International Conference on Artificial Reality and Telexistence
Eurographics Symposium on Virtual Environments (2016)
D. Reiners, D. Iwai, and F. Steinicke (Editors)

Real-Time 3D Peripheral View Analysis

M. M. Moniri, A. Luxenburger, W. Schuffert and D. Sonntag

German Research Center for Artificial Intelligence (DFKI GmbH)
Saarbruecken, Germany

Abstract

Human peripheral vision suffers from several limitations that differ among various regions of the visual field. Since these limitations result in natural visual impairments, many interesting intelligent user interfaces based on eye tracking could benefit from peripheral view calculations that aim to compensate for events occurring outside the very center of gaze. We present a general peripheral view calculation model which extends previous work on attention-based user interfaces that use eye gaze. An intuitive, two dimensional visibility measure based on the concept of solid angle is developed for determining to which extent an object of interest observed by a user intersects with each region of the underlying visual field model. The results are weighted considering the visual acuity in each visual field region to determine the total visibility of the object. We exemplify the proposed model in a virtual reality car simulation application incorporating a head-mounted display with integrated eye tracking functionality. In this context, we provide a quantitative evaluation in terms of a runtime analysis of the different steps of our approach. We provide also several example applications including an interactive web application which visualizes the concepts and calculations presented in this paper.

Categories and Subject Descriptors (according to ACM CCS): H.5.1 [Information Interfaces and Presentation]: Multimedia Information Systems—Artificial, augmented, and virtual realities

1. Introduction

Gaze input plays an important role in novel user interfaces. In virtual environments eye gaze is used for instance to improve a player's performance in games [Nav14] or as an active input modality [JGL08].As a matter of fact, many of implemented applications are limited to line of sight, since they restrict themselves to information available in the very center of gaze while excluding visual stimuli in peripheral regions. According to Barfield et al. [BHBK95], the human visual field (HVF) spans a cone-like structure in 3D space with horizontal and vertical opening angles of 190 and 135 degrees, respectively. The idea of this paper is to use 3D information of the surroundings (in virtual or real environments) for a peripheral visual perception analysis of a human observer in real-time by combining said information with a user's gaze direction in the environment while considering a 3D human visual field (HVF) model.

We present a view calculation model which provides a visibility measure for multiple objects in real-time. For this purpose, we implement a binocular model of the HVF based on two angular dimensions. With this parametrization, our task of mapping arbitrary objects in the environment to characteristic regions of the visual field constitutes a 3D problem. In this context, we opt for the concept of solid angle as a 2D angular measure in 3D space in order to determine the fractions an object occupies in different regions of the visual field from the observer's gaze direction and the position of the object. Obtained fractions are weighted considering the visual acuity in each region of the visual field to determine the total visibility of the object.

In order to prove the feasibility of our visibility measure for different analysis and interaction purposes in intelligent user interfaces, we deploy our concepts to a virtual reality (VR) application. For this purpose, we use the Unity 3D game engine[†]. We build a virtual environment based on real places, in which the user has the possibility to freely look around. Here, we compare the run-time performance of our approach against traditional peripheral analysis approaches like eccentricity, i.e. the angular distance of the object to the center of gaze. In this paper,

- we combine mathematical representations of human peripheral vision, gaze tracking and sensor-captured 3D object information such as distance and outline within a novel method for real-time peripheral view monitoring.
- We prove the real-time performance and scalability of our approach for multiple objects of interest by providing a quantitative evaluation in terms of a detailed runtime analysis.
- Finally, we visualize the concepts and numerics of our approach in an interactive web application. Therein as well as in supplementary video material, we further show how our visibility mea-

[†] https://unity3d.com/

DOI: 10.2312/egve.20161432

sure can serve for assistive, procative system functionalities of modern intelligent user interfaces in virtual and real environments.

Current available methods considering human peripheral perception use the position of an object relative to the user's gaze (eccentricity) and/or the horizontal or vertical size of an object in degrees. The novelty of our approach is that we incorporate the solid angle of the objects in our calculations and combine them with different visual acuities in the HVF. This way, we use a measure which determines the size of the object as it appears to a user.

In the following, we will discuss existing work that has been done in the context of gaze-aware user interfaces and modeling human peripheral perception. Having this overview, we will present a detailed description of our peripheral view calculation model together with its implementations. In order to show the feasibility of our approach, we will show possible applications and conduct a runtime study to prove that it can be applied in realtime.

2. Related Work

Gaze input plays an important role in novel user interfaces. Examples include a wide range of applications, such as systems which use the eye gaze to enable severely disabled individuals to control electronic devices [SGP07], responsive texts which provide interaction possibilities [BBS*10], or systems that use an eye tracking interface to store pieces of forgotten information and present them back to the user later [OTSK14]. Eye gaze is also used in virtual environments, for instance to improve a player's performance in games [Nav14], or as active input modality [JGL08]. As a matter of fact, many of these applications are limited to line of sight, since they restrict themselves to information available in the very center of gaze while excluding visual stimuli in peripheral regions. As peripheral vision plays an important role in spatial awareness, there are systems that extend the visual field of head-mounted displays by additional optical elements (for example [LBAS16] and [XB16]). According to Barfield et al. [BHBK95], the human visual field spans a cone-like structure in 3D space with horizontal and vertical opening angles of 190 and 135 degrees, respectively.

The paradigm of different HVF models is based on the limitations of human vision and associated areas in the visual field. Figure 1 illustrates two different approaches for modeling the HVF. The model of Hatada et al. [HSK80] divides the visual field into four regions with different visual capabilities. For this purpose, they use both the vertical and the horizontal spherical angle to define these four regions. As a result, each of these regions features an elliptic shape in 2D while constituting a cone-like structure in 3D. Similarly, the model of Komatsubara [Kom08] divides the HVF into four regions while discriminating different capabilities of human vision with respect to recognizing text, shape and color. However, angular parametrizations are restricted to the horizontal dimension of the visual field. Barfield et al. [BHBK95] propose a field of view model which considers also both spherical angles for defining the blind area, binocular and monocular fields of view as well as the total vertical degrees of sight.

Depending on their properties, visual field models are used by researchers in different application categories to design interfaces

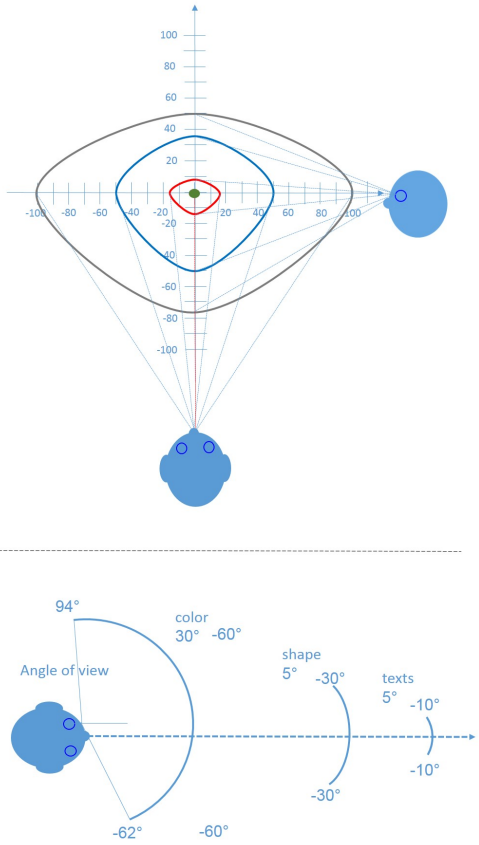

Figure 1: *Models of the human visual field.* Top: *Model of Hatada et al. using horizontal and vertical angles for parametrization.* Bottom: *Model of Komatsubara. Visual regions are defined over one angular dimension.*

based on a user's visual focus-of-attention. Ishiguro et al. [IR11] propose a gaze-operated information presentation method for mobile augmented reality systems utilizing the visual field model of Komatsubara. The gaze direction is used to control the level of detail of an overlaid information located in the peripheral area of a user's field of view. While, in this context, the main focus lies on preventing user distractions by shifting annotation information to peripheral regions with lower visual capabilities, a detailed peripheral visibility analysis in turn could identify less perceived objects of potential impact for the user while visually inducing a needed shift of attention. Tönnis et al. [TK14] propose a similar concept for information presentation at an angular offset to the user's line of sight. The information is placed directly at the axis of sight for a short time when it is demanded by the user. There are also user interfaces which use gaze without considering any peripheral view model for their applications. The potential of these interfaces was discovered by early researches from Jacob [Jac90] and Starker et al. [SB90]. In the context of more recent work, Masayuki et al. [NTT14] describe an information presentation method for head-mounted displays which is based on different gaze situations and

surrounding environments. Alt et al. [ASA*14] show that it is possible to use eye tracking systems in 3D stereoscopic displays with good accuracy. Lee et al. [LPL*11] present an interactive system that uses a user's gaze in an augmented reality application. Toyama et al. [TDSK13] present different systems based on eye tracking for assisting reading activity, facilitating interaction with virtual elements such as text or buttons by measuring eye convergence on objects at different depths [TSOK14], and for attention engagement and cognitive state analysis [TSOK15].

The mentioned systems either use directional gaze data only or in combination with the Komatsubara HVF model [Kom08]. As mentioned previously, this model divides the HVF based on horizontal angles (eccentricity). In other words, these models neither integrate the full visual appearance of an object nor a full 3D HVF model. Related systems either allow a direct interaction with the object (when the center of the gaze is on the object), or they consider the angular distance of the object to the user's focus point. The problem we see with eccentricity-based approaches is that they can hardly differentiate between objects with different sizes in terms of solid angle in a user's peripheral vision, especially in vertical directions. While using such standard metrics works well in a graphics context like for example foveated rendering [GFD*12] where objects are mapped to different circular 2D eccentricity layers centered around the current gaze point, we believe that 3D representations of both, visual measures and the HVF are more feasible in the context of 3D scene understanding if not tremendously more expensive to compute.

In this paper, we want to pave the way for bringing 3D peripheral view calculations into gaze-aware intelligent user interfaces. In this context, two important choices have to be made. The first one is concerned with an appropriate representation of the HVF in 3D that is applicable to a wide range of applications. The second choice refers to a suitable visibility measure associated with the projection of an observed object from the environment to different regions of the visual field.

Concerning the type of the underlying visual perception model, we have to differentiate between the visual models that use just one spherical angle (horizontal or vertical eccentricity) and those methods that use both angles to define their paradigms. According to our purpose, we opt for the latter while excluding 2D models of the HVF, which basically neglect one viewing dimension. Thus, we use the 3D perceptual model of Hatada et al.. Regarding the analysis of the appearance of an object, we are interested in the area an object occupies in regions of the visual field from the observer's point of view. Hence, solid angle seems to be an appropriate choice for the basis of our visibility measure. Solid angle together with the 3D perceptual model of Hatada et al., constitute the theoretical foundation of our general 3D peripheral view calculation model, for which we give detailed explanations in the following sections.

3. Peripheral View Calculation Model

In order to describe our peripheral view calculation model, we proceed as follows. First, we illustrate the Hatada model while giving a detailed review of defined regions of the visual field with their associated capabilities. We examine the concept of solid angle and, in this respect, explain how to calculate intersections between each visual field defined by the Hatada model and an arbitrary object of the 3D environment. An algorithmic realization of our model involves discrete aspects of solid angle computation where we reduce calculations for complex objects to primitive types by dividing an object's outline into rectangular patches. Finally, we relate the obtained measures to the concept of visual acuity to determine our final visibility measure.

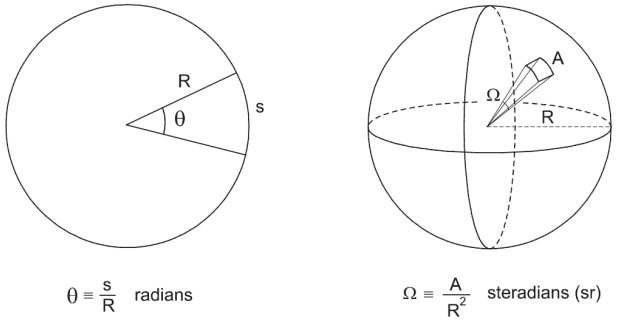

Figure 2: *The solid angle Ω is defined as 2D equivalent of a conventional angle θ. It determines how large an object appears to an observer.*

3.1. 3D Model of the Human Visual Field

Although the concepts of visibility calculation presented in this paper can be applied to any model that states a 3D representation of the HVF, we consider the Hatada model for further investigations, due to its detailed descriptions with respect to the characteristics of the defined regions and its exhaustive capture of the HVF following a two-dimensional angular parametrization. The model divides the visual field into the following four regions with corresponding angular boundaries (see Figure 1):

- The *discriminatory* visual field (3° circular).
- The *effective* visual field (3° to 15° horizontally on each side, 8° upwards, and 12° downwards).
- The *induced* visual field (15° to 50° horizontally on each side, 8° to 35° upwards, 12° to 50° downwards).
- The *supplementary* visual field (50° to 100° horizontally on each side, 35° to 50° upwards, 50° to 75° downwards).

As the name states, in the *discriminatory* visual field, an observer has high-precision discriminatory capabilities and perceives detailed information accurately with a visual acuity of over 0.5. Within the *effective* visual field, the visual acuity falls to about 0.1, while the discrimination of a simple figure can still be accomplished in a short period of time. This is the range within which an observer looks naturally at an object without head movement and is able to effectively process the information perceived. The *induced* visual field constitutes the range within which an observer has discriminatory capabilities to the extent of being able to recognize the existence of a visual stimulus. Hence, information displayed to the user which falls in this range should feature a reduced

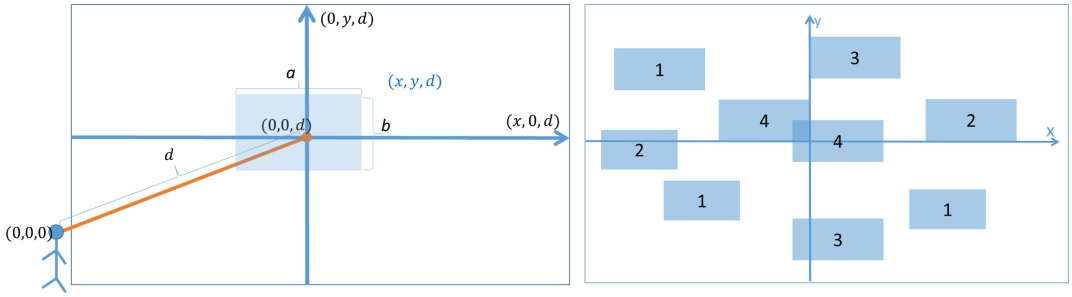

Figure 3: Left: *User's head position as center of the 3D environment. The collision plane is at distance* d *from user's head on the* z *axis. The gaze direction yields the origin of the collision plane. The bounding box of the object which is located at the gaze center has dimensions* a *and* b. Right: *Examples of the four different categories for the position of an object's bounding box (the numbers indicate the categories).*

level of detail in terms of minimalistic representations. The HVF is complemented in terms of the *supplementary* visual field which states a range with no direct functional role in the perception of visual information. All it provides is a supplementary function in the sense that a shift of the observer's gaze can be aroused in response to abrupt stimuli. Consequently, we will exploit this characteristic in our implemented use case to invoke events triggering a shift of attention.

Having a suitable 3D representation of the HVF at hand, we can now establish the mathematical basis for our peripheral view calculation model. This involves defining solid angle-based visibility measures for both the visual fields as well as discretized target objects that are projected into these fields.

3.2. Solid Angle-based Visibility Measure

A solid angle Ω [Qui06] constitutes a two-dimensional angle in 3D space that is subtended by an object from a specific point of view. This way, it provides an intuitive measure for how large an object appears to an observer. In the International System of Units, a solid angle is expressed in a dimensionless unit called a steradian (*sr*). Basically, the concept of solid angle is defined analogously to that of a conventional 1D angle θ as illustrated by Figure 2. In this respect, the measure states the fraction of a unit sphere's area covered by an observed object rather than the fraction of a circle's circumference. The general equation for calculating the solid angle of an arbitrary oriented surface subtended at a point [Mas57] is given by

$$\Omega = \iint \sin(\theta)\ d\theta\, d\phi \tag{1}$$

where θ and ϕ state the polar and azimuthal angles of a spherical coordinate system, respectively. The total solid angle of a unit sphere is 4π *sr*. In the following, we provide an analytic solution for calculating the solid angle of a visual field region.

3.3. Solid Angle of an Object's Bounding Box

In the following, the solid angle of the bounding box of an arbitrary object in a 3D environment is calculated. The position of the user's head is assumed to state the origin of a Cartesian coordinate system. At distance d from the user we define a "collision plane" which is perpendicular to the user's head orientation and whose origin is determined by the user's colliding gaze vector. The 2D projection of the 3D oriented bounding box of the target object onto this plane is assumed to have width a and height b (see Figure 3 (left)). In the Figure 3 (left), the user is looking at the center of an object. If the bounding box of the object is positioned exactly at the center of the collision plane, the solid angle of this rectangular shape can be calculated according to Mathar's formula [Mat14]:

$$\Omega(a,b,d) = 4 \arccos\left(\frac{\sqrt{1+(\frac{a}{2d})^2+(\frac{b}{2d})^2}}{\sqrt{1+(\frac{a}{2d})^2}\sqrt{1+(\frac{b}{2d})^2}} \right). \tag{2}$$

Note that this is the case when the user is looking exactly at the middle of the object. However, this formula can not be applied in its current form if the bounding box of the object is not positioned exactly at this point. This is the case when the observer's gaze is focused at another point in the environment, which shifts the target object to peripheral regions of the user's visual field. Objects in the environment can have different positions relative to local coordinate axes of the collision plane. As illustrated by Figure 3 (right), four different positions of the object's bounding box with respect to the local coordinate axes of the collision plane can be categorized:

1. The rectangle does not intersect with any axes.
2. The rectangle intersects only with the x-axis.
3. The rectangle intersects only with the y-axis.
4. The rectangle intersects with both axes.

The projection of any object in the 3D environment on the collision plane will fall into one of these four categories. The center of the coordinate system on the collision plane (where x and y axis cross) is the point where the observer is looking. The solid angle of each of these rectangles can be calculated with different variations (and combinations) of the Mathar's formula [Mat14].

3.4. Intersection of Solid Angles

In order to determine the intersection of the solid angles of the different visual field regions and the solid angle of the object's bounding box, the ellipses of the visual fields have to be projected onto the collision plane. If we assume a visual field with parameters α and β at distance d to the user to be centered at the origin of the local coordinate system having its major and minor axes aligned with the local axes, a mathematical representation is given by

$$\left(\frac{x}{\tan(\alpha)\,d}\right)^2+\left(\frac{y}{\tan(\beta)\,d}\right)^2\leq 1 \tag{3}$$

where a 2D point $(x,y)^\top$ that fulfills this equation is located inside the associated elliptic region. It should be noted that this condition has to be adapted for each half ellipse with its corresponding angles given by the Hatada model. The projected 2D bounding box of the object is divided into very small rectangles (in the Cartesian coordinate system). For the center of each of these sample patches, we use the mentioned intersection condition in order to determine the corresponding region of the HVF the sample is located in. In the case of an intersection, we calculate the solid angle of the intersecting rectangles while accumulating them. The result of this accumulation is also a solid angle which determines the amount of intersection between the corresponding visual field and the bounding box of the object (see Figure 4).

3.5. Integration of Visual Acuity

Visual acuity states an important factor influencing the visibility. As Groß mentions in [Gro94], the visual acuity V is not equal on all parts of the retina, its maximum lies in the center of the retina and decreases towards the periphery. V can be measured by the inverse of the minimum visual angle α achievable when detecting a target. A numerical approximation is given by

$$V(\alpha)\approx c_1+\frac{c_2}{\alpha+c_3} \tag{4}$$

with constants

$$c_1=-0.0323\ ,\ c_2=0.0524\ ,\ c_3=0.0507\ .$$

Note that so far this provides a one-dimensional measure defined on the unit interval for visual angles ranging from 0°, yielding a maximized acuity for the very center of gaze, to 90°, where visual observations become impossible. As already pointed out by Groß, it makes sense for each solid angle area covered on the retina to be weighted with the corresponding value of the visual acuity. We transfer these ideas to our model by weighting the fraction of the solid angle an object subtends in a visual field region with the visual acuity associated with this region, which is determined from half of the corresponding total vertical opening angle. This way, combining the concepts of solid angle and visual acuity, we end up with a two-dimensional visibility measure assigning weights of an appropriate scale to arbitrary objects in 3D space with respect to their position in the HVF.

3.6. Algorithm

As mentioned before, the different steps for determining the visibility are the following: We calculate the intersection of the solid angle of each peripheral field region and the object of interest. Then, the different small fractions of this intersection are weighted according to the visual acuity in different regions of the HVF. As a result, we get the total solid angle of the object, the total visibility of the object based on visual acuity, and also the percentage of each of these values in each visual field region. This information can be used in different intelligent user interfaces to analyze the visibility of relevant objects in the environment and also to design interaction based on the visibility of those objects. In the following section, we show a car simulation application which uses this information for proactive system behaviors which assist a driver's visual system in a daily traffic situation. We also introduce an interactive web application, which provides a visualization based on the calculations of the algorithm.

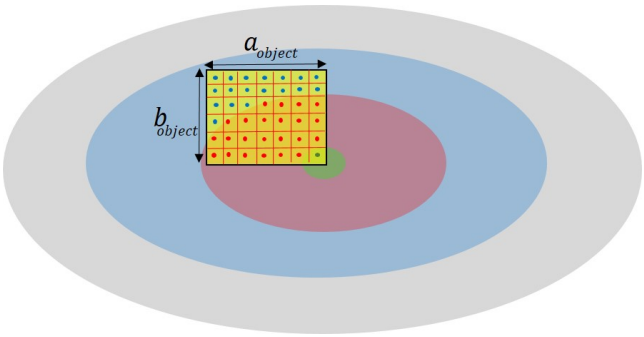

Figure 4: *The solid angle of the intersection is calculated by accumulating the solid angles of the small rectangles of the object's bounding box which intersect with respective regions of the visual field.*

4. Applications and Evaluation

Any application based on eye tracking which has interest in the visibility of relevant objects in the peripheral visual field can benefit from the provided concepts and calculations. In this section, we describe our application, which implements the described algorithm. In addition we provide a quantitative evaluation in terms of a runtime analysis of the different steps of our approach. We also briefly introduce our online tool, which provides the possibility for researchers to become more familiar with the concepts and calculations presented in this paper.

4.1. Simulated Automotive Use Case

Our VR application implements a simulated automotive use case. It includes an outdoor scene consisting of a 3D model of our campus with 492.3 K vertices, which has been reconstructed based on point clouds obtained from several laser scans. In this model we have also integrated a scanned model of our test vehicle, as well as 3D models of other vehicles, pedestrians, and other small objects like trees, road signs, traffic lights, etc. Our VR setup includes a special version of the Oculus Rift DK2 system with an integrated

Figure 5: *Our simulated automotive use case: the visibility of relevant objects (pedestrian, traffic light) in the peripheral view of the driver serve as input for assistive system functions (attention shift).*

eye tracker, which is commercially available[‡]. All implementations are based on the Unity 3D game engine. In our scenario, users play the role of a driver sitting in a car in front of a red traffic light. They are able to freely look around within the university campus. Regarding the different objects in the scene, some of them are tagged as "target object". Those objects are considered to be important in the current context. Users are expected to focus their attention and gaze on the frontal scenery or the red traffic light. In the moment it switches to green color and users are expected to move forward, a pedestrian enters the scene unexpectedly crossing the street. Consequently, besides the traffic light, the pedestrian constitutes a target object of high importance, as he or she induces a dangerous situation being situated in the user's peripheral visual field and thus barely perceived. The situation is depicted by Figure 5. Both target objects are marked with their corresponding projected bounding box which serves as input for our algorithm. In addition, one can see the projected, elliptic visual field regions which are positioned around the center of gaze (red dot) and visualized with different colors. In each frame-based pass of the main application loop, our method delivers visibility measures for each of the target objects (OOEs) and enables the system to trigger a shift of attention, in case a highly relevant object features a low visibility for the user. We encourage our readers to also have a look at the supplemented, digital video material which visualizes the described use case.

4.2. Performance Analysis

In this section, we provide a quantitative evaluation of the different steps of our algorithm and give several benchmarks for a different number of objects of interest (OOEs) and sampling patches. The goal of our evaluation is to show the scalability and real-time performance of our approach for a varied number of OOEs. The corresponding runtimes have been derived from simple time stamps which were placed before and after the invocation of each procedure. Their differences encode absolute time spans in milliseconds.

‡ http://www.smivision.com/en.html

# of patches	FPS	Sampling Time (ms)	Solid Angle Calculation Time (ms)
16 patches	75.04	0.0006	0.05380
100 patches	75.14	0.003	0.3017
400 patches	74.98	0.0035	1.0496
1600 patches	75.01	0.0184	3.9594
6400 patches	50.4	0.065	16.2518

Table 1: *Computation times for a varied number of sample patches. This parameter determines the accuracy of the solid angle calculation.*

In order to reduce the influence of distortion factors like background processes, we averaged them over 10 frames and repeated each benchmark multiple times. Thus, the presented values constitute average values. Our hardware setup consists of an ASUS Desktop PC G20CB Series featuring an Intel Core i7-6700 CPU (3.40 GHz), 16 GB RAM, and a NVIDIA GeForce GTX 980 GPU. We measured the runtimes of each step of our algorithm including the projection of the 3D bounding boxes and the visual field regions onto a common plane, the sampling of the projected bounding boxes, the computation of the intersecting solid angle of the bounding box and each visual field, and finally the calculation of our acuity-based visibility measure for each object. For the evaluation we used the described 3D model of the university, as it constitutes a typical outdoor scene with usual street layout and traffic. In the benchmarking, we varied the number of sampling patches for each projected bounding box, which determines the accuracy of the patch-based solid angle computation, and also the number of OOEs. In the following, we will describe each separately.

Table 1 shows the measured runtimes for a varied number of sampling patches. The related scene contained one target object while featuring 508.3 K vertices in total. This means that for all variations the projection time is roughly the same (about 0.013 milliseconds). However, we find variations in the number of frames per second (FPS), the sampling time, and the time for calculating the solid angle. For all variations, the time for computing the eccentricity and the object's visibility were negligible as they were in the range of a few nanoseconds.

The number of rectangular sample patches of a projected bounding box is determined as the product of horizontal and vertical subdivisions. For example, in case of 4 horizontal and 4 vertical subdivisions, we have 16 equal-sized patches. Table 1 shows that an increased number of patches yields an increased sampling time, as expected, with a maximum computation time of 65 nanoseconds. The corresponding FPS value also does not undergo an explicit change until we reach a number of 6400 patches where it drops from 75 to 50.

The time for calculating the solid angle of the patches increases in the same manner as the number of the patches. Here, we can detect a linear relation between these two values. As we quadruple the number of the patches, the time for calculating the solid angle quadruples, too.

As we found a number of 100 patches for an OOE to constitute a reasonable compromise between the runtime (about 0.3 milliseconds) and the degree of approximation of the object's solid angle,

we fix the number of patches for the following experiment and focus on a variation of the number of object of interest. The following list shows our different variations in the number of OOEs (number of patches = 100):

- Baseline (0 target objects, 492.3 K vertices):
  The university model without any OOE
- Scene 1 (1 target object, 508.3 K vertices):
  The university model with one pedestrian as OOE
- Scene 2 (3 target objects, 510.1 K vertices):
  The university model with one pedestrian, one car, and one traffic sign as OOEs
- Scene 3 (9 target objects, 545.7 K vertices):
  The university model with three pedestrians, three cars, and three traffic sign as OOEs
- Scene 4 (27 target objects, 833.7 K vertices):
  The university model with twenty one pedestrians, three cars, and three traffic signs as OOEs

For all of the listed variations, the number of FPS was between 75.14 for one target object and 74.5 for 27 target objects. The projection time and the sampling time were also less than 0.017 milliseconds and 0.0049 milliseconds, respectively. The solid angle calculation time for all variations was below 0.308 milliseconds and the time needed for computing the visibility measure and the eccentricity were negligible as again in the range of a few nanoseconds.

4.3. Interactive Web Application[§]

The interactive web application shows a room with different objects inside. The user can freely move and look around in the room. The mouse cursor represents the center of the user's gaze. The borders of different peripheral fields are visualized with different colors around the gaze center. The dimensions of these fields are selected according to the peripheral field model of Hatada et al. [HSK80]. The green circle shows the discriminatory visual field, the red and blue regions represent the effective and the induced visual field, respectively. The supplementary visual field is illustrated with a black line, however, due to the camera features of the game engine, it is not always visible. In the control panel on the upper left side of the scene, it is possible to select a target object (OOE) from a selection of the objects available in the scene. In this case, the bounding box of the target object is calculated and visualized. The intersecting solid angles of the bounding box with different peripheral fields are also calculated. If the "Show Grid" checkmark is selected, the different intersecting regions are also visualized. In the control panel, it is possible to choose the number of the samples for dividing the object's bounding box in horizontal and vertical direction. In this respect, a higher number of samples leads to an increased accuracy of the calculations. The control panel also shows the following calculated information: the distance between the object and the observer, the total solid angle of the object, the percentage of the object visible to the user (interesting in the case of occlusion) and the visibility of the object considering the solid angle of the object and the visual acuity of the visual field. Each of the last three values is also calculated separately for each of the named fields of the Hatada model. Occlusion is also considered and all of the calculations are performed for the visible part of the object.

Figure 6: *Our web application provides visualization of the presented concept and calculations in terms of a simple example. The user can move freely in the scene and change settings in the control panel.*

As the user navigates through the scene or changes the gaze direction, the different measures are calculated and displayed in real-time. All of the calculations are based on the algorithm provided in this paper. The interactive website makes it possible to examine the changes in the different values as the position or the gaze direction of the user changes. Figure 6 shows a screenshot of this application. The web application can be reached under the address: madmacs.dfki.de.

5. Conclusion and Future Work

In this paper, we introduced a general peripheral view calculation model for gaze-aware intelligent user interfaces, incorporating physiological characteristics of human vision in terms of a binocular visual field model. The model allows for a visibility analysis of several observed objects in the human visual field considering the observer's head and gaze direction as well as the relative position and outline of an object. By combining the concepts of solid angle and visual acuity, we designed a visibility measure assigning weights of an appropriate scale to arbitrary objects in 3D space with respect to their position in the HVF. Our model could benefit various eye tracking applications in two ways. On the one hand, the real-time performance of the presented algorithm allows for the continuous monitoring of a user's peripheral vision. On the other hand, this monitoring paves the way for engineering assistive technologies featuring proactive and compensating system functions. We exemplified these concepts within an interactive, task-oriented VR application for a simulated automotive use case, incorporating a head-mounted display with integrated eye tracking functionality. In each frame-based pass of the main application loop, our method delivers visibility measures for each of the OOEs and enables the

[§] Requires Firefox browser with Unity 3D plug-in.

system to trigger a shift of attention, in case a highly relevant object features a low visibility for the user.

In the presented work, our focus was on a detailed derivation of a novel real-time 3D peripheral view calculation model and a contextual adaptation of our model for interactive, head-worn VR systems based on eye tracking. In this context, we used the benefits of a controlled environment to validate our core algorithm and the corresponding application. We consider user studies and a validation of our approach as an inevitable and promising step. In order to bring our model closer to real-world use cases, we will integrate it into a mixed reality setup. In particular, this involves the use of optical or video see-through head-mounted displays with integrated eye tracking. Alternatively, this can also be realised by integrating our approach in the platforms like Microsoft HoloLens[¶] and Google Tango Project[‖]. With the use of such devices which are able to reconstruct the 3D environment in real-time, our model becomes applicable to domains beyond virtual environments.

6. Acknowledgments

This work was funded by the German Ministry of Education and Research (grant number 01IW14003).

References

[ASA*14] ALT F., SCHNEEGASS S., AUDA J., RZAYEV R., BROY N.: Using eye-tracking to support interaction with layered 3d interfaces on stereoscopic displays. In *Proceedings of the 19th international conference on Intelligent User Interfaces* (2014), ACM, pp. 267–272. 3

[BBS*10] BIEDERT R., BUSCHER G., SCHWARZ S., HEES J., DENGEL A.: Text 2.0. In *CHI'10 Extended Abstracts on Human Factors in Computing Systems* (2010), ACM, pp. 4003–4008. 2

[BHBK95] BARFIELD W., HENDRIX C., BJORNESETH O., KACZMAREK K. A.: Comparison of human sensory capabilities with technical specifications of virtual environment equipment. *Presence: Teleoperators and Virtual Environments 4*, 4 (1995), 329–356. 1, 2

[GFD*12] GUENTER B., FINCH M., DRUCKER S., TAN D., SNYDER J.: Foveated 3d graphics. *ACM Trans. Graph. 31*, 6 (Nov. 2012), 164:1–164:10. URL: http://doi.acm.org/10.1145/2366145.2366183, doi:10.1145/2366145.2366183. 3

[Gro94] GROSS M.: *Visual computing: the integration of computer graphics, visual perception and imaging*. Springer-Verlag, 1994. 5

[HSK80] HATADA T., SAKATA H., KUSAKA H.: Psychophysical analysis of the "sensation of reality" induced by a visual wide-field display. *Smpte Journal 89*, 8 (1980), 560–569. 2, 7

[IR11] ISHIGURO Y., REKIMOTO J.: Peripheral vision annotation: noninterference information presentation method for mobile augmented reality. In *Proceedings of the 2nd Augmented Human International Conference* (2011), ACM, p. 8. 2

[Jac90] JACOB R. J.: What you look at is what you get: eye movement-based interaction techniques. In *Proceedings of the SIGCHI conference on Human factors in computing systems* (1990), ACM, pp. 11–18. 2

[JGL08] JIMENEZ J., GUTIERREZ D., LATORRE P.: Gaze-based interaction for virtual environments. *J. UCS 14*, 19 (2008), 3085–3098. 1, 2

[Kom08] KOMATSUBARA A.: *Human error*. Maruzen, 2008. 2, 3

[LBAS16] LUBOS P., BRUDER G., ARIZA O., STEINICKE F.: Ambiculus: Led-based low-resolution peripheral display extension for immersive head-mounted displays. In *Proceedings of the Virtual Reality International Conference (VRIC)* (2016), ACM, p. ACCEPTED. URL: http://basilic.informatik.uni-hamburg.de/Publications/2016/LBAS16. 2

[LPL*11] LEE J. Y., PARK H. M., LEE S. H., KIM T. E., CHOI J. S.: Design and implementation of an augmented reality system using gaze interaction. In *Information Science and Applications (ICISA), 2011 International Conference on* (April 2011), pp. 1–8. doi:10.1109/ICISA.2011.5772406. 3

[Mas57] MASKET A. V.: Solid angle contour integrals, series, and tables. *Review of Scientific Instruments 28*, 3 (1957), 191–197. 4

[Mat14] MATHAR R. J.: *Solid angle of a rectangular plate*. Max-Planck Institute of Astronomy, Königstuhl, 2014. URL: http://www.mpia.de/~mathar/public/mathar20051002.pdf. 4

[Nav14] NAVARRO D.: Improving player performance by developing gaze aware games. 1, 2

[NTT14] NAKAO M., TERADA T., TSUKAMOTO M.: An information presentation method for head mounted display considering surrounding environments. In *Proceedings of the 5th Augmented Human International Conference* (2014), ACM, p. 47. 2

[OTSK14] ORLOSKY J., TOYAMA T., SONNTAG D., KIYOKAWA K.: Using eye-gaze and visualization to augment memory. In *Distributed, Ambient, and Pervasive Interactions*. Springer, 2014, pp. 282–291. 2

[Qui06] QUIMBY R. S.: *Photonics and lasers: an introduction*. John Wiley & Sons, 2006. 4

[SB90] STARKER I., BOLT R. A.: A gaze-responsive self-disclosing display. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (New York, USA, 1990), CHI '90, ACM, pp. 3–10. 2

[SGP07] SHI F., GALE A., PURDY K.: A new gaze-based interface for environmental control. In *Universal Access in Human-Computer Interaction. Ambient Interaction*, Stephanidis C., (Ed.), vol. 4555 of *Lecture Notes in Computer Science*. Springer Berlin Heidelberg, 2007, pp. 996–1005. 2

[TDSK13] TOYAMA T., DENGEL A., SUZUKI W., KISE K.: Wearable reading assist system: Augmented reality document combining document retrieval and eye tracking. In *Document Analysis and Recognition (ICDAR), 2013 12th International Conference on* (2013), IEEE, pp. 30–34. 3

[TK14] TÖNNIS M., KLINKER G.: Boundary conditions for information visualization with respect to the user's gaze. In *Proceedings of the 5th Augmented Human International Conference* (2014), ACM, p. 44. 2

[TSOK14] TOYAMA T., SONNTAG D., ORLOSKY J., KIYOKAWA K.: A natural interface for multi-focal plane head mounted displays using 3d gaze. In *Proceedings of the 2014 International Working Conference on Advanced Visual Interfaces* (2014), ACM, pp. 25–32. 3

[TSOK15] TOYAMA T., SONNTAG D., ORLOSKY J., KIYOKAWA K.: Attention engagement and cognitive state analysis for augmented reality text display functions. In *Proceedings of the 20th International Conference on Intelligent User Interfaces* (2015), ACM, pp. 322–332. 3

[XB16] XIAO R., BENKO H.: Augmenting the field-of-view of head-mounted displays with sparse peripheral displays. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems* (New York, NY, USA, 2016), CHI '16, ACM, pp. 1221–1232. URL: http://doi.acm.org/10.1145/2858036.2858212, doi:10.1145/2858036.2858212. 2

[¶] https://www.microsoft.com/microsoft-hololens/en-us

[‖] http://get.google.com/tango/