

Reviewing Nyan 2.0XT by Recreating the Burke et al. (2004) Banner Ad Study

Kyle Vessey and Anthony J. Hornof

January 11, 2013

University of Oregon

Department of Computer and Information Science

Technical Report CIS-TR-2013-01

Executive Summary

This document attempts to provide an educated review of the eye tracking software package Nyan 2.0XT (Interactive Minds Dresden, <http://www.interactive-minds.com/>) to someone who is familiar with the creation, the running, and the analysis of rigorous scientific studies. This review focuses on three important considerations that need to be considered in an eye tracking study. These considerations are: (1) During data collection the gaze data will need to be constantly monitored to ensure that accurate gaze samples are collected, (2) after data collection the gaze data may need to be cleaned up by applying post-hoc error correction techniques, and (3) gaze data will need to be organized in a clear, understandable, and usable format for analysis. This review was motivated by the University of Oregon (UO) Spatial and Map Cognition Research Lab (SMCRL) asking for our input of how we run experiments at the (UO) Cognitive Modeling and Eye Tracking (CMET) lab, and how they might be able to use Nyan 2.0XT in their research. The SMCRL wanted to figure out where is the data, what is the data, and how to get the data out of the Nyan 2.0XT environment to import into the R statistical analysis software (The R Foundation, <http://www.r-project.org/>). In order to answer these questions I (the first author) recreated the banner ad study from Burke et al. (2005). In my exploration of the Nyan 2.0XT software I came to the conclusion that I would not recommend it for use in a rigorous scientific study because it fails to provide adequate features to thoroughly address the above important considerations for running an eye tracking study.

Nyan 2.0XT is a software package that is used to create eye tracking experiments and analyze the data from those experiments. A positive feature of Nyan 2.0XT is that it handles data

well, and gives full access to that data by allowing it all to be exported out of the Nyan 2.0XT environment and to be imported into other software such as R for analysis. This data includes raw gaze data, fixation/saccade data, and other common events such as mouse clicks. Another positive feature is that areas of interest (AOIs) can be defined using the AOI definitions tool, and fixation count and duration on AOIs can then be exported out from the Nyan 2.0XT environment by using the AOI metrics tool. So there are positive features with the way Nyan 2.0XT handles data, but unfortunately there are also some major problems with the way that data is collected and formatted.

The major problem of Nyan 2.0XT during the data collection stage is that it does not provide any means to recalibrate the eye tracker during an experiment. It is good that Nyan 2.0XT allows the experimenter (that is, the lab technician who is running the experiment) to set up a second display to monitor the eye images and raw gaze samples by use of the Local Live Viewer, but the problem is that there is nothing that can be done to maintain accuracy when issues occur other than telling the participant to change his or her position. At the CMET lab we know that sometimes the recalibration of the participant is necessary, and one way we address this issue is to design fixed times within the experiment to allow for recalibration. Not only is there no way to set a time to recalibrate in Nyan 2.0XT, there is not even a way to initiate a recalibration without stopping the recording and starting over from the beginning.

The major problem with the format of the data exported is that it does not produce data that is clear, understandable, and in a usable format for analysis. The data exported from the AOI metrics tool has four troublesome issues: (1) the exporting tool creates a separate data file for each stimuli in the experiment—for example, exporting AOI metrics from ten different stimuli will produce ten different data files—that will need to be combined for final analysis, (2) there is no way to generate total fixation duration or count on a group of AOIs because the AOI metric tool will only report the mean fixation duration or count across the AOIs in the group, (3) the total number of columns in the data is excessive, and (4) there can be—and almost always will be—empty cells in the data. Some sort of post export manipulation will need to be carried out in order to get the data into a clear, understandable, and usable format for analysis.

Another major problem with Nyan 2.0XT is that there is no way to import data back into the Nyan 2.0XT environment. Importing data back into the environment would be beneficial because it is sometimes necessary to apply post-hoc error correction techniques to clean up the systematic error—the difference between the gaze reported by the eye tracker and the actual gaze maintained by the participant—from the gaze data (Hornof and Halverson, 2002). Although it may not be necessary to apply such techniques in all eye tracking experiments, having the ability to do so is important.

In my personal opinion I would not use Nyan 2.0XT to create and run an experiment because it fails to provide adequate features that thoroughly address the important considerations for running an eye tracking study. There is no simple way to recalibrate a participant to fix for inaccuracies in the gaze data while data is being collected, there is no way to apply post-hoc error correction to the gaze data after data has been collected, and the data produced by Nyan 2.0XT requires further effort and time to transform it into a clear, understandable, and usable format for analysis. Nyan 2.0XT may be useful in industry for simple user testing of software, but when it comes to rigorous scientific studies the best option for creating and running experiments is to write your own code.

Initial Nyan 2.0XT Exploration and First Impressions

To start my Nyan 2.0XT exploration I first read the manual. The manual did a decent job of explaining how to use the Nyan 2.0XT software, but it did not seem to get too in-depth on any one feature of the software. Reading the manual taught me which buttons to push so I could create an experiment, run a participant, and how to get that data out of Nyan 2.0XT. The manual made the process sound fairly straightforward and simple, but as I will explain, actually getting the data that I would want to work with is a bit more complicated than the manual would have you believe.

Project types

There are three types of projects that can be created with Nyan 2.0XT: *screenrecording*, *mediashow*, and *web projects*. *Screenrecording* projects capture the screen content as a video, *mediashow* projects present stimuli through a series of slides, and *web projects* present web pages. The selection of project type will determine what analysis tools are available. For example, mediashow and web projects have access to the *Clickmap* tool, which generates a heat map for mouse click data, while the screenrecording does not create such a map because a click map does not make sense on a video, but it does make sense on a series of slides or webpages. This is important to know because if a particular tool is needed, then the right project type must be selected. I created a project of each type and played around with the tools available to create stimuli and analyze data. I did not fully explore every feature of each project type but I believe I explored enough of each to get an overview of the capabilities of each type.

Screenrecording Projects

Screenrecording projects are good for any stimuli that cannot be displayed as a series of slides, such as a study in which the interface of a piece of software is being evaluated. The video can be divided into static sections after it is captured by use of the *VideoSectioning tool*. These static sections can be analyzed by the other tools available, such as the AOI tools.

Mediashow Projects

Mediashow projects are good for stimuli that are presentable in a controlled manner by means of a series of slides. All the normal slide stimuli options are available while creating the stimuli, such as how to advance the slide, background colors, and so on. Order of presentation is defined after stimuli have been created. Stimuli can be added to blocks which can be sequential or random, and blocks can be nested within other blocks as well. One thing I did appreciate is that the interface for defining and organizing these blocks is intuitive and clear. I remember

trying to do the same task when I first used E-prime, and I do not remember it being as quick and easy to learn as the Nyan 2.0XT interface made it.

Web projects

Web projects are good for stimuli that can be displayed in a web browser. Within the web project there are four types of stimuli: WebPages, WebTasks, WebScenes, and Dynamic Elements. WebPages are simply web pages, nothing fancy. A WebTask is a certain task that involves navigating through several web pages. Meta information can be added to the task such as a task description or task timeouts. WebScenes allow multiple webpages to be grouped together so analysis can be done on the same stimuli across multiple participants. Nyan 2.0XT will automatically assign any web page with the same URL to the same WebScene, but it also allows WebScenes to be manually edited and defined in case some web pages have different URLs but are visually identical. The automatic WebScene generation is a nice feature because manually creating WebScenes across an array of participants could quickly become tedious and time consuming. The last type of stimuli, dynamic elements, are parts of web pages that show dynamic visual behavior. These dynamic elements can be filtered out during analysis. Dynamic elements are one feature that I did not explore too thoroughly. I really do not understand them completely, so I cannot say too much about them.

Collecting Data

Actually running an experiment and collecting data is pretty similar across all project types. Participants must be added into the system by the *Manage Subjects* tool. Select the participant from the *Subjects/Stimuli* menu then push the start recording button to begin collecting data for that participant. Nyan 2.0XT has a tool called *Local Live Viewer* that can be used to monitor the eye images and raw gaze samples from another display during data collection. Figure 1 shows the Local Live Viewer on the left, and what the participant would see on the right. This is a nice feature because it is essential to monitor the status of the eye tracker

in order to get accurate eye data. One problem however, is that I do not believe there is any way to pause a recording to allow for recalibration. At the CMET lab we know that sometimes the recalibration of the participant is necessary, and one way we address this issue is to design fixed times within the experiment to allow for recalibration. If we wanted to implement these fixed recalibration times with Nyan 2.0XT, then I believe we would have to actually make separate projects. Nyan 2.0XT partially considers and provides functions for the factor of monitoring gaze data accuracy during data collection, but it does not fully consider this factor because there is no way to fix poor accuracy other than telling the participant to try to move back into his or her original position.

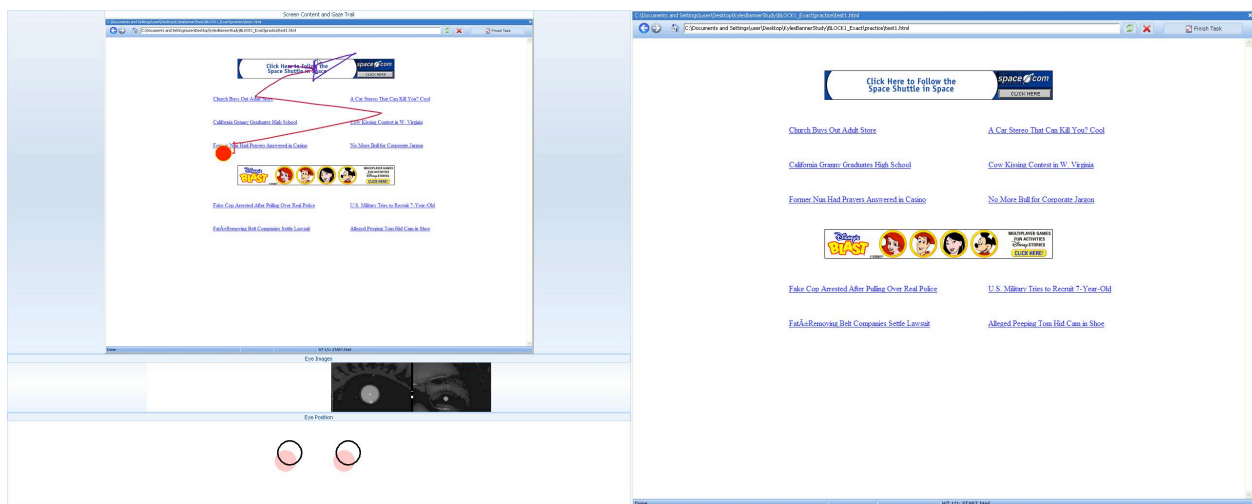


Figure 1. On the left, the experimenter's display, showing Local Live Viewer. On the right, the participant's display.

It is important to note that once any participant has been run on a set of stimuli, Nyan 2.0XT will not allow the stimuli to be changed unless the experimenter specifically pushes a button to unlock the stimuli. I think this is a good idea, but it is not practical. Those who prescribe to iterative design principles will create stimuli by making it, testing it, evaluating it, and finally make changes to it based off of the testing and evaluation. A test participant must be made in order to test out the stimuli. If changes are to be made after the test run, then Nyan 2.0XT will pop-up a warning message, and a button will need to be pushed to allow editing. I found this incredibly annoying. I think there is no reason to have an edit lock in place because

any competent scientist will know not to change his or her stimuli after its been finalized and participants are being ran.

Visualizing the Data

Nyan 2.0XT would call this step analyzing the data because it uses the Analysis tools, but I refuse to call it analyzing the data because the tools are just not useful to me as an analyst. These tools provide the common visualizations such as heatmaps, scanpaths, and video overlays, each with various options to change things up and make them look prettier. These types of visualizations are not bad, and Nyan 2.0XT does a good job of making them, but I think they are only useful for getting an overview of how much error was in the data after collecting it. The only tools in the collection of analysis tools that I found useful for actual analysis of data was the *AOI definition* and the *AOI metrics* tool because they allow measures such as fixation duration and count on AOIs to be calculated and then exported out of the Nyan 2.0XT environment. Figure 2 shows an AOI is being defined with a large polygon over the image of an airplane with the AOI definition tool on the top, and colored rectangular AOIs defined over links on a web page with the resulting spreadsheet of data with the AOI metrics tool on the bottom. I will get more into AOI definition and AOI metrics exporting later in my discussion of recreating the banner ad study.

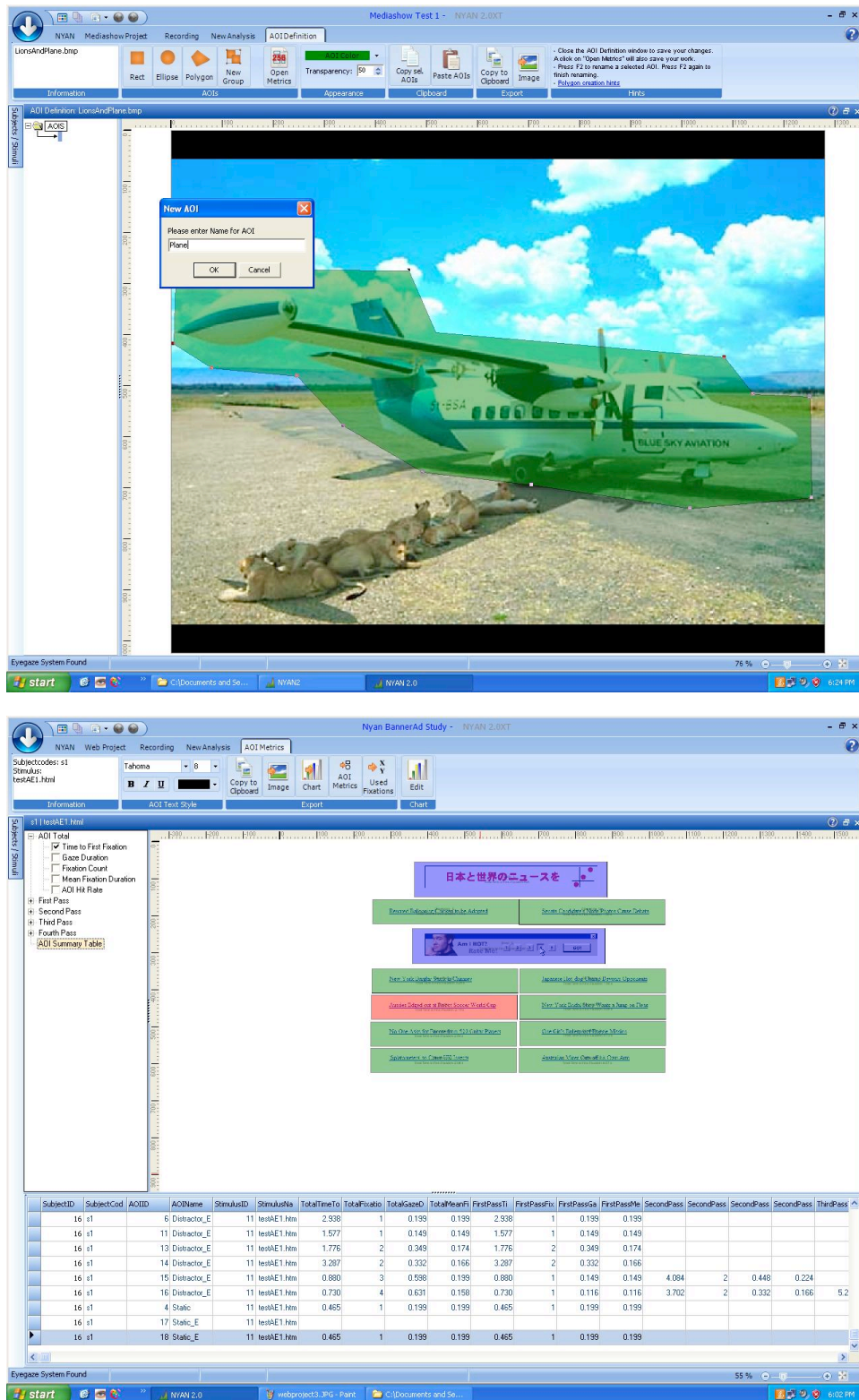


Figure 2. On the top, an AOI being defined over the image of an airplane. On the bottom, the AOI metrics tool, with rectangular AOI regions defined, and a huge detailed spreadsheet of data.

Exporting Raw Data

Various data can be exported with the data export tools provided by Nyan 2.0XT. The common raw gaze data, fixations/saccade data, mouse events, and so on can all be exported from the Nyan 2.0XT environment. The manual has a good reference section on what type of data can be exported from the various project types, so I will not get into any of those specifications.

One issue that I have with the export of data, though, is that it is not combined across data types. For example, I like to have all my data from one participant in one file, so that I can parse through one file to get the whole story. That is, the eye data is combined with the mouse data, which is also combined with event data, and it all flows together as one stream. So I know that at time 3500 the eyes were at point (x,y), then 5 ms right after that at time 3505, the participant clicked the target button. This data is available in Nyan 2.0XT, but it would be spread out across multiple data files that would require some combining to get it into the form that I want. The reason I like to have all my data in one file as one stream like this is because it makes writing parsing scripts much easier, which means that data for analysis can be extracted relatively quickly.

Nyan 2.0XT provides features to filter the raw gaze data down into fixations, saccades, and measures on AOIs, but the format that the data is in when it is exported out of the Nyan 2.0XT environment is not very usable and requires further effort and time to transform it into a clear, understandable, and usable format for analysis. During my exploration with Nyan 2.0XT, I did not find a single instance in which it would give me data that I would be satisfied with as is. In all cases I would have had to apply some sort of post-export manipulation to the data files into a usable format for me.

Final Note: the importance of allowing data to be imported to analysis software.

The ability to apply post-hoc error correction to clean up systematic error—the difference between the gaze reported by the eye tracker and the actual gaze maintained by the participant—from the raw gaze data is one of the important considerations that needs to be considered when

creating an eye tracking experiment, and Nyan 2.0XT does not provide any features that even modestly address this factor. The best solution for Nyan 2.0XT would be to implement post-hoc error correction techniques within its software like the software VizFix (Hornof & Halverson, 2002; Zhang & Hornof, 2010), which we created and use in the CMET lab. A simpler function Nyan 2.0XT could have implemented would be to have a data import option. This way the data could be taken outside of Nyan 2.0XT, post-hoc error correction could be applied by some other software or script, and the new more accurate data could be imported back into Nyan 2.0XT. The more accurate data could then be used to calculate more accurate AOI metrics, which would lead to stronger and truer results.

Case Study: Banner Ad Study Recreation

I recreated the banner ad study that was done in Burke et al. (2005)— which I also recreated using Tobii Studio back in 2009 (Vessey & Hornof, 2009)—in order to answer the questions of where is the data, what is the data, and how to get the data out of Nyan 2.0XT environment in order to import it into R.

Description of the Banner Ad Study

The experiment had 36 trials. Each trial consisted of a precue page followed by a searching page. The precue page contained the precue at the top of the page written in black 14pt Arial font and a link to the searching page labeled “GO”. Two types of precues were used: exact precues and semantic precues. *Exact* precues matched the headline exactly word-for-word, and *semantic* precues consisted of a few sentences that appeared to be from the the target headlines story. For example, the semantic precue for “Drop-outs doing just fine, thanks” is:

New research debunks the common belief that leaving school before completing
year 12 diminishes a teenager’s chance of a successful career.

A total of 24 searching pages were built, with each page containing two banner ads with one target headline and nine distractor headlines. On the searching page the target and

distractors were randomly located in a 2x5 grid. All target and distractor headlines were colored blue and in 14pt Arial font. Each searching page contained two banners ads of one of the following types: static, animated, or solid gray. *Static* banners were ads that only containing non-moving images, *animated* banners were ads that contained images that moved or flashed, and *solid gray* banners were plain gray rectangles with no images. Both banner ads on a searching page were of the same type; one banner was always located at the top of the page, and the other was located randomly between two of the five rows of headlines.

After clicking a link in the search page the participant was sent to a “pass” or “fail” page which informed the participant if he or she clicked on the correct headline. When navigating away from the pass or fail page the participant was presented with the precue for the next trial. The trials were separated into two different blocks. Each block consisted of an explanation of the search task, one example trial, and 12 experimental trials. Only exact precues were used in the first block of trials, and only semantic precues were used in the second block of trials. Four researchers from the CMET lab participated in the experiment.

Creating the stimuli and project

To recreate the banner ad study in the Nyan 2.0XT environment, I first needed to decide the project type. For a screenrecording project I could write a separate program that would present the stimuli to the participant, for the mediashow I could recreate the stimuli in Nyan, and for the webstudy I could just create web pages that link to each other.

Project goals influenced which project-type to use. The purpose of the banner ad study is to determine if animated banners attract more visual attention than other types of banner ads in a top-down visual search task. It has been shown that fixation count and length can be used to quantitatively determine how much visual attention an object receives (Buscher, Cutrell, & Morris, 2009). So we will want to extract fixation count and length from AOIs defined on the stimuli.

All three project types allow us to define AOIs and extract fixation count and length through the AOI metrics tool, but I would have to do it differently for each type of project. In the

screenrecording I would have to spend time dividing each participant recording into sections and scenes in order to define the AOIs. This would be incredibly tedious because we would have to do it for each stimuli and each participant. The best way would probably be to use a mediashow project because the presentation order of the webpages could easily be randomized within blocks.

I ended up choosing to do a web study project for a very simple reason: I already had the webpages for the banner ad study recreated in html from the 2009 technical report evaluating Tobii studio. This choice was reasonable because of Nyan 2.0XT's ability to combine web pages with the same URL into WebScenes, which would allow me to define AOIs within each WebScene one time for all participants who participated in that WebScene. The down side to this choice compared to using the mediashow type was that each participant received the same order of the web pages. I could have created different paths for each participant that would randomize the order of the web pages, but due to time constraints I did not get this involved in recreating the study.

Defining AOIs with the AOI Definition tool

As I mentioned above, each similar stimulus web page across all participants was combined into WebScenes. The good news was that I could define AOIs on each WebScene and it would allow for analyzing across all participants for that WebScene. The bad news was that there was a total of 24 different WebScenes, and I had to define all the AOIs for each one by hand. This is a common problem to these types of software packages that require manual definition of AOIs. Nyan 2.0XT only makes this tedious task a bit easier by allowing AOIs to be copied and pasted. This is one benefit to writing personalized experimental software—like we do at the CMET lab—because AOIs can be defined during stimulus generation, and then saved within some configuration file after the experiment has been ran, which is how we would normally do things in the CMET lab. That configuration file can then be loaded into the analysis software with the raw data.

I defined a total of 12 AOIs per WebScene. Nine distractor headline AOIs (called Distractor_E), one target headline AOI (called Target), and two banner AOIs. Banner AOIs were either static banners, gray banners, or animated banners (called Static, Gray, or Anim). Nyan 2.0XT allows for AOIs to be grouped, so I grouped every similar AOI into a common group. That is, there were nine Distractor_E AOIs in the group Distractor. I would need to reorganize the AOIs for each WebScene so that the AOIs lined up properly because I was copying and pasting AOIs across WebScenes. This would lead to occasional mistakes like pushing the wrong button or accidentally deleting an AOI, and this annoyance is exacerbated by the lack of an undo function in Nyan 2.0XT to rectify these little mistakes.

Exporting fixation length and count with the AOI metrics tool

Selecting the AOI metrics tool brings up a graph that can be changed to show fixation count and length. I did not find the generated graph useful for anything, but the ability to export the metrics that were used to create the graph is useful because that data can be taken and imported into R. The downside with the exporter, however, was that I had to do this for every single WebScene because although data is extracted into a single file for all participants that encountered that WebScene, a separate file is generated for each individual WebScene. This led to a total of 24 different data files that needed to be combined into a single file. This is a relatively simple task that can be completed by writing a simple script, or just by copying and pasting, but I would have liked it if Nyan 2.0XT had better ways for dealing with this. Some sort of batch export feature would have been nice.

Another issue with the AOI metrics tool is that it reports too much data. Fixation duration and count are separated into AOI Total, first pass, second pass, third pass, and fourth pass. AOI Total is the total measure for the entire WebScene. First, second, third, and fourth passes apparently separate the data based off of when the gaze first enters an AOI, the second time the gaze enters an AOI, and so on. I am not really sure about these passes because I see no reason to use them, and I was not interested in them at all.

The problem here is that Nyan 2.0XT will export all fields of the AOI metrics data table. So the data file ends up being twenty-something columns long. I find this completely ridiculous. It makes it hard to just look at the data in a text editor and have it make sense, which is the first thing I always do when I generate any kind of data file. I want to make sure each row looks right and makes sense. With this twenty-something columns ridiculousness, I practically need three wide screen monitors to be able to make my text editor wide enough so I can inspect the rows of my data files. A simple script would make my data files more manageable of course, but I just find it annoying that I have to deal with either writing a script or a ridiculous data file when Nyan 2.0XT could have simply asked me what fields I want to export during the AOI metrics exporting process.

One other small annoyance with the AOI metrics tool is that it will report a measure for grouped AOIs, but that measure is arbitrary with no way of defining it. Again, I have a total of nine Distractor_E AOIs grouped into the group Distractor. I wanted to know the total time the gaze spent on the distractor AOIs, but the Distractor field in the AOI metrics reported the mean fixation length for all nine Distractor_E. Now to get the data that I want to work with I need to go through and add up all Distractor_E fields with a script. This is not an earth shattering problem, but it would have been nice if I could choose exactly what I wanted that AOI group measure to be—total count or mean—because it would save me from spending more time writing scripts to get the data that I want to work with.

The issues discussed in this section show that Nyan 2.0XT does not fully provide features for addressing an important factor for an eye tracking experiment: Although Nyan 2.0XT is providing the necessary filtered data, it does not provide it in a clear, understandable, and usable format for the analyst. When designing software for an eye tracking experiment at the CMET lab we carefully consider the steps involved in our data analysis, and we design the software to make the process go as smoothly as possible. Raw data is created from the software that is running the experiment, the data is plugged into other tools that filter the data into fixations and saccades, some batch processing is run on all the data files generated by the experiment, and then the software exports all the data we want into a single data file that we can take and plug into R.

The data files will be exactly what we want because we define the data files at every step of the process, which makes them easy to understand and use.

Importing the data into R, sort of.

I actually never finished analyzing the data in R, but I did test it to make sure it was possible. The reason I never completed the total data import into R was because I did not want to take the extra steps to get the data that I wanted to work with. The data files from the AOI metrics tool will simply import into R. As I have previously mentioned many times, however, that data was not usable for me. For example, there were missing entries in the columns for the various weird passes that the AOI metrics tool generated, so I needed to tell R to fill in those missing entries with NA by setting the *fill* flag in the *read.table* function to true. Also, I had a total of 24 different data files for each WebScene, so I would either have to read in 24 different data files into R, write a script to combine them all into one data file, or just copy and past them by hand into one data file.

Conclusion

The purpose for recreating the banner ad study was to answer the questions of where is the data, what is the data, and how to get the data out of the Nyan 2.0XT environment to import into R. I did not spend too much time working with Nyan 2.0XT, but I do believe that I have spent enough time in order to answer these questions fairly thoroughly. The actual data that Nyan 2.0XT uses is hidden away somewhere within the Nyan 2.0XT environment, but it is relatively easy to export that data. What the data is depends on the research question being asked, and Nyan 2.0XT provides plenty of data to export such as raw gaze coordinate data, fixation data, saccade data, AOI data, and various other bits of data that can be exported out of Nyan 2.0XT. For many visual search tasks however, the most useful data would probably be the AOI metrics data, and it is nice that this data can be directly exported from Nyan 2.0XT rather than having to calculate it with the raw gaze data separately. Lastly, I believe most of the

exported data files can immediately be imported into R, but some post-export manipulation to the data files will need to take place for it to be in a format that is useable because the data is spread out across multiple data files and may include excessive useless information.

Nyan 2.0XT may be useful for simple user testing in industry, but for rigorous scientific studies it does not fully consider and provide adequately useful functions for the important considerations when running an eye tracking study. These considerations are again: (1) During data collection the gaze data will need to be constantly monitored to ensure accuracy is maintained, (2) after data collection the gaze data may need to be cleaned up by applying post-hoc error correction techniques, and (3) gaze data will need to be organized in a clear, understandable, and usable format for analysis.

I would not recommend Nyan 2.0XT for rigorous scientific studies because it fails to provide adequate features that thoroughly address the important considerations for running an experiment with an eye tracker. Once a participant recording starts, there is no way to place in predetermined pauses to allow for re-calibrations; this makes it hard to maintain accuracy while data is being collected because nothing can be done to fix error that may be seen within the live viewer. Nyan 2.0XT does not have a data import function, so there is no possible way to apply post-hoc error correction to the gaze data in order to get more accurate AOI metrics information. Finally, the data exported from Nyan 2.0XT is across multiple data files, includes excessive information, and some sort of post-export data file manipulation must be applied to create usable data files for analysis. The type of post-export data file manipulation that takes place will depend on what data files are needed. Getting a steady flow of raw data for the entire experiment will require some non-trivial script to combine multiple files into one. Getting AOI metrics information in a usable format will require combining of every single stimulus AOI metrics data file, removing empty cells, and possibly removing useless columns.

Nyan 2.0XT may be useful in industry for some simple user tests of software, but when it comes to rigorous scientific studies the best option for creating and running experiments may still be to write your own code.

References

- Burke, M. , Hornof, A., Nilsen, E. & Gorman, N. (2004). High-Cost Banner Blindness: Ads Increase Perceived Workload, Hinder Visual Search, and Are Forgotten. *Transactions on Computer-Human Interaction*, 12(4), 423-445.
- Buscher, G., Cutrell, E. & Morris, M. (2009). What Do You See When You're Surfing? Using Eye Tracking to Predict Salient Regions of Web Pages. *Proceedings of the 27th international conference on Human factors in computing systems*, 21-30.
- Hornof, A. J., & Halverson, T. (2002). Cleaning up systematic error in eye-tracking data by using required fixation locations. *Behavior Research Methods, Instruments, & Computers*, 34, 592–604.
- Vessey, K., & Hornof, A. J. (2009). A pilot study using Tobii Studio 1.3.22 to recreate the Burke et al. (2004) banner ad study. *Department of CIS Technical Report 2010-05, University of Oregon*. Seventeen pages.
- Zhang, Y., & Hornof, A. J. (2011). Mode-of-disparities error correction of eye tracking data. *Behavior Research Methods*, 43(3): 834-842.