# DynAOI: A tool for matching eye-movement data with dynamic areas of interest in animations and movies

**FRANK PAPENMEIER AND MARKUS HUFF**
*Knowledge Media Research Center, Tübingen, Germany*

Analyzing gaze behavior with dynamic stimulus material is of growing importance in experimental psychology; however, there is still a lack of efficient analysis tools that are able to handle dynamically changing areas of interest. In this article, we present DynAOI, an open-source tool that allows for the definition of dynamic areas of interest. It works automatically with animations that are based on virtual three-dimensional models. When one is working with videos of real-world scenes, a three-dimensional model of the relevant content needs to be created first. The recorded eye-movement data are matched with the static and dynamic objects in the model underlying the video content, thus creating static and dynamic areas of interest. A validation study asking participants to track particular objects demonstrated that DynAOI is an efficient tool for handling dynamic areas of interest.

In everyday life, we are surrounded by dynamic objects. In road traffic, we routinely estimate the velocity of cars in order to cross the road. In sports reports on television, we are able to follow the complex choreography of a basketball game; and in educational scenarios, we have to deal with learning environments that present us with animated depictions of chemical reactions. In order to acquire a deeper understanding of how learners explore such dynamic visualizations, eyetracking methods have increasingly become a focus of experimental research. To facilitate the analysis of eye-movement data recorded with dynamic content, the tool DynAOI is introduced in the present article.

Analyzing eye-movement data is a popular means of gaining insight into attentional and cognitive processes. It has been used extensively in reading research (e.g., McDonald & Shillcock, 2003; Rayner, 1998; Vitu, Brysbaert, & Lancelin, 2004) and in research on scene perception (e.g., Brockmole & Henderson, 2007; Henderson, Weeks, & Hollingworth, 1999; Rayner, Smith, Malcolm, & Henderson, 2009). Within this research, eye movements have usually been classified as fixations or saccades and thereafter have been analyzed using fixation and saccade metrics (e.g., fixation duration), scan paths, or areas of interest (AOIs).

More recently, researchers started analyzing gaze data that were recorded with dynamic stimulus material. There are some studies that use measures that are unrelated to the dynamics, such as the distance of eye fixation to a critical place at a single point in time (Kuhn, Tatler, Findlay, & Cole, 2007), or the probability of saccades during film edits (Smith & Henderson, 2008). The focus of the present article, however, is on studies of eye movements toward the dynamic objects within their stimuli. There are several studies that have recorded eye movements during a multiple-object tracking task (Fehd & Seiffert, 2008; Huff, Papenmeier, Jahn, & Hesse, 2009; Zelinsky & Neider, 2008). During multiple-object tracking, participants are asked to spread their visual attention spatially in order to track a particular number of independently moving targets among indistinguishable moving distractors (Pylyshyn & Storm, 1988). These studies investigated different modes of tracking, such as target jumping or center tracking, by analyzing the gaze positions in relation to the positions of the moving objects. Eyetracking has been used with more complex dynamic content, too. Landry, Sheridan, and Yufik (2001) recorded eye movements during a collision-detection task using an air traffic control display.

But what makes the difference between analyzing eye movements in relation to static stimulus material or analyzing them in relation to dynamic stimulus material? Because of the dynamics, two new issues arise. First, there is another main eye-movement category beyond fixations and saccades: smooth pursuits (Rashbass, 1961), a smooth movement of the eyes that keeps dynamic objects foveated. Most eye-movement algorithms have been constructed for distinguishing fixations and saccades only, however (Salvucci & Goldberg, 2000). Second, the objects change their positions permanently in relation to a reference frame such as the screen. This makes it impossible to define a single set of AOIs as a static frame of reference for the moving stimuli with which to match the gaze data (see Figure 1). New approaches that do not rely on the classic analysis strat-

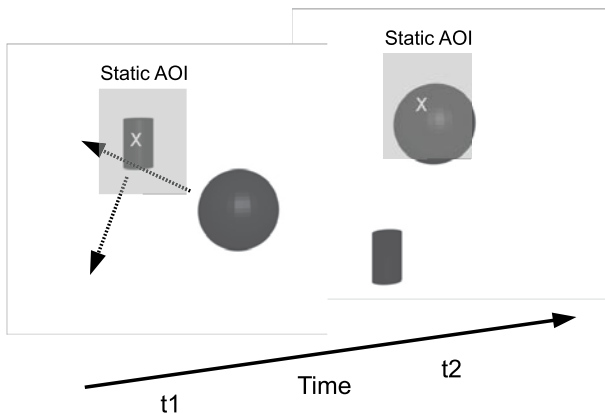F. Papenmeier, f.papenmeier@iwm-kmrc.de

**Figure 1. Example of eye movements in relation to dynamic stimulus material that cannot be analyzed using static AOIs. At both points in time, the gaze position at "X" would be matched with the static AOI; but at time t1 it represents the cylinder, and at time t2 it represents the sphere, making it meaningless across the whole trial if one is interested in distinguishing eye movements to the cylinder and sphere.**

egies for static stimulus material therefore need to be developed.

Below, we will give an overview of existing approaches; describe the main features of DynAOI, including its implementation; and report on our validation study.

## Overview of Existing Approaches

For research on eye movements toward dynamic stimuli there already exist several approaches dealing with these two issues—namely, the existence of smooth pursuits and the matching of eye movements with moving objects. Reimer and Sodhi (2006) worked on the first issue, the existence of smooth pursuits, by developing an algorithm that is capable of identifying the three eye-movement categories: fixations, saccades, and smooth pursuits. Using their algorithm, one can reduce the large amount of raw data to fewer data points representing eye movements of these categories. Thereafter, these identified eye movements need to be matched with the dynamic environment manually.

This is where the second issue of matching the recorded gaze positions with the moving objects arises. Several approaches attempt to deal with this problem. All of them work on the raw data (gaze positions) recorded by the eyetracker. A very basic method is the use of screen recordings. The dynamic stimuli that are presented on the screen are recorded, and a marker—such as a red ring—representing the participant's gaze is overlaid. The gaze data are matched with the relating objects manually using inspection of the video data (see, e.g., Sennersten, 2004). This process is very time consuming, however. Another approach is to use a shortest distance rule (Fehd & Seiffert, 2008; Zelinsky & Neider, 2008). For each point in time, the distance between the gaze position and a central position of each object is calculated. The object closest to the gaze position is the one that is counted as having been "looked at." This approach ignores the dimensions of the objects by taking the central position of each object as the

representation of the whole object. Furthermore, this procedure neglects the existence of eye movements that are not related to the objects under investigation, because each and every gaze position is matched to the closest object.

An alternative approach is the use of dynamic AOIs. In his dissertation, Bednarik (2007) explicitly asked for the development of dynamic AOIs to allow analysis to become more automatic. Dynamic AOIs are areas in dynamic stimulus material that represent the position and dimensions of important content over time. There are two main approaches in the development of dynamic AOIs: online AOIs, which match gaze position with the stimuli during the recording, and offline AOIs, which match gaze data with the stimuli after the recording.

Online AOIs are typically realized with the use of game engines (Florin, 2008; Holmberg, 2007; Sasse, 2008; Sennersten et al., 2007). Game engines allow the real-time visualization of virtual three-dimensional worlds, as is the case in first-person-shooter games such as *Quake*. The gaze positions are matched with the presented objects in real time; therefore, the static and moving objects within the virtual world themselves serve as AOIs. For example, while participants play a first-person-shooter game, several objects are presented: walls, moving opponents, weapons, and so forth. During the game, eye movements are recorded, and gaze coordinates are matched with these objects. This approach is very time efficient. Unfortunately, it is usually implemented for a specific eyetracker or for eyetrackers of a specific company only. Furthermore, it is not possible to rerun the matching process after removing some objects—for example, because they hindered gaze positions' being matched with objects behind them, such as windows.

There are fewer solutions for dynamic offline AOIs than for dynamic online AOIs. The only implementation known to the authors (besides the one being presented in the present article) has been realized by SensoMotoric Instruments in their analysis software BeGaze 2.1. Using their tool, one can overlay videos with polygons at various time points. Between any two time points, dynamic AOIs are calculated by interpolating the size and positions of the polygons. This method allows for the use of dynamic AOIs with any desired content at the cost of adding the AOIs manually, which is very time consuming. As a more efficient alternative, we will introduce a tool called *DynAOI* that allows for an automatic definition of offline AOIs for animations that are based on three-dimensional models.

## Overview of DynAOI

Working with dynamic stimulus material, one often uses animations that are based on three-dimensional models. A three-dimensional model represents a computer-simulated virtual environment. Objects are animated within the three-dimensional model and are then rendered to two-dimensional video files from the point of view of a virtual camera within the three-dimensional model. When using eyetrackers, the video files are presented and eye movements are recorded relative to the two-dimensional video files; however, it is not possible
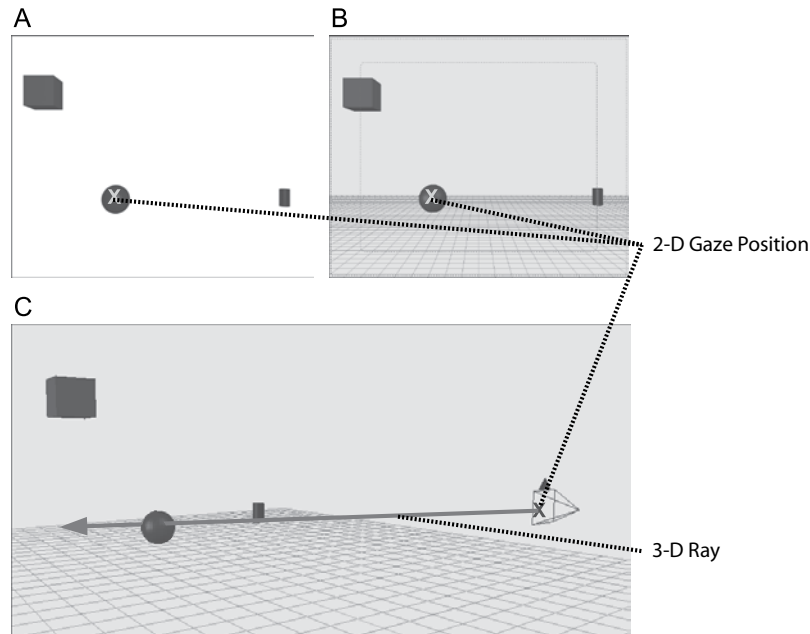
**Figure 2. Transformation of a two-dimensional gaze position relative to the presented video file (A) into a two-dimensional gaze position relative to the view of the virtual camera in the three-dimensional model (B) and, finally, into a three-dimensional ray representing all points in the three-dimensional model projecting on the two-dimensional gaze position (C).**

to match the gaze positions with the objects in the video files automatically. This is when DynAOI proves to be advantageous. DynAOI is a set of scripts released as open-source software (https://launchpad.net/dynaoi). It matches the recorded gaze positions with the three-dimensional models underlying the presented video files. The two-dimensional gaze coordinates relative to the video file (see Figure 2A) are converted to the corresponding two-dimensional coordinates in the projection plane of the camera within the three-dimensional model (see Figure 2B). Thereafter, these coordinates are transformed into a three-dimensional ray representing all points within the three-dimensional model that would project onto the corresponding two-dimensional coordinates (see Figure 2C). All objects lying on this ray were potentially looked at. Accordingly, the dynamic three-dimensional objects themselves serve as dynamic AOIs for the two-dimensional video files that are presented at the eyetracker. With regard to three-dimensional objects serving as dynamic AOIs, the present approach of defining dynamic offline AOIs is similar to the use of game engines to produce dynamic online AOIs that was mentioned above. It does differ with respect to certain features, however, which we will discuss below, including a discussion on how one might also use DynAOI with videos of real-world scenes.

**Blender and Python as development platform**. A large variety of three-dimensional modeling programs is used for the development of animations, including such programs as Autodesk 3ds Max (www.autodesk.com) and Blender (www.blender.org). For the present project,

Blender was chosen as a development platform for three main reasons. First, it is available as open-source software and is supported by a very active community. This guarantees an active development and allows for the modification of core functionalities if necessary. Second, it offers a broad functionality of accessing and manipulating objects in the three-dimensional model using the open-source programming language Python (www.python.org). This allows for a straightforward implementation of the DynAOI functionality using a set of scripts written in Python. And third, it offers a broad variety of import formats that allow the use of three-dimensional models created with different modeling software.

DynAOI was developed on, and was therefore tested on, a Windows platform. But it should be possible to port it easily to Mac OS X or Linux platforms, too, because both Blender and Python are available for those platforms.

**Data format and matching output**. When using DynAOI to match gaze data—recorded relative to two-dimensional video files—with the underlying three-dimensional model, the recorded raw data need to be exported from the recording software and then be converted to a file in the following format for each video clip: a comma-separated file with at least three columns using the headers "TIME," "X," and "Y" (as well as headers for any additional columns containing optional variables). The time 0 msec corresponds to the start of the video. The coordinates $x$ and $y$ must be given in pixels relative to the upper left corner of the video. Missing data are coded with the coordinates $-1$, $-1$. The required input data can be created easily from the raw data output of most eyetrack-

ers. A script for Tobii ClearView and Tobii Studio output is included in the DynAOI package.

During the matching process, each data point is read from the input file, and the following procedure is carried out. First, the three-dimensional model is set to the corresponding time point (frame). Next, the gaze position is transformed into a three-dimensional ray using the projection matrix of the virtual camera that defines the mapping of three-dimensional coordinates to the two-dimensional video coordinates and vice versa (see Figure 2). Intersections with objects are looked for along this ray, but only within the clipping area of the camera—that is, the part of the three-dimensional model that is "recorded" by the virtual camera. The intersection closest to the camera that contains a matchable object (explained below) is counted as "looked at" and is written to the output. This object closest to the camera occludes any other objects lying on the ray, and it is therefore visible in the two-dimensional video file. There may be cases in which the front object is not of interest and an object behind it was actually meant to be looked at—for example, when one is visually tracking a car driving past a tree, and the tree blocks the view of the car. This is one of the strengths of using DynAOI and postrecording (offline) AOIs, as it allows for a subset of objects to be defined as matchable, treating the rest as invisible. For example, it is possible to remove the foremost object (such as the tree in front of the car) and to rerun the matching process. As objects move in the three-dimensional model, they often overlap in the camera projection; therefore, the objects in their actual size themselves serve as dynamic AOIs. If the AOI size is increased by default in order to account for noise in the eyetracking system, false matching results would be possible. For instance, a gaze position lying right next to an object that is close to the camera would be attributed to this object even though the participant looked at an object that was positioned at a greater depth at this specific position. If the stimulus material is designed to avoid such ambiguities, however, the AOIs can easily be enlarged by increasing the size of the objects serving as AOIs in the three-dimensional model.

The output file of DynAOI corresponds to the input file, extended by the following three columns: "AOIMATCH," "MATCH.NUMBER," and "MATCH.DURATION." AOIMATCH holds the name of the "looked at" object or the value of a specified property variable associated with the object. In the case that no object was looked at, it stays empty. MATCH.NUMBER is a counter of changes to the AOIMATCH variable, comparable to a fixation number measure. Finally, MATCH.DURATION holds the time from the first match on an AOI to the last gaze that is still related to the AOI before the viewer looks somewhere else.

Naturally, there is also a batch mode allowing for a quick match of gaze data to AOIs for the whole experiment with a number of participants looking at different animations or movies.

**User interface and features of DynAOI**. The user interface of DynAOI (see Figure 3) allows for the following settings: *visualization*, *AOI tracking property*, and



**Figure 3. User interface of DynAOI.**

*AOI name*. The visualization option allows for a visual inspection of the matching process. When it is enabled, the three-dimensional model is opened with Blender, and the matching process is shown. If the model was saved from the camera's view, it looks similar to a gaze replay of raw data, with the differences that the three-dimensional model is shown instead of the two-dimensional video file and that the "looked at" (matched-with) object is highlighted. The AOI name option allows one to specify whether the name of the objects in the three-dimensional model or a property variable that is associated with them should be written as AOIMATCH to the output. The use of this option depends on the prior conception of the three-dimensional model. The last option is described together with the other features of DynAOI below:

1. Automatic matching of gaze data with dynamic and static objects.
2. Compatibility with virtually any eyetracker that supports the export of raw data.
3. The option to include only a subset of objects into the matching process (the "AOI Tracking Property" option in the user interface)—for example, excluding transparent objects such as windows.
4. Batch support: One can process all recordings that are located in a specified folder.
5. Versatility: No changes to the Blender three-dimensional model file are necessary.
6. Postrecording (offline) AOI matching: It is possible to edit models to meet specific needs—for example, adjusting the size of objects (AOIs).

**Using DynAOI with videos of real-world scenes**. Certainly, researchers do not work only with animations derived from three-dimensional models on the computer. One might be interested in working with video files or animations showing real world scenes, too. This is the case when using broadcasts of sporting events or clips from edited movies. DynAOI may be used with such material, too;
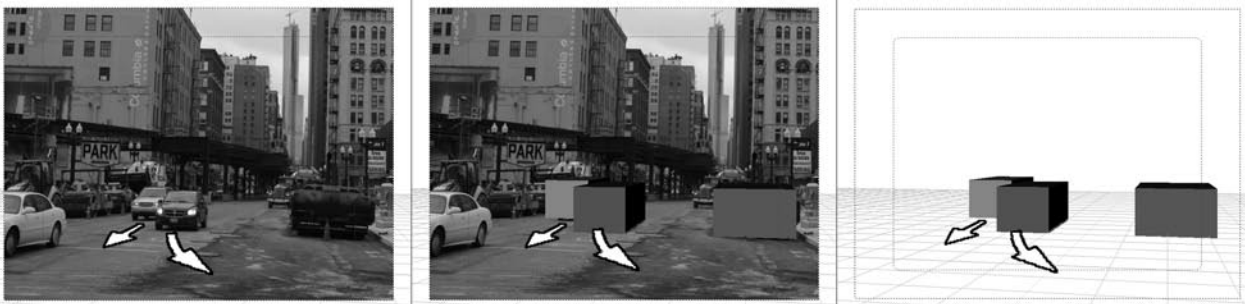
**Figure 4. Relevant content of a real-world scene (left) approximated by three-dimensional objects (middle) used as static and dynamic AOIs (right).**

however, some manual work needs to be done first. The three-dimensional model of the scene needs to be derived from the video data, since that model must be matched with gaze data. One method of doing so is to model some important objects and their trajectories manually as dynamic AOIs. To accomplish this, the video may be loaded into Blender. Then, the important content is approximated by some simple objects (e.g., spheres, cubes, or cuboids) that are added to a three-dimensional scene (see Figure 4). These objects may be animated by defining their positions at important key frames, with their intervening positions interpolated. This process of modeling the real-world scene may be very simple if the underlying trajectories are not too complex. In this case, the trajectories of the modeled objects determine not only their position but also their size in the camera projection, since they move the objects closer to the camera or farther away from it. The process of modeling a real-world scene may become very complex in the case of moving cameras or filmic cuts with abrupt viewpoint changes, however. Furthermore, this process of modeling a real-world scene can be very time consuming, so some additional tools supporting this process would be helpful and might be implemented in future versions of DynAOI.

Some approaches already exist that could support this process of creating a three-dimensional model of videos showing real-world scenes—that is, methods concerned with the automatic reconstruction of three-dimensional models from two-dimensional image sequences of static or dynamic scenes that have been recorded by a camera (see, e.g., Avidan & Shashua, 2000; Han & Kanade, 2004; Pollefeys, Koch, Vergauwen, & Van Gool, 2000). There is one active project attempting to implement some methods for three-dimensional reconstruction, or at least the reconstruction of the camera movement, into Blender (http://code.google.com/p/libmv/). This would be very helpful for the efficient use of videos of real-world scenes with DynAOI, allowing one to match gaze data with the actual objects depicted in a video.

Another approach for supporting the construction of three-dimensional models for videos of real-world scenes is the use of motion-capture techniques. Usually, human motion—by an actor, for example—is recorded using some kind of markers and is used to animate a virtual three-dimensional model. Although these techniques have been restricted to use in specialized studios in the past, new developments allow for the use of motion capture in everyday surroundings (Vlasic et al., 2007). Extending this approach to use markers with arbitrary motion and to track the position, along with rotation and zoom, of the camera, would provide for simultaneous recording of a video and a three-dimensional model of a real-world scene. The three-dimensional model of the scene could then allow DynAOI to use the real-world objects as static and dynamic AOIs.

## VALIDATION OF DynAOI

We conducted a study in order to validate the performance of DynAOI. For this purpose, a short animation was created. Participants were instructed to view it freely or to look at a specified object.

### Method

**Participants**. Twenty students from the University of Tübingen (mean age = 26.9 years) participated in this experiment. They reported normal or corrected-to-normal vision. All participants were compensated for their participation.

**Apparatus and Stimuli**. A Tobii 1750 Eye Tracker (refresh rate, 60 Hz) was used for the presentation of stimuli and the recording of eye movements at 50 Hz. Spatial accuracy is specified with 0.5 deg. A chin- and headrest ensured that participants kept a distance of 60 cm from the screen. The stimulus consisted of a video file created using the three-dimensional modeling software Blender. The video lasted 33 sec and showed three objects (cube, sphere, and cylinder) moving at the trajectories that are depicted in Figure 5. The cube, sphere, and cylinder moved at mean (and peak) velocities of 2.6 deg/sec (5.2 deg/sec), 2.9 deg/sec (4.9 deg/sec), and 11.0 deg/sec (21.7 deg/sec), respectively. They subtended the following degrees of visual angle: 3.1 deg (undistorted cube face's edge length), 3.1 deg (diameter of sphere), and 2.1 × 1.2 deg (height × width of cylinder). A blue background was used, and the video files were presented in full-screen mode at a resolution of 1,280 × 1,024 pixels.

**Procedure and Design**. After reading the instructions, participants were asked to put their head onto a chin- and headrest. A 9-point calibration procedure was performed. Thereafter, an instruction screen, followed by the video (illustrated in Figure 5) was shown four times. Participants viewed the same video four times, with the following instructions in succession: "View freely," "Track the red cube," "Track the blue sphere," and "Track the black cylinder." The experiment took approximately 10 min to complete.

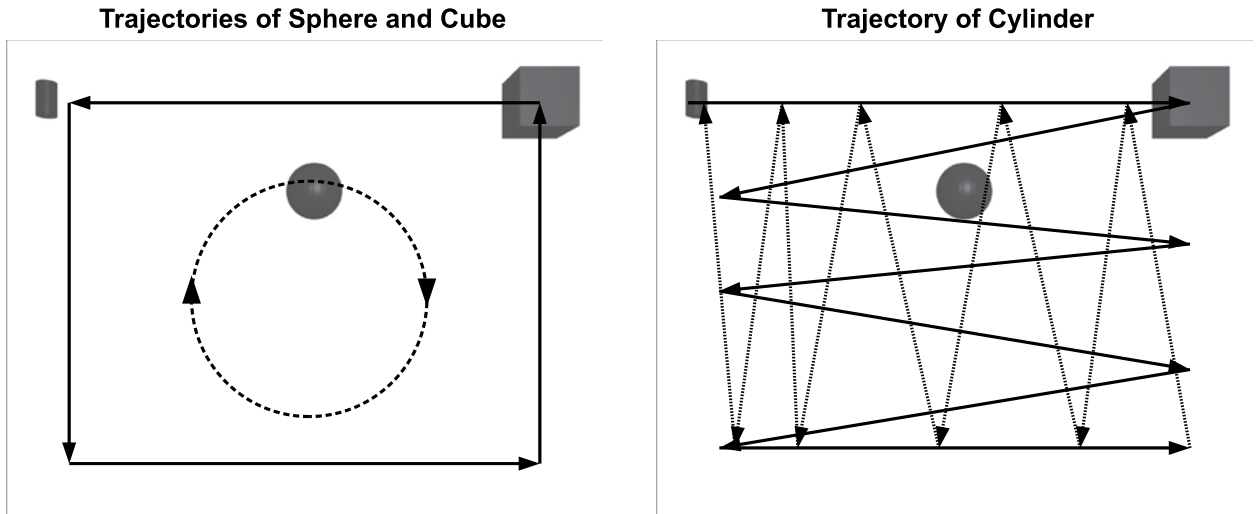**Trajectories of Sphere and Cube**          **Trajectory of Cylinder**



Figure 5. Trajectories of the objects in the video that was presented to participants. On the left are the trajectories of the sphere and the cube. The sphere traveled the depicted trajectory twice. On the right is the trajectory of the cylinder. All objects moved simultaneously, and trajectories are depicted separately for purposes of illustration only. Note that the blue background that was used in the study was removed for illustration purposes.

## Results and Discussion

A first inspection of the gaze replays revealed that 1 participant misunderstood the "track the blue sphere" instruction and actually tracked the black cylinder. This participant was therefore excluded from the following analysis.

The recorded eye-movement data were matched with the dynamic AOIs using DynAOI and the three-dimensional Blender model underlying the presented video file. Thereafter, the dependent measure *proportion matched* was computed for the three AOIs (cube, sphere, and cylinder) by dividing the number of gaze points that

were matched with an AOI by the total number of recorded gaze points separately for each instruction condition (see Figure 6).

For each instruction condition, the differences in proportion matched between the AOIs were tested using three separate two-tailed paired $t$ tests with a Bonferroni-corrected alpha criterion of 1.67% (see Table 1). In the *view freely* condition, eye movements were distributed among the AOIs, with the sphere being looked at more than the cube, and both being looked at more than the cylinder [all $t$s(18) $\geq$ 3.19, $p$s $\leq$ .005]. This gaze pattern represents the spatial and dynamic structure of the
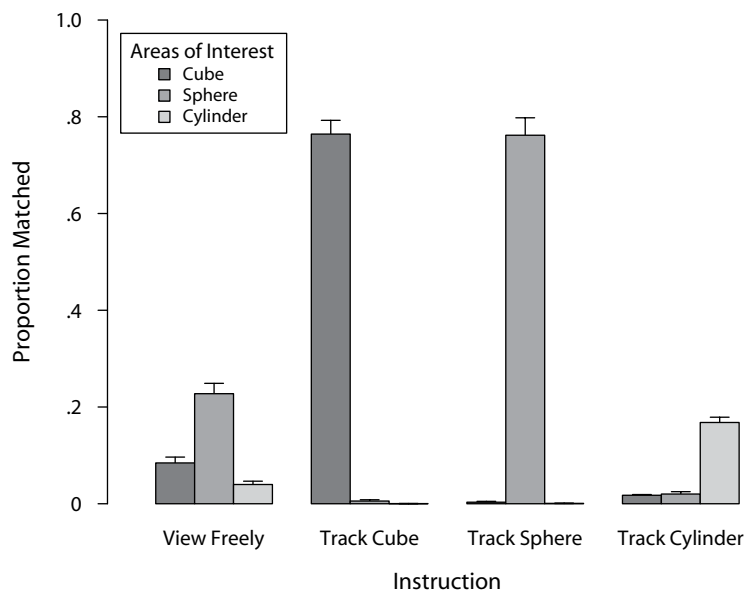


Figure 6. Proportion-matched score for the AOIs under the four different instructions. Error bars indicate the standard errors of the means.

**Table 1**
**Results of Two-Tailed Paired *t* Tests Between AOIs Under the Four Different Instruction Conditions, Tested Against the Bonferroni-Corrected Alpha Criterion of 1.67%**

| | Condition | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | View Freely | | Track Cube | | Track Sphere | | Track Cylinder | |
| Comparison | $t(18)$ | $p$ | $t(18)$ | $p$ | $t(18)$ | $p$ | $t(18)$ | $p$ |
| Cube–Cylinder | 3.19 | .005 | 26.69 | <.001 | 1.51 | .150 | −13.75 | <.001 |
| Cube–Sphere | −6.50 | <.001 | 25.25 | <.001 | −20.54 | <.001 | −0.54 | .593 |
| Cylinder–Sphere | −7.51 | <.001 | −2.15 | .046 | −21.02 | <.001 | 11.08 | <.001 |

video clip. Both size and trajectory of the moving objects determined the distribution of gazes. Object size is negatively correlated with the number of errors when one tries to fixate an object (Ware & Mikaelian, 1987), which led to the fewest gazes on the black cylinder in the present experiment. Further, the "center-of-screen bias" was responsible for more gazes located on the blue sphere, because its trajectory was located nearest to the center of the screen (Brasel & Gips, 2008; Tatler, 2007; Tosi, Mecacci, & Pasquali, 1997).

Instructing participants to track the red cube caused higher proportion-matched scores for the cube than for the cylinder or the sphere [both $t$s(18) ≥ 25.25, $p$s < .001]. Furthermore, marginally more gaze data were related to the sphere than to the cylinder, reflecting gaze behavior in the baseline condition [$t(18) = −2.15$, $p = .046$]. In the *track sphere* and *track cylinder* conditions, the specified object was looked at more than were the other objects [all $t$s(18) ≥ 11.08, $p$s < .001]. The difference between the two other objects was not significant in both cases [both $t$s(18) ≤ 1.51, $p$s ≥ .150].

Even though the relation among the objects in the track cylinder condition was as one would expect given the instructions, the proportion-matched score for the cylinder (.17) was much lower than the proportion-matched scores for the sphere (.76) and the cube (.76) in the condition in which they were to be tracked [both $t$s(18) ≥ 19.45, $p$ < .001], whereas there was no significant difference between the tracked sphere and tracked cylinder [$t(18) = .12$, $p = .905$]. The lower number of gazes on the cylinder might be the result of two things: The cylinder moves much faster than the other two objects, and its trajectory shows a lot of acceleration and deceleration. To exclude the possibility of timing issues of the DynAOI tool that emerge with the faster moving cylinder only, a gaze-overlaid playback of the recording was compared to the DynAOI matching process and showed that they were synchronous. As a result, this possibility could be rejected; however, the repetitive acceleration and deceleration of the cylinder might have caused the visual system to struggle to track the moving object because of the need for repetitive adjustments in the speed of the smooth pursuits and the need for corrective saccades (Rashbass, 1961). The gaze-overlaid playback was in accordance with this assumption, further showing predictive saccades to the upcoming position of the cylinder and therefore not lying on the cylinder too. In order to test statistically for an increase in saccades, we detected saccades using a velocity-based saccade-detection algorithm with a velocity threshold of 50 deg/sec that is far

above the peak velocity of the fastest object of 21.7 deg/sec, excluding the possibility of falsely detecting smooth-pursuit eye movements as saccades. The mean number of saccades was higher in the track cylinder condition (171) than in the track cube (132) and track sphere (119) conditions [both $t(18)$ ≥ 6.09, $p$s < .001], whereas there was no significant difference between the track cube and track sphere conditions [$t(18) = 1.28$, $p = .218$]. Additionally, since the cylinder was smaller than the sphere and cube, eye movements that were directed toward the cylinder might have been located just next to it, leading to lower proportion-matched scores. Taking advantage of the offline AOI approach of DynAOI, the underlying three-dimensional model was modified by doubling the size of the cylinder used as a dynamic AOI. Thereafter, a second matching process, using the modified model, revealed a significant increase in the proportion-matched score for the cylinder from .17 to .42 in the track cylinder condition [$t(18) = 22.82$, $p < .001$]; therefore, part of the lower proportion-matched score for the tracked cylinder could be explained by the small size of the object. The tracked cylinder's lower proportion-matched score relative to those for the tracked sphere and the tracked cube does not necessarily indicate that less attention was devoted to the quickly moving cylinder. As attention shifts to the target of a saccade prior to the occurrence of the saccade (see, e.g., Hoffman & Subramaniam, 1995; Posner, 1980; Shepherd, Findlay, & Hockey, 1986), it creates a dissociation of attentional and gaze distribution that, in the present study, was higher for the cylinder object as a result of the higher number of saccades. Additionally, it is known from research on scene perception that viewers need to view a scene for at least 150 msec during each fixation in order to process it normally (Rayner et al., 2009). It would therefore be interesting to investigate the influence of the duration of gazes at dynamic AOIs on the processing of the content of the AOIs, thus gaining further insights into how to interpret the data obtained with DynAOI.

In order to explore the influence of noise in raw data on the outcome of the matching process with DynAOI, the raw data were smoothed using a running median with a window width of five data points—that is, 100 msec. Thereafter, the smoothed raw data were matched with the dynamic AOIs using DynAOI, producing new proportion-matched scores. In each instruction condition, we compared the smoothed and unsmoothed proportion-matched score for the instructed object using paired *t* tests. This showed a significant increase in matching quality for all three instruction conditions: track cube (unsmoothed, .76;

smoothed, .78); track sphere (unsmoothed, .76; smoothed, .78); and track cylinder (unsmoothed, .17; smoothed, .18) [all $ts(18) \geq 3.92$, $ps \leq .001$].

Thus far, the validation of DynAOI was based on analysis of the data produced by DynAOI in relation to predefined instructions that were presented to the participants. In order to gain further insights into the quality of matching data that were produced by DynAOI, an additional external validation was performed. For 1 randomly selected participant, each raw gaze-position data point was overlaid on the screenshot of the current video's state and was matched to the presented objects manually. This manual scoring was compared to the automatic scoring of DynAOI using Cohen's kappa. The results showed a very high interrater agreement for all instruction conditions (view freely, .98; track cube, .95; track sphere, .98; track cylinder, .96), providing further evidence that DynAOI produces high-quality matching data for dynamic AOIs.

On the whole, participants were able to comply with the presented instruction to look at a specific moving object. As a result, we observed strong effects with significantly more gaze points related to the instructed objects. These effects were shown for all three tracking instructions, thus demonstrating that DynAOI is an efficient tool for handling dynamic AOIs.

## CONCLUSION

We introduced the tool DynAOI, which allows for matching gaze data that are recorded with two-dimensional video files, such as animations or movies, with static and dynamic AOIs. This is achieved by using the static and animated objects placed in a three-dimensional model as AOIs representing the videos. Although DynAOI works automatically with animations that are already based on three-dimensional models, some preparatory work—creating the three-dimensional models first—needs to be done when working with videos of real-world scenes. In contrast to existing approaches implementing dynamic online AOIs for specific eyetrackers using game engines (Florin, 2008; Holmberg, 2007; Sasse, 2008; Sennersten et al., 2007), DynAOI allows for a more flexible analysis. It works with virtually any eyetracker that permits exportation of the recorded raw data. Furthermore, gaze data are matched to the AOIs postrecording (offline). This allows for greater flexibility, permitting modifications such as excluding uninformative objects (e.g., transparent windows) from the model or adjusting the size of AOIs. Depending on research objectives, different AOI definitions could be tested against each other in terms of how well they describe the observed eye movements. DynAOI operates on raw data instead of fixations and saccades, because the presence of the dynamics can elicit smooth pursuits (Rashbass, 1961). Operating on data that are run through a fixation algorithm would lead to false conclusions. Nevertheless, raw data include more noise than do data run through a fixation algorithm. One should therefore consider applying a filter—for example, a running median—to the raw data. In the present validation study, this approach produced a better match of gaze-position data

and instructions. When one is working with high-speed eyetracking devices, the closing and opening of the eyelid at the beginning and end of blinks results in position data that are biased downward. The raw data should therefore be preanalyzed, with the invalid position data surrounding blinks also marked as missing data. Furthermore, DynAOI is thus far restricted to monocular measurements. When multiple objects on different depth planes in the three-dimensional model correspond to the gaze position, the object that is closest to the camera is counted as looked at. By extending DynAOI to use binocular gaze data and taking the vergence angle into account, one might be able to resolve this ambiguity and count an object at a certain depth plane as being looked at.

The validation study demonstrated the proper functioning of DynAOI. In addition, it showed that the actual motion and size of objects need to be taken into account. The cylinder was the smallest object in the scene, and it showed a repetitive acceleration and deceleration. One or both of these factors caused gaze positions to be located just next to the object instead of exactly on it. The validation study demonstrated that in such cases one should consider enlarging the AOI. Since DynAOI works offline, this can be achieved easily by enlarging such objects in the three-dimensional model and rerunning the matching process. This worked well for the present validation study, in which doubling the size of the cylinder explained a substantially higher number of eye movements directed toward it. Nevertheless, such modifications should always be based on theoretical and/or practical deliberations.

In summary, DynAOI allows for a highly efficient definition of dynamic AOIs, thus simplifying the analysis of eye-movement data in relation to dynamic stimulus material such as animations or movies, and its release as open-source software makes it available to interested researchers.

## REFERENCES

AVIDAN, S., & SHASHUA, A. (2000). Trajectory triangulation: 3D reconstruction of moving points from a monocular image sequence. *IEEE Transactions on Pattern Analysis & Machine Intelligence*, **22**, 348-357. doi:10.1109/34.845377

BEDNARIK, R. (2007). *Methods to analyze visual attention strategies: Applications in the studies of programming*. Unpublished doctoral dissertation, University of Joensuu.

BRASEL, S. A., & GIPS, J. (2008). Points of view: Where do we look when we watch TV? *Perception*, **37**, 1890-1894. doi:10.1068/p6253

BROCKMOLE, J. R., & HENDERSON, J. M. (2007). Prioritizing new objects for eye fixation in real-world scenes: Effects of object–scene consistency. *Visual Cognition*, **16**, 375-390. doi:10.1080/13506280701453623

FEHD, H. M., & SEIFFERT, A. E. (2008). Eye movements during multiple

object tracking: Where do participants look? *Cognition*, **108**, 201-209. doi:10.1016/j.cognition.2007.11.008

FLORIN, H. (2008). *Eye-tracking framework für dynamische AOIs*. Unpublished diploma thesis, Eberhard Karls Universität Tübingen.

HAN, M., & KANADE, T. (2004). Reconstruction of a scene with multiple linearly moving objects. *International Journal of Computer Vision*, **59**, 285-300. doi:10.1023/B:VISI.0000025801.70038.c7

HENDERSON, J. M., WEEKS, P. A., JR., & 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. doi:10.1037/0096-1523.25.1.210

HOFFMAN, J. E., & SUBRAMANIAM, B. (1995). The role of visual attention in saccadic eye movements. *Perception & Psychophysics*, **57**, 787-795.

HOLMBERG, A. (2007). *Eye tracking and gaming: Eye movements in Quake III. Arena*. Master's thesis, Lund University.

HUFF, M., PAPENMEIER, F., JAHN, G., & HESSE, F. W. (2009). *Eye movements across viewpoint changes in multiple object tracking*. Manuscript submitted for publication.

KUHN, G., TATLER, B. W., FINDLAY, J. M., & COLE, G. G. (2007). Misdirection in magic: Implications for the relationship between eye gaze and attention. *Visual Cognition*, **16**, 391-405. doi:10.1080/13506280701479750

LANDRY, S. J., SHERIDAN, T. B., & YUFIK, Y. M. (2001). A methodology for studying cognitive groupings in a target-tracking task. *IEEE Transactions on Intelligent Transportation Systems*, **2**, 92-100. doi:10.1109/6979.928720

MCDONALD, S. A., & SHILLCOCK, R. C. (2003). Eye movements reveal the on-line computation of lexical probabilities during reading. *Psychological Science*, **14**, 648-652. doi:10.1046/j.0956-7976.2003.psci_1480.x

POLLEFEYS, M., KOCH, R., VERGAUWEN, M., & VAN GOOL, L. (2000). Automated reconstruction of 3D scenes from sequences of images. *ISPRS Journal of Photogrammetry & Remote Sensing*, **55**, 251-267. doi:10.1016/S0924-2716(00)00023-X

POSNER, M. I. (1980). Orienting of attention. *Quarterly Journal of Experimental Psychology*, **32**, 3-25. doi:10.1080/00335558008248231

PYLYSHYN, Z. W., & STORM, R. W. (1988). Tracking multiple independent targets: Evidence for a parallel tracking mechanism. *Spatial Vision*, **3**, 179-197. doi:10.1163/156856888X00122

RASHBASS, C. (1961). The relationship between saccadic and smooth tracking eye movements. *Journal of Physiology*, **159**, 326-338.

RAYNER, K. (1998). Eye movements in reading and information processing: 20 years of research. *Psychological Bulletin*, **124**, 372-422.

RAYNER, K., SMITH, T. J., MALCOLM, G. L., & HENDERSON, J. M. (2009). Eye movements and visual encoding during scene perception. *Psychological Science*, **20**, 6-10. doi:10.1111/j.1467-9280.2008.02243.x

REIMER, B., & SODHI, M. (2006). Detecting eye movements in dynamic environments. *Behavior Research Methods*, **38**, 667-682.

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 and applications* (pp. 71-78). New York: ACM Press. doi:10.1145/355017.355028

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

SENNERSTEN, C. (2004). *Eye movements in an action game tutorial*. Unpublished master's thesis, Lund University, Lund, Sweden.

SENNERSTEN, C., ALFREDSON, J., CASTOR, M., HEDSTRÖM, J., LINDAHL, B., LINDLEY, C., & SVENSSON, E. (2007). *Verification of an experimental platform integrating a Tobii eyetracking system with the HiFi game engine* (Report No. FOI-R-2227-SE). Retrieved from FOI, Swedish Defense Research Agency: http://www2.foi.se/rapp/foir2227.pdf.

SHEPHERD, M., FINDLAY, J. M., & HOCKEY, R. J. (1986). The relationship between eye movements and spatial attention. *Quarterly Journal of Experimental Psychology*, **38A**, 475-491.

SMITH, T. J., & HENDERSON, J. M. (2008). Edit blindness: The relationship between attention and global change blindness in dynamic scenes. *Journal of Eye Movement Research*, **2**(2, Art. 6), 1-17.

TATLER, B. W. (2007). The central fixation bias in scene viewing: Selecting an optimal viewing position independently of motor biases and image feature distributions. *Journal of Vision*, **7**(14, Art. 4), 1-17. doi:10.1167/7.14.4

TOSI, V., MECACCI, L., & PASQUALI, E. (1997). Scanning eye movements made when viewing film: Preliminary observations. *International Journal of Neuroscience*, **92**, 47-52.

VITU, F., BRYSBAERT, M., & LANCELIN, D. (2004). A test of parafoveal-on-foveal effects with pairs of orthographically related words. *European Journal of Cognitive Psychology*, **16**, 154-177. doi:10.1080/09541440340000178

VLASIC, D., ADELSBERGER, R., VANNUCCI, G., BARNWELL, J., GROSS, M., MATUSIK, W., & POPOVIC, J. (2007). Practical motion capture in everyday surroundings. *ACM Transactions on Graphics*, **26**, 35. doi:10.1145/1276377.1276421

WARE, C., & MIKAELIAN, H. H. (1987). An evaluation of an eye tracker as a device for computer input. *ACM SIGCHI Bulletin*, **18**, 183-188. doi:10.1145/29933.275627

ZELINSKY, G. J., & NEIDER, M. B. (2008). An eye movement analysis of multiple object tracking in a realistic environment. *Visual Cognition*, **16**, 553-566. doi:10.1080/13506280802000752