Chapter 25
Visual Attention and Gaze Behavior in Games: An Object-Based Approach

Veronica Sundstedt, Matthias Bernhard, Efstathios Stavrakis, Erik Reinhard, and Michael Wimmer

Take Away Points

1. Although eye-tracking can tell us *where* a user is looking, understanding *what* a user is looking at can be more insightful in game design
2. Different levels of abstraction can be used to represent a stimulus in gaze analysis, ranging from pixels, shapes and polygons to objects and even semantics
3. By gaining access to the internal representation of scenes, it is possible to map gaze positions to objects and object semantics

25.1 Introduction

In the design of interactive applications, notably games, a recent trend is to understand player behavior by investigating telemetry logs as is the focus of many chapters in this book or by integrating the use of psychophysics as is the subject of Chaps. 26 and 27.

V. Sundstedt (✉)
School of Computing, Blekinge Institute of Technology, Karlskrona, Sweden
e-mail: vsu@bth.se; veronica.sundstedt@bth.se; sundstedt@gmail.com

M. Bernhard
Institute for Computer Graphics and Algorithms,
Vienna University of Technology, Vienna, Austria

E. Stavrakis
Department of Computer Science, University of Cyprus, Nicosia, Cyprus

E. Reinhard
Department for Computer Graphics, Max Planck Institute for Informatics,
Saarbrücken, Germany

M. Wimmer
Institute for Computer Graphics and Algorithms,
Vienna University of Technology, Vienna, Austria

M. Seif El-Nasr et al. (eds.), *Game Analytics: Maximizing the Value of Player Data*,
DOI 10.1007/978-1-4471-4769-5_25,

In addition to these valuable methods, measuring where players are likely to focus could be a very useful tool in the arsenal of game designers. This knowledge can be utilized to help game designers decide how and where to allocate computing resources, such as rendering and various kinds of simulations of physical properties. This leaves as many computing cycles as possible free to carry out other tasks. Therefore, the perceived realism of a game can be increased by perceptually optimizing calculations that are computationally intensive, including physically based lighting, animations (e.g. ray-tracing Cater et al. 2003, crowds of characters McDonnell et al. 2009), physically correct simulations of the interaction of materials (e.g. collision detection (O'Sullivan 2005), natural behavior of clothes or fluids etc.). Level-of Detail-variants of simulation or rendering techniques can be used in regions which are less attended by the player, while accurate simulations can be used within the expected focus of a user. Verifying or improving game mechanics and AI could be other uses.

The study of gaze behavior can provide insight about the visual attention of players and thus assist game designers in identifying problems with gameplay due to a misguided visual perception of the game environment. Moreover, knowing what a player does or does not notice can be used to control the difficulty of a game. For example, the designer may choose to make important task-relevant objects less apparent in the user's attentional field to increase the difficulty of the game, or accentuate them to decrease the difficulty. Other potentially useful computer graphics applications proposed so far include focus prediction for tone-mapping of high-dynamic range images (Rahardja et al. 2009), the selection of the optimal focal plane for depth-of-field effects (Hillaire et al. 2008) and the minimization of vergence-accommodation conflicts in stereo 3D to reduce visual fatigue (Lang et al. 2010). Further applications include the natural animation of eye-movements in agents (Itti et al. 2006) and estimating or increasing the visibility of in-game product placements (Chaney et al. 2004; Bernhard et al. 2011).

Eye-tracking can be used as a tool to study eye movements or gaze behavior (Duchowski 2003). There are many application areas for the use of eye tracking and it has previously seen extensive use in psychology, neuroscience, human factors, and human computer interaction. Eye tracking devices, commonly referred to as eye trackers, were intrusive and cumbersome to use at the beginning, but recent advancements in eye tracking technology have made it possible to use them effortlessly without distracting users. Although low-cost solutions have emerged, more robust and accurate eye tracking systems are still very expensive. Nevertheless, even with today's technology the eye tracking process still suffers from various limitations which have an impact on accuracy (Hansen and Ji 2010). Some of these issues are related to the calibration process, the ability to eye track different users, the fact that the eye is never completely still, and the extraction and interpretation of eye movements.

When an eye tracker is used to study gaze behavior in a computer game, the output data is essentially a sequence of gaze points defined by a 2D position on the display screen and a timestamp. With this information, one can establish *where* gaze was deployed in screen-space over time. Analysing gaze data for static stimuli can be time consuming, but it is even more difficult for dynamic stimuli (e.g. virtual environments such as games) (Ramloll et al. 2004; Stellmach et al. 2010b). A useful representation of gaze data are gaze point density distributions, which can quantify the amount of attention deployed to each region in the display. When a

Fig. 25.1 Example heatmap from one participant viewing game stimuli for 15 s. The visual representation of clustered areas indicate locations of a higher number of fixations

stimulus rendered to the display is static or its changes are very limited (e.g. in web pages), it is sufficient to compute gaze point density distributions in screen space. A prominent tool to illustrate screen-space gaze density distributions is a fixation map (Wooding and David 2002) or heatmap. A heatmap visualizes gaze point densities with colors such that warm colors encode high densities and cold colors low densities. An example of a heatmap can be seen in Fig. 25.1.

For computer games, we have to assume a dynamic stimulus, where temporal changes have a significant impact on the spatial distribution of gaze points on the screen. In this case one cannot accumulate gaze densities in screen space over long time periods because the viewpoint and the objects in the scene may considerably change their positions from one frame to the next. When spatial properties such as the viewpoint or object positions are changing frequently, it may not necessarily be appropriate to analyze *where* a user is looking. Instead, if we consider that semantic properties of scene objects are changing far less often, a more useful approach is to study *what* a user is looking at, especially since the meaning of game objects is supposed to have a major impact on the attention of a user.

25.1.1 Measuring "Where" But Analysing "What" Users Are Looking At

To analyze in a dynamic scene what a user is looking at, we need to record the changes in the display during the eye-tracking study. In the subsequent analysis, the recorded data is then used to reconstruct the frames depicted on the display in

temporal alignment with the corresponding gaze data. Thus, gaze analysis tools provide screen recording functions which capture the images rendered to the display during the experiment. The screen recording can then be played back as a video during the analysis stage. A synchronous visualization of the recorded gaze data superimposed on the playback of the corresponding stimuli provides an intuitive clue about the behavior of a particular participant. But with this functionality alone, one can just study the behavior of a particular subject in particular situations. In dynamically changing games, it is unlikely that different subjects are presented the same stimuli while playing a game. To complicate things further, even if the participant partakes in a number of gaming sessions, it is unlikely that the same sequence of in-game events will be triggered to generate the same stimuli. Moreover, encoding *what* a participant might actually be looking at mainly depends on the person who performs the analysis. For many purposes, an objective statistical evaluation, such as computing the gaze density distribution over different objects, might be preferable.

Commercial gaze analysis tools can accumulate gaze-points for manually defined regions of interest. To outline objects of interest in videos, the experimenter has to define regions of interest (e.g. bounding rectangles or polygons) around the objects on a frame-by-frame basis. This can be a tedious and time consuming procedure. To some extent, tools from computer vision, such as segmentation algorithms, could assist in this process. However, translating pixel regions to semantically encoded scenes remains a difficult problem in computer vision.

Fortunately, obtaining a semantic representation of the stimulus is significantly easier for computer games, where information about any game entity can be extracted from the game application directly. Game engines usually render each image from an object-based representation of a scene, from which semantic information can be obtained to a considerable extent. Recording the game engine's internal representations of game states allows the conservation of object-space information of the stimuli, which is otherwise very difficult to extract when only rendered images are available. Therefore, we could use these facilities to map gaze points back to the 3D objects that were observed during a gaming session. We assume that objects are modeled as semantically meaningful clusters of polygons, thereby allowing this approach to link gaze data to object semantics. This chapter outlines such an approach in more detail.

25.1.2 Overview

The remainder of the chapter is organized as follows. Section 25.2 introduces the reader to relevant concepts of human visual attention and eye movements. Eye tracking methodology is discussed in Sect. 25.3. The section also briefly describes some of the related work in studying visual attention and gaze behavior in computer games. Section 25.4 presents some related work in creating 3D gaze data visualizations and logging game data. Section 25.5 describes a unique pipeline that can be used to study

gaze behavior in games, while Sect. 25.6 details the underlying algorithms for mapping fixations to objects. Section 25.7 then discusses ways to use such mappings to collect statistics and draw conclusions. Representative examples of using this pipeline are presented in Sect. 25.8 and some limitations of the work in Sect. 25.9. Finally, Sect. 25.10 summarizes and discusses the work and highlights some of the key issues and important areas of further research in this emerging area.

25.2 Visual Attention and Eye Movements

Humans have different sensory systems which convert information from the environment into neural signals that are then interpreted by the brain. To derive meaning, the brain implements processes that select, organize, and interpret the information from our senses. To enable living in complex environments, humans rely strongly on vision, which consists of two broad components. The first is *perception*, which is pre-attentive. The second is *cognition*, which involves high-level processes, such as thought, reasoning, and memory (Palmer 1999). The delineation between these two is not sharp, and significant feedback and cross-talk exists between the two. When carrying out a task, the human visual perception aggregates low-level features into higher level representations, thus informing cognitive processes while affecting gaze direction. In turn, cognitive processes can guide perception, for instance by actively focussing attention on a particular part of a scene (Yarbus 1967).

Since the information-processing capacity of our brain is limited, incoming information has to be filtered so that we are able to process the most important sensory inputs. Visual attention is the control mechanism which selects meaningful inputs and suppresses those of low importance. Our eyes can sense image details only in a 2° foveal region, due to a rapid falloff of spatial acuity towards the periphery of the fovea. To reposition the image onto this area, the human visual system uses different types of eye movements. Saccades are fast and ballistic eye movements used to reposition the fovea. These movements are both voluntary and reflexive and last between 10 and 100 ms. There is virtually no visual information cognitively processed during a saccade (Duchowski 2003). Between eye movements, fixations occur, which often last for about 200–300 ms (Snowden et al. 2006). During a fixation, the image is held approximately still on the retina; the eyes are never completely still, but they always jitter using small movements called tremors or drifts (Snowden et al. 2006). According to Jacob and Karn (2003), a scan path is a spatial arrangement of a fixation sequence. A common way to visualize a *scan path*, or *gaze plot*, is to overlay a snapshot of the stimuli with fixations drawn as circles, as shown in Fig. 25.2. These circles are interconnected with lines that represent the saccadic eye movements. Their radius can be adjusted to indicate the duration an observer has been looking at that particular point.

The cognitive psychology and neuroscience literature contains a vast array of reports on models that try to predict the mechanisms of attention (Wolfe 2000). The most established is a model which divides attention into bottom-up and top-down

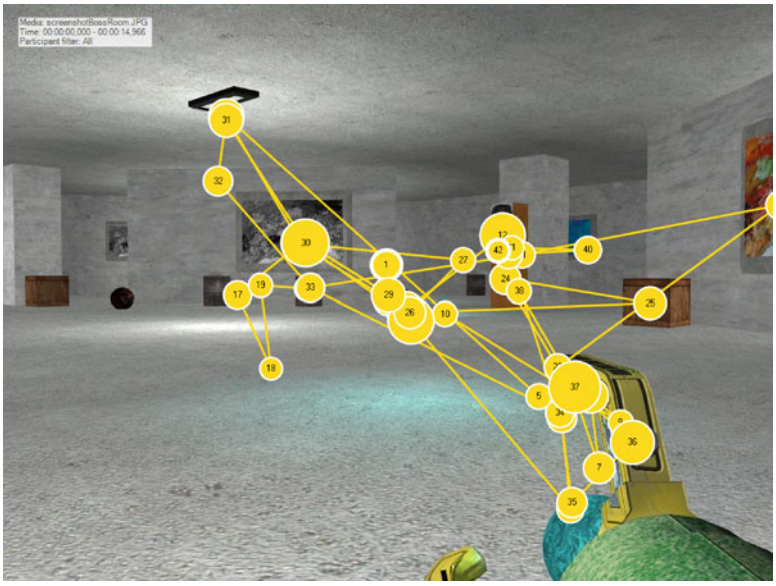


Fig. 25.2 Scan Paths: example scan path from an experiment (Bernhard et al. 2011) with a 3D First Person Shooter game featuring a dynamic camera. The *numbered circles* indicate successive fixations connected by *lines*, which denote *saccades*. The radius of each circle in the latter representation is proportional to the relative duration of each fixation

processes (James et al. 1890). In bottom-up processing, the visual stimulus captures attention automatically without volitional control (Itti et al. 1998). Low-level, bottom-up features which influence visual attention include contrast, size, shape, color, brightness, orientation, edges, and motion. In contrast, top-down processes focus on the observer's goal; they depend on the task. Low-level features in the environment that trigger pre-attentive focus are called *salient*. Features that attract attention as a result of performing a specific task are called *task relevant*. However, bottom-up and top-down processes cannot be separated perfectly, and there is much interaction between both (van Zoest and Donk 2004; Wolfe 2007).

Computational models have been developed which aim to predict which aspects of an image attract visual attention. The first models concentrated on modelling gaze behavior using low-level features, such as color, intensity and orientation (Treisman and Gelade 1980; Koch and Ullman 1985; Itti et al. 1998). Such models compute for each pixel of an image a measure of saliency, the result of which is called a *saliency map*. However, it has been shown that task-related gaze behavior can dominate over saliency (Land et al. 1999). Per-pixel measures of task relevance have more recently appeared, and these are called *task maps* (Cater et al. 2002; Navalpakkam and Itti 2005; Sundstedt 2007). Established theories about visual search assume that low-level features characteristic of target objects (e.g. color or intensity) are enhanced and guide the search (Wolfe 1994). A similar intuitive interpretation is that top-down control raises the saliency of important objects (Oliva

et al. 2003; Navalpakkam and Itti 2005; Elazary and Itti 2008). However, though there are reasonable theories for top-down mechanisms concerning visual search tasks, they do not directly explain how attention is deployed in complex and changing tasks, such as those occurring in computer games. Eye-tracking studies are a reasonable method for investigating top-down attention from the opposite perspective, that is, analyzing how visual attention behaves under particular complex stimuli and tasks. The following section will describe eye tracking methodology in more detail and how it has been used in relation to computer games.

25.3 Eye-Tracking Methodology and Games

Eye-tracking is a technology developed to monitor eye movements allowing us to determine where an observer is looking at a given time. An eye tracker is used to sample the state of the human eyes. For each sample, a *gaze point* (a 2D location in screen space) is estimated. This information can give us insight into what attracted the attention of an observer or what they found interesting (Duchowski 2003). Eye trackers measure the physical rotations of the eyes to determine the gaze direction. Gaze has also been referred to as the vector between the eye and the gaze point (Hornof et al. 2003). This information can be recorded and used in offline analysis or for real-time interaction.

The most common system for capturing eye movements is the video-based corneal reflection eye-tracker (Duchowski 2003). The main advantage with this method is that it can be non-intrusive and does not necessarily require the user to wear anything. In video-based eye tracking, a camera is focusing on one or both eyes while the eye movements are being recorded. The light source reflection on the cornea (caused by infrared light) is measured relative to the location of the pupil's center. These two points are used as reference to compensate for head movements. This is the way it works for remotely installed light sources.

Before operating a video-based eye tracker, a calibration process is necessary to fine-tune it for each individual user (Poole and Ball 2005). A common calibration method is to measure gaze at predefined strategically positioned stimuli on screen, such as the corners of a grid (Duchowski 2003). Eye trackers normally produce a large amount of raw data since humans perform several saccades per second. A typical gaze data sample includes for each eye a 2D gaze point, the pupil's 2D location in the camera image, the distance of the eye from the camera, the pupil's size, a timestamp in milliseconds and a unique sample identification number (Tobii 2006). For more information regarding gaze data samples, please see the overview by Ramloll et al. (2004). The raw data needs to be filtered and reduced before it can be analyzed. In this process it is common to identify fixations and saccades (Rothkopf et al. 2004). Sometimes blinks are also identified as separate events. The identification of fixations is a complex problem and there is no unique method for filtering the raw data (Salvucci et al. 2000; Hansen and Ji 2010).

Eye-tracking is often used under the assumption that there is a strong correlation between the focus of gaze and the actual focus of visual attention. Indeed it is

possible to focus mentally on stimuli in the peripheral visual field, outside the foveal region. In this case, the internal visual-attention system (covert visual attention) is focused on a particular place, whereas eye-movements (overt visual attention) are directed to other places. For many applications, such as rendering 3D environments, the prediction of overt attention may be sufficient to perceptually optimize rendering of specific objects since regions outside the fovea are not perceived in high detail. In such cases the focus of attention may be estimated by a predictor algorithm, while eye-tracking is used only to infer the predictor in the first place and to evaluate the performance of predictor heuristics (Marmitt and Duchowski 2002; Peters and Itti 2008).

Jacob and Karn (2003) give a comprehensive overview of eye tracking in human-computer interaction research. This is a review of work regarding the application of eye movements to user interfaces both for analyzing them (usability measurement) and as a control mechanism (input). Jacob and Karn summarise a range of usability studies and discuss what users, tasks, and eye tracking metrics were used. Some mentioned eye tracking metrics include fixations, gaze duration, area of interest, scan path, etc. In addition to estimating the position of the foveal focus, various other features useful for analysis, such as fixation counts or amplitudes of saccades, can be extracted from eye-tracking data (Duchowski 2003).

Wooding and David (2002) introduced the concept of fixation maps as a means of quantifying eye-movement traces. Wooding also explored the concept of similarity between eye-movement patterns from different individuals and to which degree their fixations covered the image. Overlapping fixations are visualised using a three-dimensional surface plot, also referred to as a landscape or terrain based on the fact that the value of any point indicates the height or amount of property (discrimination/detection/perception) at that point. Wooding pointed out that the fixation duration can be taken into account by creating a dwell map, which also represents not only the areas fixated, but also the time these were fixated upon. Notably fixation duration, which is used in this chapter to weigh fixation counts, is suggested as a good indicator for estimating how strongly cognitive functions, such as object identification (De Graef et al. 1990), memory (Henderson et al. 1999) and monitoring of task-relevant objects (Land et al. 1999) are involved. The relationship between human gaze control and cognitive behavior in real-world scene perception is reviewed in (Henderson 2003).

There are various application areas, including computer graphics, virtual reality, and games, where saliency and task models have been used with varying degrees of success. In graphics for example, these models have been used to inform global illumination algorithms (Yee et al. 2001; Haber et al. 2001; Cater et al. 2003; Sundstedt et al. 2007). Luebke et al. (2000) and Murphy and Duchowski (2001) demonstrated that geometric detail in the periphery of the visual focus can be reduced without decreasing the perceived rendering quality by using an eye-tracker for gaze-contingent rendering optimizations. Komogortsev and Khan attempted to predict the visual focus of multiple eye-tracked viewers in order to perform perceptually optimized video and 3D stream compression (Komogortsev and Khan 2006). Gaze behavior was also studied when certain tasks had to be carried out. To analyze

gaze behavior in natural tasks, several studies were conducted with easy tasks ranging from handwashing to sandwich-making (Hayhoe et al. 2003; Canosa et al. 2003; Pelz and Canosa 2001).

There are different ways of analyzing eye tracking data stemming from computer game players. First, the game can be played and eye tracked in real-time, storing the relevant information for later analysis. The second option is to show pre-recorded videos from the game, but then the observer is only passively observing the game and does not interact with the application, which might affect the eye movements. Finally, eye tracking could be used to analyze screenshots/still images from a game. The first option is the most flexible since it allows for player interaction and a more natural gaming scenario.

Recent studies suggest that in adventure games, fixation behavior can follow both bottom-up and top-down processes (El-Nasr and Yan 2006). Visual stimuli are reported to be more relevant when located near objects that fit players' top-down visual search goals. In first-person shooter games, as opposed to adventure games, gaze tends to be more focused on the center of the screen (Kenny et al. 2005; El-Nasr and Yan 2006; Bernhard et al. 2010). In an experiment involving active video game play, nine low-level heuristics were compared to gaze behavior collected using eye tracking (Peters and Itti 2008). This study showed that these heuristics performed above chance, and that motion alone was the best predictor. This was followed by flicker and full saliency (color, intensity, orientation, flicker, and motion). Nonetheless, these results can be improved further by incorporating a measure of task relevance, which could be obtained by training a neural network on eye tracking data matched to specific image features (Peters and Itti 2008).

Starker and Bolt proposed using an eye-tracker to guide synthesis of speech in a way that narration refers to the current object of the user's interest (Starker et al. 1990). Although eye-tracking is used for real-time user-to-system feedback, their models of interest map gaze to objects, and successively the user's level of interest for each object is inferred. This resembles our methodology of inferring objects' importance by mapping eye-tracking data to semantic properties. In recent years, an increased number of eye-tracking experiments have been conducted using virtual environments or computer games (Rothkopf et al. 2007; Kenny et al. 2005; El-Nasr and Yan 2006; Jie et al. 2007; Sundstedt et al. 2008). These studies support the hypothesis that in conditions where a task has to be carried out, gaze behavior is mainly dominated by task relevance rather than salient features in the stimuli, as task-relevant objects are continuously monitored by the visual system (Land et al. 1999). Note that once a target is found and monitored during a task, the models for top-down control from visual search are no longer appropriate.

In the last few years, there has been an increasing amount of work done in the field of studying visual attention in games and using gaze to control games. Sundstedt (2010) and Isokoski et al. (2009) give more extensive overviews of visual attention studies in gaming and the use of eye tracking as an interaction device. El-Nasr and Yan, for example, studied the differences between players' eye movement patterns in two 3D video games (El-Nasr and Yan 2006), assessing whether the eye movement patterns in a game follow top-down or bottom-up processes. They found

that exploiting visual attention in games can help reduce frustration and increase engagement (El-Nasr and Yan 2006). Kenny et al. presented a study which investigated eye gaze data during a first-person shooter (FPS) game in order to find which information was more important in distributing interactive media algorithms (Kenny et al. 2005). Sennersten studied eye movements in an action game tutorial and was interested in how players direct their gaze, in particular what, where and when they fixate on specific objects (Sennersten 2004). Sennersten and Lindley (2009) used a real-time gaze object logging system to investigate visual attention in an FPS game.

McDonnell et al. studied a variety of humans in crowds to determine which parts of the characters people tend to observe most (McDonnell et al. 2009). Jie and Clark developed a 2D game in which the strategy and difficulty level was controlled based on the eye movements of the player (Jie et al. 2007). Hillaire et al. developed an algorithm to simulate depth-of-field blur for first-person navigation in virtual environments (Hillaire et al. 2008). Later, they also used a model of visual attention to improve gaze tracking systems in interactive 3D applications (Hillaire et al. 2010).

25.4 Understanding Playing Behavior Based on Eye Tracking

Understanding player behavior is important for both game designers and researchers, for instance in evaluating player experience. There exist several ways to analyze such behavior, each potentially revealing different aspects of the psychology involved in playing computer games. This chapter focusses exclusively on one such approach, namely the study of eye-tracking data obtained while participants are playing games (Sundstedt 2007, 2008; Stellmach 2007; Nacke et al. 2008; Bernhard et al. 2010; Bernhard et al. 2011).

Gaze analysis on the basis of eye-tracking data (Ramloll et al. 2004) yields fine-grained information regarding objects and events that are typically attended to in games (Sundstedt 2007; Stellmach 2009). We see this as a valuable tool that can be employed during the design cycle of novel games, as it can reveal where players are focussing their attention.

One of the earlier experiments in this realm maps fixation points to objects in a pseudo-3D game scenario, which is then used to answer the question as to whether the presence or absence of a task influences fixation behavior (Sundstedt 2007; Sundstedt et al. 2008). They record the full game state, enabling the game engine to later replay all actions, thereby facilitating the mapping of fixation points to potentially moving objects. Later, it was shown that this approach extends to full dynamic 3D scenes (Bernhard et al. 2010), confirming the utility of gaze-to-object mapping techniques.

The findings of Sennersten and Lindley provide further corroboration (Sennersten and Lindley 2008), showing that analyzing gaze in terms of Volumes of Interest (VOIs) or Objects of Interest (OOIs) provides insights that are difficult to obtain with screen-space techniques only. Sennersten and Lindley (2008) integrate the HiFi game engine with an eye tracker to map gaze coordinates to objects in a scene. As mentioned previously, Sennersten and Lindley (Sennersten and Lindley 2009)

used a gaze object logging system to investigate visual attention in game environments.

Stellmach et al. (2010c) discuss the trends and requirements for gaze visualization techniques. They describe a user study with experts in the field and outline which features are desirable for the visualization of gaze data. One of their suggestions is to aggregate different visualizations. Specifically, they describe three gaze visualization techniques for superimposing aggregated fixation data over 3D stimuli: (a) projected, (b) object-based, and (c) surface-based attentional maps. These are briefly elaborated here.

Projected attentional maps are 2D planar overview representations of 3D gaze data. They can be informed by a 2D Gaussian distribution in a manner similar to contour plots (Wooding and David 2002; Stellmach et al. 2010c). If the view changes, the projected attentional maps have to be recalculated. The performance does not depend on the size of the scene or the number of objects, and it is accelerated with less gaze data.

The *object-based* approach assigns a color value to each 3D object to describe its visual attractiveness (e.g. visual attention). The performance of this method is independent of the viewpoint and is only affected by object and gaze data quantities. This bears similarities to the aforementioned gaze-to-object mapping techniques.

The *surface-based* approach displays gaze data as 3D heat maps on the surfaces of the model. A gaze ray is mapped to the triangles of the mesh and a 3D Gaussian is used to splat gaze information across the mesh surface. Here, the mesh needs to be carefully chosen to obtain smooth attentional maps (Stellmach et al. 2010b). The surface-based attentional maps are the most time consuming to compute and performance is affected by the amount of gaze data and model complexity. Similar to the object-based approach, it is also independent of viewpoint modifications.

The main contribution of Stellmach et al. (2010b) is toward better visualization techniques for 3D stimuli via the three mentioned attentional maps, albeit without the goal of improving gaze-to-object mapping techniques. These 3D attentional maps aim to assist in visually better comprehending and inspecting the various aspects of gaze data (e.g. fixations duration, count, frequency). They may also be combined to provide visualizations at different levels of detail. The work was conducted with a static 3D scene and a dynamic viewpoint, but does not include dynamic objects.

Stellmach et al. (2010a) extend upon this work by experimenting with 3D scan path visualizations. They also introduce the models of interest (MOI) timeline, which can help to determine which object was viewed at a specific instant, thereby serving the same purpose as the recording of game states (Sundstedt 2007; Sundstedt et al. 2008). Additionally, they visualize the camera path with traces pointing at each gaze position.

Alternatives to the study of eye-tracking data are also in active development. We see these alternatives as complementary sources of input. For instance, in-game events may be logged for the purpose of analysing the nature and frequency of events occurring during game-play (Nacke et al. 2008; Nacke et al. 2011; Sasse 2008), although this approach does not take into account the physiological responses from the player. Nacke et al. (2011) present a logging and interaction framework

(LAIF) which enables those inexperienced with game design and programming to develop games and analyze them in a research environment. The work includes a user study based around a 2D gaze-interaction game which is playable with mouse input. Sasse (2008) gives an extensive overview of logging techniques for games as well as presents further information regarding the game used in the LAIF framework (Nacke et al. 2011).

Finally, questionnaires could be used to gather additional information, focussing on the emotional state of the player. Nacke et al. (2008) present a psychophysiological logging framework (Stellmach 2007) and discuss how the input from such a system can be synchronized with automatic scoring of game events. Questionnaires are useful for assessing the extent of spatial presence as well as gameplay experience (Nacke et al. 2011).

25.5 Overview of a Practical Pipeline to Measure Gaze Behavior in Games

Understanding what a game player is looking at during gameplay may help improve the design of a game. However, traditional tools and techniques, typically screen recording and playback, give very limited information. Matching fixation points to *pixel* data would allow us to understand game play in terms of low level features such as pixel color and contrast. To go beyond that, a number of different approaches that focus on mapping gaze to the underlying *objects* that give rise to the observed visuals have recently emerged (Sundstedt et al. 2008; Sennersten et al. 2008; Stellmach 2009). This mapping can be performed at different levels using various algorithms, as discussed in Sect. 25.6, but it does not reveal any information regarding the semantics of game play. In this section, we will describe the principles of designing a pipeline that enables not just correlating gaze to geometric objects, but also going a step further, allowing to map gaze to semantic objects, thus affording the opportunity to learn how users interact with games. An overview of the generic pipeline described in this section is shown in Fig. 25.3.

25.5.1 Adapting Games to Study Gaze Behaviour

To allow gaze data to be mapped to geometry, the game needs to be designed in a specific manner. When designing games, it is standard practice to use structured and systematic methods of naming, categorizing and grouping content. For example, in an FPS game, a category of “enemies” may be used to group together different classes of enemy types. An enemy class may be “soldier” or “aircraft”. Furthermore, multiple individual instances of enemies belonging to the same class are commonplace in a game (e.g. “soldier_20” or “aircraft_12”). Similarly, game content can be enriched with other useful properties such as object color or shape features (e.g.

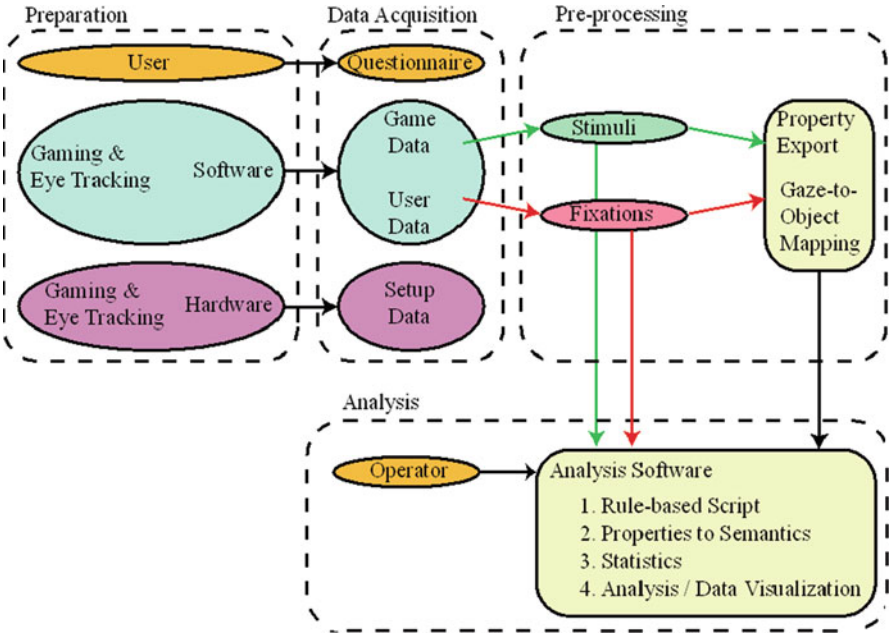


Fig. 25.3 Overview of the generic pipeline described

round, flat, etc.), status (e.g. a door may be open or closed, an enemy unit may be dead or "activated"). This wealth of information present in the game content itself provides a higher level description of the game and can be captured and processed to infer meaning of the user's gaze behavior later.

Further, games should be modified to provide game recording and playback functionality. Recording is responsible for capturing and storing the state and characteristics of the entire game for later offline use. To implement recording functionality, the game's scenegraph can be traversed and sampled at discrete time intervals. The best choice is to use the rendering loop of the game and capture the desired parameters whenever a new frame is rendered and dispatched to the screen. In turn, the playback functionality will then be able to use a recorded gaming session to load all the game state parameters necessary into the game engine and reconstruct the stimuli at a particular sample. As computing fixation points is an offline process, this functionality will allow us to determine which objects were attended to.

25.5.2 Preparation

Preparing for an eye-tracked gaming session is similar in principle to standard eye tracking studies (Sundstedt et al. 2009). However, games can be computationally demanding, thus requiring better hardware than that used when eye tracking simpler

applications such as web browsing or office applications. Most eye trackers can be operated remotely, allowing one workstation to be used exclusively for eye tracking, while a second computer is used for handling the game. Note however, that special attention must be paid when synchronizing the eye tracking data and the recorded stimuli, since clocks between two computers are unlikely to be in sync. Selecting a non-obtrusive (e.g. not worn by the player) eye tracker enables users to focus on the game itself, in contrast to head-mounted devices.

The higher the sampling rate an eye tracker can achieve the better, since fast eye movements can be more accurately recorded. It is wise to use an eye tracker that samples eye gaze at an equal or higher frequency than the display's refresh rate. This allows us to have more than one gaze sample per frame, making gaze-to-stimuli correlation more robust over temporal windows. Some commercial eye trackers currently offer sampling rates that exceed 100 Hz, while the majority of standard LCD displays operate at 60–75 Hz in native resolutions. Lighting conditions of the viewing environment should remain constant. Eye tracking data quality can be improved by the use of a chin rest, used to stabilize the head, however in most game-related studies the use of a chin rest is discouraged, because this alters the natural behavior of game players, which usually involves changes in body posture and head position in relation to the screen. Any instructions should be provided to the player a priori.

Finally, it is important to test the eye tracking hardware and be aware of the limitations it may have. For example, in some eye tracking devices, data quality degrades as the gaze moves away from the center of the screen and toward the corners. Also attention should be paid to participants wearing eye-correction glasses, occluding eye lashes and eye lids, lazy eyes, small and large pupils or pupils with low contrast. In addition, gaze behavior of participants may be affected if the setup does not resemble that of a natural gaming situation. When performing studies, care should be taken that participants may presume a certain task in computer games, even if none is provided, as this could alter their gaze patterns. Finally, participants should not partake in the same experiment more than once to avoid learning effects.

25.5.3 Data Acquisition

When studying gaze behavior in computer games, the data channels worth capturing are determined largely by the type of analysis to be performed later. Although there is no standard, there are four categories of data that one should consider recording:

1. *Setup data*: the characteristics of the environment and hardware used (e.g. eye tracker and screen), as well as its parameters (e.g. sampling rate and screen size), are important for later analysis. Parameters that belong to this category are static; that is, they remain the same throughout a study and across different subjects.
2. *User data*: in this category belong data referring to or produced by the user. This Calibration data can be both static (age and gender) and dynamic.

3. *Calibration data*: the latter includes calibration data, eye gaze over time (e.g. timestamp, position gazed in 2D screen coordinates, blinks, etc.) User input via the game's controlling interface, while technically acquired through the game engine, can also be conceptually classified as belonging to this category.
4. *Game data*: this category encompasses data produced by the game. In the past, screen recordings have been the primary game data acquired, however, as explained in this chapter, this is very restrictive. Instead, in gaming environments there is a wealth of information to our disposal, not only about the stimuli shown to the user, but also the parameters used to arrive at them, as well as temporal aspects and intrinsic parameters of the game's state. The range of data types available via the game's scenegraph is very wide, and game content can be further enriched by its designers to include properties and states. In most studies, the camera parameters, the game entities' parameters (e.g. position, orientation, color, textures, etc.) and game-generated events are the best candidate data types for capturing. Apart from dynamically changing data, games also have static data that can be recorded once, for instance the window size and position, which may differ from those of the screen, the hardware it runs on, etc.

The purpose of acquiring these data is to be able to reliably and accurately reconstruct the stimuli that affected the user's gaze behavior with the goal to study it. This decoupling of data acquisition and data analysis provides an ideal methodological partitioning that allows data reuse. The data captured from all these categories has very low storage requirements even for several minutes of data acquisition. Notably, game data consists only of parameters that allow the reconstruction of the game state at any given time, without the need of capturing thousands of images, as is the case for a screen recording.

Finally, it would be advantageous to debrief participants by means of a questionnaire. This could serve two purposes. First, a well-designed questionnaire would make it evident whether the participant understood the task that was being performed. Testing this can be important, because participants that have either misinterpreted the instructions or have second-guessed the purpose of the experiment may yield unreliable or biased data. In essence, if an outlier is detected by analysis of the data, then the questionnaire may help explain why this has occurred, providing the justification for outlier removal.

Second, the questionnaire could contain questions that query the participant regarding their response to the experiment. For instance, it would be possible to ask how difficult the different conditions were to the participant, or to what extent the task was enjoyable. Dependent on the primary aim of the experiment, answers to such questions may corroborate the data found in the main experiment. Nacke et al. (2009) studied navigation using gaze as input in a 3D first person shooter game. The purpose of the study was to investigate the gameplay experience using gaze interaction by the use of subjective questionnaires. Three questionnaires were used based on previous work which evaluated the self-reported game experience, flow, and presence.

25.5.4 Reconstruction and Pre-processing

The next step in this pipeline deals with the reconstruction of the stimuli and pre-processing of the recorded data for further analysis. With the recorded data in hand, a playback-like simulation of the game is performed by reconstructing all states the game has gone through during the gaming session on a frame-by-frame basis. The data is processed off-line without any performance constraints. Several tasks are carried out:

- **Fixation detection**. Raw gaze data captured by the eye tracking hardware are fuzzy and should not be directly used to infer a subject's gaze behavior. Instead, raw gaze data is processed using fixation detection algorithms that cluster raw gaze into fixations for subsequent use (Duchowski 2003).
- **Gaze-to-object mapping**. Gaze is correlated with objects. Here, detected fixations are used to map gaze data back to scene objects, using a so-called gaze-to-object mapping algorithm (see Sect. 25.6). In this process, each fixation is in turn correlated to one or more objects which are potentially the targets of that fixation.
- **Property exporting**. Game entities carry properties assigned to them at design time. These may be static or change in the course of the gaming session. They should be linked to the objects to determine the semantics of each object.

The output of the pre-processing can be stored in a single file (e.g. in XML format) comprising an entry for each fixation. Each fixation entry contains the ID of the fixation target object(s) and a sequence of frame entries, encoding the states of the stimuli within the duration of a fixation. Each frame entry comprises a set of visible objects, a set of audible sounds, a set of user events and further attributes reflecting those properties of the game's context which may be relevant for understanding the behavior of the player. The entry corresponding to each object, sound event etc., comprise an identification code (ID) and a set of properties which characterize meaning and state of the respective entity. Correct alignment in the time domain between the recorded gaze points and game-states is achieved by comparing time-stamps of both sequences.

25.5.5 Analysis Tools

The final step of the pipeline is the analysis of gaze data, for which we describe our approach here. We first perform an explorative analysis by means of visual interpretation of gaze density histograms. The main analysis will then map gaze data to objects with a set of pre-defined semantic properties. The user has many degrees of freedom in the definition of the semantic properties of interest, a task aided by a user interface that allows not only the selection of semantic properties, but also enables the definition of clusters of semantic properties (e.g. assigning objects of a similar category to one super-class) or to define new semantic properties which depend on

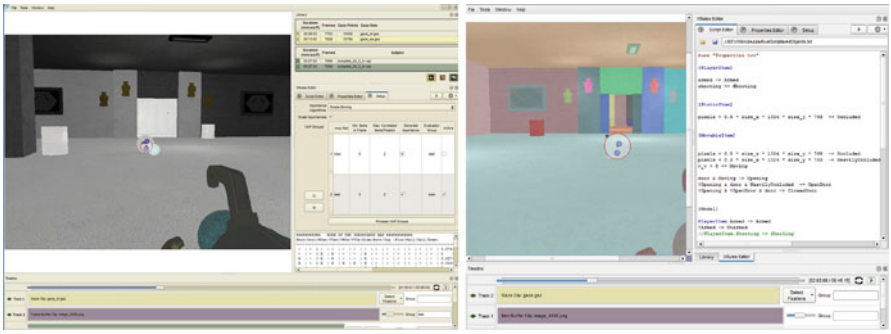


Fig. 25.4 Screenshots of the analysis software toolbox of the experimental pipeline

dynamic events in the scene (e.g. an enemy is destroyed after a successful shot). To this end, a scripting language is used in the analysis software, where rules can be defined to describe how certain properties should be interpreted prior to building gaze histograms (see Sect. 25.7).

The Graphical software tools we designed for studying gaze behavior are similar in look-and-feel to video editing suites and includes the following components:

- **Library**. A library widget designed as a front-end to gaze and stimuli data which the user can load from disk. This is useful so that the operator can select collectively which stimuli data and eye tracked subjects are relevant to his current analysis. The library holds pointers to data, but need not load the data.
- **Timeline**. A timeline widget with different stacked tracks allows the operator to instantiate stimuli and gaze data so that they can be played back in parallel. The timeline offers typical controls of temporal data (e.g. start, stop, etc.) and allows for seeking arbitrary frames within the datasets, enabling intuitive non-sequential access to them.
- **Views**. To visualize the data, viewing widgets use the timeline tracks to sequentially overlay visual representations of the respective data at the time the timeline's head is positioned or a temporal window around it. For example, fixations can be easily overlaid and played back over the stimulus that produced it.
- **Script editor**. A scripting editor enables an operator to define, execute and debug scripts that transform low level extracted game entity properties into semantic properties.

The combination of these tools into a single graphical user interface enables an operator of the analysis software to potentially gain insight and assist him in scripting rules for transforming properties to semantics. This graphical user interface, shown in Fig. 25.4, is effectively an Integrated Development Environment (IDE) for studying gaze behavior that not only offers the tools to setup and perform an analysis task, but also provides visual feedback and can potentially leverage the experience and intuition of the operator.

The following section describes the various algorithms required to implement such a pipeline, including those that enable the analysis of data.

25.6 Object-Based Gaze Analysis Algorithms

The input assumed for the processing pipeline is a gaze data set from the eye tracker and a reconstruction of the game states. The goal is now to process gaze data and obtain gaze statistics scoring the amount of attention deployed to particular objects, object categories or semantic properties. To achieve this goal, two basic steps are required: (a) gaze has to be *mapped* to objects, which can then be further abstracted by their category, other properties or their meaning to the user, and (b) a method has to be defined to build gaze *statistics* with respect to the independent variables we are interested in (e.g. object IDs, properties or meaning; see Sect. 25.7). Initial solutions were prototyped in the work of Sundstedt et al. (2008) and Bernhard et al. (2010). In the following sections, the ideas behind these approaches are presented.

First, an important step of the proposed methodology is to map gaze data to objects. This is done by a gaze-to-object mapping algorithm which specifies the potential target(s) of each fixation. Fixation targets are individual objects which are represented by an identification number (ID). In some cases, it might be interesting to quantify how often an individual object was attended, but for realistic game levels we have to assume that each player has a unique game experience when navigating a spatially large environment containing many objects. Under these circumstances, gaze is distributed very sparsely and is not suited for a statistical analysis. Therefore, rather than focusing on particular object instances (e.g. "AlienMonster_57") it is more promising to compute gaze statistics for object categories or semantics.

Overall, gaze analysis can be performed at different levels of abstraction. We distinguish four layers in which the stimulus can be represented in the analysis:

- **Screen space**: Gaze points in 2D (e.g., position = [0.1,0.5])
- **Object space**: Object instances (e.g., ID = 2,933)
- **Property space**: An object's category, state and behavior (e.g. category = "Alien Monster", distance = 5 m, behavior = "approaching", avatar health state = 10%, etc.)
- **Semantics**: An object's meaning to the user according to game task (e.g. "attacker", "close", "dangerous", "high risk")

In Fig. 25.5, we illustrated the levels of abstraction with an example of a game where a user has to move a pedestrian across the street: In the first abstraction layer (top), we see pixels as seen by the player of the game. The next abstraction is the object level, where we have particular instances, such as cars and trees, with unique IDs. In the third layer, individual objects are abstracted in terms of their properties including the object category (e.g., "car") and spatial properties (e.g. velocity or position). In the semantic layer (bottom), the scene is abstracted according to the meaning of the objects to the user and the task at hand. In this example, the user has to move the avatar across the street, the avatar hence becoming a pedestrian. For a pedestrian, objects which are most task relevant are the oncoming car and the car currently passing, whereas the car which has already passed by is not important. On the other hand, details of objects behind the street (e.g., houses, trees and sky) are of low relevance and can be abstracted as background.

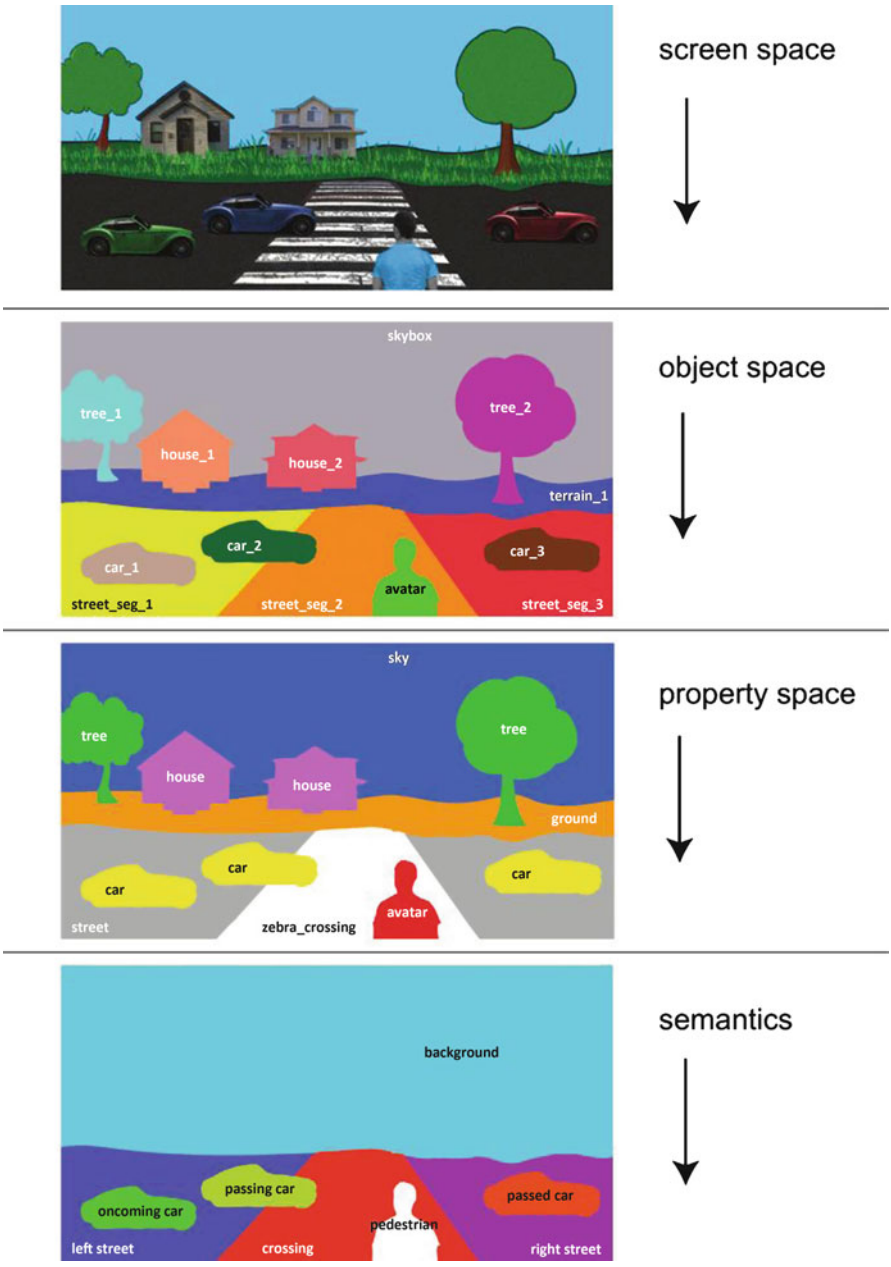


Fig. 25.5 Example for layers of abstraction in a pedestrian road crossing task

Most commercial gaze analysis tools operate only with screen space data which is readily available: the images rendered to the screen and the gaze data which is output by the eye tracker in screen space. But for computer games, we can

fortunately assume that an object space representation is available, which can be obtained when the internals of the game engine can be accessed.

However, for gaze analysis in object space we also need a representation of gaze data in object space (e.g., the ID of a fixated object). This can be obtained by mapping gaze to objects, as described in Sect. 25.6.1. Some additional modifications of the game engine are also needed to derive properties from object space. These will be discussed in Sect. 25.6.2. Semantics are then derived from the properties of a scene. Since inferring semantics from properties is a cognitive process, this requires the assistance of a human operator who provides an ontology which defines the mapping from properties to semantics (Sect. 25.6.3).

25.6.1 Mapping Gaze to Object Space

A common way to pre-process gaze data is to filter for fixations, since a user's attention correlates with fixation locations only and not the saccades in between (Duchowski 2003). Thus, the input of gaze-to-object mapping is assumed to be fixations and the reconstructed states of the game during the start and end time of each fixation.

Note that the position of a fixation is fuzzy, as it corresponds to a cluster of jittered gaze points sampled by the eye tracker during the time the fixation occurred. On the other hand, we have objects of a 3D scene which are rendered to a 2D image by a perspective projection from the camera viewpoint. Mapping fixations to objects is therefore done by computing the degree of intersection between fixations and scene objects. Different ways to achieve this have been proposed, reaching from straight-forward solutions to more sophisticated methods. The methods along with their advantages and disadvantages are briefly described in the following (see also Fig. 25.6).

25.6.1.1 Point-Based Methods

The simplest way to map fixations back to objects in a scene is to use the center of a fixation (e.g. the mean position of raw gaze samples). With this simplification, one has to find only the object which was projected to one pixel position. This can be carried out directly in object space by casting a ray through the fixation center into the scene and computing the nearest object intersected by that ray.

Another solution would be to solve the problem in screen space. Each scene object is rendered using a unique color ID and stored in an image buffer, referred to in the literature as an *item buffer* (Sundstedt et al. 2008) or *id buffer* (Saito and Takahashi 1990). This operation can be performed directly on the GPU with minimal computational effort. The color of the pixel which corresponds to the fixation position is then queried in the item buffer and decoded to obtain the respective object ID.

Using only one point to map a fixation to an object is a reasonably simple and efficient solution, which may be advantageous for real-time applications. It should

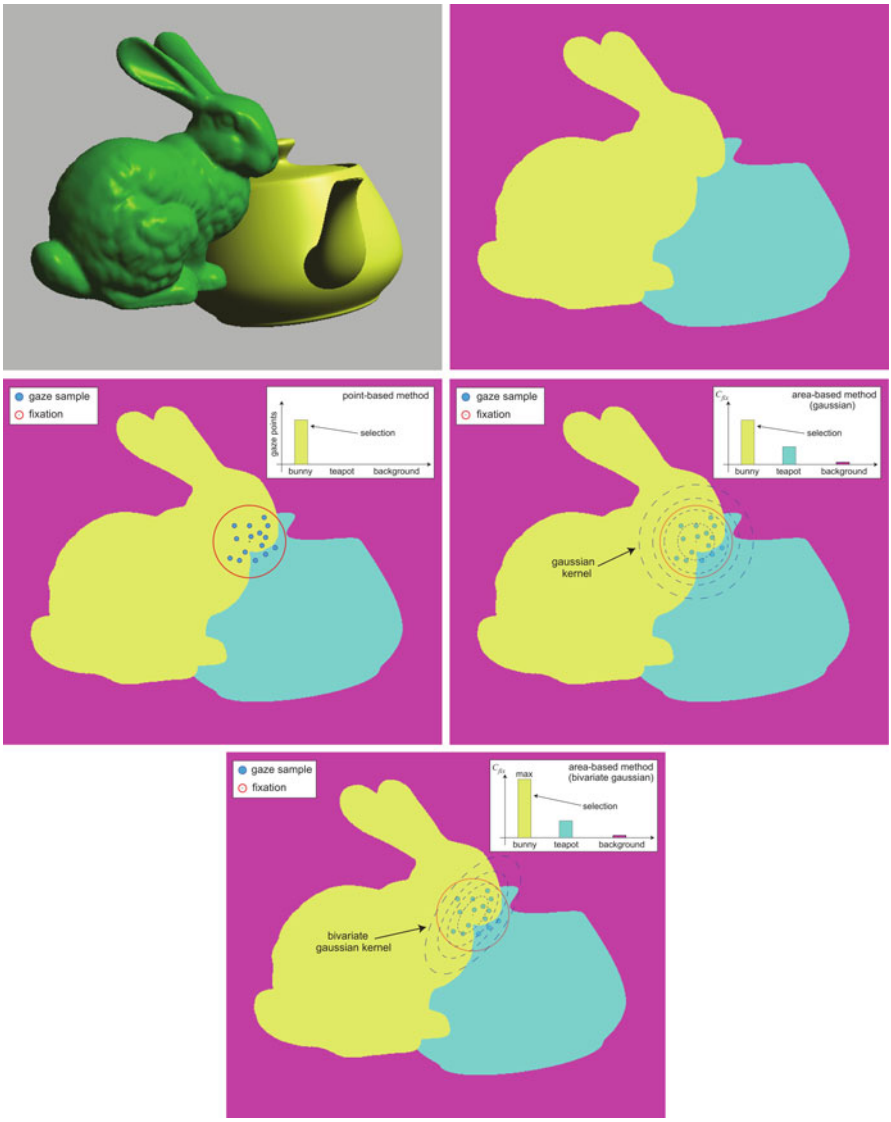


Fig. 25.6 Gaze to object mapping methods: to identify the object underneath each pixel, the scene (*top left*) is rendered to an item buffer (*top right*). Fixations can be mapped to objects by simply picking the pixel at the center of a fixation (*middle left*) or integrating the energy spread by a 2D Gaussian kernel (illustrated with *dashed rings*), which models the foveal acuity (Eriksen and St James 1986) (*middle right*) or the distribution of gaze points (Bernhard et al. 2011) (*bottom*), over the area of the respective objects

work well as long the scene is simple, that is if there are only few, relatively big and well separated objects. But in many cases, the fixation center might not necessarily intersect the object a user is actually attending.

There are two factors that may play an important role when mapping gaze to objects: first, the fovea resolves all objects in sharp resolution in a frustum of about 2° of visual angle (Palmer 1999), and second, a fixation is a cluster of a fuzzy cloud of gaze-points distributed over space and time being sampled in discrete time intervals from an actually continuous motion path of two eyes. It is, therefore, not always appropriate to evaluate the intersection of a fixation and scene objects at a single point. Rather, it may be advantageous to account for the region spanning the potential focus of attention, thus leading to area-based methods.

25.6.1.2 Area-Based Methods

An area-based approach was first proposed by Sundstedt et al. (2008). They render the scene into an item buffer and intersect a kernel K_{fix} with the objects contained in the buffer. The kernel is centered at the fixation's mean position $(\overline{x}_{fix}, \overline{y}_{fix})$ and for each visible object o, an integral is computed which accumulates the energy contributed by that kernel over the area $A(o,t)$ covered by this object in the item buffer at time t. To account for possible changes in the item buffer, integration is also done over time between start time stamp $t_{s,fix}$ and end time stamp $t_{e,fix}$. We define this integral as the correlation $C_{fix}(o)$ between a fixation fix and an object o:

$$C_{fix}(o) = \int_{t_{s,fix}}^{t_{e,fix}} \int_{(x,y)\in A(o,t)} K_{fix}(x-\overline{x}_{fix}, y-\overline{y}_{fix}) \mathrm{d}(x,y) \mathrm{d}t \tag{25.1}$$

Note that this implies that the fixation duration t_{fix} is given by $t_{e,fix} - t_{s,fix}$. Two variants of the kernel have been proposed so far: Sundstedt et al. (2008) proposed a kernel which simulates the visual acuity of the human retina, whereas Bernhard et al. (2010) proposed to use a kernel to model the distribution of the gaze points corresponding to the fixation.

25.6.1.3 Foveal Sensor Density Model

Sundstedt's foveal sensor density model begins by approximating the fall-off of spatial acuity from the fovea to the periphery using a normal distribution N (Sundstedt et al. 2008):

$$K_{fix}(\Delta x, \Delta y) = N(\sqrt{\Delta x^2 + \Delta y^2}) \tag{25.2}$$

The Euclidean distance $\sqrt{\Delta x^2 + \Delta y^2}$ between any pixel position and the center of a fixation point is inserted into the univariate Gaussian density distribution:

$$N(\sqrt{\Delta x^2 + \Delta y^2}) = exp\left(-\frac{\sqrt{\Delta x^2 + \Delta y^2}}{2\sigma}\right) \tag{25.3}$$

Taking into account that the foveal region of human vision spans approximately 2° of visual angle, the area over which a fixation point bears relevance corresponds to a circle which is determined by an intersection of the cone of foveal vision and the screen plane. The size of this circle, which we denote as σ_{fovea}, is then used to define the standard deviation of the Gaussian kernel. Assuming that the distance between eyes and the display is d, we can compute σ_{fovea} :

$$\sigma_{fovea} = d\tan(\alpha) \tag{25.4}$$

Note there is a subtle but reasonable simplification in the computation σ_{fovea} as the intersection of the foveal cone and the screen plane actually depends on the eye's viewing angle and would vary with gaze position if computed accurately. However, it is more important to account for the limited accuracy of the eye-tracker. Thus, we add to this the eye-tracker error σ_{error} :

$$\sigma = \sqrt{\sigma_{fovea}^2 + \sigma_{error}^2} \tag{25.5}$$

Using Eq. (25.1), we compute a weight C_{fix} for each object, which serves as an estimate for the likelihood for it to be attended by the user. This model assumes that the a priori probability for a fixation increases with the number of pixels covered by the object. Hence, if the size of scene objects varies too much, this may result in a bias toward large objects, as the amount of attention received does not necessarily correlate linearly with size. To control large variations in size, it may be necessary to perform a subdivision of objects (as done in Sundstedt et al. 2008).

Another assumption of this approach is that several objects may be attended during one fixation, as the output is a value for each object scoring its potential attentional relevance in the current fixation. This corresponds to spatial models for attention such as the spotlight (LaBerge 1983) or zoom-lens models (Eriksen and St James 1986; Castiello and Umiltà 1990), which assume that attention is enhanced for all objects within the focus region. However, some experimental results suggest that human cognition is better at attending only one object at a time (Duncan 1984; Baylis and Driver 1993; Behrmann et al. 1998). Especially, during execution of a task, unexpected objects or events may go unnoticed even if they appear within the foveal focus of a viewer (Simons and Chabris 1999).

Under these assumptions, it is not necessarily appropriate for a single fixation to compute an attention weight for several objects. Hence, Bernhard et al. proposed to assume that during a fixation, attention is focused on one object only and may not be directly related to the foveal sensor density (Bernhard et al. 2010). Instead of approximating foveal sensor density distributions, they account for the fact that a fixation is made up of a cluster of spatially distributed gaze points, as discussed next.

25.6.1.4 Gaze-Point Distribution Model

As the eye-tracker has limited precision, and the human oculomotor system cannot hold gaze stable on a fixed position, we have to account for the fact that the gaze

points, which are sampled at discrete points in time, are distributed within a known uncertainty region (Bernhard et al. 2010). The density distribution of the continuous gaze paths during a fixation can be approximated, for instance with a bivariate Gaussian kernel. The parameters of the kernel are derived from the constant uncertainty of the eye-tracker and the spatial distribution of gaze points clustered within the fixation. A bivariate kernel provides a better fit to unidirectional drifts of gaze, which were frequently observed in the gaze data:

$$K_{fix}(\Delta x, \Delta y) = exp\left(-\frac{1}{2\left(1-(\rho_{fix})^2\right)} \left(\frac{\Delta x^2}{(\sigma^x_{fix})^2} + \frac{\Delta y^2}{(\sigma^y_{fix})^2} - \frac{2\rho_{fix}\Delta x \Delta y}{\sigma^x_{fix}\sigma^y_{fix}} \right) \right) \quad (25.6)$$

In this case, the parameters of the kernel depend on the distribution of gaze-points clustered with the current fixation *fix*. The parameters σ^x_{fix} and σ^y_{fix} denote the standard deviations of the fixation's uncertainty region in both dimensions, and ρ is their correlation in (x, y). After evaluating Eq. (25.1) with this kernel for each object, the object which is most likely the target of the fixation is determined by selecting the object with the maximum value correlation weight C_{fix}:

$$o_{fix} = arg\,\max_{o} \quad C_{fix}(o) \quad (25.7)$$

It should be noted that current gaze-to-object mapping techniques are still in a premature state and their accuracy could be improved considerably, as discussed in the following section.

25.6.1.5 Limitations

First of all, the most important concern is that though these techniques provide reasonable results in proof-of-concept studies, their accuracy has not been evaluated yet. Unfortunately, evaluating accuracy for general scenes is a difficult problem, as it requires us to compare the result of the gaze-to-object mapping algorithm with the actual focus of a user. Such a comparison would require an experiment which uses other methods than eye-tracking to reliably determine which object is attended by the user.

We expect that the current algorithms fail particularly in situations where objects or the camera are moving fast. The algorithm's accuracy is also limited when objects are relatively small, consist of thin parts, are placed very close to each other or even occlude each other partially. Another problem arises when a user tends to scan the silhouette of an object, as this provides more information about its shape. In this case, unattended objects in the background may be incorrectly marked as fixated upon.

In current methods, fixations are treated as static. To account for fast motion in the scene or a moving view port of the camera, algorithms may need to incorporate the temporal dimension in the distribution of gaze samples clustered in one fixation. Therefore, appropriate gaze-to-object mapping methods should be developed for smooth pursuits, which are drifting fixations occurring when the eyes track a moving object.

25.6.2 From Object-Space to Properties

The so-called property space describes the properties of a scene or even the entire state of the current application. Ideally, such a description is generated for the entire scene, or at least for all visible objects. The reason why it is useful to consider, apart from the fixated object, other objects in the scene is that we should account for the context under which a fixation occur red. If the scene and the viewpoint changes, this is important to identify or note, because we need to track how many other objects could be concurring alternative targets for the fixations being issued.

There are several properties which could be of interest. Overall those can be divided into properties of objects and properties reflecting the current behavior of the user and the avatar being controlled.

25.6.2.1 Object Properties

- **Visibility**: all objects which were visible in the camera's field-of-view had a potential influence on a users's behavior and could be potential fixation targets. Visibility can be determined directly from the item buffer, as only visible objects may cover any pixels.
- **Object category**: the most important property is the category of an object, which allows us to link an object to semantics. As we reasonably assume that the category of an object is a static property, we just need a look-up-table where each object ID is mapped to the respective category of an object.
- **Spatial properties**: spatial properties, like size, motion or position in the screen space could also be of some interest in the analysis.

25.6.2.2 Player Related Properties

- **Game/player state**: the current state of the game and the player may also affect user behavior. For instance, a low health state could cause the player to focus on searching health items.
- **Interaction**: it might also be interesting to analyze gaze behavior with respect to the way the user is interacting with the application. Hence it is useful to include the actions of the avatar (e.g., "running" or "shooting") or input events from mouse, keyboard or joypad.

25.6.2.3 Logging Tool

To extract scene properties, a light-weight interface to the game-engine is defined, which is used to notify a logging tool about changes in the game's internal parameters, such as the view-matrix of the player camera or variables of scene entities (e.g., objects and other relevant of the game state). Having access to those very basic

parameters, the logging tool then computes properties which might be useful for the analysis or the inference of semantics. For example, the tool infers screen space positions, bounding windows or motion vectors from camera parameters and object world-space bounding boxes.

25.6.2.4 Log Format

Though many of the properties are static throughout the game (e.g., category), we have to assume changes in many other properties (e.g., visibility or user input events), which hence have to be logged for each frame. If only fixations are considered in the analysis, it is useful to define a format where for each fixation the fixated object and a description for all frames between the begin and end time of that fixation are logged. For each frame description, one should log the time-stamp, the IDs of the visible objects, their dynamic properties and a description of the current game state and user input events.

In XML an example for the log format could look like this:

```
<fixation>
  <duration>0.532</duration>
  <fixated object>
    <id>12423</id>
    <confidence>0.9</confidence>
  </fixated object>
  <frame>
    <timestamp>54.334</timestamp>
    <object>
       <id>12423</id>
       <visibility>1.0</visibility>
       <category>Tree</category>
       <screen_bounding_window>
         <min_x>0.1 </min_x>
         <min_y>0.6 </min_y>
         …
       </screen_bounding_window>
       …
    </object>
    <object>
    …
    </object>
    …
    <player>
       <health_state>0.7</health_state>
       <action>"running"</action>
       …
    </player>
```

```
    </frame>
    …
  </fixation>
```

25.6.3 From Properties to Semantics

Having a full description of the stimulus for each frame, it is now possible to analyze various aspects of human behavior, for instance by focussing on specific semantically meaningful object properties. Assuming top-down attention is mainly influenced by high-level processes, the most appropriate way to represent the stimulus is a description accounting for the meaning of objects according to the current task a user is performing. However, inferring meaning from object properties is a complex problem and requires introducing knowledge into the analysis pipeline. At this stage, the user of the analysis software has to specify a set of rules on how raw object properties should be translated into meaning. Therefore, Bernhard et al. proposed a simple scripting interface, which is integrated into the user interface of the analysis software.

A user may write rules defining relations, such as for instance "palmtree **is a** tree" or conditional statements, such as "**if** car.position.x < center.x **and** car.motion.x > 0 **then** car **is** approaching." The transformation from object properties to semantics is then carried out by an interpretation unit which applies the rules specified by the user.

25.6.3.1 Keeping Degrees of Freedom Low

To avoid problems of sample size, it is important to keep the degrees of freedom low, i.e. avoid many dimensions and use a small number of semantic properties in the analysis. If there are too many semantic categories, one can reduce this number by clustering similar categories or defining semantic super-classes. The degrees of freedom can be reduced in an additional selection pass proposed in Bernhard et al.'s work (Bernhard et al. 2010). This selection pass is specified by the user of the analysis software and projects the output of the semantic transformation to those values in which the user is interested most.

25.7 Collecting Fixation Statistics

Let us assume that we have mapped each object to one semantic property to be further denoted as x. Of course, it is possible that an object has more than one semantic property, but, for simplicity, we will only assume a single property case (see Bernhard et al. 2010 for a multidimensional example). The next step is to derive a

statistic which scores the amount of attention given to each semantic property as an importance value. This statistic will be denoted as importance map $I(\mathrm{x})$, which links each semantic property x to an importance value. The importance ideally corresponds to the probability that an object holding x is fixated by the user.

To derive the importance map, Sundstedt et al. (2008) proposed to accumulate fixation times for each semantic property. However, in their study the viewpoint was fixed and the set of observable objects remained constant. For the general case, we have to assume a viewpoint which is not fixed and the set of objects in the camera's field-of-view may vary considerably from one frame to another. Thus, Bernhard et al. (2010) proposed a heuristic normalization strategy accounting for the different amounts of time certain objects are visible to the user.

To calculate $I(\mathrm{x})$, the time t_{fix} that objects with that x were fixated is first accumulated and than normalized by the accumulated time t_{vis} that objects with that x were visible during a fixation (i.e. the number of frames they were potential fixation targets):

$$I(\mathrm{x}) = \frac{t_{fix}(\mathrm{x})}{t_{vis}(\mathrm{x})} \tag{25.8}$$

The normalization factor $t_{vis}(\mathrm{x})$ corrects for variations in the visibility of different semantic properties. If an object is visible in many frames but it is fixated only in a few of them, the importance value should be low, and if an object is fixated most of the time it is visible, the importance value should be high. The maximum importance value is 1 and occurs if a semantic property is fixated in every frame it is visible.

This model takes into account changes due to the visibility of objects. However, generalizing the solution to contextual dependencies is a difficult problem, as it would increase the dimensionality of the statistic to the size of all possible combinations of semantic properties, and a sufficient density of gaze samples would be difficult to acquire.

25.7.1 Limitations

Practically, it is not possible to define a normalization strategy which perfectly corrects for all latent effects resulting by the variation of the viewpoint and changes in the scene. Hence, this heuristic involves many simplifications, such as the assumption that each semantic property is perceived as one unit of attention. Defining the units of attention, which make up the number of alternative targets a user can fixate in a given view of the scene is a hard problem. For future work, it could be useful to investigate strategies which are inspired by models for pre-attentive object detection from vision research. Those could potentially allow to better quantify the amount of visual information a user perceives within the field-of-view.

Another important factor arising from the uniqueness of the game experience of each user is the variation of the contexts in which a particular object may be seen, which may also significantly influence attention. One strategy to reduce the varia-

tion is to divide large game levels into sections where a similar context can be expected and perform the analysis for those sections separately.

25.8 Results

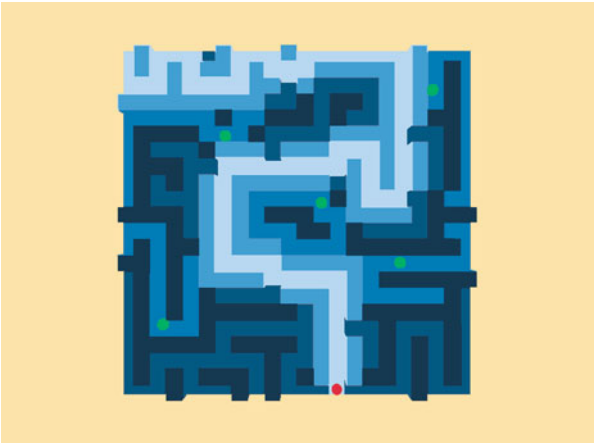
The pipeline described in this chapter is a flexible framework that offers the technical means of setting up an experiment to study gaze behavior in a game, discusses what data may be useful to capture, how to achieve this, and how to analyze them. The pipeline has been presented in a manner that can be adapted to various studies to enable researchers and practitioners to tailor it to their own needs. We will describe two sample experiments here to offer first hand examples of how the pipeline and algorithms have been put into use by the authors.

25.8.1 Example 1

The first example consists of an eye-tracking experiment that was carried out to generate an importance map based on high-level properties in a computer game (Sundstedt et al. 2008). The task of the game was to navigate a small ball through a maze, which was in 3D, but rendered from a fixed bird's-eye view. All items in the maze were tagged with high-level properties, such as the correct and incorrect path, to encode the relevance of certain parts of the maze in relation to the task of finding the exit of the maze. These items were also referred to as *object classes* and can be seen in Fig. 25.7 along with a more detailed description. Accumulating fixations over different object classes provides a fruitful approach in understanding where game players focus their attention. Such information cannot currently be extracted from an analysis of low-level salient features alone.

The analysis process depends on three main steps. In the first pass, the player plays the game while being recorded using an eye tracker. All game states are logged so that it is possible to reconstruct each frame later. The novelty of this approach is that it also allows playback of the game in real-time, which can be used for another condition or another group of players watching the same game stimuli passively, for example. After the first pass, each frame can be reconstructed and the additional data, such as the item buffer, frame buffer and object data, can be generated. Fixations can then be mapped to object types using the item buffer in order to find out which of them are the most significant with respect to the gameplay. The item buffer for the maze can be seen in Fig. 25.8. Finally the analysis tool can be used to get useful information from the stored game and gaze data to generate an importance for each object class. The distribution of fixations directly relates to the importance each object type carries for executing the game's tasks.

The area-based approach (Sect. 25.6.1.2) is used, which renders the scene into an item buffer and convolves with a kernel simulating the visual acuity of the human retina, with the objects contained in the buffer. The Foveal Sensor Density Model is used to map fixation points back to semantic object classes in the game. After



Object Class	Description
Correct path	The floor and walls of the path that must be traversed to go from the starting point to the end point.
Incorrect path	The floors and walls of dead ends.
Adj. to correct	The top surface of the walls that are adjacent to the correct path.
Top surface	The top surface of the walls that are not adjacent to the correct path.
Closed paths	These are the parts of the maze that are separate from the main paths and may contain distracting elements.
Main walls	The four walls enclosing the playing field.
User-contr. ball	The ball under the participant's control.
Distractor balls	Balls not under participant control.

Fig. 25.7 Maze Object Classes: the image shows the different object classes used in the experiment (Eriksen and St James 1986) as well as their configuration. Below this, the description of the object classes

classification, each participant produces a normalized distribution of fixations per object class. This set of distributions is then subjected to further analysis using traditional statistical tests, such as a one-way ANOVA (Cunningham and Wallraven 2011), to reveal statistical differences in fixation behavior of participants in different conditions (for example actively playing a game versus passively watching a game) (Sundstedt 2008).

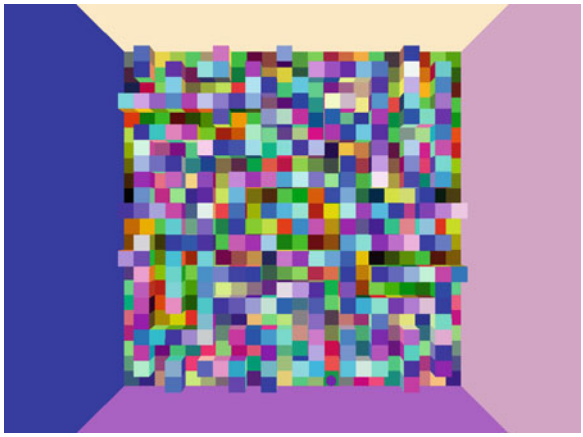


Fig. 25.8 Maze Item Buffer: showing all *color coded* objects which enable us to relate fixation points to objects and object classes

To check the validity of the experimental design, fixation distributions are matched to projected object sizes. The hypothesis is that if fixations are randomly distributed, they would fall on large objects more often than on smaller objects. It was found, however, that the fixation distributions are markedly different from the distribution one would obtain by counting the number of pixels that are covered by each object type. This indicates that none of the results can be explained by random fixation behavior.

The experimental design allows a comparison between fixation behavior while the game is played against fixation behavior while observing a recording of a previously played game. The hypothesis is that passive viewing would lead to different behavior than active gameplay. Further, if this is the case then the concept of saliency could be applied to predicting gaze behavior in the absence of a task, while simple saliency measures would not predict fixation behavior in the presence of a task.

However, this experiment led to a surprising result in that even passive user behavior is task dominated and cannot be statistically distinguished from active gameplay behavior. In this particular game design, saliency is therefore a very poor predictor for task relevance. This observation may extend to other game designs. However, it should be noted that in this study, the camera was locked so that each user had access to similar visual content at all times. This improves the rigor of the experimental design, leading to better control of the experimental set-up, and thereby fewer risks of introducing bias.

On the other hand, this study reduces the problem of inferring gaze distributions to a very limited case by assuming a fixed camera and a constant set of objects. In the second example, the approach is generalized to a representative 3D scenario with a dynamic viewpoint and a field-of-view with variable content.

25.8.2 Example 2

The second example is from Bernhard et al. (2010), who implemented an early prototype of the entire pipeline described in this chapter. A 3D First Person Shooter game was used to perform a proof-of-concept study of their system.

Fig. 25.9 Predicting visual attention from fixation statistics: (**a**) An example reconstructed framebuffer of the game is overlaid with the visualization of a fixation in the current frame. In Figure (**b**), fixation statistics were used to predict the importance for each object in the scene which is visualized by the brightness of the objects (brighter objects are more important than darker ones). Figure (**c**) shows the corresponding saliency map. Since fixation statistics account for semantics, they can predict better the high importance of doors or objects in the center, while saliency maps are less selective and predict the importance of pixels rather than objects

The actual goal of this work was to derive a gaze prediction heuristic which is learned from gaze data recorded from several participants of an eye-tracking study. Learning is regarded as the process of inferring an importance map (essentially a statistical model of the data) by utilizing the gaze analysis pipeline. The importance map is then used to estimate the likelihood for each object to be attended in a particular frame. Figure 25.9 shows an example frame of the FPS game, the corresponding importance map learned and the respective saliency map for comparison.

The following sections will give a very brief description of this work. For readers who prefer a more detailed description, we recommend to read the original article by Bernhard et al. (2010).

25.8.2.1 Inferring Importance Maps

The pipeline can be adapted, as shown in Fig. 25.10, to enable deriving importance maps for gaze prediction, which is structurally similar to the methodology described in Sects. 25.6 and 25.7. The input of this pipeline is a gaze file and a replay file recorded during an eye-tracking study. This information is then used twofold. First, an abstraction of the stimulus in terms of high-level semantic properties is derived. Second, the object which was fixated in that stimulus can be determined. The information as to which object is fixated and the abstraction of the corresponding stimulus then forms the input of an algorithm which learns an importance map. A straightforward estimate of the importance map can be obtained by accumulating fixation times for all semantic properties as described in Sect. 25.7.

25.8.2.2 Gaze Prediction at Runtime

At runtime, the importance map forms the basis for a per-object estimate of the probability of being attended. This probability is computed for those objects located within the field-of-view of the current frame, as illustrated in Fig. 25.11.

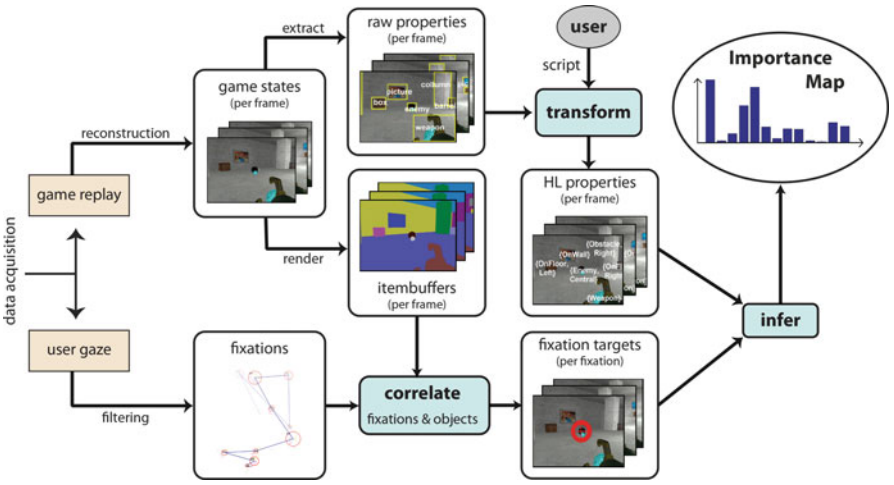


Fig. 25.10 An overview of the complete pipeline used by Bernhard et al. (2011) to derive importance maps

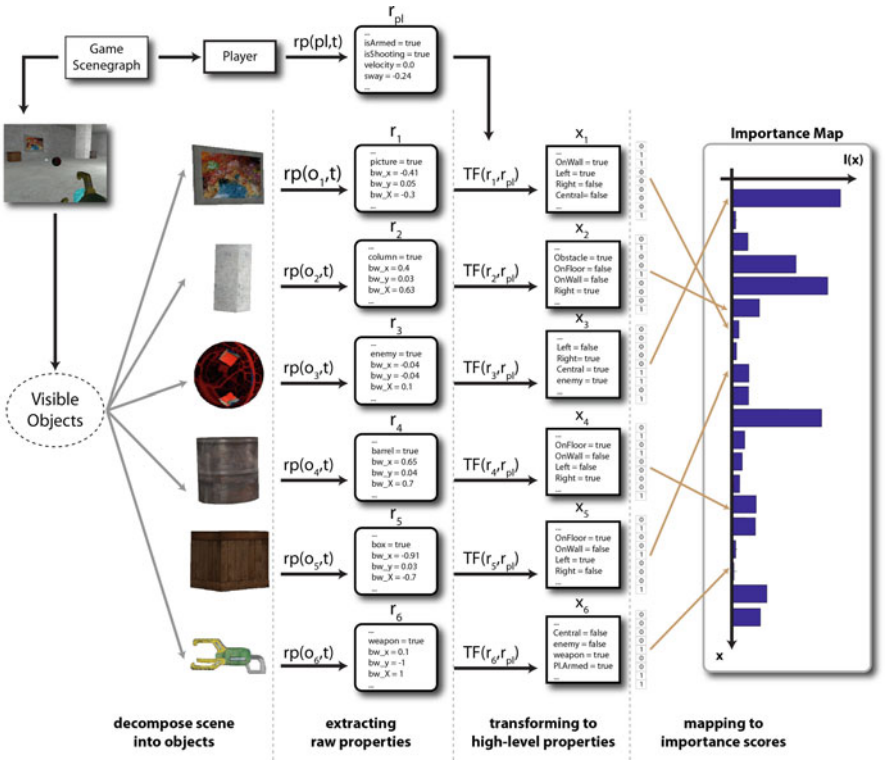


Fig. 25.11 The process of assigning importance values to objects in the player's field of view

First, a visibility algorithm determines the set of visible objects. For each of these objects, their properties are extracted from the scenegraph (Sect. 25.6.2), which are then transformed to high-level properties with a user specified function (discussed in Sect. 25.6.3). Boolean values are used to encode whether an object exhibits a certain high-level property or not. Therefore, the output of this mapping is a vector of boolean values which are then used as keys to perform a look-up into the importance map. With this process, a normalized importance value is determined for each visible object in the scene.

25.8.2.3 Discussion

Bernhard et al. (2010) evaluated their importance maps in the context of a first person shooter game. Included in their experiments are both a navigation task as well as a fighting task. Their importance maps are task dependent, for instance assigning lower importance to pictures on a wall when a player is engaged in opposing an enemy than when a player is navigating the virtual environment.

They found that the predictions of this importance map are of moderate quality during tasks where the player is less focused on a task, for instance when the player is navigating the environment. During periods when the player is very focused, i.e. during fighting against attacking enemies, up to 80% of the fixation time is deployed to enemies and explosions. This experiment reveals that under such conditions, objects representing enemies and explosions attract an exceptional amount of attention.

Since the game is a first person shooter, it can be expected that there is a strong bias toward fixating the center of the screen. This tendency can be exploited by encoding a measure of the degree of eccentricity from the spatial location of the objects. Bernhard's experiments show that this measure outperforms importance maps which rely on semantics, particularly in periods when there is less action in the game (e.g., during pure navigation tasks).

However, the highest predictive power is obtained when semantic and spatial information is combined. To keep the degrees of freedom low, object categories can, therefore, be clustered according to their importance values and then combined with eccentricity as an additional property.

25.9 Limitations

The work described in this chapter has opened a new avenue of gaze analysis methodology. Due to the fact that the authors were doing the very first steps into this field, many research efforts and technical investigations are required to bring their ideas forward to generally applicable tools.

One important technical limitation is the limited accuracy of the gaze-to-object mapping methods, which is discussed in Sect. 25.6.1.5. Another serious technical

problem is the collection and analysis of fixation statistics under the circumstance that each user has a unique game experience. As discussed in Sect. 25.7.1, there are no optimal strategies to fully compensate the latent effects caused by strong variations of the set of visible objects in the stimuli from one frame to another.

Moreover, the overall approach assumes a simplified world for games, which is composed of a set of objects with a clear semantic category and a clear geometrical outline. Though for many game objects this assumption holds true, commercial games frequently comprise more difficult content, e.g., large environment models or vegetations. Novel solutions need to be investigated, which allow decomposing all elements of a scene adequately, so that gaze can be analyzed according to the key features of major impact. Particular extensions to be considered are hierarchical decompositions of difficult objects, like trees or houses, and screen-space approaches to subdivide large models or background into regions in such a manner that features with a different response on visual attention can be spatially separated.

Finally, it is also important to further investigate the practical value of these tools. This includes evaluating their performance and value for a variety of different game types and examining in particular how these tools can be used to improve games.

25.10 Conclusion and Outlook

As this book discusses, evaluation of computer games is becoming increasingly important. While many chapters of the book focus on telemetry and game log data analysis, this chapter investigates eye tracking, which can be integrated with telemetry analysis and be used as one of the many tools at the disposal of the game user researcher. In addition to eye tracking and telemetry, there are also other techniques for gathering information from the player and to evaluate the gameplay experience, as discussed in several chapters in this book. Nacke et al. (2009) evaluated the experience of gaze-based interaction for example using different types of questionnaires; Chapter 24 in this book also reviews the use of questionnaires more extensively. More recently, eye tracking has been used in conjunction with psychophysiological data and game telemetry to evaluate the player experience. One way of evaluating the player experience is to gather quantitative data including biometric information from an electro encephalography (EEG), electromyography (EMG), galvanic skin response (GSR), heart rate (EKG), blood volume pulse (BVP), and breathing (Zammitto et al. 2010). This is also a subject discussed by Nacke et al. and McAllister et al. in Chaps. 26 and 27 of this book. Using additional input techniques in addition to gaze could give even further information regarding the state of the player.

The main focus of this chapter, however, is on the mapping of fixation points obtained with an eye tracker to semantic objects as defined by game designers. The methods employed are necessarily more involved than recording screen shots, but

the opportunities for understanding game players' behavior are numerous. The work presented here only begins to scratch the surface. We see this approach as a viable technique for game designers to test their designs prior to bringing their products to market. At the same time, our enhanced and extended mapping techniques could form the basis for further research, for instance in understanding driver behavior in driving simulators.

About the Authors

Veronica Sundstedt is an Assistant Professor at the Blekinge Institute of Technology in Sweden where she coordinates the Computer Graphics research subgroup. She was previously a lecturer in the GV2 (Graphics, Vision, and Visualisation) Group in the School of Computer Science and Statistics at Trinity College Dublin, Ireland. She worked as a Postdoctoral Research Associate in the Department of Computer Science at the University of Bristol and the University of Bath, UK. She holds a Ph.D. in Computer Graphics from the University of Bristol and an M.Sc. in Media Technology from the University of Linköping, Sweden. Her research interests are in computer graphics and perception, in particular perceptually-based rendering algorithms, experimental validation, novel interaction techniques, and eye tracking technology. She organized and co-chaired the first Novel Gaze-Controlled Applications (NGCA) conference in 2011 and co-chaired the ACM APGV conference in 2007 and the Eurographics Ireland workshop in 2009. She is also on the editorial board of the ACM Transactions on Applied Perception. Veronica is Program Co-Chair for the ACM Symposium on Applied Perception (formerly APGV) in 2012 and the lead author of the book: Gazing at Games: An Introduction to Eye Tracking Control.

Efstathios Stavrakis is currently Visiting Lecturer at with the University of Cyprus. He holds a Ph.D. in Computer Science from the Vienna University of Technology (Austria) and has studied for an M.Sc. in Computer-Aided Graphical Technology Application and a BA (Hons) in Creative Visualisation at the University of Teesside (UK). He has conducted and published research in computer games, graphics and vision, eye-tracking and psychophysics, non-photorealistic rendering, as well as 3D audio rendering for VEs. He brings a wealth of experience in graphical algorithms, interface design and software development. Previously, he has held posts at the Technical University of Vienna (Austria), at INRIA Sophia Antipolis – Méditerranée (France) and the Glasgow School of Art (UK).

Matthias Bernhard is a Ph.D. Student at the institute of Computer Graphics and Algorithms of the Vienna University of Technology. He received a Bachelor degree Information Engineering at the University of Konstanz in 2004 and his Master Degree in Medical Computer Science at Vienna University of Technology in 2006. He follows an interdisciplinary perspective and his current research interests include bimodal perception and the role of visual attention in virtual environments.

Michael Wimmer is an Associate Professor at the Institute of Computer Graphics and Algorithms of the Vienna University of Technology, where he received an M.Sc. in 1997 and a Ph.D. in 2001. His current research interests are real-time rendering, computer games, real-time visualization of urban environments, point-based rendering and procedural modeling. He has coauthored many papers in these fields, and was papers co-chair of EGSR 2008 and Pacific Graphics 2012. He also co-authored the book "Real-time Shadows".

Erik Reinhard received his Ph.D. in Computer Science from the University of Bristol in 2000, having worked on his Ph.D. at Delft University of Technology, as well as in Bristol. After holding a post-doctoral position at the University of Utah (2000–2002) and an Assistant Professorship at the University of Central Florida (2002–2005), he returned to Bristol as a lecturer in January 2006 to become senior lecturer in 2007. Erik founded the prestigious ACM Transactions on Applied Perception, and has been Editor-in-Chief since its inception in 2003, until early 2009. He is currently Associate Editor for this journal, as well as for Computers and Graphics. Erik is lead author of two books: 'High Dynamic Range Imaging: Acquisition, Display, and Image-Based Lighting' and 'Color Imaging: Fundamentals and Applications'. He is keynote speaker for Eurographics 2010, the Computational Color Imaging Workshop 2011 as well as the 6th European Conference on Color in Graphics, Imaging, and Vision (CGIV 2012). He is program co-chair for the Eurographics Symposium on Rendering 2011 and area co-chair for the high dynamic range imaging track at Eurographics 2011. Finally, his interests are in the application of knowledge from perception and neuroscience to help solve problems in graphics and related fields.

References

Baylis, G. C., & Driver, J. (1993). Visual attention and objects: evidence for hierarchical coding of location. *Journal of Experimental Psychology: Human Perception and Performance, 19*(3), 451–470.

Behrmann, M., Zemel, R. S., & Mozer, M. C. (1998). Object-based attention and occlusion evidence from normal participants and a computational model. *Journal of Experimental Psychology: Human Perception and Performance, 24*, 1011–1036.

Bernhard, M., Stavrakis, E., & Wimmer, M. (2010). An empirical pipeline to derive gaze prediction heuristics for 3D action games. *ACM Transactions on Applied Perception (TAP), 8*(1), 4:1–4:30.

Bernhard, M., Zhang, L., & Wimmer, M. (2011). Manipulating attention in computer games. *IVMSP workshop, 2011 IEEE 10th* (pp. 153–158), Ithaca, NY.

Canosa, R. L., Pelz, J. B., Mennie, N. R., & Peak, J. (2003). High-level aspects of oculomotor control during viewing of natural-task images. In B. E. Rogowitz & T. N. Pappas (Eds.), *Human vision and electronic imaging VIII. Proceedings of the SPIE in presented at the Society of Photo-Optical Instrumentation Engineers (SPIE) conference* (pp. 240–251). Santa Clara, CA.

Castiello, U., & Umiltà, C. (1990). Size of the attentional focus and efficiency of processing. *Acta Psychologica, 73*(3), 195–209.

Cater, K., Chalmers, A., & Ledda, P. (2002). Selective quality rendering by exploiting human inattentional blindness: Looking but not seeing. *Proceedings of the ACM Symposium on Virtual Reality Software and Technology* (pp. 17–24). Hong Kong, China.

Cater, K., Chalmers, A., & Ward, G. (2003). Detail to attention: Exploiting visual tasks for selective rendering. In *Proceedings of the 14th Eurographics workshop on Rendering* in EGRW '03 (pp. 270–280). Aire-la-Ville, Switzerland: Eurographics Association.

Chaney, I. M., Lin, K.-H., & Chaney, J. (2004). The effect of billboards within the gaming environment. *Journal of Interactive Advertising, 5*(1), 37–45.

Cunningham, D., & Wallraven, C. (2011). *Experimental design: From user studies to psychophysics*. Natick: A K Peters.

De Graef, P., Christiaens, D., & d'Ydewalle, G. (1990). Perceptual effects of scene context on object identification. *Psychological Research, 52*(4), 317–329.

Duchowski, A. T. (2003). *Eye tracking methodology: Theory and practice*. New York: Springer.

Duncan, J. (1984). Selective attention and the organization of visual information. *Journal of Experimental Psychology. General, 113*(4), 501–517.

Elazary, L., & Itti, L. (2008). Interesting objects are visually salient. *Journal of Vision, 8*(3:3), 1–15.

El-Nasr, M. S., & Yan, S. (2006). Visual attention in 3D video games. In *ACE 06: Proceedings of the 2006 ACM SIGCHI international conference on advances in computer entertainment technology* (p. 22). New York: ACM.

Eriksen, C., & St James, J. (1986). Visual attention within and around the field of focal attention: A zoom lens model. *Attention, Perception, & Psychophysics, 40*, 225–240.

Haber, J., Myszkowski, K., Yamauchi, H., & Seidel, H.-P. (2001). Perceptually guided corrective splatting. *Computer Graphics Forum, 20*(3), 142–152.

Hansen, D. W., & Ji, Q. (2010). In the eye of the beholder: A survey of models for eyes and gaze. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 32*(3), 478–500.

Hayhoe, M. M., Shrivastava, A., Mruczek, R., & Pelz, J. B. (2003). Visual memory and motor planning in a natural task. *Journal of Vision, 3*(1), 49–63.

Henderson, J. (2003). Human gaze control during real-world scene perception. *Trends in Cognitive Sciences, 7*(11), 498–504.

Henderson, J. M., Weeks, P. A., & Hollingworth, A. (1999). The effects of semantic consistency on eye movements during complex scene viewing. *Journal of Experimental Psychology Human Perception & Performance, 25*, 210–228.

Hillaire, S., Lécuyer, A., Cozot, R., & Casiez, G. (2008). Using an eye-tracking system to improve camera motions and depth-of-field Blur Effects in Virtual Environments. *VR* (pp. 47–50).

Hillaire, S., Breton, G., Ouarti, N., Cozot, R., & Lécuyer, A. (2010). Using a visual attention model to improve gaze tracking systems in interactive 3D applications. *Computer Graphics Forum, 29*(6), 1830–1841.

Hornof, A., Cavender, A., & Hoselton, R. (2003). Eyedraw: A system for drawing pictures with eye movements. *SIGACCESS Accessibility Computers* (pp. 86–93), Atlanta, GA, USA.

Isokoski, P., Joos, M., Spakov, O., & Martin, B. (2009). Gaze controlled games. *Universal Access in the Information Society, 8*, 323–337.

Itti, L., Koch, C., & Niebur, E. (1998). A model of saliency-based visual attention for rapid scene analysis. *IEEE Transaction on Pattern Analysis and Machine Intelligence, 20*(11), 1254–1259.

Itti, L., Dhavale, N., & Pighin, F. (2006). Photorealistic attention-based Gaze Animation. In *Proceedings of the IEEE international conference on multimedia and expo* (pp. 521–524). Toronto, Ontario, Canada

Jacob, R. J. K., & Karn, K. S. (2003). Eye tracking in human-computer interaction and usability research: Ready to deliver the promises. In J. Hyönä, R. Radach, & H. Deubel (Eds.), *The mind's eye: Cognitive and applied aspects of eye movement research* (pp. 573–605). Amsterdam: Elsevier.

James, W., & Anonymous. (1890). *The principles of psychology*, Vol. 1, volume reprint edition. New York: Dover Publications.

Jie, L., & Clark, J. J. (2007). Game design guided by visual attention. In L. Ma, M. Rauterberg, & R. Nakatsu (Eds.), *Entertainment computing, ICEC 2007 in Lecture Notes in Computer Science* (pp. 345–355), Shanghai: Springer.

Kenny, A., Koesling, H., Delaney, D., McLoone, S., & Ward, T. (2005). A Preliminary investigation into eye gaze data in a first person shooter game. In *19th European Conference on Modelling and Simulation*, Riga.

Koch, C., & Ullman, S. (1985). Shifts in selective visual attention: Towards the underlying neural circuitry. *Human Neurobiology, 4*, 219–227.

Komogortsev, O., & Khan, J. (2006). Perceptual attention focus prediction for multiple viewers in case of multimedia perceptual compression with feedback delay. In *ETRA '06: Proceedings of the 2006 symposium on eye tracking research & applications* (pp. 101–108). New York: ACM.

LaBerge, D. (1983). Spatial extent of attention to letters and words. *Journal of Experimental Psychology: Human Perception and Performance, 9*(3), 371–379.

Land, M., Mennie, N., & Rusted, J. (1999). The roles of vision and eye movements in the control of activities of daily living. *Perception, 28*(11), 1311–1328.

Lang, M., Hornung, A., Wang, O., Poulakos, S., Smolic, A., & Gross, M. (2010, July). Nonlinear disparity mapping for stereoscopic 3D. *ACM Transaction on Graphics, 29*(4), 1–75. doi:http://doi.acm.org/10.1145/1778765.1778812, URL: http://doi.acm.org/10.1145/1778765.1778812. New York: ACM.

Luebke, D., Hallen, B., Newfield, D., & Watson, B. (2000). Perceptually driven simplification using gaze-directed rendering.

Marmitt, G., & Duchowski, A. T. (2002). Modeling visual attention in VR: Measuring the accuracy of predicted scanpaths. In *Eurographics 2002, Short Presentations* (pp. 217–226). Saarbrücken, Germany.

McDonnell, R., Larkin, M., Hernández, B., Rudomin, I., & O'Sullivan, C. (2009). Eye-catching crowds: saliency based selective variation. *ACM Transactions on Graphics, 28*, 55:1–55:10.

Murphy, H., & Duchowski, A. T. (2001). Gaze-contingent level of detail rendering. In *Proceedings of EuroGraphics 2001 (Short Papers)*. EuroGraphics Association. Manchester, England.

Nacke, L., Lindley, C., & Stellmach, S. (2008). Log who's playing: Psychophysiological game analysis made easy through event logging. In P. Markopoulos, B. de Ruyter, W. IJsselsteijn, & D. Rowland (Eds.), *Fun and games in lecture notes in computer science* (pp. 150–157). Berlin/Heidelberg: Springer. 10.1007/978-3-540-88322-715.

Nacke, L., Stellmach, S., Sasse, D., & Lindley C. A. (2009). Gameplay experience in a gaze interaction game. In A. Villanueva, J. P. Hansen, & B. K. Ersbōll (Eds.) *Proceedings of the 5th conference on communication by Gaze Interaction Ð COGAIN 2009: Gaze Interaction for Those Who Want It Most* (pp. 49–54), Lyngby, Denmark. The COGAIN Association.

Nacke, L. E., Stellmach, S., Sasse, D., Niesenhaus, J., & Dachselt, R. (2011). LAIF: A logging and interaction framework for gaze-based interfaces in virtual entertainment environments. *Entertainment Computing, 2*(4), 265–273. <ce:title>Special Section: International Conference on Entertainment Computing and Special Section: Entertainment Interfaces</ce:title>.

Navalpakkam, V., & Itti, L. (2005). Modeling the influence of task on attention. *Vision Research, 45*(2), 205–231.

O'Sullivan, C. (2005). Collisions and attention. *ACM Transactions on Applied Perception, 2*(3), 309–321.

Oliva, A., Torralba, A., Castelhano M. S., & Henderson, J. M. (2003). Top-down control of visual attention in object detection. In *Proceedings of the IEEE International Conference on Image Processing (ICIP '03).* Barcelona, Catalonia, Spain.

Palmer, S. E. (1999). *Vision science: Photons to phenomenology*. Boston: MIT Press.

Pelz, J. B., & Canosa, R. (2001). Oculomotor behavior and perceptual strategies in complex tasks. *Vision Research, 41*, 3587–3596.

Peters, R. J., & Itti, L. (2008). Applying computational tools to predict gaze direction in interactive visual environments. *ACM Transactions on Applied Perception, 5*(2), 1–19.

Poole, A., & Ball, L. J. (2005). Eye tracking in human-computer interaction and usability research: Current status and future prospects. In C. Ghaoui (Ed.), *Encyclopedia of human-computer interaction*. Pennsylvania: Idea Group, Inc.

Rahardja, S., Farbiz, F., Manders, C., Zhiyong, H., Ling, J. N. S., Khan, I. R., Ping, O. E., & Peng, S. (2009). Eye HDR: Gaze-adaptive system for displaying high-dynamic-range images. *ACM SIGGRAPH ASIA 2009 Art Gallery & Emerging Technologies: Adaptation* in SIGGRAPH ASIA '09 (pp. 68–68). New York: ACM.

Ramloll, R., Trepagnier, C., Sebrechts, M., & Beedasy, J. (2004). Gaze data visualization tools: opportunities and challenges. In *Information Visualisation, 2004. IV 2004. Proceedings. Eighth International Conference on* (pp. 173–180). London, UK

Rothkopf, C. A., & Pelz, J. B. (2004). Head movement estimation for wearable eye tracker. In *Proceedings of the 2004 symposium on eye tracking research & applications* in ETRA '04 (pp. 123–130). New York: ACM.

Rothkopf, C. A., Ballard, D. H., & Hayhoe, M. M. (2007). Task and context determine where you look. *Journal of Vision, 7*(14), 1–20.

Saito, T., & Takahashi, T. (1990). Comprehensible rendering of 3-D shapes. *SIGGRAPH Computation Graphics, 24*(4), 197–206.

Salvucci, D. D., & Goldberg, J. H. (2000). Identifying fixations and saccades in eye-tracking protocols. In *Proceedings of the 2000 symposium on eye tracking research & applications* in ETRA '00 (pp. 71–78). New York: ACM.

Sasse D. (2008). *A framework for psychophysiological data acquisition in digital games*. Master's thesis, Otto-von-Guericke-University Magdeburg, Magdeburg.

Sennersten, C. (2004). *Eye movements in an action game tutorial*. Master's thesis, Lund University, Lund.

Sennersten, C., & Lindley, C. (2008). Evaluation of real-time eye gaze logging by a 3D game engine. In *12th IMEKO TC1 & TC7 joint symposium on man science and measurement* (pp. 161–168). Annecy, France.

Sennersten, C., & Lindley, C. (2009). An investigation of visual attention in FPS computer gameplay. In *Conference in games and virtual worlds for serious applications, VS-GAMES '09* (pp. 68–75). Coventry, UK.

Simons, D. J., & Chabris, C. F. (1999). Gorillas in our midst: Sustained inattentional blindness for dynamic events. *Perception, 28*, 1059–1074.

Snowden, R., Thompson, P., & Troscianko, T. (2006). *Basic vision: An introduction to visual perception*. Oxford University Press, USA.

Starker, I., & Bolt, R. A. (1990). A gaze-responsive self-disclosing display. In *CHI '90: Proceedings of the SIGCHI conference on human factors in computing systems* (pp. 3–10). New York: ACM.

Stellmach, S. (2007). A psychophysiological logging system for a digital game modification. Unpublished Internship Report, *Department of Simulation and Graphics*. Otto-von-Guericke-University, Magdeburg.

Stellmach S. (2009). *Visual analysis of Gaze Data in virtual environments*. Master's thesis, Otto-von-Guericke-University Magdeburg, Magdeburg.

Stellmach, S., Nacke, L., & Dachselt, R. (2010a). Advanced gaze visualizations for three-dimensional virtual environments. In *Proceedings of the 2010 symposium on eye-tracking research & Applications* in ETRA '10 (pp. 109–112). New York: ACM.

Stellmach, S., Nacke, L., & Dachselt, R. (2010b). 3D attentional maps: Aggregated gaze visualizations in three-dimensional virtual environments. In *Proceedings of the international conference on advanced visual interfaces* in AVI '10 (pp. 345–348). New York: ACM.

Stellmach, S., Nacke, L. E., Dachselt R., & Lindley C. A. (2010c). Trends and techniques in visual gaze analysis. *CoRR*, abs/1004.0258.

Sundstedt, V. (2007). *Rendering and validation of high-fidelity graphics using region-of-interest*. PhD thesis, University of Bristol, Bristol.

Sundstedt, V. (2010). Gazing at games: Using eye tracking to control virtual characters. *ACM SIGGRAPH 2010 Courses* in SIGGRAPH '10 (pp. 5:1–5:160). New York: ACM.

Sundstedt, V., Gutierrez, D., Anson, O., Banterle, F., & Chalmers, A. (2007). Perceptual rendering of participating media. *ACM Transaction on Applied Perception, 4*(3), 15.

Sundstedt, V., Stavrakis, E., Wimmer, M., & Reinhard, E. (2008). A psychophysical study of fixation behavior in a computer game. In *APGV '08: Proceedings of the 5th symposium on applied perception in graphics and visualization* (pp. 43–50). New York: ACM.

Sundstedt, V., Whitton, M., & Bloj, M. (2009). The whys, how tos, and pitfalls of user studies. *ACM SIGGRAPH 2009 Courses* in SIGGRAPH '09 (pp. 25:1–25:205). New York: ACM.

Tobii. (2006). *User manual: Tobii eye tracker, ClearView analysis software*.

Treisman, A. M., & Gelade, G. (1980). A feature-integration theory of attention. *Cognitive Psychology, 12*(1), 97–136.

van Zoest, W., & Donk, M. (2004). Bottom-up and top-down control in visual search. *Perception, 33*, 927–937.

Wolfe, J. M. (1994). Guided search 2.0: A revised model of visual search. *Psychonomic Bulletin and Review, 1*(2), 202–238.

Wolfe, J. (2000). Visual attention. In K. K. De Valois (Ed.), *Seeing* (pp. 335–386). San Diego: Academic.

Wolfe, J. M. (2007). Guided Search 4.0: Current Progress with a model of visual search. In Gray, W. (Ed.), *Integrated models of cognitive systems* (pp. 99–119). New York: Oxford University Press.

Wooding, D. S. (2002). Fixation maps: Quantifying eye-movement traces. *Proceedings of the 2002 symposium on eye tracking research & applications* in ETRA '02 (pp. 31–36). New York: ACM.

Yarbus, A. L. (1967). Eye movements during perception of complex objects. In *Eye movements and vision* (pp. 171–196). New York: Plenum Press.

Yee, H., Pattanaik, S., & Greenberg, D. P. (2001). Spatiotemporal sensitivity and visual attention for efficient rendering of dynamic environments. *ACM Transaction on Graphics, 20*(1), 39–65.

Zammitto, V., Seif El-Nasr, M., & Newton, P. (2010). Exploring quantitative methods for evaluating sports games. In *CHI 2010 workshop on brain, body and bytes: Psychophysiological user interaction*.