AUGMENTING USABILITY WITH EYE MOVEMENT PROCESS METRICS

CHI 2008 Course #14

Supplemental Student Guide

Amy Alberts, Ed Cutrell, Andrew Duchowski

Index

TOBII STUDIO: A PRIMER

In this brief section, Studio's main features and concepts are introduced. The basic workflow concept of Studio is generally a three-step process: (1) design a study and record eye movements, (2) play back recorded scanpath, and (3) visualize the data. Generally, one creates a project composed of at least one test. Each test consists of a number of visual stimuli over which eye movements are captured. Stimuli may include still images, video clips, web pages, or the computer screen (desktop). Stimulus presentation may be *self-paced*, i.e., terminated by the user, or *machine-paced*, i.e., ending after a experimenter-defined timeout period. Examples of such tests are given in this student guide. It is also possible to insert screen-based user instruction (text) among the stimuli, also known as*media elements*.

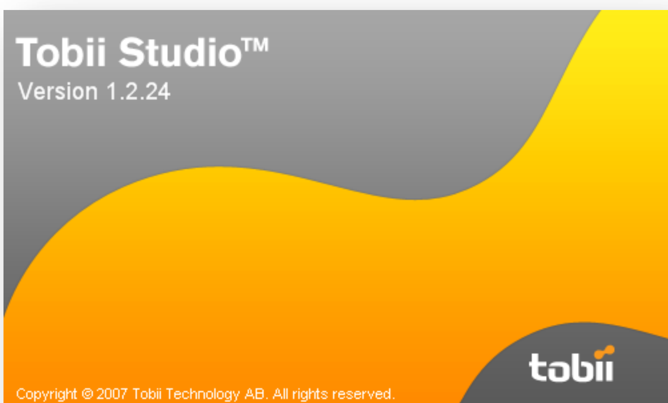
Studio maintains a database of participants, generally populated by the experimenter when participant data is available. Careful design of experiments determines which stimuli are shown to which participants, how many times, and in what order. A participant's eye movement recording can be played back immediately after recording.

Meaningful insights of attentional properties of the stimuli and/or of viewers' behavior may be made from various visualizations offered by Studio. Of these, gaze plots and heat maps offer single-image views of aggregate scanpath data distilled from sets of participants' eye movements over selected sets of stimuli. Areas Of Interest, or AOIs, may be defined by the experimenter to provide more quantitative views of eye movement metrics over specific regions in the imagery.

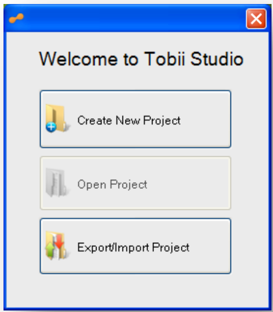
For subsequent dissemination and robust analysis, Studio offers means of exporting the captured data, either as images (e.g., hot spots), video, or as data files (e.g., for spreadsheet or other forms of statistical analysis).

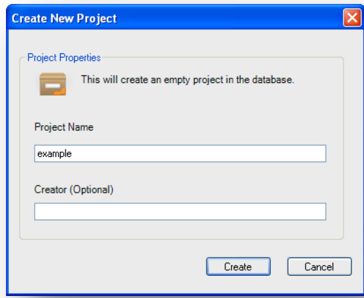
STUDIO

To explore the user interface, start Tobii's Studio, either via the Start menu or by double-clicking the Tobii icon.



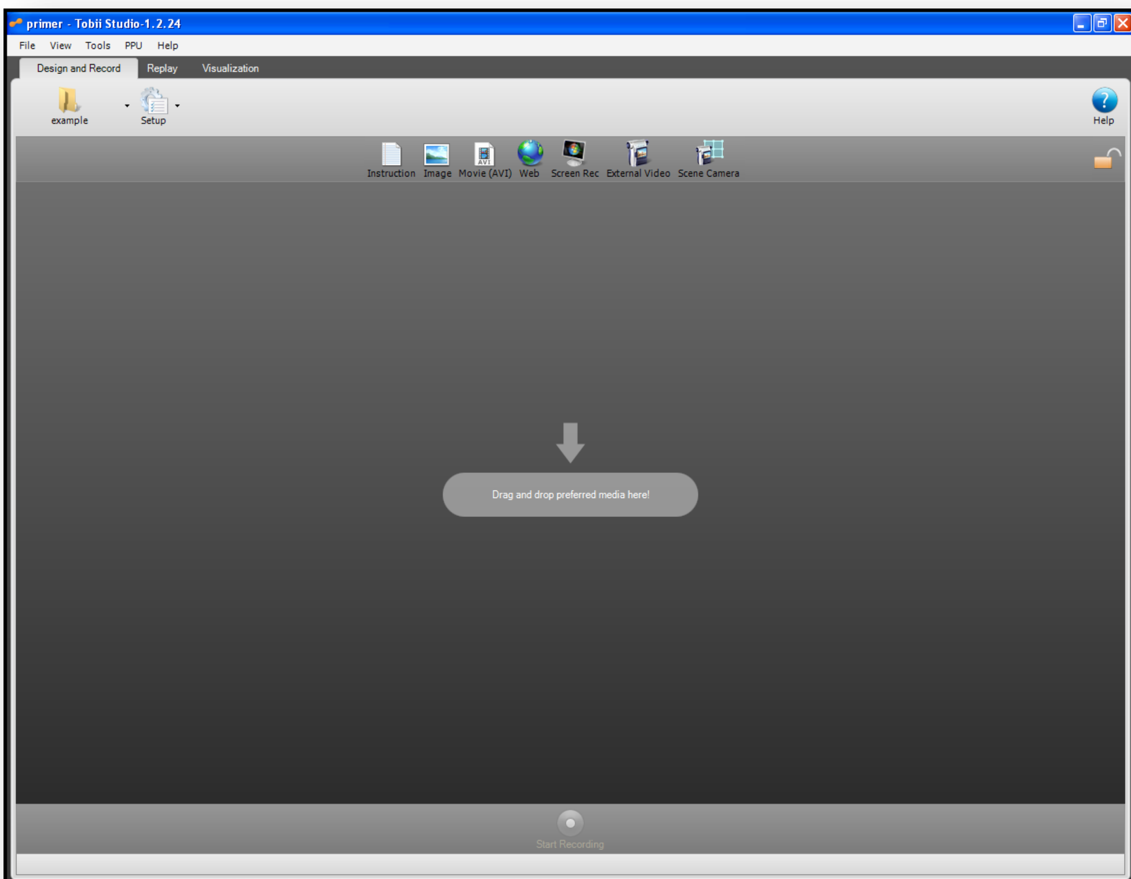
If no projects have yet been created, the **Open Project** option will be grayed out from Tobii's **Welcome to Tobii Studio** dialog.





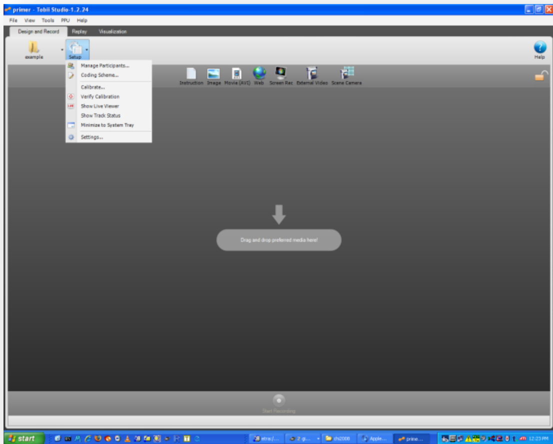
Create a new project, providing a new project name and (optional) creator identity. Studio's database is made up of projects where each project contains at least one test. Each test, in turn, is made up of an instance of a visual stimulus such as a still image, a video clip, a web page, or screen recording (desktop), or a sequence of the former types of stimuli. Once you provide a project name, Studio will automatically create an instance of an empty test, providing you with a chance to name that test (initially the default test takes the name of the project).

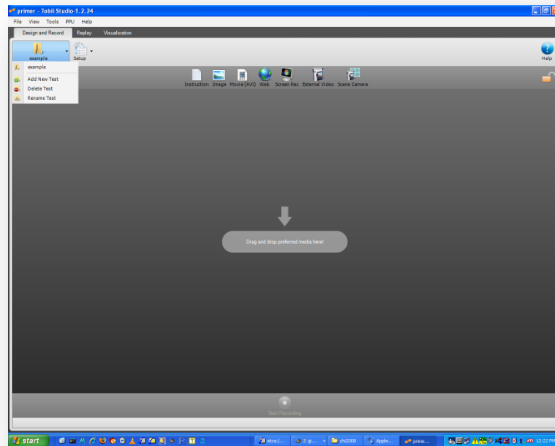
Studio's main window is split among its three main tasks via tabbed windows, with tabs labelled **Design and Record**, **Replay**, and **Visualization**.



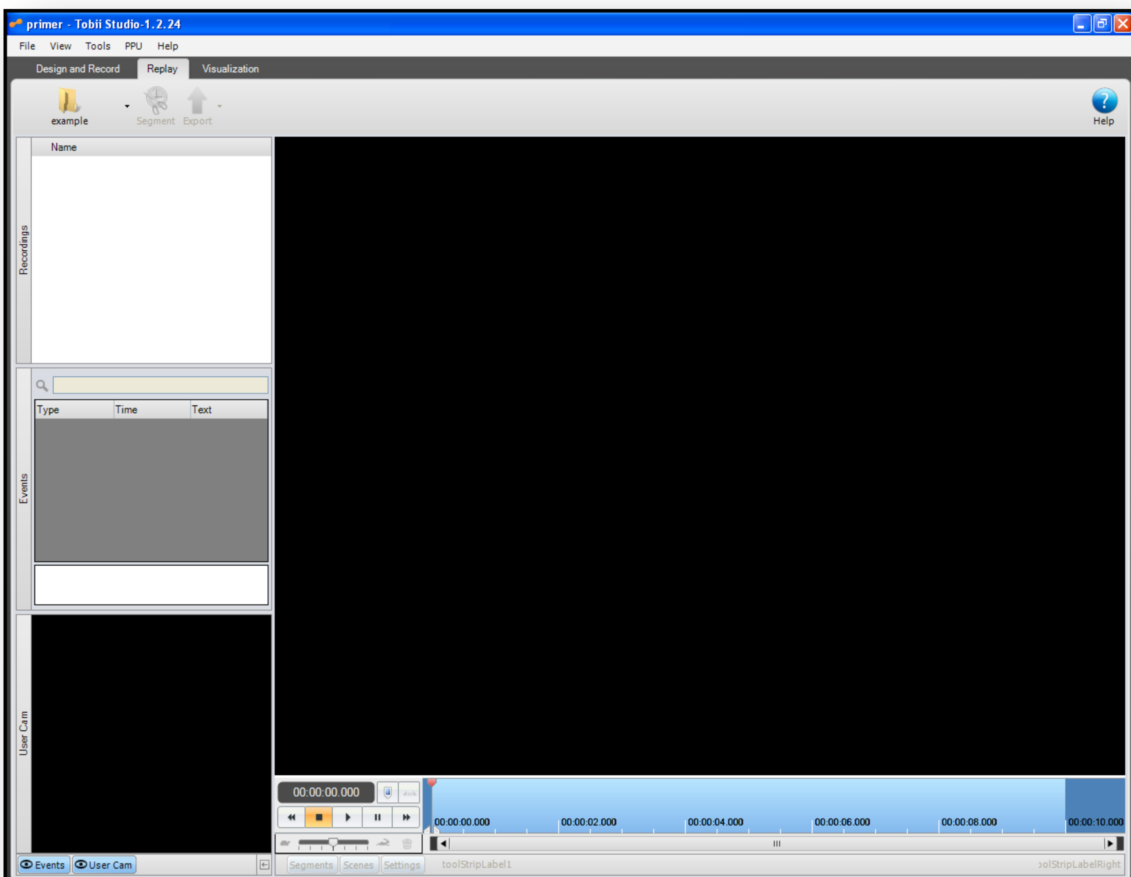
The **Design and Record** tab allows assembly of stimulus media elements (e.g., images, movie clips, web pages, etc.), that are dragged and dropped into the central window region, forming a timeline of events to be viewed by the study participants. The top menus, **Test** and **Setup**, indicate which test is presently being developed and what setup options are available, resp.

Note that the Test menu can take on the name of the first test contained therein. The Test menu allows creation, deletion, and renaming of tests, while the Setup menu provides control over the participant database, calibration parameters, and other options.



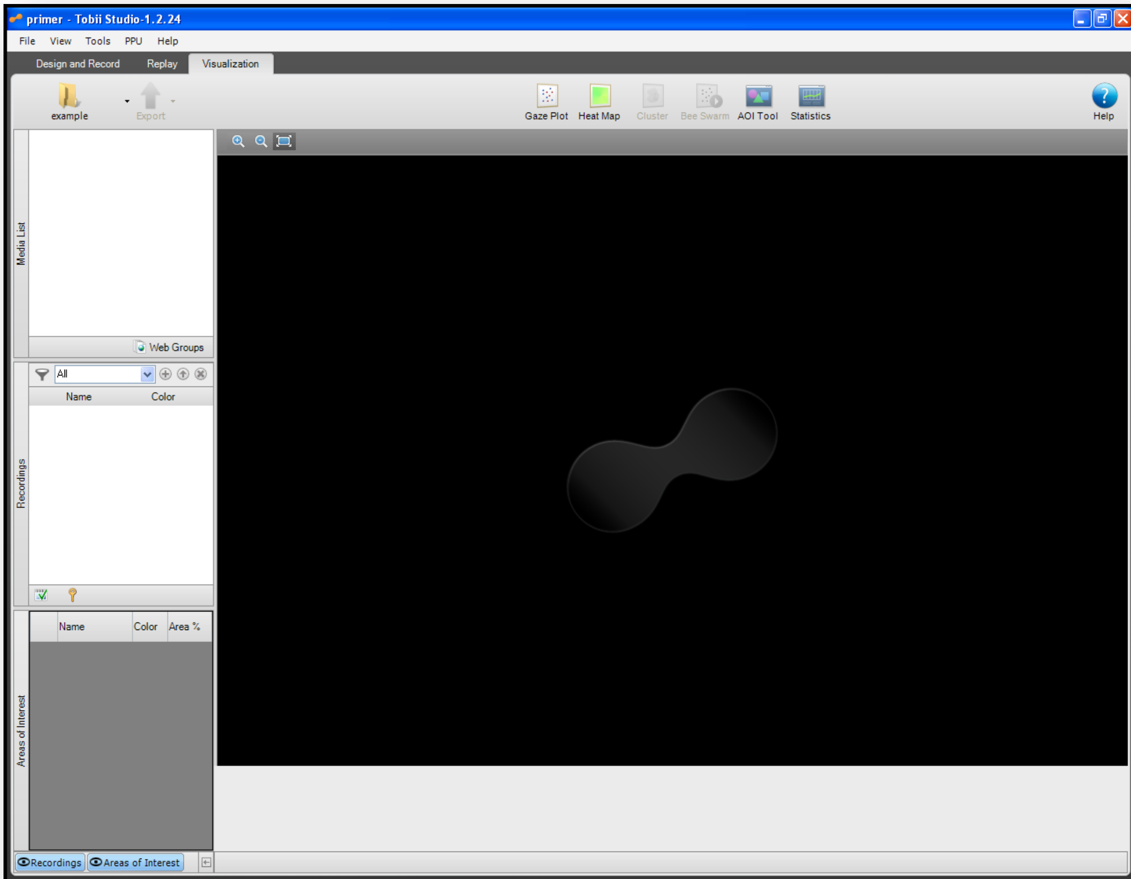


The **Replay** tab view allows playback of captured eye movement data (scanpaths). The **Recordings** and **Events**window panes at left allow selection of specific eye movement scanpath to be displayed. At bottom are VCR-like controls for controlling scanpath playback as well as for marking specific events.



Buttons below the VCR controls allow control over **Segments**, **Scenes**, and **Settings**. The **Settings** button is particularly useful for resizing the apparent dimensions of the playback, e.g., to fit the screen.

The **Visualization** tab displays aggregate views of the data. The **Media List**, **Recordings**, and **Areas of Interest**window panes at left allow selection of stimuli and recordings (and user-specified AOIs atop these) for visualization.



The rest of this guide contains examples of a simple study of image layout effectiveness, somewhat advanced studies of static images and video, web page analysis, as well as analysis of eye movements over an application.

Effectiveness of Image Layout

Collecting eye movements over a single static image can reveal the viewer's attentional deployment, which could be indicative of the viewer's evaluation of the image or the viewer's *intent*. In the former case, where the viewer attended to could suggest those image regions that the viewer found interesting. For example, regions in artwork that draw the viewer's attention, or some other pleasing forms of color, contrast, etc. In the latter case, the sequence of the viewer's fixations, their *scanpath*, could be indicative of a cognitive strategy used to direct gaze. For example, a left-to-right scanning strategy is typical of English readers when reading text. Although less obvious, a scanning strategy between text and images may also indicate some form of coherence the viewer is striving to achieve between related pictures and descriptions. In usability and/or marketing applications, one is often interested in detecting a strategy used when making decisions, e.g., when considering a purchase. Eye movements may help determine whether the viewer is seeing relevant information (product descriptions), and/or whether particular image regions (advertisements) are being viewed at all.

Background

Viewers, acting as consumers, may use a combination of strategies in a decision environment of interest. The success of any marketing program rests on the ability to accurately predict which choice strategy the consumer is likely to use (Wright, 1975). For example, consumers may be sensitive to the differences between the strategies of simplification and optimization, when engaging in a covert cost-benefit analysis. Although individuals can apparently adapt their decision strategies based on situational priorities, the pattern of these adaptations is not well understood. Of the numerous methodological approaches to the study of cognitive factors at play during the processof consumers' decision-making (Lynch & Srull, 1982), we focus on therecording, depiction, and analysis of overt visual attention deployment, as evidenced by the movement ofthe eyes. Although limited attentional capacity leading to selective aspects of attention have been known forsome time, attentional deployment strategies used by consumers have not been well studied. Contrastingwith well-known strategies employed in reading(Rayner, 1998), little is known abouthow consumers perform visual search. Here, we will simply look at one specific aspect in the composition of an ad: the effectiveness of *deictic reference*, as employed in the image.

Hypothesis

Deictic reference is simply a reference made to some point or direction in space through indirect indicators such as verbal utterances (e.g., "I'm looking *at that*") or by simply looking at something. Humans, as well as other animals, are remarkably good at being able to follow one another's gaze direction. The experimental hypothesis to be evaluated in this mock study is whether deictic reference does indeed draw viewers' gaze toward a particular region on the screen. More formally, we would have to state the null hypothesis as the assumption that deictic reference does not lead to a measurable effect. We hope to be able to reject the null hypothesis and accept its alternative that deictic reference is precisely the only effect that is responsible for the difference in our measured, or dependent, variable: a count of fixations over the region of interest.

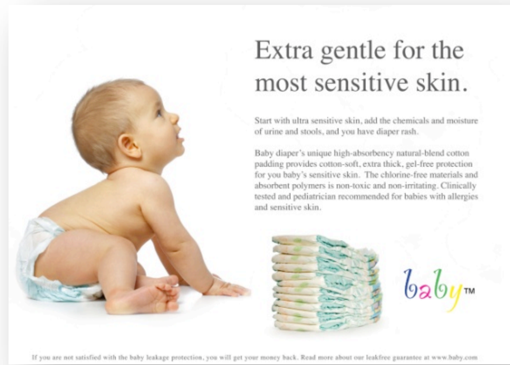
Generally, the magnitude of the difference in measurements isevaluated by an analysis of variance, or ANOVA, which in turn is based on the F-ratio: the variation due to an experimental effectdivided by the variation due to experimental error (Tangen, 2002). Thenull hypothesis assumes $F = 1.0$, or that the effect is the sameas the experimental error, hence no significant difference isexpected (between means of the sampled responses, assumedto be normally distributed). This hypothesis is rejected if theF-ratio is significantly large enough that the possibility of itequaling 1.0 is smaller than some pre-assigned probability,e.g., $p = 0.01$, or one chance in 100, meaning that if $p < 0.01$ then the observed difference is > 99% certain to besolely due to experimental effect (the means are sufficientlyfar apart that the distributions do not overlap).

Experimental Design

To test our hypothesis, we consider the number of fixations as our main *dependent variable*. More specifically, we would like to count the number fixations in a specific region of interest that we define in our given image. To test the effect of deictic reference (our operational *treatment,* or *independent variable*) we want two images that are identical to each other in all respects but one: the use of deictic reference. The study can be performed within-subjects, such that each viewer sees both images. Oneaspect of this design thatwe needto worry about is the order of presentation of the image pair. Let's call the two images A and B. If we were to show them in sequence AB to every viewer, then whatever measurements we record could be the result of order effects, e.g., whether viewers looked at a particular region in the image could be due to fatigue or memory, either of which would confound our desired measurement of just the effect of deictic reference. Instead, to counteract these order effects, we would like to either explicitly control the presentation order, or have it be randomized.

Stimulus

To control for the appearance of deictic reference while holding everything in the pair of images as similar as possible, we choose an image where only the pose of the principal actor in the advert is changed (in this case it is the image of a baby that is either looking at the viewer, or looking at the intended region of interest).



Notice how in one of these images the baby, by looking at the viewer, draws attention to the baby's face as well as drawing attention to the diaper. Notice the finger pointing to the object of interest? That too is a form of deictic reference. (The white of the diaper against the baby's skin tone also provides two high-contrast edges, a so-called "low-level" visual feature that can draw gaze more or less automatically.) In the second image, by having the baby look at the headline, gaze is (intended to be) drawn to the text. Because we are used to reading following a certain strategy (left-to-right and top-to-bottom), in this image, gaze may also be drawn more quickly to the brand name found towards the bottom of the right hand side of the page.

For the sake of fair comparison, we cannot readily test the visibility of the diaper via numbers of fixations because one of the two images does not present the same kind of stimulus. In other words, this particular stimulus is only present in one image. All else being equal, the text appears to be identical in both images, allowing a more fair comparison. The text can be broken up into three subregions: headline, description, and brand.

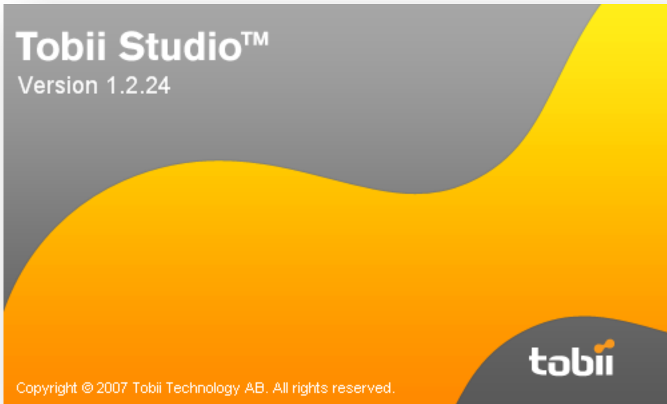
Apparatus

We will use Tobii's Studio (v1.0) software for preparation of the visual search task as well as eventual analysis. At the time of this writing, a Tobii ET-1750 video-based corneal reflection eye tracker was used to test the procedure. The Tobii ET-1750 operates at 50 Hz at an accuracy of about 0.5 degrees visual angle (bias error). At the time of testing, screen resolution was set to 1280x1024. The eye tracking server ran on a Sun model w2100z dual 2.0 GHz AMD Opteron 246 machine equipped with 2 G RAM. The server was running Windows XP.

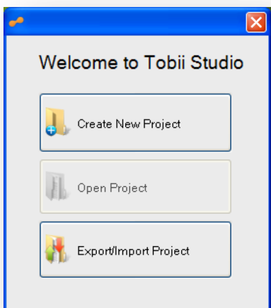
For reproducibility reasons, it is quite common, if not entirely expected, to provide system specifications. For eye tracking experiments specifically, one typically needs to list the sampling rate (50 Hz) and accuracy (0.5° visual angle) of the device, its mode of operation (video-based corneal reflection), as well as any eye-tracking-specific procedures such as number of calibration points used as well as eventual method of eye movement analysis. Note that newer equipment from Tobii, the T120 models, operate at 120 Hz.

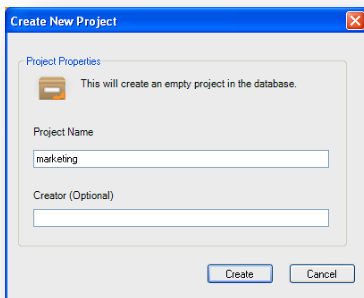
Studio

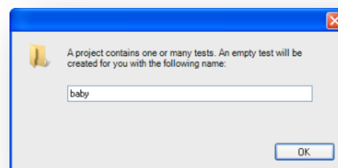
Start Tobii's Studio, either via the Start menu or by double-clicking the Tobii icon.



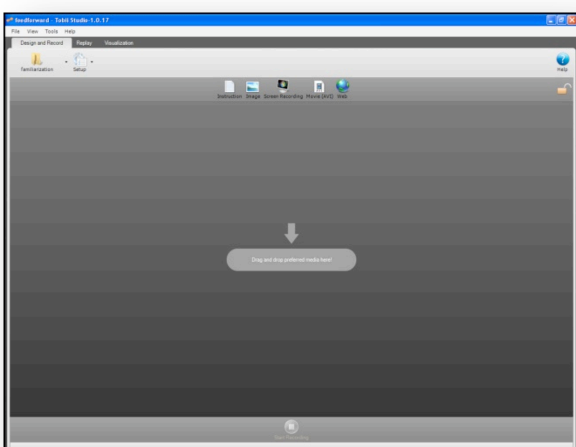
If no projects have yet been created, the **Open Project** option will be grayed out from Tobii's **Welcome to Tobii Studio** dialog.







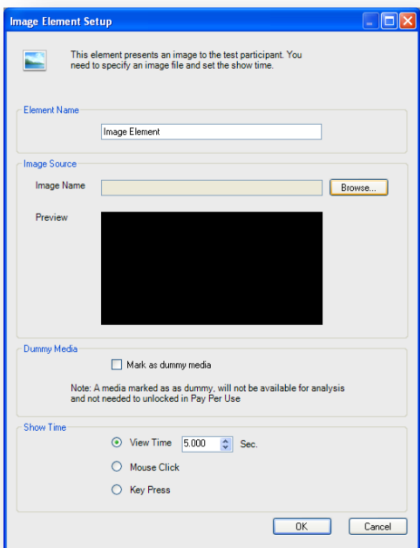
Create a new project, providing a new project name and (optional) creator identity. Studio's database is made up of projects where each project contains at least one test. Each test, in turn, is made up of an instance of a visual stimulus such as a still image, a video clip, a web page, or screen recording (desktop), or a sequence of the former types of stimuli. Once you provide a project name (e.g., **marketing**), Studio will automatically create an instance of an empty test, providing you with a chance to name that test (initially the default test takes the name of the project, here we can use **baby** as the test name).



Design and Record: treatment

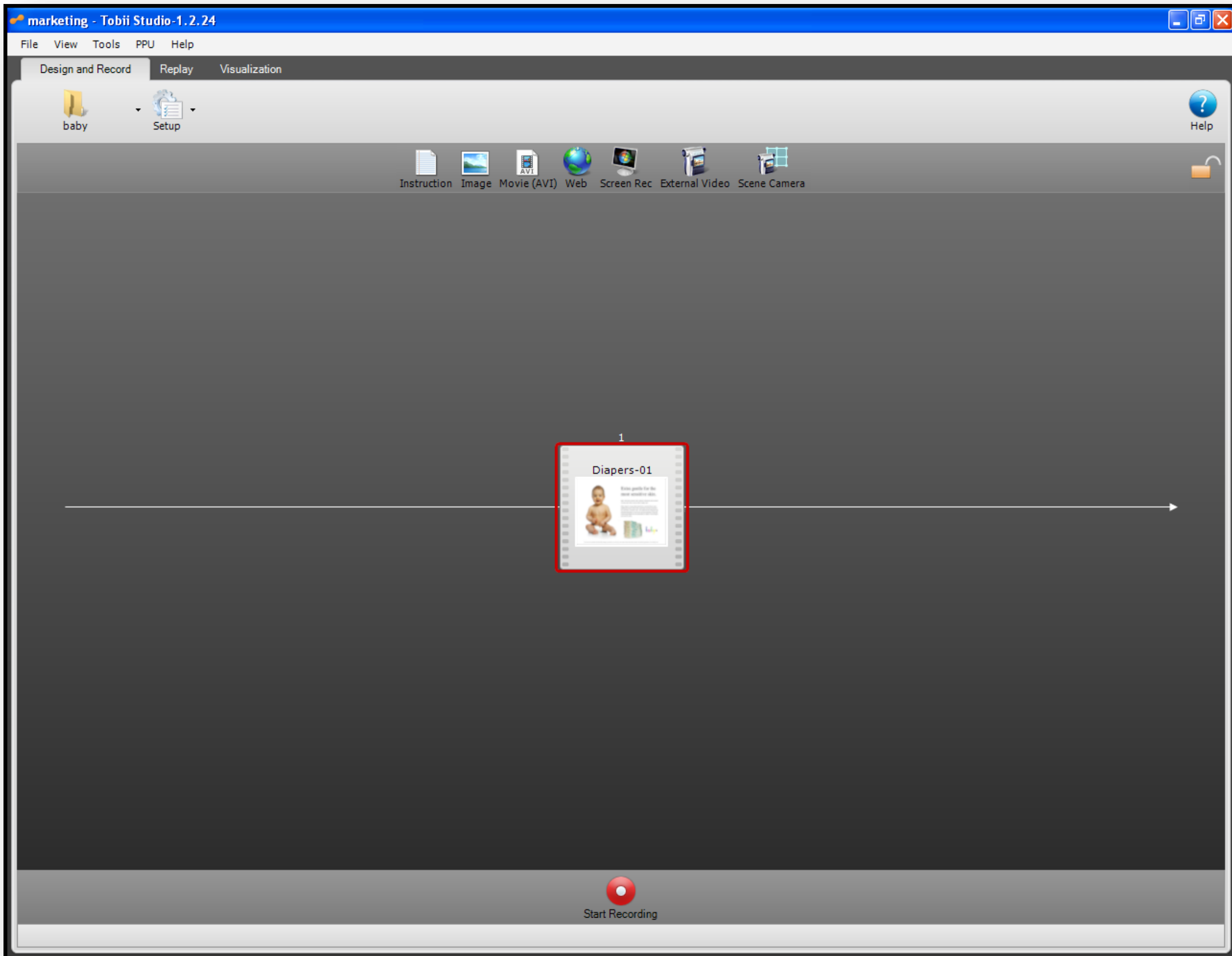
Underneath the main tab selector Studio shows the possible types of media that you string together to assemble a stimulus sequence for the viewer. When you mouse-hover over each media type its name is highlighted, as shown at right when hovering over the **Image** media type.Let's start by selecting the **Image** stimulus and click-and-dragging that to the gray area where Studio wants the media dropped.

When you drag and drop a placeholder for an image into the central media viewer, Studio pops up an **Image Element Setup** dialog which let you select the image of your choice.

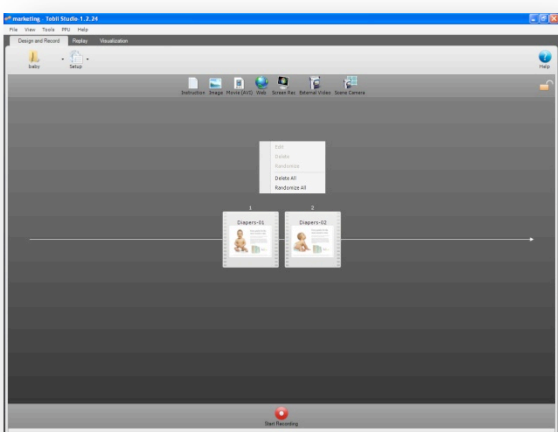


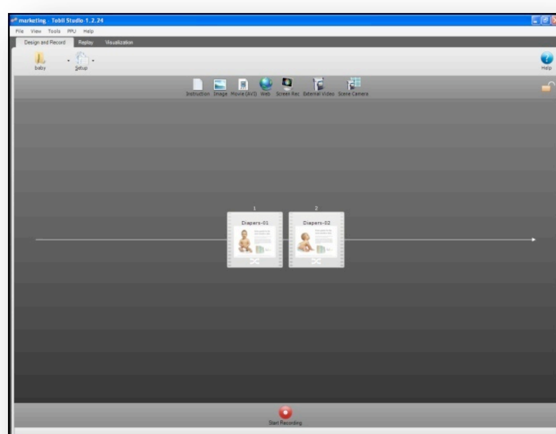
In this dialog the **Element Name** refers to the media label that you assign to the given stimulus. The element name does not have to be the same as the file name, although that is the default. It is more efficient to first press the **Browse…** button in this dialog and then make the decision to either keep the default element name or provide something more or less meaningful. The element name will be visible in the main media viewer, so it should be informative to the experimenter but yet should be sufficiently ambiguous to the participant so that they are not necessarily aware what is being shown.

Press the **Browse…** button for Studio to pop up the **Open** file browser. From within the file browser, select the **Diapers-01** image from the **stimulus→images→tobii_fake_ads** directory. In the **Show Time** dialog GUI group, check the **View Time** option and use 5.000 seconds as the view time. This results in a *machine-paced* stimulus presentation, allowing us to, more or less, analyze the number of fixations between conditions (different images) in a manner that can be thought of as normalized with respect to time. This is important for the "all things being equal" approach to the experiment. Alternatively, clicking the **Mouse Click** radio button would make playback of this image element *self-paced*, with the user controlling advancement to the next element by pressing one of the mouse buttons. Although useful in other scenarios, e.g., reading instructions, giving up this control of stimulus duration would give us a potential confound in the number of fixations recorded over each of the pair fo images. The**Key Press** radio button would also set the display of the element to be*self-paced*, advancing after a keyboard button press. The result of dropping in the first baby image element into the media viewer is a single media element.



In a similar manner, drop in the second **Diapers-02** image. The result is a two-image sequence. In order to randomize the order, right-click anywhere on the background to pop up a menu that allows one to **Edit**, **Delete**, or **Randomize** an individual media element, or that allows one to **Delete All** or **Randomize All**. Choose **Randomize All** to randomize the sequence. You should notice that the image elements now contain a kind of “shuffle” symbol at bottom.



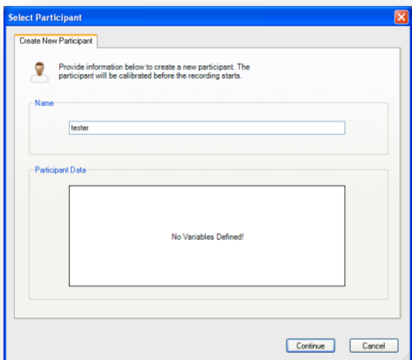


This completes the basic image study design. You are ready to run your first participant through the study.

Running the Study

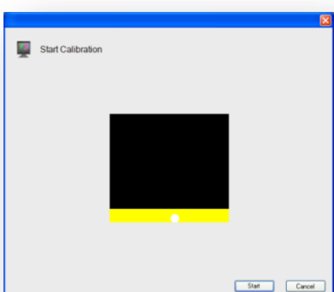
Run the simple randomized test by pressing the **Start Recording** button at the bottom of the Studio window. Run yourself through first to make sure that you included the proper stimulus images and check the machine-paced display timings.

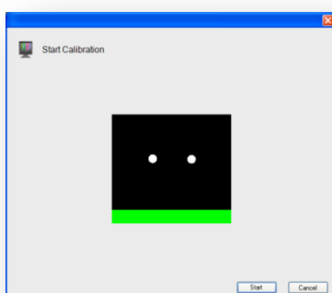
Once you press the **Start Recording** button, if no participants have previously been defined, a **Create Participant** dialog box pops up.

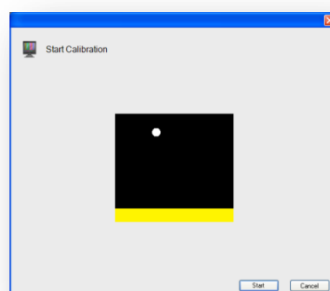


Enter in an ambiguous name, e.g., **tester**. Keeping participant names ambiguous is generally a good idea since many Institutional Review Boards (IRBs as they are sometimes known) require that all participant records are *anonymized*, i.e., no personal information is ever kept. Thus it is important to get into the habit of using participant labels instead of actual names, e.g., S1, S2, and so on.

Clicking **Continue** displays a track status window showing the location of the eyes from the camera's point of view. Make sure that the eyes are centered and level on the screen with the participant sitting at an appropriate distance. If the participant is too far, the discs representing the eyes will be too low in the track status window; if the participant is too close, the discs will be too high up in the track status window.

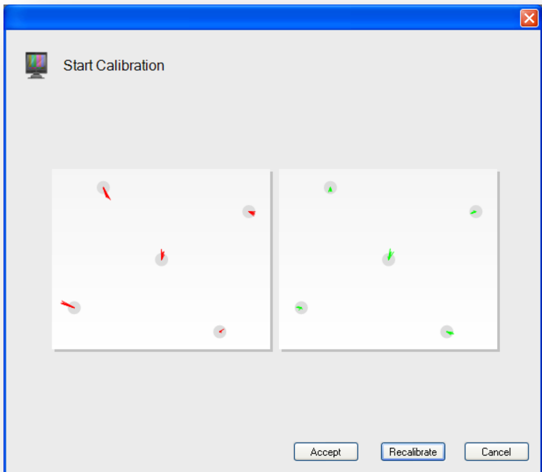






A good track status, with both eyes visible to the eye tracker's cameras will show a green bar at the bottom. If one eye is not seen by the cameras, the status bar will show in yellow. If neither eye is visible to the camera, the bar will be shown in red.

Next, system calibration is performed. Calibration consists of a colored disc moving from spot to spot on a blank screen. The intent is for the viewer to follow the disc as it moves and to fixate it when it stops periodically at one of 2, 5, or 9 calibration locations. The entire calibration step takes only a few seconds. At each stopped position, the camera records the direction of the viewer's gaze and calculates an error from the known location of each of the stopped calibration disc locations. The idea is that the viewer look at each of the "calibration dots" in turn. A good experimenter strategy is to instruct the participant to follow the dots and maintain gaze on them when they come to a stop, and not to anticipate where the next dot might be (i.e., don't race ahead of the dots). When calibration is finished, Studio shows a summary of the calculated error for each of the left and right eyes at every stopped position of the calibration disc.

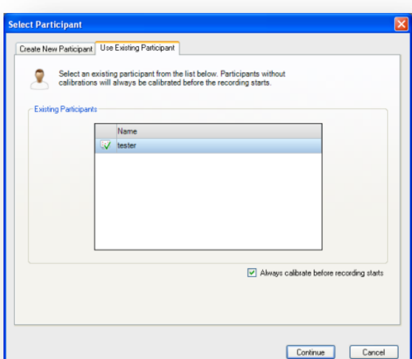


For an acceptable calibration, the error lines extending from the center of each calibration disc should be short. If there are really long lines extending from any of the calibration discs, the eye tracker should be recalibrated.

This is a rather important point: it is the eye tracker that is being re-calibrated, and not the viewer! Some participants may be concerned if the eye tracker has trouble calibrating to their eyes. This may be due to glasses worn by participants (with very shiny rims for example), heavy mascara, post-Lasik surgery (flattened corneas), or something more mysterious. In such cases, remember to be sensitive to participants' feelings: it is always easier to blame the equipment than the person. Poor calibration may mean that a participant's recording might not be used in the final analysis, but the participant need not know this.

Once calibration is accepted, the eye tracker will record the viewer's scanpath over each stimulus in turn, in this particular case, in a randomized order.

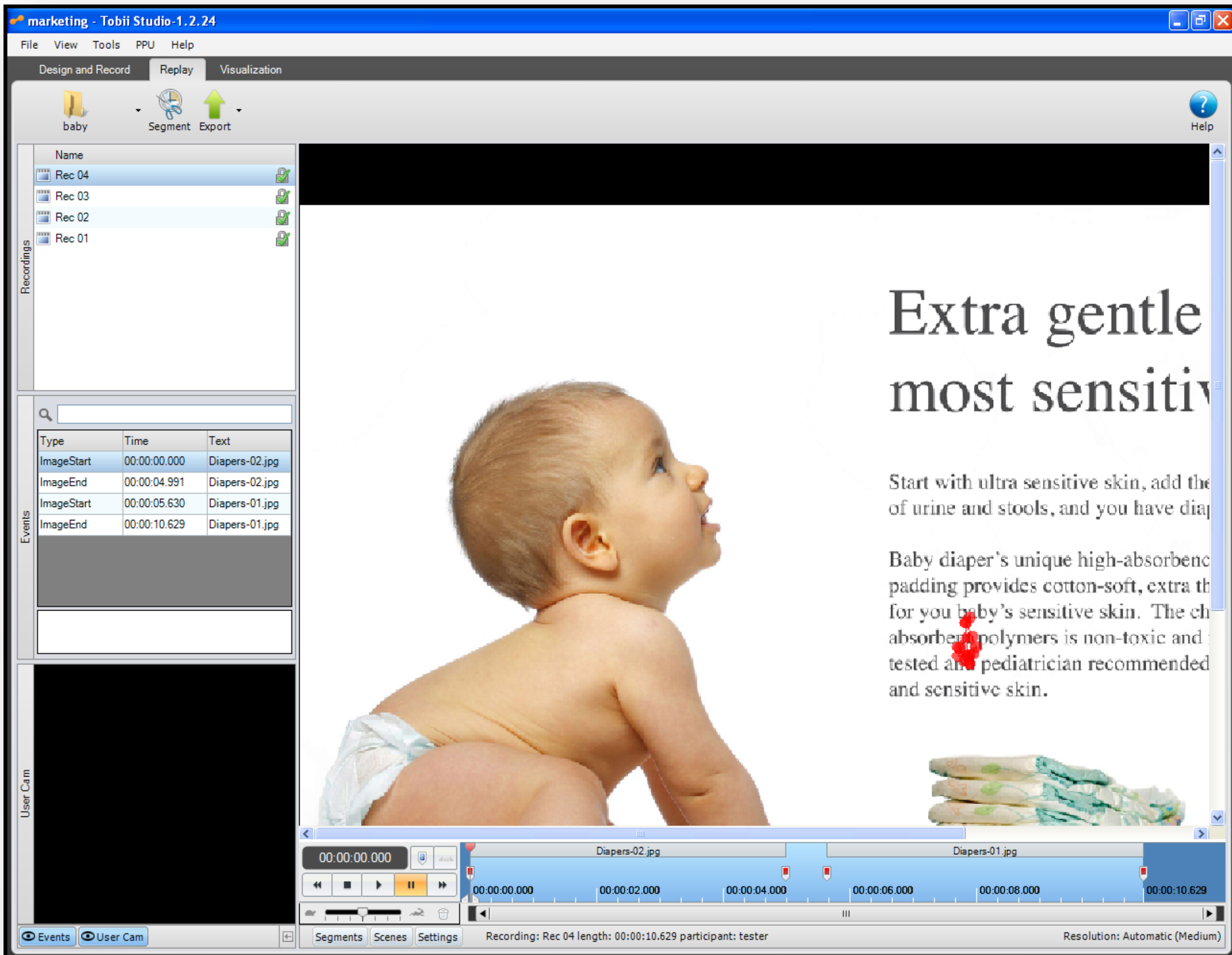
Repeat the above process a number of times (e.g., record four scanpaths). Notice that once a participant has been defined, a dialog window shows up asking to **Select Participant**. Click the **Use Existing Participant** tab and select **tester**.



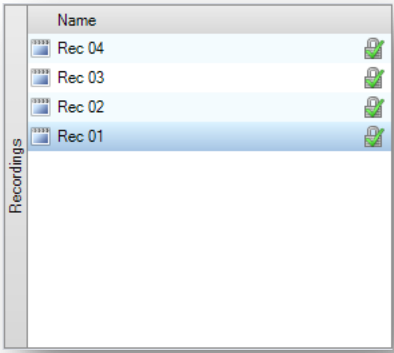
Note the **Always calibrate before recording starts checkbox**. Checking this checkbox ensures that calibration is performed every time, regardless of whether a previous calibration exists for a participant. Generally a new calibration is a good idea unless you can ensure that the given participant will be seated in almost the same position that s/he was in when the original calibration was taken. Since this is often difficult to guarantee, even in the best laboratory conditions, performing a new calibration is often prudent. When ready, press **Continue** and repeat again to record four recordings.

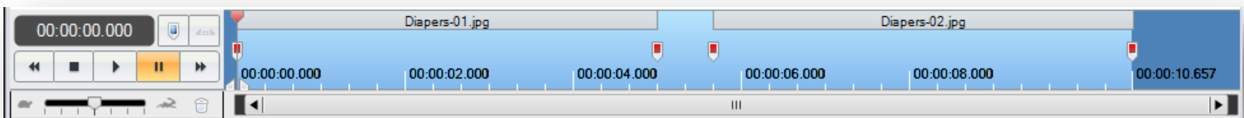
REPLAY: TESTER SCAN

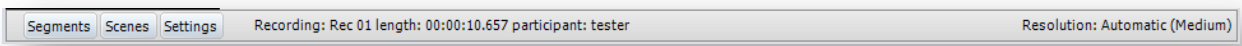
Select the **Replay** tab to inspect the tester's scanpath(s) (in this case 4 recordings were made).



Choose a recording from the **Recordings** pane and then press the play button using the VCR controls to play back the selected scanpath. Note the status bar and additional buttons below the VCR controls and timeline.







Each of the buttons below the VCR controls, namely **Segments**, **Scenes**, and **Settings**, expands to provide additional playback options. For example, the **Settings** buttons provides control over the appearance of the playback, including the color of the scanpath, the tail duration, and the opacity (transparency) of the display. Click on the **Fit to screen** radio button and press the play button to see the dynamic replay of the tester's scanpath. Click on different recordings to view each of the four experimental runs.

You should see small circles representing fixations moving from spot to spot as the tester fixated each element in the display. Fixations are joined by straight line segments that indicate saccadic jumps between fixations.

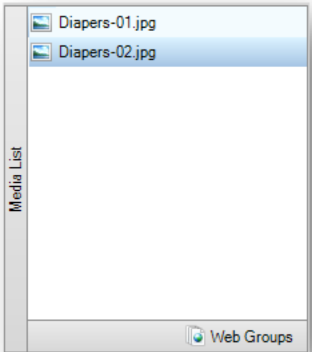
Visualization: Analysis and Results

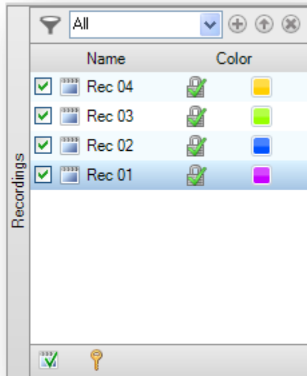
The goal of the analysis of this experiment is to determine whether the image content (the baby's pose) had any effect on the participants' scanpath. We are mainly interested (in this mock example) in counting the number of fixations over the three textual regions, not so much over the baby.

The main result we are after is statistical support for our hypothesis (the baby's pose makes a difference). Note that this does not constitute proof of any kind (scientific or otherwise), but merely provides evidence for acceptance of the hypothesis. More formally we would have declared the hypothesis in the statistically neutral sense by specifying the null hypothesis, H_o, which states that no effect is expected. Statistical analysis of the data then would either accept H_o or reject it in favor of the alternative, which is what we are hoping for. The statistics we use essentially state that if there is a significant difference in the means of the dependent variable (number of fixations), it is due to the effect and not chance alone, therefore there is a statistically significant reason for rejecting the null hypothesis.

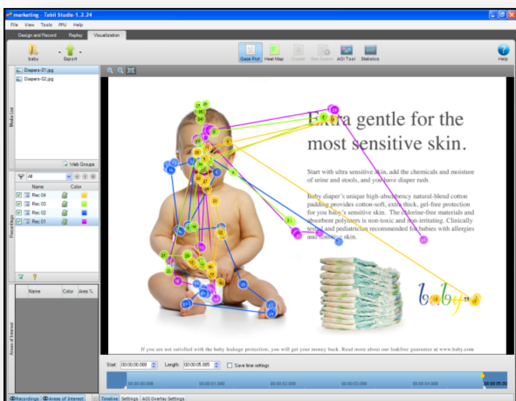
Before "crunching" the data, we can use Studio's visualization tools to see if this effort is worthwhile. Intuitively, we would expect to see an absence of fixations over the textual areas when viewing the "facing straight" baby (consider this the control, and the baby looking at the textual regions as the treatment). Studio provides several visualization tools, including **Gaze Plot**, **Heat Map**, **Cluster**, **Bee Swarm**. Some statistical analysis is available by defining Areas Of Interest, or AOIs, with the **AOI Tool** and the **Statistics** button. For the present case we are mainly interested in comparing the aggregate eye movement data of the fixations over the two images.

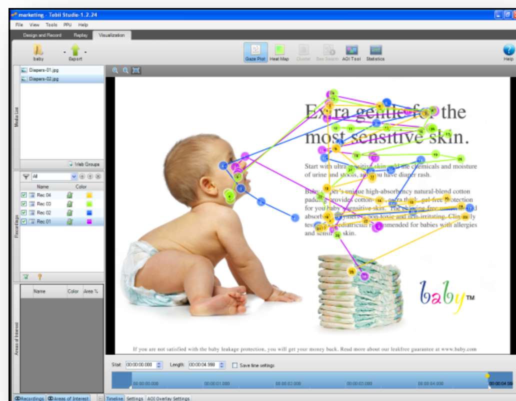
For any of the visualizations available, use the **Media List** to select the stimulus image on which to render the graphics. For example, image **Diapers-01.jpg** is the image containing the baby facing forward (control) and **Diapers-02.jpg** is the image of the baby looking at the text (treatment). For each of the control and treatment visualizations, all(4) recordings are used from which to obtain the aggregate visualizations.





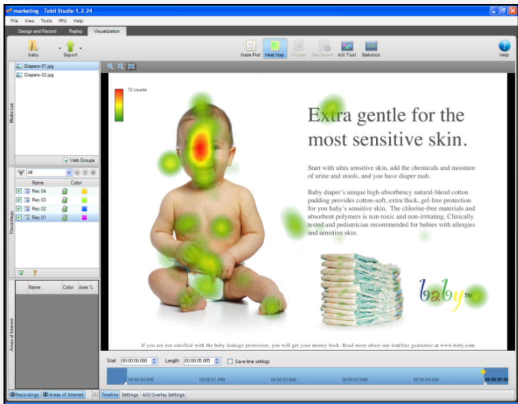
Using the **Gaze Plot** button, we can examine the general pattern of eye movements exhibited over each control and treatment images.





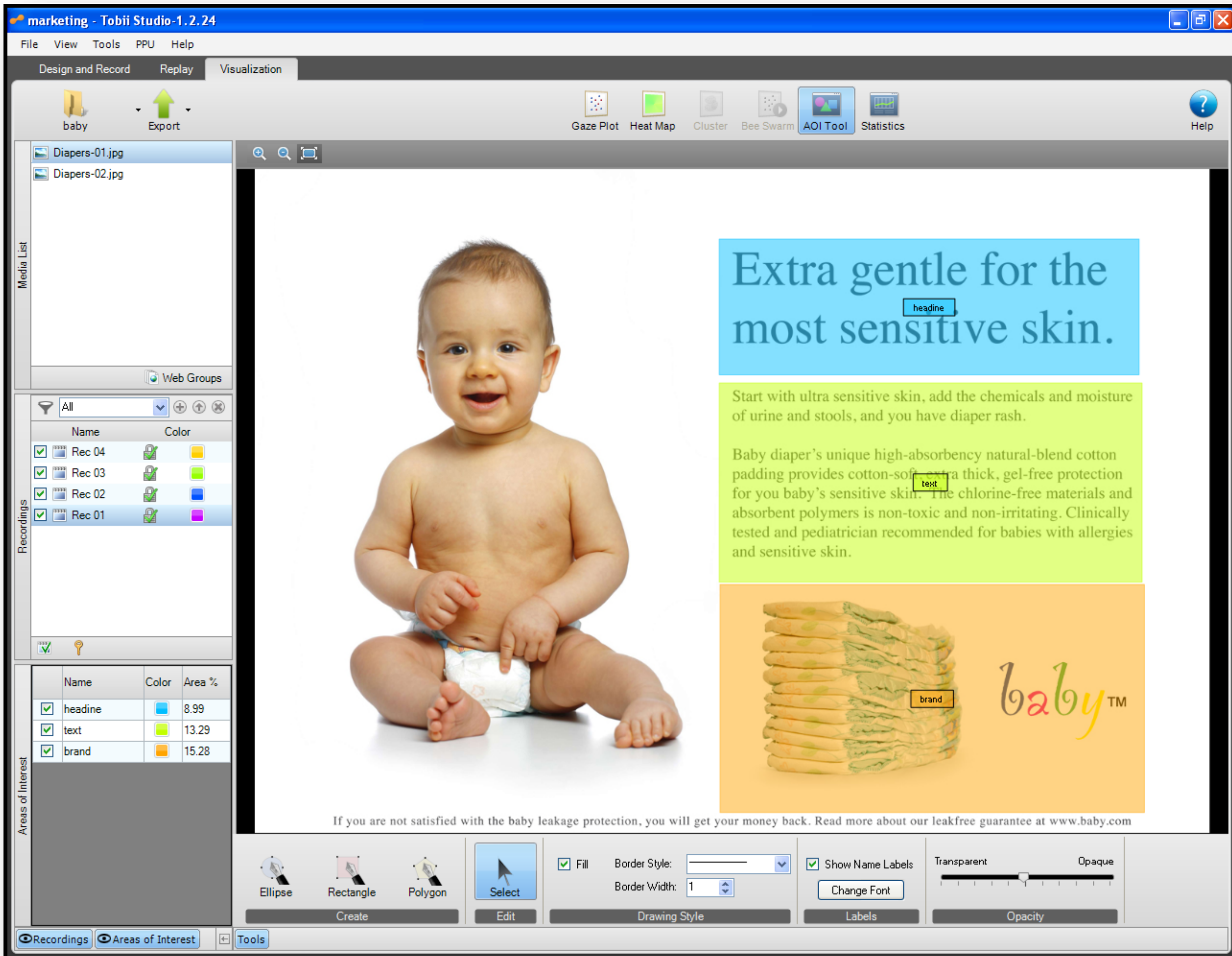
If there is a large number of recordings, due to a large number of participants or replications (recordings), gaze plots can become somewhat congested. The left gaze plot belongs to the control, the right to the treatment. Notice that, as expected, the treatment appears to be drawing gaze to the right side of the image, the textual region. Meanwhile the control's gaze plots seem to show interest in the visual component (the baby), not the text.

Heatmaps are better suited to the aggregate view since these tend to better depict data collected over a large number of recordings.

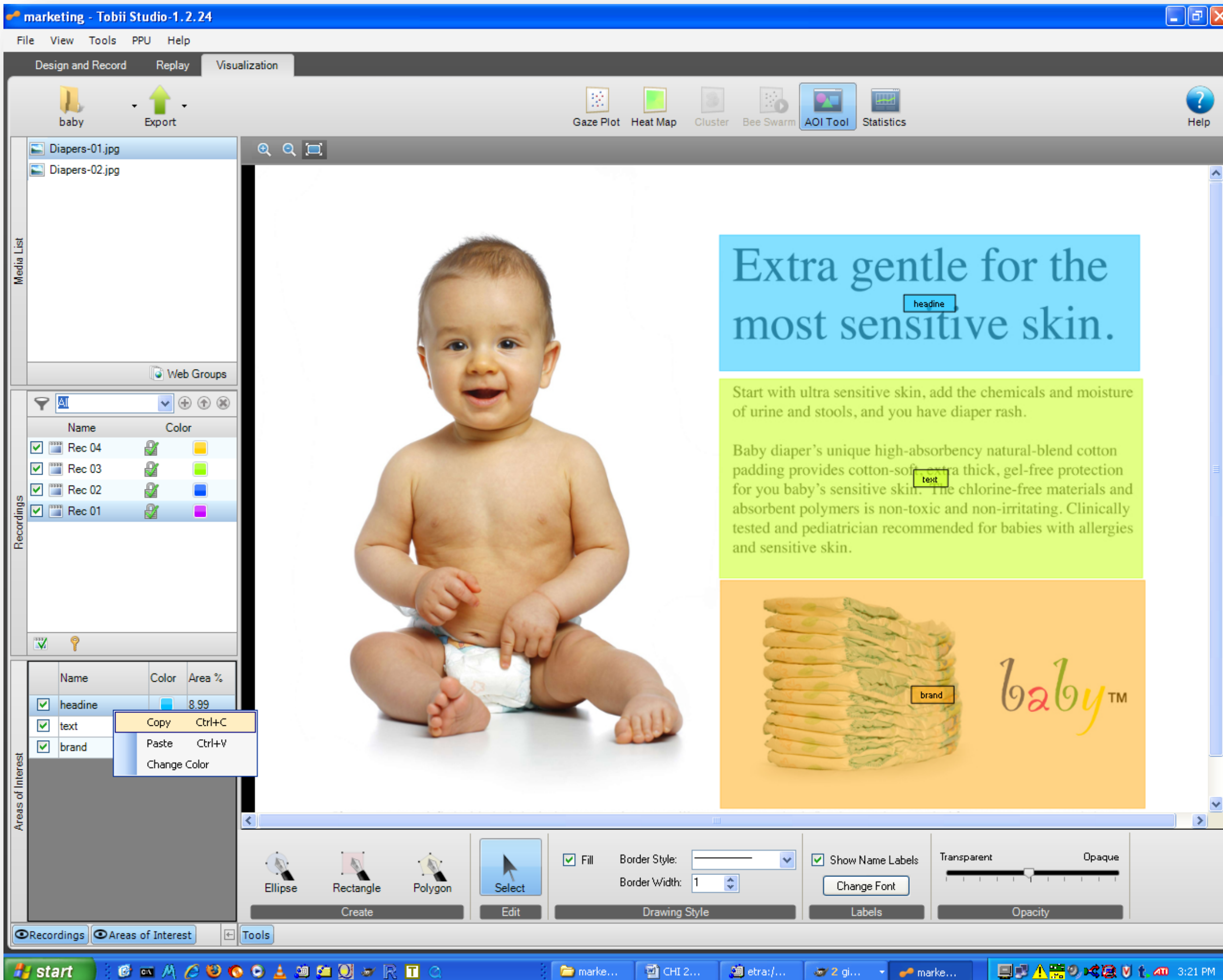


As a preliminary tactic toward statistical analysis, the **AOI Tool** and **Statistics** buttons can be used to generate some simple plots. The aim here is to generate statistics describing eye movement metrics per Area Of Interest. In this case, the AOIs are the textual regions: headline, description (text), and brand. For example, we would like to obtain statistical analyses regarding the number of times each AOI was seen in each image. The first step is to define the AOI regions per stimulus image.

Click on the **AOI Tool** button at top (to the right of **Gaze Plot** and **Heat Map**), and use the **Create** and **Edit** tools to define, position, and shape AOIs on the stimulus image. The **Ellipse**, **Rectangle**, and **Polygon** are fairly self-explanatory shapes not uncommon to typical drawing packages. Use the **Areas Of Interest** pane at bottom left to provide informative names to the AOIs, e.g., **headline**, **text**, and **brand** to identify the three textual regions.

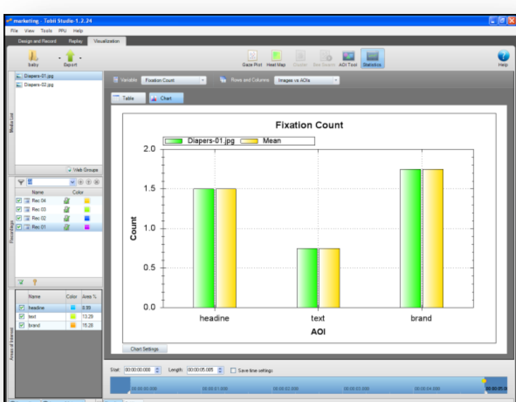


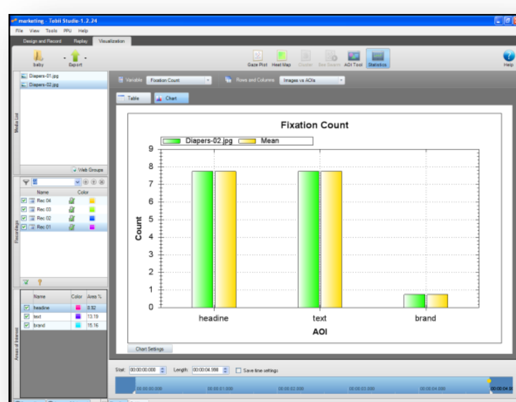
Note that AOIs are defined within a stimulus image. What we really want here are AOIs that span both images. To do so, first create the AOIs atop the **Diapers-01.jpg** image, as seen above. Click over the default AOI names in the **Areas of Interest** pane to rename the AOIs to **headline, text,** and **brand.** Then, right-click on each to **Copy** an AOI to memory. Then select **Diapers-02.jpg** in the **Media List** pane and **Paste** the AOI into the **Areas of Interest** pane. Repeat for each AOI.



With the stimulus and recordings selected in the **Media List** and **Recordings** panes, and AOIs defined, use the **Statistics** tool to generate plots of eye movement data per AOI. To specify the type of data to display, use the **Variable** and **Rows and Columns** drop-down lists to specify the plot to generate.

For example, use **Fixation Count** as the variable of interest and **Images vs AOIs** to generate data plots for the control and treatment images.





On average (over all recordings), the plots show that the fixation counts appear higher on the headline and text over the **Diaper-02.jpg** (treatment) image. About 7.5 fixations vs. 1.0 fixations (note the scale at left). The fixation count over the brand icon appears to be about the same, namely about 1.0 fixation, give or take (a bit more than 1.5 over **Diaper-01.jpg** and just under 1.0 over **Diaper-02.jpg**).

Are these results *significant* (in the statistical sense)? It is difficult to tell just by looking at the plots. To perform a more robust analysis of the data, more advanced methods are required. Studio allows data export, which can then be used by other packages to perform statistical analyses. A good exercise here would be to attempt to do just that: we would like to perform ANOVA (analysis of variance) on the fixation counts (and possibly other metrics, e.g., fixation durations) per AOI. This is left as an exercise for the reader, but the next (advanced) example shows how this can be accomplished.

Improving Visual Search Over an Image Sequence: Advanced

Collecting performance and process measures while viewing a (sequence of) static image(s) is at once the simplest form of eye tracking usability testing as well as potentially the most robust. Because the stimulus consists of static media you can exercise a great deal of control of how the sequence is displayed. For example, you can control the order of image presentation as well as the exposure duration of each image in the sequence. In contrast, movie clips are often constrained in their display rate (e.g., 30 fps), whereas web pages often contain a great deal of information that may be difficult to keep track of (e.g., scrolling). Indeed, the introduction of any form of dynamic media tends to raise the complexity of the experiment.

Background

In this example, a sequence of static images will be used to evaluate the utility of *feedforward* (visual) information in a visual inspection training task. This example follows the work of Nalanagula et al. who showed that the presentation of eye tracking data of an expert inspector can help improve the visual search performance of novice inspectors(Nalanagula, Greenstein, & Gramopadhye, 2006). We will design a similar study.

Hypothesis

The experimental hypothesis to be evaluated states that the depiction of an expert's visual search strategy, represented by an exemplar scanpath, leads to more efficient novice visual search. The reason for this supposition is the belief that explicit depiction of a systematic search strategy will prompt the novice to adopt a similar strategy and therefore conduct a more efficient search than simply "hunting and pecking" at random (assuming this is what novices do prior to gaining experience).

Experimental Design

To test our hypothesis, we consider an expert's scanpath as our main independent variable. This immediately leads us into a design where in one condition we want to display this scanpath (our operational *treatment*) and in another we display everything but the scanpath.

Our study lends itself well to a between-subjects design, wherein the treatment group of participants is trained "by example", being shown the expert scanpath, and the control group is just shown a static image of the search field. Following Nalanagula et al., we will use the Printed Circuit Board (PCB) image data set as our visual stimulus. Unlike their study, wherein they evaluated three forms of expert scanpath display (static, dynamic, and hybrid), we will restrict ourselves to only the dynamic form of display.

The control and treatment groups are schematically depicted in Figure 0. During the familiarization phase, each group is to be shown a sequence of self-paced images of a "clean" PCB followed by examples of each of the possible board defects participants are to search for in the test phase.

If possible, stimulus used in the familiarization phase should be similar but different from what is used in the test phase. That is, we do not wish to "give anything away" during familiarization. In fact, while the familiarization phase should be fairly simplistic and easy to comprehend, the test phase should be somewhat difficult. If the test phase is made simple enough such that everyone performs at a high level of proficiency, there may be no apparent difference in performance between participants.

Note also that the test phase consists of several repetitions of the same task. Thus we can perform repeated measures (or within-subjects) analysis of individuals' performance on the search task. This is to provide sufficient power in statistical analysis, but also to give us further insight into the task complexity relative to the nature of the targets being sought. That is, we can include a subordinate independent variable into the design by varying the type of visual search target. This of course depends on the variability (and availability) of the stimulus.

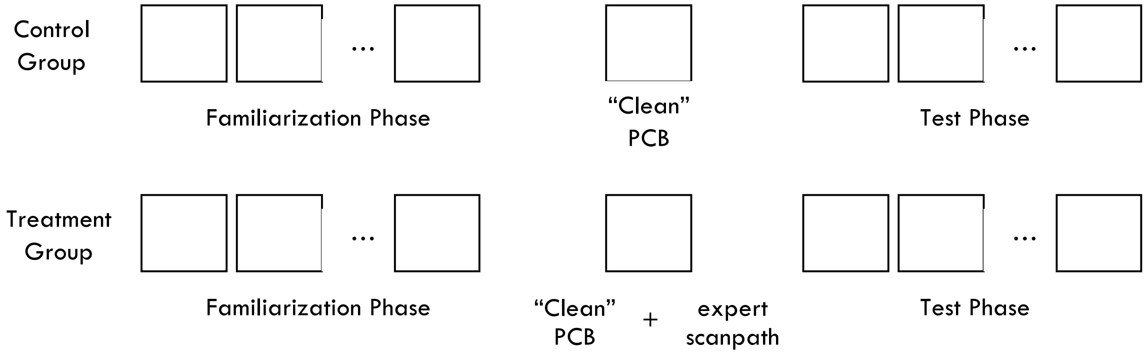
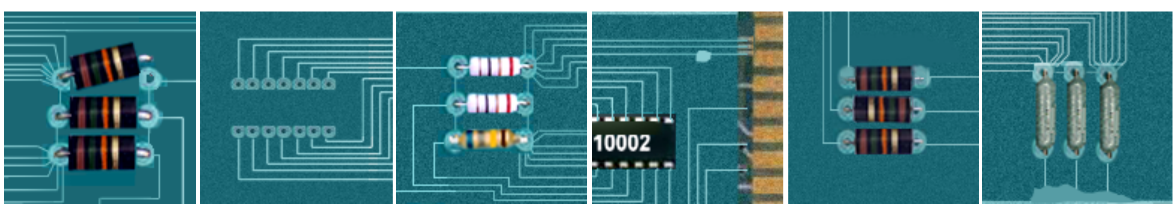


FIGURE 0: BETWEEN-SUBJECTS DESIGN

STIMULUS

From the study of Nalanagula et al. (2006), and courtesy of Anand K. Gramopadhye at Clemson University, we have a feature-rich stimulus data set consisting of a large number of Printed Circuit Board examples. The objective of the visual search (inspection) task is to locate defects on the (simulated) PCB images. There are six types of possible defects, due to simulated problems of component alignment, inclusion (or omission), selection, trace, polarity, and board (edge) flaws, shown in Figure 1. There are also two sets of PCB images, one with 34 components per board and one with 68 components per board. To satisfy the simplicity of the familiarization phase, the sparser boards can be used to ease localization of the defects, while the denser boards are used for the test phase.



(a) alignmei (b) inclusion (missing component) (c) selection (wrong component) (d) trace (wiring defect) (e) polarity (f) edge

FIGURE 1: PRINTED CIRCUIT BOARD DEFECTS

All that remains is obtaining an expert's scanpath over the 34-component "clean" PCB to show the strategy employed for a systematic search for defects. Once this is obtained, what remains is the assembly of stimulus for presentation.

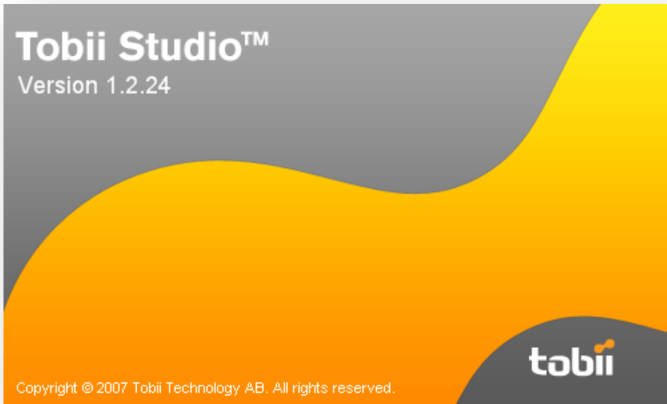
APPARATUS

We will use Tobii's Studio (v1.0) software for preparation of the visual search task as well as eventual analysis. At the time of this writing, a Tobii ET-1750 video-based corneal reflection eye tracker was used to test the procedure. The Tobii ET-1750 operates at 50 Hz at an accuracy of about 0.5 degrees visual angle (bias error). At the time of testing, screen resolution was set to 1280x1024. The eye tracking server ran on a Sun model w2100z dual 2.0 GHz AMD Opteron 246 machine equipped with 2 G RAM. The server was running Windows XP.

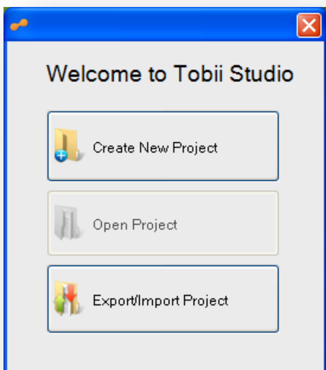
For reproducibility reasons, it is quite common, if not entirely expected, to provide system specifications. For eye tracking experiments specifically, one typically needs to list the sampling rate (50 Hz) and accuracy (0.5° visual angle) of the device, its mode of operation (video-based corneal reflection), as well as any eye-tracking-specific procedures such as number of calibration points used as well as eventual method of eye movement analysis. Note that newer equipment from Tobii, the T120 models, operate at 120 Hz.

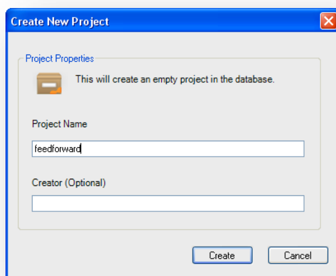
STUDIO

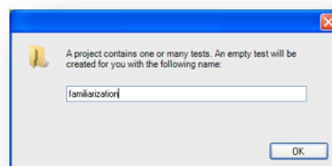
Start Tobii's Studio, either via the Start menu or by double-clicking the Tobii icon.



If no projects have yet been created, the **Open Project** option will be grayed out from Tobii's **Welcome to Tobii Studio** dialog.



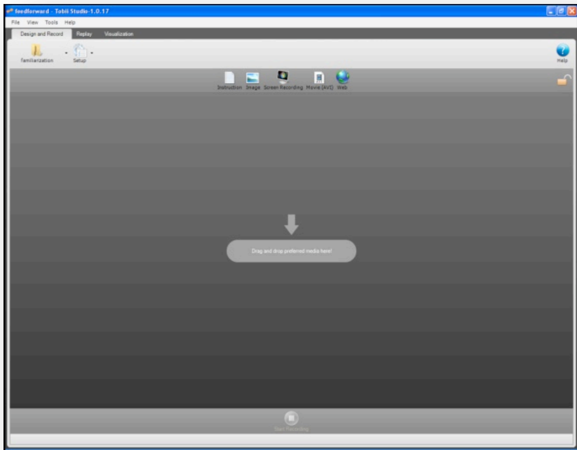




Create a new project, providing a new project name and creator identity. Studio's database is made up of projects where each project contains at least one test. Each test, in turn, is made up of an instance of a visual stimulus such as a still image, a video clip, a web page, or screen recording (desktop), or a sequence of the former types of stimuli. Once you provide a project name (e.g., **feedforward**), Studio will automatically create an instance of an empty test, providing you with a chance to name that test.

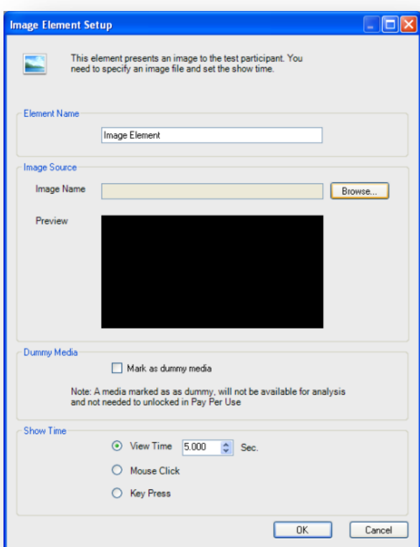
Design and Record: familiarization

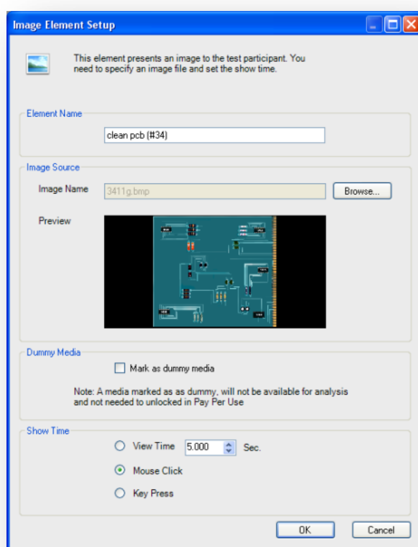
Let's start with the familiarization scenario that we will use to acquaint participants to defect types with. Enter **familiarization** into the dialog box. The Studio window should then appear with an empty test.



Underneath the main tab selector Studio shows the possible types of media that you string together to assemble a stimulus sequence for the viewer. When you mouse-hover over each media type its name is highlighted, as shown at right when hovering over the **Image** media type.

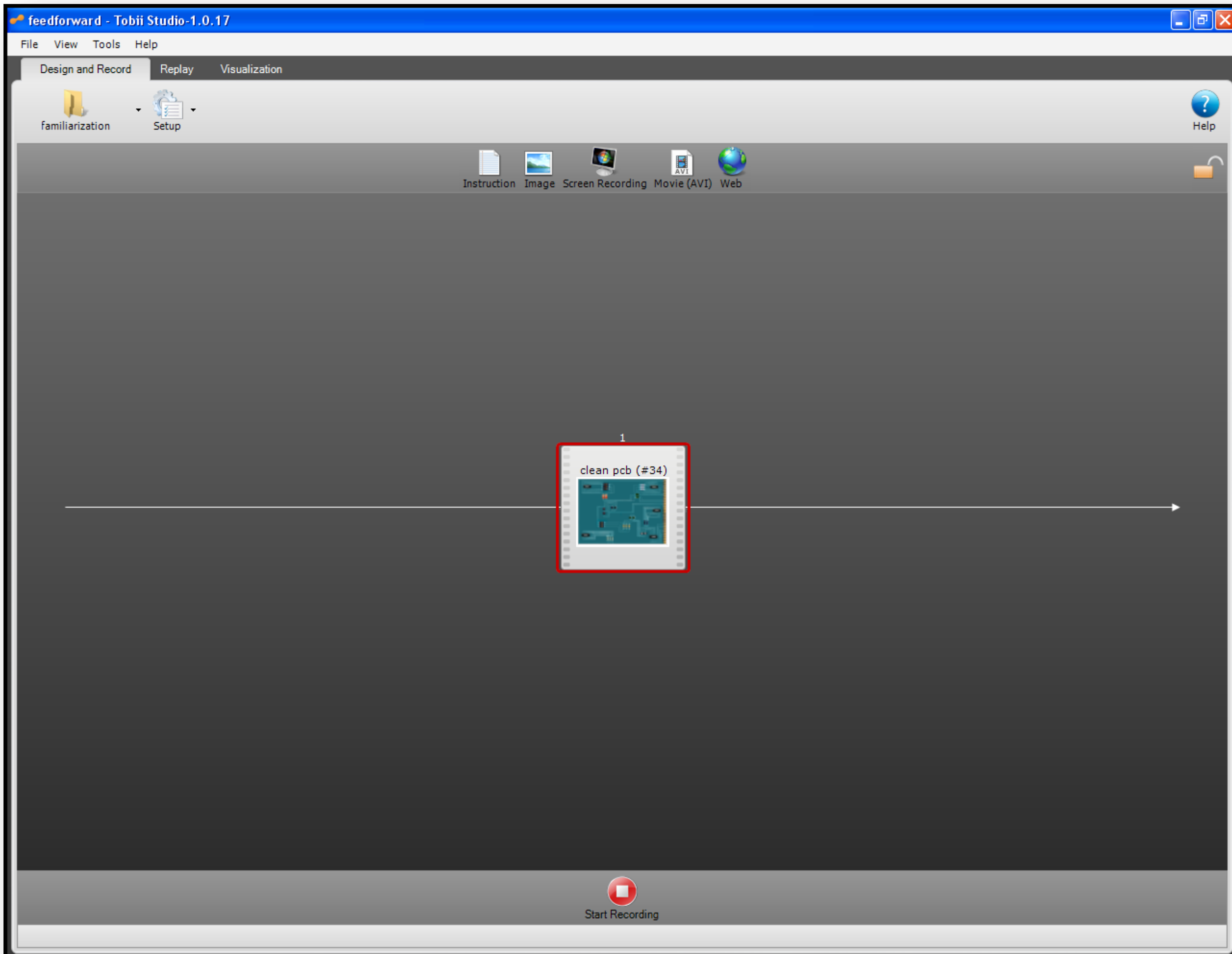
When you drag and drop a placeholder for an image into the central media viewer, Studio pops up an **Image Element Setup**dialog which let you select the image of your choice.





In this dialog the **Element Name** refers to the media label that you assign to the given stimulus. The element name does not have to be the same as the file name, although that is the default. It is more efficient to first press the **Browse…** button in this dialog and then make the decision to either keep the default element name or provide something more or less meaningful. The element name will be visible in the main media viewer, so it should be informative to the experimenter but yet should be sufficiently ambiguous to the participant so that they are not necessarily aware what is being shown.

Press the **Browse…** button for Studio to pop up the **Open** file browser. From within the file browser, select the **3411g.bmp**"clean" PCB image from the **stimulus→images→PCB→34→clean** directory. Rename the default **3411g** element name with **clean PCB (#34)**. In the **Show Time** dialog GUI group, click the **Mouse Click** radio button. Setting this option makes playback of this element *self-paced*, with the user control advancement to the next element by pressing one of the mouse buttons. Similarly, the **Key Press** radio button would set the display of the element as *self-paced*, advancing after a keyboard button press. The **View Time** radio button provides *machine-paced* display of the element, with its duration specified in the adjacent text box. The result of dropping in the first clean PCB image element into the media viewer is a single media element.

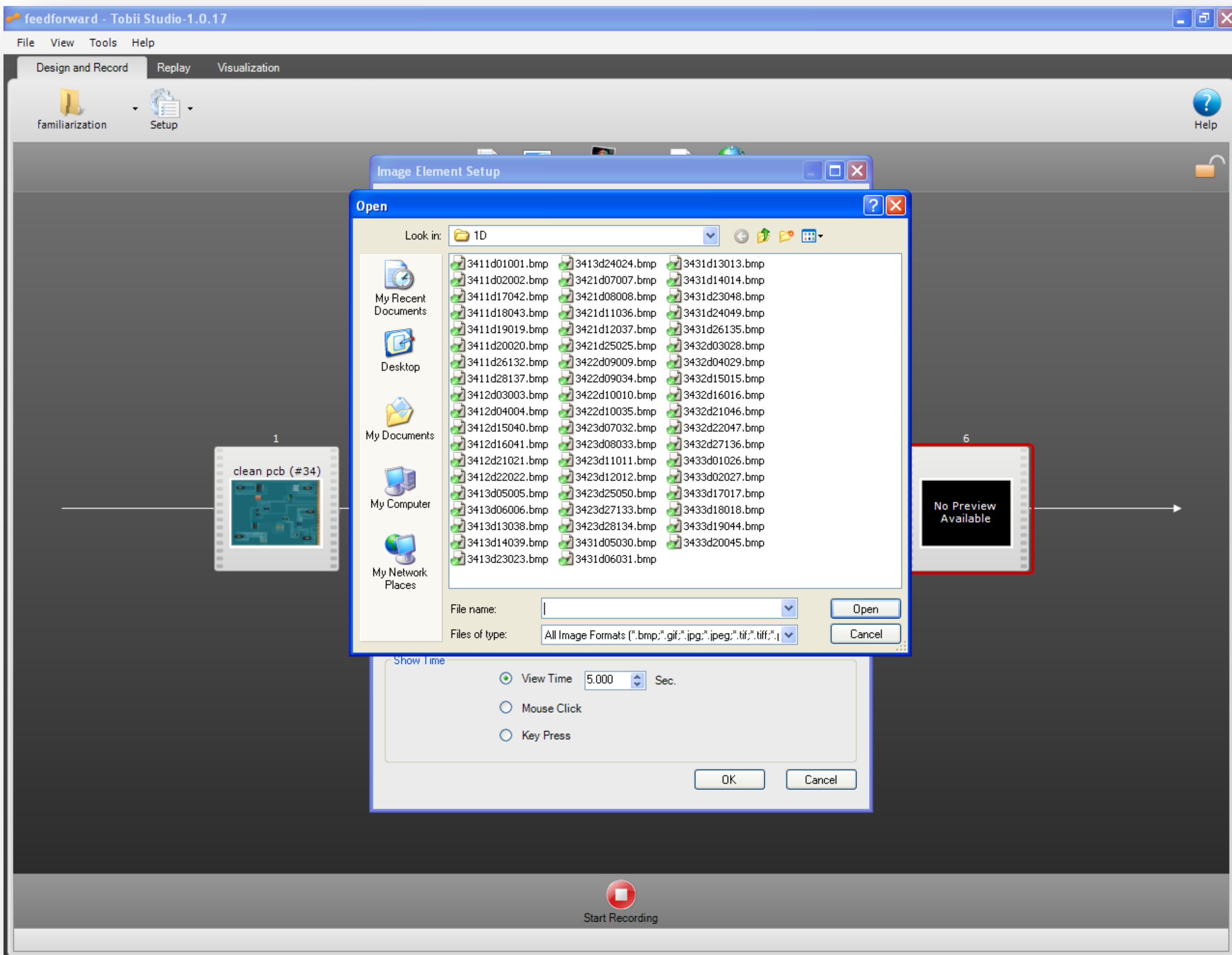


Continue dropping in the following image media elements:

stimulus→images→PCB→34→1D→3411d17042.bmp
stimulus→images→PCB→34→1D→3411d19019.bmp
stimulus→images→PCB→34→1D→3412d03003.bmp
stimulus→images→PCB→34→1D→3412d04004.bmp
stimulus→images→PCB→34→1D→3413d05005.bmp
stimulus→images→PCB→34→1D→3413d06006.bmp
stimulus→images→PCB→34→1D→3413d23023.bmp
stimulus→images→PCB→34→1D→3413d24024.bmp
stimulus→images→PCB→34→1D→3422d09009.bmp
stimulus→images→PCB→34→1D→3422d09034.bmp
stimulus→images→PCB→34→1D→3423d25050.bmp

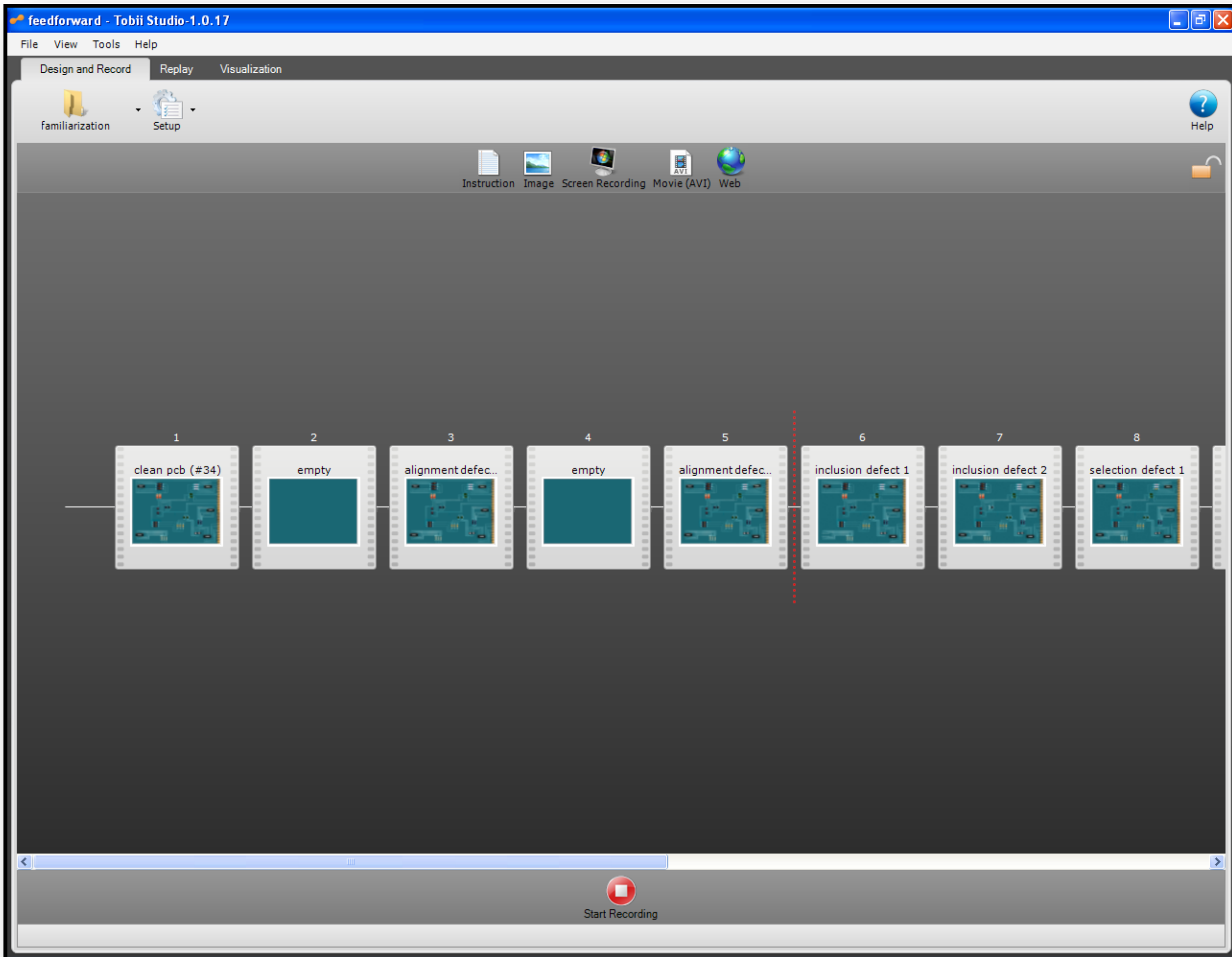
stimulus→images→PCB→34→1D→3423d28134.bmp

This forms the list of familiarization PCB images, with one defect on each board. The 12 example images show two examples of each of the six defects: alignment (misaligned components), inclusion (missing components), selection (wrong components), trace (wiring), polarity (reversed components), and edge (damaged board). Change the element name of each media element to reflect its content, overwriting the default ambiguous numerical name, e.g., use **alignment defect 1, alignment defect 2**, **inclusion defect 1**, **inclusion defect 2**, and so on.

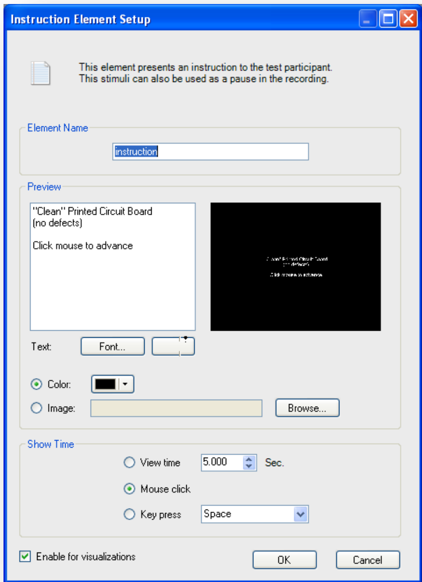


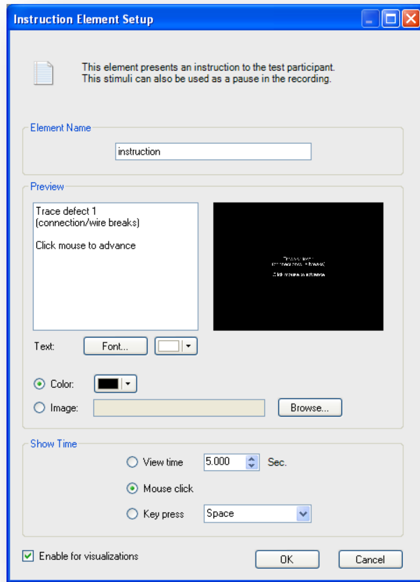
To prevent obvious "pop-out" effects when viewing boards in quick succession, insert a blank (**empty**) image in between every board image. Set the empty image to be machine-paced with duration of **0.500 s**. Inclusion of this empty "mask" image tends to reduce image after-effects.

4



The familiarization test is almost complete. All that remains is to provide the viewer with written instruction indicating what they should be paying attention to and how to interact with the display, in particular reminding them to press the mouse button to advance the presentation. Drop in an **Instruction** element at the beginning of the sequence and before each defect example. The instruction appearing before each defect example should indicate what the defect is (e.g., **Trace defect 1 (connection/wire breaks)**) and remind the viewer to **Click mouse to advance**.





The leading instruction element should introduce the participant to the entire sequence, e.g.,

Familiarization Stage

There are 6 types of defects: alignment, inclusion, selection, trace, polarity, edge

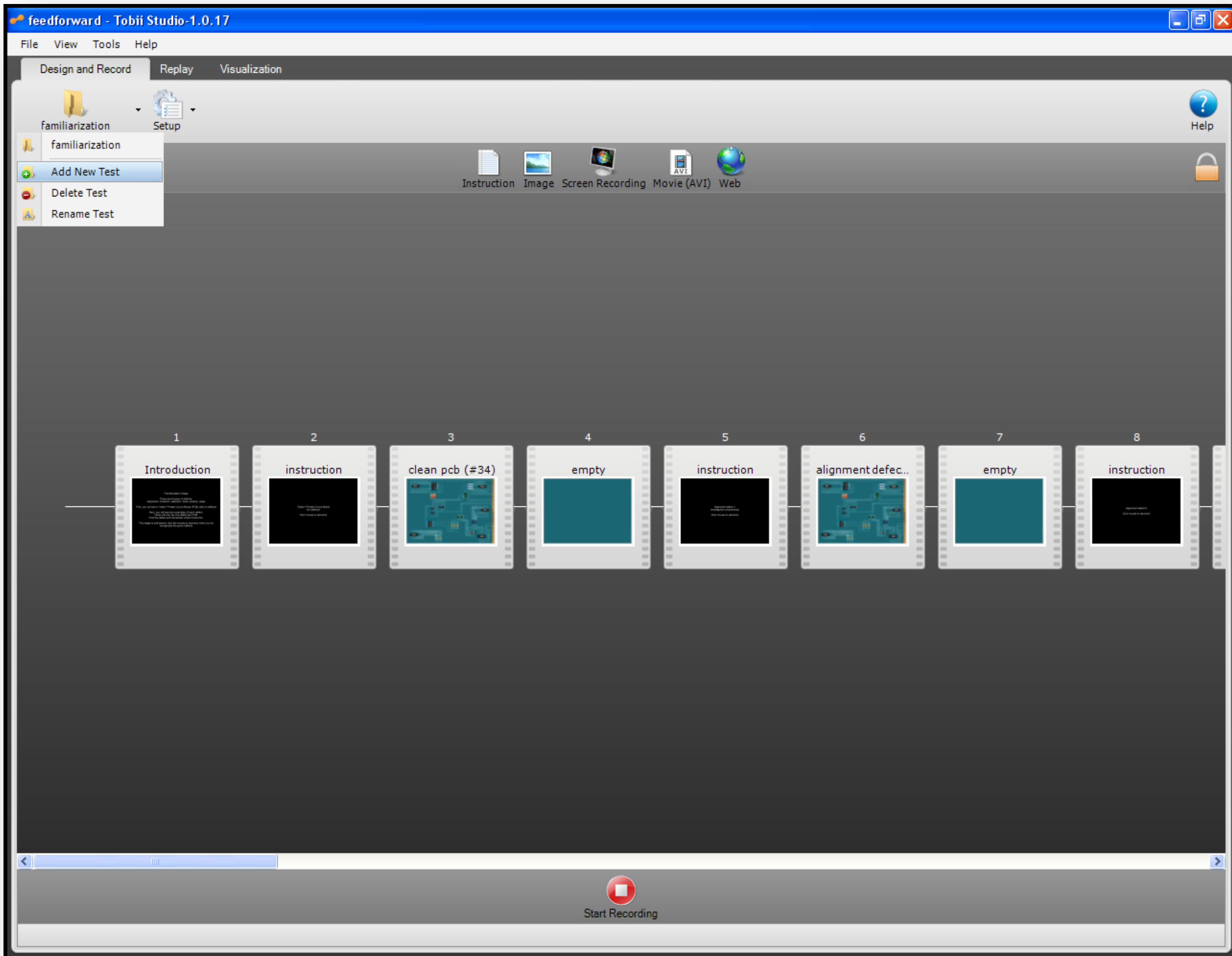
First, you will see a "clean" Printed Circuit Board (PCB) with no defects

Next, you will see two examples of

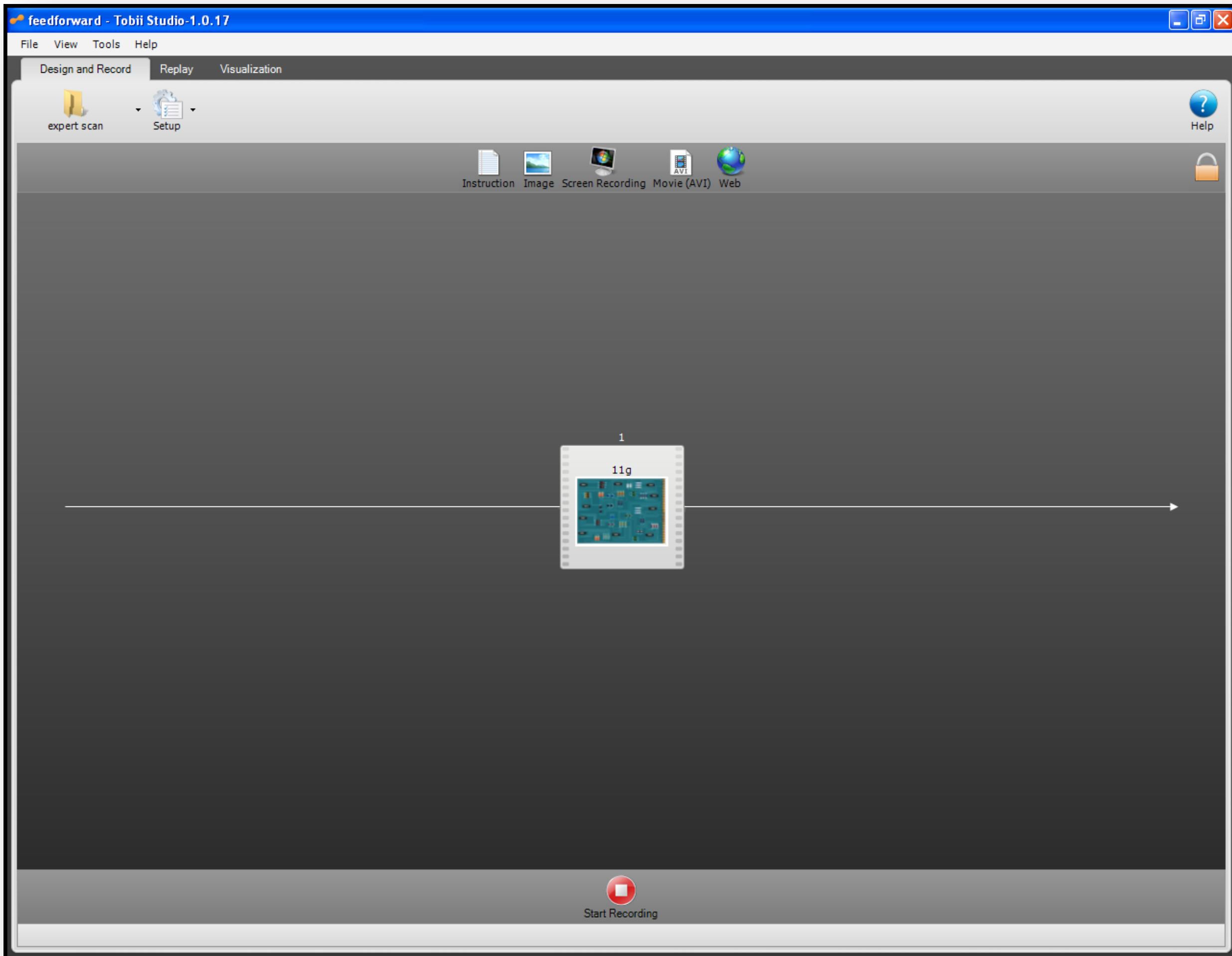
Run the familiarization test by pressing the **Start Recording** button at the bottom of the Studio window to make sure that you included the proper stimulus images and set the self- and machine-paced timings.

Design and Record: expert scan

The control and treatment tests are a bit shorter than the familiarization test, however, the treatment test requires an expert scanpath as its critical component (independent variable) differentiating it from the control test. So we need to create a special test to record the expert sequence. Using the **Open** menu from the **Design and Record** tab, select **Add New Test** and label it **expert scan**.

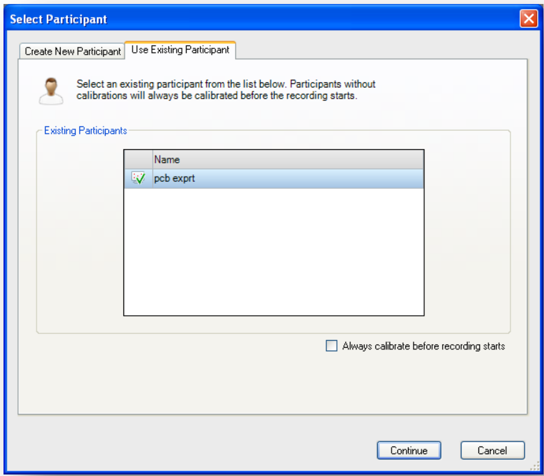


For the expert, choose the **stimulus→images→PCB→68→clean→11g.bmp** image of the clean PCB with 68 components. You should just have one self-paced image to display to the expert.

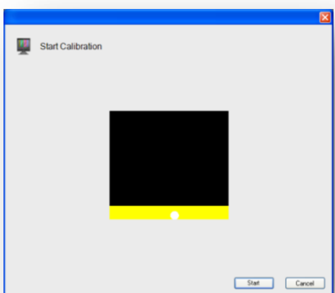


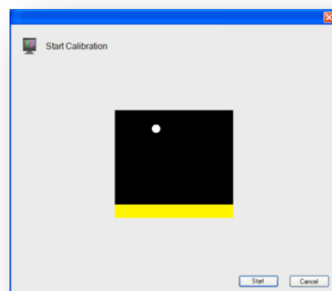
When the expert is seated in front of the eye tracker, press **Start Recording** to obtain the expert's eye movements over the test board. This board contains no defects, so instruct your expert to employ a fairly methodical scan pattern, e.g., left-to-right, top-to-bottom, fixating each component in succession. After scanning all components, perform a circular scan of the Integrated Circuit (IC) chips verifying that their chip numbers fall into a counter-clockwise increasing sequence (10001-10010). A final visual sweep of the trace elements (wire connections) should be performed, followed by a quick sweep of the board edges. The entire visual scan should take about 50-60 seconds. Several recordings may be required for the expert to generate this rather fast scanpath.

After pressing **Start Recording**, you will be asked to either **Create New Participant** or **Use Existing Participant**. This is an opportunity to add a new participant to the participant data base. When entering participant names, human subject protocols (e.g., as governed by Institutional Review Boards) generally require maintaining anonymity. You should therefore record subject information without including participants' names or other information that could potentially identify them. In the present example, a **pcb expert**participant was created earlier.



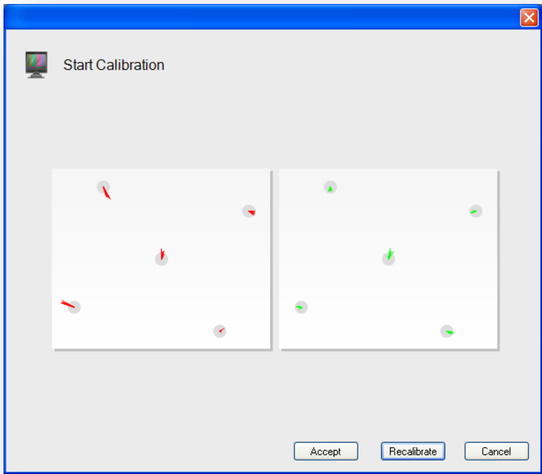
Checking the **Always calibrate before recording starts** checkbox ensures that calibration is performed every time, regardless of whether a previous calibration exists for this participant. Generally a new calibration is a good idea unless you can ensure that the given participant will be seated in almost the same position that s/he was in when the original calibration was taken. Since this is often difficult to guarantee, even in the best laboratory conditions, performing a new calibration is often prudent. When ready, press **Continue**. A track status window will appear showing the participant's eyes from the camera's point of view. Make sure that the eyes are centered and level on the screen with the participant sitting at an appropriate distance. If the participant is too far, the discs representing the eyes will be too low in the track status window; if the participant is too close, the discs will be too high up in the track status window.





A good track status, with both eyes visible to the eye tracker's cameras will show a green bar at the bottom. If one eye is not seen by the cameras, the status bar will show in yellow. If neither eye is visible to the camera, the bar will be shown in red.

Calibration consists of a colored disc moving from spot to spot on a blank screen. The intent is for the viewer to follow the disc as it moves and to fixate it when it stops periodically at one of 2, 5, or 9 calibration locations. The entire calibration step takes only a few seconds. At each stopped position, the camera records the direction of the viewer's gaze and calculates an error from the known location of each of the stopped calibration disc locations. When calibration is finished, Studio shows a summary of the calculated error for each of the left and right eyes at every stopped position of the calibration disc.



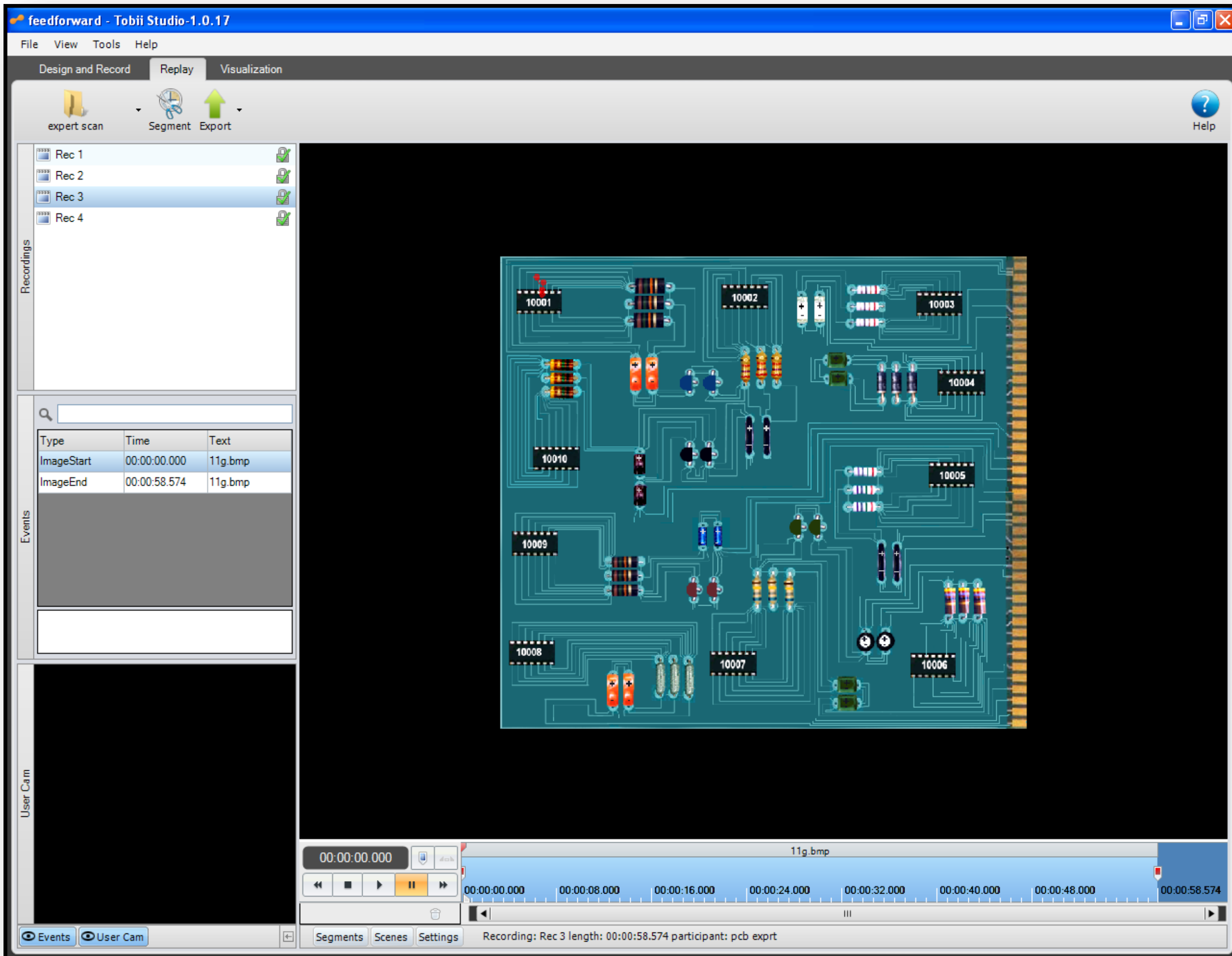
For an acceptable calibration, the error lines extending from the center of each calibration disc should be short. If there are really long lines extending from any of the calibration discs, the eye tracker should be recalibrated.

This is a rather important point: it is the eye tracker that is being re-calibrated, and not the viewer! Some participants may be concerned if the eye tracker has trouble calibrating to their eyes. This may be due to glasses worn by participants (with very shiny rims for example), heavy mascara, post-Lasik surgery (flattened corneas), or something more mysterious. In such cases, remember to be sensitive to participants' feelings: it is always easier to blame the equipment than the person. Poor calibration may mean that a participant's recording might not be used in the final analysis, but the participant need not know this.

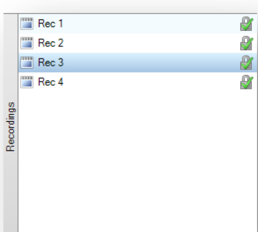
Once calibration is accepted, the eye tracker will record the expert's scanpath until the expert presses a mouse button.

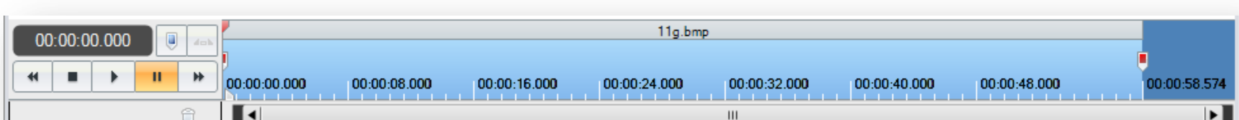
Replay: Expert Scan

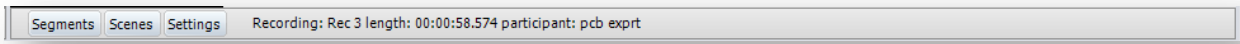
Select the **Replay** tab to inspect the expert's scanpath(s) (in this case 4 recordings were made).



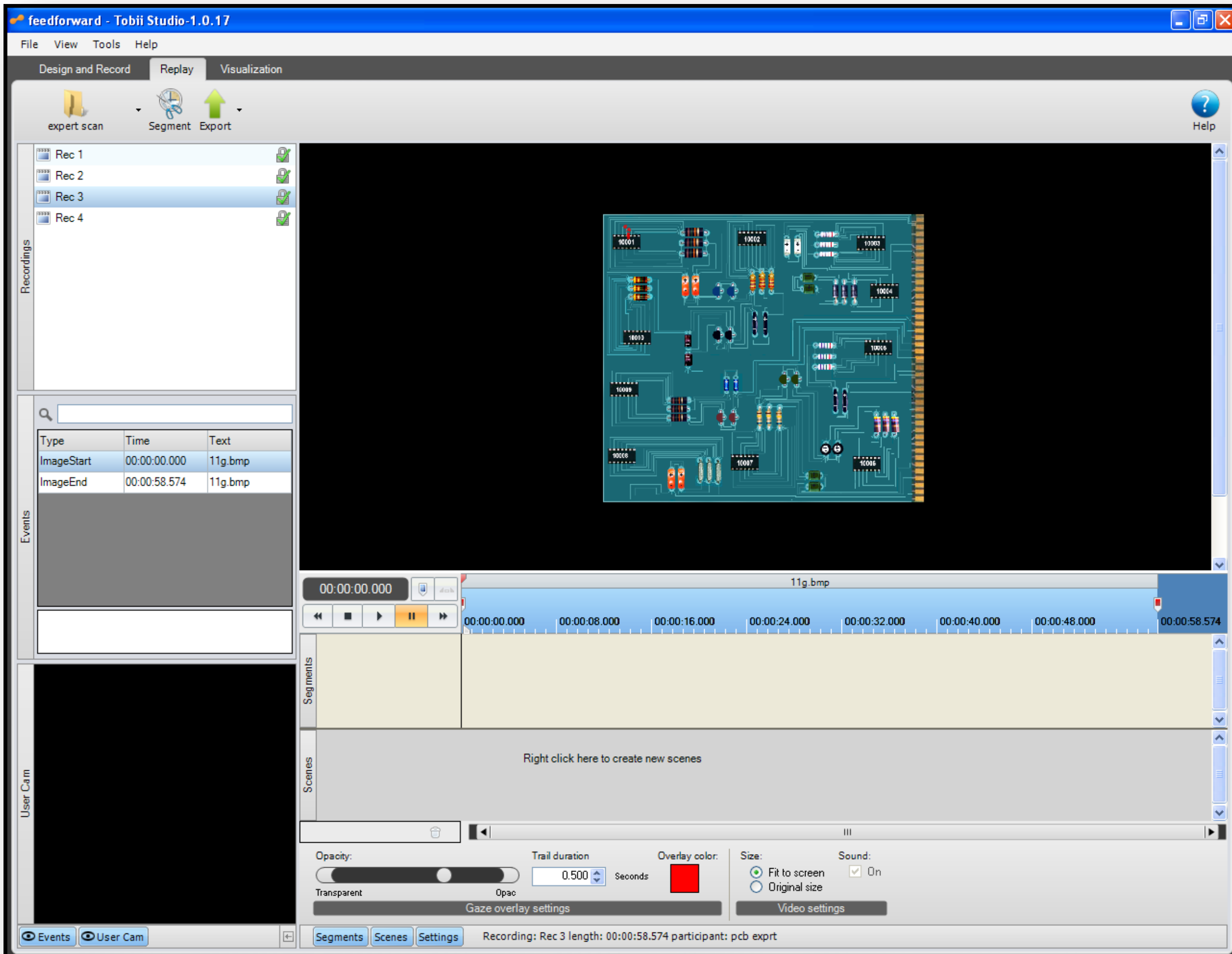
Choose a recording (there may be only one) from the **Recordings** pane and then press the play button using the VCR controls to play back the selected scanpath. Note the status bar and additional buttons below the VCR controls and timeline.



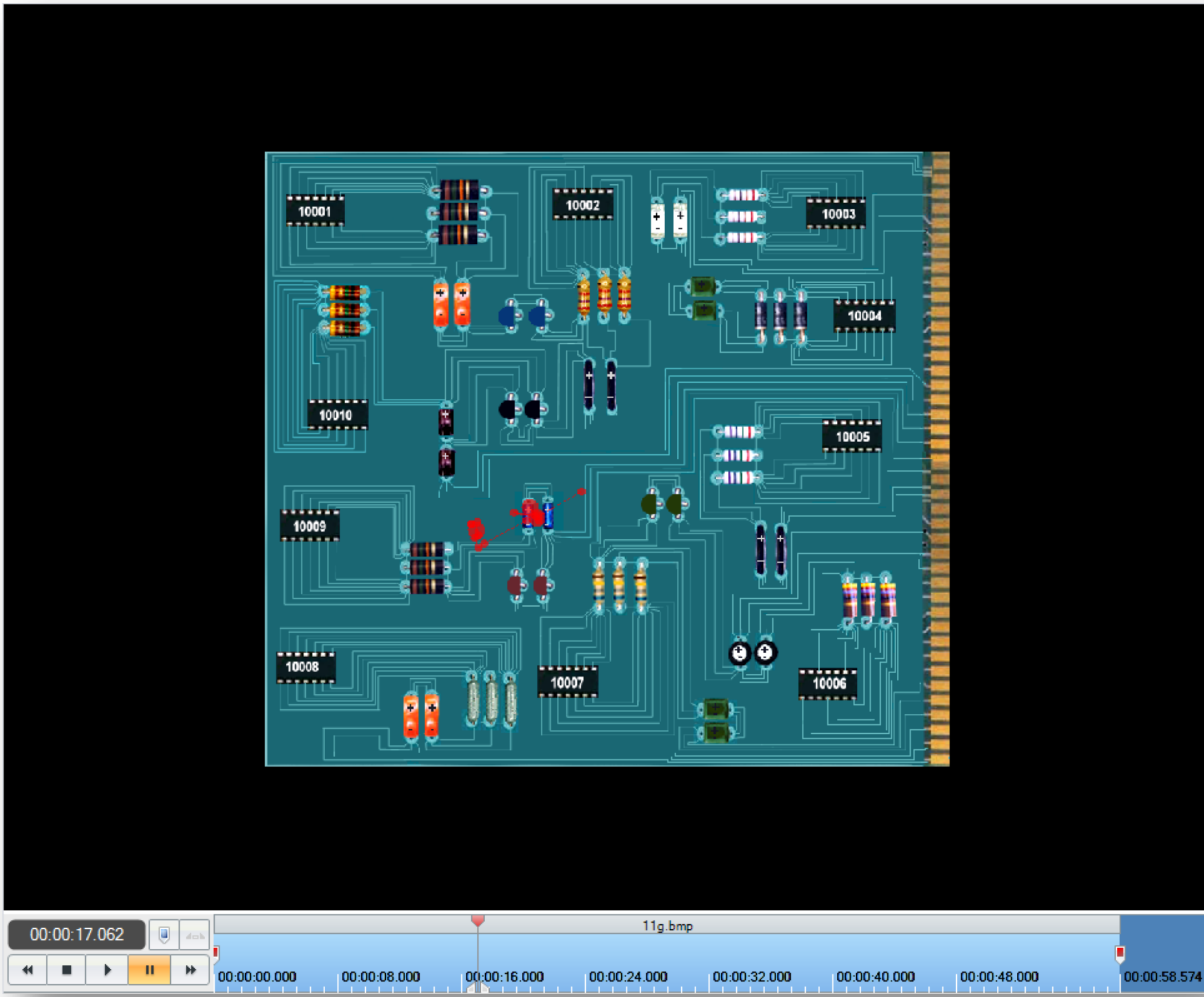




Each of the buttons below the VCR controls, namely **Segments**, **Scenes**, and **Settings**, expands to provide additional playback options.

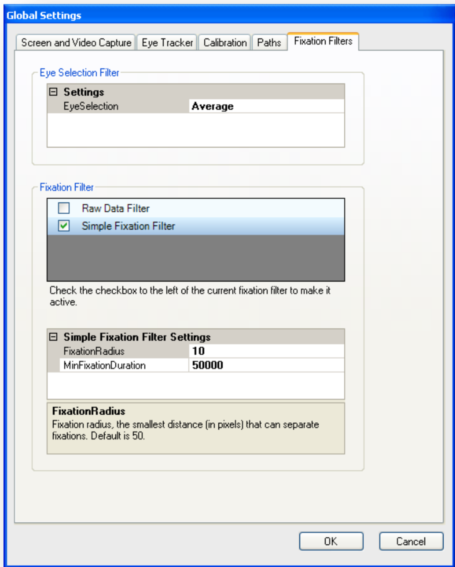


For example, the **Settings** buttons provides control over the appearance of the playback, including the color of the scanpath, the tail duration, and the opacity (transparency) of the display. Press the play button to see the dynamic replay of the expert's scanpath.



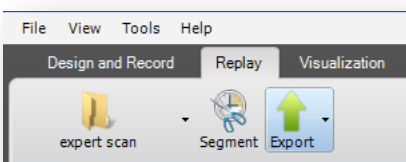
You should see small circles representing fixations moving from spot to spot as the expert fixated each element in the display. Fixations are joined by straight line segments that indicate saccadic jumps between fixations.

The default settings that govern estimates of fixation extents and durations may be a bit too large for the present case. That is, setting the radial extent of what Studio considers to be a fixation may obscure the detailed inspection that the expert was performing. To get a better fine-grained visualization of the expert's eye movements, choose the **Settings** option from Studio's **Tools** menu. Select the **Fixation Filters** tab from the **Global Settings** pop-up window. Then click on the **Simple Fixation Filter** checkbox in the **Fixation Filter** GUI group. Change the **FixationRadius** to **10** and the **MinFixationDuration** to **50000**.

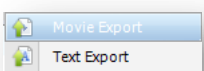


These fixation filter settings will reduce the size of the expert's fixation discs and show faster, shorter eye movements, providing a better visualization of the expert's inspection of individual board components.

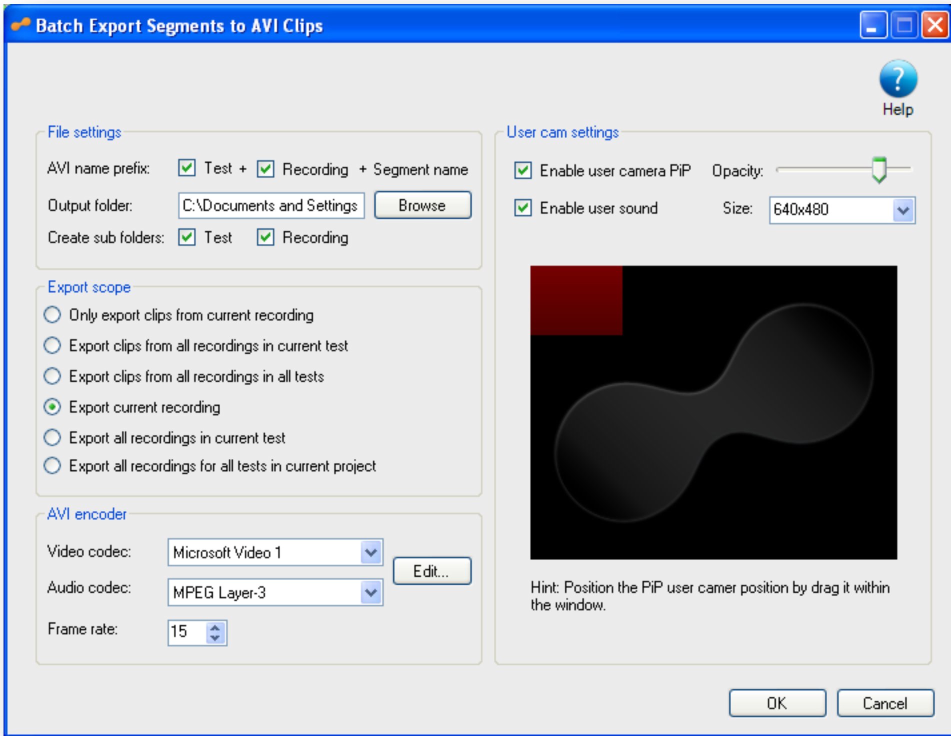
Once you are happy with the quality of the expert's scanpath—it should suggest a good, systematic strategy to the novice viewer—select **Export** from the **Replay** tab menu.



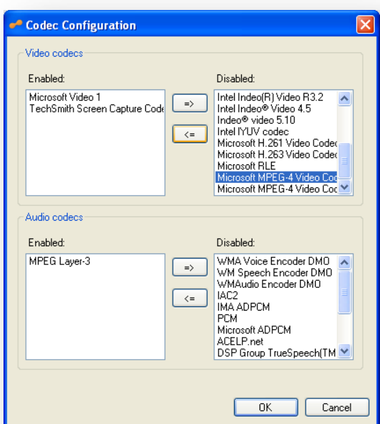
Select **Movie Export** from the pop-up menu.



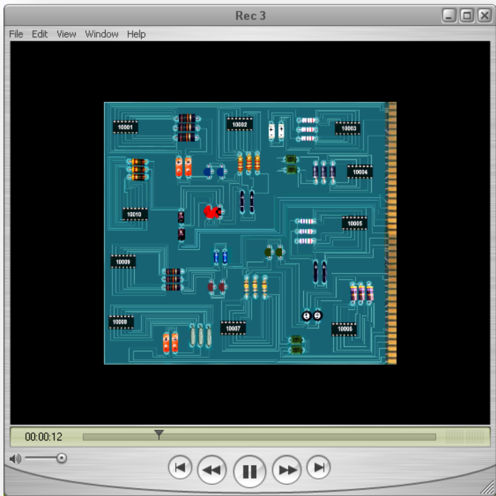
The subsequent pop-up dialog presents several options for movie export. With a specific recording selected in the **Recordings** pane, click the **Export current recording** radio button in the **Batch Export Segments to AVI Clips** to export that recording to an AVI movie clip. In this example, Rec 3 was chosen as the best representative scanpath made by the expert. Select the **640x480**, and choose an appropriate codec.



Microsoft Video 1 appears to be a good choice. If you want to use another codec, click the **Edit...** button within the **AVI encoder** GUI group. You can then add other codecs to the drop-down list if you think you might need them.

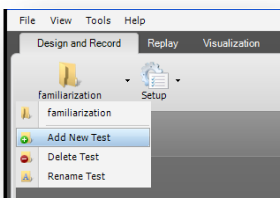


Export the clip by pressing **OK**in the **Batch Export Segments to AVI Clips** window. We will use the .AVI clip as the important feedforward scanpath in the control group test. When the export completes, be sure to play the clip in a media viewer such as the Windows or the VLC media player, or Apple's QuickTime.



Design and Record: control

In the **Desing and Record** tab, choose the **Add New Test** option to create a empty **control** test.

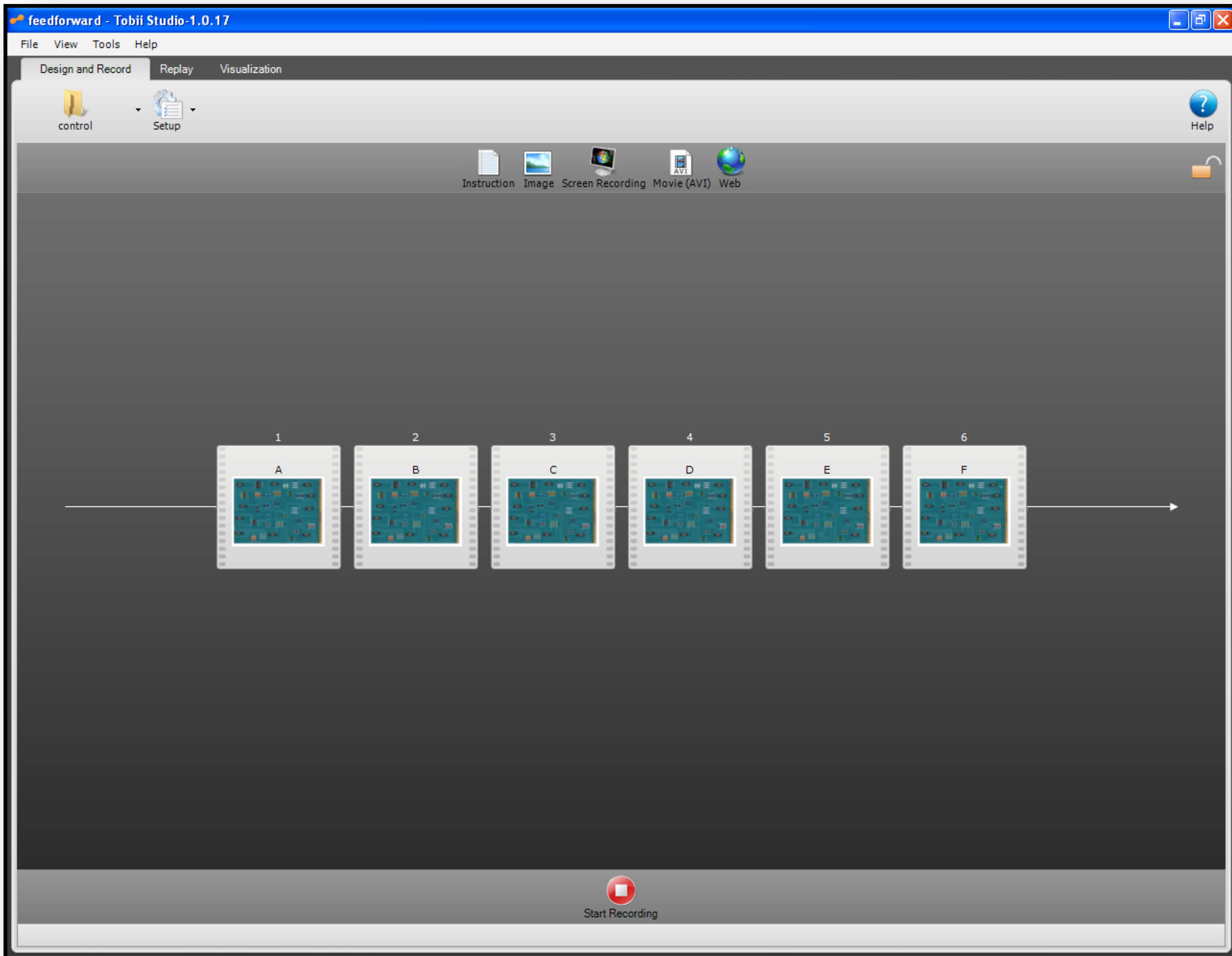


The treatment and control display conditions differ only in the display of the expert's scanpath. The treatment group will see the expert's scanpath recorded over the 68 element PCB, while the control group will only see a static image of the board without the expert's scanpath. Beyond this brief training phase, both test sequences will be composed of the same sequence of test images (or search fields as they are often referred to in visual search studies).

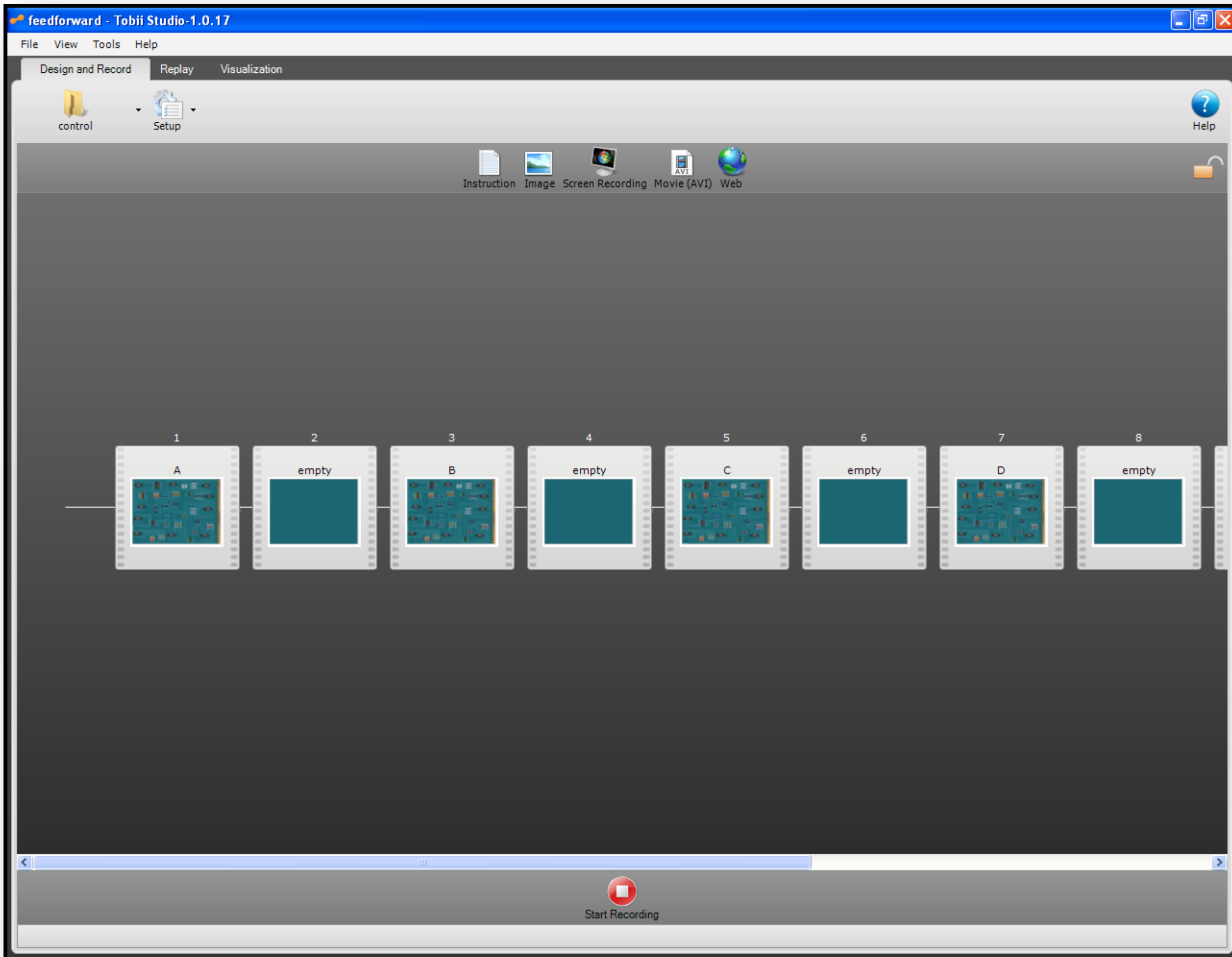
Because there are many possible search fields to choose from, we will restrict ourselves to those boards that contain 3 defects per PCB. Each image will simply be labeled by an alphanumeric character, starting with A. The list of chosen images is as follows:

A: stimulus→images→PCB→68→3D→22d072123237.bmp
B: stimulus→images→PCB→68→3D→22d082225238.bmp
C: stimulus→images→PCB→68→3D→22d092509482.bmp
D: stimulus→images→PCB→68→3D→22d132326476.bmp
E: stimulus→images→PCB→68→3D→23d0317233233.bmp
F: stimulus→images→PCB→68→3D→23d280927213.bmp

Using the **Image**media element drag and drop method we used to construct the familiarization test, construct the above sequence of still image stimuli.



Continue to drag and drop **Image** media elements, inserting the **empty** image after each stimulus image. Remember to set each image element for machine-paced display, with each stimulus image displayed for 60 seconds and each empty mask image displayed for 0.500 seconds.



Next, drag and drop the clean **stimulus→images→PCB→68→clean→11g.bmp** image to the front of the sequence, followed by an empy image element. Finally, prepend to the sequence an Instruction media element with the following instructions:

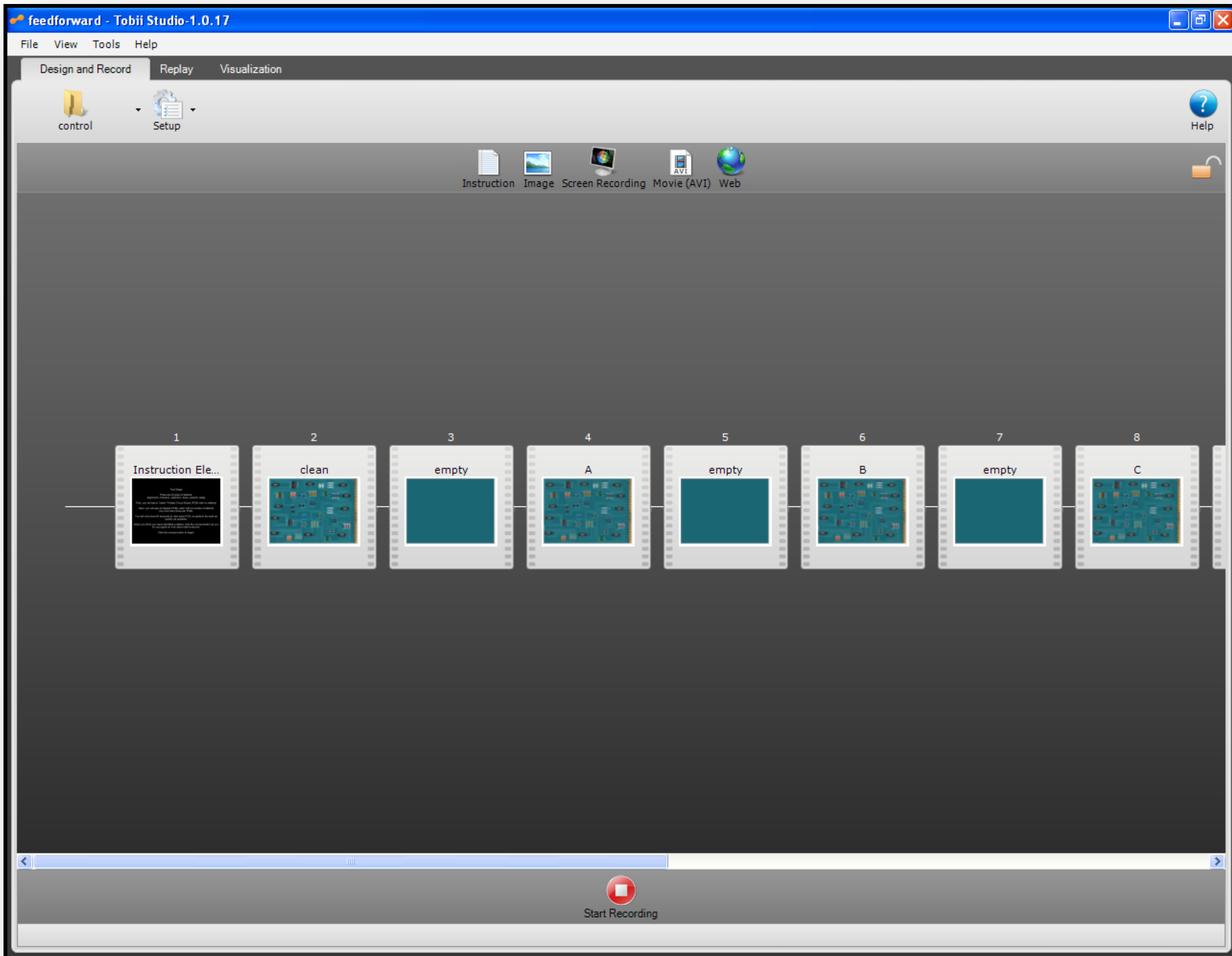
Test Stage

There are 6 types of defects: alignment, inclusion, selection, trace, polarity, edge

First, you will see a "clean" Printed Circuit Board (PCB) with no defects

Next, you will see six flawed PCBs,

This completes the control sequence.

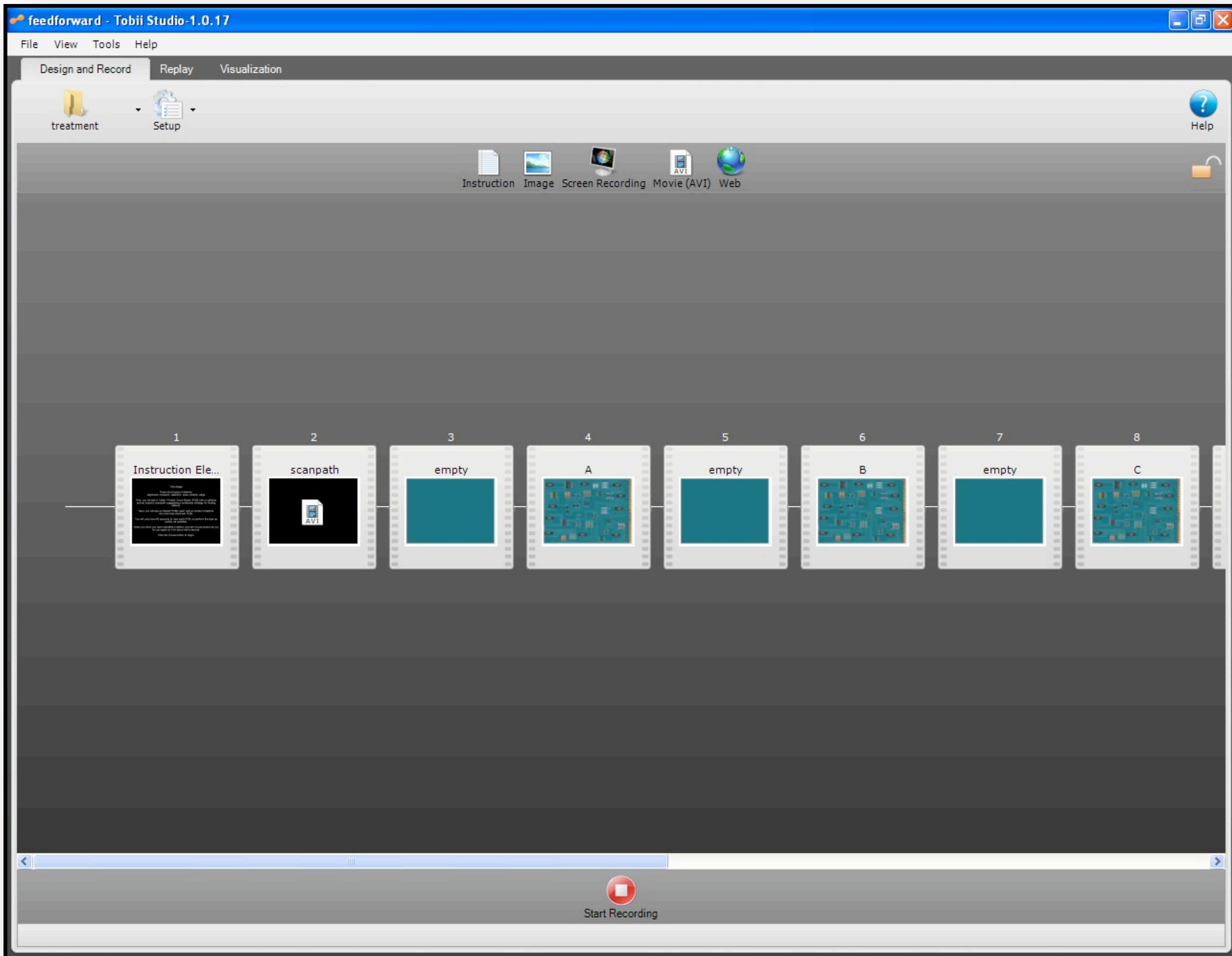


Note that the image element sequence order ABCDEF is but one order of viewing the images. To counter order effects (e.g., fatigue, learning), the order of stimulus presentation should be presented in a counterbalanced fashion. For example, a Latin square can be used to provide six sequence orderings, requiring at least 6 participants:

ABCDEF
BCDEFA
CDEFAB
DEFABC
EFABCD
FABCDE

Design and Record: treatment

The treatment sequence is identical to the control sequence except that it replaces the clean static 68-component PCB image with the expert's scanpath. The scanpath .AVI video clip contains the same PCB image, however, hence the only real difference is the expert's visual scanning performance. The control stimulus sequence is constructed in exactly the same way as the control sequence, except that instead of inserting the clean image, one uses the **Movie(AVI)** media element selection to insert the expert's scanpath video.

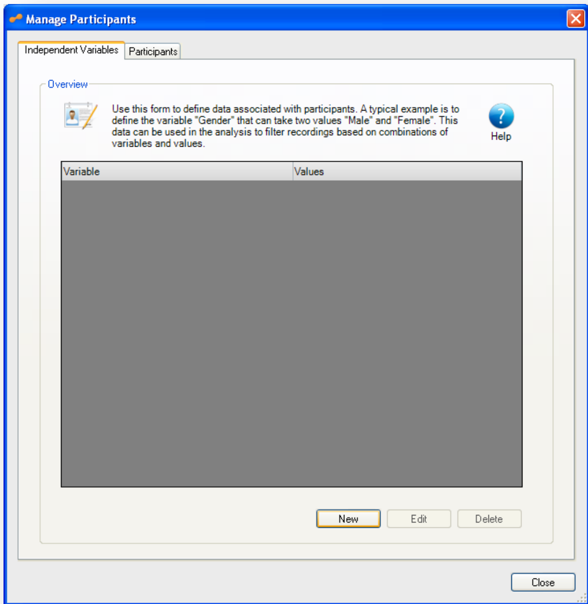


The instructions to participants differ only slightly.

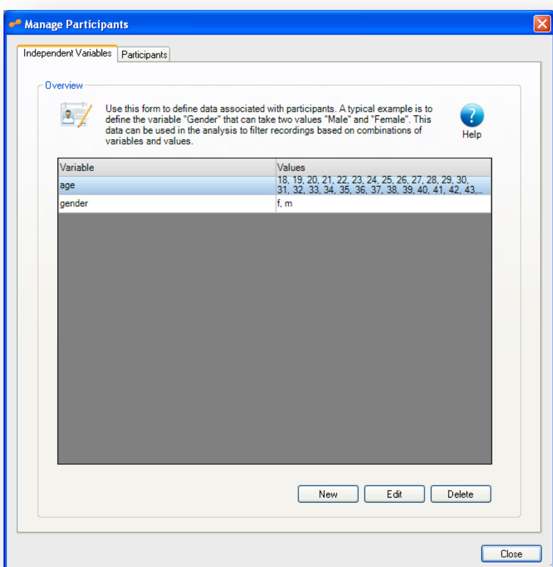
Test Stage **There are 6 types of defects: alignment, inclusion, selection, trace, polarity, edge** **First, you will see a "clean" Printed Circuit Board (PCB) with no defects and an expert's scanpath suggesting a systematic strategy for finding defects.**

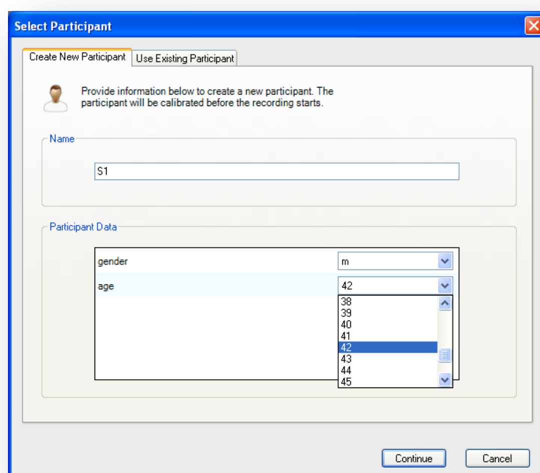
Running the Study

When running the study, take care to collect anonymous demographics data from participants. One generally cannot store personal information such as names, addresses, etc. for privacy concerns, however, data pertaining to age and gender should be stored. Studio provides a mechanism for defining arbitrary data along with possible data values. This should be set up before any participants are asked to sit in front of the eye tracker. From the Design **and Record** tab's **Setup** menu, choose **Manage Participants**.



This pop-up dialog allows the specification of "Independent Variables" describing participants' demographic data. Press **New** to create a new variable. As an example, type in **age** in the **Variable Name** field. In the **Values** list, type in **18** then press the down arrow key to add in another value for age, **19**. Repeat this until you have a list of possible (expected) ages of the participants, e.g., 18-50.





Repeat these steps for variable gender and values m,f. Whenever a new participant is added, you will just select the appropriate instances of age and gender for that participant. Remember to use anonymous names, e.g., S1.

For reasons of generalizability, the experiment would typically involve as many participants as possible to increase statistical power. In this "fully cooked" example, however, in reality there was only one participant pretending to act as member of both control and treatment groups. For some semblance of statistical analysis, each sequence was viewed twice. Nevertheless, the data obtained serves as an illustrative example of what may really have happened with real participants.

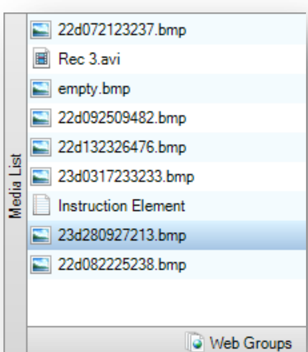
Visualization: Analysis and Results

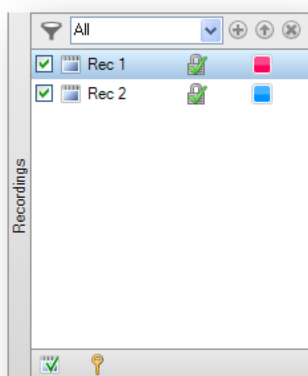
The goal of the analysis of this experiment is to determine whether the expert's scanpath had any effect on the participants' performance and to use captured eye movement data to examine why. Typically performance would be measured in terms of dependent measures such as speed and accuracy. Because the experiment involved machine-paced stimulus display, however, only accuracy measures make sense, i.e., number of defects found. Ultimately we would like to examine rates such as false positives, but here we will restrict the analysis to just true positives.

The main result we are after is statistical support for our hypothesis. Note that this does not constitute proof of any kind (scientific or otherwise), but merely provides evidence for acceptance of the hypothesis. More formally we would have declared the hypothesis in the statistically neutral sense by specifying the null hypothesis, H_o, which states that no effect is expected. Statistical analysis of the data then would either accept H_o or reject it in favor of the alternative, which is what we are hoping for. The statistics we use essentially state that if there is a significant difference in the means of the dependent variable, it is due to the effect and not chance alone, therefore there is a statistically significant reason for rejecting the null hypothesis.

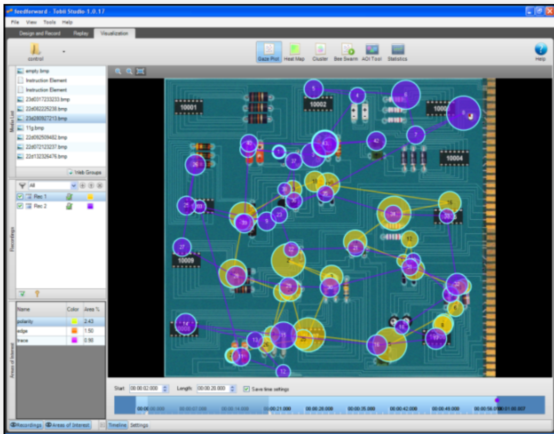
Before "crunching" the data, we can use Studio's visualization tools to see if this effort is worthwhile. Intuitively, if the control group missed defects, we should see some defects not selected by participants and possibly not seen by them. Thus we would expect to see an absence of fixations over defects as well as a lack of mouse click events over those defects. Studio provides several visualization tools, including **Gaze Plot**, **Heat Map**, **Cluster**, **Bee Swarm**. Some statistical analysis is available by defining Areas Of Interest, or AOIs, with the **AOI Tool** and the **Statistics** button. For the present case we are mainly interested in comparing the aggregate eye movement data of the control and treatment groups.

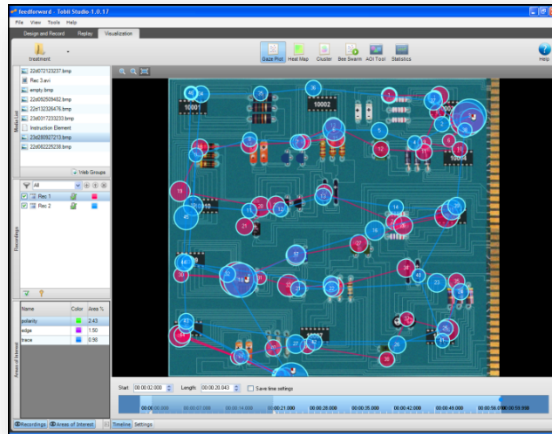
For any of the visualizations available, use the Media List to select the stimulus image on which to render the graphics. For example, image **23d280927213.bmp** is the sixth (F) image in the sequence and contains the polarity, trace, and edge defects. For each of the control and treatment visualizations, both recordings are used from which to obtain the aggregate visualizations.



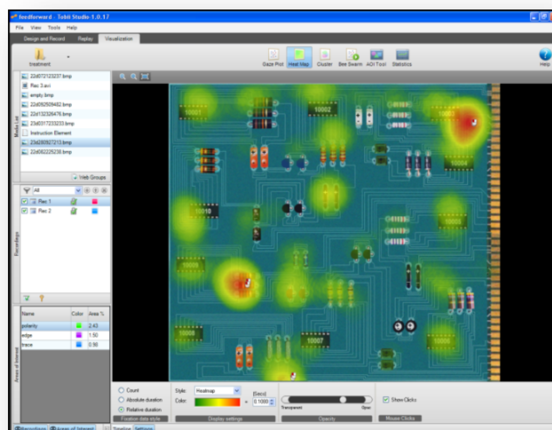


Using the Gaze Plot button, we can look at the general pattern of eye movements exhibited by both control and treatment groups.

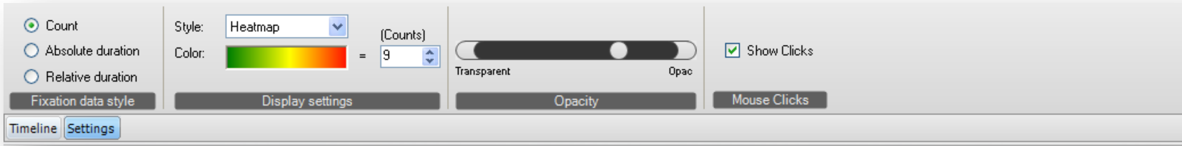




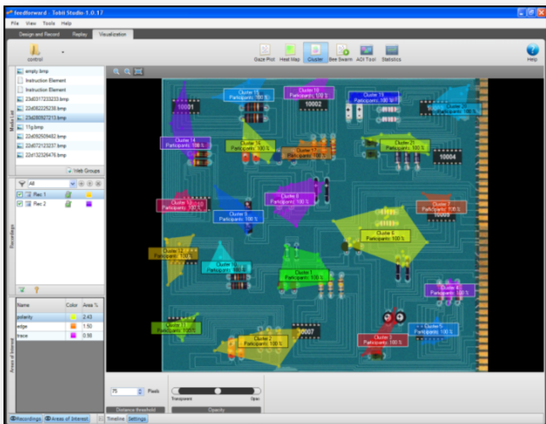
If there is a large number of participants, gaze plots can become somewhat congested. Use the **Timeline** below to restrict viewing of the gaze plot to a limited period of time. In this case, examine the gaze plots captured over the sixth (F) image stimulus above, both restricted to the first 2-20 second time period. The left gaze plot belongs to the control group, the right to the treatment group. Notice that, as expected, the treatment group appears to be viewing the image in a left-to-right, top-to-bottom pattern as indicated by the expert's scanpath. Meanwhile the control group's gaze plots do not seem to be exhibiting any kind of pattern but are more scattered and disorganized. On the plots you can also detect small mouse icons indicated where participants have clicked to select targets. These are particularly visible on the treatment gaze plot, at the lower left and upper right portions of the image. Heat map images reveal the relative distribution of fixations over the entire 60 s viewing duration.

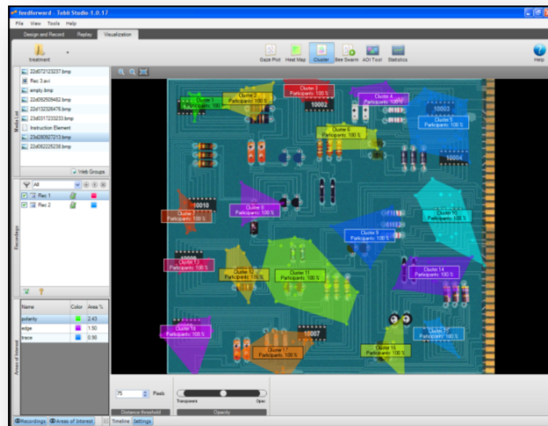


The relative distribution of fixations (aggregated from multiple viewers once again over the sixth (F) stimulus search field) shows participants responding to the given instruction: longer dwell times are visible over detected PCB defects. Notice, however, that the control group seems to have missed the edge defect at bottom left. Notice that in this instance the relative number of aggregate fixations is shown. The **Settings** button below allows depiction of **Absolute duration** of fixations as well as just their **Count**, along with their **Relative duration**. Other options allow selection of heatmap colors, transparency, as well as the inclusion of mouse clicks.



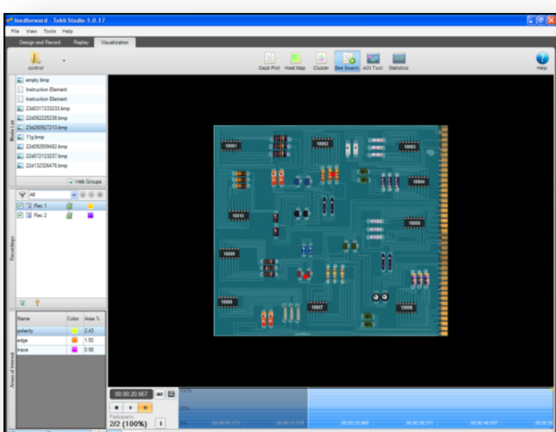
Note that eye movement analysis generally involves classification of raw data into fixations and non-fixations (saccades) since the raw data tends to be rather meaningless (although visually tantalizing). Fixations, however, can be clustered further into groups, generally via some sort of algorithmic approach. Such visualizations can show further high-level abstraction of the raw data.

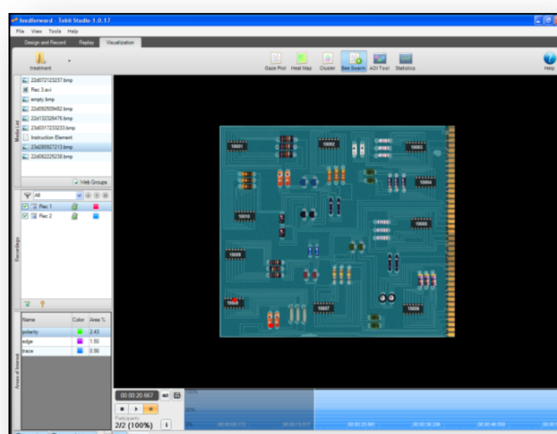




At first glance the cluster visualization just shows that participants are generally inspecting the Printed Circuit Boards in at least an intuitive manner: by fixating the PCB components. Upon closer inspection, one can see what was evident in the gaze plots, namely that the sequence of clusters appears to follow that of the expert sequence. However, this is no more than anecdotal evidence and cannot be readily used to infer the significance of the effect of the expert's scanpath on novices adopting the strategies, it just appears they may have done so. More sophisticated approaches are needed.

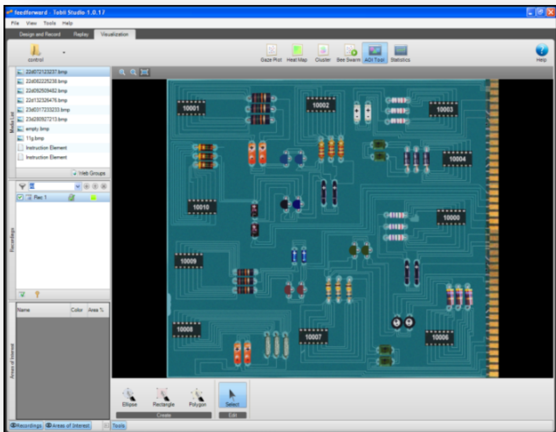
Note that while clusters show spatial grouping of fixations, **Bee Swarm** data shows such tendencies in time. That is, if the treatment group had adopted the expert's sequence, in space as well as in time, then most participants would be fixating at the same AOIs at the same time.

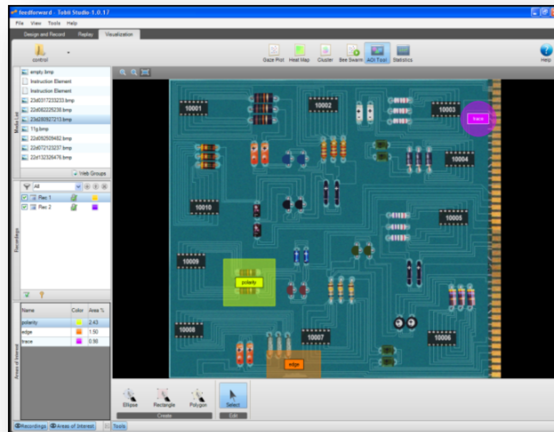




The present cooked example only contains data from two runs in each control and treatment condition. This is hardly enough for effective display of swarm data. Still, at the same instant in the Timeline, fixations of the control participants show disparate locations. Meanwhile, the two participants in the treatment group seem to be fixating nearly in the same location at the same instant. Once again, in the present instance there is hardly enough data for the above visualization to be meaningful. Temporal visualization provided by the Bee Swarm will likely show interesting trends, particularly with large sample populations. For the present case, statistics are still required to determine statistical significance of the effect, if any.

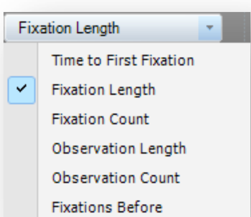
As a preliminary tactic toward statistical analysis, the **AOI Tool** and **Statistics** buttons can be used to generate some simple plots. The aim here is to generate statistics describing eye movement metrics per Area Of Interest. In this case, the AOIs are fairly obvious, namely the defects. For example, we would like to obtain statistical analyses regarding the number of times each AOI was seen by each group of participants. The first step is to define the AOI regions per stimulus image.

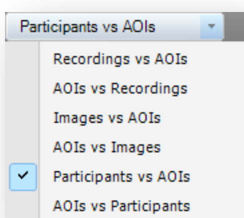




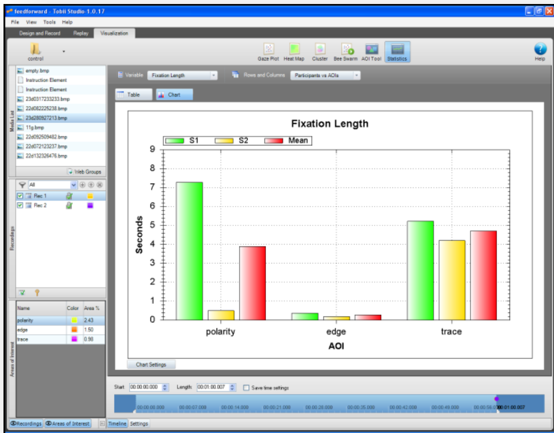
Use the **Create** and **Edit** tools to define, position, and shape AOIs on the stimulus image. The **Ellipse**, **Rectangle**, and **Polygon** are fairly self-explanatory shapes not uncommon to typical drawing packages. Use the **Areas Of Interest** pane at bottom left to provide informative names to the AOIs, e.g., polarity, edge, and trace to identify the known defects on the given PCB.

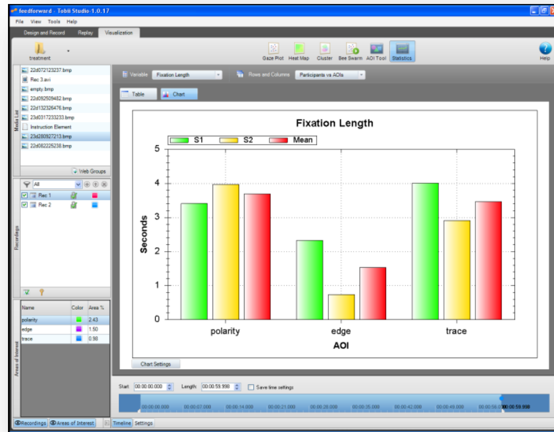
With the stimulus and recordings selected in the **Media List** and **Recordings** panes, and AOIs defined, use the **Statistics** tool to generate plots of eye movement data per AOI. To specify the type of data to display, use the **Variable** and **Rows and Columns** drop-down lists to specify the plot to generate.





For example, use **Fixation Length** as the variable of interest and **Participants vs AOIs** selection to generate data plots for the control and treatment groups.





The plots show that the fixation lengths appear shorter over the edge defect for both control participants and are shorter on the polarity defect for one of the control participants. It appears that both control participants fixated the trace defect. Both treatment participants seem to have fixated all three defects. Because the instructions to participants asked that participants fixated defects as they counted them, it appears that at least over this particular stimulus there is some difference in the way the groups (small as they were in this cooked example) performed. However, these plots are mainly descriptive in nature and do not offer a means for making any further inferences. Statistics are still required to draw further meaning from the data.

As an example, consider tabulating the number of defects detected by participants under each condition as an aggregate measure of accuracy. As it happened, all participants in the treatment group (2) managed to find all defects (3) per stimulus bard. Meanwhile, the control group participants did not fare as well. The data is shown in Table 1.

TABLE 1: NUMBER OF DEFECTS FOUND

	Control	Treatment
A	5	6
B	2	6
C	3	6
D	4	6
E	3	6
F	3	6

Graphically, the data can be replotted as show in Figure 2. The bar plot shows average defects found with whiskers representing standard error.

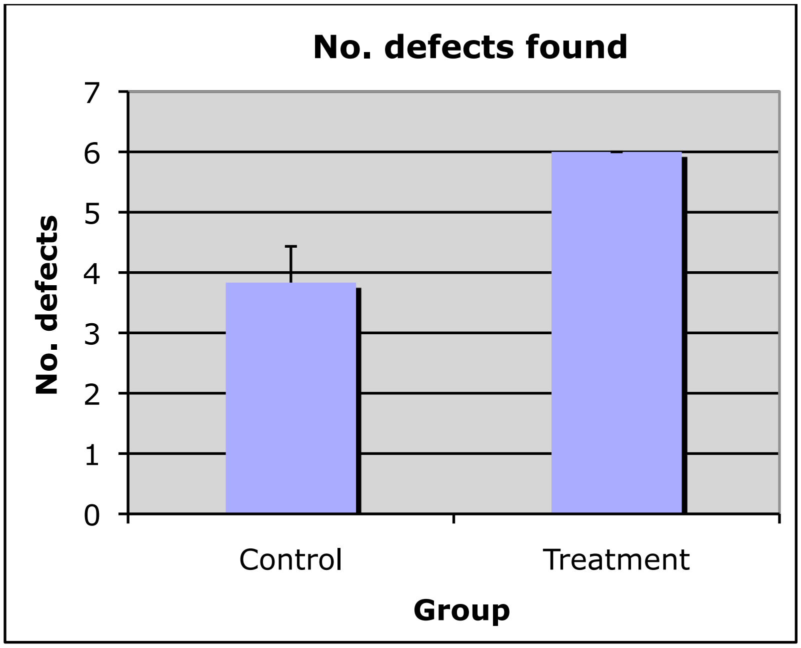


FIGURE 2: NUMBER OF DEFECTS (AGGREGATE)

Because the number of defects can be considered parametric data, Student's two-sample *t*-test (assuming unequal variances) can be used to evaluate whether the average number of defects found in each condition differs significantly. The between-subjects effect appears to be significant ($t(df=5) = -3.60$, $p < 0.05$). Keep in mind this is only a simple example with a small sample size. Normally we would want a larger number of trials an subjects to bolster confidence of the results. Still, the analysis shows that in this case (cooked example) the behavior of the participants (or rather, participant), differs significantly.

The example, as developed, shows a potentially powerful "recipe" for conducting a meaningful eye tracking study with static images. The PCB data set is rather rich and offers a myriad of analytical approaches, e.g., effect of feedforward training, difficulty of visibility of individual defect types, and so on.

Testing the Visibility of a Moving Target in a Video Presentation

Video is a naturally compelling type of stimulus to want to study. However, its dynamic and voluminous nature makes it one of the more difficult types of data to work with. The main complication is synchronization of eye movement data to the moving objects within the video stream. In a word this is known as the registration problem and it poses unique challenges for analysis. Further complications are drawn from a myriad of video formats and file types exemplified by a large number of coders/decoders or *codecs*. There are numerous codecs for compressing video data (e.g., MPEG, Cinepak) as well as audio data (e.g., PCM, mp3) that accompanies it. Video and audio streams are encapsulated by container file formats, of which there are several varieties (e.g., .asf, .mov, .avi). Currently there are few analytical options for analysis of eye movements over video. Nevertheless, there is certainly a great deal of potential for interesting work.

Background

For an example of a challenging eye tracking study, we revisit a classic experiment exploring inattentional blindness. In their well-known work on this topic, Simons and Chabris describe inattentional blindness as the phenomenon of failing to perceive an unexpected object, even if it appears at fixation, when attention is diverted to another object or task(Simons & Chabris, 1999). That is, when engaged in some unrelated activity, one can "look at" but not "see" an object, even if gaze happens to fall over the object. Simons and Chabris demonstrated this phenomenon during a video-viewing task. In the video, two teams of three basketball players are passing a basketball to each other. One team wears white shirts the other black. After some time (Simons and Chabris provide the specifics omitted here for brevity) an unexpected event occurs: a person wearing a gorilla suit passes through the players. Depending on the task assigned to viewers, up to 92% of the viewers could miss perceiving the unexpected event. The percentage of viewers noticing the unexpected event when viewing a superimposed video of the teams and gorilla is given in Table 2. The tasks were differentiated by twoaspects. First, viewers were instructed to attend to either the white-shirted players or the black-shirted players. Second, the task was made easy when asked to count all passes made by each team to each other. The task was made difficult when asked to count only bounce passes (ignoring the aerial passes). The players as well as the unexpected event appeared to be somewhat transparent due to the nature of overlapping (composited as it were) video streams.

TABLE 2: PERCENTAGE OF VIEWERS NOTICING THE UNEXPECTED EVENT IN EACH CONDITION.

Easy Task		Difficult Task	
White Team	Black Team	White Team	Black Team
8	67	8	25

Simons and Chabris' gorilla video was based on an earlier but similar video presentation by Neisser and colleagues where instead of a gorilla, a woman holding an umbrella appeared to walk through the two teams of basketball players(Neisser & Becklen, 1975). Of twenty-eight naïve observers performing the pass-counting task, only six reported the presence of the umbrella woman, even when questioned directly after the task. When participants just watched the video without performing any task, they always noticed the umbrella woman.

A particularly interesting question is how many times the unexpected event is fixated by participants reporting not to notice the unexpected event. That is, eye tracking could be used to explore the "looking but not seeing" aspect of the inattentional blindness phenomenon. (An example of such a study was presented as a poster at the 2005 Human Factors and Ergonomics Society meeting (Pappas, Fishel, Moss, Hicks, & Leech, 2005)).

Hypothesis

In an attempt to replicate previous work, we can re-use the hypothesis employed by previous researchers, namely that given the task to attend to the white-shirted players, only a small percentage of viewers is likely to notice the unexpected event (or conversely, a larger proportion of viewers will notice the unexpected event if attending to the black-shirted players). Furthermore, we can then hypothesize that the viewers attending to the black-shirted players will devote a larger proportion of fixations to the unexpected event (thus stipulating one of our dependent variables: number of fixations over target).

Experimental Design

To keep this example experiment tractable, we will restrict ourselves to the easier task of simply counting all the passes made by either team. That is, we will only consider the easy task and not bother with the difficult task. Hence our experimental design is between-subjects where we simply have two groups, one assigned to attend to the black-shirted players, the other assigned to attend to the white-shirted players.

STIMULUS

The key to this demonstration is the availability of the video stimulus. Courtesy of Ulrich Neisser, we can use a digital video clip of the tape he originally used in his studies (Professor Neisser holds the copyright to the material, hence permission was obtained prior to including it in this example). To do so, we must ensure that it is in the proper video format for display by our software. Tobii Studio is currently capable of playing back AVI movie clips (containing Cinepak or MPEG4-encoded video with PCM-encoded audio; Studio may eventually support a larger variety of formats, but presently these combinations are known to be supported). Prior to setting up our single trial test, we can attempt to preview the video via a movie playing application such as Windows Media Player. If the video plays in Media Player, chances are that it will play in Studio as well.

APPARATUS

Once again we will use Tobii's Studio (v1.0) software and a Tobii ET-1750 video-based corneal reflection eye tracker (or the faster T120 if available). The Tobii ET-1750 operates at 50 Hz at an accuracy of about 0.5 degrees visual angle (bias error). (The T120 operates at 120 Hz.) At the time of testing, screen resolution was set to 1280x1024. The eye tracking server ran on a Sun model w2100z dual 2.0 GHz AMD Opteron 246 machine equipped with 2 G RAM. The server was running Windows XP.

STUDIO

As with the visual search study, we create a new project, providing a project name and creator identity. Once you provide a project name (e.g., **video**), Studio will automatically create an instance of an empty test. We will explore two ways in which to use Studio to carry out this experiment.

DESIGN AND RECORD: VIDEO

As with our previous test examples (expert scan, familiarization, etc.), in our first attempt at setting up an eye tracking test, drag and drop a **Movie (AVI)** media element to the test. Select the **stimulus→videos→umbrella woman→neisser_umbrella.avi**video clip. There is only one video clip, however, as you will see below, we need to decide which is better: a Studio test per group of between-subjects participants, or one common Studio test for both groups.

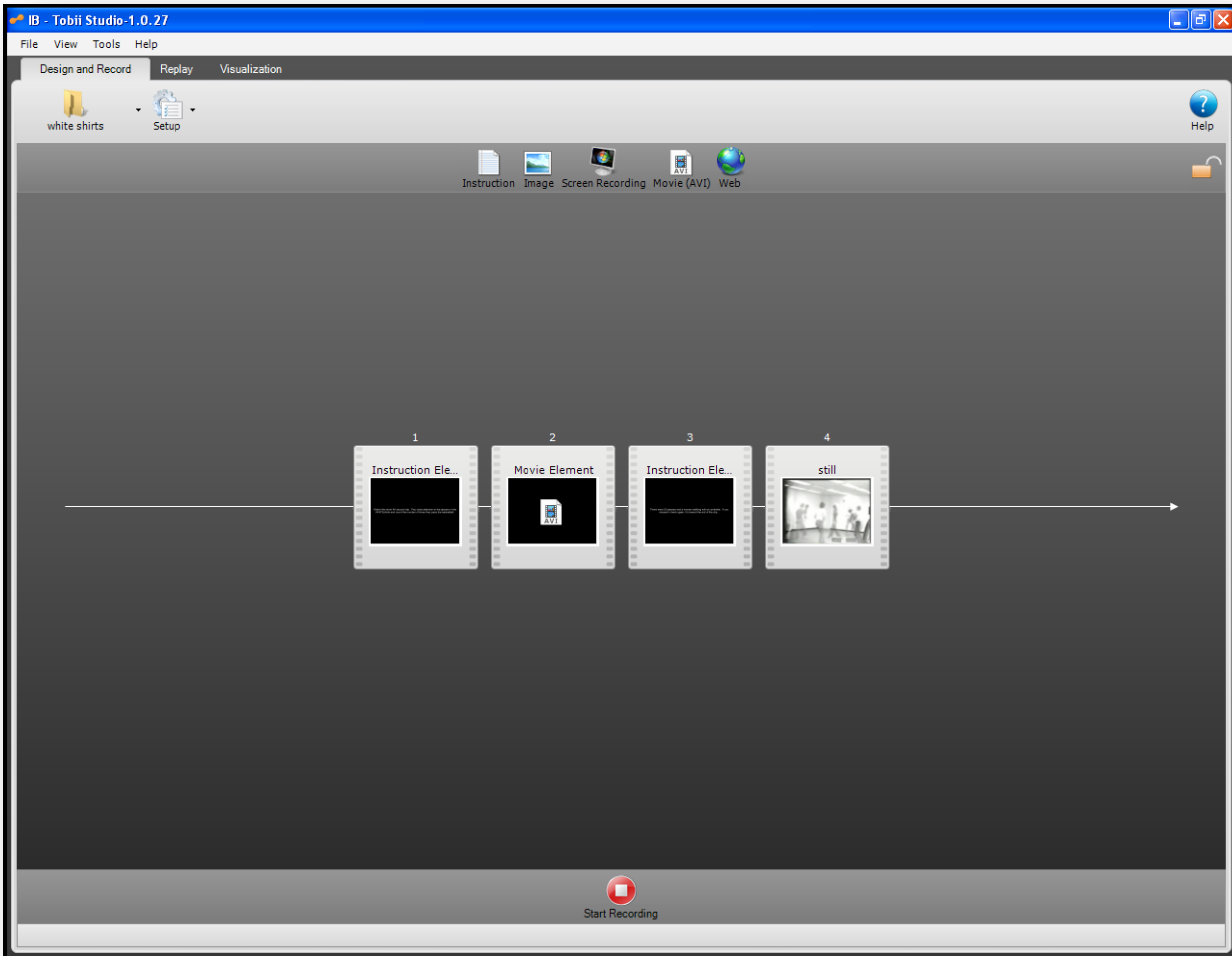
First, we will create one Studio test per between-subjects groups, wherein one group will be asked to attend to the black-shirted players, the other group to the white-shirted players. Use the **Instruction** element to prepend the video clip with instructions to either watch the white-shirted or black-shirted players, e.g.,

Watch this short 40 second clip. Pay close attention to the players in the BLACK shirts and count the number of times they pass the basketball.	**Watch this short 40 second clip. Pay close attention to the players in the WHITE shirts and count the number of times they pass the basketball.**

Following the video clip, append another instruction to debrief the viewer to tell them what they just saw, e.g.,

There were 20 passes and a woman walking with an umbrella. If you missed it check again, it's toward the end of the clip...

Finally, add a still image of the umbrella woman to convince the viewer of her presence in the video clip (a still image is provided: **stimulus→videos→umbrella woman→still.png**). (Inspiration for this format was found in a clip that can be found on YouTube; search for "visual illusion attention experiment" to find it.) Each of the two test sequences (called **white shirts** and **black shirts**, resp.) should look like this, with the appropriate text used in the leading**Instruction** element.



An alternative approach to the two separate Studio tests is to simply create just one test (call it **common stream**, for example), with instructions that are intended to be supplemented verbally by the experimenter, e.g.,

Watch this short 40 second clip. Pay close attention to the players in the _____ (experimenter will tell you which) shirts and count the number of times they pass the basketball.

The reason this alternative approach will be made clear during visualization and analysis below.

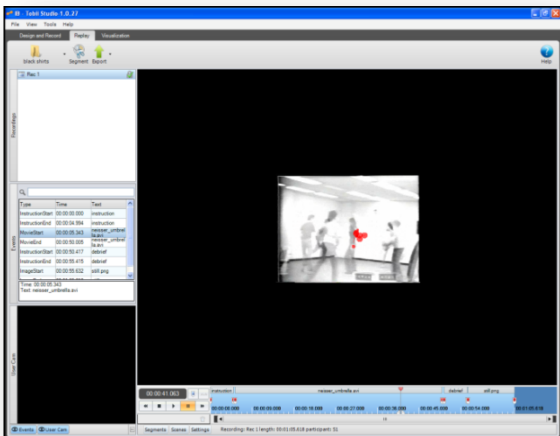
Running the Study

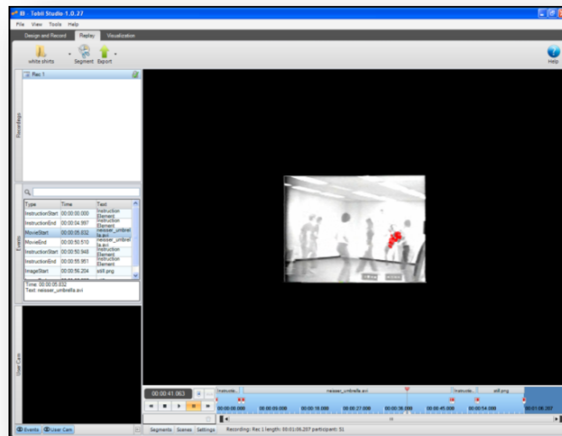
Run the separate Studio test study twice. First, run two participants, one viewing the **black shirts** sequence, the other viewing the **white shirts** sequence. Then run the study again with two more participants, both viewing the **common sequence** Studio test but with differing instructions (ask one to watch the white-shirted players, the other to watch the black-shirted players). When recording using the separate sequences, the recording names are not as important, you can use the default **Rec 1** in both cases. However, when making the recordings using the common sequence, label each recording properly, again using **white shirts** and **black shirts** as labels.

REPLAY

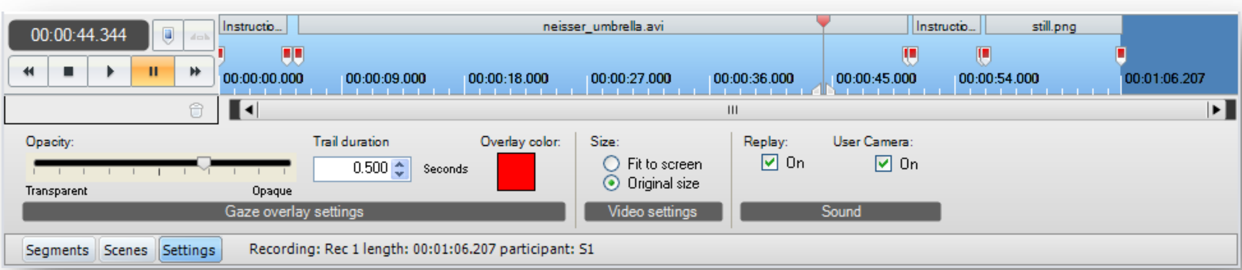
Select the **black shirts** test from the test menu. Using Studio's **Replay**playback feature (click the **Replay** tab), examine the viewer's eye movements at some interesting instant where the umbrella woman is visible. The idea is to discern whether when attending to the black shirts viewers fixated the umbrella woman and whether attending to the white shirts did the viewers not fixate the umbrella woman.

Select the **MovieStart** event in the **Events** list. Next, position the timeline cursor at some interesting timestamp where the umbrella woman appears, say **00:00:41:063** for example. Now switch to the **white shirts** test in the test menu and advance (or *scrub* in movie editing parlance) forward to the same timestamp. There are two things worth noting here. First, hopefully the viewers' eye movements land where you expect them to, e.g., when attending to the black shirted players, the viewer fixated the umbrella woman at the same instant when the viewer attending to the white shirted players did not.

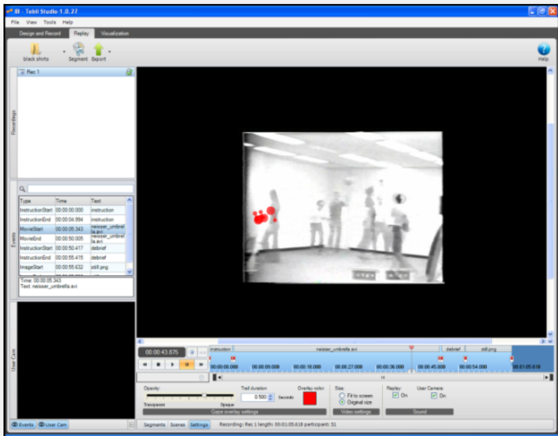


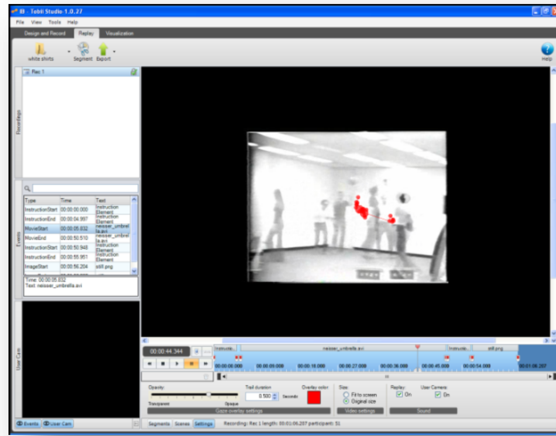


Second, notice that the video frames, even at the same timestamp, are different. Scrub forward in both Studio tests to find the frame where one of the black-shirted players is in mid-leap and the umbrella woman is about to walk off-screen. To increase the size of the image frame, expand the Settings tab below the timeline and select the Original size radio button.



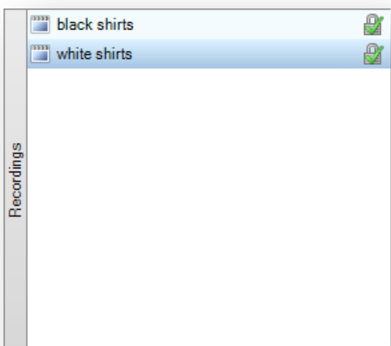
Once you locate the same frame in both test sequences, you may find that the timestamps differ, e.g., for the black shirt sequence the time stamp may be 00:00:43:875 while for the white shirt sequence it may be 00:00:44:344. Why the mismatch?



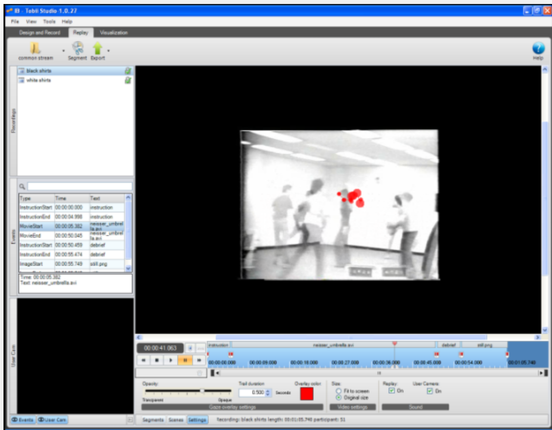


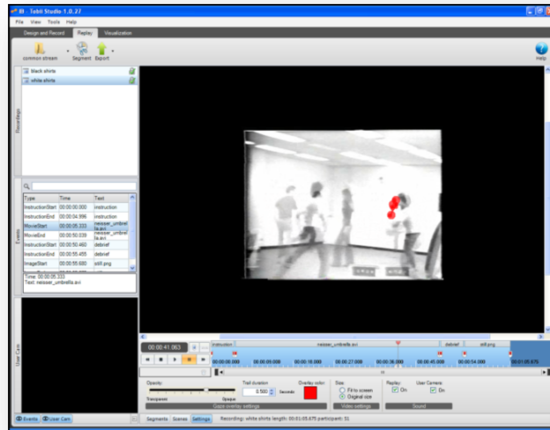
The mismatch may be due to the way the computer's internal clock works, or may be due to some slight error in setting up both sequences. The point is that even though the experimenter meant for the sequences to be identical, precise timing may not be possible if two separate sequences are used. To ensure proper timing, use the same test sequence and provide the missing instruction to the viewer verbally.

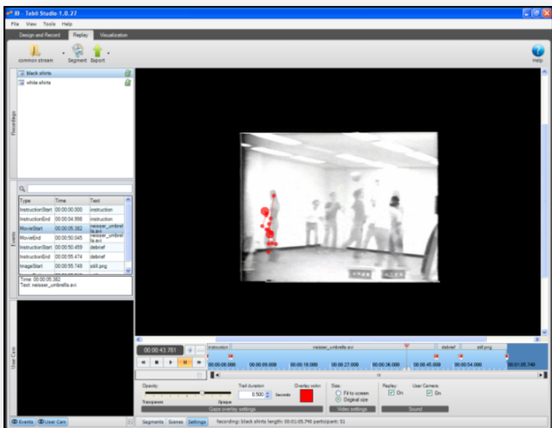
Select the **common sequence** test from the test menu and click the **Replay** tab. Now notice that instead of just one recording per test sequence, we now have both recordings in the **Recordings** list.

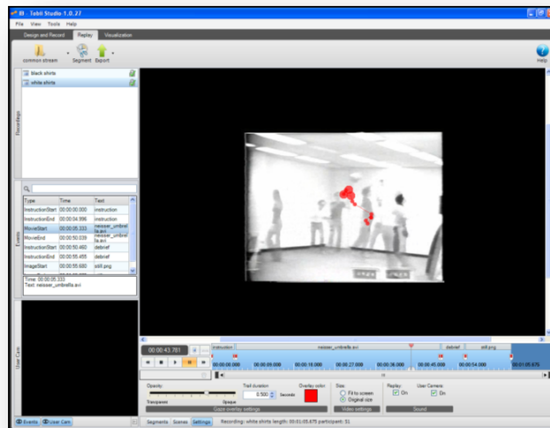


Select one recording or the other, then again select the **MovieStart** item in the **Events** list. Now scrub forward to the same frames as before. Try timestamps **00:00:41:063** and **00:00:43:781**. Do this for each of the recordings.







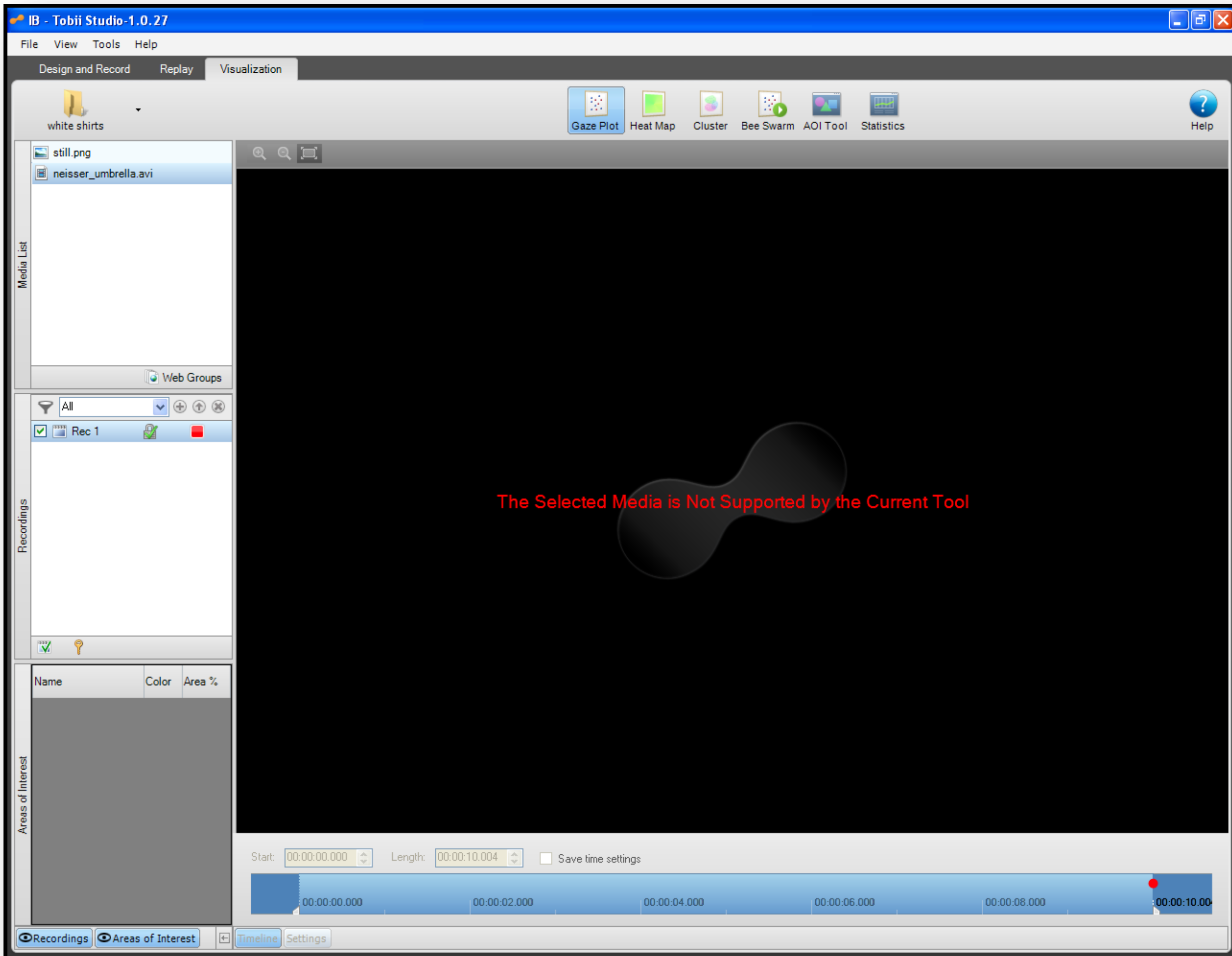


We now have better agreement in terms of timestamps. Note that the frames still differ slightly, but they are in closer agreement than before. And luckily, we see that when attending to the black-shirted players (at left), viewers fixated the umbrella woman. Meanwhile, when attending to the white-shirted players (at right), gaze falls on the players and not on the unexpected event (remember that these examples are "cooked" and the data is fake, but you get the idea).

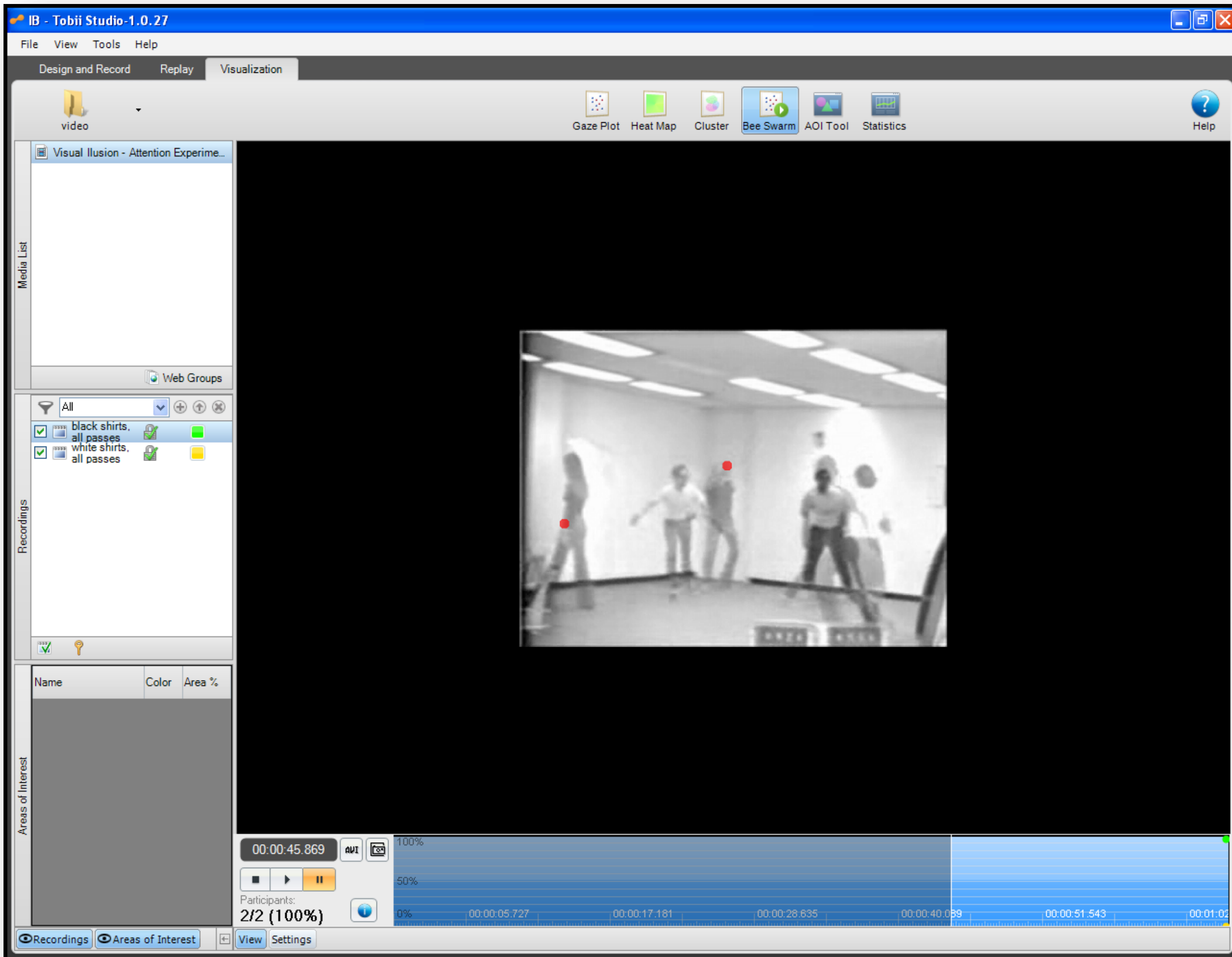
The playback provides an important depiction regarding fixation points, but is it enough to show statistical effect? For that we would need to count the number of fixations at identical (or nearly so) time instants and average them over the course of the video stream. Then we would need to correlate this data with viewers' verbal reports confirming (or denying) perception of the unexpected event.

Visualization Analysis and Results

Unfortunately eye movement analysis options over video are presently limited. For the separate sequence approach, they are simply unavailable.



With the exception of the **Bee Swarm**, none of the analysis tools supports video media. The **Bee Swarm**, however, while providing an interesting visualization, does not offer any further analytical tools.



The **AOI Tool** may in the future support dynamic media, but presently it does not. How might such a tool function in the future? And how can we analyze the eye movement data with current tools?

What we really want to do here is count the number of fixations within an AOI defined over our unexpected event, the umbrella woman. Ideally, this AOI should of course move (translate) across the video frame matching the woman's movement. Because the unexpected event's shape may change due to its projection onto the image (the camera is stationary in this case but it may not be), it would be even better if we could let the AOI change shape matching the outline of the unexpected event (the woman and the umbrella both). This is a fairly common problem in film production known as *match moving*. Various tricks of the trade are available for defining and automatically establishing the track of dynamic events. Changing the shape of the AOI from frame to frame entails *keyframing* the AOI shape. Some software packages call this a *rotoshape*, from the term *rotoscoping* (based on the *rotoscope*, a mechanical device patented by Max Fleischer in 1917 for projecting single frames of live-action footage onto an animator's drawing board; think Disney's *Snow White* and *Cinderella*).

The present version of Tobii's Studio does not provide a rotoscoping AOI tool unfortunately, but it may be possible to define such a shape with Apple's Shake, for example. Shake is (was) Apple's film compositing software. Among its many tools for video processing (compositing being the application's main tool), Shake provides a means for defining a rotoshape.

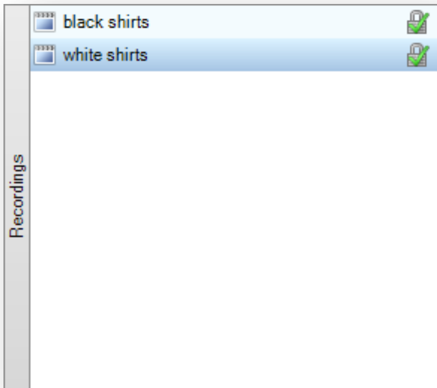
Just like Tobii's Studio, Shake's rotoshape is a basic polygon. Its position can be entered manually by scrubbing through the video, or it can be automatically created by a match moving operator that is set to match a block of pixels from frame to frame (e.g., the umbrella woman's cuff above her left wrist since it is a fairly well-defined region in terms of luminance difference). The match mover does not always perform flawlessly (far from it in fact on this particular video clip), and so in this case the rotoshape may be keyframed manually. Establishing its position on certain key frames is sufficient for Shake to interpolate its position linearly in between the key frames (a process known as *inbetweening*, which often requires tedious human intervention).

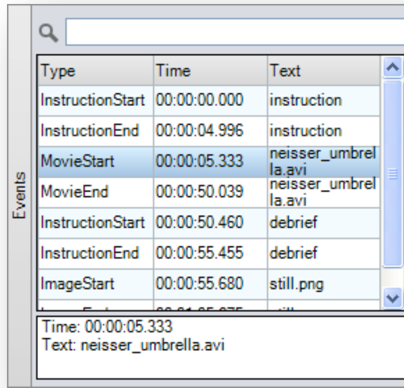
For our purposes, the most convenient feature of Shake is its capability of exporting the user-defined rotoshape as a plain text file. The first few lines of the exported rotoshape are shown in the text box. The important data elements to note are the `num_vertices` that define the number of vertices of the rotoshape, `num_key_times` that define how many keyframe values there are, and the `center_x` and `center_y` position of the rotoshape. This latter data value will change from keyframe to keyframe.

```
shake_shape_data 4.0
motion_blur 0.000000
shutter_timing 0.500000
shutter_offset 0.000000
num_shapes 1
shape_name Shape1
parent_name
closed 1
visible 1
locked 0
tangents 1
edge_shape 1
num_vertices 5
num_key_times 170
key_time 952.000000
center_x 531.190002
center_y 104.557999
color_r 1.000000
color_g 1.000000
color_b 1.000000
color_a 1.000000
vertex_data 517.852905 219.235275 517.852905 219.235275 517.852905 219.235275 517.852905
219.235275 517.852905 219.235275 517.852905 219.235275 484.205902 16.264690 484.205902
16.264690 484.205902 16.264690 484.205902 16.264690 484.205902 16.264690 484.205902
16.264690 561.882385 11.235271 561.882385 11.235271 561.882385 11.235271 561.882385
11.235271 561.882385 11.235271 561.882385 11.235271 579.205933 172.176437 579.205933
172.176437 579.205933 172.176437 579.205933 172.176437 579.205933 172.176437 579.205933
172.176437 554.617676 210.735245 554.617676 210.735245 554.617676 210.735245 554.617676
210.735245 554.617676 210.735245 554.617676 210.735245
key_time 953.000000
center_x 531.190002
center_y 104.557999
color_r 1.000000
color_g 1.000000
color_b 1.000000
color_a 1.000000
vertex_data 517.852905 219.235275 517.852905 219.235275 517.852905 219.235275 517.852905
219.235275 517.852905 219.235275 517.852905 219.235275 484.205902 16.264690 484.205902
16.264690 484.205902 16.264690 484.205902 16.264690 484.205902 16.264690 484.205902
16.264690 561.882385 11.235271 561.882385 11.235271 561.882385 11.235271 561.882385
11.235271 561.882385 11.235271 561.882385 11.235271 579.205933 172.176437 579.205933
172.176437 579.205933 172.176437 579.205933 172.176437 579.205933 172.176437 579.205933
172.176437 554.617676 210.735245 554.617676 210.735245 554.617676 210.735245 554.617676
210.735245 554.617676 210.735245 554.617676 210.735245
```

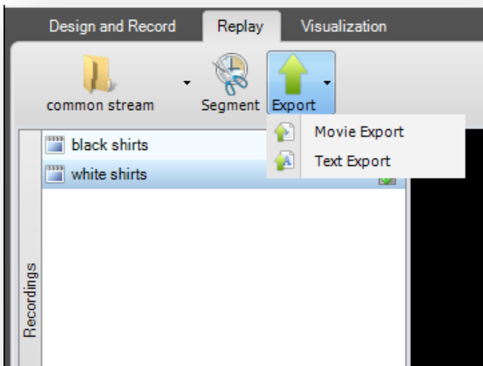
The large number of vertices expressed by the `vertex_data` element pertains to the tangents defined at each rotoshape vertex. Some of these values may probably be ignored.

Studio is also capable of exporting recorded scanpaths as plain text files. Select the **common sequence** test from the test menu and click the **Replay** tab. Choose either of the recordings from the **Recordings** list.

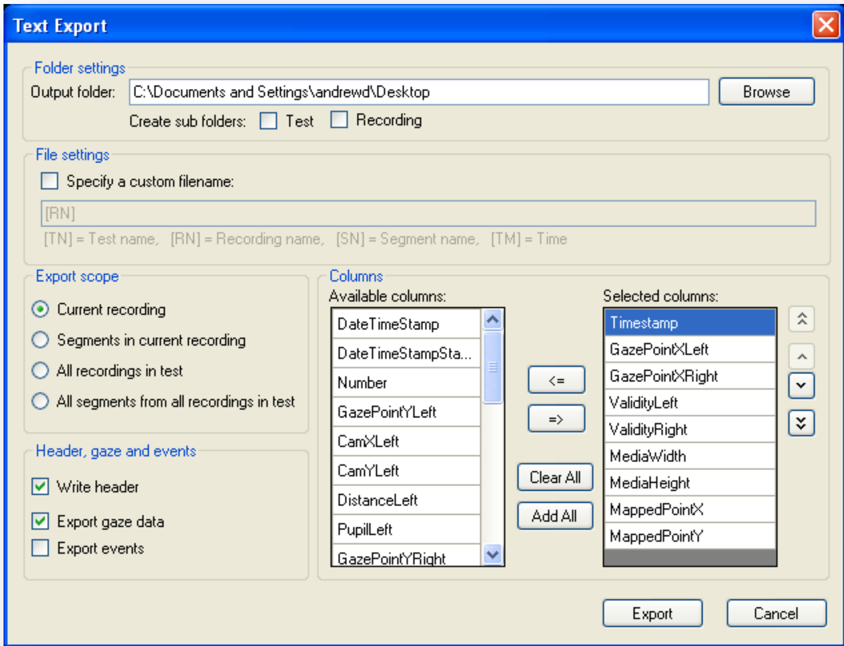




Select the **MovieStart** item in the **Events** list. From the **Replay** tab's **Export** menu, choose **Text Export**.



In the pop-up **Text Export**dialog, first choose the **Output folder**. From the **Export scope**GUI group, select **Current recording** to export the scanpath from only the **black shirts** or **white shirts** recording. Select the data you want output, and how it is arranged in columns in the output file. In this example, choose the **Timestamp**, **GazePointXLeft**, **GazePointXRight**, **ValidityLeft**, **ValidityRight**, **MediaWidth**, **MediaHeight**, **MappedPointX**, and **MappedPointY** then press the **Export** button.



The first few lines of the output data should appear as in the textbox.

```
System Properties:
    Operating System: Microsoft Windows NT 5.1.2600 Service Pack 2
    System User Name: andrewd
    Machine Name: TOBII

Data properties:

    Recording name: white shirts

    Recording date: 10/23/2007
    Recording time: 10:34:17 AM

    Export date: 10/23/2007
    Export time: 3:02:39 PM

Timestamp      GazePointXLeft GazePointXRight       ValidityLeft   ValidityRight
        MediaWidth      MediaHeight    MappedPointX    MappedPointY
0       564.6595        597.7616        0       0       0       0
20      572.1125        601.4225        0       0       1280    1024    589     494
40      574.1533        605.4293        0       0       1280    1024    589     494
60      573.8362        600.2247        0       0       1280    1024    589     494
80      562.0909        606.5455        0       0       1280    1024    589     494
100     568.2215        606.6851        0       0       1280    1024    589     494
120     569.7728        620.5757        0       0       1280    1024    589     494
140     571.2032        617.2695        0       0       1280    1024    589     494
160     572.0243        615.8949        0       0       1280    1024    589     494
```

The next task is of course nontrivial, as it requires writing a program to parse both data files, and then to integrate the data. The main concerns are mapping of spatial coordinates between the two data sources as well as synchronizing the temporal parameters. Spatial coordinates should either be normalized to the [0,1] range, or should be mapped to some common reference frame, e.g., Tobii's screen resolution, specified by the `MediaWidth` and `MediaHeight` values above (1280x1024). Shake's `center_x` and `center_y` dynamic position of the rotoshape will also need to be mapped to this frame, with a clever guess regarding the dimensions of the reference frame they were defined in to begin with.

Temporal synchronization of frames has to do with lining up the rotoshape over the umbrella woman over the proper frame numbers. Shake's rotoshape was defined over keyframes within the entire video frame sequence [1,1340]. This frame number must be properly matched to Tobii's `Timestamp`, which is relative to the time course of collected eye movements, and not the video sequence. Timing is important here. Eye movements are collected every 20 ms (since the eye tracker operates at 50 Hz), where as each frame is displayed at a rate of 30 frames per second, or fps, hence one frame per 30 ms. There is no one-to-one mapping between eye movements and video frames, hence it is possible that certain video frames will contain no eye movement data points, while others will contain several.

Proper analysis requires some care and a good bit of effort. A similar approach to the one outlined above was taken by researchers from the University of Bristol when evaluating the efficacy of sensor fusion, e.g., combination of infra-red video with visible-spectrum RGB data (Dixon, et al., 2007). Eye movement (fixation) data was compared with pre-drawn 'target maps' (rotoshapes essentially) drawn around a target to be tracked (a soldier). A similar keyframing operation to Shake's was performed manually with inbetweening performed automatically via interpolation (Shake provides a similar mechanism). In the experiment, the rotoshape serves as a ground truth that is used to evaluate human tracking performance.

FIGURE 3: TARGET-TRACKING TOOL (LEFT) AND GAZE LOCATION COMPARISON WITH ROTOSHAPE (AT CENTER IS A HIT, AT RIGHT IS A MISS). COURTESY OF STAVRI NIKOLOV; IMAGES ARE PUBLICLY AVAILABLE AT HTTP://WWW.SCANPATHS.ORG.

Figure 3 shows an example of the desired outcomes of the analysis: given synchronized eye movement and video data, it is possible to report accuracy scores for target tracking by human observers. In their work, Dixon et al. compared different computational approaches to sensor fusion. In the present example the above approach would be used to evaluate the effect of task on tracking accuracy, but the procedure would be very similar.

UNDERSTANDING HOW USERS EXPERIENCE WEB SITES

The previous sections showed examples of experiments that use static images and video to understand some basic cognitive processes. Eye tracking is also an excellent tool for exploring issues in human computer interaction (HCI) and usability. One important application of eye tracking for both basic and applied research is probing how users experience Web sites. While some usability engineers will simply take screen captures of Web page designs of interest and use some of the techniques described above for static images, this is sometimes insufficient. Often, it is important for users to interact with a real Web browser and live HTML. For example, a researcher interested in information architecture may want to understand how users interact with an entire site how they explore multiple pages for different links. Other researchers may be interested in making sure that users see important information such as alerts or warnings on live Web pages.

Studying live Web sites is fraught with its own set of difficulties. One problem is that Web pages are often bigger than the hosting browser, so users need to scroll the page to see all the content. However, the eye-tracking system is recording the eyes' positions relative to the screen, NOT the page. This leads to a registration problem similar to that described above for video analysis. When a user scrolls a page, she may be looking at the same place on the screen, but very different Web content. How do we get the screen-based coordinates of the eyes registered with the variable browser-based coordinates of the Web page? Typically, this is done with some kind of a plug-in to the browser that tracks the position of Web content relative to the screen and this is then mapped to the eye-tracker recording. Happily, there are several options currently available. One of the first of these plug-ins was developed by researchers at PARC (Reeder, Pirolli, & Card, 2001). For this example, we will be using the plug-in provided by Tobii Studio.

Before we dive into the example study, there are a few limitations you should be aware of. Most importantly, the plug-in provided with Tobii Studio works very well with standard HTML content. Unfortunately, dynamic content such as javascript, Flash or Silverlight doesn't work very well. Often, pages with this kind of content will show up as blank pages in Tobii's analysis software. Web-based applications based on these technologies are becoming increasingly common, and you should understand that currently the only way to do eye-tracking analysis of such pages is to use screen-capture techniques such as those described in the section for eye tracking for applications.

BACKGROUND

In this example, we will design an experiment to explore the effect of the placement of advertising in Web search. It is commonly assumed that the best location for a business to place their advertising on a Search Engine Results Page (SERP) is at the top of the page, above the first "algorithmic" result (i.e., the results that are not explicitly paid for by advertisers). However, placing an ad in a top position is often more expensive than placing the same ad to the right of the results. A business may want to know: How much more visibility does an ad at the top of the SERP really have versus an ad placed on the right of the SERP? If we opt for the less expensive ads, will searchers ever see them? What difference does it make if the only ads shown are on the right?

HYPOTHESIS

For this simple example experiment, the hypothesis is very straight-forward: advertising at the top of the SERP will be viewed more than ads to the right of the SERP. In addition, we will hypothesize that this effect is lessened if the only ads on the page are on the right—i.e., for an identical search, advertising at the side of the page will be viewed more if there are no ads at the top of the page.

EXPERIMENTAL DESIGN

Controlled experiments in Web search are notoriously difficult to perform because of the extreme variability in different queries. In addition, between-subject variability can be very large because of differences in searchers' knowledge, experience and search sophistication and the fact that the nature of the tasks can be subjective. Therefore, we want to do as much within-subjects analysis as possible by showing every participant every combination of our independent variables possible. To keep this example tractable, we will assume the minimum number of tasks/SERPs necessary to adequately counterbalance the experiment (3). In reality, we would want to use some multiple of this number of tasks.

For this experiment, the independent variable (the position of advertising) has three levels: Top, Side, and Both. Since a given search task can be performed only once by a given participant, we will need to counterbalance task with ad position between subjects; we want to make sure that the variability associated with the task is shared by all levels of our variable. So, for our 3 search tasks we will need a multiple of 3 participants to make sure that each ad position is seen for each search task. E.g.,

Ppt 1:	1-Top	2-Side	3-Both
Ppt 2:	1-Side	2-Both	3-Top
Ppt 3:	1-Both	2-Top	3-Side

(In reality we would also need to counterbalance the order of tasks as well, but for this example we will ignore this.)

STIMULUS

Eye-tracking research is particularly sensitive to small changes in the arrangement of visual stimuli. This can be a serious problem for Web search because two searches with identical queries issued 5 minutes apart can generate very different results. Such differences could easily sabotage all the variability we're attempting to control with the counterbalancing described above. To control for this, we will cache the initial SERPs that all searchers will see for each task. To maintain the realistic sense of Web searching, the SERPs will be live—that is, searchers will be able to view links from the results page and issue new queries. However, we will focus our analyses on the first pages they interact with, as these will be controlled across participants.

To make this work, will need to do some basic HTML editing to create our stimuli and experimental control. In contrast to the previous examples, all the instructions, stimulus order, etc., will be done outside of Tobii Studio and within our Web pages. We will need to generate 3 control Web pages, one for each of the 3 groups of participants. These pages will include task instructions and links to the cached SERPs we will be testing. In addition, we will need to save and edit 3 different SERPs (for each task) to generate pages with advertising in the different positions. One of the control pages is below:

APPARATUS

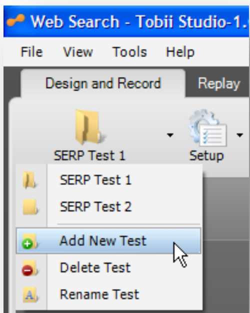
Once again we will use Tobii's Studio software and a Tobii ET-1750 video-based corneal reflection eye tracker (or the faster T120 if available). The Tobii ET-1750 operates at 50 Hz at an accuracy of about 0.5 degrees visual angle (bias error). (The

T120 operates at 120 Hz.) At the time of testing, screen resolution was set to 1280x1024. The eye tracking server ran on a Sun model w2100z dual 2.0 GHz AMD Opteron 246 machine equipped with 2 G RAM. The server was running Windows XP.

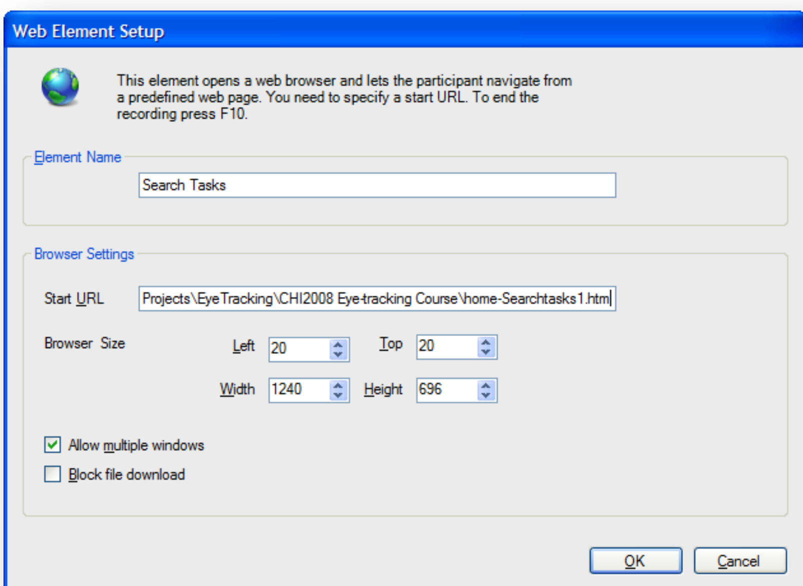
Studio

Design and Record: Web

As in previous examples, we begin by creating a new project. Once you provide a project name (e.g., **Web Search**), Studio will automatically create an instance of an empty test. Because our experiment calls for 3 different sets of pages to be balanced, we will create 3 different tests. First we create the three tests and then add the content to each. We can create a new test through either the **File Menu** or by clicking on the down-arrow next to the current Test. In the screen shot below, we've already created 2 tests (SERP Test 1 and SERP Test 2) and are about to create the third:



Now we must provide the starting Web site for each test. Select the first test (e.g., SERP Test 1) and drag the **Web** element icon onto the central media viewer. Studio will pop up the **Web Element Setup** dialog where you can specify the name of the element and the starting URL. Because all our experimental control happens from the HTML we authored, we need only supply the corresponding page for the current test. Below, we're pointing to "home-Searchtasks1.html." Repeat this for each of the other two tasks (e.g., SERP Test 2→home-Searchtasks2.html and SERP Test 3→home-Searchtasks3.html).

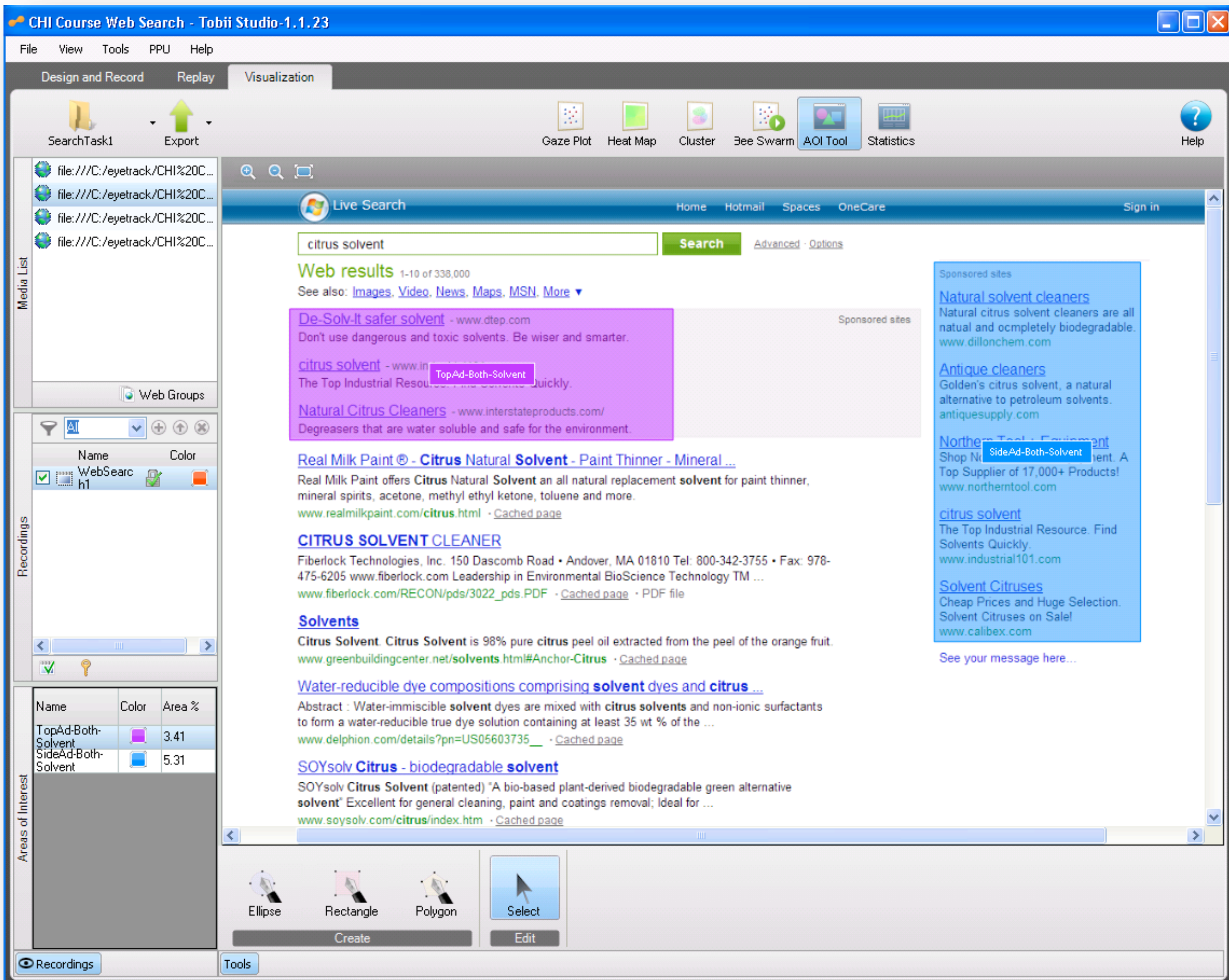


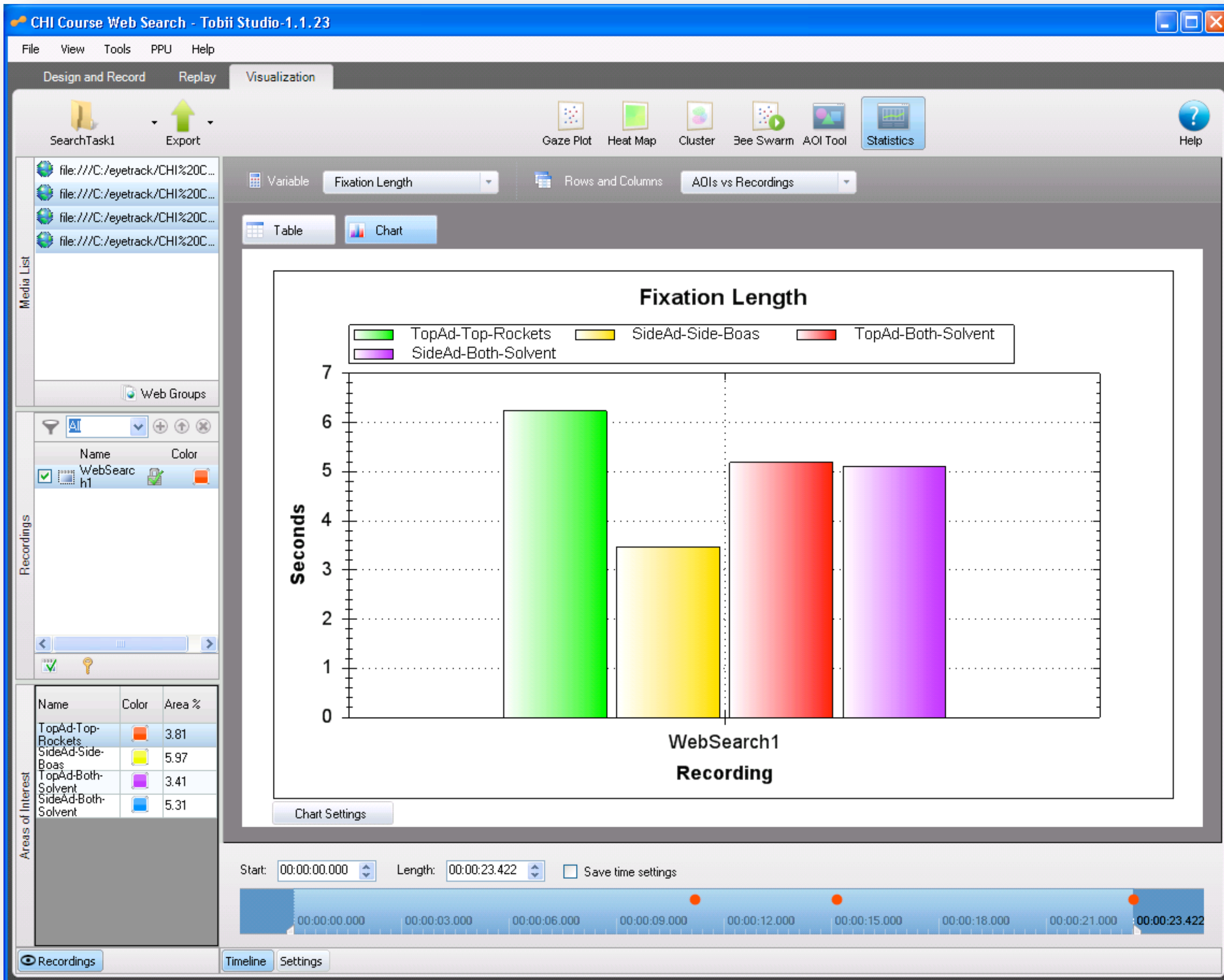
Running the Study

Run a participant in each test you created for a balanced design. Since we're using eye tracking in support of a normal Web task that users perform frequently, there's really nothing special that you need to do here. Ideally, you would run several participants in each test.

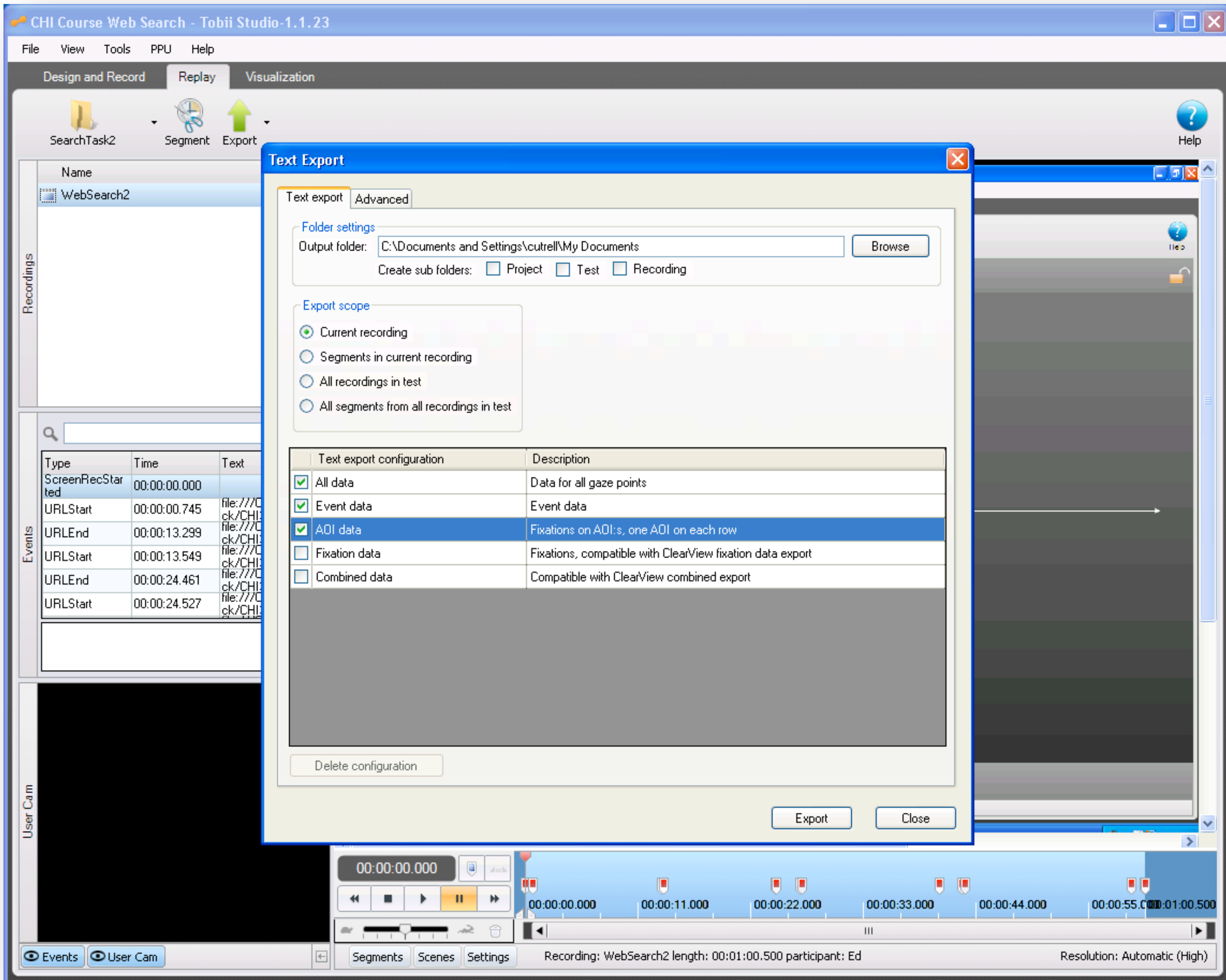
ANALYSIS & RESULTS

We can do a first pass qualitative analysis by viewing the gazeplot or heatmap visualizations (as described earlier). If the results are really obvious, we may need to do nothing more. However, to do any quantitative analysis, we need to compare gaze metrics for the areas of interest (e.g., the advertisement positions) within and between participants. As in the PCB experiment, we can use the **AOI Tool** and **Statistics** buttons to generate some simple plots. As a first pass, the AOIs we'll identify are fairly simple: the top ad region, the side ad region, the main body of search results. We can now use the **Statistics** buttons to do some simple comparisons of gaze behavior in our regions of interest for each query.





If our experiment were a bit more complicated (for example if we were dynamically changing things for each page presented or we just wanted a great deal more detail and resolution of AOIs for each page viewed), defining AOIs by hand at the end of each test could get very tedious. To deal with this, we can define AOIs either on the fly as users encounter pages, or do it post-hoc. In either case, we will want to export all our data in a text file and merge the gaze data with our AOIs (e.g., with scripts). We would then do all our analyses off-line with scripts and other statistics packages. For many questions, particularly usability settings or when using eye tracking as a supplemental information source, Studio's visualizations and simple comparisons may be sufficient. However, if we want to fully exploit our experimental design (collapsing across queries for each condition), we need to compile the data outside of Studio for further analysis. First, we must export the data using the **Export** button as described above for the Video experiment. As with that experiment, the next step requires us to write a script to parse the files and output the data into a useful form for further analysis by a statistics programs (such as SPSS, SAS, JMP, R, etc.). In the case of our example, we'll need to abstract the 3 conditions and arrange the data for a within-subjects analysis for the relevant dependent variable(s) (e.g., fixation duration, number of fixations, or compound variables). Proper analysis depends on good experimental design and a good deal of care in dealing with the output data. However, with a little patience and luck, eye-tracking can reveal excellent insights into Web page design. A similar approach to this was taken by (Cutrell & Guan, 2007), looking at the effect of snippet length and task type on the experience of Web search experience.


CHI Course Web Search - Tobii Studio-1.1.23
File View Tools PPU Help
Design and Record
Replay
Visualization
SearchTask2
Segment
Export
Help
Text Export
Text export
Advanced
Folder settings
Output folder:
C:\Documents and Settings\cutrell\My Documents
Browse
Create sub folders:
Project
Test
Recording
Export scope
Current recording
Segments in current recording
All recordings in test
All segments from all recordings in test
Text export configuration
Description
All data
Data for all gaze points
Event data
Event data
AOI data
Fixations on AOI:s, one AOI on each row
Fixation data
Fixations, compatible with ClearView fixation data export
Combined data
Compatible with ClearView combined export
Delete configuration
Export
Close
Name
WebSearch2
Recordings
Events
User Cam
Type
Time
Text
ScreenRecStarted
00:00:00.000
URLStart
00:00:00.745
URLEnd
00:00:13.299
URLStart
00:00:13.549
URLEnd
00:00:24.461
URLStart
00:00:24.527
00:00:00.000
00:00:11.000
00:00:22.000
00:00:33.000
00:00:44.000
Segments
Scenes
Settings
Recording: WebSearch2 length: 00:01:00.500 participant: Ed
Resolution: Automatic (High)

IMPROVING APPLICATION USABILITY

In a software development environment, the momentum for change comes from 2 primary sources: customer satisfaction, and the advent of new technology. In a traditional software development cycle the technology and user interface design are developed in parallel, inextricably entwined. At each step of the process assumptions are made about how the software 'speaks' to the user, the sequence by which the user and the software are communicating their state, purpose, position in a task sequence. The iterative process of interaction design assumes the principles by which the application was developed are communicated via the user interface. Eye tracking can help verify these assumptions about how the UI is consumed by the user, both in real-time (as the user is completing a task) and post-hoc (aggregate user analysis).

The use of eye tracking in a corporate environment will yield great benefits to the researcher, technology teams, and ultimately the customer. The addition of eye tracking data to the usability metrics will round out the data in a highly salient and actionable manner. Most eye trackers are very easy to integrate and use in the corporate usability lab. However, depending on the stimulus (in this specific example, a Windows application), the analysis of the data post-hoc can be time consuming. The crux of the issue is that unlike web-pages, eye tracking software does not recognize Windows-level application events. Simply put, the eye tracker can record the screen and monitor the eye movements of a study participant in real time, but it has no awareness of what the user is *doing*. The tracker software does not know what application(s) are in use, and therefore it does not recognize the actions the user is performing (e.g. open a dialog, click a menu item, etc). Essentially the output for analysis is a movie that must be manually tagged, segmented, and aggregated to produce the typical eye tracking data visualizations (e.g. heat maps, gaze plot, bee swarm, etc).

It must be noted that the limitations of eye tracking software for application usability typically have little affect on the researcher (and the value of eye tracking in this environment). This is due to the fact that eye tracking data is usually observed, recorded and interpreted in situ of the greater usability study. While observing the user clicking on a menu the research is also observing the eye movement pattern associated to said menu'ing behavior, as well as the participant's skill level, apparent emotional state, and the state of their path to successful completion of the task. Therefore, the eye tracking data is just another piece of overt behavior that is being observed, noted, and interpreted in the context of the observed behavioral patterns.Only in the case of specific research questions that require eye tracking visualization output is there a requirement to spend the extra time analyzing the data.

BACKGROUND

In this study, a series of common Microsoft Word tasks will be performed with Microsoft Office Word 2007 to assess the effectiveness of the Fluent© User Interface system. Specific components of the Fluent UI will be isolated to determine their ability to support core user document creation and editing tasks in Word. The usability of features, measured by their discoverability, efficiency of usage, and overall support of task completion will be assessed. This example follows the protocol of a typical software development usability study.

RESEARCH QUESTIONS

Due to the schedule and cost associated with software development, succinct, scoped and specific research questions dictate the usability study design. In the case of Office 2007 and the introduction of the Fluent UI system in Word, the following research questions were asked:

1) Are users able to **efficiently** find commands in the new Office UI?

   An operational definition of 'efficiently' is necessary to answer this question. For this study 'efficiency' is defined as the aggregation of:

   a) task success
   b) task time
   c) scan path patterns that match what was expected of the interaction designer.

2) Were the problems from the previous UI **solved**?

The Fluent UI system was introduced into the Microsoft Office 2007 release of Word to address several usability issues with the previous version of Office, namely:

a) Improve user's effectiveness in accessing functionality and completing tasks (using the right controls at the right time)
b) Improve access and interaction with formatting tools
c) Promote experimentation and foraging for new/different functionality.

HYPOTHESIS

The experimental hypothesis to be evaluated states that the visual search pattern for feature discoverability is consistent with the primacy of design elements on the Word interface for each given user scenario and task. It is expected that the Fluent UI will be 'consumed' in a manner consistent with that which was expected of the interaction designer, both in fixation sequence over UI elements and in behavioral interaction (e.g. mouse clicks). These fixation and behavioral data points will vary as a function of task, but should remain consistent within and between subjects.

Given the specific research questions delineated above, the following are our specific hypotheses:

a) The user will exhibit behavior that suggests he is able to efficiently access functionality based on task type, independent of familiarity with said task
b) The user will exhibit behavior that suggests he is able to efficiently format text and objects during document creation
c) The user will exhibit behavior that suggests he is in control of Office

EXPERIMENTAL DESIGN

VARIABLES

To test our hypotheses, 2 primary variables will be isolated: Word 2007 and Task type.

1) Word 2007 - the Fluent UI, specifically:

- Tabs
- Contextual tabs
- Auto-preview
- Mini bar
- Galleries
- Quick-access toolbar

2) Task Type: document creation:
    a) Frequent & Familiar tasks
        i) Open a file
        ii) Bulleting
        iii) Bolding
        iv) Adjust margins
        v) Save a document
    b) Infrequent & Unfamiliar
        i) Custom bullets
        ii) Custom table border
    c) Core & Ubiquitous UI
        i) Accessing Help
        ii) Inserting objects/Common File Dialog
        iii) Zooming
        iv) Customizing the Quick Access Toolbar
    d) New Features
        i) Contextual tabs
            (1) Access Table Tools (format)

(2) Apply Table formats 2 (border)
(3) Add picture effect
ii) Galleries & Formatting
(1) Scroll a gallery (Font QuickStyle)
(2) Scroll a gallery (presentation design)
(3) Use a gallery (bullet style)
(4) Use a gallery (margins)
iii) Office button
(1) Change Excel app options
(2) Opening a file
(3) Saving
(4) Saving as a new version

CONTROL

User research studies in a software development environment tend to not reflect as much scientific rigor as traditional research studies in an academic or laboratory environment. However, attempts are always made by the user researcher to ensure data speaks to the research questions in a reliable manner. These control variables are even more important for eye tracking studies. Real-time eye tracking data is highly compelling for any observer, especially observers who are not trained to filter any overt behavioral expression thru the lens of data triangulation. Therefore it's even more important for the task list to be robust and the method sound.

Task list control: participants in this study will have varying levels of exposure to the Word 2007 UI. To help control for the variance, initial exposure to each element of the Word 2007 will be built into the warm-up tasks.

Eye movement control: traditional usability studies require participants to think out loud (verbal protocol) as they work. Due to the effect verbalizing has on user eye movements, verbal protocol will be eliminated from this study.

Study administrator training: In most corporate usability studies the researcher is intimately familiar with the user interface under investigation. Further, the researcher has worked with the interaction designer to create the optimal solution for the user and can predict the 'correct' scan paths necessary for efficient task completion.

To ensure the study administrator in this training is equally primed to see the 'correct paths', an eye tracking video of correct and efficient task completion has been created. Each study administrator should watch this video before their study sessions so observations of participants are *inherently* in comparison to this optimal/correct model.

ANALYSIS COMPARISONS

The primary research question for this study is if the participant 'consumes' the Word 2007 UI in the same manner the Interaction Designer predicted. The primary output of this exercise is the study administrator's observation log. This log should include as much detail as possible about the approach the participant took to complete each task. The log should identify the eye movement behavior as observations (the same as mouse clicks, etc) to ensure the participant's behavior is recorded. Again, the observations should be made about all observed behavior, and should inherently compare the participant's task completion behavior to the optimal/correct paths.

In the case of task completion breakdown, the study administrator must decide the most appropriate analysis method to employ. Will a video snippet, heat map, bee swarm or gaze plot going to accurately capture the issues observed? What is the most compelling output based on your audience and time constraints.

ANALYSIS EXAMPLES

HEAT MAP

During the development of Office 2007 file menu, there were concerns about the complexity of the menu. Of particular interest was the user's attendance to the icons, flyout buttons, and sub-menu titles. A number of participants completed tasks relating to the file menu. A subsequent heat map was created to visualize the concentration of eye movements across the menu. The heat map efficiently communicated the high concentration of fixations on the primary file menu titles (New, Open., Share).

However, data suggested little attention paid to the icons, nor to the flyout titles. This data helped the team focus discussions on the file menu, and its more salient interaction elements.

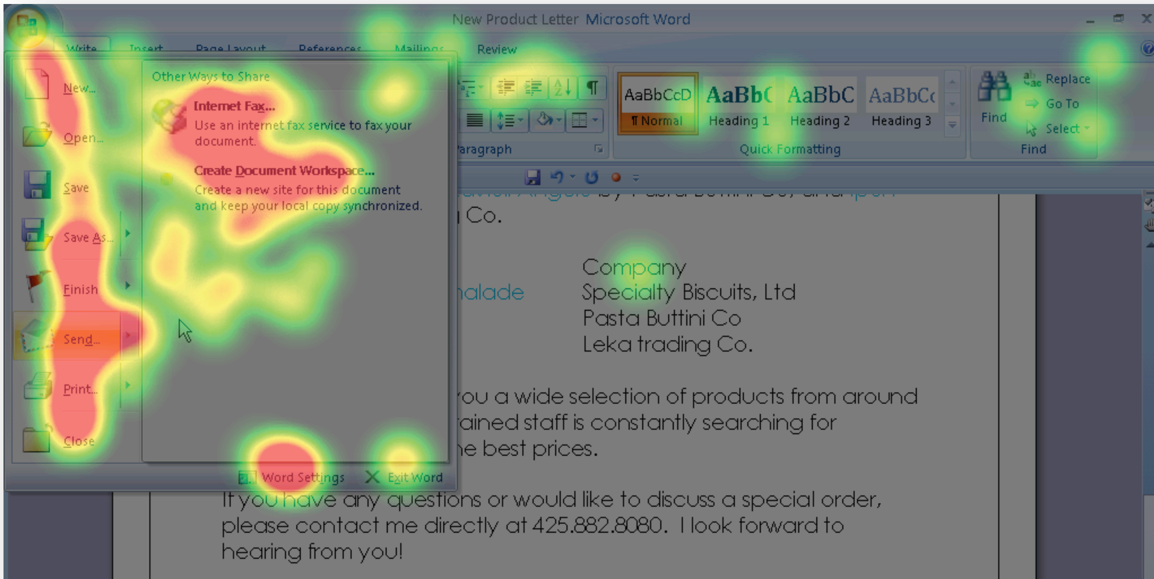


FIGURE 4: WORD 2007 HEAT MAP

Gaze Plot

As stated in the previous Heat Map example, the Office 2007 file menu was studied for various reasons. One concern was participants' comprehension of the menu fly-outs (their interaction rules, purpose, etc). A study was conducted to evaluate this part of the File menu. A particularly interesting finding was not directly related to an item with a flyout. Rather, the interesting data was in the case where a file menu item did *not* have a flyout. Data suggested that participants were unable to predict when an item had a flyout and when it did not. This was evidenced by the fact that when users moused over the "Workflows" menu item, their eye movements displayed the same scan behavior (up to the right and down) as when on a flyout item. Results from this study helped to justify further simplification of the flyout behavior in the file menu.

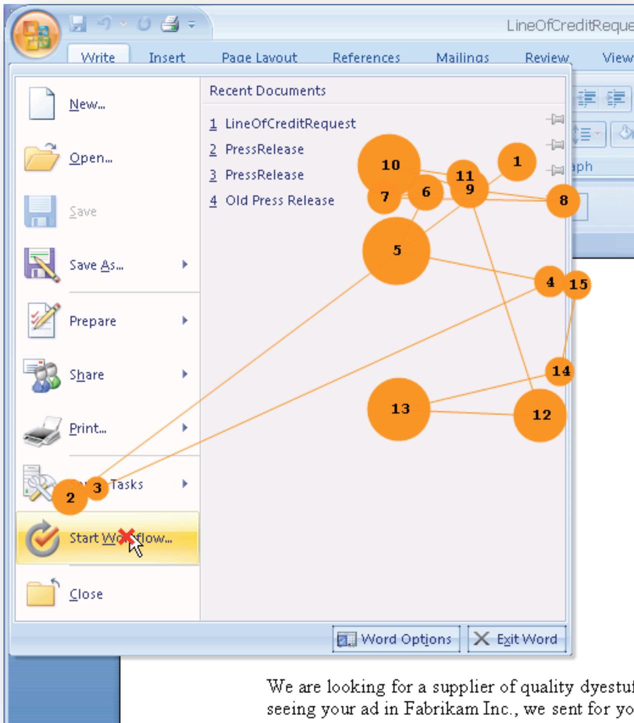


FIGURE 5: WORD 2007 FILE MENU GAZE PLOT

Live (real-time), Movies

Probably the most compelling of all eye tracking data visualizations is in real time, in the usability lab with your technology team watching a 'real person' interact with the application. The addition of eye tracking data draws your teammates into the study, and helps create an empathetic connection between themselves and the user. In the case where a participant is failing a task, the eye movements help make sense of the behavior to the observer (who tend to think all the task should be easy to complete ☺).

In the case where your team cannot observe in the lab, movies of the eye tracking sessions are a reasonable substitute to the experience of real-time observation. In the interest of brevity and efficiency, the eye tracking data should be specific to issues observed, and be indicative of issues with many/all users (depending on the severity of the problem). It's important to note that an eye tracking movie will always be of one user (never in aggregate). Therefore it's important to select compelling (but accurate) illustrations of issues observed in the lab.

Stimulus

Microsoft Office Word 2007 will be the stimulus for this exercise. There are several areas of the Word user interface that will be of interest for this study:

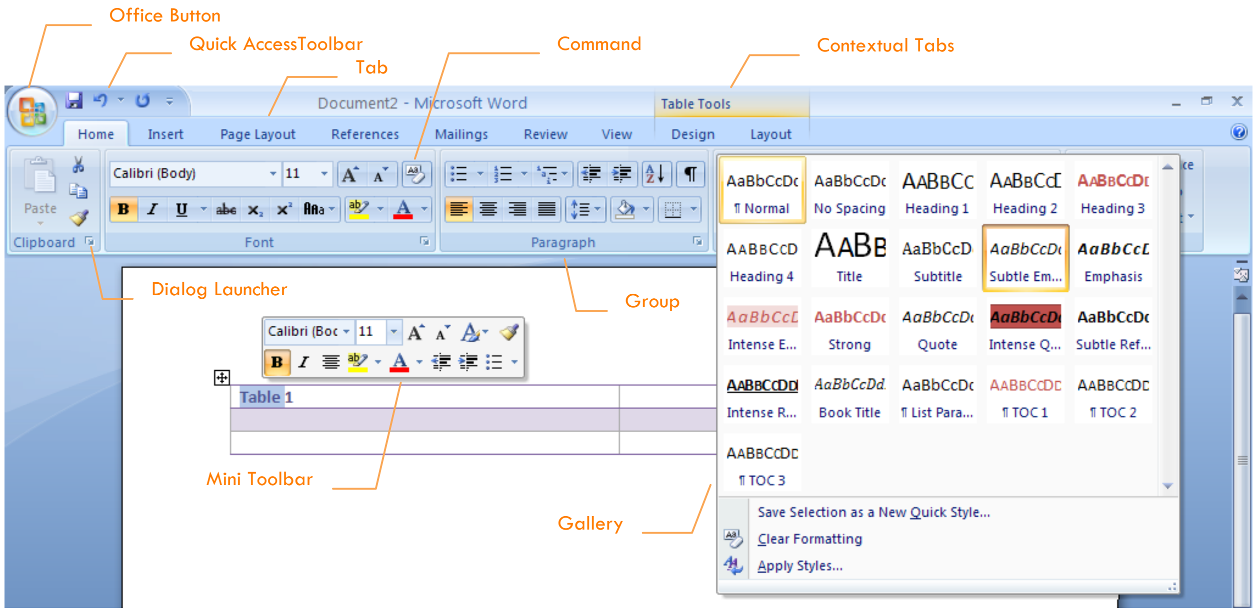


FIGURE 6: ANATOMY OF WORD 2007 FLUENT USER INTERFACE

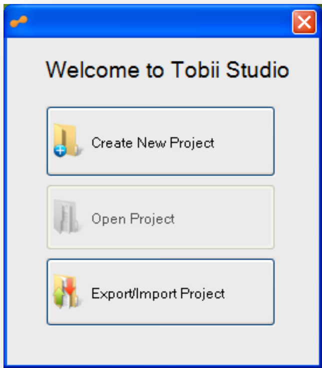
APPARATUS

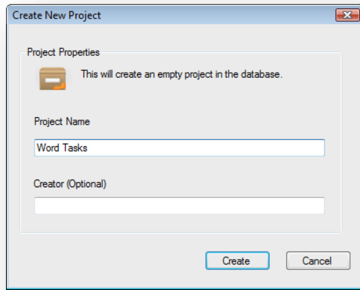
We will use Tobii's Studio (v1.0) software for real-time observation of task completion in Microsoft Office Word 2007 as well as eventual analysis. At the time of this writing, a Tobii ET-1750 video-based corneal reflection eye tracker was used to test the procedure. The Tobii ET-1750 operates at 50 Hz at an accuracy of about 0.5 degrees visual angle (bias error). At the time of testing, screen resolution was set to 1280x1024. The eye tracking server was running Windows XP.

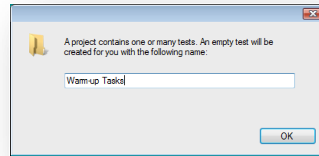
STUDIO

Start Tobii's Studio, either via the Start menu or by double-clicking the Tobii icon.

If no projects have yet been created, the **Open Project** option will be grayed out from Tobii's **Welcome to Tobii Studio** dialog.



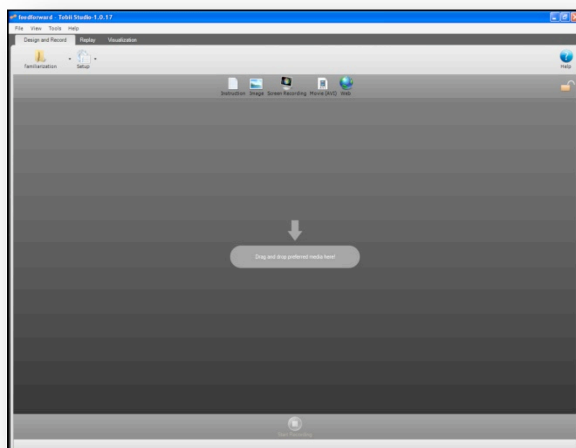




Create a new project, providing a new project name and creator identity. Studio's database is made up of projects where each project contains at least one test. Each test, in turn, is made up of an instance of a visual stimulus such as a still image, a video clip, a web page, or screen recording (desktop), or a sequence of the former types of stimuli. Once you provide a project name (e.g., **Word Tasks**), Studio will automatically create an instance of an empty test, providing you with a chance to name that test. You will be creating 2 tasks lists (warm-up and formal). Therefore, the first test you will create is the Warm-up Tasks test.

Design and Record: Build Task List Flow

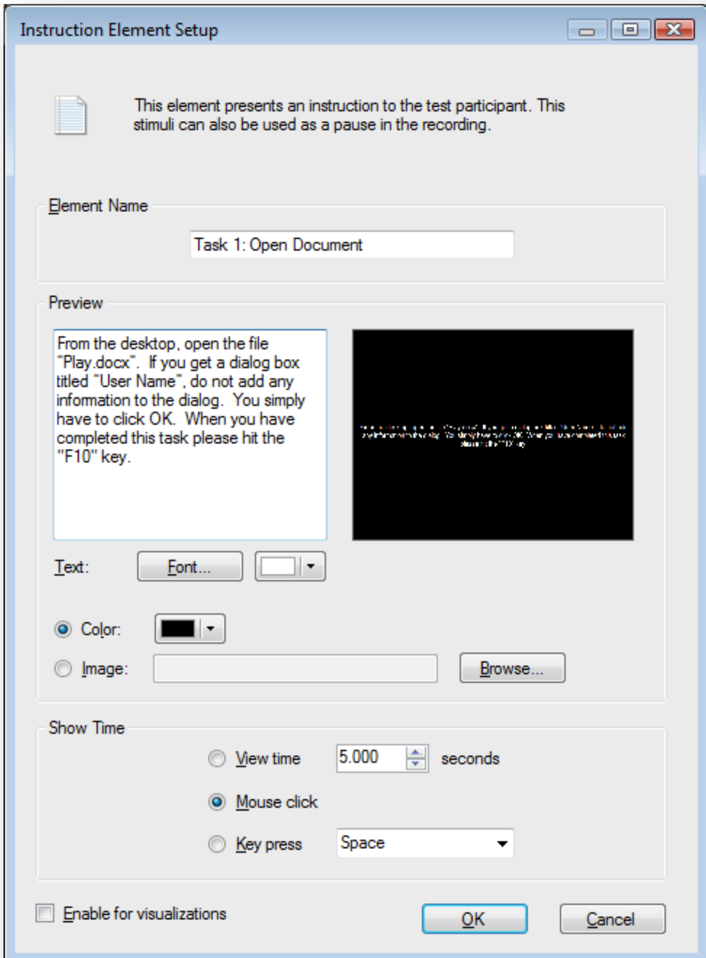
You will create this study by building your task list into the study flow. You will build a flow of **Instruction, Image** (in some cases)then **Screen Recording** sequences per task.The default test is called "Warm-up Tasks" (as specified when you first created the Word Tasks Study).



Underneath the main tab selector Studio shows the possible types of media that you string together to assemble a stimulus sequence for the viewer. You will be using the **Instruction** and **Screen Recording** media types for this experiment.

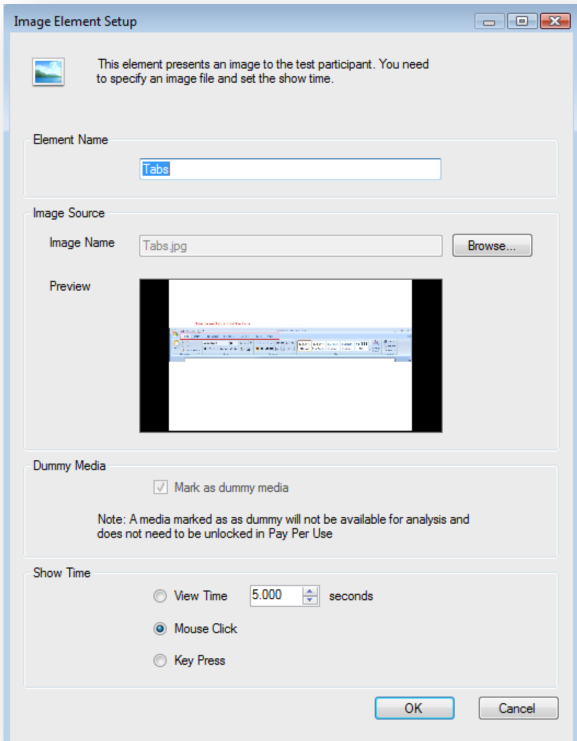
When you drag and drop a placeholder for an **Instruction** into the central media viewer, Studio pops up an **Instruction Element Setup** dialog which lets you define the task and navigation sequence to move to the next screen.

You will have one **Instruction Element** per task. For your own reference and data organization, name the task number and its basic goal in **Element Name**. Copy the Task instruction into the ***Preview*** area (this is what the participant will see). Finally, change the ***Show Time*** default to ***Mouse Click***. This will allow the user to transition from the task instruction to the screen recording via a mouse click.

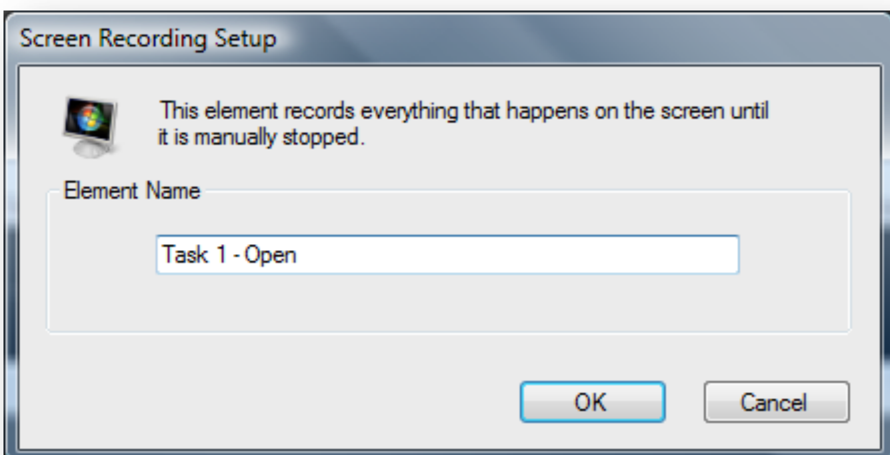


Verbiage: All tasks should conclude with the following sentence: *When you have completed this task please hit the "F10" key*.F10 will stop the screen record and transition the user to the next task (but not end the study).

There will be cases in the warm-up tasks where you will want to give the participant a picture before he begins the task (e.g. a visual of the "Tabs" so the Participant understands the purpose of the warm-up).



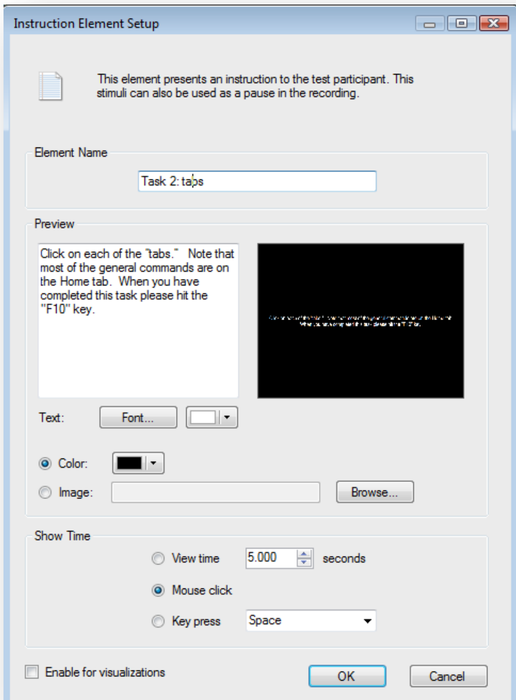
Once the task is defined (and image, if appropriate) and saved to the timeline, add a Screen Recording instance. Your only option for Screen Recording is to name the element, which you should name by task.

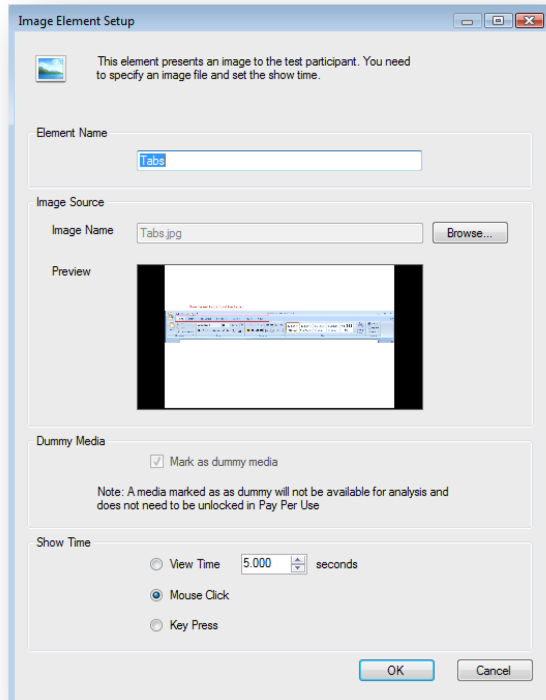


WARM-UP TASKS

Tasks 1-5 are warm-up tasks for your participant. The warm-up tasks ensure the participant is familiar with the basic elements and vernacular of the Microsoft Office Word 2007 user interface. Use the Task List document to copy and paste the warm-up tasks into your study design. You should plan to observe your participant performing the tasks to ensure tasks are performed correctly and the participants is exposed to the correct areas of the Word 2007 user interface before you begin the formal task list. This will ensure a level of consistent knowledge of the Word UI between participants.

For tasks with images, you will paste the task instructions into the ***Preview*** area. You will then add an Image type and add the tasks' corresponding picture via the ***Image: Browse…*** feature. All task images are saved to your machine's desktop. Note, tag this image element as "Dummy Media" (check the box "Mark as dummy media"). This will omit the image from your eye tracking analysis. Finally, ensure you use "Mouse click" as the transition method between elements.



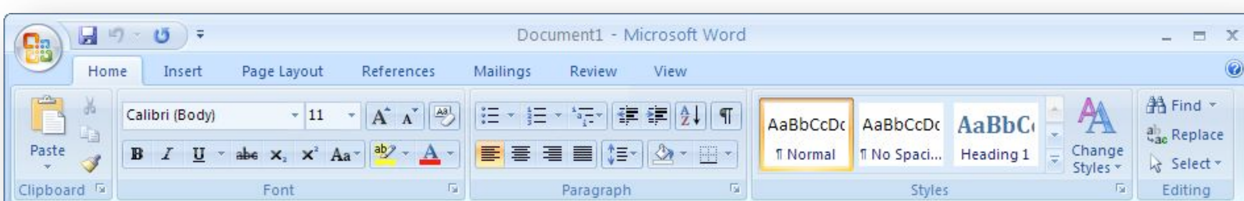


Task 1 – Open a document

From the desktop, open the file "Play.docx". If you get a dialog box titled "User Name", do not add any information to the dialog. You simply have to click OK.

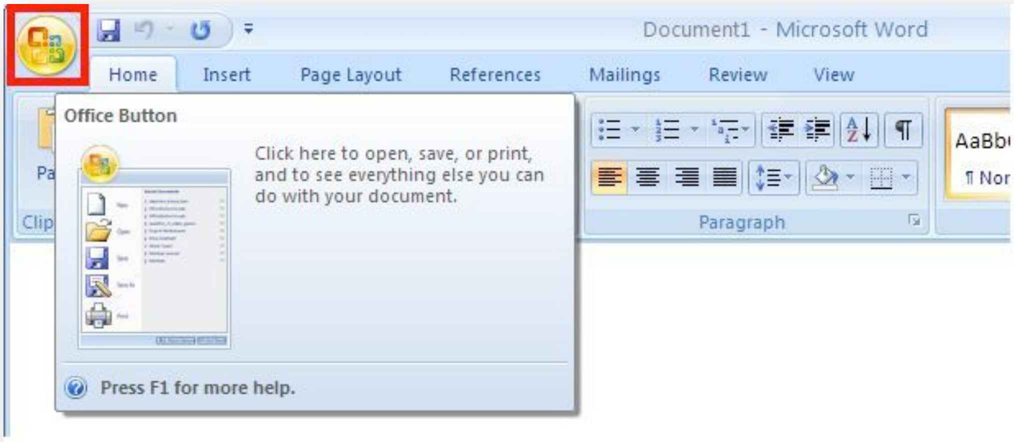
Task 2 – Become familiar with Tabs

Click on each of the "tabs." Note that most of the general commands are on the Home tab.



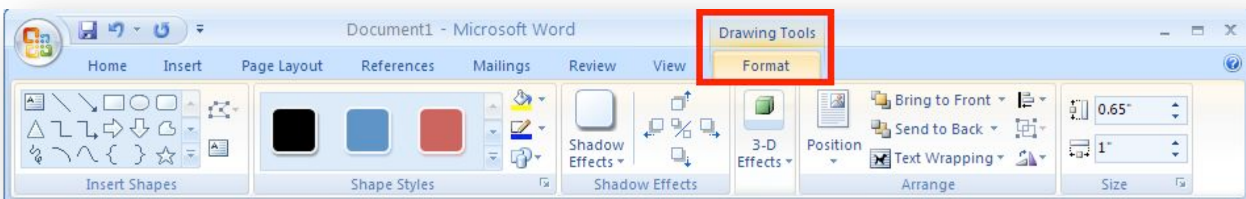
Task 3 – Access and review the Office Menu

Click the Office Button to find file commands like Open and Office tasks like Share. Note that frequent commands like Save and Undo appear in the Quick Access Toolbar next to the Office Button.



Task 4 – Become familiar with Contextual Tabs

Click the blue box and then click the Drawing Tools Format tab to the right of the normal tabs. Some objects have tabs that appear only when the object is selected. Notice that as you click outside of the blue box, the tab goes away; when you click the blue box again, it reappears.

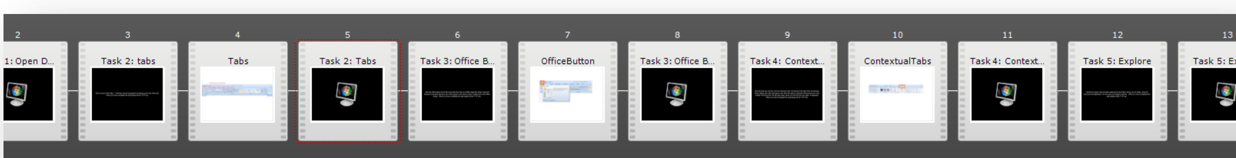


Task 5 – Explore Word

Feel free to spend a few minutes exploring the new Office. When you are ready, close the document and application. Do not save any changes to the file.

Study Timeline

Your warm-up task timeline should look something like this once completed:

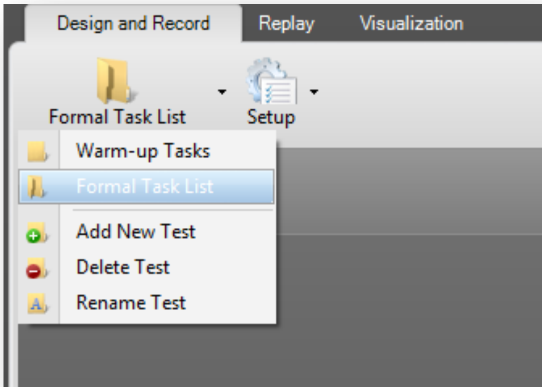


Formal Task List

The formal task list follows. These are the tasks you will observe the user perform in real-time, taking observational notes about their behavior that is indicative of their understanding of the Word 2007 UI. These behavioral notes should include specific eye movement patterns, mouse clicks, facial expression, apparent emotional state and verbal comments. As noted previously, these notes should always be within the context of the task. The study administrator should be continually comparing the 'correct' path against the observed behavior.

Unlike the warm-up tasks, images that are part of the formal task list will not be included in the Tobii task sequence. This is due to the fact that participants will need to continually refer to the pictures in the tasks. To ensure the participant can refer to the pictures at will, these tasks should be **printed out as a formal task list as supplementary material to the study**.

To ensure the eye tracking data you collect against the formal task list is easy to access, you should create a New Test in the Design and Record tab. Name this task list "Formal Task List".



STUDY ADMINISTRATOR PREPARATION

The study administrator should perform the following tasks to prepare to run participants:

1. Read and understand each task
2. Watch the "Correct" path sequence video which can be found in the Replay section of Studio

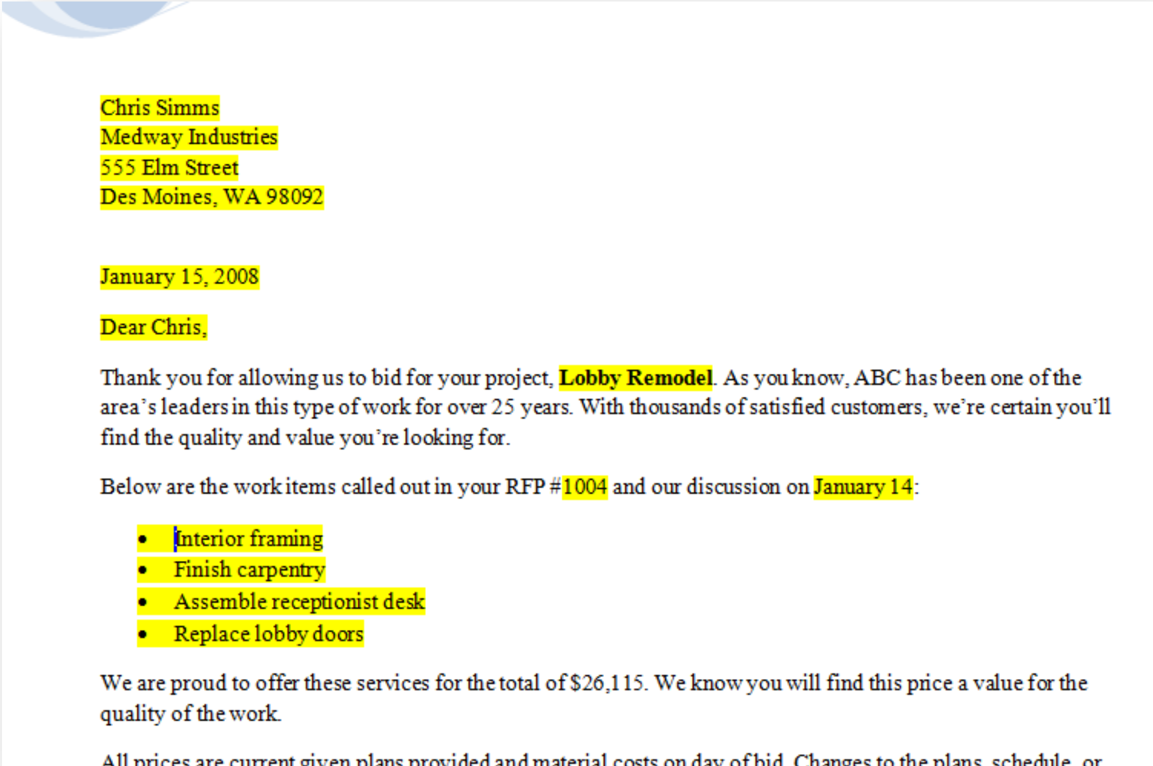
TASK 1 – OPEN FILE

From within Word, find the Open command.

Using Word's Open dialog, open the 'Proposal.docx' file. It is located on your desktop, but not leave Word to open it—the task is to open the file from within Word.

TASK 2 – MATCH FORMATTING

Change the document on screen to match the document below. I've highlighted the changes you need to make.



Chris Simms
Medway Industries
555 Elm Street
Des Moines, WA 98092

January 15, 2008

Dear Chris,

Thank you for allowing us to bid for your project, **Lobby Remodel**. As you know, ABC has been one of the area's leaders in this type of work for over 25 years. With thousands of satisfied customers, we're certain you'll find the quality and value you're looking for.

Below are the work items called out in your RFP #1004 and our discussion on January 14:

- Interior framing
- Finish carpentry
- Assemble receptionist desk
- Replace lobby doors

We are proud to offer these services for the total of $26,115. We know you will find this price a value for the quality of the work.

All prices are current given plans provided and material costs on day of bid. Changes to the plans, schedule, or

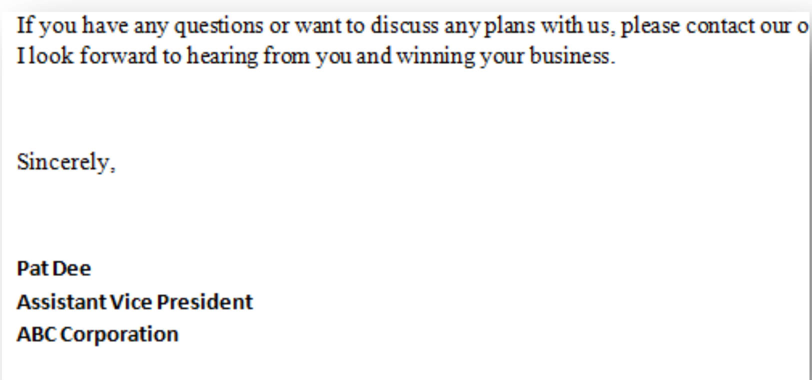
Task 3 – Set Margins

Set the margins to exactly 1" on top and bottom and exactly 1'' on left and right. (The margins must be exact--do not use the Ruler.)

Task 4 – Text Styles

You notice that the signature is not using the proper style as prescribed by your company. This corporate template has a style called Strong.

Find a way to apply the **Strong** style to the signature so your document looks like this:



If you have any questions or want to discuss any plans with us, please contact our o
I look forward to hearing from you and winning your business.

Sincerely,

Pat Dee
Assistant Vice President
ABC Corporation

Task 5 – Table Formatting

There is a table of references on the last page. You want to change the appearance of the table to make it look better. Make your page look like the example below. Note the text color, shading, and borderlines:

References

Name	Number	Type of Work
Katja Fascher 4208 869th Ave NE Somerville, WA 98073	206.234.5678	First floor remodel
Elena Volpone 43 Sage St Sammamish, WA 98052	425.908.3621	Exterior porch addition
Alexander Moore 7277 Ross Lane Copperville, WA 98034	800.234.9845	AmTech Lobby
Kimberly Alden 10322 W Battenkill Troutdale, WA 98234	310.432.9300	GenCo waiting room

Task 6 – Change Default Bullets

The list of work items on the first page uses the default bullet appearance, a simple filled circle. Find a way to change the bullets to checkmarks, so your list appears like this:

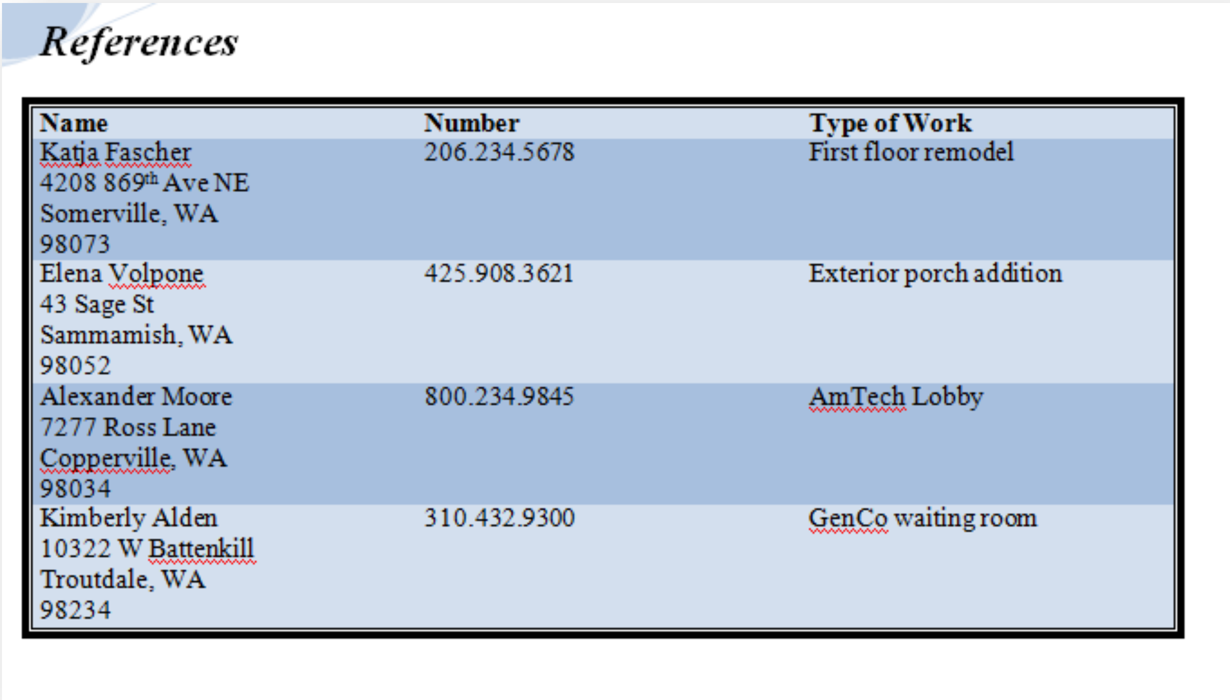
Below are the work items called out in your RFP #1004 and our

- ✓ Interior framing
- ✓ Finish carpentry
- ✓ Assemble receptionist desk
- ✓ Replace lobby doors

We are proud to offer these services for the total of $26,115. We the quality of the work.

Task 7 – Add Table Borders

You want to change the border box around the outside of the table. A colleague has told you that Word has a border style which you can apply to the whole table to make your table look like this (a thick line and a thin line):



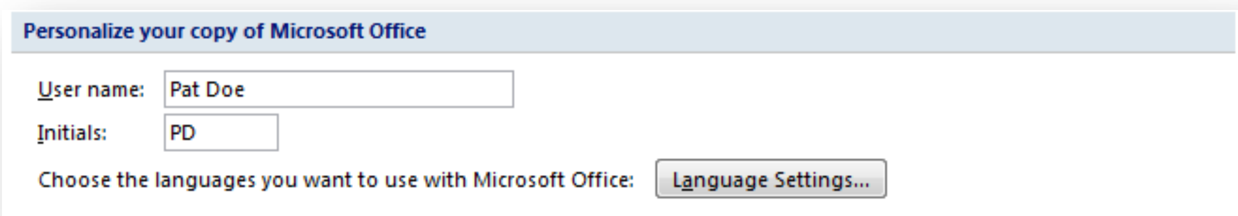
References

Name	Number	Type of Work
Katja Fascher 4208 869th Ave NE Somerville, WA 98073	206.234.5678	First floor remodel
Elena Volpone 43 Sage St Sammamish, WA 98052	425.908.3621	Exterior porch addition
Alexander Moore 7277 Ross Lane Copperville, WA 98034	800.234.9845	AmTech Lobby
Kimberly Alden 10322 W Battenkill Troutdale, WA 98234	310.432.9300	GenCo waiting room

Task 8 – Add User Name

There are many features in Office that work better if you personalize your copy of Office by entering your name and initials.

Find a way to tell Word that your name is “Pat Doe”.



Task 9 – Use Help

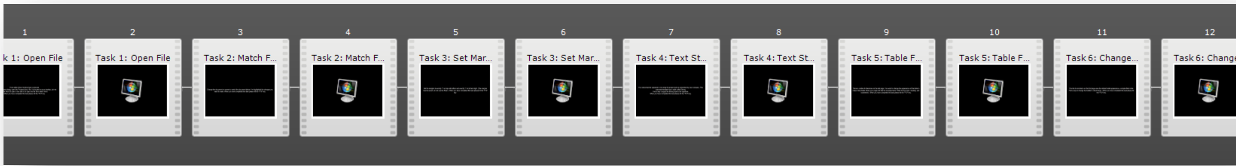
Find instructions for how to add page numbers to your document.

Task 10 – Customize Quick Access Toolbar

You like to use the Print Preview option often while creating a new document. Find a way to add a shortcut to Print Preview to the Quick Access Toolbar so it’s easy preview your document.

Study Timeline

Your warm-up task timeline should look something like this once completed:



Running the Study

Run a single participant at a time, with all participants attempting to complete the full task list. Participants should work quietly, as the practice of verbal protocol will compromise the eye movement data. As with any traditional usability study, it will be important to observe the participant's interaction with the UI, mouse clicks, task outcome, and any other overt behavior (e.g. eye movements) throughout the study. Your observational notes should include eye movement data as it occurred in situ, for later analysis.

You'll notice that all users will have different start and stop time stamps for each task. The task duration will also be different per user, and should be expected. If necessary, you will coordinate the task data between users during analysis.

Analysis and Reporting

Degree and depth of analysis is driven by observations made during the study. Eye tracking analysis plan should be outlined during the study when particularly interesting eye movement patterns are detected. Eye movement patterns of interest include:

1. Patterns that precede task errors
2. Patterns that elongate task time
3. Patterns that illustrate common user confusion (focus on misleading UI)
4. Patterns that help to 'diagnose' issues with the user interface (e.g. Office file menu examples)

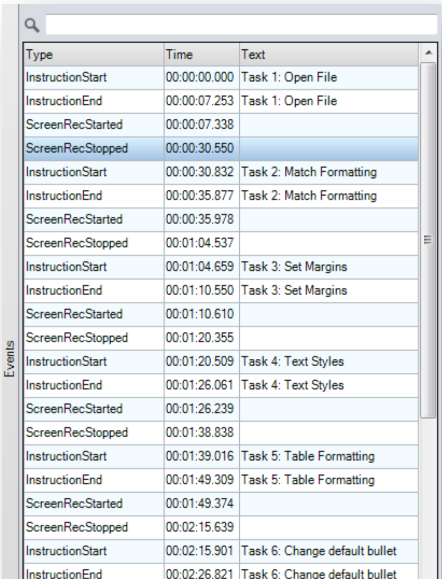
Replay: Scenes of Issues

Use replay to verify assumptions, observations from the log, and tag task-level that you plan to help visualize observed issues. Each user and study is different, so you should expect to create a post-hoc (or at least in situ) analysis plan.

As mentioned in the Analysis Examples section, there are a number of eye movement visualizations that can you can create in Tobii Studio. Depending on what you observed, you can create video snippets, heat map, gaze maps, bee swarms, etc. To accomplish these analyses in Studio, each eye tracking video must be manually tagged.

Tagging Scene Video for Analysis

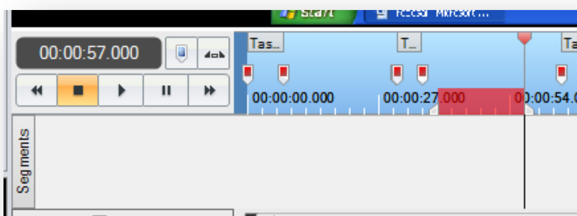
In the Replay section of Studio, all recording are pre-segmented based on the original study protocol timeline (e.g. you can navigate your video based on each timeline element, like the start record time for the "Task 2 Instruction Start"). These markers will help you navigate the video recordings.



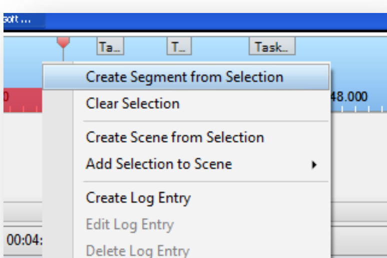
Type	Time	Text
InstructionStart	00:00:00.000	Task 1: Open File
InstructionEnd	00:00:07.253	Task 1: Open File
ScreenRecStarted	00:00:07.338	
ScreenRecStopped	00:00:30.550	
InstructionStart	00:00:30.832	Task 2: Match Formatting
InstructionEnd	00:00:35.877	Task 2: Match Formatting
ScreenRecStarted	00:00:35.978	
ScreenRecStopped	00:01:04.537	
InstructionStart	00:01:04.659	Task 3: Set Margins
InstructionEnd	00:01:10.550	Task 3: Set Margins
ScreenRecStarted	00:01:10.610	
ScreenRecStopped	00:01:20.355	
InstructionStart	00:01:20.509	Task 4: Text Styles
InstructionEnd	00:01:26.061	Task 4: Text Styles
ScreenRecStarted	00:01:26.239	
ScreenRecStopped	00:01:38.838	
InstructionStart	00:01:39.016	Task 5: Table Formatting
InstructionEnd	00:01:49.309	Task 5: Table Formatting
ScreenRecStarted	00:01:49.374	
ScreenRecStopped	00:02:15.639	
InstructionStart	00:02:15.901	Task 6: Change default bullet
InstructionEnd	00:02:26.821	Task 6: Change default bullet

CREATE VIDEO SNIPPET, GAZE PLOT, ETC.

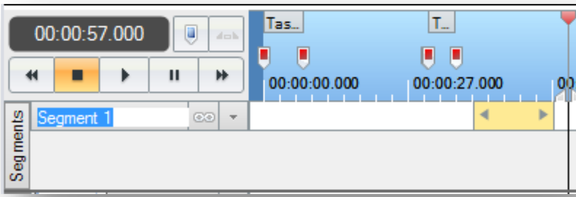
Your first task is to manually tag your recordings. To do this, first position the segment tool to mark the beginning and end of the snippet, e.g.:



Once the segment has been defined right-click inside the red area and select "Create Segment from Selection":

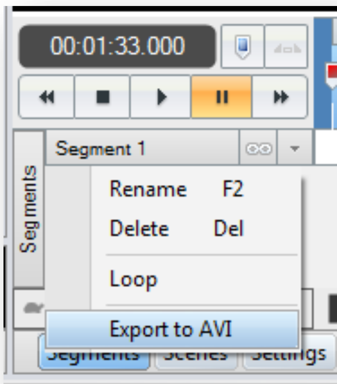


This will create a "Segment" that you can name:

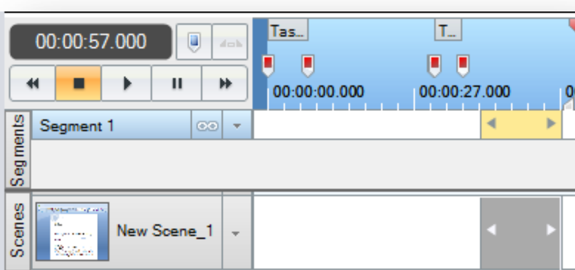


At this point you have a couple options:

a) Create a video snippet (AVI) of the segment



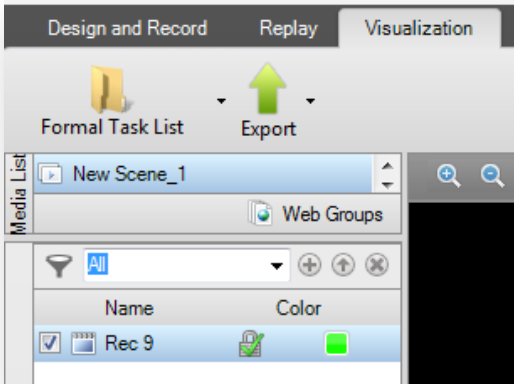
b) The segment can be dragged down to the "Scene" area where a new Scene will be created:



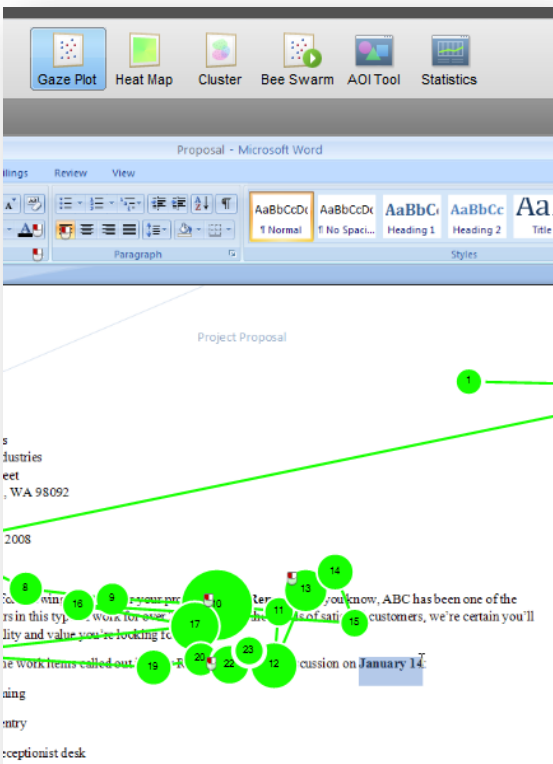
Note: once a segment has been transformed into a Scene, that Scene is now available as a tag for all other recordings.

VISUALIZATION: IMAGES OF ISSUES

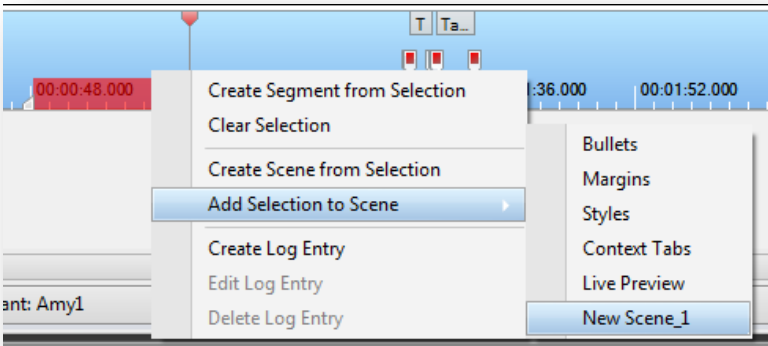
All scenes are available in the "Visualization" tab in the Media list. You'll notice is only has one recording associated to it:



Any visualization type you select will display an analysis for just that recording. In this case, a Gaze Plot is most appropriate for a single user analysis:



It's also available as a 'tag' for the other recordings in your study in the Replay tab. Therefore, if you want to create a heat map or bee swarm for that particular issue/task with multiple recordings, you can create a Segment, then use the "Add Selection to Scene" and select said scene.



Once a Scene has multiple recordings associated to it, the visualization options such as the Heat Map, or Bee Swarm are appropriate to use. However, this should only be the case if your research findings support the need for said visualizations, e.g.:

Heat Map and Cluster

- a single map that boldly illustrates user lack of attention to the "correct" area of the UI
- multiple maps, each for a particular user group (e.g. successful vs unsuccessful)

Bee Swarm

- illustration of common or different user confusion (where there is convergence, then deviation