NEWTRACK - reading experiments with the Fourward DPI eyetracker on modern PCs

Eric Auer <eric@coli.uni-sb.de>

13th April 2005

The NEWTRACK term project is intended to provide a modern alternative to the well-known PCEXPT software by Charles Clifton for Fourward Technologies DPI eyetrackers. While supporting modern and cost-effective PC hardware, it does not support all script language and post processing features of PCEXPT. Instead, a subset of the script language of the Saarbrücken READING software for SMI / SR Research EyeLink eyetrackers has been implemented. The log files use a text format which is modelled after the text messages produced by READING, so we can re-use the READING analyzers to analyze NEWTRACK log files. Both the dependency on now hard to get old hardware and the possibility to reduce complexity by using modern standard PC features instead of specialized drivers motivated me to write a completely new eyetracking software instead of trying to improve PCEXPT. The result, NEWTRACK, can be freely reused under the terms of the GNU public license - NEWTRACK does not contain code from PCEXPT.

Contents

1 Introduction 3

2 From PCEXPT to NEWTRACK 3

2.1 Timing, memory and CPU 5

2.2 Dual display support 6

2.3 Eyetracker and buttonbox connection 9

2.4 Calibration 12

3 Compiling and portability 12

4 File formats 13

4.1 Scripting language 13

4.2 Log file format 15

5 Appendix 16

5.1 References 16

5.2 Sample stimulus script file 18

5.3 Sample log file 19

5.4 How to initialize the dual display system in DOS 21

5.5 Printer port data aquisition device 23

5.6 NEWTRACK usage 26

1 Introduction

Modern eyetracking devices allow to sample the gaze position of a human subject several 100 times per second, with an accuracy of up to 0.01 degrees. With a typical measurement range of +/- 20 degrees, this allows researchers to know exactly where and when the subject looked at items on a computer screen, ideally with pixel precision coordinates. Calibration problems and uncontrolled movement of the subject, even when fixed to the sensor with a head band (SMI) or using a bite bar (DPI) will often make it impossible to reach such accuracy, but still it is realistic to assume that it is possible to decide on which letter on a displayed text the subject looked at a certain display frame, even when CRT screens with high refresh rates are used.

Both the SMI and the Fourward design rely heavily on hardware to achieve their speed and resolution. SMI is using high speed video cameras along with a signal processing board (for ISA or PCI bus), while the DPI eyetracker sold by Warren Ward / Fourward Technologies is implemented as an entirely standalone device. EyeLink software, originally for Windows and DOS, now supports Linux and MacOS to some degree. The DPI system, on the other hand, does not come with any software at all. The hardware - a clever system of servo control loops which use an infrared light beam to track eye movements - provides the measured coordinates as analogous voltages (-5 Volts to +5 Volts), along with some TTL (0 or 5 Volts, representing a boolean value) compatible signals for e.g. track loss and blink events. The Fourward Technologies homepage lists about 40 sites which have a Fourward DPI eyetracker installed.

2 From PCEXPT to NEWTRACK

Both NEWTRACK and PCEXPT can be used for eyetracking experiments with Fourward DPI eyetrackers. The eyetracker implements all necessary tracking algorithms in hardware, so the PC only has to read out a pair of voltages to know the (uncalibrated) eye coordinates. Still, PCEXPT is a very complex piece of software and relies on specialized hardware which will often be no longer available in 2004. The system uses dual screen output, one screen for the operator and one for stimulus presentation, and it employs special data acquisition hardware to read out the coordinates from the eyetracker. PCEXPT is maintained by Charles Clifton, Jr, Dept. of Psychology, University of Massachusetts at Amherst, USA.

Both the dual screen output and the coordinate input are written for hardware which is optimized to work on old and slow computers. The dual screen output requires an ISA monochrome or Hercules graphics card: Modern PCs no longer have ISA slots anymore – unless they are expensive ones for industry use, where replacing custom ISA boards by newly developed alternatives is often not an option. The data acquisition hardware, used for reading the voltages from the eyetracker as well as for timestamping and for the button box signal inputs, is designed towards use with slow computers and quite advanced and expensive. The original hardware requires one or two ISA slots for that, but there are now versions for PCI slots as well. Most extension boards for a modern PC will actually be for PCI slots. However, the PCI version of the data acquisition hardware is even more expensive – it is sold as a compatible replacement for the classic ISA board(s).

For using the PCEXPT software, the DPI (Dual-Purkinje Image) eyetracker is connected to a PC with both a Hercules or monochrome graphics card and a normal color graphics card. There is a buttonbox (normally 4 buttons: Yes, No, Calibrate, Next) for subject and operator, which is normally connected to a combined ISA digital I/O and timer/counter (used for timestamps and delays) board. The X/Y signals are usually measured by an high-end 12bit precision A/D converter board (e.g. one 250kSample/sec converter combined with a multiplexer that connects it to up to 16 input lines in a round-robin way and a DMA system which copies the sample data into the memory of the PC without requiring the main processor to collect it from the board. The 12 bit precision lets you use a 4096 by 4096 coordinate grid if the input signals are adjusted to full range amplitudes. This still has to be corrected for linearity and calibrated to each subject, but you will be able to read the gaze position from the DPI at almost screen pixel level even if the screen is much smaller than the maximum viewing angle of the DPI.

PCEXPT contains several no longer maintained drivers for now exotic hardware, but it also has only limited abilities to work with modern hardware, apart from the abovementioned ISA bus requirement. One of the most important points is that a plain 16 bit compiler is used, so only the first megabyte of the system RAM can be used to store data. There are no drivers to use extended or expanded memory in PCEXPT.

In spite of the limitations of 16 bit DOS operation, PCEXPT includes many features, added by several different people and often not well integrated into the whole system. A few examples are support for Korean text display, audio output, image presentation, stimulus presentation with masking, and so on. PCEXPT even contains drivers to display text with unusually high refresh rates on Hercules, Tandy, EGA and VGA graphics cards as well as on an oscilloscope (!), which is used as a fast vector graphics display. All those additions make the PCEXPT system hard to maintain and hard to modify. Most of the hardware tuning is obsolete with modern hardware. For example you can use your VESA VBE3 compatible VGA BIOS (or add the free SciTech Display Doctor / UNIVBE VBE3 driver software) and the free VBEHz tool to select video modes with high refresh rate on standard VGA graphics hardware now. So there is no refresh rate tuning done inside NEWTRACK itself.

PCEXPT comes with many driver test programs, data analysis programs and both built-in and standalone support for special stimulus processing like controlled randomization of trial order. I found many of those features being replaced or not in use now: You analyze the experiment results with your favourite statistics software (e.g. the OpenStat system or something written in the Octave language, which is similar to Mathlab). For stimulus shuffling, you will often use a separate Perl script or similar tools. Some of the features in PCEXPT have to cope with the memory limitations of a 16 bit environment and cannot be used for some experiments that way.

Rather than using a 32 bit operating system for NEWTRACK, only a 32 bit compiler is used, while the system itself runs in plain DOS. Running 32 bit software in DOS is not a problem, and the operating system itself only has to provide simple services like file access and keyboard / mouse drivers for NEWTRACK. The hardware drivers in NEWTRACK support both the most frequently used PCEXPT compatible hardware as well as some widely available modern hardware, as discussed in more detail below.

Only for the data acquisition hardware, you still have to choose between several evils: Either use PCEXPT compatible but expensive hardware or build a custom data acquisition system and connect it to the printer port. The third choice would be using the joystick port for the button box and medium price fast serial port A/D converters, but then you have to adjust the drivers of NEWTRACK to support that third option. NEWTRACK is able to run in a fallback mode on a PC with only one screen, mouse and keyboard: This allows you to test and modify the software without having to use your actual eyetracker and dual screen system for that.

Both PCEXPT and NEWTRACK should be used in plain DOS, because DOS as a single-tasking operating system has good real-time processing properties. You would get worse timing accuracy and slower access to the displays and to the eyetracking hardware in a multitasking operating system like Linux, Mac OS or Windows, unless you use special real-time support. In Linux, real-time support would mean using an RTLinux kernel (real-time kernel which is doing the normal Linux kernel activities only while no time-critical things are pending) or at least using the real-time scheduling priority (allowing one task to grab all available CPU time). One freely available open source DOS implementation is FreeDOS, which should run on any old and new DOS compatible PC and which supports harddisk sizes of up to 2048 GB as long as the system BIOS can handle them.

2.1 Timing, memory and CPU

The DPI eyetracker in Saarbrücken is used for fast precision measurements during reading experiments now, so I suggested to write a new software from scratch which would be easier to maintain and free of unneeded drivers and features, while on the other hand being able to run on more modern but cheaper hardware. Starting with Intel Pentium, processors contain a so-called Time Stamp Counter, which counts the CPU clock cycles since the last reboot. It is a 64 bit value, so it will not overflow back to 0 even on fast systems with clock speeds of several GHz.

Even older technology is 32 bit computing and 32 bit address pointers – introduced for the PC platform with the Intel 386, about 15 years ago. Using this allows NEWTRACK to access large data structures easily, without having to split them into chunks of less than 65536 bytes or applying other unnecessarily complex programming techniques. NEWTRACK assumes that your system will have several megabytes of RAM installed, but even 16 MB should be more than enough. For floating point calculations, the presence of a floating point unit is assumed (otherwise it will be too slow). From the 486dx family on, an FPU will be present as part of the main CPU, so you do not have to worry about that.

Should you be using an old Pentium clone without Time Stamp Counter (e.g. the pre-MMX versions of Cyrix 6x86 CPUs), you still do not have to worry: The used DJGPP C compiler has a library function which provides microsecond timestamps by programming the built-in timer chip of every PC appropriately, and NEWTRACK uses this library function as a fallback if a Time Stamp Counter cannot be used. Be warned that you can block access to this counter on newer CPUs. This should not happen in DOS, but if the counter is present but still not usable, check if something did block access to it.

2.2 Dual display support

Both PCEXPT and NEWTRACK are designed for use on PCs with two screens: One will display the stimulus to the subject, and the other screen is used for the operator. In NEWTRACK, only text displays were used in the initial version. Later versions allow displaying both PCX images and text to the subject. The subject will see stimulus text, fixation marks (for calibration and gaze triggered stimulus presentation) and sometimes a gaze controlled cursor (for calibration tests). The operator will see the same, but will also see log events (similar to a status bar) and measurement results from the A/D (eye coordinates) and the buttonbox. In graphics mode, the operator will only see a text display, with fields of letters indicating where the automatic runtime image analysis has detected graphical objects in the stimulus display. The subject will see the text parts of the stimulus rendered on the graphical canvas, using the 8x16 font provided by the VGA BIOS scaled to a 16x32 pixel font. This allows both text and images to be shown to the subject at the same time.

NEWTRACK highlights stimulus words on the operator display when they are being looked at by the subject. This allows direct real-time evaluation of the experiment progress and will make calibration errors show up quickly. PCEXPT always uses a monochrome operator display and has no highlighting. Both PCEXPT and NEWTRACK have fallback modes for systems with only one screen, but those are not useful for other purposes than testing.

While it is hard to get monochrome graphics cards like the Hercules Graphics Card – and an ISA bus to plug them in – for a PC now, the system offers an easy way to connect two screens to one computer. Color and mono boards will use separate port ranges for control and separate address spaces for the frame buffers where the screen contents are stored. The mono board can be controlled by the system BIOS directly, while the color board is equipped with an own BIOS (basic input output system, containing basic initialization and driver software).

Normal PCs in 2004 will be equipped with an on-board or, more often, an AGP (accelerated graphics port) graphics card. Those are both configurable through PCI/AGP bus controls. There will only be one AGP slot, but also several PCI slots. The PCI bus is still widely used for adapter cards, but the market for PCI graphics cards has become very small. PCI cannot offer the high speed data transfer of AGP which is used for recent 3d games during texture upload, but it is fast enough for screen content updates in 2d applications like NEWTRACK. Because AGP graphics cards often have plenty of local RAM, 3d games can upload all textures once at level start and send only geometry data later. It might be useful for NEWTRACK to upload more than one image in graphics mode and then run several trials in a row, but this is not implemented yet. For normal text mode, the bus bandwidth even of the PCI bus is much better than needed, and for graphics mode a small delay (with blanked screen) before each trial will not cause real problems either.

It should stay easy enough to acquire a PCI graphics card during the next years, when all monochrome cards will have either fallen apart, ended up on dumpsters or will not be useable anyway because ISA bus support will no longer be available even on special motherboards – which is a pity, because ISA hardware design is very straightforward and many specialized industry ISA boards have been built for custom applications, although ISA is not a fast bus at all. Another upcoming trend is having more than one video output connector on one single graphics card, but there is no support for that in NEWTRACK yet. I assume that a generic driver for this could be written, using VESA VBE3 BIOS functions. VESA VBE3 is an interface standard for video BIOSes and drivers, and you can use drivers like UNIVBE or SciTech Display Doctor (DOS versions are now freeware) for older graphics card where the BIOS has no built-in support for it.

For maximum compatibility, the current version of NEWTRACK uses two separate graphics devices instead of one with two output ports. First, NEWTRACK scans for the presence of a classic color / monochrome dual head setup. If found, it will assume that both cards are set to text mode and write the text which it wants to be displayed to the two framebuffer memory areas. You can use the DOS MODE command to initialize both cards to text mode.

If no classic dual head hardware is found, NEWTRACK will scan the PCI/AGP bus system to find at least 2 VGA compatible devices attached to the bus. If so, it detects which of the devices is logically turned on. Because resources overlap, you cannot have two VGA devices turned on at the same time, but luckily this affects only the connection to the CPU. The disabled device will still continue to show whatever it has been showing on screen before it got disabled. That way, NEWTRACK can toggle between both devices. The one device which is enabled is the one where the screen contents can be updated. This should work for all VGA compatible PCI, AGP or on-board graphics cards, no matter how old or new they are. On PCI-enabled systems, the on-board graphics unit, if one exists, will be attached logically as if it would be a normal pluggable AGP or PCI device.

If NEWTRACK still did not find suitable hardware dual display, it will drop to a fallback mode: Every second line on the single existing screen will be used for the subject display, and every other line will show the operator display. This is meant only for testing purposes and allows you to do NEWTRACK development on any PC. It is recommended to use the DOS MODE command to select a mode with more than 25 lines on screen before using NEWTRACK in this fallback mode.

The general idea of virtually plugging and unplugging VGA cards in order to access two of them on the same PC is based on the RBdualVGA software by Ralf Brown. Apart from RBdualVGA, Ralf Brown (a computational linguist) has also collected the RBIL (Ralf Brown's Interrupt List), a comprehensive guide to DOS / PC programming interfaces (APIs) and hardware resources. Without those publications, writing NEWTRACK would have been much harder. Although RBdualVGA is open source software, the dual screen support of NEWTRACK has been rewritten from scratch. The 16 bit DOS Assembly language RBdualVGA programs could not have been merged easily into 32 bit DOS C language NEWTRACK.

Ralf Brown recommends that both VGA devices should be of the same kind, because only one BIOS can be used. While it is true that you can virtually plug and unplug PCI / AGP devices, the main system BIOS will only select one device at boot time and copy the BIOS of that device into later write protected RAM and run it, which will initialize the VGA device to text mode and show some product identification message. Even when you swap the plugging state to the other device later, you would have to unprotect the RAM to be able to replace the BIOS copy. Moreover, other programs will already have adjusted to the BIOS of the first VGA, and changing the BIOS will lead to a crash then. Now if you have two identical VGA devices, you can swap the plugging state and tell the BIOS of the first VGA device to program the VGA to the desired resolution and mode (e.g. with the DOS MODE command). It will make no difference for the BIOS that it is actually initializing the second VGA although it shipped with the first VGA. It is very common that a VGA device does not display anything at all after a reset or reboot until it is initialized by the VGA BIOS.

Because NEWTRACK aims to work on simple platforms, I looked for a way to make it run with two different VGAs. On our test systems, we use normal PCs with one standard AGP VGA card and one additional PCI (there is only one AGP slot) VGA card, the latter taken from any source of old PCI VGA cards. You can select in the system BIOS setup whether the PCI or the AGP bus should be scanned first, and the first VGA device which is found will become the primary system display during boot. You can copy the BIOS of the primary / active VGA into a file for later use, as explained in the appendix.

Now the problem is how to initialize the secondary display: To achieve that, you have to logically connect it first. NEWTRACK has a command line option which simply swaps connection status and then exits to DOS again. Now, you have to initialize the VGA hardware. If both devices are compatible to each other, as simple MODE command will be enough. However, it is more likely that using MODE at this point will not bring the secondary display to live. Rather, it will crash the VGA hardware because you used one highly hardware specific VGA BIOS to program another, different, VGA hardware.

To escape this sad situation without having to acquire two identical VGA devices, I conceived a method for loading a VGA BIOS from DOS. This works particularily well if your VGA BIOS contains a string IDX which means that a data structure for loading the VGA BIOS to other places in RAM or even running it from protected mode is present. In other cases, your mileage may vary. Some, even old, BIOSes will just accept being run from an unusual location: We can only load the BIOS of the secondary VGA to DOS memory, not to the normal address which is already used by the BIOS of the primary VGA. Other BIOSes are less flexible, starting them in DOS RAM will crash. You have to try which of your two

VGA BIOSes will accept running from DOS. Then make the other one – where the VGA BIOS can not be run from DOS – the primary display which is initialized at boot time.

This will allow you to 1. boot, 2. swap the plugging state with help or NEWTRACK, 3. run the VGA BIOS of the secondary display from DOS, as explained in the appendix, and 4. swap the plugging state again. Now finally both displays will be running in text mode, and the plugged VGA is again the one for which you have the BIOS activated and permanently loaded at boot time, so you can use it even in other display modes. If none of your VGA BIOSes accept running from DOS, you are out of luck. Either replace one of the devices by a different model or take care that both devices use the same chipset. If both devices use the same chipset, the initialization sequence is: 1. Boot, 2. swap, 3. use DOS MODE command to select a display mode (e.g. CO80 for 80x25 color text mode), 4. swap again. In this case, you do not have to run any BIOS from DOS.

NEWTRACK will use the VGA which is logically connected when you start NEWTRACK for the stimulus (subject) display and the other one for the operator display. If you use the classic color plus monochrome hardware, the monochrome display will be the operator display. You can use NEWTRACK to swap the logical connection state before starting the actual experiment if you want to swap the subject / operator assignment of both VGA devices in the dual VGA case.

2.3 Eyetracker and buttonbox connection

Because NEWTRACK aims to work on a PC without ISA bus, the button box hardware had to be replaced by something which does not rely on the digital I/O board which is used by PCEXPT. However, because the ISA I/O board might already be present on a PC anyway, drivers are included even in NEWTRACK. This case is automatically detected. You can actually run NEWTRACK on most PCEXPT compatible PCs as long as they PC have a fast 486dx or better processor and a few megabytes of system RAM. Only the CIO-CTR / DAS16 hardware is supported, while PCEXPT also has drivers for an alternative I/O and A/D board each, but those are used far less often nowadays.

The recommended way of running NEWTRACK, however, is on a Pentium compatible or newer PC with 2 VGA cards, so there had to be a way to work without the data acquisition hardware which is used PCEXPT. I decided to design a custom hardware which has to be connected to the printer port. That way it does not depend on ISA, PCI or whatever else bus system you use, but you should not use it with an USB to parallel printer port adapter: DOS does not support USB devices at all, and the system BIOS only allows DOS compatible access to USB devices in very specific cases. The special printer port device used by NEWTRACK is not such a BIOS supported case! The circuit diagram of the new device can be found in the appendix.

This design decision might be a bit nonstandard, but I found no other suitable and compatible way in the desired price range than using a custom built printer port device. You can use an existing joystick port to connect up to four buttons (adding drivers to NEWTRACK for this will be easy), but the joystick port is far too inaccurate and slow for measuring the analogous eye coordinate signals.

For the eye coordinate signals, both a certain speed and some minimum resolution have to be reached by the A/D conversion hardware. I was thinking about using serial port A/D converters which could sample thousands of values per second with 12 bit precision. The driver has to read the data bit by bit, because the serial port hardware is not able to use word lengths of more than 8 bits and because it can be somewhat slow. The problem with this is that you have to use either two devices (need 2 free serial ports, so if you want to connect a mouse as well, it has to be a PS/2 one, as most PCs have at most 2 serial ports unless you add extra hardware).

Alternatively, you have to use one device with a multiplexer. This means that you can tell the device which of two or more input lines has to be measured. By switching lines after each measurement, you still can get fast enough sampling of both the X and the Y coordinate and maybe optionally the track loss / blink signal, but the driver gets complex and the data exchange keeps the PC busy for a while. If you add a driver for using either one multiplexed or two separate serial port A/D converters to NEWTRACK, please submit it for inclusion with future versions of NEWTRACK.

This is why I decided to design an entirely new device, with exactly those features which are needed for eyetracking but without expensive extra features. This device connects to the parallel printer port and exploits the fact that starting with 386 based PCs (maybe even earlier) printer ports became bidirectional. This is not to be confused with ECP or EPP printer port functions: ECP / EPP allows you to transfer data to and from the printer in a very fast and heavily hardware assisted way. My device is already happy if the data bus between PC and printer works in reverse direction.

For that, you tell the port hardware to stop sending a data byte. Then you can read back data on the bus which is sent to the PC. This input byte is of course not wide enough for the 12 bit A/D conversion result (coordinate, in 0 to 4095 range) which the PC should be able to read from the device. Therefore the device uses two monolithic serial A/D converters. The other 6 bits of the byte after subtracting two bits for the serial X and Y data are used for the four buttons and up to two digital signals from the eyetracker like for example the track loss / blink signal.

As button and track loss / blink signals are already digital and already follow the 0 / 5 Volt signal levels of the printer port, no extra chips are needed to process the digital signals. However, signals which are connected to bidirectional data bit lines of the printer port should never force the line to high without making sure that the printer port is in input mode. Forcing the line to low is always safe for data bit lines (the port uses open collector output and pull-up resistors).

NEWTRACK uses the $\overline{\mathsf{AutoFeed}}$ line to tell the A/D chips when data may be sent to the PC, and the $\overline{\mathsf{Strobe}}$ line as clock for the serial data transfer from the chips to the PC. Two converters are used in parallel. This allows to read both coordinates simultaneously. With only one A/D, they would have to be read in an alternating pattern, switching between the two coordinate inputs after each measurement.

I recommend to use the LTC1286 12 bit monolithic A/D converter for this program: The clock for the serial data transmission can be up to 200kHz, so reading the coordinate data only takes a fraction of a millisecond, which is more than enough speed given that the eyetracker hardware itself is not able to update the signal fast either. Reading the eye coordinates 1000 times per second still gives you some so-called empty resolution beyond what is actually useful given the frequency response of the eyetracker. The selected converter chips are a cost-effective solution (less than 15 Euros each) and allow enough speed and accuracy for our purposes.

Using only 256 steps for each coordinate might be just enough for text presentation, but the margin gets quite narrow given that you have to pass the values through a calibration formula and usually only part of the possible viewing angle / voltage range of the eyetracker will be used for an experiment. Using more steps for each coordinate than the 4096 which you get with a 12 bit converter will usually mean that you will use a 16 bit converter which allows 65536 steps. Those converters are, at least at useful speeds, far more expensive. And the actual resolution of an eyetracker like the Fourward DPI system only reaches at most a few thousand steps per axis anyway, although there are no actual steps in the DPI system. Instead, it will simply fail to detect small eye movements and small differences in gaze positions.

Taking all factors into account, I recommend to use NEWTRACK with 12 bit A/D converters. The DAS16 data acquisition board (used by PCEXPT) uses one fast multiplexed 12 bit converter, and I have added a driver for that to NEWTRACK. So if you already have a PCEXPT compatible system, you can use it for NEWTRACK as well. Adding a second VGA card will be good for NEWTRACK and will be ignored by PCEXPT, which will continue to use one VGA and one monochrome card. If you do not add a second VGA card, NEWTRACK can also use the two existing cards, but your operator display will stay in monochrome and with annoyingly low screen refresh rate that way.

If NEWTRACK can find neither a DAS16 nor the described printer port device with attached serial A/D converters, or if it did find either of the converters but you have chosen not to use it, NEWTRACK will use the mouse as coordinate source instead. You have to load a mouse driver to make use of this feature. Some mouse drivers can take quite a while, more than one millisecond, to check for mouse movements. In that case, you will see lag warnings flashing on the screen whenever NEWTRACK noticed that acquiring the new coordinate pair took too long.

The mouse mode will not use screen coordinates in the normal way. Instead, when you move the mouse, the imaginary voltage measurements from the eyetracker are updated. That way you can calibrate NEWTRACK to any desired mouse sensitivity, but be aware that you can also end up calibrating it to inverse coordinate directions or non-linear behaviour. This might even be called a feature. Calibration is limited in that you can only use X movements for the X axis and Y movements for the Y axis, so if you rotate the mouse or if the subject screen is not aligned with the eyetracker, NEWTRACK cannot correct the effects of such rotation yet.

2.4 Calibration

Because hardware linearity is quite good both in the used A/D converter devices and the Fourward DPI eyetracker itself, the measured voltages are converted into screen coordinates (available both as pixel coordinates and as text mode coordinates) by using only 3 parameters for each coordinate. There is no support for crosstalk or rotation correction. The screen coordinate is the voltage multiplied by some factor plus some offset plus, for nonlinearity correction, the square of the voltage multiplied by some factor. PCEXPT uses a more advanced calibration scheme here. Both PCEXPT and NEWTRACK use a grid of 9 calibration marks. Three each at the top, in the middle, and at the bottom. The marks are not at the screen edge but offset by some margin.

During calibration, the operator sees which mark is active and which voltages are sampled (or actually the value in zero to 4095 range which corresponds to the voltage). For already processed marks, the last sampled voltage is displayed. The operator has to confirm each mark gaze with a button press in NEWTRACK while PCEXPT can automatically assume that the subject is fixating a mark when eye movement stops.

When all marks have been calibrated, the six calibration parameters are displayed and an eye cursor becomes visible. The operator can now check the calibration and repeat it if needed. The subject will only see one of the calibration marks blinking and color-cycling at a time, without getting the numerical feedback. The blinking of the marks is a compromise between drawing lots of attention and being too nervous and annoying.

When all marks have been calibrated, an equation system is solved to adjust the six parameters. After the first calibration, the operator will see an eye cursor (based on the previous calibration) during subsequent calibrations. The marks will be calibrated in left to right, top to bottom order only if no valid calibration is loaded, e.g. only initially and after calibrations which led to parameters which were rejected by NEWTRACK because they were out of plausible or computable range. For all other calibrations, the order in which the marks will be calibrated is randomized.

3 Compiling and portability

PCEXPT is meant to be compiled with Borland C 3 / Turbo C 2 (the latter is now freely available from the Borland online community museum), along with Borland Turbo Assembler for one or two performance critical code fragments. I have ported those fragments to the free NASM Assembler for convenience. You could also use the free Arrowsoft ASM to compile TASM code.

For better portability, and to make full use of 32 bit CPUs, NEWTRACK is written for the DJGPP compiler. DJGPP is the DOS version of the well-known gcc/g++ compilers which are popular among Linux programmers. It is open source and freely available. The C library of the DJGPP platform is modelled after the GNU C libary of Linux, but

NEWTRACK itself is not meant to work in Linux!

Small parts of NEWTRACK are written in inline assembly language: A tiny delay which must not be optimized away by the C compiler in the A/D interface readout and the code which reads the Time Stamp Counter. This inline assembly is processed with the GNU AS assembler which is part of the DJGPP compiler suite. Note that AS uses AT&T syntax which has some important differences from classic Intel Assembly language syntax (as used in MASM, TASM and NASM assemblers, for example).

If you ever want to create a Linux version of NEWTRACK, you will probably need root privileges to run it. Some important points: NEWTRACK is real-time software, so for a Linux version, you have to make sure that no other tasks can take CPU time at the wrong moments. Only root is allowed to reserve the CPU for a single task. NEWTRACK also accesses hardware directly. You can rewrite most of this, but you will still need privileges to access the corresponding devices. Used devices are the text mode frame buffers of two (!) graphics cards, the time stamp counter, the printer port, the keyboard, optionally the classic (known from PCEXPT) ISA devices, optionally (if Linux itself does not support dual display usage for your hardware) the PCI/AGP bus configuration. But in the latter case, you even have to initialize the second display, which will often involve running the BIOS of the second graphics card. For graphics mode, the BIOS of the graphics card which displays the stimulus must be working, and the card must support VESA VBE 2.0 frame buffers.

Adding support for new kinds of A/D devices or buttonboxes should be pretty straightforward as long as you have some kind of technical data sheet for the device available. Support for VESA VBE3 controlled dual head displays with only one single graphics card which has two separately useable outputs should be possible, too, but you will have to remove the PCI/AGP bus control and add linear framebuffer or BIOS access methods for the two frame buffer areas of both displays instead.

4 File formats

4.1 Scripting language

To keep the NEWTRACK system simple, it does not support the fancy scripting language extensions of PCEXPT. Instead, it uses the proven scripting language of our READING software, where applicable. See the appendix for details. The READING scripting language (sometimes also referred to SMI scripting language, although it is not designed by SMI; only the first implementation was in the READING software for SMI trackers) allows comment lines, definition lines and stimulus lines.

A definition line assigns a name to a set of properties. One of the properties is the trigger method: Showing the stimulus at once, or gaze-activation with a target mark at the place where the text will start. Or gaze-activation with a centered target mark (drift check). Gaze-activation can always be bypassed using the operator buttons. Other properties are whether or not the eye movement data should be logged (usually not for fillers and other general text display) and which buttons (and keyboard keys) are valid subject responses. Pressing a valid response key ends a trial. It is common to use arrow or buttonbox keys for yes / no responses and the space bar or buttonbox keys for “I am done with looking at this trial”. In NEWTRACK and PCEXPT, both keyboard and buttonbox are useable. READING is used without a buttonbox normally, as operator and subject each have their own keyboards in front of them.

Basic stimulus lines for both NEWTRACK and READING – the script files are completely different for PCEXPT – consist of a category selection (you have to define the category in a define line before), a label which will be used for logging, a millisecond timeout (to limit the time of stimulus presentation if the subject fails to press a valid button at all), the keyword “inline” and some text.

With the READING software, stimulus text can also be read from external files. In that case, the file name is written instead of the “inline” keyword. Both READING and NEWTRACK scripting languages allow image presentation, but syntax differs: For READING, you give two bitmap (bmp) image file names (one for the stimulus and one for a template) and optionally a wave audio (wav) sound file name. The template consists of colored areas in the same layout as the stimulus, and READING will log a color code taken from the template for gazes at graphical objects. If you want to present images and text at the same time, you have to put the text into the bitmaps manually.

NEWTRACK syntax for image presentation uses the keyword “image” or “image2”, followed by the desired minimal vertical distance between image contents and text (negative if the image is below the text, otherwise positive), a PCX image file name and finally some (optional) inlined stimulus text. Image2 is for single lines of text automatically placed close to the image, while image is for centering the text in the unused area of the screen. Template image files are not used, as NEWTRACK analyzes image contents on the fly and enumerates objects on the image from left to right. You can use the PCX2TPL too which is part of the NEWTRACK package to create a template image file for a given stimulus image file, if you want. Audio stimuli are not supported yet.

You can use the included test program to get a visualization of how NEWTRACK “understands” a particular image. Make sure that objects do not contain large areas of background color and that they are not too close to each other. On the other hand, if parts of an object are separated from the rest of the object by too much space, they will be counted as separate objects.

You can split the inline text over several lines with NEWTRACK (not with READING) by placing a backslash right before the end of a line to indicate that the line is continued on the next line. The backslash / linebreak pair is treated like a normal space when displaying the stimulus. This allows you write stimulus scripts which are easy to read, print and edit, because you can keep lines short. Display line breaking in both READING and NEWTRACK is done automatically: If a line is too long to display, it will be wrapped to the next line at a suitable place (word boundary).

In addition to that or to override automatic line breaking completely, you can insert $\backslash n$ at any place to force a line break, which gives you extra control over the presentation. Last but not least, _ can be used to mark a boundary which will not be displayed. This allows you to have e.g. gaze events for the parts of a compound word as if each part would be a word on its own. Line spacing is automatically selected in a certain range depending on how many lines of text will be displayed. If you use the automatic line wrapping, NEWTRACK will check if it can use less lines by reducing the left and right margins. If so, it will prefer smaller margins over more lines.

4.2 Log file format

The log file is a text file which has a millisecond timestamp at the beginning of each line. For each stimulus, the positions of all words are logged, as well as the display size and trial id string. Then the display is enabled and the clock starts to run at exactly that moment. Whenever the subject starts or stops looking at a word region, a log entry is created. This log entry contains the area number, average and current gaze coordinates and the word (enter case) or the duration / dwell time of looking at that word (leave case).

When an image is shown as part of the stimulus, the objects on the image are automatically numbered (using run-time image analysis). Looking at an object is logged as a special case of looking at a word, as shown in the log file example below. Exact object shapes can be complex, so they are not logged! You have to check manually which number is assigned to which object, by creating a test log file or by reviewing the image analysis result display on the operator display for each image. In both cases you have to test all images, to see if analysis results match your expectations. The image is scanned from left to right and object numbers are quite predictable. You can also use the included test program to get a graphical visualization of the analysis results for a particular image.

The current version of NEWTRACK allows only PCX file format for images. Color depth has to be 8 bits per pixel. Presentation will use VESA VBE 2.0 graphics modes with 16 bits color depth. This allows using separate text and image color spaces. You should convert images to use between 20 and 190 colors, if possible without dithering. Useful free tools for mass conversion of images are 'convert' and 'mogrify' of ImageMagick (see references). The analysis assumes that the most used color is the background, and areas in background color are treated as background even if they are inside an object as judged by humans. Then NEWTRACK blurs the non-background areas a bit, to reduce the effect of small holes and dithering, and uses a recursive fill algorithm to find coherent objects. Areas are searched from left to right, so the assigned object numbers will be lowest for objects on the left. Finally, the object number (if any) for each screen pixel is stored in an array to allow fast gaze analysis.

The gaze analysis is done in real-time during stimulus presentation, and some filtering is applied to remove glitches (and optionally blinks). When the presentation is gaze-triggered, an extra event is logged for the gaze at the trigger area and an extra timestamp marks the moment when the whole stimulus is revealed after the gaze trigger. For each trial, the response button (or a timeout event) which triggered leaving the trial is logged, as well as the status (ok, error, aborted, repeated) and the result (usually equal to the response). Repeated trials are created if the operator interrupts a trial for recalibration.

The log file format is modelled closely after the log messages of the READING system, but uses a different file format: READING creates a binary log file, possibly with an eye coordinate data stream embedded, and only an extra conversion step extracts the textual log messages from it. The syntax of the resulting text, however, is almost the same in both systems, so it is easy use one common analysis software for log files of both programs. Have a look at the sample log file and sample script file to find out more.

Take care to convert the log files to your preferred encoding, e.g. with unix2dos: The raw NEWTRACK log files always use DOS line breaks and use the DOS codepage, which affects the representation of non-ASCII characters like accents and umlauts. The scripts should also be converted to DOS style before using them with NEWTRACK: While NEWTRACK has flexible linebreak parsing, it does not translate all accents and umlauts to DOS encoding, so you have to do that conversion for NEWTRACK. Only the German umlauts in ISO 8859 1 (also known as Latin 1) encoding are understood, other encodings and other accented characters just get displayed wrong if you fail to provide DOS (cp437) encoded scripts. Take care: Newer versions of unix2dos and the dos2unix counterpart need extra command line options to enable umlaut conversion in ISO mode, as they would otherwise default to using plain ASCII or to Unicode encodings.

To run an experiment with text stimuli in NEWTRACK, just start NEWTRACK and give the script file name and subject name. The log file name will be generated from the subject name. You can also use command line options instead of using the interactive configuration, or mix both. See the appendix for the help message of NEWTRACK.

---

5 Appendix

5.1 References

Fourward Technologies, Inc. (DPI eyetracker):
`http://www.fourward.com/`

SensoMotoric Instruments GmbH (head-mounted camera eyetrackers):
`http://www.smi.de/`

EyeLink II by SR Research (head-mounted camera eyetrackers):
`http://www.eyelinkinfo.com/`

GNU General Public License (by the Free Software Foundation, Inc.):
`http://www.gnu.org/licenses/gpl.html`

Borland Developer Network Museum:
http://community.borland.com/museum/

NASM Assembler:
http://nasm.sourceforge.net/

DJGPP Compiler by DJ Delorie:
http://www.delorie.com/djgpp/

ImageMagick image processing tools:
http://www.imagemagick.org/

SciTech Display Doctor (free for DOS VESA VBE3 driver):
http://www.scitechsoft.com/products/ent/free_titles.html

PCEXPT eyetracking software by Charles Clifton, Jr.:
http://www-unix.oit.umass.edu/~cec/software.html

FreeDOS, a DOS operating system:
http://www.freedos.org/

Ralf Brown's Files (RBdualVGA, RBIL, others):
http://www-2.cs.cmu.edu/afs/cs/user/ralf/pub/WWW/files.html

VBEHz, a free tool to change your screen refresh rate:
http://home.arcor.de/g.s/vbehz.htm

OpenStat, a free statistics system, similar to SPSS:
http://www.statpages.org/miller/openstat/

Octave, a free mathematical programming language:
http://www.octave.org/

Parallel printer port pinout:
http://www.hamhud.net/ka9mva/regpins.html

Joystick port pinout and information:
http://www.epanorama.net/documents/joystick/pc_joystick.html

5.2 Sample stimulus script file

You can find a longer example in the ZIP file with the source code and the NEWTRACK binary.

```
# Both ; and # can introduce comments. Empty lines are ignored.
# Important: "inline" text cannot contain comments, so ; and #
# are interpreted as part of the text there.
# \n forces line break. \_ marks invisible area break.
# Any amount of whitespace counts as word boundary, which will
# be displayed as a space of automatically selected width, or
# as line break if the line would get too long otherwise.

# The SMI EyeLink based Saarbrücken READING software did not
# break areas at every space, but NEWTRACK does. You can pool
# data again with other tools later.

# READING left/right/down are interpreted as Y/N/Y respectively.
# Subject button box buttons are equivalent to Y/N.
# Single letters (except # ;) can be defined as allowed response
# as well. They stand for those letters being typed at the PC
# keyboard. Timing will be less exact for those.

define Statement gaze stream down      # define a trial class
define Intro nogaze nostream down # no wait for gaze trigger
define FourQ gaze stream left right t f # four possible answers
define Quest gaze stream left right # two possible answers
# the (no)gaze / (no)stream keywords must be in this order!
# Apart from gaze / nogaze, driftcorrect mode is also possibe.

Intro welcome 60000 inline Welcome to our demonstration\
  eyetracking experiment.\nPress the YES or NO button after\
  reading each page.\nIf the text did not contain a question,\
  use the YES button.
# 2nd word is a trial label, 3rd is a msec timeout.
# For now, only inline type trials are supported.
# Note that READING doesn't allow '\' for 'ignore end-of-line'.

Statement story-1 60000 inline 'What was that?' inquired Alice.\
  'Ree\_ling and Wri\_thing, of course, to begin with,' the\
  Mock Turtle replied; 'and then the different branches of\
  Arith\_metic - Ambi\_tion, Dis\_traction, Ugli\_fication,\
  and Deri\_sion.'
# This is an example of how to use the \_ 'invisible boundary'.
```

```
Statement story-1 60000 image -10 turtle.pcx The Mock Turtle,\
  which you can see above, now told Alice about Draw\_ling:
# This is an example of how to use images in NEWTRACK. Note that
# READING uses different syntax. In NEWTRACK, you can optionally
# present some text along with the image. The -10 means that the
# image will be shown 10 (real) pixels from the bottom edge of
# the screen, and the text (in 16x32 real pixel font) will be
# shown on top. Use positive values for positions relative to the
# upper edge, with the text at the bottom.
# Images have to be in (exactly) 8 bit per pixel PCX file format.

Statement story-eel 60000 inline We also learn Draw\_ling. The\
  Draw\_ling\_-master was an old conger\_-eel, that used to\
  come once a week: he taught us Draw\_ling, Stretching, and\
  Fainting in Coils.'

Quest question-eel 60000 inline You have reached the end of\
  the story. So now tell me:\nWas the Drawling\_-master a\
  concrete eel?
```

5.3 Sample log file

This sample log file is very incomplete: Many lines have been removed to keep the example short. You can find a more complete example in the ZIP file with the source code and the NEWTRACK binary. Lines which start with ';' are not part of the log file but have been inserted as comments.

```
0000000 DISPLAY_COORDS 0 0 640 192
; The size of the display, in text mode pixels (see below).
0000000 TRIALID intro
; The label for that trial
0000000 INFO WORD 0 108 24 171 55 Welcome
; For each word, a field number and coordinates are logged.
; Numbers are, respectively: field number, x1, y1, x2, y2.
0000000 INFO WORD 1 172 24 195 55 to
0000000 INFO WORD 2 196 24 227 55 our
0000000 INFO WORD 3 228 24 339 55 demonstration
0000000 INFO WORD 4 340 24 435 55 eyetracking
0000000 INFO WORD 5 436 24 531 55 experiment.
; Everything except whitespaces and \_ and \n is a 'word'.
; (some lines omitted)
```

```
0000001 DISPLAY ON
0000001 SYNCTIME
; This marks the moment when the subject display is turned on.
0000549 ENTER WORD 0 111 42 112 43 Welcome
; After 0.55 seconds, the a gaze at the word 'Welcome' is found.
; The numbers are: 1. field number, 2. x1, 3. y1, 4. x2, 5. y2.
; The (x1,y1) is the average, the (x2,y2) the current position.
0001007 LEAVE WORD 0 144 43 190 43 457 0
; The numbers are: 1. ... 5. as above, 6. dwell time, 7. zero
; (Times are in msecs. The zero means 'no audio timestamp'.)
; (some lines omitted)
0002003 ENTER WORD 5 437 41 438 42 experiment.
0002380 LEAVE WORD 5 471 46 472 60 376 0
; Now we have a jump in the timestamp - reading the next line.
0002857 ENTER WORD 6 134 71 134 72 Press
0003375 LEAVE WORD 6 136 83 160 88 518 0
; (some lines omitted)
0008196 ENDBUTTON 8
; Trial ended by a press of button 8 (4/8: buttonbox buttons).
; For keyboard responses, an extra ANSWERBUTTON line is logged:
; ANSWERBUTTON x KEY k, where k is the ASCII value or a 16 bit
; value describing the keypress and x is 128 plus the ENDBUTTON
; interpretation code of the keypress. If the subject did not
; respond at all, a TIMEOUT line will be logged.
0008196 TRIAL OK
; No errors. Other possible events: TRIAL ERROR, TRIAL ABORT
; (aborted by operator), TRIAL REPEATED (after recalibration).
; Whole experiment cancelled is logged as 'EXPERIMENT ABORTED'.
; An example error is 'TRIAL ERROR notype' which would mean
; that you tried to define a trial of an undefined type / class.
0008196 TRIAL_RESULT 8
; The result usually equals the endbutton if trial was okay.
; For keyboard responses, the value will be 128 higher. If
; no response found (error/timeout/repeated), value will be 0.
; On to the next trial!
0008196 TRIALID story-1
; (some lines omitted)
0000000 TARGET ON 0
0002138 TRIGGER MAIN 0 44 40 44 40 400
; This is a gaze activated trial: It took 2 seconds before the
; subject gaze at the activation mark reached 400 msec dwell
; time. Numbers: area 0, coordinates (44,40) / (44,40).
0000001 DISPLAY ON
; Possible timeout would be 'DISPLAY STILL OFF BUT TIMEOUT'.
```

```
0000001 SYNCTIME
; Now the actual stimulus is revealed and the clock is reset.
0000009 ENTER WORD 0 44 41 44 42 'What
; After 9 msec, the gaze was judged to be at the 1st word now.
; (rest of the log omitted)
```

If your experiment contains stimulus items of 'image' or 'image2' type, the displayed image will be analyzed automatically at run-time and the visible objects will be numbered. The generated `ENTER WORD` and `LEAVE WORD` events will use special area numbers and the 'word' assigned to an area will have the format `PCX/5` where 5 is the object number. The area number is always some constant plus the object number here.

Right before the `DISPLAY ON` log line there will be a log line like `IMAGETRIAL 84 383 -19 8 stimulus.pcx`, the numbers being the range of Y positions where image objects are found on screen, the number of text lines occupied by the image (counting from the upper or lower edge of the screen, negative values mean that the image is at the bottom) and the number of objects which were found in the image. The first two numbers are not necessarily in screen pixel coordinates!

Logged coordinates are scaled to a coordinate system where each text character is 8 pixels wide and 16 pixels high. The actual font size is usually 16 by 32 pixels in graphics mode for NEWTRACK, so the normal scaling factor will be two. In text mode, some graphics cards will use other fonts than 8 by 16 pixels, but the logged coordinates will be scaled – 80 logged pixels to the right always means ten characters to the right, no matter what font or resolution is used, for text mode and for graphics mode.

Make sure to check the analysis result on the operator screen when you use an image for the first time. Possible problems include objects falling apart into parts, adjacent objects being counted as one and holes in objects – areas which accidentally match the background color and are analyzed as belonging to the background.

5.4 How to initialize the dual display system in DOS

I have written an helper tool to save the VGA BIOS of the primary VGA (the one which is activated at boot time) into a file and load it from DOS later. This will not work with all VGA BIOS brands, but you must have either at least one VGA graphics card with a BIOS which can be loaded with my tool or your 2 VGA graphics cards have to be able to work both with the same BIOS (i.e. use the same chipset). In the latter case, my tool is not needed: You will only have to use NEWTRACK to flip displays, then run the DOS MODE command to select a suitable display mode and (optionally) flip back. MODE might sometimes think that you already are in the desired mode. As a workaround, use MODE twice to select two different modes. Example of this method:

```
CLS
NEWTRACK flip
MODE co40
MODE co80
NEWTRACK flip
```

After those steps, the secondary graphics card should be initialized and in 80 column / 25 line text mode. You may have to do CLS again to get the cursor position corrected at the end. You will probably not be able to see anything on the secondary display until the end of the procedure, so prepare to type the commands blindly or use a batch script.

If your VGA boards differ too much, the above steps will only crash at least one of your VGA boards, so you will not get both boards to display text mode. Getting one of the two boards to work is easy: Just select it as the one which is to be activated at boot time as primary system display. The problem is activating the other board. For that, you have to load the VGA BIOS of the other board from DOS. You need the VGA BIOS of the other board in a file for that, so select the desired board as primary one in your BIOS / CMOS setup. Then, boot to DOS and run

```
USEVBIOS S yourvga.rom
```

Now reboot and change the setup back, so that the other VGA will be the primary display again. You only have to do the above step once and you can leave your display selection unchanged from now on. Boot to DOS and run

```
MODE co80
NEWTRACK flip
USEVBIOS L yourvga.rom
NEWTRACK flip
CLS
```

This will temporarily connect the VGA board for which you saved the VGA BIOS into a file, run the initialization part of that saved VGA BIOS, and connect the other VGA board again. Now both displays should be in text mode and ready for use with NEWTRACK. If the saved VGA BIOS cannot be run from DOS – which depends on the version and brand of your VGA BIOS – you will have to try all (!) the steps with the opposite system setup. Save the VGA BIOS of the other card to another file and try running that from DOS. If still one of the displays is not properly initialized, or the system crashes in the `USEVBIOS L` step (for both possible combinations of primary / boot time and secondary / initialized from DOS display), you are out of luck. You have to change one or both cards then, as explained above. Alternatively, you might be able to initialize the secondary display with help of UNIVBE, but this method has not been explored yet.

If you are using a classical VGA / monochrome combination, the VGA board should be selected as primary display. The DOS MODE command should then be able to initialize the monochrome (e.g. Hercules Graphics Card) board:

```
MODE MONO
MODE CO80
```

Even if switching to or from mono mode clears the other display, it should still be in text mode and useable by NEWTRACK after the mode switch.

5.5 Printer port data aquisition device

To read the coordinates from the Fourward DPI eyetracker, you need 2 A/D converters with -5 Volts to +5 Volts input range. Because we are not using the maximum trackable viewing angle of the system, we reduce the A/D input range to -2.5 Volts to +2.5 Volts. Further reduction would get clipping limits too close to the used range, measurement would get too non-linear. Clipping is not needed if you use the full -5 Volts to +5 Volts range, but then only a small part of the A/D range is used and each screen character corresponds to fewer A/D units.

I decided to use 2 monolithic LTC1286 converters with 12 bit resolution (4096 steps) and up to 12500 samples per second. Speed depends on the used serial data transmission clock speed. Please check the LTC1286 data sheet for details about the timing and communication protocol. As the LTC1286 are only 15 Euros each, I decided to use one per channel, not using a multiplexer. This allows to read both the X and the Y voltage at the same moment and reduces the data transfer time because both values are read in parallel.

Using the printer port in the input direction (possible since PS/2) allows you to read eight digital signals, so the device is used for four buttons and the track loss / blink signal as well. The A/D chips have built-in logics to disconnect their output signals from the bus while the printer port is in output mode. The $\overline{\text{AutoFeed}}$ signal is used to tell the device when the printer port is in input direction.

Although the A/D converters consume very little power, you have to get a 5 Volt power supply for the device. You can either connect to a floppy power connector in the PC and have a cable to the device from there. The 5 Volt supply of the joystick port is often too weak. Other possibilities are the serial port control lines (can be programmed to either positive or negative 12 Volts and can supply 20 mA each, far more than needed for the device) and the keyboard connector. If you use a 12 Volt source, you can regulate to 5 Volt for the device, which should give you less noise on the supply voltage compared to using 5 Volts directly. Make sure that the device is not connected to the printer port while not being connected to the power supply – it can damage the chips to have data lines connected but no power present.

The button box and DPI eyetracker itself are hard to damage by wrong connections: The Fourward DPI eyetracker has protection against signal short-circuit events of any duration on the X and Y voltage outputs and the track loss / blink signal.

Note that you cannot replace the 78L02 (2.5 Volt voltage regulator by a simple Zener diode, otherwise the voltage would change when the current taken from the reference changes. Zener diodes or precision reference diodes are okay when combined with an Op-Amp driver, though: The Op-Amp draws constant current from the reference diode even if the reference voltage load at the output of the Op-Amp is changing.

NEWTRACK is configured to read the track loss / blink bit by testing the printer port status bit 2 (mask 8). The X and Y data stream from the A/D chips is read through the data port bits 5 and 4 respectively (masks 32 and 16). The buttons for YES, NO, CALIBRATE and NEXT are assigned to bits 3, 2, 1 and 0 of the data port (mask 8, 4, 2 and 1) respectively. If you decide to connect the signals to other bits, make sure to adjust the corresponding #define lines. One possible change would be modifying the button box input to use the joystick port and use two printer port status signals for the A/D data stream. Then you no longer have to use the input mode of the printer port.

Pin assignment information: Drive power plugs use black for ground, yellow for 12 Volts and red for 5 Volts. The joystick port pins are: Pin 2, 7, 10 and 14 for buttons 1...4 (normally open buttons, connected to ground, which is at pins 4, 5 and 12). Pin 3, 6, 11 and 13 for X1, Y1, X2 and Y2 respectively. Pin 1 and 9 for 3.3 or 5 Volt supply to which the X/Y potentiometers are connected. Pin 12 and 15 are sometimes used as midi output / input.

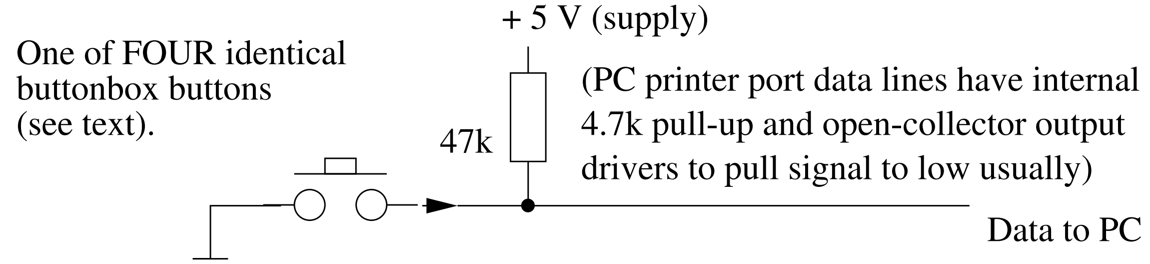
Printer port pins – at the sub D 25 pin connector – are: Data bit 0...7 are pins 2...9. Pin 1 is $\overline{\mathsf{Strobe}}$ and pin 10...17 are, respectively, $\overline{\mathsf{Ack}}$, Busy, PaperEnd, Select, $\overline{\mathsf{AutoFeed}}$, $\overline{\mathsf{Error}}$, $\overline{\mathsf{Init}}$ and $\overline{\mathsf{SelectIn}}$. Pins 18...25 are ground.

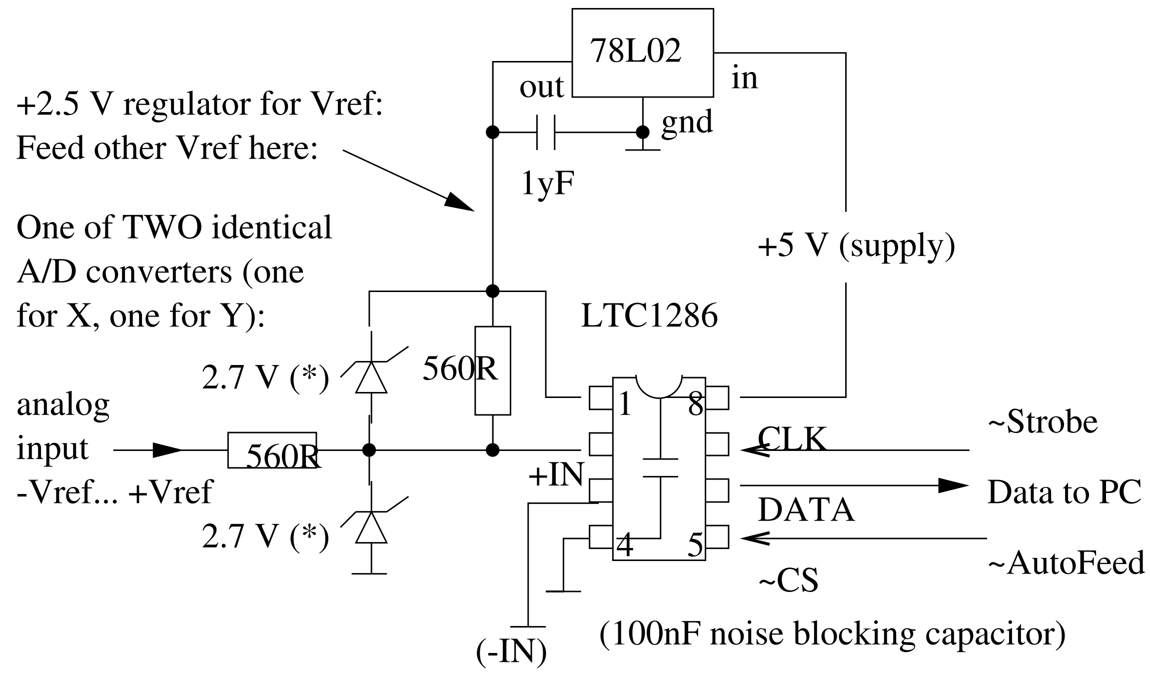
If your printer port uses 3.3 Volt signal levels, or if you want to improve signal quality and protect the A/D chips, use one 74HCT125 (four drivers with separate $\overline{\mathsf{enable}}$ lines) for the clock, $\overline{\mathsf{CS}}$, X and Y bus lines between device and printer port. The $\overline{\mathsf{enable}}$ lines for the X and Y drivers should be connected to the $\overline{\mathsf{CS}}$ signal after the driver, and the $\overline{\mathsf{CS}}$ and clock drivers should be permanently on ($\overline{\mathsf{enable}}$ connected to ground). The 74HCT...series treats all signals above 2 Volts as logical HIGH, so it has no problems with 3.3 Volt signals.

Printer ports which send 3.3 Volt signals (but can receive 5 Volt signals without problems) are quite common in modern PCs. TTL chips like 74HCT...have no problems with this, but CMOS chips and compatibles define HIGH as at least two thirds of the supply voltage, and 3.3 Volts are just below two thirds of 5 Volts. Pin assignment of 74HCT125: 1 nen

Circuity for printer port data acquisition module

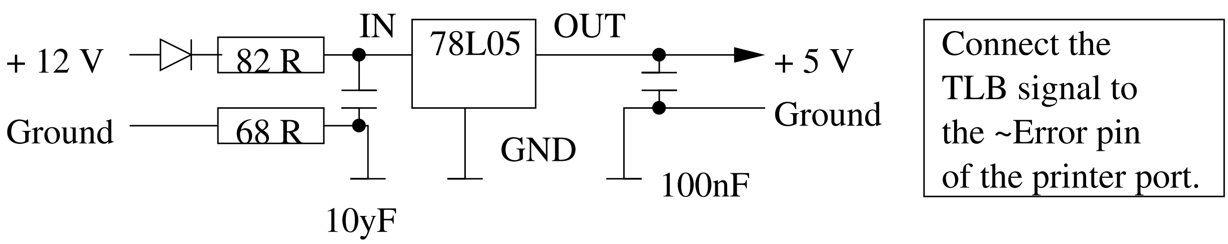
The four buttons and two A/D converters are all connected to six different "Data to PC" lines as defined in tracker.h





(*) Zener diodes to limit input range / protect chip.
Not needed if Vref is +5 V. Use sockets for the LTC1286s.

Voltage regulation circuit, to get 5 Volt supply from a 12 Volt supply:



A, 2 in A, 3 out A, 4 nen B, 5 in B, 6 out B, 7 ground, 14 +5 Volt, 13 nen D, 12 in D, 11 out D, 10 nen C, 9 in C, 8 out C (nen meaning not-enable).

If you prefer not to build the described device, you can also buy serial A/D converter modules which connect to the serial port (the control lines of it, not the normal data lines - the A/D converter uses no standard timing and word size, so you still have to convert the serial data stream back to parallel in software. In addition, you have to connect the button box to some other port. I recommend the joystick port, which provides four digital (0/5 Volts) inputs and a simple power supply. The four simple A/D converters of the joystick port are too bad to be used for eyetracker signals – apart from maybe sampling the (digital) track loss / blink signal with them (simply because you have no fifth digital input on the joystick port). You will have to modify NEWTRACK in order to be able to use the described serial port modules and printer port button box connection.

5.6 NEWTRACK usage

At startup, NEWTRACK displays a message about itself and the built-in hardware drivers:

```
NEWTRACK eye tracking software - (c) Eric Auer 2004/2005 - License: GPL 2

  Buttons:    parport, keyboard (enable with scroll-lock) or CIO.
  ADC:        2 LTC1286 (parport), mouse (simulation) or DAS16.
  Dual-head:  2 PCI/AGP, interlaced (simulation) or mono+color.
  Graphics:   VESA 1024x768 mode, PCX file display.
  ySec clock: Processor TSC or normal PC timer chip.
```

Type `NEWTRACK /?` to display the help screen, which is also displayed when an argument syntax error is detected:

```
Allowed arguments are: graphics (enables graphics mode)
  subject=name     (define subject name, no spaces allowed)
  script=filename  (use specified script file)
  logfile=filename (defaults to SUBJECTNAME.log)
Alternative syntax is using 'flip' as the only argument,
  to switch between the available PCI/AGP graphics cards
  (does not start the experiment but returns at once)
```

On most systems, NEWTRACK will automatically detect the current hardware configuration. In some cases, you are given the chance to reject using a detected device. For example if you do have an eyetracker connected, you can still opt to use the mouse instead of the eyetracker. This is possible to allow you to do simulation runs without a subject even on the PC which is connected to the eyetracker. For debugging, you can suppress the usage of two VGA cards by having ScrollLock active when NEWTRACK initializes, and you can work without eye coordinates if you abort the calibration by pressing Ctrl-D.

If you do not specify any command line arguments, you are asked for a subject name and a stimulus script file name. The log file name is derived from the subject name. When all file names and the subject name are known, the button and coordinate values from the eyetracker are displayed until you press the CAL button (all buttonbox buttons have keyboard aliases, which are displayed at the startup screen). This starts the initial calibration: Press the NEXT button to make the next fixation mark show up and store the coordinates of the current fixation mark. All coordinates are displayed on the operator screen. When you enter calibration mode again later (with the CAL button), the fixation marks will be displayed in randomized order. After calibration, the calculated correction factors are displayed. Now you can either accept with the NEXT button or calibrate again with the CAL button. In the second and subsequent calibrations, an eye cursor is displayed based on the correction factors of the previous calibration.

After the initial calibration, the experiment starts. For normal stimuli, the NEXT button triggers the change from a test pattern to the stimulus display. For gaze activation stimuli, the subject has to stare at a fixation mark (which is located at the same place where the first word of the stimulus will start) to trigger the display. You can bypass the gaze trigger by pressing the NEXT button. While a stimulus is shown, the looked-at words are highlighted on the operator screen in real-time. Most logged events are shown in a status bar on the operator screen, as well as information about the elapsed time. For each stimulus, either the timeout or a button press by the subject ends the stimulus display. When all stimuli have been shown, the log file is saved and the program exits. You can abort single trials or the whole experiment with hotkeys. If you abort a trial by entering calibration mode, the trial will be repeated after the calibration.