Chapter 9
Table-Mounted System Hardware Installation

This chapter focuses on installation of a fourth-generation table-mounted eye tracker (cf. the analog-video, or third-generation table-mounted eye tracker discussed in Chap. 6). Although the modern counterpart is based on similar underlying principles as previous generation technology (video-based, corneal-reflection eye tracking), it is considerably easier to transport, install, and use. The table-mounted eye tracker may appear no different from a common flat panel display, and that is intentional. Unlike a typical monitor, however, a camera and infra-red LED optics are embedded beneath the LCD flat panel. The eye tracker on which this discussion is based is commercially available from Tobii Technology AB, based in Stockholm, Sweden. The dual-head installation at Clemson University, on which this and the remaining table-mounted system chapters are based, is shown in Fig. 9.1. The particular hardware devices installed at Clemson are described here for reference.

Each of Clemson's eye tracking stations is centered on Tobii's ET-1750 eye tracker, which operates (samples) binocularly at 50 Hz at an accuracy of about 0.5° (bias error). The Tobii display is a 17 in. TFT flat panel running at 1280 × 1024 resolution. Tobii's eye tracking server runs on a dual 2.0 GHz AMD Opteron 246 Sun W2100z with 2 GB RAM running Windows XP. Its display is driven by an NVidia NVS280 graphics card. The application computer is a 2.2 GHz AMD Opteron 148 Ultra 20 with 1 GB RAM running Fedora Core 4 (Linux 2.6.11 with gcc v4.0.0). It is equipped with an NVidia GeForce 7800 GTX graphics card. Both computers are connected to a 1 Gb ethernet LAN. Both client and server machines are connected to a Belkin OmniView SOHO Series (F1DS104T) 4-Port KVM switch with audio PS/2 and USB support (firmware v2.0 7/7/2005). The KVM switch allows sharing of the keyboard, display, and mouse between the two computers.

A.T. Duchowski, *Eye Tracking Methodology*, DOI 10.1007/978-3-319-57883-5_9

Fig. 9.1 Tobii dual-head eye tracking stations at Clemson University

9.1 Integration Issues and Requirements

The salient characteristic of the table-mounted eye tracker is its calculation and delivery of instantaneous (x, y) coordinates of the the user's gaze. Computing applications receiving this signal can then be developed to make use of this real-time information in a number of different ways (see Chaps. 21–24). The most basic application is diagnostic in nature where the application collects gaze coordinates while the user is watching some form of stimulus (e.g., images, video, Web pages, desktop). Following eye movement (signal) analysis, one can infer the user's attentional strategies and in turn the stimuli's attentional qualities. Diagnostic applications are thus off-line because the display generally does not change contingent on the user's gaze location.

In contrast, interactive applications can make use of the viewer's real-time gaze location by either changing their appearance in some way or by directly invoking the user's gaze to affect control of the application. The former type of passive interface is termed *gaze-contingent* and is often used to evaluate characteristics of the human visual system. For example, one can provide imagery degraded in the user's peripheral visual field. If the user reports no perceived effect, then such a gaze-contingent display has successfully matched the human visual system's resolving capacity. The latter type of active interface allows the user to control aspects of the interface directly through eye movements. The classic example is using the (x, y) gaze coordinates in place of a manual mouse.

In general, most eye tracking devices allow the following functionality.

1. Connection: establish connection with the eye tracker (e.g., serial port or TCP/IP).
2. Calibration: display calibration points at the appropriate location and time.
3. Synchronization: display stimulus at the appropriate time (the eye tracker should be able to inform the application program of its state, or vice versa).
4. Data streaming: use eye tracker to capture data and/or update the stimulus scene in a gaze-contingent manner.

Two of the four key integration concerns identified in Chap. 6 have been for the most part removed, namely:

1. The capability of the eye tracker to provide fine-grained cursor control
2. The capability of the eye tracker to transmit its operating mode along with gaze (x, y) coordinates

Elimination of these two concerns is one of several reasons for the eye tracker's usability improvement. The fine-grained cursor control was previously used to obtain a mapping of the gaze coordinates over the display. This is still important, however, the Tobii transmits normalized gaze coordinates, i.e., in the range [0, 1]. Thus, mapping from the eye tracker's reference frame to a given application's is performed by simply scaling the normalized gaze coordinates by the extents of the application window. This greatly simplifies manipulation of gaze coordinates by the client application.

The second point concerning the tracker's operating mode has also been removed mainly due to the transfer of mode control to the application. That is, although now the host/client roles have been reversed, i.e., the Tobii is referred to as the server and the (interactive) application is now considered the client, it is the client application that controls what mode the eye tracker is in, i.e., idle, calibrating, or running. This transfer of control makes development of interactive applications considerably easier than with previous equipment. With older technology, a developer often needed a (very) patient secondary individual to serve as viewing subject when testing an eye tracking application. It was very difficult to self-calibrate because one had to control the eye tracker from a dedicated console, while operating the application from another. Two keyboards, mice, etc. were needed. Transferring control to the software allows the client application to control the eye tracker from within. Thus, self-calibration is now easy to perform.

The two other major integration concerns still exist, but due to Tobii's selection of a particular standardized display technology and release of Software Development Kit (SDK), the concerns have been greatly alleviated. They are:

1. Knowledge of the video format the eye tracker requires as input (e.g., currently VGA, hence no longer an issue)
2. Knowledge of the data format the eye tracker generates as its output (provided by the SDK reference)

Note that these two concerns, and particularly knowledge of the gaze data format and other details contained within the SDK, are mainly directed toward development of gaze-aware applications. This means that these issues are more of a concern for developers of interactive applications, or for those who wish to reinstrument their applications for off-line eye movement diagnostic analysis. For users who simply wish to evaluate currently available desktop applications (currently restricted to the Windows environment), images, Web pages, or videos, Tobii provides an even simpler solution with their ClearView software. ClearView is simply a Windows-based client application that Tobii has written that can collect eye movements over various forms of stimuli including Web pages (using Tobii's reinstrumented browser) and the Windows desktop (sacrificing a portion of the display refresh rate while obtaining data in this "screen" mode).

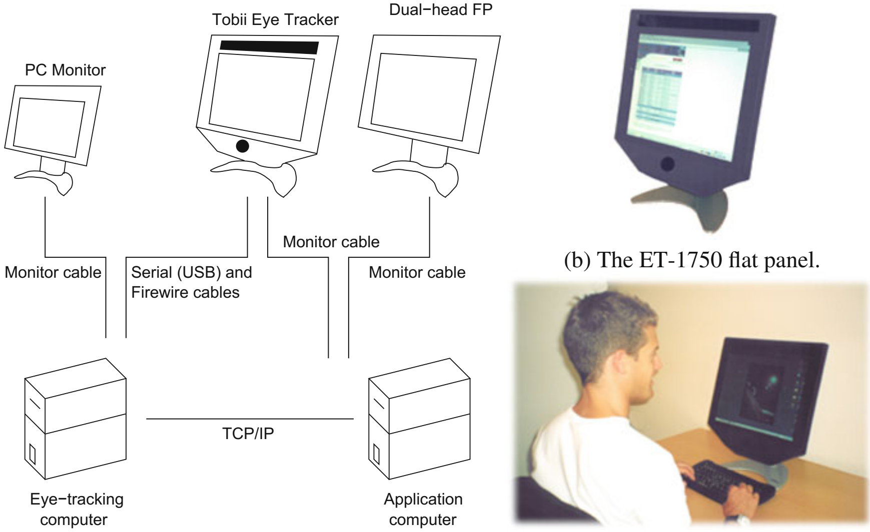
For diagnostic users of eye tracking technology, Tobii (as well as other manufacturers) have made great strides toward "plug-and-play" usability. Following basic installation, or rather simply computer-display connection (described below) and installation of eye tracking software, the experimenter is ready to begin collecting eye movements. The remaining details of Tobii's data format, SDK, etc. generally concern those wishing to develop their own gaze-aware applications (e.g., gaze-contingent displays).

9.2 System Installation

The Tobii ET-1750 eye tracker can be configured in several ways, one of which is acting as a server for a possibly remote eye tracking client application. Connecting over TCP/IP, a client application controls user calibration and then synchronizes with the eye tracker through Application Program Interface (API) callbacks. In the simplest case, both the client and server reside on the same machine; e.g., this would be typical of a single-PC such as a laptop running Windows, connected to the eye tracking display and camera and infra-red light optics.

An example configuration more suitable for development on a secondary application computer (e.g., Linux or Mac workstation or laptop) connected to the Tobii eye tracker is shown in Fig. 9.2. Figure 9.2b shows the eye tracker display in isolation. Figure 9.2a shows the behind-the-scenes connections of the eye tracking station. An application computer drives the display's LCD flat panel. This computer is connected (via Ethernet) to the eye tracker computer processing images obtained from the camera embedded in the display. The eye tracking server also connects to the eye tracker, but not to its display, only its camera (e.g., via the IEEE 1394 or "firewire" interface) and infra-red lights (e.g., via the USB serial interface).

Note that the configuration in Fig. 9.2 is a triple-head eye tracking station (in comparison, Fig. 9.1 only shows a two-headed eye tracking station). Using all three monitors provides the simultaneous advantage of a wide-screen dual-head application (e.g., Linux) display and a monitor for the eye tracking server. The eye tracking server display may be somewhat extraneous, particularly if the server runs as a background process and does not offer much in the way of a status display. Indeed, the lack of server status is expected because such status information should be obtained and displayed by the eye tracking client application. This is typical of the traditional



(a) Schematic configuration.

(b) The ET-1750 flat panel.

(c) Typical use.

Fig. 9.2 Single Tobii eye tracking station hardware setup. From Ashmore et al. (2005) © 2005 Canadian Information Processing Society. Reprinted by permission

client/server programming paradigm. For the triple-headed installation, both the eye tracking server and application client computers should be equipped with dual-headed graphics cards if one wishes to also perform diagnostic eye tracking work on the server machine.

Although the triple-headed installation may be somewhat of an overkill, eventually one must interact with the operating system on which the eye tracking server runs, and therefore, one must be able to connect the display to the eye tracking server. If a small monitor is not available, then the alternative is to share the eye tracking display between both server and client computers via a KVM, or Keyboard-Video-Mouse switch. The KVM switch is omitted from the schematic in Fig. 9.2, as are the keyboard and mouse. The KVM switch is visible in Fig. 9.1, however.

9.3 Lessons Learned from the Installation at Clemson

Compared to the previous eye tracking installation (see Chap. 6), setting up the current workstations was almost trivial, if not pleasurable. However, just as the previous installation had one troublesome component (video cable pinout differing between manufacturers), this installation was also not entirely trouble-free. Note that the video signal is still a concern, particularly if working with different manufacturers, e.g., Apple, who used to use specialized Apple Display Connectors. Contemporary video cards should support VGA, perhaps requiring a DVI-VGA adaptor (usually included with the display). If Tobii upgrades the display to DVI, the adapter would not be needed. Speculations aside, the video signal was a minor issue.

The component that required an appreciable amount of cajoling in the modern installation was the KVM switch. The specific KVM switch used shipped with a bug causing erroneous recognition of the mouse input when switching between Linux and Microsoft Windows. The symptom was a frozen mouse cursor requiring a reboot to unfreeze. The firmware update made available by the KVM switch manufacturer eventually alleviated the problem, but firmware installation was itself most annoying. The problem here was extremely poor feedback provided by the firmware upgrade utility. No status was given as the firmware was loaded and upon completion only an error was given. The error was apparently commonplace (as reported by technical support) and was meant to be ignored. However, to install the firmware, one had to disconnect all other signals to switch and connect only the computer loading the firmware. What's worse, the loading computer had to have a parallel port, which is something of a rarity. Thus, an older laptop had to be found which could install the firmware upgrade after everything else was disconnected.

Apart from the KVM switch annoyance, other pointers worth remembering are the typical requirements of sufficient power, network, keyboard, mouse, etc. When installing multiple multiheaded stations such as at Clemson (three stations are available for students), six network jacks were required. Depending on available IT support, this may or may not be a problem (at the Clemson installation, these jacks were not available initially, and a network switch was required).

9.4 Summary and Further Reading

This chapter presented key points for installation of a modern, table-mounted eye tracker. As with earlier head-mounted systems, successful installation still depends on appropriate signal routing and synchronization. Modern systems have simplified these concerns, however, particularly if signal acquisition is based on a networked client/server model. Rather than relying on the interpretation of vendor-specific serial data, communication over TCP/IP facilitates faster development due to the reliance of accepted standards, e.g., socket programming. Although the data received may still be specific to a particular vendor's format and protocol, their receipt from a socket stream is much simpler than byte assembly of a serial interface. An added benefit is the lifting of the proximity constraint placed by the length of a serial cable.

Video signal routing is still a (minor) concern, although this too has been simplified by the adoption of standardized displays, e.g., VGA. The installation of a three-headed system, described above, made simple by the use of a KVM switch, is a relatively complex one. Typical eye tracking "stations" need not be this elaborate and can for the most part resemble a typical computer workstation: computer and monitor along with eye tracking optics (possibly embedded).

Apart from hardware issues, the use of normalized screen coordinates (e.g., $(x, y) \in [0, 1]$) removes the problematic need for reference frame mapping, typically

(a) Classroom with 20 Gazepoint trackers. (b) Close up of Gazepoint tracker mount.

Fig. 9.3 Eye tracker classroom installation with 20 eye trackers from Gazepoint

done in software, or at least the more daunting portion of coordinate registration that was needed previously. Given normalized point of regard coordinates, these are simply scaled to a given display or window over which eye movements are collected.

Since the typical installation of one or two eye trackers in a lab, a recent trend is on developing larger eye tracking labs or classrooms, e.g., see installation at Clemson shown in Fig. 9.3. Other similar classrooms have appeared at the SWPS University of Social Sciences and Humanities in Warsaw, Poland, the Slovak University of Technology in Bratislava, Slovakia, the University of the Free State in Bloemfontein, South Africa, and Lund University in Lund, Sweden. One of the main issues with such installations is synchronization among multiple eye trackers. Collecting data from multiple eye trackers simultaneously opens up new possibilities for interesting research and applications.

As with previous systems, the two primary sources where further information can be obtained on system installation and setup are the manufacturer's manual and any user community groups that may have assembled. Users of eye trackers still report descriptions of their apparatus and any specific technical innovations required for system development, specifically to foster replication of their experiments by other researchers. Reports can still be found in journal articles such as *Vision Research*, *Behavior Research Methods, Instruments, and Computers (BRMIC)*, and conference proceedings. There are various conferences that deal with eye tracking, either directly or indirectly. For example, conferences that deal with computer graphics (e.g., SIGGRAPH, EuroGraphics, or Graphics Interface), human–computer interaction (e.g., SIGCHI), or virtual reality (e.g., VRST), still carry papers that discuss the use of eye trackers. Two main conferences dealing directly with eye tracking run on an interleaved biennial basis: the European Conference on Eye Movements (ECEM), and the U.S.-based Eye Tracking Research & Applications (ETRA). Finally, the eye movement email listservs *eye-movements* and *eyemov-l* are excellent on-line "gathering places" of eye tracker researchers.