gaussian function is, that it is not possible to find out from which direction the observer viewed the object, because the weighting is independent of the incident angle of the line of sight. Furthermore, a large support of the 3D Gaussian weighting and a relatively small geometry lead to colorized triangles which are actually occluded by the geometry itself. When analyzing, this makes the user think, that the back of the geometry was looked at, which in fact was not the case. These two problems can also be observed at the duck mesh on the left of Figure 3 in [Stellmach et al. 2010].

Pfeiffer et al. [2012a] described a direct volume rendering approach where a fixation is visualized with the help of a continual, three-dimensional Gaussian function centered around the 3D position of the gaze samples. The distribution represents the visual acuity around the visual axis. Additionally, the distribution is distorted perpendicular to the line of sight so that it gets broader from the observer's point of view depending on the distance. The amplitude is scaled due to the duration of the fixations. Using a direct volume rendering approach, the attention volumes can be integrated over time or different persons and visualized according to a color mapping function. The problem here is, that the visualization is not realistic and therefore not correct, because no projection into the scene and no occlusions are considered. Actually, this approach is similar to rendering an oriented ellipsoid at the computed gaze point instead of a sphere.

Duchowski et al. [2012] presented another approach utilizing the GPU for fast heatmap generation of a 2D scanpath. For each gaze point or fixation, a full-screen rectangle is drawn over the scene. On each rectangle, a 2D Gaussian distribution representing the gaze probability is rendered being centered around the gaze point or fixation in image space. Next, all the Gaussian distributions are accumulated using the blending functionality of the graphics pipeline to get a 2D scalar field. This scalar field needs to be normalized to map the scalar values to color according to a given color mapping function. For this, they implemented a parallel reduction algorithm using shaders for finding the maximum value in the scalar field. Also utilizing shaders for colorization, it can easily be switched between different color mappings. They further suggested using their GPU shader approach to render heatmaps on 3D surfaces on a per-fragment basis, but no details were given.

In summary, current state-of-the-art visualization techniques for 3D heatmaps do not create a correct visualization of the scene perception due to utilizing a 3D Gaussian distribution around the gaze point or fixation describing the viewed area in the scene. This yields a strong dependency between the size of the Gaussian distribution and the size of the geometry which can lead to false colorizations at the back of the geometry. The contribution of this paper is to provide a more realistic visualization of the scene perception by

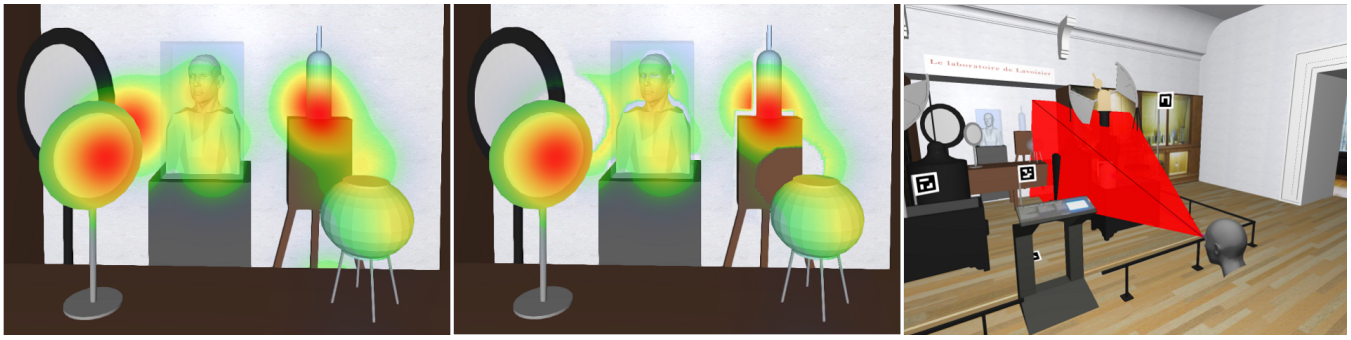

Figure 4: *Projection of about 100 gaze samples into the scene without (left) and with (middle) consideration of occlusions. The image space involved in the projection computations can be limited to a sight cone approximation (red pyramid on the right).*

projecting the gaze into the scene instead of using 3D Gaussian distributions at specific locations.

3 Realistic Heatmap Generation

3.1 3D Gaze Data as Input

Using our approach for a fast heatmap generation extending [Duchowski et al. 2012], 3D gaze input data is required which consists of a line of sight in the 3D environment for each gaze sample. This line is defined by the nodal point of the corresponding eye [Guestrin and Eizenman 2006] and the viewing direction. Acquisition of this data highly depends on the used eye tracking system. Its eye tracking must deliver the line of sight for the viewing direction and it further needs a computation of the extrinsic parameters of the scene camera of the eye tracker to be able to estimate the nodal point of the eye in 3D coordinates. Additionally, a geometric representation of the scene is needed for rendering and can also be utilized for computing the exact 3D position of the most likely gaze point by intersecting the line of sight and the scene.

3.2 Gaze Projection into Scene

Typically, a 2D Gaussian function is used for describing the visual acuity on a plane. The standard deviation of the Gaussian distribution can be chosen respectively to the field of view of the fovea which is typically spanning less than 2° [Holmqvist et al. 2011]. Since this does not change, the 2D Gaussian function describing the gaze probability can be saved in a static texture. For a specific position of the observer and arbitrary 3D points in the scene, their coordinates only need to be projected onto the image plane of the eye to get the gaze acuity from the pre-computed texture. This process is illustrated in Figure 2 and technical details are described below.

For each gaze point, a full-screen rectangle is drawn, so that for each pixel of the rectangle a fragment program is executed. In the fragment program, the 3D point of the surface corresponding to the looked at pixel will be transformed into normalized device coordinates of the tracked person's eye. Therefore, the 3D points corresponding to all pixels will be written to a texture when rendering the scene. These points are given in world coordinates and thus need to be transformed into the camera coordinate system of the eye. This is done using the position of the computed gaze point and the nodal point of the eye, which are stored in graphics memory using a vertex buffer object. The line between those two points in world space needs to be the z-axis in eye space while the origin of the coordinate system is the nodal point. To accomplish this, the tangent space along this line is computed and used as the view transformation. Next, these coordinates in eye space will be transformed to normalized device coordinates applying the projection matrix and a perspective division. Finally, the normalized device coordinates, which are in the interval of $[-1, 1]^3$, need to be transformed to texture coordinates $[0, 1]^2$ to read the visual acuity from the Gaussian texture. The following accumulation using blending, the normalization and the colorization are implemented utilizing the approach described in [Duchowski et al. 2012].

3.3 Consideration of Occlusions

Using the above described projection of gaze into the scene, surfaces, which are occluded from the tracked person's view but visible in the virtual camera, may also be colorized, because 3D points on these surfaces will also be projected on a valid texel inside the Gaussian texture. To prevent this incorrect visualization, an occlusion test is performed for each eye position. In Figure 4, a visualization without occlusion test can be seen on the left image and with occlusion test on the image in the middle. Comparing both images, it can be seen that in the left image areas are colorized which have not been looked at. Those areas are clearly visible as shadows of the projection in the central image. Thereby a correct visualization of the scene perception can be achieved.

To reuse previous computations already done for projecting the Gaussian texture into the scene and therefore minimizing computations per fragment, Shadow Mapping [Williams 1978] can be utilized for an occlusion test illustrated in Figure 3: For each gaze point, a shadow map or depth map of the scene is computed from the view of the nodal point of the eye. This is done once per frame for all new gaze samples at the beginning of the rendering computations. Before projecting a 3D point onto the Gaussian texture to read the probability value, an occlusion test is executed: Using the previously computed normalized device coordinates of a 3D point visible in the virtual camera, the depth value can be read from the shadow map. This depth value will be compared to the z-value of the normalized device coordinates of the 3D point to check for occlusion. Is the depth value from the shadow map, representing the distance from the nodal point of the eye to the nearest surface in the direction of the 3D point, less than the distance from the nodal point to the 3D point itself, the point is not visible. Hence, no further computations inside the fragment shader need to be done. In summary, the fragment shader doing the projections only needs another texture read from the depth map and a comparison to facilitate an occlusion test.

Since shadow maps will be needed for each eye position, the resolution of the shadow maps should be quite low for visualizing many gaze points because of the limited graphics card memory. On the

contrary, a low resolution corresponding to a low sampling of the scene's depth leads to typical artifacts of the shadow mapping algorithm like aliasing. To address this, texture compression methods and shadow mapping extensions for soft shadow edges can be used to improve quality and memory demands.

3.4 Limiting the regarded area in image space

Until now, each pixel will be dealt with, although only a few of the corresponding surfaces lie inside the sight cone. To speed up the accumulation, the image space, for which the projection computations and occlusion tests are executed, should be limited as much as possible. This is accomplished by rendering a pyramid approximating the sight cone instead of drawing a full-screen rectangle. The pyramid will be drawn along the line of sight where the top of it lies at the nodal point of the eye and the faces are extruded to infinity. Usually, this leads to a far smaller area in image space which needs to be processed by the shader programs for heatmap computation for a single gaze point. In Figure 4 on the right, this pyramid for a specific gaze point is illustrated. The projection computations only need to be done for the red pixels.

3.5 Reusing previous accumulations

If the position and orientation of the virtual camera do not change on subsequent frames, the accumulations of the previous frames can be reused. In this case, only the contribution of the new gaze points needs to be added to the current accumulation texture. If a limited time slot is analyzed, additionally the contribution of gaze points, which lie outside the time window, must be subtracted from the current accumulation texture. Utilizing this method, an unlimited number of gaze points can be accumulated for heatmap visualization as long as the camera is not moving.

Extending this approach by using a texture atlas similar to the one Stellmach used in [2010], the projection results can be stored in a texture representing the surfaces. This facilitates reusing previous computations even if the virtual camera changed and might lead to a significant improvement in interactivity for visualizing heatmaps consisting of an arbitrary number of gaze points.

4 Conclusion

A novel real-time capable approach for realistic heatmap generation and visualization of 3D gaze data was presented. In contrary to state-of-the-art approaches, projecting the tracked person's gaze into the scene on a per-fragment basis and considering occlusions using a shadow mapping algorithm yields a correct visualization of the perceived scene independent of the tessellation of the underlying scene geometry. On the other hand, in addition to typical shadow mapping problems, the use of shadow mapping for occlusion handling leads to a huge graphics memory demand which can be minimized using e.g. texture compression methods. That is why further work should be done evaluating the trade-off between computation time and needed memory using different shadow algorithms for the occlusion test.

Implementation details including two acceleration techniques facilitating an interactive steering of the online and offline gaze analysis in 3D scenarios like e.g. in museums were presented. The first method limits the number of fragments for which the projection computations need to be executed. The second acceleration method reuses previous accumulation results on a per-fragment basis, so that only for new gaze points the projection computations need to be performed and accumulated. This allows for real-time generation and visualization of an arbitrary number of gaze points as long as the virtual camera does not move. To also facilitate these benefits when the camera is moving, an extension to this technique was proposed.

References

CAMPBELL, M., MORAN, A., AND KENNY, I. 2013. Characteristics of expertise in able and disabled elite golfers: the role of vision and technique. In *British Association of Sport and Exercise Sciences Annual Conference*, British Association of Sport and Exercise Sciences. `http://hdl.handle.net/10344/3372` (last accessed 2014-01-30).

DUCHOWSKI, A. T., PRICE, M. M., MEYER, M., AND ORERO, P. 2012. Aggregate gaze visualization with real-time heatmaps. In *Proceedings of the Symposium on Eye Tracking Research and Applications*, ACM, New York, NY, USA, ETRA '12, 13–20.

GUESTRIN, E., AND EIZENMAN, E. 2006. General theory of remote gaze estimation using the pupil center and corneal reflections. *Biomedical Engineering, IEEE Transactions on 53*, 6, 1124–1133.

HAMMER, J. H., MAURUS, M., AND BEYERER, J. 2013. Real-time 3D gaze analysis in mobile applications. In *Proceedings of the 2013 Conference on Eye Tracking South Africa*, ACM, New York, NY, USA, ETSA '13, 75–78.

HARWOOD, T., JONES, M., AND CARRERAS, A. 2013. Shedding light on retail environments. In *Proceedings of the 2013 Conference on Eye Tracking South Africa*, ACM, New York, NY, USA, ETSA '13, 2–7.

HOLMQVIST, K., NYSTRM, M., ANDERSSON, R., DEWHURST, R., HALSZKA, J., AND VAN DE WEIJER, J. 2011. *Eye Tracking : A Comprehensive Guide to Methods and Measures*. Oxford University Press, Oxford [u.a.].

PALETTA, L., SANTNER, K., FRITZ, G., MAYER, H., AND SCHRAMMEL, J. 2013. 3d attention: measurement of visual saliency using eye tracking glasses. In *CHI '13 Extended Abstracts on Human Factors in Computing Systems*, ACM, New York, NY, USA, CHI EA '13, 199–204.

PFEIFFER, T. 2012. 3D Attention Volumes for usability studies in virtual reality. In *Proceedings of the 2012 IEEE Virtual Reality*, IEEE Computer Society, Washington, DC, USA, VR '12, 117–118.

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

STELLMACH, S., NACKE, L., AND DACHSELT, R. 2010. Advanced gaze visualizations for three-dimensional virtual environments. In *Proceedings of the 2010 Symposium on Eye-Tracking Research &Applications*, ACM, New York, NY, USA, ETRA '10, 109–112.

TOYAMA, T., KIENINGER, T., SHAFAIT, F., AND DENGEL, A. 2012. Gaze guided object recognition using a head-mounted eye tracker. In *Proceedings of the Symposium on Eye Tracking Research and Applications*, ACM, New York, NY, USA, ETRA '12, 91–98.

WILLIAMS, L. 1978. Casting curved shadows on curved surfaces. In *Proceedings of the 5th annual conference on Computer graphics and interactive techniques*, ACM, New York, NY, USA, SIGGRAPH '78, 270–274.