Simon Lessing, simon@home.se
Ludvig Linge, ludvig.linge@home.se

IICap – A New Environment for Eye Tracking Data Analysis

Abstract

IICap is an environment for analysing eye tracking data on the Windows platform. Interviews with personnel were conducted to identify problems in the current process of eye tracking data analysis at the University of Lund eye tracking lab. A review of previous research on visualisation of eye tracking data and important eye tracking statistics was conducted. An IICap design proposal with suggested functionality that solves a majority of the problems found is presented.

1. Introduction

1.1 Background

The collection of eye tracking data is a science more than fifty years old. Early researchers such as Buswell (1935) and Yarbus (1967) performed experiments with simple photographs as stimuli. Today the method is used in the field of experimental psychology, cognitive science and human factors. Commercial applications include quantitative tests of website usability and advertisement observations. The eye tracking hardware has constantly been improved, from Yarbus's search coil technique to today's advanced equipment; infrared cameras connected to computers.

Early eye tracking data analysis was often qualitative, due to the simple fact that computers were not widely available. Some experimenters even collected data based strictly on observations (Rayner 1978). Though computers are extensively used today, there are still elements of qualitative analysis left when analysing eye tracking data, as some eye tracking experiments are recorded to video only.

Even when collecting data with an advanced, computer based system it is still a cumbersome and tedious process to analyse it. Many researchers in the field spend up to 90% of their valuable time analysing data (Interviews with Holmqvist 2001. See also Aaltonen 1999). The process is often manual. Significant amounts of time could be saved if the analyses were automated.

In the field of eye tracking data analysis Itoh et al (1998) and Velichkovsky et al (1996) have presented ways of visualising eye tracking data in new manners. Attentional landscapes and state diagrams simplify the presentation of data, especially to people not acquainted with the eye tracking terminology. These and other representations will become increasingly important as the practical applications of eye tracking expand to commercial areas. One of the most important areas of applied eye tracking is HCI testing, where the people expected to participate in the interpretation of the results will be designers and programmers rather than researchers.

1.2 Main Objective

The main objective of this thesis is to present a design proposal for a new eye tracking analysis software to be used at the eye tracking lab at the Lund University Cognitive Science department - LUCS. Interviews with personnel were conducted to identify problems in the current analysis process, the results are to be found in chapter 2. A review of previous research on visualisation of eye tracking data and important eye tracking statistics was conducted. These results and their relevance to this thesis are presented in chapter 3. Chapter 2 and 3 serve as a base for the development of the new software, called IICap. The IICap design proposal in chapter 4 includes descriptions of the workflow and suggested functionality with corresponding screen layouts. The suggested functionality will include new ways of visualising eye tracking data. Chapter 5 includes a discussion on the application developed and suggestions on further research.

2 The Current Process at LUCS

2.1 Staff at LUCS Eye Tracking Lab

To obtain an overview of the problems that face the personnel at the LUCS eye tracking lab, two people who have worked frequently in the lab were interviewed during the late fall 2001:

- **Kenneth Holmquist** – Assosciate Professor at LUCS and responsible for the eye tracking lab. Kenneth has been working in the lab for 5 years.

- **Maria Barthelson** – Currently writing her master thesis at LUCS. Maria is studying the behaviour of subjects visiting news sites on the Internet. She has been working with the eye tracker and analysis software at LUCS frequently but only during a few months. Maria was interviewed since she is a new user in the lab and therefore has a greater ability to question current routines and habits.

The material presented in this chapter is based on interviews with the personnel mentioned above if no other reference is given.

2.2 Static Data Capture and Analysis

There are two main types of data analysis found at the LUCS eye tracking lab. The first type of data analysis is performed on data collected by a table-mounted eye tracker device continuously tracking the subject’s gaze. In this case, raw eye tracking data is generated as a sequence of time stamped gaze coordinates and pupil diameters, sampled at 50Hz. Since the eye tracker is mounted in a fixed position and extensive head movements are not allowed, the coordinates collected are all relative to the stimuli and can easily be used to superimpose the gaze movements and fixations on top of the stimuli.

2.3 Non-Static Data Capture and Analysis

The second type of data analysis is performed on data collected to video. In this case, a head-mounted eye tracker collects the eye gaze position of the subject and the coordinates are used to continuously superimpose a white dot over the stimulus as recorded by a video camera mounted on the subject’s head. Since the coordinates collected by the eye tracker are relative to the position of the subject’s head and not relative to the actual stimuli as in the first case, this method requires not only analysis of the coordinates generated by the eye tracker but also analysis of the video recording of the stimulus.

This thesis will focus on the analysis of data collected by a static, table-mounted eye tracking device. The process of data analysis involving video data will not be covered.

2.4 Equipment

The main equipment used for static eye tracking at the LUCS eye tracking lab is:

- Table-mounted eye tracking device (iView 50 Hz remote set) from SMI, GmbH.
- iView eye tracking data analysis software from SMI, GmbH.
- An AMD Athlon XP 1700+ PC running Windows 98. Used for stimuli generation.
- An Intel Pentium PC running Windows 95. Used for data analysis.

2.5 Data Collection Issues

2.5.1 Calibration

At the start of every experiment, LUCS-personnel calibrate the eye tracking hardware by letting the subject look at nine calibration points on the screen. In many cases this procedure is sufficient to produce valid data, but some of the subjects generate data that has poor calibration. This can be due to contact lenses, glasses or individual differences in the physiology of the eye. For instance, if the subject starts to squint more than s/he did during calibration, data will also be distorted.

Even if the personnel notice poor calibration during the experiments, they cannot interrupt and restart the testing since this would yield ecologically invalid data. The result is that some of the collected data is distorted. Often the distortion is obvious to LUCS-personnel when analysing the data. The data can be indicating that a subject has been reading a text, but the actual fixation points may for instance be 1-2 degrees below the lines of text. Although the subject was obviously reading the text, there is not much the personnel can do about it, since data cannot be manipulated in retrospect with current software.

2.5.2 HCI Experiments

Another problem during data collection occurs in the field of HCI upon usability testing of web sites and computer software. It has been shown that when eye tracking data is used to support verbal retrospection the quality of usability testing results are enhanced (Hansen 1991).

However, interface evaluation and usability testing are expensive, time-intensive exercises, often done with poorly documented standards and objectives. They are frequently qualitative, with poor reliability and sensitivity. An improved tool for rapid and effective evaluation of graphical user interfaces is clearly needed (Goldberg & Kotval 1999).

At LUCS the personnel unfortunately have to restrict their HCI experiments in different ways. Since there is no specific support for HCI testing in current software, there is no difference between computer software and other stimuli, such as photographs. Every time the user generates an event such as a click on a link in a browser the personnel manually have to take notice of this and send a sync signal to the eye tracker by pressing a keyboard button twice to start a new set of data. Furthermore, there is no automatic logging of which events the user has trigged and when the events were trigged.

2.5.3 Web Page HCI Studies

The most urgent problem to solve regarding HCI testing at LUCS is the scrolling of web pages in a web browser. Whenever a subject scrolls a page the eye tracking data collected is distorted, since the absolute coordinates are lost. The subject could be reading a single line of horizontal text while s/he scrolls down the page, resulting in incorrect data that indicates gaze movements going in an up/right direction on screen. These severe limitations have forced the personnel to exclude scrolling in some tests involving usability of websites. This has had a negative impact on the ecological validity of the tests.

Interpreting the data is difficult, if test software contains scrollable and multiple windows, which can be freely positioned on the screen (Aaltonen 1999).

In addition to scrolling compensation, personnel wish to capture the URL of visited pages as well as mouse and keyboard data of subjects, since this would be useful when performing usability tests.

2.6 Organizing Data

2.6.1 Sets

Data generated from an experiment is stored in .DAT text files containing each subject’s gaze coordinates and pupil diameter sampled and time stamped at 20 ms interval. A data file can include many different sets. A set is defined as data within a data file, limited by a start and stop time. Typically, sets are used to mark a change of stimulus shown to a subject.

2.6.2 Keeping Track of Subjects and Stimuli

The first step in data analysis is often to organize the different data files and the sets included in these. A set contains no other information than start/stop time; there is no reference to subject or stimuli. Data files can be named, but the sets are only numbered. Data files and their sets is the only way to discriminate between different subjects and different stimuli. Often, one data file contains data for one subject and many stimuli. For instance, a data file called TP7.DAT could include 15 sets, each representing data from 15 stimuli showed to test person 7. As these sets cannot be named, the personnel have to keep track of the sets and which stimulus they represent manually. In the iView, stimuli are chosen independently of data files, which add another variable to keep track of. All this is especially time consuming when stimuli have been shown in a randomised order. Then, a set number can correspond to any arbitrary stimulus. Currently, personnel enter data in Microsoft Excel and print it to keep track of sets, subjects, stimuli and their relations.

2.6.3 Area Of Interest Definitions

Common practise at LUCS is to define one or more Area Of Interest (AOI) for each stimulus. An AOI is a spatial region of a stimulus that is given a descriptive name and used in statistical purposes. AOIs are often used to study the distribution of fixations on a stimulus. The transitions of gaze from one AOI to another might also be of interest.

In iView an AOI is defined as either a rectangle or a circle. An Object/Time view based on the AOIs defined is available. This view shows where and when fixations have fallen and the total amount of gaze time spent in each AOI. Number of saccades within each AOI is also showed. There are no statistics available that show the number of transitions between the AOIs. As these statistics are of interest in many experiments, LUCS-personnel have to print the object/time view and count the number of transitions between AOIs manually.

Furthermore, LUCS-personnel wish to define AOIs of any shape, not only rectangles and circles.

The AOIs defined in iView are stored in files independent of stimuli; this adds another variable to keep track of manually.

2.6.4 Lack of Temporal AOI Definitions

When working with her dissertation Holsanova (2001) had to correlate verbal recordings with spatial eye tracking data. In iView there is no way of defining a temporal Area Of Interest. A temporal AOI would be defined by a start and stop time and applied to a subject’s observation of a certain stimulus. Due to the lack of temporal AOIs, Holsanova had to manually keep track of a subject’s verbal keywords and the times they were said.

2.7 Numerical Data Analysis

2.7.1 Identifying Subject Behaviours

Identifying subject behaviours is often desirable. One such behaviour can be reading, where one is looking for a specified number of saccades to the right followed by a return sweep of saccades to the left. There is no automated way to identify patterns of saccades, LUCS-personnel therefore have to export eye tracking data and write their own algorithms in Microsoft Excel. This is often done from scratch every time data is needed, and algorithms are rarely saved.

2.7.2 Numerical Measures

The following numerical measures are characteristic of eye tracking, but not available in iView. They have to be calculated manually in Microsoft Excel every time they are needed:

Average saccade length (in pixels)
Average saccade duration (in ms)
Average saccade velocity (pixels/ms)
Average number of saccades per second
Average fixation duration (in ms)
Fixation/saccade time ratio
Average pupil diameter

The above statistics could be limited by spatial- or temporal AOI definitions or be calculated from the entire stimulus and observation time interval.

Statistics showing numbers of saccades/fixations in a single set are available in iView. However, sometimes statistics based on several subjects are required. This is particularly common when looking at averages such as average saccade length and average fixation length.

2.7.3 Missing Data

It is very rare that 100% of data is captured during experiments. This is not shown in iView, which sometimes can be confusing. Saccades can be interpreted as being 10 seconds long, resulting in erroneous statistics. This impacts the validity of the majority of statistical analysis that can be calculated.

2.8 Visualisation

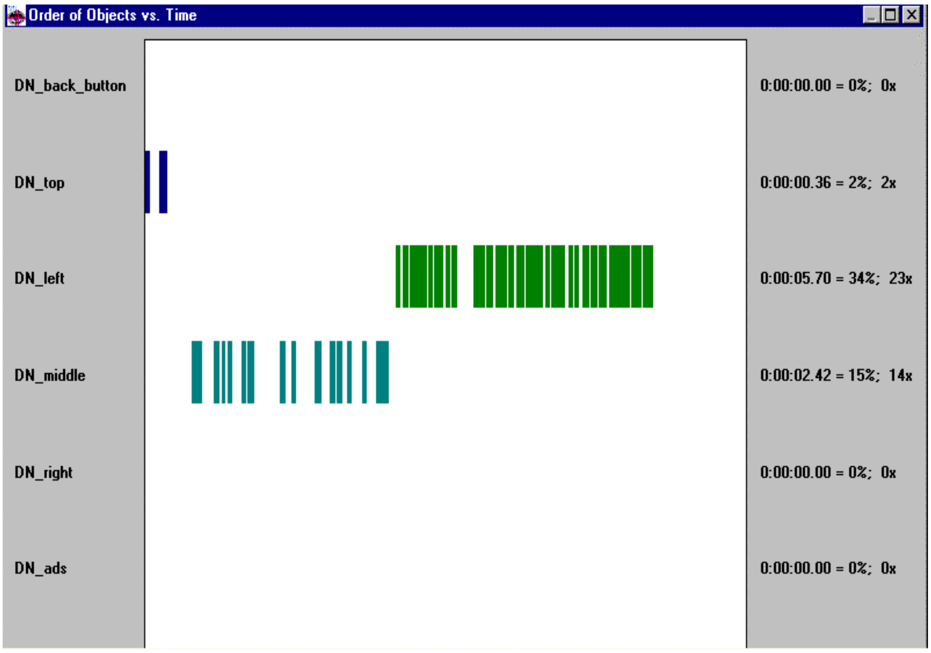
2.8.1 The Need for New Visualizations

In iView eye movements are visualised as scan paths. Scan paths are defined by a saccade-fixate-saccade sequence on a display (Goldberg & Kotval, 1998). A bitmap picture is imported and scan paths are superimposed over the stimuli.

Scan paths provides a good way of visualization for short recordings where the number of fixations of an experiment do not exceed 10-20. However, in a several-minute recording the number of fixations increases and the screen become cluttered with scan path visualizations. Also, when showing the data to people not familiar with eye tracking terms it is desirable to avoid having to explain the mechanisms of saccades and fixations.

2.8.2 AOI Transitions

In iView, the object/time view is the only diagram showing transitions between AOIs. Though it shows time, the AOIs are not presented in a spatially preserved way.



DN_back_button	0:00:00.00 = 0%; 0x
DN_top	0:00:00.36 = 2%; 2x
DN_left	0:00:05.70 = 34%; 23x
DN_middle	0:00:02.42 = 15%; 14x
DN_right	0:00:00.00 = 0%; 0x
DN_ads	0:00:00.00 = 0%; 0x

Object vs. Time from iView.

When personnel want to present their eye tracking data visually they have to export relevant data to Microsoft Excel and use graphs available there. In Excel there are no graphs specifically designed for visualising eye tracking data.

2.8.3 Diagram Comparisons

In current software, the diagrams available can only be viewed one at a time, which means that side by side comparisons between different diagrams is impossible. The only way to overcome this problem is to print diagrams and compare them on paper. The same situation occurs when the user wants to view diagrams from two or more subjects side by side, with the addition that s/he have to load a new data file for every subject.

2.8.4 Real-time Playback

Current software offers no way of playing back eye tracking data in real-time. The only way to see when a specific saccade has taken place is to select a time interval and see if the saccade is within this interval. The interval can then be made smaller until it covers only the saccade of interest, and the user can get some understanding of the temporal aspect of the saccades by moving the selection forward in time.

3. Previous Work of Relevance

Quite a lot of work has been done trying to find new ways of interpreting, analysing and visualizing eye tracking data. However, inventions in these fields often emerge as side effects of eye tracking based experiments where researchers, due to the lack of existing methods, have to come up with their own solutions. Very few articles focus directly on the problems of eye-tracking data analysis. Listed in this chapter are articles that have given aid and inspiration to the process of designing the IICap software.

3.1 Data Collection

3.1.1 Web Page Analysis

As mentioned in section 2.5.2 and 2.5.3, data collection problems occur when the stimuli software contains scrollable and multiple windows.

Greg Edwards, former researcher at the Stanford university and now founder of Eyetools Inc, has been rewarded a patent (US pat no 6,106,119) which states a method of correlating test software events with eye tracking data. In his invention, Edwards suggests a number of methods for the correlation. One of the methods presented is based upon a software that is fed in real-time with eye tracker data and has a connection to the operating system hosting the stimuli software. The correlation software is able to fetch event information such as scrolling, mouse clicks, hyperlink activations and screen resizes from the operating system. Every event occurring is either correlated with live eye tracking data or is given a time stamp and then stored in a log file for later correlation and/or presentation in an eye tracking data analysis software. A snapshot of the window is also stored together with a time stamp upon every event generated.

The correlation software then uses the log files and snapshots to build a continuous picture of the stimuli software with the eye movement superimposed over it.

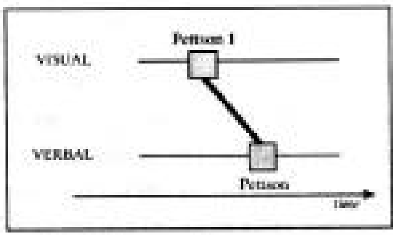
It is clear that certain parts of the techniques described by Edwards would help solve some of the problems described in section 2.1.3. However, Edwards does not present any details of the technique used to fetch events from the operating system. Nor does Edwards discuss the performance loss caused by his method when taking multiple snapshots at a high rate that would be necessary upon fast scrolling of any window.

3.2 Data Organization

3.2.1 Temporal AOIs

Faced with the problems of visualizing the correlation between temporal data such as verbal communication and spatial data such as eye gaze positions, Holsanova (2001) invented her own timeline visualization.

The suggested timeline consists of two different ‘stream’ time axes - the visual and the verbal. Blocks, visualized as rectangles on each stream represent visual and verbal focus. Visual focus typically is a set of fixations occurring in a certain AOI at a certain time. Verbal focus is a single word or a couple of words in a sentence pronounced by the subject.



By connecting the verbal- and visual blocks with lines, Holsanova was able to visualize which verbal object described which visual object. With this visualization, it was easily seen that most often, the verbal focus corresponding to a visual focus was delayed by 2-4 seconds.

Further on, there are different kinds of mappings between the visual and the verbal objects. The most basic is the one to one mapping, however, there can also be 1-n mappings where a single verbal object is mapped to multiple visual objects. These mappings are easily visualized in the timeline with lines connecting multiple temporal objects to each other.

The concepts of temporal objects and timeline visualization as suggested by Holsanova are interesting and could be used as a ground from which to implement temporal objects and temporal-spatial visualization in the design of the IICap software.

3.3 Statistics

Goldberg & Kotval (1999) present an introduction to a framework for eye tracking data analysis, especially suitable for user interface evaluation. Goldberg & Kotval clearly emphasize the need of quantitative methods for rapid and more effective analysis. In their work, Goldberg & Kotval present suggested measures and algorithms in the light of an experiment presenting users with good and poor user interfaces. Conclusions are drawn that more sophisticated measures than AOI transitions are needed in order to approach some of the usability issues regarding screen design. Considering the complex nature of scan paths, measures such as scan path duration and scan path length are becoming important.

3.3.1 Scan Path Length

Goldberg & Kotval suggest that the scan path length, i.e. the sum of the consecutive saccade lengths between fixations on interface cues, is a measure of efficiency in visual search. An optimal visual search indicates a minimum in scan path length.

3.3.2 Scan Path Duration

According to Goldberg & Kotval (1999), scan path durations relate more to cognitive load and complexity of information processing than to visual search efficiency. More relative time is spent in fixations than in saccades.

3.3.3 Number of Fixations

Goldberg & Kotval's (1999) results emphasize that the number of fixations in an AOI indicates the efficiency in finding a desired object among others. Significant differences in number of fixations were found when comparing the good user interface with the poor one.

Van Orden et al (1999) conclude that fixation frequency and blink frequency are the best indicators of cognitive workload.

3.3.4 Maximum and Average Fixation Duration

Goldberg & Kotval (1999) suggests that maximum fixation duration and average fixation duration in an AOI are good measurements of how informative the components of a user interface are in a certain area.

Van Orden et al (1999) concludes that several visual search studies have shown that more effortful search, as indicated by poorer performance accuracy and longer search times, is associated with increasing numbers of fixations.

3.3.5 Saccadic Extent

Van Orden et al (1999) notice a significant decline in saccadic extent during high cognitive load.

In a study of eye movement behaviour while driving a car, Cohen (1978) concludes that perceptual narrowing emerges from cognitive workload such as stress or mental fatigue. Further on, drivers influenced by alcohol as well as inexperienced drivers suffered from a perceptual narrowing compared to normal drivers.

Another use for the saccadic extent measure is when examining the meaningfulness of user interface cues. A well-designed interface should provide sufficient cues to direct the user's scanning to desired targets very rapidly, with few interim fixations. (Goldberg & Kotval 1999)

3.3.6 Pupil Diameter

Pupil diameter, while also affected by changes in illumination, stimulus, and accommodative behaviour, has been shown to generally increase with higher cognitive processing levels. A number of reports show that pupil diameter correlates with cognitive workload irrespectively of the visual demands imposed by a task (Van Orden et al, 1999)

Partala et al (2000) studied the pupillary responses to emotionally provocative stimuli. They showed that the pupil size differed significantly between exposure of negative and exposure of positive emotional stimuli.

3.3.7 Fixation / Saccade Ratio

In interface evaluation, this measure can be of importance when examining the processing time vs. searching time for components and interface cues (Goldberg & Kotval 1999).

3.3.8 Direction of Saccades

Backtracking is introduced as a saccade that deviates more than 90 degrees from the most previous one. Backtracking indicates that the subject probably was mislead by the interface and had to change direction completely. Further on, such 90-degree saccade deviations can also emerge from comparison tasks (Goldberg & Kotval 1999).

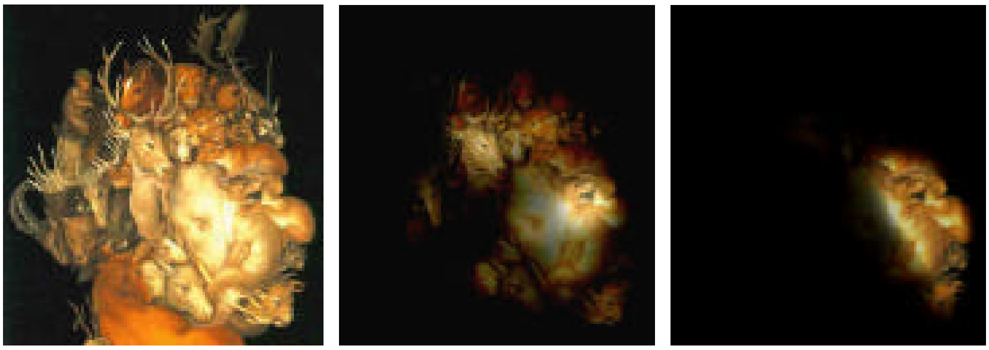
3.4 Visualizations

3.4.1 Attentional Landscapes

Velichkovsky & Hansen (1996) introduce “Attentional Landscapes” in order to reconstruct the subjective interpretation of an ambiguous picture on the basis of spatial distribution of eye fixations. They are suggesting a filter function that changes intensity from the focus out to the peripheral vision. By applying this filter to the stimulus viewed one can improve the understanding of the subjective interpretation of the stimulus.

Velichkovsky & Hansen (1996) suggest a use of their attentional landscapes for medical purposes. By studying how skilled doctors examine ambiguous medical images to detect deceases and syndromes, it might be possible to detect which parts of the images that are given extra attention and which are not. From these studies, attentional landscapes are created giving less skilled doctors aid in the image examination process.

These Attentional landscapes fill the need of an easy to understand form of visualization described in section 2.4.1.

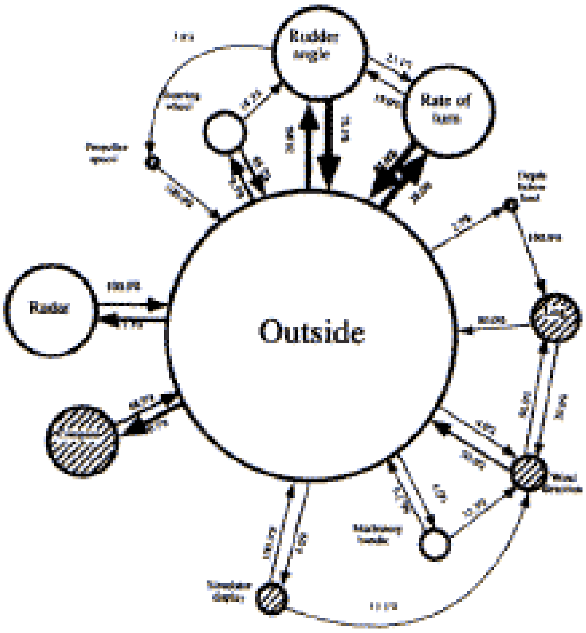


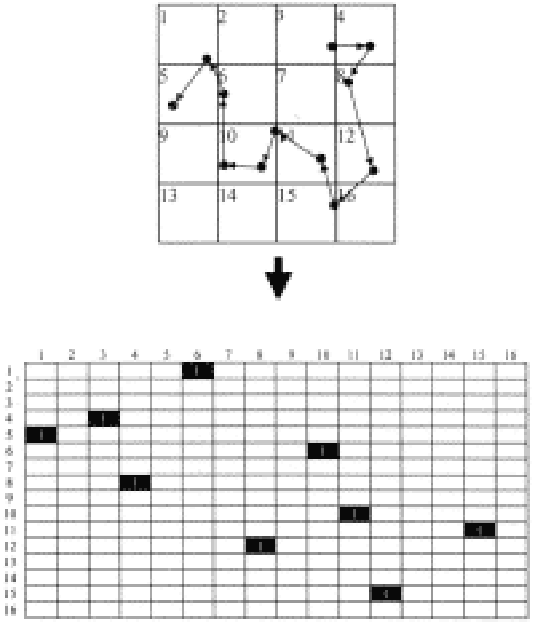
Pictures from Velichkovsky & Hansen (1996) showing intensity falloff based on eyetracking data.

3.4.2 State Diagram and Transition Matrix

Itoh, Hansen and Nielsen (1998) present a method of visualising eye movements among areas of interests as a state transition network. Each state (AOI) is visualised as a circle with the radius corresponding to the accumulated fixation time for the particular AOI. Arrowed arcs connecting one circle with another represent the possible transitions from the state. The size of the arcs shows the frequency of transition with the relative percentage of outgoing eye-movements from one source to another. Two different circle fill patterns are used to separate “situation awareness“ states from “state monitoring” states.

The state diagram representation provides a way of visualizing and simulating the viewing of a scene. The simulation begins in one state and then the following simulated eye movement (state transition) is depending of the probability of jump to another state, as indicated by the arrow arcs.

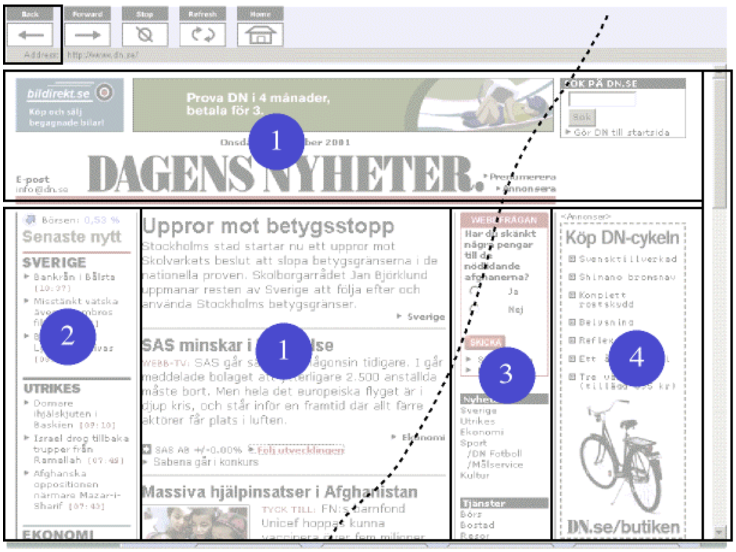




Instead of using a state transition diagram, Goldberg and Kotval (1999) divide the scene into a number of small rectangular cells. Then, by checking saccade transitions from one cell to another, they develop a transition matrix showing the transitions from one cell to another. In this way, every part of the visual scene is considered to be an area of interest, and thus it is possible to compute the transition density for the entire scene.

3.4.3 First Occurrences AOI Sequence Diagram

In her master's thesis, Maria Barthelson (2002) studies subjects' behaviour when reading on line news. Barthelson suggests a way of visualizing the first occurrences of a subject's fixations of the web page AOIs. Her results might be of interest to web designers when deciding how to structure the layout of newspapers on the web.



The first occurrences AOI sequence diagram was constructed in the following way:

An Object/Time diagram in iView was used to extract the first occurrences of fixations for each subject. Each AOI got an ordering number for each subject. A specific ordering number was only added once for each subject and AOI.

The ordering numbers for each AOI were finally summed and a mean value was calculated for each AOI. The first occurrences AOIs sequence diagram was created by sorting the mean values created.

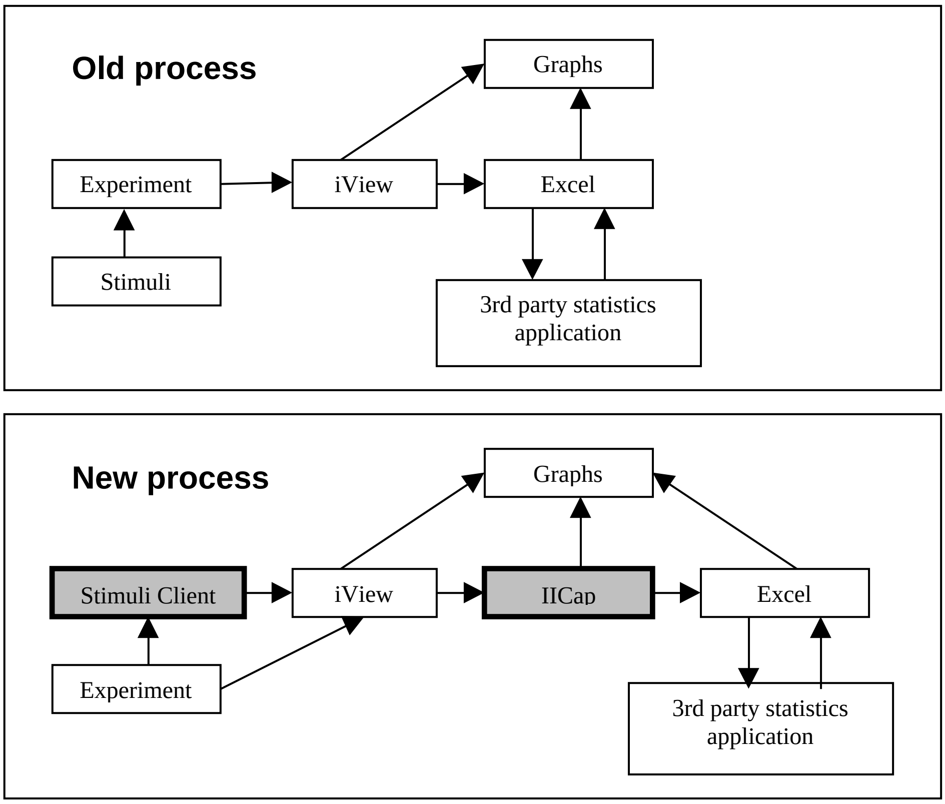
4. The Application

4.1 Targeted Users

The IICap software will primarily be used by personnel at the LUCS eye tracking lab. Users of the software are assumed to be familiar with eye tracking vocabulary and have basic knowledge of occomular research. Furthermore, users are expected to have intermediate experience of computers, and to be familiar with the Microsoft Windows operating system.

4.2 Scope of the Application

The application will be used in conjunction with existing software, as the application will not include support for eye tracking hardware and data collection.



4.3 The Stimuli Client

IICap consists of two modules; the stimuli client and the main program. The stimuli client will run on a separate computer, communicating via serial port with the computer where the eyetracking hardware is installed to synchronise. The client can be programmed to show either static stimuli such as illustrations and photographs or dynamic stimuli like web pages or video. In many aspects this client is similar to PsyScope (1993) but unlike PsyScope it will seamlessly integrate with data analysis software.

Stimuli can be fully randomised or shown in a pattern specified by the user. The client keeps track of subject, stimuli and time. This means that randomising stimuli no longer will impact analysing time since the sorting of data will be automatic.

The stimuli client includes a web browser based on a Microsoft Internet Explorer plug-in. A log file is created to keep track of which links the subject has clicked and how the subject has scrolled. Users can scroll web pages and the stimuli client generates screenshots that have a larger y-height in pixels than the height resolution of the screen.

The scrolling compensation, inspired by the technique described in chapter 3.1.1, continuously samples the scrollbar position of the browser window to get a vertical and horizontal offset in pixels. The sample rate is at least 50Hz to sync with the eye tracking hardware. Scroll positions are time stamped and saved into a log file. After the experiment session, a merging algorithm applies the offset values from the log file to the raw data file from the eye tracker. The merged output is a new raw data file which is which is then loaded into iView. The iView fixation algorithm converts it to fixations that is saved as .FIX files and then imported into and further analysed in IICap.

The stimuli client was developed as a solution to the problems presented in section 2.5.2, 2.5.3 and 2.6.2.

4.4 Features of the Main Program

This section aims to give a brief overview of the features implemented in IICap.

Data Modification

- Data that has been collected with insufficient calibration can be corrected in retrospect by manipulating a grid of vector control points.

Projects

- The workflow in IICap is based on projects. The project architecture was chosen in view of the fact that there was, as presented in section 2.6.2, a substantial need to organize data of subjects and stimuli to avoid the process of keeping track of the information manually
- A project file includes multiple subjects' specified by names, ages and genders.
- A project file includes multiple stimuli specified by names and corresponding bitmap image files.
- Actual eye tracking data is represented as observations, containing information on which subject have observed which stimuli.
- IICap allows import of eye tracking data from SMI's iView as well as Fixplot by ASL. These two software solutions ship with the biggest selling hardware and are therefore most widely spread.
- The import function conserves the set information in the .FIX files imported. The sets from multiple subjects are incorporated into the project architecture of IICap as observations.

AOIs

- IICap allows creation of objects of three forms; circles, rectangles and polygons. AOIs are editable in retrospect.
- AOIs are saved in the project file.
- Users are able to create temporal objects, defined as a time interval in a specified observation.

Chart Analysis

- AOI vs. Time analysis gives an overview of the distribution of saccades in objects in the temporal domain. The functionality is similar to that of iView, with the addition of temporal objects and the ability to see more data related to object and time.
- Transition diagrams give an overview of the distribution of saccades in AOIs in the spatial domain. Transition data shows how many transitional saccades have fallen between AOIs. The spatial layout of each AOI is preserved, as AOIs and transition data are superimposed over the stimuli. Data from multiple subjects can be averaged or summarized to present a single diagram.
- Data from multiple subjects can be averaged to present a single diagram showing in which order fixations have fallen onto a stimuli. The AOI Sequence diagram shows a ranking of the AOIs as discussed in section 3.4.3.
- Attentional Landscapes, as introduced in section 3.4.1, give an overview of where saccades have fallen without the need to present scan paths. Data from multiple subjects can be averaged or summarized to present a single diagram.

- It is possible to play back eye tracking data in real time. The controls of this function are similar to those of a standard VCR with play, stop, rew, ff.

Statistics
The following statistics are available in IICap:

- Average saccade length (in pixels) as specified in 3.3.1
- Average saccade duration (in ms) to fill the need in specified in 3.3.2
- Number of saccades/fixations to fill the need in specified in 3.3.3
- Maximum and average fixation duration to fill the need in specified in 3.3.4
- Average pupil diameter to fill the need in specified in 3.3.6
- Fixation/saccade ratio to fill the need in specified in 3.3.7

Furthermore, more basic eye tracking statistic measures are available:

- Average saccade velocity (pixels/ms)
- Average number of saccades per second
- Average fixation duration (in ms)

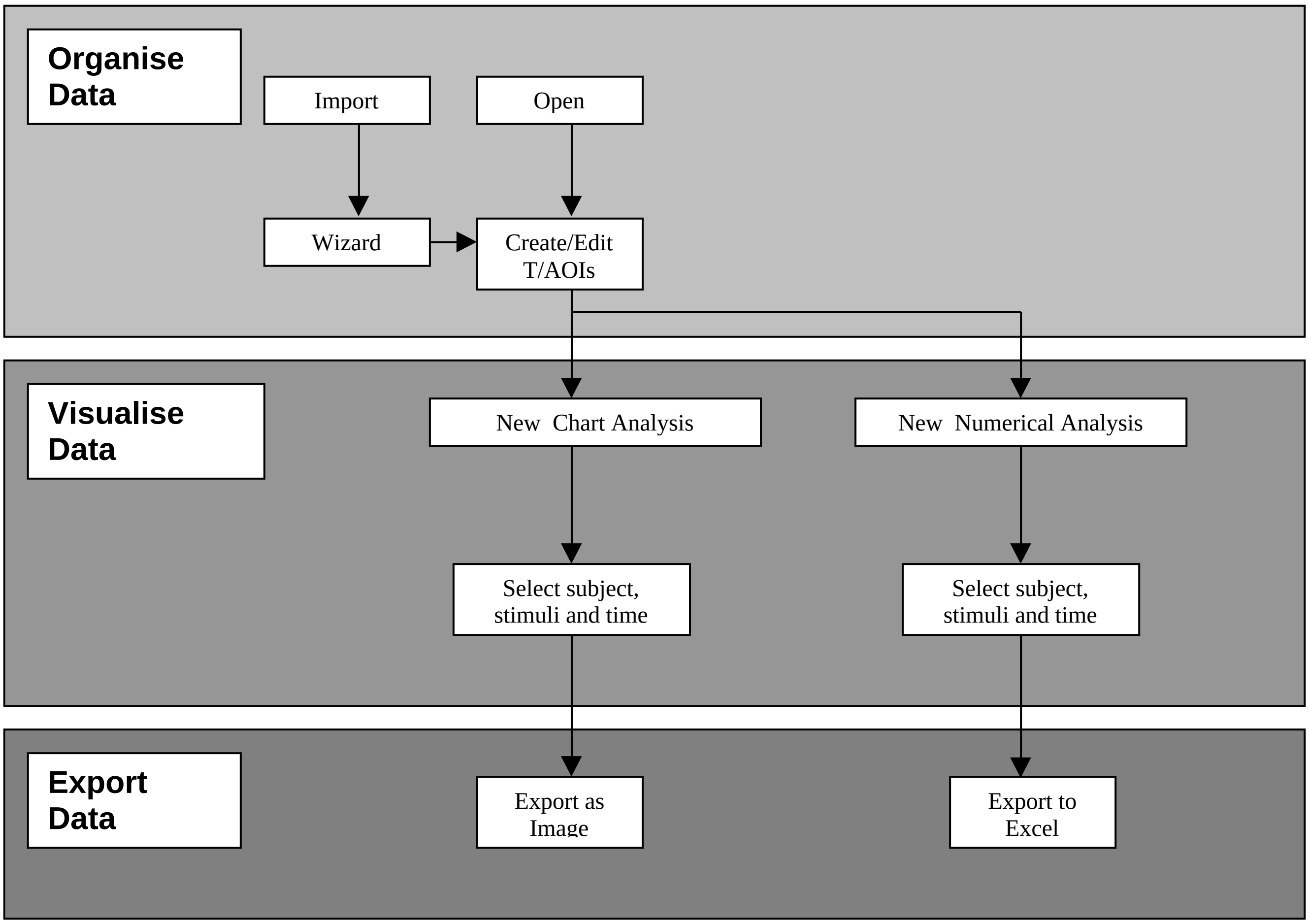
Not all statistics are applicable in all analysis. Selection of data that render statistics is based on one or more of subject, stimuli, time and AOI.

Export

- Export of eye tracking statistics to Microsoft Excel.
- Export of diagrams and graphs as either JPEG or BMP files.

4.5 Workflow

The workflow in IICap is divided into three main tasks; Organise data, Visualise Data and Export Data.



Organise data task

- *Import data/Open project* The user either opens an existing project or creates a new project by importing existing iView data. If import is chosen, the user goes through the process of entering set information.
- *Create/Edit AOIs* The user has the option of defining multiple spatial- or temporal AOIs for each stimulus defined.

Visualise data task

- *Create new analysis* Analyses show either numerical data or diagrams. Several analyses are available, each showing a different diagram, graph or table.
- *Select subject, stimuli and time* When a new analysis is created, the user selects one or multiple subjects and stimuli, and a time interval. This selection is then visualised in the analysis window.

Export data task

- *Save bitmap image.* A bitmap image snapshot of the active chart analysis is saved to disk either in BMP or JPEG format.
- *Save Excel file* An .XLS file of the active numerical analysis is saved to disk to allow further statistic analysis and visualisation in Microsoft Excel.

4.6 Interaction

IICap is developed in a Windows environment and users are assumed to have intermediate knowledge of the Windows operating system. The interface and method of interaction therefore resembles the one of Windows as much as possible. IICap uses standard Windows components; windows, icons, and menus.

The main interaction technique in IICap is Direct Manipulation. The term Direct Manipulation was first presented by Shneiderman (1983) and is characterized by:

1. Continuous representation of the objects and actions of interest
2. Physical actions or presses of labeled buttons instead of complex syntax
3. Rapid incremental reversible operations whose effect on the object of interest is immediately visible.

Systems that use Direct Manipulation are easy to learn and reduce the need for error messages. Furthermore users experience less anxiety because the system is comprehensible and because actions can be reversed more easily (Shneiderman 1998).

The same Direct Manipulation techniques that are available in Windows are used in many parts of IICap:

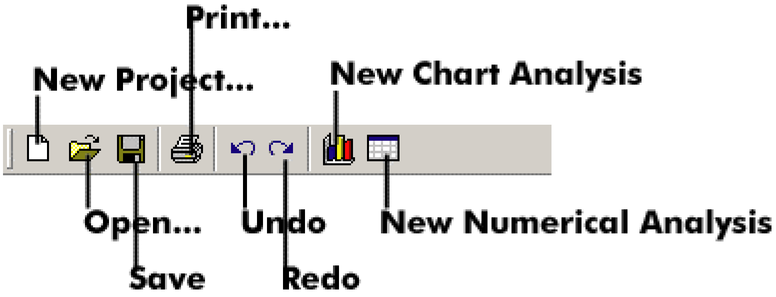
- Subjects and stimuli are continuously visible in the project tree. Dragging and dropping subjects and stimuli into a newly created analysis give the user immediate feedback as it creates a diagram or graph.
- Manipulating windows ease the organization of multiple analyses.

- The mouse pointer is used to make direct selections in diagrams and graphs.
- The mouse pointer changes shape depending on what is under it, so that users can see what actions are possible.

Analyses are presented in windows that can be moved around, resized, and tiled like any Windows window. Users are able to create multiple analyses, each shown in a separate window to allow comparisons between different diagrams or graphs, solving the problem presented in section 2.8.3. Analyses are programmed as independent modules; thus introducing the possibility of third party plug-in development that visualize data in new ways.

Pull down menus are used to lower the users' cognitive load by presenting commands in a hierarchy that can be remembered in chunks. (POMS 1991)

Icons are used in a toolbar, which offers short cut access to functions that are frequently used. Aimed at expert users, short cut keys are also available for a selection of commands. In the pull down menus, icon shortcuts will be shown to the left of the commands and keyboard short cuts to the right. This assists learning of icons and keyboard short cuts (Dix 1995).



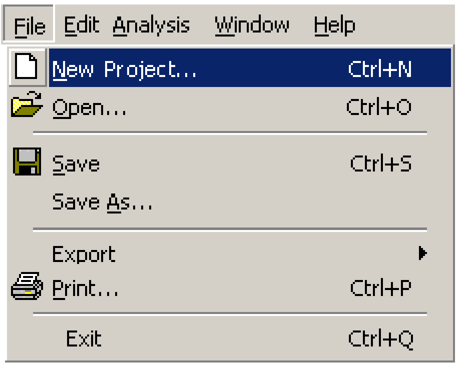
Floating toolboxes are used to set properties of analysis windows. The content of the toolboxes are dependent on what kind of analysis view is active, as different settings are offered for each analysis type. Floating toolboxes are often used in design programs like Adobe Illustrator and Macromedia Freehand. They offer a good space to put commands that are not always available; as the pull down menus should not contain commands that appear and disappear depending on what window is selected.

Toolboxes can be closed when not needed; thereby providing an effective way of saving screen space (Shneiderman 1998).

All data associated with a project is stored in a project hierarchy. Hierarchies are common in the windows environment and most users should be comfortable with navigating a tree structure like the one found in Windows Explorer. The project hierarchy will help users to keep track of subjects, stimuli and AOIs as the whole project is saved to a single file, representing an experiment setup.

With a tree structure, the user can form a mental model of the structure and the relationships among the menus (Schneiderman 1998).

4.7 Menus



The menus in IICap are the standard Windows menus (File, Edit, Window) plus an IICap specific menuitem called Analysis. In Analysis, the user finds commands relating to analysis views, most importantly the New Analysis command. Icons and keyboard short cuts are showed in the menus to aid learning.

The matrix below shows the menu structure IICap.

Menu-Submenu	**Menu Item**	**Description**
File	New Project…	Opens the Import Wizard, where the user imports eye tracking data to create a new project.
File	Open…	Opens an existing project.
File	Save	Saves the current project to disc
File	Save As…	Saves the current project to disc, with a new name
File	Print…	Prints a chart or table from the currently selected analysis.
File	Exit	Exits the program
File – Export	Export Chart Analysis as Image	Saves the currently selected chart analysis as a bitmap image, either in JPEG or BMP format.
File – Export	Export Numerical Analysis to Excel	Saves the currently selected numerical analysis as an Excel file.

Edit	Undo	Undoes the latest action
	Redo	Redoes the latest undo
	Define AOIs…	Opens the Define AOIs dialog
	Define Temporal AOIs.	Opens the Define Temporal AOIs dialog
	Modify Data	Opens the Data Modification window

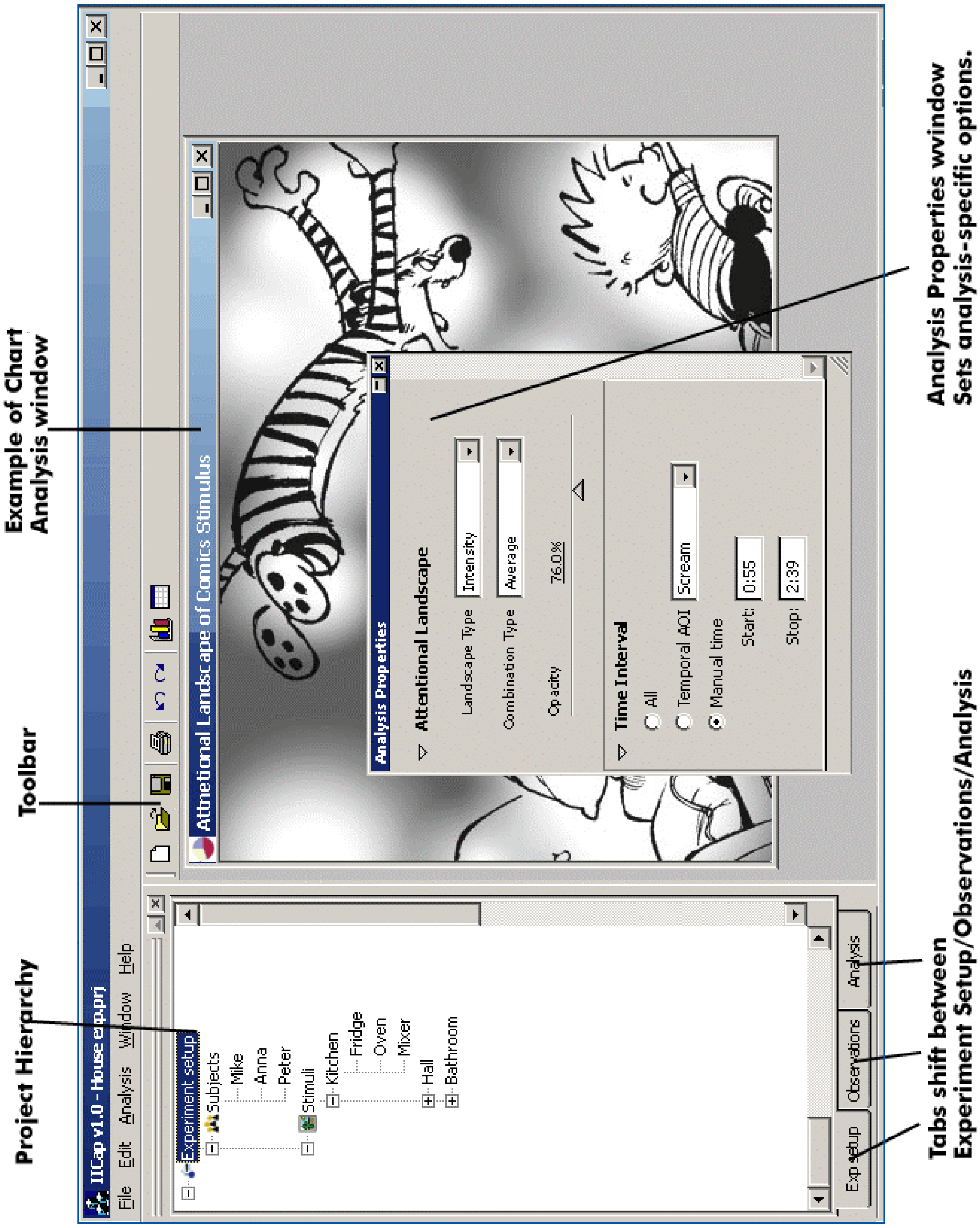
Analysis – New	Chart…	Opens a dialog where the user can choose to create a new chart analysis window.
Analysis – New	Numerical…	Opens a dialog where the user can choose to create a new numerical analysis window.
Analysis	Subjects…	Opens a dialog box where subjects can be chosen to be included in or deleted from the currently selected analysis.

Window	Cascade	Arranges the analysis windows on top of each other.
Window	Tile	Arranges the analysis windows tiled with each other

4.8 Screens

4.8.1 Main Screen

The interface in IICap consists of two main areas; the project hierarchy and the desktop area that holds the analyses. The project hierarchy is found on the left side of the screen and contains three tabs that hold the complete project; Experiment Setup, Observations and Analysis.



Landscape Type: Intensity

Combination Type: Average

Opacity: 76.0%

○ All

○ Temporal AOI: Scream

● Manual time

Start: 0:55

Stop: 2:39

Project Hierarchy

Toolbar

Example of Chart Analysis window

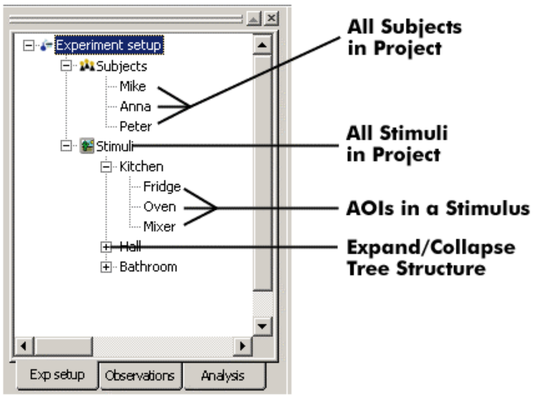
Analysis Properties wwindow
Sets analysis-specific options.

Tabs shift between
Experiment Setup/Observations/Analysis

4.8.2 The Project Trees

Experiment Setup Tree

This tree holds information on the subjects; Name, Age and Gender. The defined AOIs are found under Stimuli.



All Subjects in Project

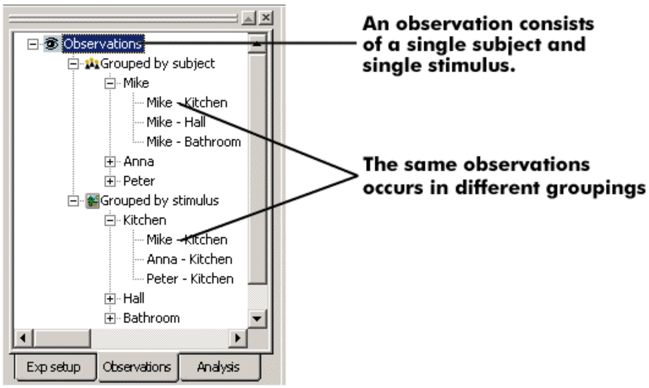
All Stimuli in Project

AOIs in a Stimulus

Expand/Collapse Tree Structure

Observations Tree

This tree holds information on which subjects have observed which stimuli. Temporal AOIs are found here, as they occur during an observation.

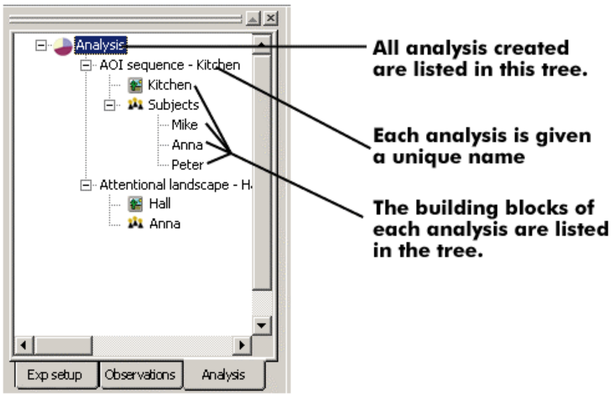


An observation consists of a single subject and single stimulus.

The same observations occurs in different groupings

Analysis Tree

The analysis tree keeps track of all analyses created. The stimuli and subjects that make up the analysis are showed for each analysis.



All analysis created are listed in this tree.

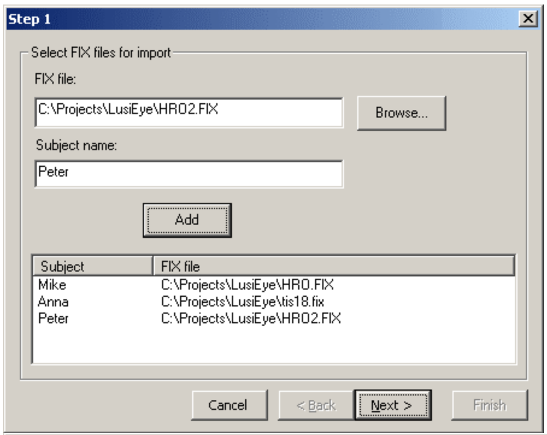
Each analysis is given a unique name

The building blocks of each analysis are listed in the tree.

4.8.3 Import Wizard

The import function is created as a Wizard (a Microsoft term) to aid users in the process of experiment set up. Shneiderman (1998) suggests wizards as a way to guide users through a complex decision-making process, by presenting one decision at a time.

User can enter some subjects and stimuli and then jump between these dialog steps as they wish, adding information until they are satisfied. Users feel safe by knowing that they do not make final decisions until they press the finish button.



Step 1

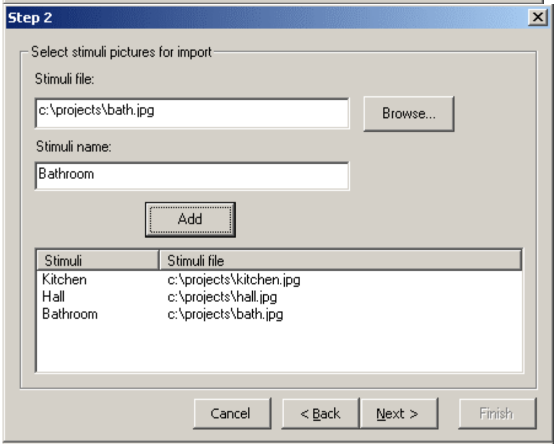
Select FIX files for import

FIX file: C:\Projects\LusiEye\HR02.FIX

Subject name: Peter

Subject	FIX file
Mike	C:\Projects\LusiEye\HR0.FIX
Anna	C:\Projects\LusiEye\tis18.fix
Peter	C:\Projects\LusiEye\HR02.FIX

Step 1/3 .FIX files containing actual eye tracking data from iView are selected, and name of the subject that generated the data is specified, to help keep track of all subjects.



Step 2

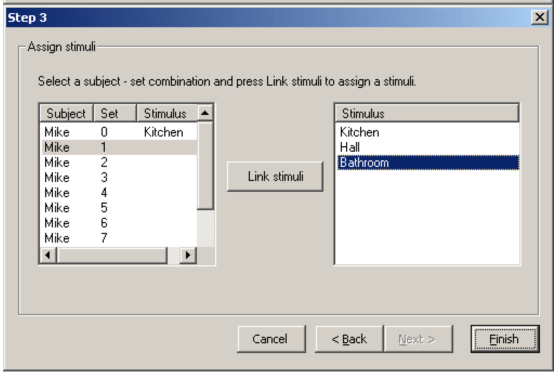
Select stimuli pictures for import

Stimuli file: c:\projects\bath.jpg

Stimuli name: Bathroom

Stimuli	Stimuli file
Kitchen	c:\projects\kitchen.jpg
Hall	c:\projects\hall.jpg
Bathroom	c:\projects\bath.jpg

Step 2/3 .JPG/.BMP files containing bitmap images of stimuli are selected. The names of stimuli are specified, to help keep track of all stimuli.



Step 3

Assign stimuli

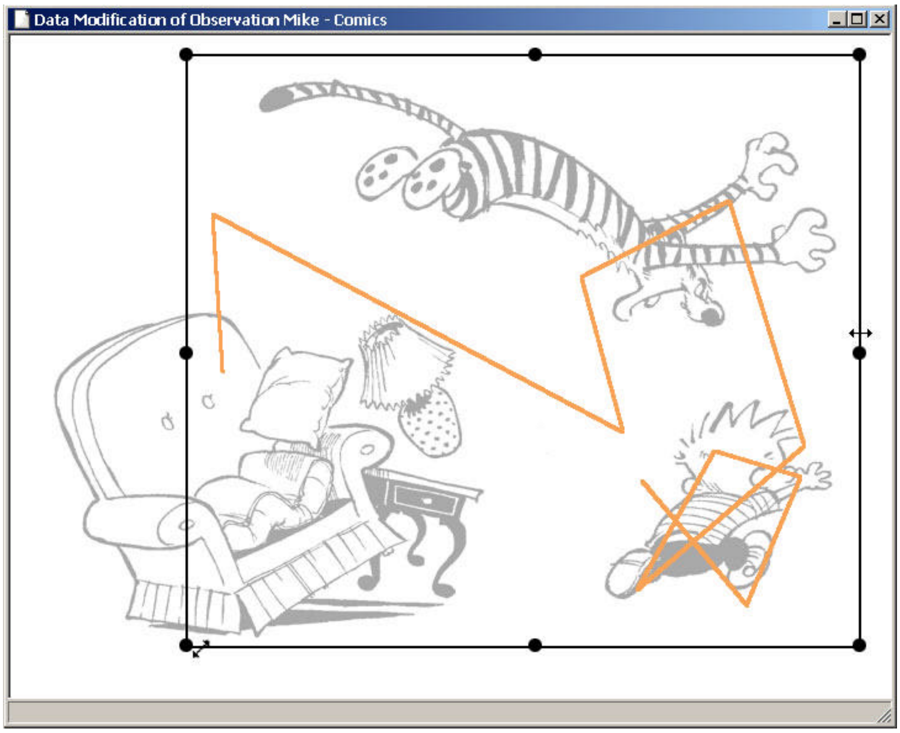
Select a subject - set combination and press Link stimuli to assign a stimuli.

Subject	Set	Stimulus
Mike	0	Kitchen
Mike	1	
Mike	2	
Mike	3	
Mike	4	
Mike	5	
Mike	6	
Mike	7	

Stimulus
Kitchen
Hall
Bathroom

Step 3/3 Each .FIX file contains one or more sets, each corresponding to a stimuli. The user specifies this information by selecting a set and double clicking on the correct stimuli. This step applies only when data was collected without the use of the IICap stimuli client. IICap will automatically perform this step if data was collected with the use of the IICap stimuli client for stimuli presentation.

4.8.4 Data Modification Window



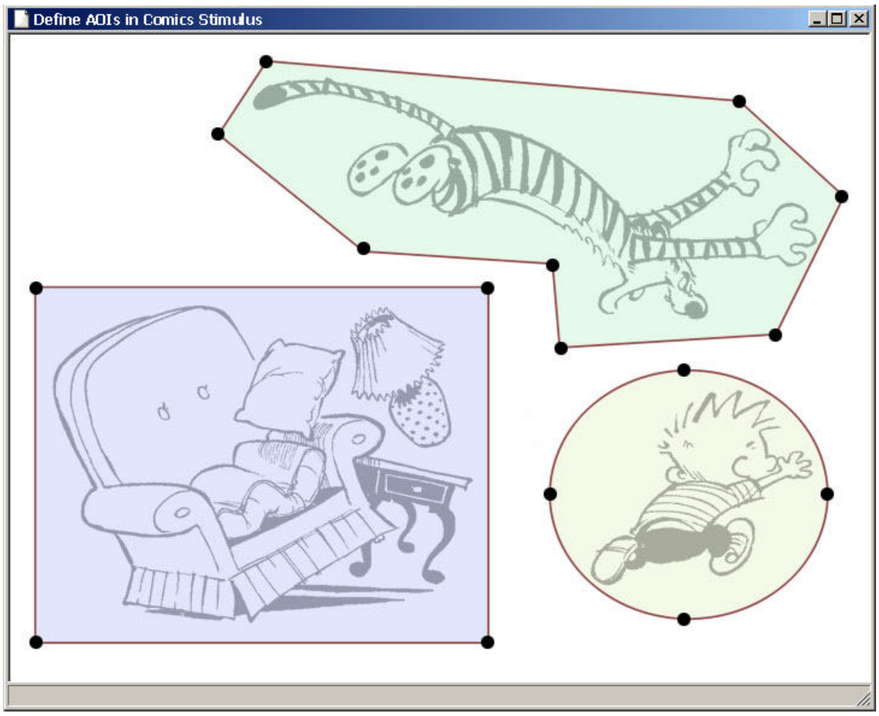
This window is used to modify data that has been collected with faulty calibration; it is only available under the edit menu if an observation is selected in the observations tree. When the window opens other windows are grayed out to show that only interaction in the current dialog is possible.

Scan paths are shown, as they are easy to adjust to underlying stimuli. No toolbox is available since the only two functions needed are move and scale. The user either places the mouse over the box to click and move all data, or places the mouse over the border of the rectangle to scale the data.

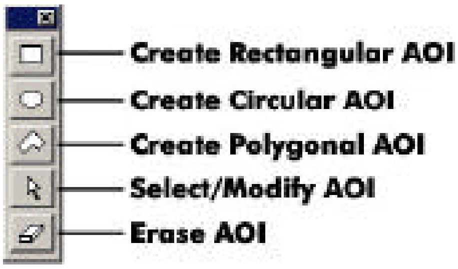
Upon closure of this window a dialogue will ask for confirmation of the modifications made. If confirmed, all analysis windows are automatically updated to reflect the data modifications made.

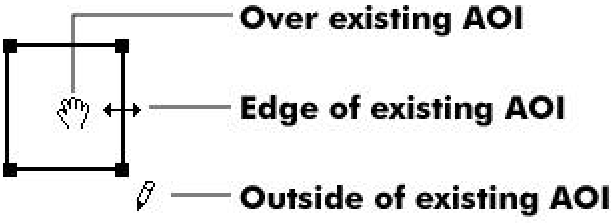
The data modification window is presented as a solution to problems presented in section 2.5.1.

4.8.5 Define AOIs



This window is used to create and modify AOIs, it is only available under the edit menu if a stimulus is selected in any of the trees. When the window opens other windows are grayed out to show that only interaction in the current dialog is possible. A floating toolbox with essential tools is shown:



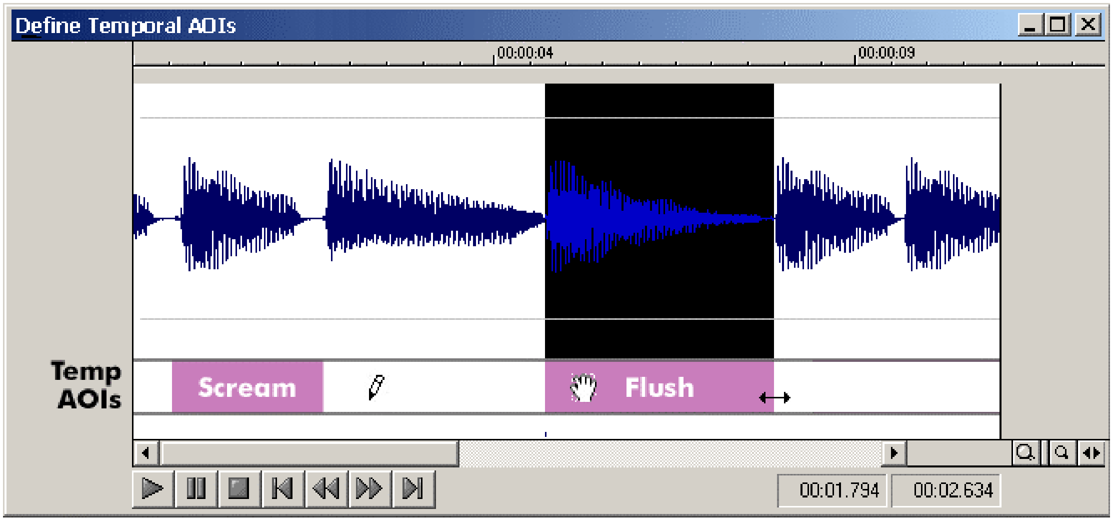


The mouse pointer changes depending on background content to give feedback to the user, showing what actions are possible.

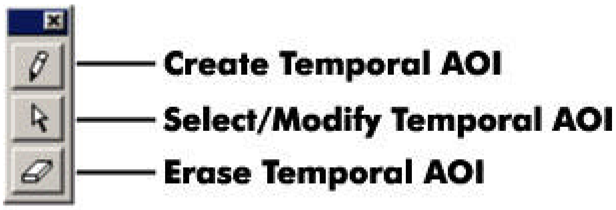
When a new AOI is created a dialog box appears where the user may enter the desired name of the AOI.

Possible interactions in this analysis are shown in appendix, section 1.

4.8.6 Define Temporal AOIs



This window is used to create and modify Temporal AOIs. When the window opens other windows are grayed out to show that only interaction in the current window is possible. A floating toolbox with essential tools is shown:

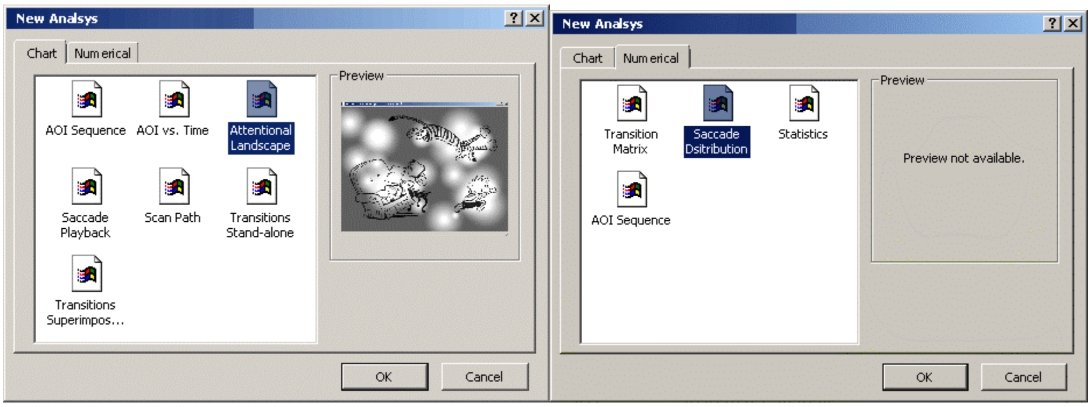


In order to assist the creation of Temporal AOIs there is an opportunity of loading a wave sound file into the Temporal AOI window. The waveform of the sound loaded is displayed along with the time-axis making it easy to identify important sound passages during which Temporal AOIs should be defined. By selecting one of the sound control buttons in the lower left part of the dialog window the sound loaded can be played back through the computer soundcard.

Temporal AOIs are presented as a solution to problems presented in section 2.6.4

Possible interactions in this analysis are shown in appendix, section 2.

4.8.7 Creating a New Analysis



In this dialog the user can choose to create one of the two types of analyses available in IICap: chart or numerical. Numerical analyses show only figures, while chart analyses are used to visualize data in many different ways. When a new analysis is selected a new window is presented with instructions on how to select subjects and stimuli. Depending on analysis type, either one or multiple subjects and stimuli can be used:

Please drag and drop the following
from the project tree on your left
to this window to create an analysis:

One stimulus and
One subject

Please drag and drop the following
from the project tree on your left
to this window to create an analysis:

Multiple stimuli and
Multiple subjects

Please drag and drop the following
from the project tree on your left
to this window to create an analysis:

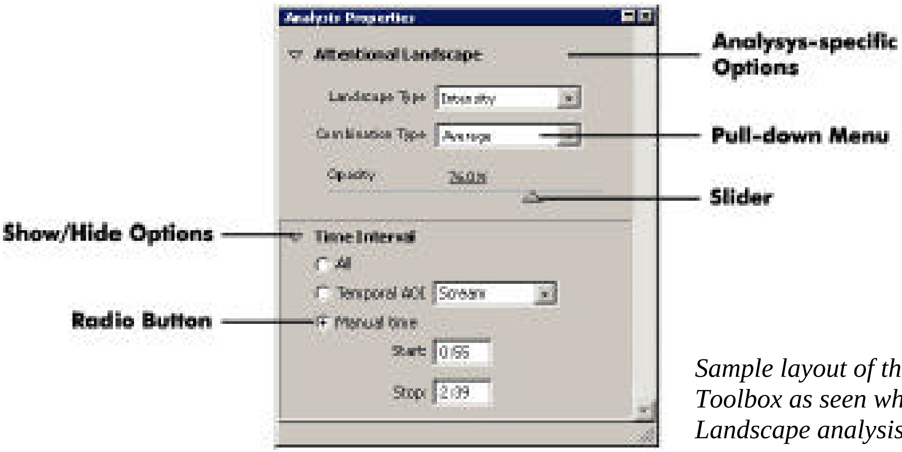
One stimulus and
Multiple subjects

The user then drags a subject or stimulus from the project tree and drops it in the window. Once either is dropped, the corresponding instruction disappears. When both subject and stimulus have been dropped in the window, the analysis is presented. This method offers immediate feedback of user actions and makes use of the existing tree structure to aid selection of desired subject and stimulus. The dragging and dropping of subjects and stimuli to create an analysis can be seen as a metaphor for building a construction using basic building blocks. The object of the design is to make users aware of the fact that an analysis is built from stimuli and subjects.

4.8.8 The Analysis Properties Toolbox

When a new analysis is created a floating Analysis Properties Toolbox is shown. This toolbox contains options that are specific for each type of analysis. The options available change dynamically depending on which type of analysis is selected.

In most of the analysis types the user can specify which time interval the analyses should be based on, this is always done in the Properties Toolbox.



Analysis Properties

Attentional Landscape

Landscape Type: Intensity

Combination Type: Average

Opacity 76.0%

Time Interval

○ All

○ Temporal AOI Screen

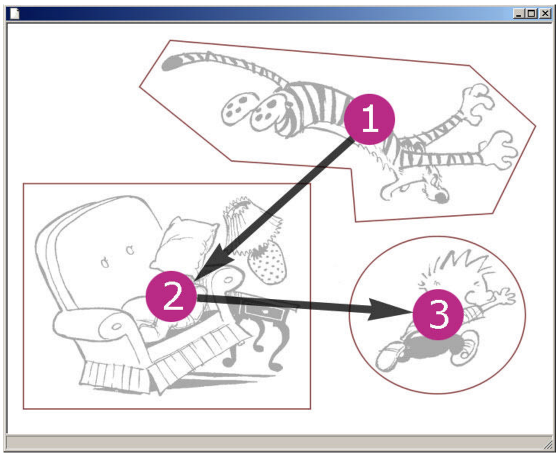
● Manual time

Start: 0:55

Stop: 2:09

Sample layout of the Analysis Properties Toolbox as seen when an Attentional Landscape analysis is currently selected

4.8.9 AOI Sequence

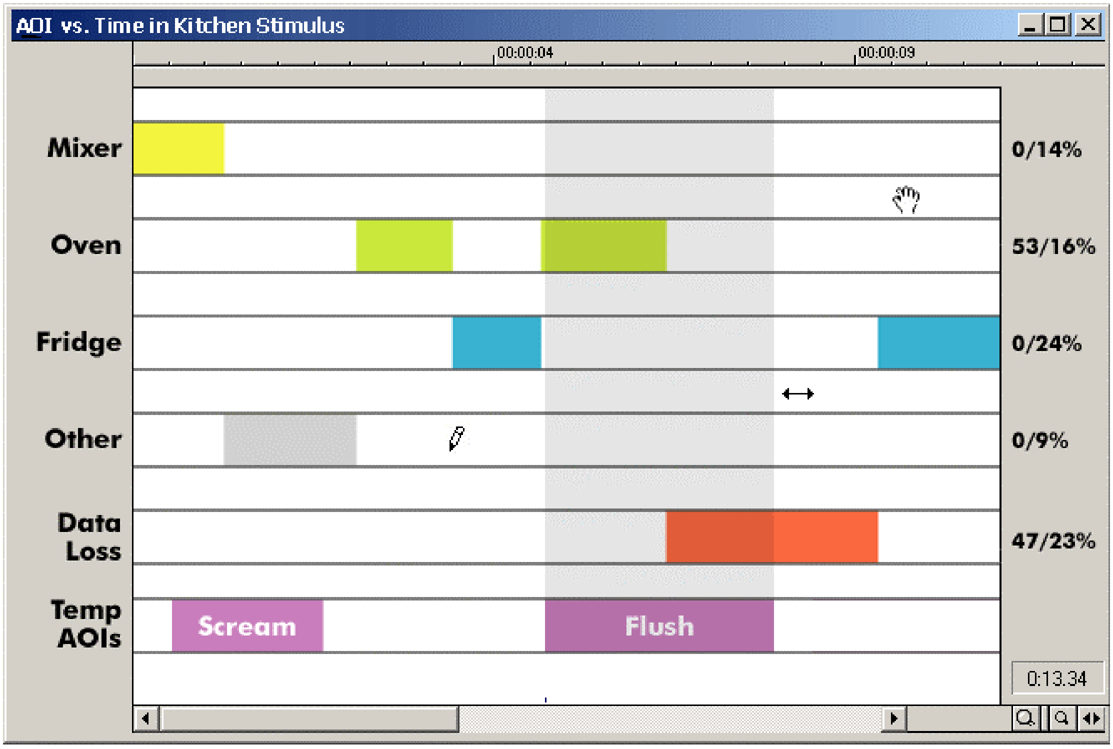


This chart analysis shows in which order one or multiple subjects have looked at the AOIs in a stimulus. The data is calculated on first occurrences, numbers are superimposed on the stimulus and arrows going from AOI 1 to 2 and so on are shown. AOI sequence is based on a single stimulus and multiple subjects. If multiple subjects are selected an index is built to rank AOIs before numbers are shown, as described in section 3.4.3.

The analysis presents the sequence in a way that most people can understand, since no knowledge of eye tracking is required. It is superior to a scan path view that does not show any temporal aspects, and should be a help when designers want immediate feedback on their design.

Possible settings for this analysis are shown in appendix, section 3.

4.8.10 AOI vs. Time



This chart analysis is similar to the one found in iView, but with some features added. Users are able to continuously zoom in on and out of the timeline thus enabling even short intervals to be seen in detail. In addition to the defined AOIs three rows have been added; Temporal AOIs, Other and Data Loss. The Temporal AOIs are used so select a specific time interval for analysis. Other shows when a subject has looked in an area not covered by the AOIs defined.

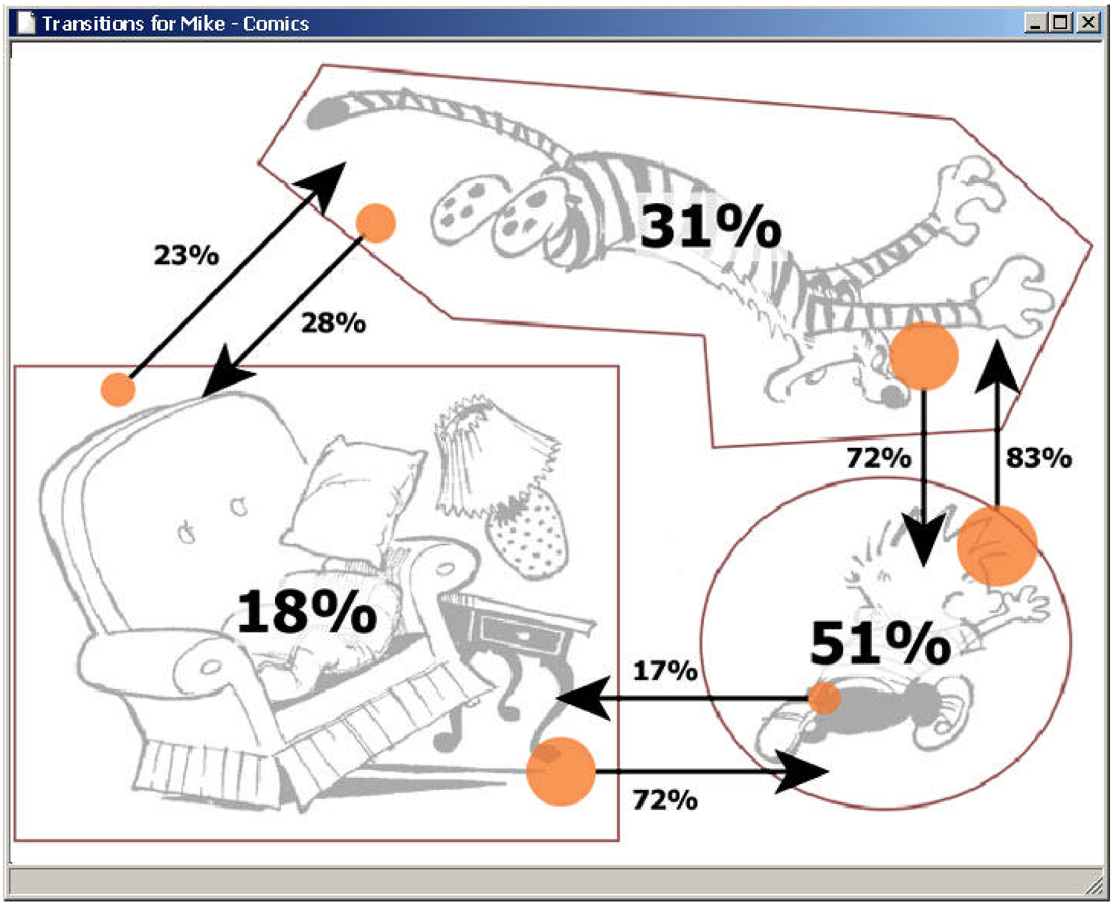
Data Loss indicates that during a time interval, data collection has failed and gaze position cannot be secured. The data loss indication is presented as a solution to problems presented in section 2.7.3

When a selection is active, the first figure on the right shows fixation distribution based on selection only. Figures showing fixation distribution based on all time is always available.

AOI vs. Time is based on a single stimulus and a single subject. Multiple subjects are not allowed; as this would yield parallel temporal AOIs that could clutter the screen. For users who wish to visualize multiple subjects, it is possible to create multiple views and arrange the windows side by side.

Possible settings for and interactions in this analysis are shown in appendix, section 4.

4.8.11 Transitions - Superimposed on Stimulus

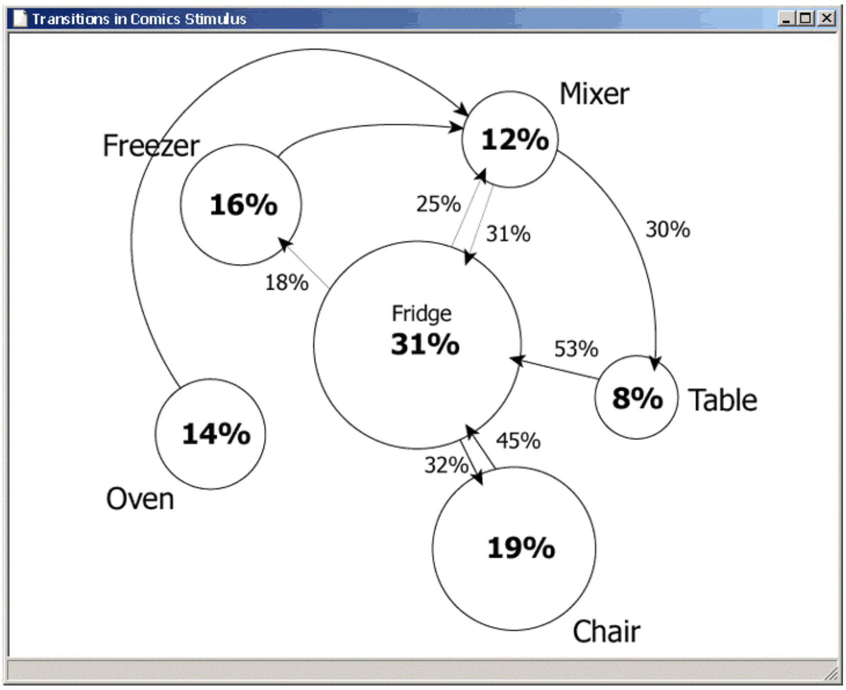


This chart analysis shows percentage of transitions between AOIs with arrows. The percentage of fixations in each AOI is also shown. The AOIs are superimposed on the stimulus so that spatial information of the stimulus is preserved.

When the analysis is first presented there are no arrows, as arrows from every object to every other object can clutter the screen. Instead, the user draws arrows manually by selecting the Create Arrow Tool from a floating toolbox. When an arrow between two AOIs is created, percentage corresponding to transitions between the AOIs is shown. At the base of each arrow is a circular analogous representation of the percentage to complement the figures in the chart. An analogous representation is faster when doing relative comparisons, though not that exact Tufte (1990).
The analysis is based on a single stimulus and one or multiple subjects. If multiple subjects are chosen, they are averaged to present a single diagram.

Possible settings for and interactions in this analysis are shown in appendix, section 5.

4.8.12 Transitions – Stand-alone diagram



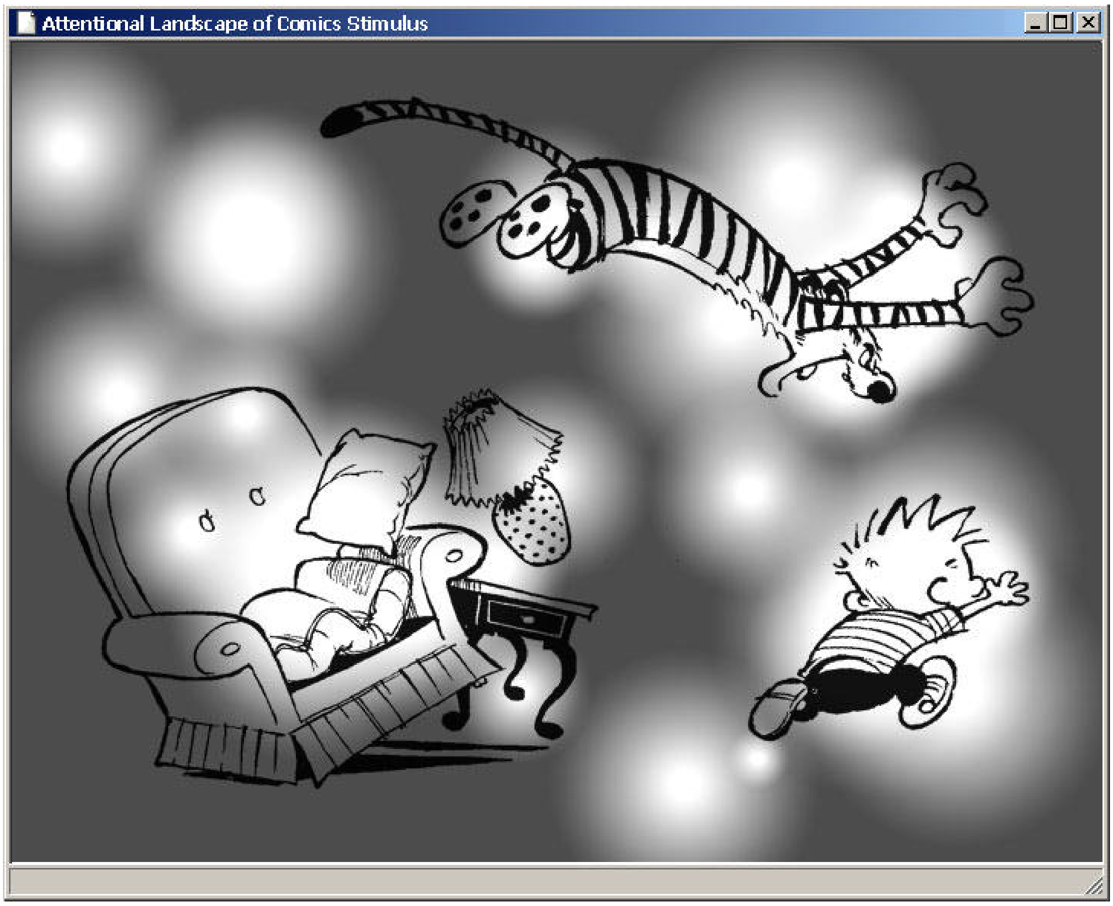
This chart analysis visualizes each AOI defined as a state represented by a circle. Arrows represent possible state transitions. The percentage given for each transition arrow tells the probability of transition from the source state (arrow start) to the destination state (arrow end). The percentage of fixations in each state (AOI) is shown in the circle center. In this analysis the actual stimulus is not shown. A central state is selected – most often the one where most fixations have fallen – and other states are placed around this central state. The circle representing a state is sized to represent the amount of fixations that have fallen in it, giving an analogous representation that is not available in the chart that is superimposed on the stimulus.

When this analysis is presented, all states and transitions are drawn. However, to prevent the screen from getting cluttered with low probability transitions the user can set a probability threshold beneath which no transitions are drawn.

The Transitions stand-alone analysis is based on a single stimulus and multiple subjects.

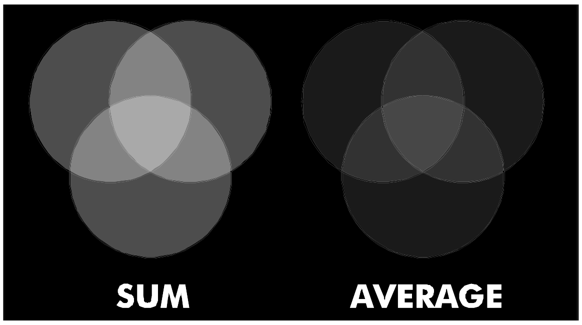
Possible settings for this analysis are shown in appendix, section 6.

4.8.13 Attentional Landscapes



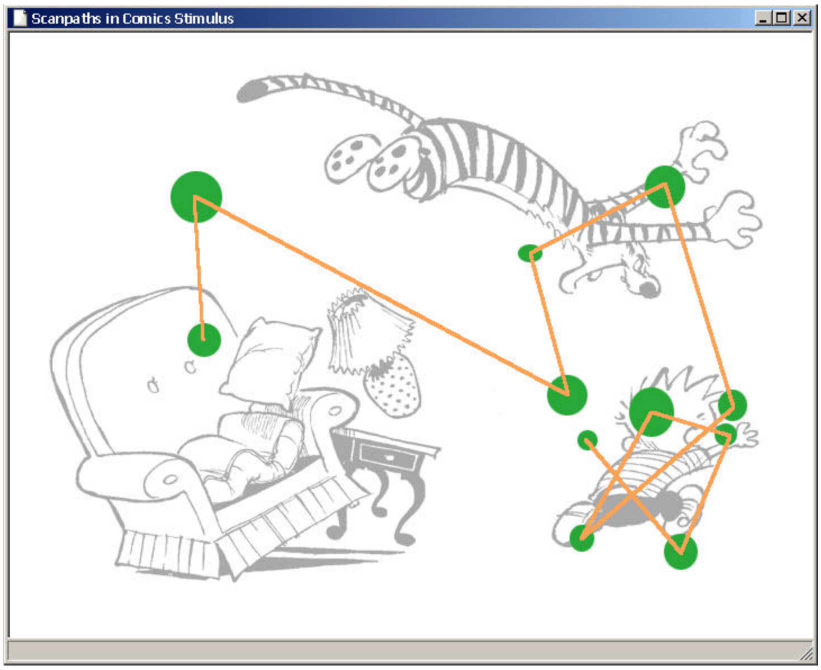
This chart analysis shows where one or multiple subjects have looked as an attentional landscape. The user can choose between two different kinds of landscapes; Intensity or Heat Signature. The Intensity Landscape is the look presented by Velichkovsky et al. (1995), with a gradient stretching from black to transparent. The Heat Signature has a gradient stretching from blue via yellow to red, to indicate where the most fixations have fallen.

The analysis is based on a single stimulus and one ore multiple subjects. If multiple subjects are chosen, the landscape can be either a sum or an average of the subjects' fixations.



Possible settings for this analysis are shown in appendix, section 7.

4.8.14 Scan Paths

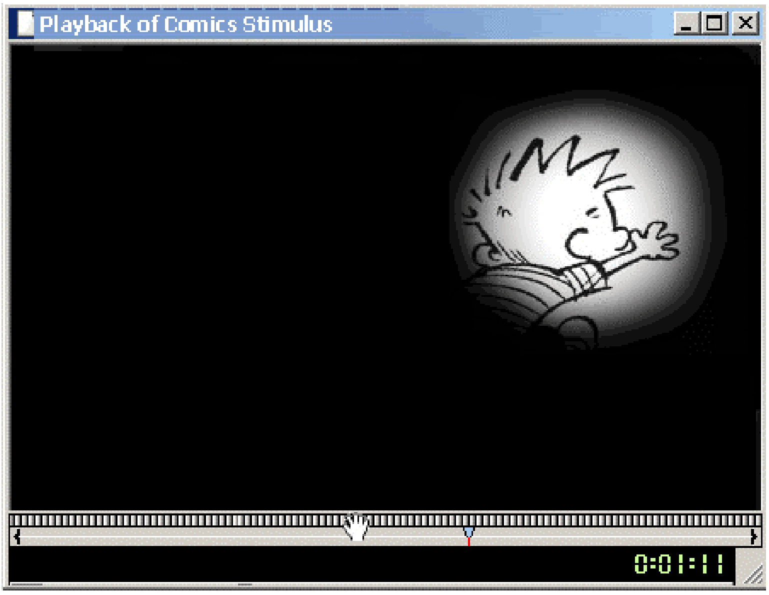


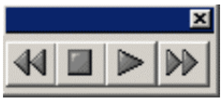
This chart analysis shows actual fixations and saccades of a single subject. The visualization type is very common in eye tracking software, as it is the most straightforward way of showing saccades and fixations. Fixations are shown as circles with size mapped to corresponding fixation time.

The analysis is based on a single stimulus and a single subject.

Possible settings for this analysis are shown in appendix, section 8.

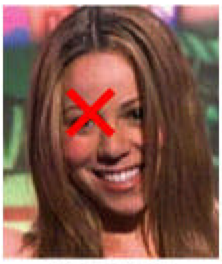
4.8.15 Playback

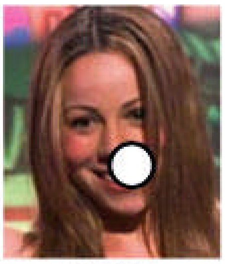


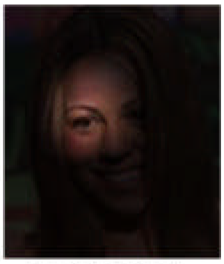


This chart analysis shows actual gaze position of a single subject. The visualization will be very similar to playing back an eye tracking experiment that has been recorded to video, where gaze position is shown as a superimposed dot. When the analysis is created a floating toolbox is shown. The toolbox contains functions that are normally found on a video player; Play, Stop, Rewind and Fast Forward.

There are three different types of gaze indicators; Cross, Dot and Torch.


CROSS


DOT


TORCH

The Torch mode is similar to attentional landscapes, the stimuli can only be seen where the subject is currently looking. To aid the viewing of saccade playback the user has the option to set a trail that follows the gaze indicator.

The analysis is based on a single stimulus and a single subject and is presented as a solution to problems presented in section 2.8.4

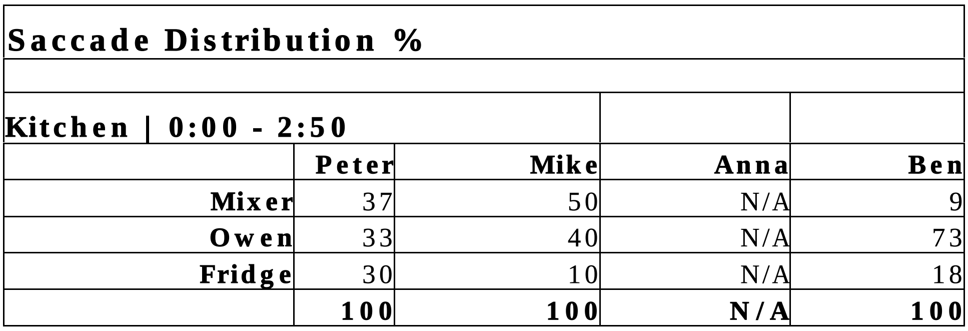
Possible settings for this analysis are shown in appendix, section 9.

4.8.16 Statistics

All the numerical analyses are used to export data to Microsoft Excel.
There are four different kinds of numerical analyses available

Saccade Distribution	Shows percentage of fixations in different AOIs.
AOI Sequence	Shows an index of first occurrences.
Transition Matrix	Shows number of transitions between AOIs.
Statistics	Shows detailed numerical statistics, based on saccade direction, length and velocity and fixation duration.

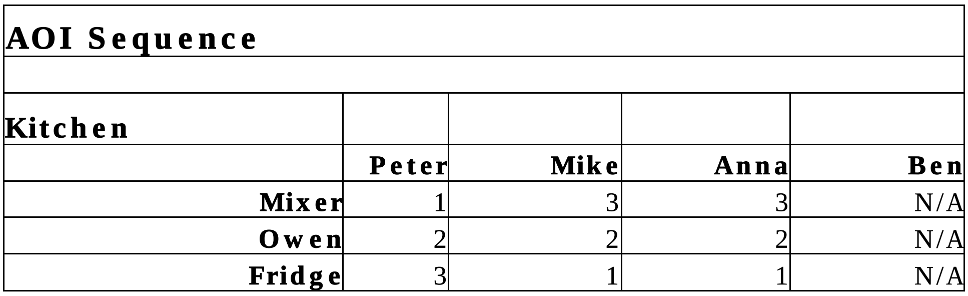
Below are sample spreadsheets from each analysis:



Saccade Distribution %

Kitchen | 0:00 - 2:50

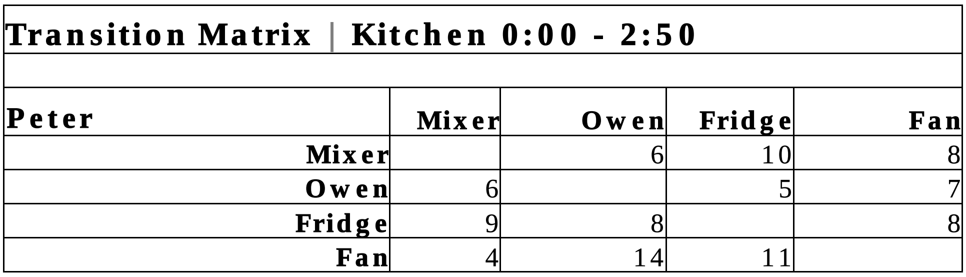
	Peter	**Mike**	**Anna**	**Ben**
Mixer	37	50	N/A	9
Owen	33	40	N/A	73
Fridge	30	10	N/A	18
	100	**100**	**N/A**	**100**



AOI Sequence

Kitchen

	Peter	**Mike**	**Anna**	**Ben**
Mixer	1	3	3	N/A
Owen	2	2	2	N/A
Fridge	3	1	1	N/A



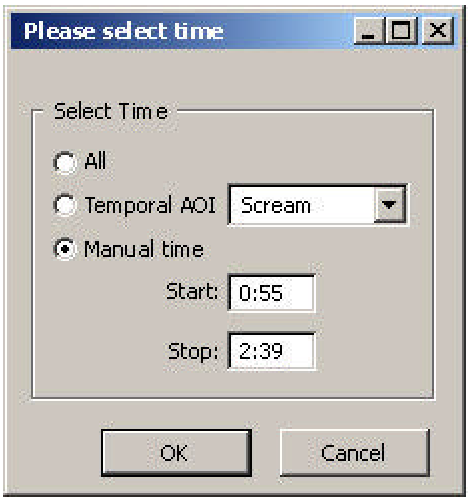
Transition Matrix | Kitchen 0:00 - 2:50

Peter	**Mixer**	**Owen**	**Fridge**	**Fan**
Mixer		6	10	8
Owen	6		5	7
Fridge	9	8		8
Fan	4	14	11	

Statistics						
Peter						
	Kitchen | 0:00 - 2:33			**Bathroom | 0:00 - 2:33**		
		Owen	Fridge		Shower	Bathtub
Average saccade length (in pixels)	**43**	19	24	**21**	10	11
Average saccade duration (in ms)	**210**	180	240	**200**	190	210
Average saccade velocity (pixels/ms)	**24**	60	33	**2**	8	3
Average number of saccades/S	**4.2**	4.1	4.3	**5.4**	5.2	5.6
Maximum fixation duration (in ms)	**1750**	1700	1800	**1350**	1300	1400
Average fixation duration (in ms)	**250**	240	260	**320**	310	330
Fixation/saccade time ratio	**3.2:1**	3.0:1	3.4:1	**2.8:1**	2.8:1	2.8:1
Number of saccades/fixations	**6500**	3000	3500	**5700**	2600	3100
Average pupil diameter (mm)	**4.2**	3.9	4.3	**4.1**	4.1	4.1

All numerical analyses support multiple subjects. If one table per subject is required multiple tables are created in one spreadsheet. Saccade Distribution, AOI Sequence, and Statistics analysis support multiple stimuli, while the Transition Matrix only accepts a single stimulus. If no observation data is available, N/A is shown.

When creating a numerical analysis, all subjects have to be added first, to avoid mismatching of temporal AOIs. Then stimuli are added and for every stimulus the user is asked to specify the time interval used to calculate statistics from. The user can choose to include all observation time or select a Temporal AOI that matches the desired time. There is also an option to enter manual start and stop times.



Select Time

○ All

○ Temporal AOI: Scream

● Manual time

Start: 0:55

Stop: 2:39

5. Discussion

5.1 The Application

Implementation of the fully suggested design proposal was not in the scope of this thesis and we have not had the time to perform a usability test of the application. However, the suggested functionality is not as dependent on a user test as the user interface. Thus, we argue that the functionality presented in IICap - such as the automated statistics, the project concept and export functions – by itself will be sufficient to save the LUCS personnel significant amounts of time.

The project hierarchy should decrease the cognitive load of the personnel, as they do not need to keep track of stimuli, subjects and AOIs manually. Furthermore, the analogy of a real life project that includes everything related to an experiment should help the personnel feel that they are in control of all relevant data.

By presenting a design proposal that was developed using the latest Windows technology we were able to take advantage of new improvements in the Windows user interface. iView was originally designed for a Windows 3.1 environment. A Windows 95 environment is more convenient to the personnel, as the operating systems used are Windows 95 and Windows 98.

We have presented features that are useful from a usability point of view, but missing in iView; Undo and Redo that decreases anxiety in users, wizards that guide users through a complicated process, toolbar icons in menus that aid learning and multiple windows to allow comparisons between different subjects and stimuli without the need to open new files.

Many different chart analyses are available in IICap, as people with diverse background are expected to participate in the interpretation of experiment results. People have different preferences in how information is visualized and this should be taken into account when an application like IICap is designed.

5.2 Suggested Features

The design proposal presented includes full functionality, but during our work with this thesis, some ideas that were not presented in chapter 4 have come up.
Though not in the scope of the IICap software, we suggest that these ideas - presented below - are further explored.

5.2.1 Plug-ins

The user interface design of IICap provides the possibility to design homogenous third party plug-ins. Once a new plug-in analysis module is available it emerges as a new icon in the New Analysis dialog.

The selection of subjects, observations and stimuli are performed in the same way as for the basic analysis modules, by dragging and dropping items from the project trees. Plug-in developers are not able to change the layout of menus or project trees thus reducing the risk of divergent interface design.

The only way for a third party developer to provide custom controls for an analysis is through the Analysis Properties tool window. A development kit with documentation should be available, including guidelines for the design of custom controls in the tool window.

The introduction of a proper plug-in structure would allow researches around the globe to develop and exchange their own plug-in analysis modules. As worldwide exchange of plug-ins is the primary goal of the plug-in structure, it is of great importance to provide a well functioning website were this exchange can take place.

5.2.2 Correlating Eye Tracking Data and Web Statistics

Most web hotel service providers keep detailed statistics of click rates on links from pages hosted. Detailed statistics of number of page loads are often kept as well. In research on web interface design and web user behaviour it might be interesting to correlate eye tracking data with link click and page load statistics. This would provide an opportunity to see if users actually click were they look, and if some behaviours such as scanning or reading are associated with greater or lesser click through rates.

5.2.3 Export of Attentional Landscape as Video File

This feature would allow users to export the real-time playback of saccadic data in torch mode as a greyscale video file. When imported into an editing program such as Adobe Premiere, this mask could be overlaid onto video stimuli. The result would be a video with an overview of where a subject has been looking, even though the stimuli is video and not still images.

5.2.4 3-way or More Transition Diagram

Currently the transition diagrams only show transitions between two AOIs. It is possible to see how many saccades have gone from AOI A to B, B to C and A to C. However there is no way to see how many percent of the fixations landed in AOI A that are followed by a saccade on to AOI B and C respectively. We suggest a feature that will allow this kind of analysis that involves more than two AOIs. Furthermore, the user should be able to specify a start and stop AOI and get statistics and diagrams on the different transitions from AOI A to B. Some saccades might have gone straight from A to B while others might have gone from A to X to Y and then to B. A realistic approach would be to maximize the number of intermediate AOIs to two, since the number of possible ways grows at an exponential rate. This type of diagram would provide even more detailed analysis than the current 2-way transition diagrams.

5.2.5 Simulation of Peripheral Falloff

We suggest that the torch playback function in IICap could be used in real-time to simulate peripheral falloff.

Users could set up a specific function for falloff from the fovea to the peripheral field of vision. Falloff could be set up as a function of degree. While Velichkovsky & Hansen (1996) only included brightness falloff in their function we suggest letting degree affect one or more of the following:

Brightness
Resolution (blur)
Saturation
Contrast

These functions should be customisable in all aspects and could be used in the stimuli client to modify the stimuli in real-time according to where a test person looks. By providing the tools needed, we hope that further experimentation on peripheral falloff will take place. It will be possible to simulate tunnel vision, macular degeneration etc. The experiments will take off from the results of Velichkovsky & Hansen (1996), but with a temporal domain and more variables added.

Parkhurst et al. (2000) have shown that gaze dependent variable resolution displays might be effective by allocating more CPU power to where a person is currently looking on a screen. However, more research on the cognitive aspects of this method is needed, and the suggested feature would be a tool for this.

5.2.6 Rule-based Eye Gaze Pattern Detection

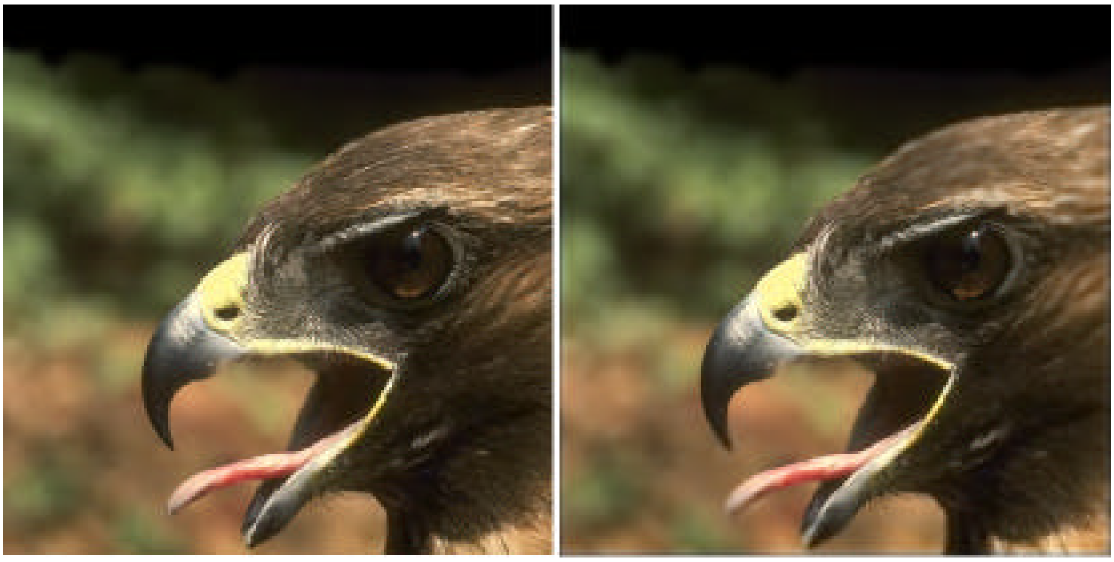
In her thesis work, Barthelson (2002) used a filter to extract statistics of certain fixation patterns corresponding to reading activity. By using this filter Barthelson was able to distinguish the reading activities from other activities such as searching and dwelling.

Such a filter could be implemented in IICap to address the problems identified in section 2.7.1. Users would be able to identify patterns by setting up rules for data selection. For instance; one could look for patterns of »four or more short consecutive saccades to the right, in a direction 90 +/-10 degrees, followed by a long saccade to the left« to find reading patterns. These rules could specify any of the saccade characteristics mentioned in section 2.3.1, in any given order.

5.3 Beyond IICap - Eye Movements and Image Compression

The study of eye movements introduces an opportunity of enhancing existing image compression techniques. A compression algorithm could be fed with an averaged attentional landscape as generated by the attentional landscape analysis module in IICap. The attentional landscape should be created from an experiment where several subjects look at the same picture for approximately ten seconds each. From this averaged attentional landscape, the compression algorithm could extract information to decide compression ratio at different parts of the stimulus picture. Areas not likely to be observed by a subject are compressed harder than areas more likely to be observed.

This technique was presented by Johansson (Forthcoming) and the result of his brief pilot study performed at LUCS is shown below.



Original picture (left) was compressed using an attentional landscape mask based on eye tracking data from five subjects; the result was an image compression ratio of 100:1 (right)

This technique would be especially useful for images that are to be presented only during a short time and that are to be transmitted over a low bandwidth transmission medium.

If this technique shows to be of interest, the IICap software should be equipped with an export function to format and export averaged attentional landscapes to suit eye tracking aided image compression. Another opportunity is to encapsulate the complete image compression process into IICap.

6. Acknowledgements

We would like to thank the personnel at LUCS for their theoretical and practical help. We will also like to thank the people at Picofun AB in Lund for their encouragement and support.

7. References

Aaltonen, A. (1999) *Eye Tracking in Usability Testing: Is it Worthwhile?* CHI'99 Workshop The Hunt for Usability: Tracking Eye Movements, Pittsburgh, PA, May 1999

Barthelson, M. (2002) *Behaviour in online news reading.* Unpublished

Buswell, G. T. (1935) *How people look at pictures*. Chicago: University of Chicago Press. Huey, E. B.

Cohen, J. D., Macwhinney B., Flatt M., Provost J. (1993) *PsyScope: A New Graphic Interactive Enviroment for Designing Psychology Experiments.* Beahaviour Reasearch Methods, Instruments, & Computers, 25: 257-271.

Dix, A. (1995) *Accelerators and toolbars: learning from the menu*
http://tina.lancs.ac.uk/computing/users/dixa/papers/menus95/toolbar.html

Hansen, J.P. (1991) *The use of eye mark recordings to support verbal retrospection in software testing.* Acta Psychologia 76, 31-49.

Goldberg, J. H., & Kotval, X. P. (1999) *Computer interface evaluation using eye movements: methods and constructs*. International Journal of *Industrial Ergonomics*, 24(6), 631-645

Holsánová, J. (2001): *Picture Viewing and Picture Description: Two Windows on the Mind.* Lund University Cognitive Studies 83

Johansson, K. (2001) *Investigating the application of eye tracking data on image compressibility*. Unpublished.

Kotval, X.P. & Goldberg, J.H. (1998) *Eye Movements and Interface Components Grouping: An Evaluation Method.* Proceedings of the 42nd Annual Meeting of the Human Factors and Ergonomics Society, Santa Monica: HFES.

Norman, K., The Psychology of Menu Selection: Designing Cognitive Control of the Human/Computer Interface, Ablex, Norwood, NJ (1990).

Parkhurst, D., Culurciello, E., Niebur, E. (2000) *Evaluating Variable Resolution Displays with Visual Search: Task Performance and Eye Movements*. Presented at Eye Tracking Reasearch & Applications Sumposium 2000, Palm Beach Gardens.

Rayner, K. (1978) *Eye movements in reading and information processing*. Psychological Bulletin, 85, 613-660.

Tufte, E.R. (1990) *Envisioning Information*. Graphic Press. Cheshire, Conn.

Van Orden, K.F., S., Makeig, T.P., Jung, W., Limbert (1999) *Eye activity correlates of workload during a visuospatial memory task.* Abstract submitted May 1, 1999 to the Human Factors Society.

Velichkovsky, B.M. & Hansen, J.P. (1996) *New technological windows into mind: There is more in eyes and brains for human-computer interaction.* In CHI-96: Human factors in computing systems. NY: ACM Press

Yarbus, A. L. (1967) *Eye movements during perception of complex objects*, *in* L. A. Riggs, ed., `Eye Movements and Vision', Plenum Press, New York, chapter VII, pp. 171-196.

Shneiderman, B. (1983). *Direct manipulation: A step beyond programming languages*, IEEE Computer 16, 8, (August 1983), 57-69.

Shneiderman, B. (1998) *Designing the User Interface: Strategies for Effective Human-Computer-Interaction*. Addison Wesley.

Itoh, K., Hansen J.P., Nielsen, F.R (1998) *Cognitive Modelling of a Ship Navigator Based on Protocol and Eye-movement Analysis*. Le Travail Humain, tome 61, n0 2/1998, 99-127

Appendix

1. Define AOI Dialog

Possible interactions

Tool selected	Mouse over	Pointer shown	Mouse action	Result
Any of the create tools	Area not covered by an existing AOI	Pen	Click and drag	A new AOI matching the shape of selected tool is created
	Existing AOI	Hand	Click and drag	Moves the current AOI
	Edge of existing AOI	Arrow	Click and drag	Changes size of current AOI
Select/Modify arrow	Existing AOI	Arrow	Click	Selects object
	Existing AOI	Arrow	Multiple clicks while holding down the SHIFT key	Selects multiple AOIs
	Area not covered by an existing AOI	Arrow + bounding box	Click and drag over multiple AOIs	Selects multiple AOIs
Eraser	Existing AOI	Eraser	Click	Deletes object if confirmed in dialog

2. Define Temporal AOI Dialog

Possible interactions

Tool selected	Mouse over	Pointer shown	Mouse action	Result
Create tool or Select/modify arrow	Temporal AOI	Hand	Click and drag	Move AOI
	Edges of a temporal AOI.	Arrow	Click and drag	Compress/Expand current AOI
	Area not covered by Temporal AOI	Pen	Click and drag	Create new Temporal AOI
Eraser	Existing temporal AOI	Eraser	Click	Deletes object if confirmed in dialog

3. AIO Sequence

Analysis Properties Toolbox:

Function	Settings	Comment
Show numbers	Check box	Shows or hides numbers
Show arrows	Check box	Shows or hides arrows

4. AOI vs. Time

Possible interactions

Mouse over	Pointer shown	Mouse action	Result
Selection	Hand	Click and drag	Move selection
Edges of selection	Arrow	Click and drag	Compress/Expand current selection
Area not covered by selection	Pen	Click and drag	Create new selection
Temporal object	Arrow	Double click	New selection matching the temporal AOI clicked

Analysis Properties Toolbox:

Function	Settings	Comment
Show Other	Check box	Shows or hides the Other row
Show Data Loss	Check box	Shows or hides the Data Loss row

5. Transitions – Superimposed on Stimuli

Possible interactions

Tool selected	Mouse over	Pointer shown	Mouse action	Result
Create Arrow	Existing AOI	Crosshair	Click and drag to another AOI	A new arrow between the two AOIs is created
Select/Modify arrow	Start or end point of existing arrow	Arrow	Click and drag to another AOI	Changes the arrow to point to or from a new AOI
Eraser	Existing arrow	Eraser	Click	Deletes arrow if confirmed in dialog

Analysis Properties Toolbox

Function	Settings	Comment
Show numbers	Check box	Shows or hides numbers
Show arrows	Check box	Shows or hides arrows
Circles	Check box	Shows or hides circles

6. Transition – Stand Alone Diagram

Analysis Properties Toolbox:

Function	Settings	Comment
Set Threshold	Slider	Filters what transitions are shown
Show numbers	Check box	Shows or hides numbers
Show arrows	Check box	Shows or hides arrows
Central AOI	Pull down menu	Selects which AOI should placed in the center of the chart

7. Attentional Landscape

Analysis Properties Toolbox:

Function	Settings	Comment
Landscape Type	Pull down menu	The user can select two different types: Intensity or Heat Signature
Opacity of Landscape	Slider	Adjusts how much the stimulus shines through
Combination Type	Pull down menu	The user can select two different types: Sum or Average

8. Scan Paths

Analysis Properties Toolbox:

Function	Settings	Comment
Show Paths	Check box	Shows or hides scan paths
Show Fixations	Check box	Shows or hides fixation spots

9. Playback

Analysis Properties Toolbox:

Function	Settings	Comment
Gaze Indicator	Radio buttons	The user can select three different types: Dot, Cross or Torch
Size	Slider	Sets width of the gaze indicator in pixels.
Trail Length	Slider	Sets length of the trail in seconds.