OGAMA (Open Gaze and Mouse Analyzer): Open-source software designed to analyze eye and mouse movements in slideshow study designs

ADRIAN VOßKÜHLER, VOLKHARD NORDMEIER, LARS KUCHINKE, AND ARTHUR M. JACOBS
Freie Universität, Berlin, Germany

In the present article, a new software is introduced that allows the recording and analyzing of eye- and mouse-tracking data from slideshow-based experiments in parallel. The Open Gaze and Mouse Analyzer (OGAMA) is written in C#.NET and has been released as an open-source project. Its main features include slideshow design, the recording of gaze and mouse data, database-driven preprocessing and filtering of gaze and mouse data, the creation of attention maps, areas-of-interest definition, and replay. Eyetracking and/or presentation soft- and hardware recordings in ASCII format can be imported. Data output is provided that can be used directly with different statistical software packages. Because it is open source, one can easily adapt it to suit one's needs.

Eyetracking is used in a fast-growing field of research (Duchowski, 2003; Rayner, 1998). Measuring gaze positions and eye movements is used in fundamental research in psychology (research on reading—e.g., Rayner, 1998; visual search—e.g., Gitelman, Parrish, Friston, & Mesulam, 2002; or developmental disorders, e.g., Hutzler, Kronbichler, Jacobs, & Wimmer, 2006), as well as in applied research areas like usability engineering (e.g., Goldberg & Kotval, 1998; Nielsen, 2004). In recent years, eyetracking research has benefited from the fact that hardware and applications development started to complement one another so that, currently, the hardware is stable and tailored to a wide range of applications. Still, the software for analyzing the data "is usually proprietary, nonextensible . . . and nonmodifiable in the sense that the user cannot add additional analyses or modify the algorithms" (Gitelman, 2002, p. 605).

Inspired by this challenge, we decided to start an open-source project for developing a gaze analyzer. We think that the "open source" idea is the best way to combine the power of different researchers into one tool that provides them with the opportunity to fulfill their own needs. Such a project should combine commonly known "state-of-the-art" gaze-analyzing tools, such as attention maps, areas-of-interest (AOI) definition, and real-time replay, along with an easy-to-use graphical user interface. Because of the huge amounts of data that are recorded during eyetracking experiments, the tool should rely on visual representations of the results so that the human image processing can identify the crucial aspects on which to concentrate in both further processing and the statistical analysis. From our point of view, graphical representations and visualizations of gaze data are a key to eye-movement research, especially for the facilitation of the data-mining processes

(Tan, Steinbach, & Kumar, 2006). Therefore, such a tool should be designed in a powerful programming language that handles large databases, supplies flexible graphical interfaces, and provides the researchers with a basis for customization—for example, the ability to define new saccade-related variables.

At the same time, coming from our own applied research in the field of physics education, it would be nice to have a tool that can handle mouse-tracking data as well. It is known from usability research during Web browsing that there is a strong relationship between gaze position and mouse position (Chen, Anderson, & Sohn, 2001).

In physics education, we investigate students interacting with direct manipulative media (Shneiderman & Plaisant, 2004) in screen-based environments. Does the close relationship between gaze and mouse movements hold true for interactive screen experiments (ISEs) used in learning situations (Kirstein & Nordmeier, 2007), or do gaze and mouse patterns differ depending on previous knowledge? To improve the conclusions drawn from tracking ISE, we required a tool that merges eyetracking data with mouse-movement data.

In the present article, we describe a software application that is intended for recording and analyzing gaze and mouse data in parallel. We called this tool OGAMA—the Open Gaze and Mouse Analyzer. OGAMA is suitable for experimental setups with screen-based slideshow stimuli.

The main goals in designing OGAMA were the following:

1. The recording and analyzing of gaze- and mouse-movement data together.

2. Open-source publishing to provide extensibility to the research community.

A. Voßkühler, adrian.vosskuehler@physik.fu-berlin.de

3. The implementation of state-of-the-art eyetracking analysis functions such as replay, AOI definition, and attention maps in an up-to-date user interface.

4. Visual-guided data mining through the use of different visual representations.

5. Database-driven software layout to handle highly sampled eye- and mouse-tracking data.

The article is split into two main sections. In the Application Design section, we describe in detail basic principles of the underlying application structure, the database, and the fixation detection algorithm. The other section consists of a brief description of the currently implemented application interfaces to give an overview of the processing steps to follow during the analysis from the data import to the computation of the statistical output variables.

2. Application Design

In order to have full freedom in developing algorithms as well as their fast implementation, and—on the other side—to have fast and powerful tools for the graphical output and the user interface, we decided to use the Microsoft Visual Studio development suite with the language C#.NET. This is a low-level programming language whose performance enables real-time support and fast data access.

2.1. OGAMA's storage files. Each experiment resides in its own folder, with four subdirectories. The .oga file, which has an xml layout, stores the key values for the description of each experiment. These are the database file, the path to stimuli, sampling rates, eyetracker screen size, and the fixation calculation parameters. This file can be used to start the OGAMA session by double-clicking on it. The "Database" subfolder contains the two database files. They are pure SQL standard, so one can access the database without OGAMA with any SQL-enabled language or application (e.g., Microsoft Access). For use during analysis, all slide images should be copied to the "Stimuli" subfolder. Their corresponding thumbnails are stored in the "Thumbs" subfolder in a single binary file. In addition, all items that are copied to the "SlideResources" subfolder can be used for slideshow creation when designing one's own experimental presentation.

2.2. Database layout. OGAMA stores the eye- and mouse-tracking data in SQL Express local data files that are created for each experiment. The underlying database is shown in Figure 1. The whole database is subdivided into six tables. The subjects table contains the list of subjects with their associated descriptive attributes, like age, sex, and category, and it also includes a field for comments. The trials table contains the meta-information of all trials. Information on each trial of the experiment for each subject is stored with trial-associated variables, like stimulus file names of the presentation slides, start time, trial duration, and category, along with columns for subjects' responses and correctness information. For each subject, a raw data table is created that contains the imported eye- and mouse-tracking data and the corresponding timing information. For each time stamp, this can include up to two columns for pupil diameter data (PupilDiaX, PupilDiaY), two columns for gaze position data (GazePosX, GazePosY), two columns for mouse position data (MousePosX, MousePosY), a column associated with subjects' responses (response), and a column for recorded Flash movie states (FlashState). If AOIs are defined during the analysis, they are stored in the AOI table. AOIs are defined to additionally analyze subregions in a stimulus image. For example, AOI information can be used to calculate fixation durations on a special feature/object of a slide. OGAMA provides the possibility
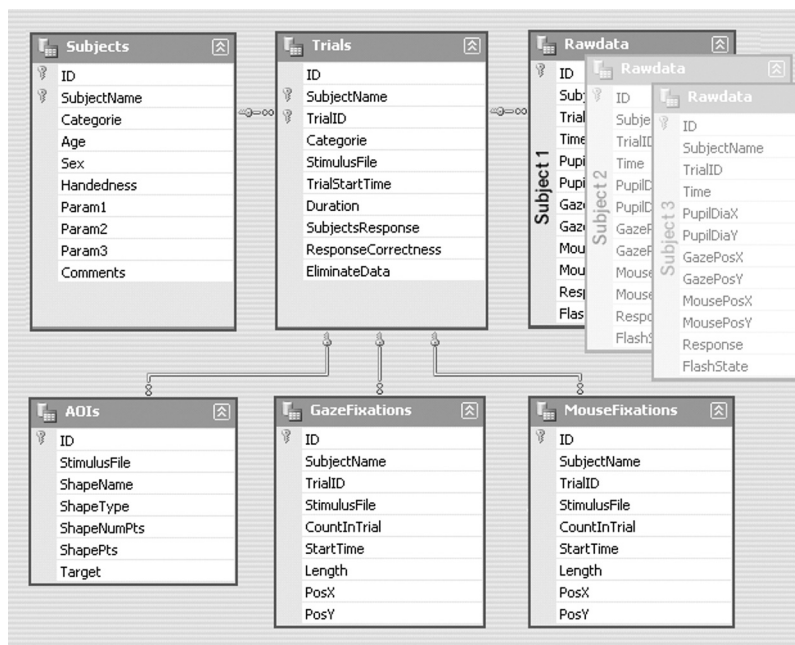

Figure 1. Database design of OGAMA's SQL database template file.

of creating rectangular, ellipsoid, and polygonal-shaped AOIs (see section 3.7). Each shape is stored together with a name (ShapeName), the owning image (ImageFile), its type (ShapeType), its location and size (ShapePts), and a column for AOI categorization (target).

Finally, two fixation tables list the gaze and mouse fixation information for each subject and for each trial with its associated variables: the number of the fixation in a trial (CountInTrial), the starting time (StartTime), the fixation duration (length), and the position (PosX, PosY).

All six tables reside in the experiments' locally stored SQL database file. Data integration is ensured over foreign key constraints between these tables, as is shown in Figure 1.

2.3. Fixation detection. The fixation calculation is done using the fixation detection algorithm published by LC technologies (www.eyegaze.com/doc/FixationSourcecode.htm). It is a dispersion-type algorithm with a moving window (Salvucci & Goldberg, 2000). We ported the source code to C# and added a time estimation support, but the working principle was not changed. This is described in the source code documentation:

The function detects fixations by looking for sequences of gaze-point measurements that remain relatively constant. If a new gaze point lies within a circular region around the running average of an ongoing fixation, the fixation is extended to include the new gaze point. (The radius of the acceptance circle

is the above user specified maximum distance.) To accommodate noisy measurements, a gaze point that exceeds the deviation threshold is included in an ongoing fixation if the subsequent gaze point returns to a position within the threshold. If a gaze point is not found, during a blink for example, a fixation is extended if

a) The next legitimate gaze point measurement falls within the acceptance circle, and

b) There are less than the above minimum numbers of samples of successive missed gaze points. Otherwise, the previous fixation is considered to end at the last good gaze point measurement.

(source code description, www.eyegaze.com/doc/FixationSourcecode.htm)

Prior to the fixation calculation, the fixation detection algorithm applies a two-step filter. In the first step, empty values are ignored in the calculation. This is necessary because of the different sampling rates of the gaze and mouse data. The second step is to omit samples with both x- and y-coordinates equal to 0, which often marks the eyetracker output during a blink, and to omit samples that lie out of the screen.

In the experimental settings, one has to specify two parameters for this algorithm: (1) the maximum distance (in pixels) that a point may vary from the average fixation point that is still considered to be a part of the fixation,
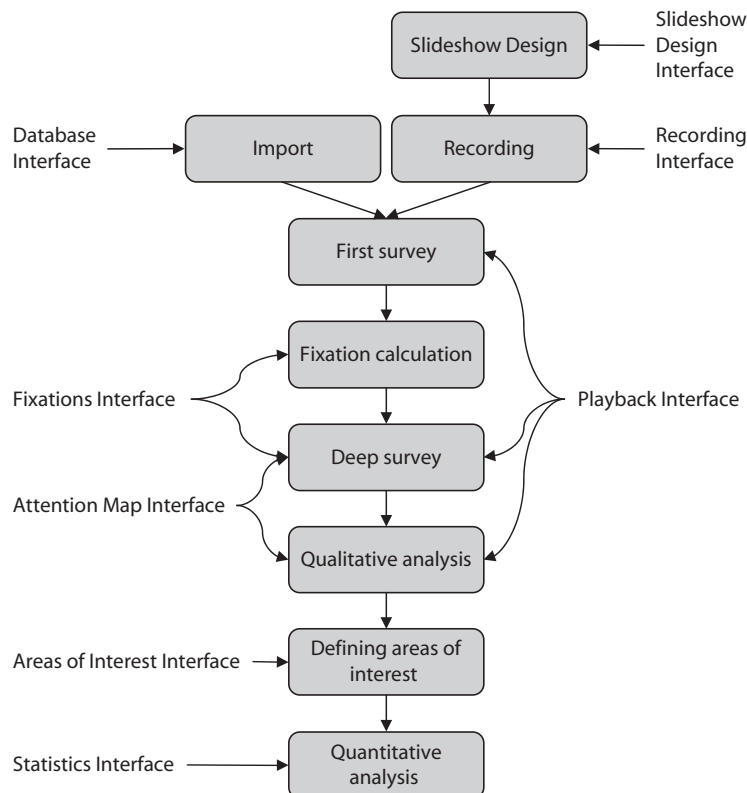

Figure 2. OGAMA's workflow with the supporting interfaces.

and (2) the minimum number of samples that defines a fixation. Depending on the experimental sampling rate and the research domain, these highly interdependent parameters have to be chosen carefully (see Karsh & Breitenbach, 1983, for a discussion).

3. Interfaces

This section contains a brief description of OGAMA's capabilities. When using OGAMA, we suggest that the user follows the subsequently described processing steps. OGAMA's workflow is depicted in Figure 2. A new project is started either by designing a slideshow presentation (see section 3.1) and recording new data (see section 3.2), or by importing new sampling data with OGAMA's import assistant (see section 3.3). Yet, regardless of whether either recording or importing succeeded, each step of the analyzing process is supported by one of the six different interfaces. Each interface is specialized on its intended purpose. The interfaces can be grouped into three sets for visual data analysis, which are replay (see section 3.4), fixations (section 3.5), and attention maps (section 3.6); two interfaces for statistical data analysis, which are AOIs (section 3.7) and statistics (section 3.8); and the database interface (section 3.9), which is used for controlling the underlying data. The interfaces reside in a main window that also hosts the menu, a toolstrip, and a context panel. The context panel contains lists of subjects and stimuli, along with an interface context help. By using the context panel, the user can easily select a stimulus and/or a subject for the analysis. Figures 3A–3H show exemplary screen shots of the interfaces created with the OGAMA demo project (available from the OGAMA Web site).

3.1. Slideshow design interface. The slideshow design interface enables the creation of slides to be presented during an experiment (Figure 3A). Five different stimulus

types are supported: verbal stimuli, image stimuli, shape stimuli, blank stimuli, and Flash movies. The newly created slide sequence is depicted in a preview window that also provides the possibility to easily modify all slide properties. Via double-clicking the mouse, the user opens a slide properties dialog consisting of the following five areas (the creation process is WYSIWIG; i.e., each modification is instantly shown in the preview window):

1. Stimulus creation: This section covers the properties needed to add a specific stimulus to the slide. For example, adding instructions to a slide is done by writing the instruction, choosing its style properties—such as font, size, and color—and clicking on the "add instruction" button. The position on the slide is then defined by dragging the mouse on the slide.

2. Timing: This is to define the conditions under which the slide should be exchanged. That can be after a specific amount of time (time stop condition), after a specific or arbitrary key response (key stop condition), or after a specific or arbitrary mouse click (mouse stop condition). Multiple stop conditions can be defined for one slide.

3. Background: An image or color can be assigned to the background of each slide.

4. Testing: Within this area, targets/responses can be defined (e.g., a target shape to click on or a valid key or mouse response) to specify a correct response associated with a slide. This enables OGAMA to track correct or incorrect answers that are stored in the response correctness column of the database.

5. Mouse: Starting positions and the visibility of the mouse cursor can be defined.

In addition to these five areas, slide names and categories/conditions can be entered before adding the created slide to the preview area. The slides are stored in xml format in the .oga file and can be exported to other OGAMA ex-
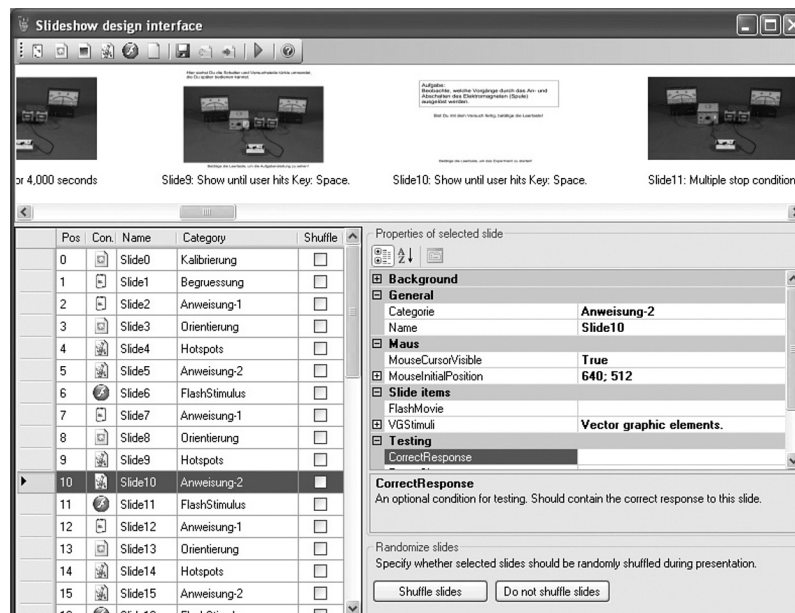

Figure 3A. Screenshot of OGAMA's slideshow design interface.

periments via a special xml slide format. Image files and flash movies are copied to the "SlideResources" folder of the experiment's main folder to provide independence from the Windows file system (e.g., when moving the experiment to another computer). When the slideshow is finished, all slide images are automatically copied to the "Stimuli" subfolder (see section 2.1).

3.2. Recording interface. The recording interface is designed to capture subjects' gaze and/or mouse movements during the presentation of the experimental slideshow (Figure 3B). Two different tracking systems are currently implemented: OGAMA is able to control and record from Tobii gaze-tracking systems (Tobii Technology), and OGAMA includes a software mouse-tracking device.

In general, four steps are required to start a recording:

1. Connect: Start the communication between the eyetracker and OGAMA.

2. Subject: Opens a dialog to enter subject information (e.g., identification code, age).

3. Calibration: Start the (selected) eyetracker calibration process.

4. Record: Start data recording and show slideshow on the presentation screen.

The settings associated with the selected tracking interface are edited via a customized dialog (e.g., specifying the trackers' IP Address, the number of calibration points, their speed, size, and color). At the end of the experimental procedure, the tracked data are stored in OGAMA's database and are immediately available for analysis or replay.

3.3. Data import. OGAMA's import assistant is designed to flexibly support data import from different data sources of various eyetracking systems. The data import is performed in six steps:

1. *Select the log file with the gaze and/or mouse samples*. OGAMA can read ASCII files with delimiter separated columns. The log file should contain the whole raw data samples from the recording.

2. *Specify the parsing conditions*. To successfully read the log file, one has to specify the column delimiter (e.g., a semicolon or a tab) and the string that identifies quotes that should be ignored by the import assistant. It is also possible to skip lines from the log file that don't start with a number. In addition, the user can select whether the first readable row contains the column names. After each modification, a preview of the log file with the detected columns is shown in a table, where the column names can be modified.

3. *Assign columns*. To guarantee a matching between the import file columns from different sources and OGAMA's internal database columns, the next step is to assign the relevant columns of the log file to those of OGAMA. For example, assume a "Tobii 1750" EFD log file with the columns "Timestamp," "Found," "GazepointX," and "GazepointY." In a dialog, the correct assignment is done by selecting the log file columns from the drop-down buttons. If no subject name column is specified, then it is assumed that the whole data file is from the same subject, and a dialog asks for a unique subject name.

4. *Specify how to distinguish different trials of the experiment*. OGAMA's import assistant provides three different trial specification modes:

(a) In each raw data row, a column with the current trial number is present (a column with an incremental trigger—e.g., iViewX System, SMI).

(b) The trial-separating rows contain a unique string like "MSG" (iViewX System, SMI) or "TRIALID" (EyeLink System, SR research).

(c) There is a table with starting time and trial ID columns. This can be imported from an additional trial definition file (Tobii 1750 EVD file) or entered manually.

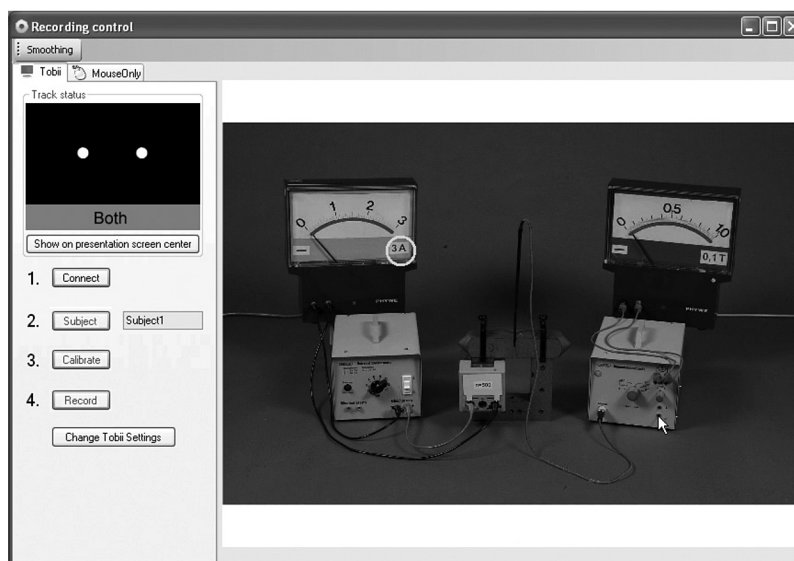In this step, the user is also prompted to specify the time code unit.

Figure 3B. Screenshot of OGAMA's recording interface.

5. *Specify how to detect or read the stimulus images for each trial*. In the last step, one has to extract the stimulus image file names from the log file to assign them to the imported trial numbers. This step works similarly to the trial import (see Step 4).

Every change is previewed in a data grid view, so the settings can be controlled instantly.

6. *Start importing*. To facilitate the import of similarly structured log files, the user is asked to save the import settings to a file that can be used at the start of the next import. During the data import, all raw data (including 0 values or values that are out of screen) are wrapped into OGAMA's database, which enables a customized handling in the fixation detection algorithm and visualization.

3.4. Replay interface. The replay interface is designed to replay the data logged during an experiment (Figure 3C). Its intended use is to easily visualize and record single subject gaze and mouse-tracking segments. To support this use, different drawing modes are available for gaze and mouse-tracking data (or both). Depending on visualization needs, the drawing modes can be combined arbitrarily.

For the replay of the sampling data, the user can select the following modes:

1. Cursor: Draw a cursor at the current sample point.

2. Path: Draw a polyline consisting of the sample points up to the current sample point.

3. Fixations: Draw circles that represent fixation position and fixation duration.

4. Connections: Draw straight lines that connect the fixation circles.

5. Spotlight: Show the current sample point position as a circle of the original image overlaid on the grayed stimulus slide.

To replay a certain tracking segment, a specific subject and trial can be selected from drop-down controls. Further controls provide the possibility to change the replay speed (real-time, faster, or slower); the drawing modes; the shape, size, and color of the cursors; and the pen styles of all drawing elements. The timing control is achieved by both a time slider and "play," "pause," "stop," and "reset" buttons. Video recording is done via the "record" button. The recorded .avi files with custom size and compression are saved on hard disk. The replay interface is a stand-alone tool that calculates fixation data independently of precalculated fixations (see section 3.5) in real time for the selected trial. These allow a quick check of fixation detection parameters (see section 2.3).

3.5. Fixations interface. This fixations interface is designed to calculate, store, and display the gaze and mouse fixations made by the subjects (see Figure 3D). It is recommended that one calculate the fixations for all subjects right after data import. These are stored in OGAMA's database and are available afterward for the attention map interface and the statistics interface. Note that a new calculation of the fixations is recommended if one changes or adds raw data.

The fixation interface contains a canvas for visualization and a data table for each gaze and mouse fixation. There are six display modes available:

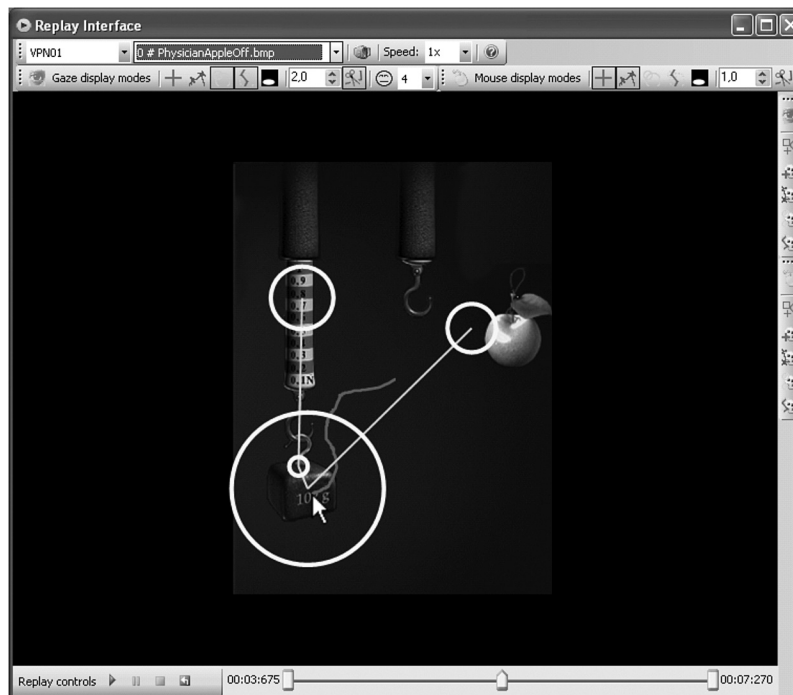1. Connected dots: Draw the fixation number and position connected by small lines.

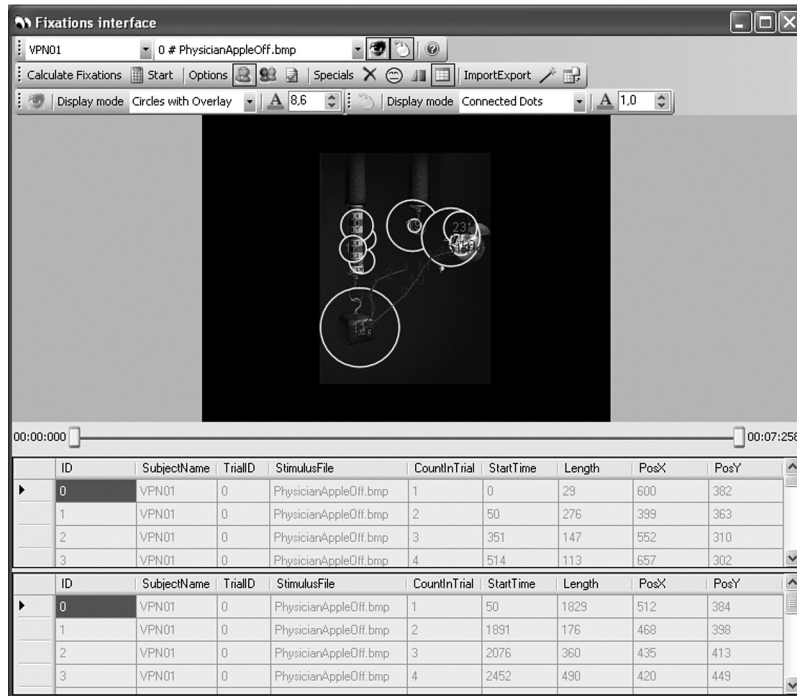**Figure 3C. Screenshot of OGAMA's replay interface.**

Figure 3D. Screenshot of OGAMA's fixations interface.

2. Connected circles: Draw the fixation number and position connected by small lines with a circle size that is proportional to the fixation duration.

3. Circles fast: Draw the fixation number and position with a circle size that is proportional to the fixation duration.

4. Circles with overlay: Draw the fixation number and position with a circle size that is proportional to the fixation duration. Fixations are overlaid by subsequent fixation circles.

5. Spotlight: Draw the fixation number and position as a circle of the original image overlaid on the grayed stimulus slide.

6. Attention map: Draw fixations as a colored heat map. We implemented heat maps as superimposed Gaussian distributions of all fixations in a trial (see section 3.6 for a description of the algorithm).

The data table lists the number of the fixation, the calculated starting time, the duration, and the position of each fixation for the current trial. Trials can be selected by subject name and trial number from a drop-down control. To facilitate analysis, defined AOIs (see section 3.7) can be displayed. It should be noted that the fixation detection parameters are fixed for the fixation calculation of a whole experiment, but they can be edited in the experiment's "options" dialog of the main window.

3.6. Attention maps interface. The attention maps interface is designed to visualize merged fixation data from selected subjects (see Figure 3E). This can be used to identify regions of special attention in a specific stimulus slide. In addition to the attention maps computed in the fixation interface, the attention maps interface allows

one to build averaged maps across a subset of some (or all) subjects.

The attention maps are calculated as aggregated Gaussian distributions of each fixation in a stimulus slide. The summed Gaussian distributions are then overlaid on the original stimulus slide. Doing this allows a quick visualization of a landscape of visited and unvisited locations on the stimulus slide (Spakov & Miniotas, 2007; see also Figure 3E). The attention map calculation uses a two-dimensional Gaussian kernel with an editable size to enable adjustable smoothing of fixation distribution landscapes. With a mean value of $\mu = 0$, a standard deviation of σ, and an isotropic distribution, the Gaussian kernel is given by the following formula:

$$f(x, y) = \frac{e^{-\frac{x^2 + y^2}{2\sigma^2}}}{2\pi\sigma^2} \, x, y \in \left[-s, s\right].$$

The standard deviation is internally set to $\sigma = s/5$. Each value of this template kernel is multiplied for each fixation with a factor that is given by the duration of the fixation. This factor represents the weighting of this fixation. All multiplied kernels in a trial are then added to a stimulus-sized array at the positions of the respective fixations. Finally, the whole array is normalized. The result is a stimulus-sized height map or landscape (see, e.g., Spakov & Miniotas, 2007). This landscape can be easily transformed into a color map by a predefined or custom gradient. Figure 3C shows an example image that uses a "traffic sign" gradient in which the hot spot is overlaid in red. Another predefined gradient is called *black mask*, and it hides unseen areas of the original image with a black mask.
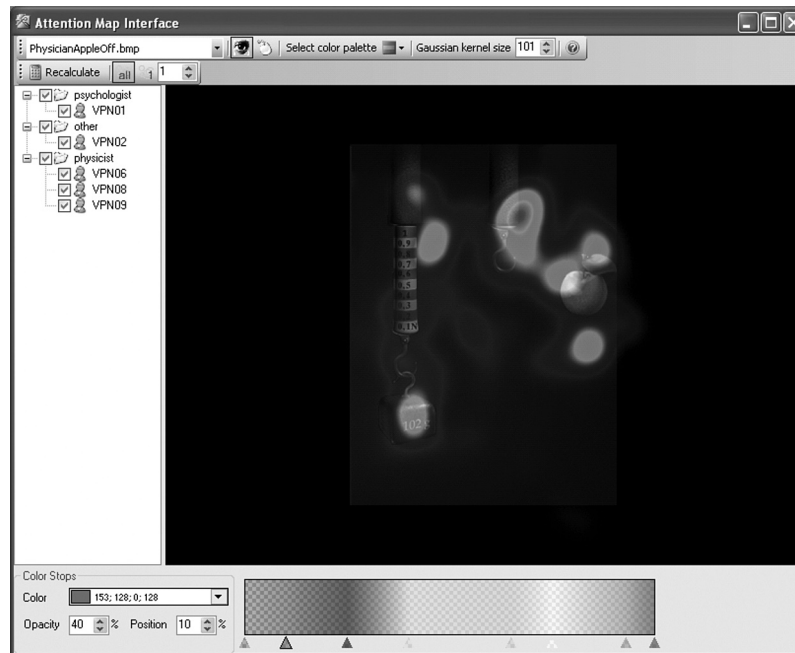
Figure 3E. Screenshot of OGAMA's attention maps interface.

The user can choose the subjects to include in the calculation along with whether the calculation should be based only on the first or on all fixations. Thus, one can easily answer questions like "Where does the average subject have its first look?" and "What locations of the stimulus are not noticed by the average subject?"

3.7. AOI. The AOI interface is designed to define and display different AOIs (see Figure 3F). Three types of AOI shapes are available: simple rectangles, ellipses, and polygons. After selecting the type of the AOI, one specifies location and the size of the AOI with the mouse on the stimulus slides. Shape and size can still be modified in later processing stages, with the help of grab handles.

The AOI data are stored in a separate database table together with the stimulus file name they belong to. The database's entries are shown in a data table in which editing is allowed for either renaming and fine tuning or deleting the whole AOI. In the data table, the AOIs can be categorized by grouping them with specific names. These categories are used in the statistical output. Predefined values are "Target" and "SearchRect," which are used, for example, to generate variables like "time left until first fixation in 'target' AOI" or "number of fixations in 'SearchRect.'"

Similar to the foregoing interfaces, the user can change the pen style to allow visible differentiation. Furthermore, the AOI interface provides the possibility of importing a list of rectangular AOIs from a text file through an import assistant similar to that described in section 3.3.

3.8. Statistics interface. The statistics interface is designed to calculate empirical parameters that are useful for further analyses (see Figure 3G). There are four different scopes of variables available: subject parameters, trial parameters, gaze parameters, and mouse parameters.

The parameters are marked for calculation by selecting checkboxes. Once this is finished, the parameters are calculated and submitted to a results data table. This table can be exported to an ASCII-file, which is importable in postprocessing software, like SAS or SPSS.

Subject parameters. Subject parameters are objective parameters to describe the subject's properties. These predefined variables refer to the subject table of OGAMA's database. In detail, there are the unique subject name (e.g., a subject ID, "AZ012"), a subject category (e.g., "physicist"), age, sex, handedness, three undefined parameters that can be assigned by the user (called "param1," "param2," "param3"), and a comments column for additional notes.

Trail parameters. Trial parameters categorize the different trials. Trial parameters are split in four sections: "name and timing," "areas of interest," "loss of data," and "further options."

The section "name and timing" consists of the unique trial ID, the stimulus slide file name, a descriptive stimulus category (which can be used for grouping trials, like "first seen" or "second visit"), the trial duration, and the starting time (in milliseconds).

The "areas-of-interest" section contains Targetsize, the relative size of the "target" AOI as a percentage (in relation to the total screen size). This size can be used for calculation of Fitts's law (Accot & Zhai, 2003). "Areas of interest" further contains the AOISize, which gives the relative size of all AOIs as a percentage (relative to the screen size).

The third section, "loss of data," provides a count of the data lost due to blinks and out-of-monitor samples.

The "further options" section contains three columns. The first contains subjects' responses recorded during the
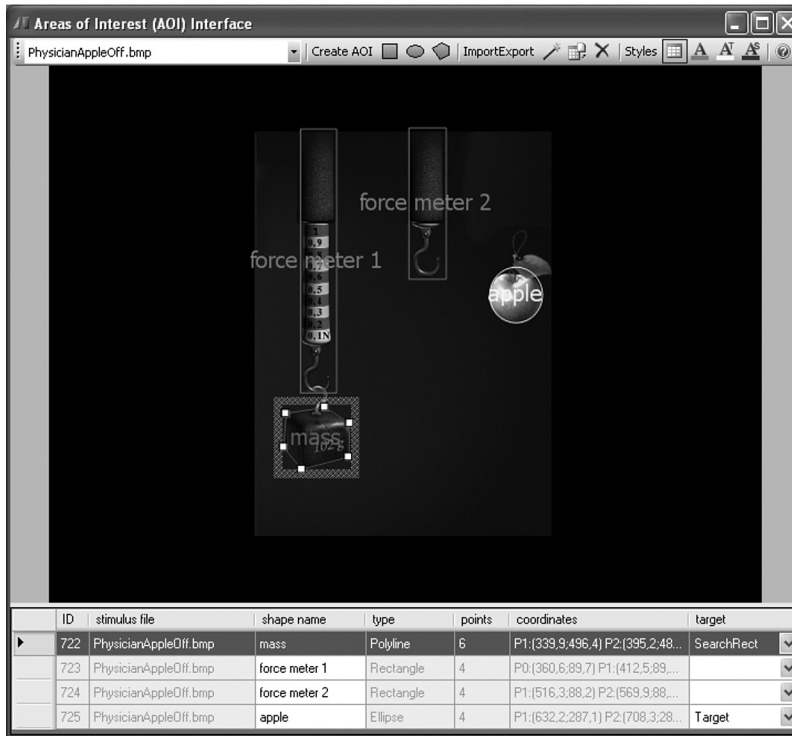
Figure 3F. Screenshot of OGAMA's AOI interface.

experiment; the second is an optional parameter (e.g., for correctness data); and the third column indicates manually excluded trials. The latter column contains "yes" values when the trial is manually marked for exclusion in the fixations interface.

Gaze parameters. Gaze parameters are split in four sections: "fixation general," "fixation timing," "gaze path," and "gaze mouse interaction" (see Figure 3G). General fixation parameters in the "fixation general" section include the number of fixations, the average fixation dura-
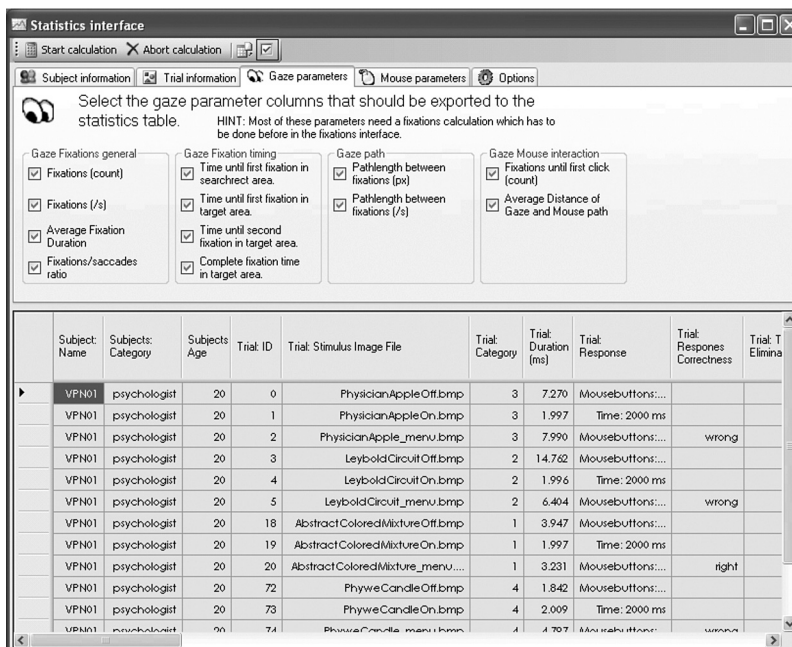

Figure 3G. Screenshot of OGAMA's statistics interface.

tion, and the fixations–saccades ratio. The term *fixation timing* refers to categorized AOIs. This section evaluates the time until the subject first fixated the AOI referred to as "target" or "SearchRect," and the amount of time that the subject fixated the "target" AOI at all. Additionally, it includes the calculation of the time until the second fixation at the target AOI.

The term *gaze path* refers to the gaze path length in a trial. It is calculated by adding the lengths of the fixation connections to exclude microsaccades and data collection noise. The user can select between two output parameters: absolute gaze path length in pixels and gaze path velocity in pixels/second.

The last section, "gaze mouse interaction," contains two parameters. The first reports the number of fixations until the first mouse click, and the second calculates the average distance between gaze and mouse path.

Mouse parameters. In the "mouse parameters" tab of the statistics interface, the gaze parameters are also calculated for mouse movements (except the average gaze and mouse path distance). Additionally, the user can select the evaluation of click counts and click rates for the left and right mouse buttons and the time until the first mouse click event.

3.9. Database interface. The database interface is designed to manually view and edit the subjects table, trials table, and all raw data tables with sampling data (see Figure 3H). The user can edit and revise each database entry. Through the database interface, the import assis-

tant for importing custom sample data can be started (see section 3.3).

4. Example Study[1]

As an illustrative example, we will provide the data of an experimental study from usability research in physics education. In physics education, we are interested in how students perceive and process multimedia representations of real physics experiments (Kirstein & Nordmeier, 2007). What properties of these multimedia representations support learning of physics? In particular, recent research has focused on so-called "interactive screen experiments." In the present experiment, the learner can directly manipulate the experimental setup of physics experiments in multimedia systems within a sequence of digital photos that originate from the conduction of the actual experiment. Within the ISE approach, multimedia are used to create a realistic image from a real experiment in which the learner is able to make her or his own findings. It should be noted, however, that ISEs have limited manipulation possibilities in comparison with real-world experiments. In this particular study, we were interested in the effect that different representational formats of the experimental setups in ISEs have on the way subjects solve basic physics problems. In physics education, there is a persistent discussion on whether formal and abstract representations of physics knowledge support the learning due to its reduced complexity (Lenzner, Müller, Horz, & Schnotz, 2008). Recent developments in commercial learning software are in favor of more realistic or even
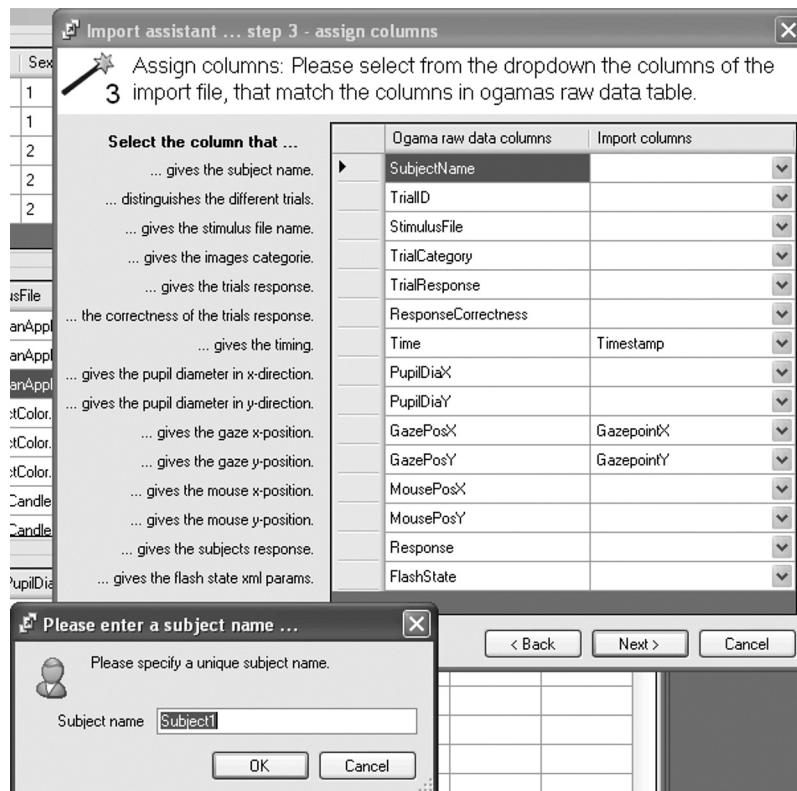

Figure 3H. Screenshot of OGAMA's database interface.

photorealistic representations. Mainly, these products differ from classical physics teaching materials in that their material is reduced in complexity (to direct the learners' attention to the crucial objects), and they are particularly suitable for children (for an example, see Figure 3C, which is a part of the physics learning game, *Physicus*).

This pilot study was designed to compare the effects of different representational formats in ISEs. Is it possible to find evidence that representations with reduced complexity like that of abstract formats and that of modern learning software help the user to solve basic physics problems? Is this advantage of representations with reduced complexity mirrored in the subjects' eye movements? If so, then we expected to observe shorter fixation durations associated with complexity-reduced representation formats as an index of facilitated processing during the solving of basic physics problems (Rayner, 1998). In addition, in an exploratory manner, we tracked subjects' mouse-position data. To illustrate the capabilities of OGAMA, two mouse parameters are reported here: the "time until mouse over target" and the "average distance between gaze and mouse path" (see section 3.8).

4.1. Method. A total of 32 subjects (22 female, mean age = 27.7 years) participated in this study. All had normal or corrected-to-normal vision. The stimulus material consisted of 52 photos of 13 basic physics experimental setups in four different representational formats that ranged on a dimension from low complexity to high complexity (see Figure 4): (1) an "abstract" representation consisting of formal line drawings, (2) photorealistic and complexity-reduced snapshots of experimental setups taken from the physics learning game *Physicus*, and (3) and (4) photos of representative experimental setup from two different standard German teaching material companies (Leybold, Phywe).

When making these photos of the same experimental setups, we took special care to be as comparable as possible across the different representational formats, thereby covering different areas of physics, such as optics and thermodynamics. For example, one way to control the comparability across the different formats was to present the same number of objects in a similar spatial order.

A training trial and all 52 experimental setups were randomly presented on a 17-in. computer screen, and the subjects were asked to solve basic physics problems (e.g., "Turn the light on"). For every problem, there was exactly one "interactive" possibility to solve it by clicking with the left mouse button on the correct object. When the subject did so, the manipulated object changed in the intended manner. For reasons of standardization, no other object was animated in this study. Subjects were given no time restriction to solve a problem. We recorded the subjects' eye and mouse data while they were solving the problems. Subjects were seated with their head stabilized in a chinrest (approximately 0.6-m distance between their eyes and the computer screen). An infrared-sensitive camera recorded gaze position at a sampling rate of 240 Hz using a video-based iView X Hi-Speed eyetracker (Senso-Motoric Instruments, Teltow, Germany). Presentation 10.0 software (Neurobehavioral Systems Inc.) was used to synchronize stimulus presentation and the eyetracker, and to record mouse events and mouse-position data. Eye- and mouse-position data were analyzed using OGAMA. AOIs were defined as rectangular or polygonal shapes covering the contour of the target object that solved the task. The statistical analysis was done using SPSS, computing repeated measures ANOVAs at a predefined significance level ($\alpha = 0.05$).

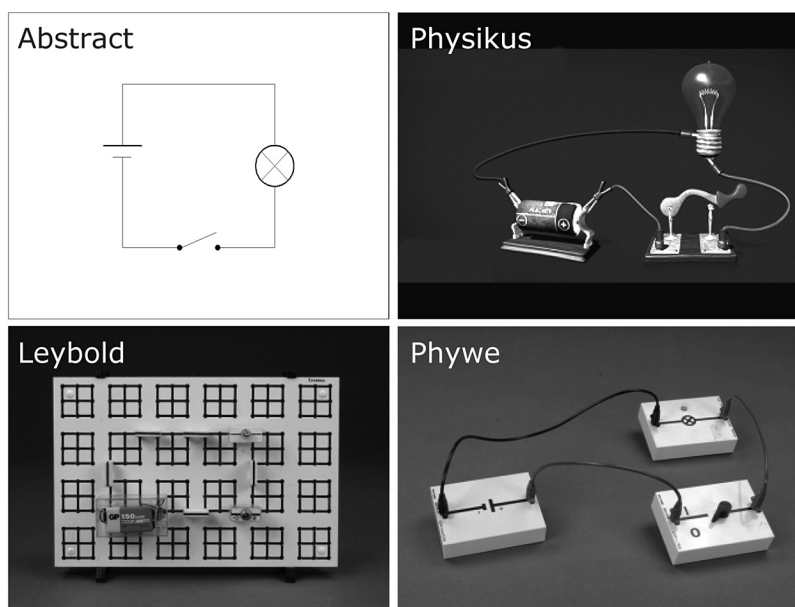**4.2. Results and Discussion**. Following artifact detection, 6.07% of all trials had to be discarded because

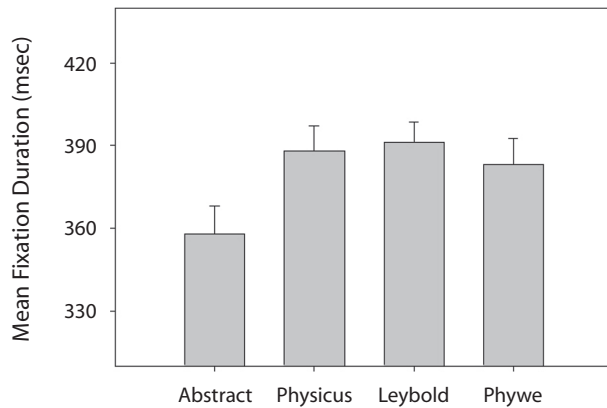**Figure 4. Four different representations of a physical experiment.**

Figure 5. Results of mean fixation duration.

of excessive blinking or decalibration during a trial. An analysis of mean fixation durations revealed a significant main effect of representational format [$F(3,93) = 9.915$, $p < .001$, $\eta^2 = .242$]. Follow-up paired t tests showed that abstract representations differed from all other conditions (all ps $< .003$). No other comparison reached significance. Thus, in line with our initial hypothesis, subjects had shorter fixation durations when confronted with formal abstract representations of physics experiments (see Figure 5). With this finding, it seems probable that abstract representations of physics experiments are processed faster than real-world-like scenarios. It is obvious that the formal and complexity-reduced setup of abstract representations facilitate the perception of this basic physics problem (Rayner, 1998).

For a first analysis of mouse-position data, the variable "time until mouse over target" was calculated. The repeated measures ANOVA revealed a significant main effect for representational format on the variable "mouse first time over target" [$F(3,93) = 3.565$, $p = .017$, $\eta^2 = .242$]. Again, the follow-up paired t tests revealed significant differences between abstract representations and the other three conditions (all ps $< .049$), whereas no other comparison reached significance (see Figure 6). As with
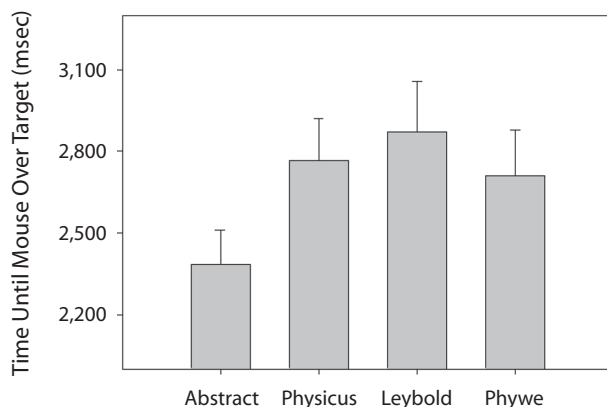
the fixation duration analysis, an advantage for processing abstract representations was observed for the variable "time until mouse over target." Thus, subjects were faster in moving the mouse over the target object when a formal and complexity-reduced representation was presented. We attribute this result, which is analogous to the fixation durations, to facilitated processing. Still, the possibility appears that fixation durations and the variable "time until mouse over target" represent different aspects of underlying cognitive processing (Chen et al., 2001). Although in this study both variables act in parallel, one might speculate whether the variable "fixation durations" focuses on perceptual processing (see, e.g., Unema, Pannasch, Joos, & Velichkovsky, 2005), whereas the variable "mouse over target times" might be a parameter more closely related to decision making (Magnuson, 2005).

When analyzing the average distance between gaze and mouse path, the repeated measures ANOVA revealed a significant main effect of representational format [$F(3,93) = 15.712$, $p < .001$, $\eta^2 = .336$]. Here, the follow-up t tests revealed a different pattern: Representations taken from the learning game *Physicus* differed significantly from all other conditions (all ps $< .001$), and the other representational formats did not differ significantly (see Figure 7). Investigating this variable, the photorealistic representations led to a smaller average distance between mouse and gaze path, whereas the other representations did not show this strong connection. It is probable that the representational format of the images elicits different strategies in the subjects that are reflected in the average gaze mouse distance (Byrne, Anderson, Douglass, & Matessa, 1999; Chen et al., 2001). One should consider the type of images used in the *Physicus* game. These images are designed to encourage the operating subject to directly manipulate the physical content. Since we cannot prove this speculation on the basis of the present pilot study, an evaluation of these results should be addressed to future research in physics education.

Still, we believe that measuring and analyzing gaze and mouse data in parallel might give new insights not only in the processes involved in solving physics problems, but also more generally in usability or psychological research (Magnuson, 2005).
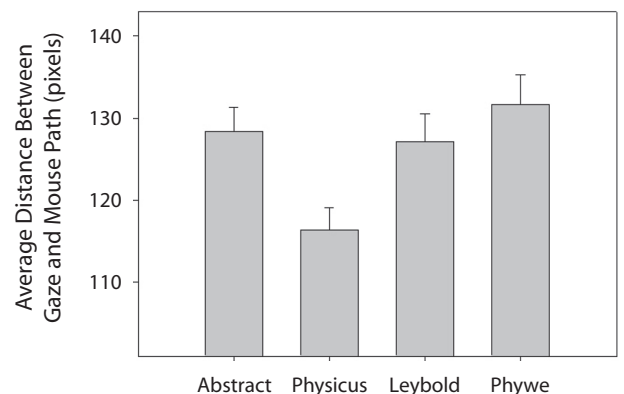

Figure 6. Results from testing the variable "time until mouse over target."

Figure 7. Results from testing the variable "average distance between gaze and mouse path."

5. Conclusion

We introduce OGAMA to expand the research capabilities in screen-based eyetracking studies. OGAMA is a stand-alone application running on Microsoft Windows platforms and is designed to handle data from a wide range of experiments in psychology and usability research. Thus, it is optimized for the analysis of slideshow-based stimulus presentations but is not suitable for the analysis of movie stimuli. It handles mouse movements just the same as gaze movements and has the ability to calculate connecting parameters between mouse and gaze data. It also supports most of the commonly used qualitative analysis and visualization tools in the field of eyetracking research, like replay, attention maps, and AOIs, along with tools for a quantitative look at the data.

Researchers from the community are encouraged to further develop their own statistical variables by adapting OGAMA's source code. To support this process, we provide OGAMA with an extensive source documentation. Its open-source character allows also the development of custom interfaces—for example, a specialized reading analysis interface or a salience map calculation interface. At this point, OGAMA is flexible and powerful software, and we would be happy to see it as a useful tool in the eyetracking research community. OGAMA is available along with an extensive demo project at http://didaktik.physik.fu-berlin.de/ogama.

REFERENCES

Accot, J., & Zhai, S. (2003). Refining Fitts' law models for bivariate pointing. In *Proceedings of ACM CHI 2003 Conference on Human Factors in Computing Systems* (pp. 193-200). New York: ACM Press.

Byrne, M. D., Anderson, J. R., Douglass, S., & Matessa, M. (1999). Eye tracking the visual search of click-down menus. In *Human factors in computing systems: Proceedings of CHI '99* (pp. 402-409). New York: Addison-Wesley.

Chen, M.-C., Anderson, J. R., & Sohn, M.-H. (2001). What can a mouse cursor tell us more? Correlation of eye/mouse movements on Web browsing. In *CHI '01 extended abstracts on human factors in computing systems* (pp. 281-282). New York: ACM Press.

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

Gitelman, D. R. (2002). ILAB: A program for postexperimental eye movement analysis. *Behavioral Research Methods, Instruments, & Computers*, **34**, 605-612.

Gitelman, D. R., Parrish, T. B., Friston, K. J., & Mesulam, M.-M. (2002). Functional anatomy of visual search: Regional segregations within the frontal eye fields and effective connectivity of the superior colliculus. *NeuroImage*, **15**, 970-982.

Goldberg, J. H., & Kotval, X. P. (1998). Eye movement-based evaluation of the computer interface. In S. K. Kumar (Ed.), *Advances in occupational ergonomics and safety* (pp. 529-532). Amsterdam: IOS Press.

Hutzler, F., Kronbichler, M., Jacobs, A. M., & Wimmer, H. (2006). Perhaps correlational but not causal: No effect of dyslexic readers' magnocellular system on their eye movements during reading. *Neuropsychologia*, **44**, 637-648.

Karsh, R., & Breitenbach, F. W. (1983). Looking at looking: The amorphous fixation measure. In R. Groner, C. Menz, D. Fisher, & R. A. Monty (Eds.), *Eye movements and psychological functions: International views* (pp. 53-64). Hillsdale, NJ: Erlbaum.

Kirstein, J., & Nordmeier, V. (2007). Multimedia representation of experiments in physics. *European Journal of Physics*, **28**, S115-S126.

Lenzner, A., Müller, A., Horz, H., & Schnotz, W. (2008). Dekorative und instruktionale Bilder in Lerntexten des Physikunterrichtes [Decorative and instructive pictures in learning texts in physics education]. In V. Nordmeier (Ed.), *Didaktik der Physik. Beiträge zur Frühjahrstagung der DPG—Berlin 2008*. [CD zur Frühjahrstagung des Fachverbandes Didaktik der Physik in der Deutschen Physikalischen Gesellschaft]. Berlin: Lehmanns Media.

Magnuson, J. (2005). Moving hand reveals dynamics of thought. *Proceedings of the National Academy of Sciences*, **102**, 9995-9996.

Nielsen, J., (2004). *Usability engineering*. San Diego: Kaufmann.

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

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

Shneiderman, B., & Plaisant, C. (2004). *Designing the user interface* (4th ed.). Boston: Addison-Wesley.

Spakov, O., & Miniotas, D. (2007). Visualization of eye gaze data using heat maps. *Electronica & Electrical Engineering*, **2**, 55-58.

Tan, P. N., Steinbach, M., & Kumar, V. (2006). *Introduction to data mining*. Boston: Addison-Wesley.

Unema, P., Pannasch, S., Joos, M., & Velichkovsky, B. M. (2005). Time-course of information processing during scene perception: The relationship between saccade amplitude and fixation duration. *Visual Cognition*, **12**, 473-494.

NOTE

1. Most of the figures presented in this article are images taken from this pilot study.