VLAB User’s manual

Fangtu Qiu

March 9, 2003

Contents

1. Overview

VLab is a system providing task controlling, data collecting, and graphics rendering for the physiologists as well as psychologists in the field of vision or related researchs. It has two parts: one runs on a MS-Windows platform (Windows NT 4.0, Windows 2000 and Windows XP), the other in an unix box (SGI Iris, Linux). The Windows part controls the task, collects the data, displays the online plots and performs the online statistics. The unix part is an Open Inventor program. It loads the Inventor script files written in Virtual Reality Modeling Language (VRML) to render the stimulus graphics in 3D space. Those two parts communicate with each other through TCP/IP network and the timing is synchronized against the retrace of stimulus rendering monitor.

2. Task file

The VLab on the PC has a built in c-like compiler to parse the task file into an executable function list. The entry point of the task file should be a function named “TaskEntry()” with empty parameter list. The TaskEntry will be inserted into the main loop of the VLab and executed repeatly when the “Resume/Pause” button is in active status (pressed down). The main loop of the VLab goes through several things including collecting the spikes, rendering the graphics, accepting user’s inputs (mouse, keyboard and joystick, etc.), communicating with Open Inventor, as well as executing the task file. Any event will be saved together with the receiving moment in ms to the data buffer, except neural spike with which the time saved is the occurring time. The faster is the computer, the shorter is the delay between occurring moment and the receiving moment of an event, and therefore, the more accurate is the timing of the entire system. A typical time period consumed by one loop is about 1~2 ms in a Pentium III PC. This gives an accuracy of 2ms for the VLab system. In order to maintain the timing accuracy, operations consuming too much time are not suitable within the task file.

The syntax supported by the task file compiler includes:

a. Constant values including Arabic numbers and symbolic constant values, but not string values. A string within a pair of double quote will be treated as a name of graphic or behavior parameter.

b. Variables, “int” only, can be “static” or not.

c. Arithmetic operators: + (plus), - (minus), * (multiple), / (divide), += (plus equal), -= (minus equal), *= (multiple equal), /= (divide equal), ++ (increase by 1), -- (decrease by 1), - (negative), = (assignment).

d. Logic operators: > (greater than), < (less than), == (equal), >= (greater than or equal), <= (less than or equal), != (not equal), && (and), || (or), ! (not).

e. Flow control: "if", "else", "else if", "?:".

f. Functions. Functions should be defined before being used. Only empty parameter list is supported. Parameters should be transferred by means of defining global variables. Every function should return an integer value (int).

g. Comments. Anything preceded with “//” in the same line is treated as comments and will be ignored by the parser. “/*” and “*/” are the open and close signs for block comments. Anything between them will also be ignored by the parser.

A task file usually has a “c” extension. It is specified as “Command File” in the “Test” panel.

a. Symbolic constant values

name	description	value
TRUE	1	1
FALSE	0	0
ITI	Task state indicates interval between trials.	0
ITI2	Task state indicates time out.	1
TARGET_ON	Task state indicates that the target is on.	2
TARGET_FLIP	Task state indicates that the target is flipped.	4
TARGET_OFF	Task state indicates that the target is off.	5
FOREP	Task state indicates fore period (onset of stimulus).	3
FOREP1	Task state	8
FOREP2	Task state	9
FOREP3	Task state	10
FOREP4	Task state	11
FOREP5	Task state	12
FOREP6	Task state	13
FOREP7	Task state	14
FOREP8	Task state	15
FOREP9	Task state	16
FOREP10	Task state	17
ISI	Task state: inter-stimulus interval.	18
ISI2	Task state: inter-stimulus time out.	19
USER_STATE	Task state	100
KEY_CENTER	Key state indicates that the key is in the center (released).	
KEY_BACK	Key state indicates that the key is pulled backward.	
KEY_FORTH	Key state indicates that the key is pushed forward.	
KEY_LEFT	Key state indicates that the key is moved to the left.	
KEY_RIGHT	Key state indicates that the key is moved to the right.	
TRIAL_NONE	Trial state indicates that there is no trial.	0
TRIAL_EARLY	Trial state indicates that the trial is ended prematurely.	1
TRIAL_HIT	Trial state indicates that the subject successfully performed the task.	2
TRIAL_MISS	Trial state indicates that the subject made a wrong response.	3
TRIAL_CONTINUE	Trial state indicates that the trial is on going.	4
TRIAL_END	Trial state indicates that the trial is ended.	5
TRIAL_FAIL	Trial state indicates that the subject unsuccessfully performed the task. The trial will be discarded.	6

TRIAL_REPEAT	Trial state: current trial needs to be repeated.	7
VK_CONTROL	Key code: both control keys.	
VK_LCONTROL	Key code: left control key.	
VK_RCONTROL	Key code: right control key.	
VK_SHIFT	Key code: both shift keys.	
VK_LSHIFT	Key code: left shift key.	
VK_RSHIFT	Key code: right shift key.	
VK_LEFT	Key code: left arrow.	
VK_RIGHT	Key code: right arrow.	
VK_UP	Key code: up arrow.	
VK_DOWN	Key code: down arrow.	
FIG_ON	Indicates onset of figure (not used in task file)	6
FIG_OFF	Indicates offset of figure (not used in task file)	7

b. Internal variables

nCurrentTime	System time in ms. It goes from 0 to 2^{32}-1 and again from 0. Approximately 49 days in one cycle. Assigned by VLAB program.
nKeyPos	Key position. 67: center; 70: forward; 66: backward; 76: left; 82: right.
nTaskState	Current state of the task, ITI, TARGET_ON, or FOREP, etc.

c. Internal functions

GetTime()	Return system time in ms by Windows API call. More accurate than nCurrentTime but more expensive to use.
OnFP()	Return 1 if the eye position is within the fixation window, otherwise return 0.
GetKeyPos()	Return the key position. It is same as what nKeyPos indicates.
TurnOnTarget()	Turn on the target.
TurnOffTarget()	Turn off the target.
FlipTarget()	Flip the target in orientation discrimination task.
TurnOnFig()	Turn on the testing figure. The render delay time of the figure is compensated.
TurnOffFig()	Turn off the figure.
DoEarlyAct()	Do action for a trial ended prematurely. Counter of total trials increased by one.
DoHitAct()	Do hit action for a correctly performed trial. Counter of total trials and hit trials are both increased by one. The subject is rewarded at the same time.
DoMissAct()	Increase total trial number by one with time out.
Recording()	If recording in VLAB is on going, it returns 1, otherwise 0.
GetFPH_Min()	Returns horizontal eye movement data in minute of arc.

GetFPV_Min()	Returns vertical eye movement data in minute of arc.
GetGraphParam("param")	Returns the value of graph parameter "param".
GetGraphParam10("param")	Returns 10 times up-scaled value of graph parameter "param".
GetBehavParam("param")	Returns the value of behavior parameter "param".
GetBehavParam10("param")	Returns 10 times up-scaled value of behavior parameter "param".
SaveKeyState(param)	Store key state param to the data buffer.
sin_1000(param)	Returns 1000 times up-scaled sine of param.
cos_1000(param)	Returns 1000 times up-scaled cosine of param.
tan_1000(param)	Returns 1000 times up-scaled tangent of param.
asin_1000(param)	Returns 1000 times up-scaled arcsine of param.
acos_1000(param)	Returns 1000 times up-scaled arccosine of param.
atan_1000(param)	Returns 1000 times up-scaled arctangent of param.
abs(param)	Returns absolute value of param.
DisplayVariable(param)	Print the value of param in the syslog window.
KeyDown(param)	Returns 1 if key param is pressing down, otherwise returns 0.
Rand(p1, p2)	Returns a random value between p1 and p2.
ReadAICh_mv(ch, g)	Returns value from A/D convert card channel ch with gain g in mini-volt.
SendUserMsg("p1", p2)	Send "p1" in value of p2 to Inventor.
SetGraphParam("p1", p2)	Same as SendUserMsg.
SetGraphParam10("p1", p2)	Send "p1" in 10 times up-scaled value of p2.
FPCircle(x, y, r, c)	Draw a circle at position (x, y) with radius r and color c on the eye movement display panel.
FPRectangle(x, y, w, h, c)	Draw a rectangle at position (x, y) with width w, height h, and color c on the eye movement display panel.
SaveResponse(s, t, v1, v2, v3)	Store task state s, response time t together with 3 additional values v1, v2, and v3 to the data buffer.

3. Command file

A command file consists of conditions to be tested by the experiment. It is specified as "Command File" on the "Test" panel. It usually has a "vlc" extention.

The VLab can test a single scalar parameter without using a command file by clear the "Use Command file" check mark, and specify "Param to Test", "Begin", "End" and "Step".

If more than one parameter or a vector parameter will be tested, a command file is necessary to specify the testing conditions. It is a text file with one condition per line. Each line should contain the same parameter list in the same order. Use spaces to separate parameter name and its value. Use comma to separate different parameters. Blank lines or lines beginning with "#" sign are ignored. The same condition can be repeated to balance other conditions.

One or multiple commands can be executed per testing trial at the begin of the inter trial interval (ITI) or inter stimulus interval (ISI) randomly or sequentially according to the "Random" check mark in the “Test” panel. Randomization will be restricted within the command file if “balanced” check mark is checked or expanded to repetitions (specified by “Accum Times”) if unchecked. Conditions can be grouped together by specifying “xxx commands in a group by first xxx parameters” with “balanced” check mark being unchecked.

4. Inventor script file

An Inventor script file has a “iv” extension. It is specified as “Open Inventor File” on the “Test” panel. Its path should be a real unix file path it resides.

In a Inventor script file, all “GlobalField” with tags begin with “UF-“ or “CF-“ are treated as graphics parameters and their names and initial values are transferred to PC upon loading. They will be listed in the “GraphParam” panel at the same order as they appear in the script file. The parameter list can be separated by inserting “#@” in the script file.

5. Launching VLab

The first time to run VLab should be done by an user with administrator’s privilege to register the VLab Object in the OLE database upon the system registry.

After VLab is started, check “SysLog” panel. “vstat.dll” should be loaded for the statistical functions to work. The experiment settings shown in “Test” panel and behavior parameters shown in “BehavParam” panel should have already been loaded from the user’s registry at this moment. Now click “Pause<=>Resume” button. It will start the Open Inventor program in the unix box specified by “SGI” behavior parameter. “GraphParam” will be filled with parameters from the Inventor script file. Check “SysLog” panel for any errors related to loading of the Inventor script file and task file. “nidaq32.dll” and “nidex32.dll” should be properly loaded for the joystick to work. Device driver “SPIKE” is for collecting the neural spikes, while device driver “PCDIO” is for awarding. They are necessary for animal experiment. Serial port specified by behavior parameter “SerialPort” should available to accept eye movement data from ISCAN.

If the VLab warns that the data directory does not exist, close the VLab, create the data directory and start VLab again. The data directory is specified by behavior parameter “DataPath”.

Specify the “Accum Times” in “Test” Panel and click “Start” button. The experiment is now started. The responses of the subject are displayed on the “Resp” panel. Turn off the control panel by click “Control Panel” button in the tool bar. The online plots will be displayed on the left panel of the main window. The testing condition together with spikes and eye movement curves are displayed on the upper right panel trial by trial. The real time fixating location of the subject is displayed on the lower right panel. Click “Save” button to save the data after all conditions have been tested.

VLab allows recording of neural data of multiple channels simultaneously. The eight big buttons in the channel bar indicate the status of the channels. First number on the button indicates the channel number from 0 to 7. The second number indicates the electrode number assigned to that channel. When CapsLock is enabled, this number can be increased by holding down “Shift” key while press F8, F1~F7 keys for each channel, and decreased by holding down “Ctrl” key wile press related F keys. Without CapsLock, the same key combination will increase or decrease the cell name indicated by the lower case character on the buttons. The aster character indicates that the channel is active. It can be toggled by pressing F keys directly. “@” key indicates the main channel. Information of this channel will be used to form the data file name. The main channel can be changed by holding down Shift and Ctrl keys simultaneously while pressing corresponding F key. Clicking one of the eight buttons down will let the graph view and trial view to display the data of that channel, otherwise, the graph view will display average data of all channels, and the trial view will display all eight channels of spike trains. See below for definitions of graph view and trial view.

6. Test settings

A new test is created by clicking “New Test” button on the “Test” panel. Input a name for the new test in the pop up dialog box. The ExpParams (those displayed on “Test” panel), initial GraphParams and intial BehavParams are copied from old test and should be modified to represent the difference from the old. Initial GraphParam and BehavParam are those custom parameter values for this specific test. They can be added by selecting a parameter in the main list and clicking the “Add to init param” button, or removed by selecting it from the initial test parameters list and clicking “Del init param” button. When a specific test is selected from “Test Type” on the “Test” panel, its associated ExpParams, initial GraphParams and BehavParams are loaded. Default GraphParams and BehavParams, which are modified by the initial params, are restored when another test type is selected, and new set of initial params for the later test type are then loaded. Unwanted set type can be deleted by clicking “Delete Test” button.

Besides those mentioned in this manual, rest of the behavior parameters can be added or deleted by clicking “Add behav param” or “Del behav param” buttons. And all parameters on the three control panels can be used in the task file.

Modification of the test settings are saved to the registry by selecting “Configure—>Save parameters” from the menu, or discarded by selecting “Configure—>Reset parameters”.

7. Manipulate displays of the results

a. Graph view

The display panel on the left shows the neuronal response curves, bar charts, bubble plots, and spike train in raster plots as well as behavior responses in histograms. This is the graph view.

Right click at locations other than the hotspots will bring a menu. From the pop up menu we can modify the properties (page layout, figure size, pen width, grid tick length, axis labels and ranges, etc.), font, plot type (bar, bubble, line, roof — a plot type optimized for roof edge experiment, edge — a plot type optimized for step edge experiment, etc.). From the pop up menu, we can also toggle the figure guide lines, cross hair cursor, copy the data or graphics to the clipboard, or export the data to a file in different formats including DBF and CSV and print the graph as well. The graphics copied to the clipboard can be pasted into other applications such as MS-Word, PowerPoint or CorelDraw as VLab Object, Picture (Metafile), or Picture (Enhanced Metafile, non-preferred, since it scales down the size of the graphics). VLab Object contains both the graphics and the data.

When the plot type is in bar or line, mouse cursor can be used to select a value of the parameter shown along the horizontal axis and apply it to the GraphParam by choose “Set Optimal Value” from the pop up menu.

When right click at the two axis titles or the figure title, menus will pop up for us to select the parameter to display. The extra choice “Time (ms)” in the horizontal axis title pop up menu is for selecting the raster display of the spike train. “Spikes/S” in the vertical axis pop up menu is for selecting the bar plot, while “Depth matched” for the histogram plot of joystick settings. Behavior parameters “Param2ShowX”, “Param2ShowY”, and “Param2ShowZ” will reflect the changes immediately. Those parameters can also be modified directly to change the plots.

New variables can be calculated from existing testing parameters by choosing “Data—>Transform...”. Type in variable name to the left and expression to the right in the pop up dialogue box. Then hit “OK”. New variable is available immediately for drawing as horizontal, vertical or title variable. If its name matches one of the graph or behavior parameter, it will be treated as a testing parameter when the experiment is started again. Transformation equations will be saved together with the data and available upon reopen. The new variables will be available for the online statistics as well. If you choose to export the data, the new variables will appear in the testing parameter list. The disadvantage of the transformation is that the testing parameter list is not the same as the original. Thus it will cause problem when you want to export different data files of the same test type into a single CSV or DBF file. Therefore, it will be better not to save the data file again after doing transformation.

Supported syntax in the transform expression includes:

Values: numeric (both integer and floating) or string (in double quote).

Variables: The name should match one of the testing parameters.

Arithmetic operators: + (plus), - (minus), * (multiple), / (divide), - (negative).

Logical operators: > (greater than), < (less than), ! (not), >= (greater than or equal), <= (less than or equal), == (equal), != (not equal), && (and), || (or).

Functions: sin (sine), cos (cosine), tan (tangent), asin (arc sine), acos (arc cosine), atan (arc tangent).

Switch: condition ? choice_for_true : choice_for_false.

Grouping: (,).

When the experiment is running, i.e. “Start” button on “Test” panel is pressed, the menus can not be opened by right clicking. At this moment, dragging the axis and figure titles will do the trick.

Dragging the axis or the figure guide lines will change the figure size and margin. Dragging the vertical axis label at the upper end will change the range of the vertical axis. All the hot spots are only available within the first graph.

Choose one of the macros from “Data —> Macro” will enable preset transformations and plot type and properties optimized for:

Random dot roof edge experiment (Draw RDRoof);

Random dot step edge experiment (Draw RDEdge);

Uniform solid roof edge experiment (Draw SolidRoof);

Border ownership and stereoscopic edge experiment (Draw BorderOwnership).

The spike rate for each trial is calculated from stimulus onset (when TurnOnFig is called in the task file) till the time length specified by behavior parameter “AnalysisTime”. Trials shorter than this time period are given -1 as the spike rate, and therefore excluded from plots display and online statistical analysis.

b. Trial view

The trial view is the upper right display panel. It displays the results trial by trial. Its layout can also be changed by dragging the lines. Its right clicking pop up menu allows us to zoom vertically (FP Range) or horizontally (Horizontal Zoom), or to change the display font. When the input cursor is in the comment edit box to right of the tool bar, press key F8, F1~F7 will enable/disable collected trial in channel 0, 1~7. Disabled trial will be displayed in gray. Up and Down keys are used to move from trial to trial. Comments can be input for specified trial at this moment. Be aware that press key F8, F1~F7 will enable/disable the channel for upcoming trials if input cursor is not in the comment window.

c. Eye movement view

The eye movement view is located at the lower right display panel. It displays the fixating locations of the experimental subject. Right clicking pop up menu allows changing the zoom level (FP Range), toggling the grid, to select different display font, or to initiate the calibration routine (same as the "Calibrate" button in the tool bar). Dragging the axis labels or rotating the mouse wheel will change zoom level as well. The coordinate is in degree of visual angle.

Optionally, squares or circles can be drawn in selected colors from the task file through FPRectangle() or FPCircle() function call.

Calibration should be performed at the beginning of each experimental session. Once the calibration routine has been initiated, it will move the fixation target through five locations including center, left, right, bottom and top. The parameter names, whose values control the fixation target locations, are specified by behavior parameters "FPParamH" and "FPParamV". The routine will collect number of frames of eye movement data specified by "FPCalFrames" for each target locations during the "FOREP" task state (see symbolic constant value of task file) when fixating location is within a circular fixation window whose radius is specified by "FPWinRad". The amplitude of the movement of the target location from the center is specified by "FPCalDevH" and "FPCalDevV" in minute of arc unit. "UnitsPerDeg" designates how many units in the Open Inventor display equals one degree in visual angle. "FPUpdate" indicates dynamic update of the origin (1) or not (0). Multiple runs of the routine enable more accurate calibration. Use a bigger fixation window for the first run to ease the calibration, then a smaller one for higher accuracy. In the behavior parameter list, "H0" and "V0" are the calibrated origin of the eye movement, while "DtoAH" and "DtoAV" indicate the digital (from ISCAN) to analog (in degree of visual angle) ratio. The "DtoAH2" and "DtoAV2" compensate the rotation of the ISCAN camera. Frame period of the ISCAN signal is specified by "EyePointBin" in ms.

8. Data file

a. Convention of the file name

A data file will be created when the "Save" button of the "Test" panel is clicked or the menu item "File—>Save or Save As" is chosen. The file name is formed automatically according to information of the experiment. The first two character of the file name are from the "SubjectName" in the behavior parameter list. The third character of the file name indicates the hole number (HoleNum) inside the recording chamber wile the fourth character indicates the position number (PosNum) with that hole. For psychophysical experiment, these two characters can be set to arbitrary value, such as "aa". The name of the test type comes next. Then is the repetition number in two digits. After the file has been saved, the repetition number will be increased by one if the "New File" button is clicked. When select a different test type, if the same test has already been done for the same cell, the next repetition number will be used to form the file name. Finally, the extension of ".vlb" is appended to indicate a VLab data file: double clicking the data file will open it with VLab.

b. Browsing through the files

VLab files can be browsed through by typing full path names with wild character in the "File" edit box to left of the tool bar, then click the ">" or ">|" buttons. The files retrieved are ordered by creation time. Click ">" button go to next file, while click "<" button go to previous file. Click ">|" to last file, "|<" to first file. The file names can also be retrieved from ODBC database. The syntax is as follows:

odbc://username:password@data_source_name:sql_query_string

For example:

odbc://tiger:scott@mydatabase:select filename from test where testname='Orient'

Here "tiger" is database username with "scott" as the password. "mydatabase" is the name of the ODBC data source. "filename" is a column holding full pathname to the data file in table "test". "testname" is another column in the table holding the name of the test type. This example will retrieve all the files with test type name of "Orient". If the SQL statement is getting too long to edit, double click it will bring an editor box for easy input.

Uncheck "View—>Load Format", then VLab will ignore the display format for the graph view from the opened the file. This is useful if you don't want to be disturbed by the different display format of each opened file.

If the Open Inventor program is running at the same time, browsing through each file will also load the Inventor script file accordingly. If the task file and the Inventor script file are the same as before, you will discover that the display of the Open Inventor is exactly the same as the moment before you saved the data file. By opening an old data file, you can restore the old settings for a particular test. Just by clicking "Start" following "New File" button, you can do the same experiment with exactly the same settings again.

In order to browse through the data files properly or to repeat the same test with the same settings again at a later time, it is important not to modify, rename or delete the original task file and Inventor script file. If modification is necessary, at least you need to keep it back ward compatible, and keep a change log together with those files. Command file is always saved together with the VLab data file. By clicking at "View" button besides "Command File" on the "Test" panel after open an data file, you could see the original command file used. Start the experiment without click "New File" first, the same command as in the data file will be used again. Otherwise, command file specified by the edit box will be used. At this moment, one need to be aware that the command file might have been changed since last time when it was saved with the data file.

c. Open in MatLab

The VLab data file can also be opened in MatLab by a mex function "ReadVLabData.dll" with following format:

[nTestType, szComment, nSpikeChannel, aTestType, aExpParam, aBehavParam,
aGraphParam, aTestSeries, aTrial, aFOREP, aResponseTime, aTrialStatus, aDepth2Match,
aCursorDepth, ucMajorChannel, nCellNum, ucElectrodeNum, nElectrodeDepth] =

ReadVLabData (filename);

9. Export of data

a. Export of trials

The trial by trial data can be exported by right clicking at the graph view panel and choose "Export..." from the pop up menu. Then in the pop up dialogue box, input the full path of the files to export in "Input Files". Wild card characters can be used here. Next specify the output file name (for output to single file) or directory in "Output file, dir or table". Then choose the output file format. CSV is a comma delimited text file. First line of the file lists the field names. Fields are separated by comma. Text field is quoted by double quotes. DBF is a native file format of DBase III, which also carrys the information about data type and width of the fields.

If you want all the data from selected files go to a single file, check “output to single file” and specify a filename rather than a directory name in “Output file, dir or table”. Only files with the same testing parameters, which are in the same order, can be exported to a single file properly.

a) Basic event

Basic events include fields of CellID, Repeat, TrialNum, Set, Item, Testing parameter list, Status, RTime, V1, V2, V3, Resp, Resp0, Resp1, Resp2, Resp3, Resp4, Resp5, Resp6, Resp7, ActiveCh, MainCh.

CellID is an unique six-character name for the cell under testing.

Repeat is the last part in the file name just before the “.vlb” extension.

TrialNum is the sequential number of that trial starting from 0.

Set is the repeat number of the command for that trial. It starts from 0.

Item is the command number for that trial. It also starts from 0.

Testing parameter list is the parameter list from the command file.

Status is the result of the behavior task. It can be 1—TRIAL_EARLY, 2—TRIAL_HIT, or 3—TRIAL_MISS. This is the first parameter being saved by “SaveResponse” in task file.

RTime is the second parameter being saved by “SaveResponse” in task file.

V1~3 are the third through fifth parameters being saved by “SaveResponse” in task file.

Resp is the neural response in spikes/sec of the main channel. Negative 1 indicates bad trial.

Resp0 ~ Resp7 are the neural responses of channel 0 ~ 7. Negative 1 indicates bad trial or channel is not active.

ActiveCh is a 8-bit value. Each bit indicates the status of corresponding channel (1—active, 0—non-active).

MainCh is a decimal number which indicates the main channel.

b) Spike events

Spike events include CellID, Repeat, TrialNum, SpikeEvent, SpikeTime.

First three fields are the same as those in basic events.

SpikeEvent is a 16-bit value. A bit value of 1 indicates that there is a spike in the corresponding channel. Negative 1 indicates the end of the trial.

SpikeTime is a floating value in ms which indicates the moment when SpikeEvent occurs. Its accuracy is 0.1 ms. Time 0 is the moment when “TurnOnFig” is called in the task file, and it is synchronized with the retrace of the display monitor of the Open Inventor.

c) Eye movement data

Eye movement data consist of CellID, Repeat, TrialNum, FPX, FPY, FPTime. FPX and FPY are the coordinates of the fixating location at time of FPTime in ms.

d) Joystick readings

The data table has fields of CellID, Repeat, TrialNum, KeyAction, KeyTime. KeyAction is the one saved by calling SaveKeyState from the task file at time of KeyTime.

b. Export of parameters

The GraphParam and BehavParam list can be exported by choosing “Data—>Export Param...” from the main menu. In the pop up dialogue box, type the full path of the file, from which the parameter will be exported, at “Input files”. Use wild characters “*” or “?” to select multiple files. Then type in the output file path at “Output file”. Now specify the parameters to export at “Param to export”. Separate multiple parameters by comma. After choose “Graph Param” or “Behav Param”, hit “OK”.

The format of the output parameter file is the same as that of CSV file. The first line lists the field names. Fields are separated by comma. The first three columns are CellID, TestName, Repeat. TestName is the name of the test type. The rest columns are the parameter list you just typed in the export dialogue box.

10. Online statistics

Online statistical routines have been implemented by calling MatLab math library. On “Statistcs” control panel, the field names of the internal data table are listed to the left. Their meanings have been explained in the previous session.

Select one of the variables, typically Resp, as dependent variable by clicking “Set Depend”. Pick several factors as you wish by selecting them first on the left list then clicking “Add Factor”. You can remove unwanted one by clicking “Remove”. You can select sub set of the data by checking “Cases” and specify the condition to the right. Interaction levels are specified by “Level”.

After choosing a function from the “Function” list, for example “anova”, then click “Estimate”. The result will be displayed on the panel.

Since the statistics routines consume a lot of CPU time, it is not appropriate to run them during an ongoing trial. They should be run between trials at the inter trial interval (ITI) or at time when the test is not running.

11. Play back of the trials

After the experiment, the data files can be opened. All trials can be played back in a similar way as doing the experiment. When Open Inventor program is running, from “Play back” panel, set the parameters and their values dealing the events, then click “Start”.

12. Installation and setup

a. Device drivers for Windows

Device drivers “SPIKE” and “PCDIO” need to be installed before VLab can collect neural spikes and give reward for the task. The driver files “spike.sys” and “pcdio.sys” should be copied into system32\drivers directory of Windows, and following registry keys should be installed as well:

```
\registry\machine\system\currentcontrolset\services\Spike
    Type = REG_DWORD 0x00000001
    Start = REG_DWORD 0x00000002
    Group = Extended base
```

ErrorControl = REG_DWORD 0x00000001

\registry\machine\system\currentcontrolset\services\Pcdio

Type = REG_DWORD 0x00000001

Start = REG_DWORD 0x00000002

Group = Extended base

ErrorControl = REG_DWORD 0x00000001

The SPIKE device driver is for a home made data collection card for ISA bus. This card has 16 input lines for collecting 16 channels of spike event simultaneously. It has an internal buffer which can hold up to 20KB data. Whenever a spike appears on the 16 input lines, 2 Bytes data will be saved at the buffer together with the spike occurring time in tens of ms. At meanwhile, IRQ 7 of the host computer is triggered. The event port address is 0x300, which is also the control port address. Time port address is 0x302, while status port address is 0x304.

The PCDIO device driver is for a PC-DIO-96 card from National Instrument. This card is a 8255A based I/O card for ISA bus. The base port address used is 0x180. IRQ 5 is its interrupt request line.

A PCI-6023E analog to digital (A/D) convert card from National Instrument is used to read the joystick settings. The device driver from the manufacture is used to control this card.

b. Serial port

A spare serial communication port is need for VLab to read eye movement data from ISCAN. Which COM port to use is specified by behavior parameter “SerialPort” with its baud rate specified by “BaudRate”. The ISCAN computer should send out exactly two values through the serial port, one for horizontal position, another for vertical position. The position data can be pupil center position, position of the corneal reflection point, or the difference between them. The time period between two subsequent frames is specified by “EyePointBin” in ms.

c. Open Inventor program

Both Iris and Linux version of the Open Inventor can be obtained from SGI free of charge. At least its run time component should be installed for the VLab Inventor program to run properly. If custom nodes are used, the environment variable LD_LIBRARY_PATH should be set to include the directory containing these nodes. Environment variable DISPLAY should be set to local host as:

```
export DISPLAY=:0.0
```

Open GL and Motif libraries should also be installed properly. Xdm (Iris) or XFree86 (Linux) should be running before starting the VLab. In most of cases, an user should login interactively to the xwindow before the display on the local host can be opened by VLab. Graph parameter BCOL will be intercepted by the VLab Inventor program and used to set the background color of the display.

d. Network setup

Network adapters with speed of at least 100 MPS should be used in both PC and Unix box to insure that the timing accuracy is at 1-2 ms level. Two TCP ports are used for communication between PC and Unix. Port 20000 is listened by inetd (Iris) or xinetd (Linux) in the unix box to initiate the vlabd. Vlabd will in turn launch vlabx, which is an Open Inventor program. Vlabx will be listening to port 22222 and accept connection from PC to exchange controlling and acknowledge messages. A typical vlabd script in SGI Iris system will be:

```
#!/bin/sh

# no more than one copy of vlabx running simultaneously.
temp=`ps -u vlabuser -o comm | grep vlabx`
if [[ -z $temp ]]
then
    # open the display of the local host
    export DISPLAY=:0.0
    # turn off screen saver
    /usr/bin/X11/xset s reset
    /usr/bin/X11/xset s off
    # include directory containing custom nodes
    export LD_LIBRARY_PATH=/home/vlabuser/vlab
    cd /home/vlabuser/vlab
    # launch vlabx
    ./vlabx
    # after the experiment, turn on the screen saver
    /usr/bin/X11/xset s on
    /usr/bin/X11/xset s reset
    # and activate the screener saver immediately
    /usr/bin/X11/xset s activate
fi
```

The vlabd is launched through inetd as appending the following line to /etc/inetd.conf:

```
vlabd stream tcp nowait vlabuser /home/vlabuser/vlab/vlabd vlabd
```

and the following line to /etc/services:

```
vlabd           20000/tcp
```

If your system uses xinetd rather than inetd, then a file named vlabd as follows should be created in directory /etc/xinetd.d:

```
service vlabd
{
    socket_type        = stream
    wait               = no
    user               = vlabuser
    server             = /home/vlabuser/vlab/vlabd
    log_on_success     += HOST DURATION
    log_on_failure     += HOST
    disable            = no
}
```

Both vlabd and vlabx should have “x” bit set in their file mode by command:

```
chmod a+x vlab*
```

13. Miscellaneous tips

Comments can be input into the large edit box on the “Test” panel while the experiment is on going. The depth of the electrodes can be input on “Electrode” panel and saved to the data file. When “Pause/Resume” is pressed down, the task file will start running. For psychophysics experiment, you would like to return from the task file immediately if the test has not been started yet. This is done by checking the return value of function “Recording()”. Returning value 1 means the test is running. For neural recording, you would like the subject to keep doing the task while the test has not been started yet.

When “Pause/Resume” is on pressed down position, the thread priority of VLab is set to high. If you want to run other applications in the same computer, for example to make a new command file, you would like to toggle “Pause/Resume” button to up position to set VLab’s thread priority back to normal to give your editor more CPU time.

The first item displayed on the status bar is accumulated times of data in term of command file. Click “Discard” button on the “Test” panel will discard trials from the last repetition of the command file. This is useful if the cell was dying or killed recently.

The second item displayed on the status bar is the size of the command file or the total number of the testing conditions including repeated items. “S” on the status bar means the value of the horizontal axis at the location of the mouse tip, “R” value of vertical axis under mouse tip in spikes/s. “H” and “V” indicate the receptive field position, merely the value of graph parameters “POSH” and “POSV”. The next item is the subject name. It followed by key position (level key for doing the task, different from joystick), task state, and percentage of correct trials.

The receptive field position is designated by POSH and POSV (offset from the fixation target), OR (optimal orientation), and POSX and POSY (offset in coordinates rotated by OR). Typical routine for fine tuning the receptive field position is to record POSX and POSY profiles, choose the peak position from the profile curves. Then click “XY to HV” button on the “GraphParam” panel to apply POSX and POSV to POSH and POSV, and reset POSX and POSY to zeros.

Set behavior parameter “SwitchJoystick” to 1 enable you to take over control of the subject joystick by experimenter’s joystick, which allows you to perform the depth matching task without going over to the subject room.

14. Index