

EyeLinkServer

Michael Stephan

December 18, 2019

Contents

1	Purpose	1
2	How it works	1
A	Commands	2
A.1	General Commands	2
A.2	Target Creation	3
A.3	Target Commands	3

1 Purpose

This server is intended to be controlled by a client application. It also provides information for that application. It monitors the gaze locations of a subject by means of an EyeLink eye tracker. Any number of client applications can read the current eye position from the server through “shared memory”. It can also monitor target regions on the subject’s screen and create events whenever the gaze position enters or leaves such a region. The regions are shown in the server’s main window and if EyeLink is in idle mode when the target is created, also on the EyeLink computer’s screen.

Events are also generated when blinks are detected by EyeLink. Optionally the server can be configured to generate “leave” events when blinks are detected (this may simplify the client’s event handling logic).

The server communicates with the EyeLink through a dedicated Ethernet link. Communication with the client is through a pipe and various kernel32 events.

2 How it works

Coordinates: (0,0) is the center of the screen. Positive values going to the right and up respectively. The auto reset event named **EyeServerDone** is signaled after server startup and when a “stop” command finishes (this may take some time when an **.edf** file is transferred).

The regions are drawn in the server's main window and if EyeLink is in idle mode when the target is created, also on the EyeLink computer's screen.

share memory name and structure

A Commands

Commands are sent through the named kernel32 pipe `EyeServerPipe`.

A.1 General Commands

[00 0 0] stop recording. If a file was specified in the start command, a file selection dialog will open. You may specify a destination for the `.edf` or abort the dialog (then the `.edf` will not be copied).

[00 0 0 filename] stop recording (and copy `.edf` to the specified file).

[00 0 1 width height] set screen size. This command is not needed if the EyeLink `.INI` files contain valid entries.

[00 0 2] start recording without `.edf`

[00 0 2 filename] start recording. The `filename` has to be specified with the extension (`.edf`) included. Note that this specifies the name of the file on the remote EyeLink computer. The filename is restricted to 8 characters (plus extension). Consider to always use the same filename. This will prevent the remote disk from filling up over time.

[00 0 3] remove transformation

[00 0 3 x0 y0 x1 y1] set coefficients (single precision floating point) for a linear transformation

[00 0 3 x0 y0 x1 y1 x2 y2] set coefficients (single precision floating point) for a quadratic transformation

[00 0 3 x0 y0 x1 y1 x2 y2 x3 y3 x4 y4 x5 y5] set coefficients (single precision floating point) for a mixed term quadratic transformation:

$$\begin{aligned}x' &= x_0 + x_1 \cdot x + x_2 \cdot y + x_3 \cdot xy + x_4 \cdot x^2 + x_5 \cdot y^2 \\y' &= y_0 + y_1 \cdot x + y_2 \cdot y + y_3 \cdot xy + y_4 \cdot x^2 + y_5 \cdot y^2\end{aligned}$$

[00 0 4 message] pass `message` to EyeLink to be stored into the `edf` file.

A.2 Target Creation

Each target creates two named manual reset kernel32 event objects: `nameIn` and `nameOut`. These events are set when the eye position enters or leaves the target region. In general the other event is reset at the same time. This behaviour can be switched off through a command (see below).

You have to read the key of the newly created target as a 16 bit unsigned integer after issuing this command.

If these commands are issued while the tracker is in offline mode (not recording) then the target regions are also drawn on the tracker's screen.

[00 1 *x y r name*] create a circular target with radius *r* at position (*x,y*). *x, y, r* are single precision floating point values.

A.3 Target Commands

[kk 0] remove target.

[kk 1] force target to generate `Out` events on blinks.

[kk 2] disable the exclusive signaling mode for the target.