Calibration of the VBEGT system camera is required to set a number of operational parameters, which relate to image capture and the mechanism used to transfer it to the VBEGT system. These parameters are listed in Table 4.5.

User calibration is considered to be within the domain of VBEGT system calibration as the physical attributes of the user are used by the system during feature extraction and to determine the user’s visual line of gaze. These parameters include the physical characteristics of the user, such as the distance at which the user sits from the computer monitor and the size of their iris at this distance (see Table 4.6). In addition to these characteristics, the VBEGT system requires three calibration points to show the distance travelled by the iris when looking from the screen centre to top right and bottom left corners (see Figure 4.28 and Figure 4.29). These points are used when determining the user’s visual line of gaze to limit the degree of systematic error that may be exhibited by the gaze points determined by the VBEGT system during operation.

![Three calibration points mark the distance travelled by the user’s iris as they traverse from the centre of their computer monitor to either side.](image)

Figure 4.28: Three calibration points mark the distance travelled by the user’s iris as they traverse from the centre of their computer monitor to either side.
**Figure 4.29:** Calibration grid used to determine the calibration points used for a given User Configuration Script.

**Table 4.4:** Description of the VBEGT system software variables that require configuration/calibration prior to testing.

<table>
<thead>
<tr>
<th>Variable</th>
<th>Description</th>
<th>Accepted Values</th>
</tr>
</thead>
<tbody>
<tr>
<td>Image Threshold Value</td>
<td>Used to determine whether a given pixel is termed 'black' (part of the current line segment) or 'white' (not a part of the current Line Segment). This value may be influenced by the colour of the user's iris.</td>
<td>0-255</td>
</tr>
<tr>
<td>Fixation Number</td>
<td>The number of gaze points within a specified vicinity that denote whether a user is fixating upon a given point or the eye is still in transit from one point of fixation to another.</td>
<td>0 – maximum value of the unsigned integer data type^36</td>
</tr>
<tr>
<td>Fixation Resolution</td>
<td>The margin that determines whether consecutive gazes are centred about the same point.</td>
<td>0 – width or height of the image depending on which value is larger.</td>
</tr>
<tr>
<td>Line Segment Deviation</td>
<td>The deviations between the start and end positions of a given Line Segment. Used when matching line segments to hypotheses.</td>
<td>0 – width of the image</td>
</tr>
</tbody>
</table>

^36 Depending on the operating system and compiler the maximum value of an integer may vary due to the length of the data type. On 32-bit systems an integer is 32 bits long, therefore an unsigned integer (i.e., one supporting only positive values) may have a value from 0 to 4294967295 (2^{32} - 1)
<table>
<thead>
<tr>
<th>Variable</th>
<th>Description</th>
<th>Accepted Values</th>
</tr>
</thead>
<tbody>
<tr>
<td>Line Segment Pixel Threshold</td>
<td>Represents the maximum number of pixels that may be missed before a break in that Line Segment occurs.</td>
<td>0 – width of the image</td>
</tr>
<tr>
<td>Minimum Line Segment Length</td>
<td>The minimum size of a line segment retrieved from a given image row.</td>
<td>0 – width of the image</td>
</tr>
<tr>
<td>Lines To Skip</td>
<td>The number of rows in the image to skip during processing. For example, setting this value to the number 2 would process every second line of the image.</td>
<td>0 – height of the image</td>
</tr>
<tr>
<td>Eye Data Array Size</td>
<td>The size of the array storing recent eye movement. This array is used when determining the current state of the eye, i.e., moving (saccade) or focused (fixation)</td>
<td>0 – maximum value of the unsigned integer data type</td>
</tr>
<tr>
<td>Left Screen Position</td>
<td>The left most screen position, in pixels, marking the edge of the user's screen.</td>
<td>0 – maximum horizontal resolution of the user's monitor</td>
</tr>
<tr>
<td>Right Screen Position</td>
<td>The right most screen position, in pixels, which marks the edge of the user's screen.</td>
<td>0 – maximum horizontal resolution of the user's monitor</td>
</tr>
<tr>
<td>Top Screen Position</td>
<td>The top most screen position, in pixels, which marks the edge of the user's screen.</td>
<td>0 – maximum vertical resolution of the user's monitor</td>
</tr>
<tr>
<td>Bottom Screen Position</td>
<td>The bottom most screen position, in pixels, marking the edge of the user's screen.</td>
<td>0 – maximum vertical resolution of the user's monitor</td>
</tr>
</tbody>
</table>
Table 4.5: Description of the VBEGT system camera variables that require configuration/calibration prior to testing.

<table>
<thead>
<tr>
<th>Variable</th>
<th>Description</th>
<th>Accepted Values</th>
</tr>
</thead>
<tbody>
<tr>
<td>Maximum Exposure Time</td>
<td>The maximum exposure time in milliseconds that may be applied to an image before reducing the camera's frame rate to a value below the minimum acceptable frame rate specified for this camera.</td>
<td>0 - 1000</td>
</tr>
<tr>
<td>Decimation</td>
<td>The decimation value (no decimation, decimate by 2, or decimate by 4) used by the camera during image captures. This value affects the size of the final image captured (see description for sub-window width and height).</td>
<td>No decimation:</td>
</tr>
<tr>
<td></td>
<td>PXL_NO_DECIMATION</td>
<td>Decimate by 2:</td>
</tr>
<tr>
<td></td>
<td>PXL_DECIMATE_BY_2</td>
<td>Decimate by 4:</td>
</tr>
<tr>
<td></td>
<td>PXL_DECIMATE_BY_4</td>
<td></td>
</tr>
<tr>
<td>Data Transfer Size</td>
<td>The size of the pixel information returned by the camera. The data transfer size selected (8 or 16 bit) affects the clock rate selected. 8 bits per pixel (256 shades of red, green and blue) is the default for this study. 16 bits per pixel is not supported.</td>
<td>8 bits per pixel data size:</td>
</tr>
<tr>
<td></td>
<td>DATA_8BIT_SIZE</td>
<td>16 bits per pixel data size:</td>
</tr>
<tr>
<td></td>
<td>DATA_16_BIT_SIZE</td>
<td></td>
</tr>
<tr>
<td>Clock Rate</td>
<td>The clock frequency set for the imaging device (24, 16, 12, 8, 6 or 4 MHz). When the image transfer size is set to 16 bits a clock frequency of 24 MHz will crash the system.</td>
<td>4 MHz clock frequency:</td>
</tr>
<tr>
<td></td>
<td>PXL_4MHZ</td>
<td>6 MHz clock frequency:</td>
</tr>
<tr>
<td></td>
<td>PXL_6MHZ</td>
<td>8 MHz clock frequency:</td>
</tr>
<tr>
<td></td>
<td>PXL_8MHZ</td>
<td>12 MHz clock frequency:</td>
</tr>
<tr>
<td></td>
<td>PXL_12MHZ</td>
<td>16 MHz clock frequency:</td>
</tr>
<tr>
<td></td>
<td>PXL_16MHZ</td>
<td>24 MHz clock frequency:</td>
</tr>
<tr>
<td></td>
<td>PXL_24MHZ</td>
<td></td>
</tr>
<tr>
<td>Sub-window Width</td>
<td>The width of the sub-window in pixels for the captured image. Note this value does not denote the width of the final image as image width is defined as: image width = sub-window width / decimation value.</td>
<td>PXL_MIN_WIDTH – PXL_MAX_WIDTH</td>
</tr>
</tbody>
</table>

37 Sub-window left position + sub-window width must be less than – PXL_MAX_WIDTH. The sub-window width must be greater than PXL_MIN_WIDTH.
<table>
<thead>
<tr>
<th>Variable</th>
<th>Description</th>
<th>Accepted Values</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sub-window Height</td>
<td>The height of the sub-window in pixels used for image captures. Note this value does not denote the final height of the captured image, as: image height = sub-window height / decimation value.</td>
<td>PXL_MIN_HEIGHT - PXL_MAX_HEIGHT</td>
</tr>
<tr>
<td>Sub-window Left Position</td>
<td>The position of the sub-window within the camera's field of view (1280 x 1024 pixels) in pixels from the left side of the imager (camera) window.</td>
<td>PXL_MIN_WIDTH - PXL_MAX_WIDTH</td>
</tr>
<tr>
<td>Sub-window Top Position</td>
<td>The position of the sub window, in pixels, from the top of the imager (camera) window.</td>
<td>PXL_MIN_HEIGHT - PXL_MAX_HEIGHT</td>
</tr>
<tr>
<td>Capture Mode</td>
<td>Defines the camera mode used to capture images. The camera selected for this study is able to operate in either video mode or still mode.</td>
<td>Video:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>CAPTURE_MODE_VIDEO_MODE</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Still Mode:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>CAPTURE_MODE_STILL_MODE</td>
</tr>
<tr>
<td>Colour Mode</td>
<td>The colour conversion applied to the image before processing i.e., the image is converted to an RGB24 colour image, RGB24 greyscale image or RGB24 black and white image.</td>
<td>RGB24 colour image:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>IMAGE_COLOUR</td>
</tr>
<tr>
<td></td>
<td></td>
<td>RGB24 greyscale image:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>IMAGE_GREYSCALE</td>
</tr>
<tr>
<td></td>
<td></td>
<td>RGB24 black and white image:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>IMAGE_BLACK_AND_WHITE</td>
</tr>
</tbody>
</table>

38 Sub-window top position + sub-window height must be less than – PXL_MAX_HEIGHT. The sub-window width must be greater than PXL_MIN_HEIGHT.
Table 4.6: Description of the VBEGT system user variables that require configuration/calibration prior to testing.

<table>
<thead>
<tr>
<th>Variable</th>
<th>Description</th>
<th>Accepted Values</th>
</tr>
</thead>
<tbody>
<tr>
<td>Maximum Iris Radius</td>
<td>The maximum size for a circle (in pixels) that may be recommended to the VBEGT system as the user’s iris.</td>
<td>0 – width of the image</td>
</tr>
<tr>
<td>Minimum Iris Radius</td>
<td>The minimum size (in pixels) for a circle that may be recommended to the VBEGT system as the user’s iris.</td>
<td>0 - width of the image</td>
</tr>
<tr>
<td>Top Left Calibration Top</td>
<td>The top most point of a rectangle encasing the user’s iris. The coordinates of this position are relative to the position of the iris in a given eye image when the user is looking at the top left position on a given calibration grid.</td>
<td>0 – Top Left Calibration Bottom</td>
</tr>
<tr>
<td>Top Left Calibration Bottom</td>
<td>The bottom most point of a rectangle encasing the user’s iris. The coordinates of this position are relative to the position of the iris in a given eye image when the user is looking at the top left position on a given calibration grid.</td>
<td>Top Left Calibration Top – width of the image</td>
</tr>
<tr>
<td>Top Left Calibration Left</td>
<td>The left most point of a rectangle encasing the user’s iris. The coordinates of this position are relative to the position of the iris in a given eye image when the user is looking at the top left position on a given calibration grid.</td>
<td>0 – Top Left Calibration Left</td>
</tr>
<tr>
<td>Top Left Calibration Right</td>
<td>The right most point of a rectangle encasing the user’s iris. The coordinates of this position are relative to the position of the iris in a given eye image when the user is looking at the top left position on a given calibration grid.</td>
<td>Top Left Calibration Left – height of the image</td>
</tr>
<tr>
<td>Bottom Right Calibration Top</td>
<td>The top most point of a rectangle encasing the user’s iris. The coordinates of this position are relative to the position of the iris in a given eye image when the user is looking at the bottom right position on a given calibration grid.</td>
<td>0 – Bottom Right Calibration Bottom</td>
</tr>
<tr>
<td>Variable</td>
<td>Description</td>
<td>Accepted Values</td>
</tr>
<tr>
<td>----------</td>
<td>-------------</td>
<td>-----------------</td>
</tr>
<tr>
<td>Bottom Right Calibration Bottom</td>
<td>The bottom most point of a rectangle encasing the user’s iris. The coordinates of this position are relative to the position of the iris in a given eye image when the user is looking at the bottom right position on a given calibration grid.</td>
<td>Bottom Right Calibration Top – height of the image</td>
</tr>
<tr>
<td>Bottom Right Calibration Left</td>
<td>The left most point of a rectangle encasing the user’s iris. The coordinates of this position are relative to the position of the iris in a given eye image when the user is looking at the bottom right position on a given calibration grid.</td>
<td>0 – Bottom Right Calibration Right</td>
</tr>
<tr>
<td>Bottom Right Calibration Right</td>
<td>The right most point of a rectangle encasing the user’s iris. The coordinates of this position are relative to the position of the iris in a given eye image when the user is looking at the bottom right position on a given calibration grid.</td>
<td>Bottom Right Calibration Left – width of the image</td>
</tr>
<tr>
<td>Centre Calibration Top</td>
<td>The top most point of a rectangle encasing the user’s iris. The coordinates of this position are relative to the position of the iris in a given eye image when the user is looking at the centre position on a given calibration grid.</td>
<td>0 – Centre Calibration Bottom</td>
</tr>
<tr>
<td>Centre Calibration Bottom</td>
<td>The bottom most point of a rectangle encasing the user’s iris. The coordinates of this position are relative to the position of the iris in a given eye image when the user is looking at the centre position on a given calibration grid.</td>
<td>Centre Calibration Top – width of the image</td>
</tr>
<tr>
<td>Centre Calibration Left</td>
<td>The left most point of a rectangle encasing the user’s iris. The coordinates of this position are relative to the position of the iris in a given eye image when the user is looking at the centre position on a given calibration grid.</td>
<td>0 - Centre Calibration Right</td>
</tr>
<tr>
<td>Centre Calibration Right</td>
<td>The right most point of a rectangle encasing the user’s iris. The coordinates of this position are relative to the position of the iris in a given eye image when the user is looking at the centre position on a given calibration grid.</td>
<td>Centre Calibration Left – width of the image</td>
</tr>
</tbody>
</table>
Configuration and calibration of the test environment is essential for operational system testing. The test environment is composed of the MUM, MSG and TI. The MSG provides Movement Scripts used to control the movement of the MUM during testing. It is essential that the MUM is calibrated/configured correctly so that the EMM and HMM are able to move to the positions specified by the Movement Scripts. EMM and HMM movement uses a trigonometry based approximation for the number of steps required to move to a given point based upon the distance of the MUM from the computer monitor, the dimensions of the monitor’s display area (height and width) and the physical characteristics (i.e., resolution) of the MUM (see Table 4.7). The resolution of the MUM is defined by the accuracy of the stepper motors used to construct it. Table 4.8 outlines the resolution of these stepper motors.

Table 4.7: Description of the MUM variables that require configuration/calibration prior to operational system testing.

<table>
<thead>
<tr>
<th>Variable</th>
<th>Description</th>
<th>Accepted Values</th>
</tr>
</thead>
<tbody>
<tr>
<td>Active Screen Width</td>
<td>The width of the user’s screen in pixels, i.e., the horizontal resolution of the user’s computer monitor. For example, if the screen resolution is 640 x 480 the active screen width is 640.</td>
<td>1 to the width of the user’s desktop is pixels.</td>
</tr>
<tr>
<td>Active Screen Height</td>
<td>The height of the user’s screen in pixels, i.e., the vertical resolution of the user’s computer monitor. For example, if the screen resolution is 640 x 480 the active screen height is 480.</td>
<td>1 to the height of the user’s desktop is pixels.</td>
</tr>
<tr>
<td>X Screen Divisions</td>
<td>The number of horizontal divisions a screen may be divided into for user input.</td>
<td>1 to Active Screen Width</td>
</tr>
<tr>
<td>Y Screen Divisions</td>
<td>The number of vertical divisions a screen may be divided into for user input.</td>
<td>1 to Active Screen Height</td>
</tr>
<tr>
<td>Variable</td>
<td>Description</td>
<td>Accepted Values</td>
</tr>
<tr>
<td>-------------------------------</td>
<td>------------------------------------------------------------------------------</td>
<td>----------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Distance From The Screen</td>
<td>The distance between the Eye Model and the computer monitor in millimetres.</td>
<td>0 - maximum value of the unsigned integer data type</td>
</tr>
<tr>
<td>Physical Screen Width</td>
<td>The width of the computer screen in millimetres</td>
<td>0 - maximum value of the unsigned integer data type</td>
</tr>
<tr>
<td>Physical Screen Height</td>
<td>The height of the computer screen in millimetres</td>
<td>0 - maximum value of the unsigned integer data type</td>
</tr>
<tr>
<td>Starting Division</td>
<td>The division number upon which the Eye Model gazes at the beginning of each test.</td>
<td>0 to ((Active Screen Width - Active Screen Height) -1)</td>
</tr>
<tr>
<td>Eye X Step Range</td>
<td>The maximum number of horizontal steps that the Eye Model may travel from left to right.</td>
<td>131</td>
</tr>
<tr>
<td>Eye Y Step Range</td>
<td>The maximum number of vertical steps that the Eye Model may travel from top to bottom.</td>
<td>78</td>
</tr>
<tr>
<td>Eye X Degrees Per Step</td>
<td>The number of horizontal degrees travelled per step.</td>
<td>0.9</td>
</tr>
<tr>
<td>Eye Y Degrees Per Step</td>
<td>The number of vertical degrees travelled per step.</td>
<td>0.9</td>
</tr>
<tr>
<td>Head X Step Range</td>
<td>The maximum number of horizontal steps that the Head Model may travel from left to right.</td>
<td>23510</td>
</tr>
<tr>
<td>Head Y Step Range</td>
<td>The maximum number of vertical steps that the Head Model may travel from top to bottom.</td>
<td>13367</td>
</tr>
<tr>
<td>Head Z Step Range</td>
<td>The maximum number of steps that the Head Model may travel back and forth.</td>
<td>14557</td>
</tr>
<tr>
<td>Head X Distance Per Step</td>
<td>The number of horizontal degrees travelled per step.</td>
<td>0.01</td>
</tr>
<tr>
<td>Head Y Distance Per Step</td>
<td>The number of vertical degrees travelled per step.</td>
<td>0.005</td>
</tr>
<tr>
<td>Head Z Distance Per Step</td>
<td>The number of degrees travelled per step.</td>
<td>0.01</td>
</tr>
</tbody>
</table>

39 Defined by the physical characteristics of the MUM (see Table 4.8).
Table 4.8: Physical characteristics of the MUM (Holme, 2003).

<table>
<thead>
<tr>
<th>Degree of Freedom</th>
<th>Range in Steps</th>
<th>Displacement per Step</th>
</tr>
</thead>
<tbody>
<tr>
<td>Head Movement Model:</td>
<td></td>
<td></td>
</tr>
<tr>
<td>X Stepper</td>
<td>23510</td>
<td>0.010mm</td>
</tr>
<tr>
<td>Y Stepper</td>
<td>13367</td>
<td>0.005mm</td>
</tr>
<tr>
<td>Z Stepper</td>
<td>14557</td>
<td>0.010mm</td>
</tr>
<tr>
<td>Eye Movement Model:</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Pan Stepper</td>
<td>131</td>
<td>0.9 degrees</td>
</tr>
<tr>
<td>Tilt Stepper</td>
<td>78</td>
<td>0.9 degrees</td>
</tr>
</tbody>
</table>

Table 4.9: Description of the Movement Script Generator variables that require configuration/calibration prior to operational system testing.

<table>
<thead>
<tr>
<th>Variable</th>
<th>Description</th>
<th>Accepted Values</th>
</tr>
</thead>
<tbody>
<tr>
<td>Max X Divisions</td>
<td>The maximum number horizontal divisions that the user's screen may be divided into for input.</td>
<td>0 - maximum value of the unsigned integer data type</td>
</tr>
<tr>
<td>Maxy Divisions</td>
<td>The maximum number vertical divisions that the user's screen may be divided into for input.</td>
<td>0 - maximum value of the unsigned integer data type</td>
</tr>
<tr>
<td>Number of MUM Movements</td>
<td>The number of MUM movements generated per Test Script.</td>
<td>0 - maximum value of the unsigned integer data type</td>
</tr>
</tbody>
</table>
4.4 Summary

In order to provide an improvement over existing VBEGT systems, this study required a high-resolution camera that supported high image transfer rates. In order to achieve maximum response times the processing algorithms developed were developed such that they made use of the features provided by the selected camera and maintained an efficient design so as to maximise system performance in an attempt to satisfy the real-time system requirements.

Determining the feasibility of VBEGT for mainstream computer use, required the development of a prototype VBEGT system and suitable test apparatus. This test apparatus was designed such that it is able to provide repeatable movements during operational system testing and allow the prototype VBEGT system to be evaluated in terms of Hallet's (1986) critical list and the research questions. This apparatus is composed of a replica eye, which is covered by a mask so that it has a more human appearance.
FINDINGS

As described in Section 3.5, VBEGT system testing occurs in two phases. The first phase, algorithm testing, required a series of static test images generated by SCIS Research Support. These images provided input to the VBEGT system via a modified interface that bypasses camera input to allow the use of bitmap images.

The second phase of testing, operational testing required the imaging device attached to the VBEGT to capture images of the MUM as it gazes upon the Test Grid. Findings from both phases of VBEGT system testing are located in the following sections. The data collected will be used to answer the research questions and critique outlined by Hallet (1986), using the date analysis methods outlined in Section 3.6.

5.1 Algorithm Testing Using Static Images

Testing of the VBEGT system developed for this study commenced with testing the IHM. The SCIS Research Support Department of Edith Cowan University supplied a series of test images for this purpose. These images and details of their generation are located in Appendix H. As per the method outlined in Section 3.5.1, these test images provided input to the VBEGT system as a replacement for images of the MUM. By using static images, we are able to determine both the accuracy of the IHM and the time taken to determine the position of a user's iris under controlled conditions. Results for these tests appear in the following subsections.

Movement of the test subject's head influenced the pupil positions depicted by the images supplied. Therefore, this study is unable to use these images as a control when determining the accuracy of the method implemented to calculate the user's visual line of gaze. Testing the accuracy of the visual line of gaze is deferred until operational system testing outlined in Section 5.2.
5.1.1 Determining the Accuracy and Robustness of the Iris Hypothesis Method

Determining the accuracy and robustness of the VBEGT system implemented for this study begins by determining whether the IHM is able to produce identical output upon representation of a given test image to the VBEGT system. After establishing the reliability of the IHM, its accuracy may be determined by comparing the xy position of the iris recorded by SCIS Research Support (see Table H. 3 in Appendix H) to the position determined by the IHM. IHM testing followed the procedures outlined in Section 3.5.1.1.

Repeated testing of the IHM demonstrated that it was able to determine the location of the iris at the pixel positions shown in Figure 5.1 for all tests performed. From this information, one may conclude that the IHM performs reliable feature extraction - the output produced for each test was identical given the corresponding image as input.

<table>
<thead>
<tr>
<th>Image #</th>
<th>Iris Co-ordinates (x1,y1) (x2,y2)</th>
<th>Iris (w x h)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>(225,276) (344,395)</td>
<td>120 x 118</td>
</tr>
<tr>
<td>1</td>
<td>(162,254) (281,373)</td>
<td>120 x 118</td>
</tr>
<tr>
<td>2</td>
<td>(170,284) (289,403)</td>
<td>120 x 118</td>
</tr>
<tr>
<td>3</td>
<td>(255,268) (374,386)</td>
<td>120 x 118</td>
</tr>
<tr>
<td>4</td>
<td>(250,208) (369,326)</td>
<td>120 x 118</td>
</tr>
<tr>
<td>5</td>
<td>(225,134) (345,253)</td>
<td>120 x 118</td>
</tr>
<tr>
<td>6</td>
<td>(235,128) (354,246)</td>
<td>120 x 118</td>
</tr>
<tr>
<td>7</td>
<td>(312,122) (431,241)</td>
<td>120 x 118</td>
</tr>
<tr>
<td>8</td>
<td>(310,86) (429,204)</td>
<td>120 x 118</td>
</tr>
<tr>
<td>9</td>
<td>(270,68) (389,187)</td>
<td>120 x 118</td>
</tr>
<tr>
<td>10</td>
<td>(275,80) (394,199)</td>
<td>120 x 118</td>
</tr>
<tr>
<td>11</td>
<td>(319,70) (438,189)</td>
<td>120 x 118</td>
</tr>
<tr>
<td>12</td>
<td>(314,50) (433,168)</td>
<td>120 x 118</td>
</tr>
<tr>
<td>13</td>
<td>(268,66) (387,184)</td>
<td>120 x 118</td>
</tr>
<tr>
<td>14</td>
<td>(287,56) (407,174)</td>
<td>120 x 118</td>
</tr>
</tbody>
</table>

Figure 5.1: Results from the IHM showing the xy pixel positions for the upper left and lower right corners of the user's iris for each test image supplied by SCIS Research Support.

---

40 As per the method outlined in Section 3.5.1.1, the IHM algorithm was tested on no less than fifty occasions to determine the reliability of the output using the series of fifteen test images provided by SCIS Research Support.
Once the reliability of the IHM was established, testing focused upon determining the accuracy with which the IHM was able to discern the location of a user's iris within a given test image. As discussed in Section 4.3.1, the VBEGT system is highly configurable. The IHM may be configured to skip rows of the Eye Image during processing in order to improve execution time (see Section 5.1.2). Such configuration, however, may reduce the vertical accuracy of the calculated iris locations (see Section 5.1.2.1). Subsequently, algorithm testing proceeded with row skipping disabled.

The xy pixel positions for the lower left hand and upper right hand corners of the user's iris calculated by the IHM were compared to the positions recorded by SCIS Research Support. In total, fifteen comparisons were made based upon the assumption that the user's gaze was focused upon one of the fifteen calibration points used during the generation of each test image (see Appendix H).

The results of algorithm testing showed that the IHM was able to determine the exact position of the test subject's iris relative to the locations supplied by SCIS Research Support as shown Table 5.1. The IHM achieved an error rate equal to zero percent. Such a low level of error is not unexpected given the nature of the test images supplied - "each image comprises a black circle equal in diameter to the subject's iris superimposed over a larger white disc (representing the white of the eye)" (Wild, 2004, p. 3) (see Figure 5.2).
Comparison between the iris locations recorded by SCIS Research Support and those determined using the IHM.

<table>
<thead>
<tr>
<th>Calibration Point</th>
<th>Iris Location Recorded By SCIS Research Support</th>
<th>Iris Location Determined By The IHM Algorithm</th>
<th>Percentage Error</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Left (x-Axis)</td>
<td>Bottom (y-Axis)</td>
<td>Left (x-Axis)</td>
</tr>
<tr>
<td>A</td>
<td>225</td>
<td>395</td>
<td>225</td>
</tr>
<tr>
<td>B</td>
<td>162</td>
<td>373</td>
<td>162</td>
</tr>
<tr>
<td>C</td>
<td>170</td>
<td>403</td>
<td>170</td>
</tr>
<tr>
<td>D</td>
<td>255</td>
<td>386</td>
<td>255</td>
</tr>
<tr>
<td>E</td>
<td>250</td>
<td>326</td>
<td>250</td>
</tr>
<tr>
<td>F</td>
<td>226</td>
<td>253</td>
<td>226</td>
</tr>
<tr>
<td>G</td>
<td>235</td>
<td>246</td>
<td>235</td>
</tr>
<tr>
<td>H</td>
<td>312</td>
<td>241</td>
<td>312</td>
</tr>
<tr>
<td>I</td>
<td>310</td>
<td>204</td>
<td>310</td>
</tr>
<tr>
<td>J</td>
<td>270</td>
<td>187</td>
<td>270</td>
</tr>
<tr>
<td>K</td>
<td>275</td>
<td>199</td>
<td>275</td>
</tr>
<tr>
<td>L</td>
<td>319</td>
<td>189</td>
<td>319</td>
</tr>
<tr>
<td>M</td>
<td>314</td>
<td>168</td>
<td>314</td>
</tr>
<tr>
<td>N</td>
<td>268</td>
<td>184</td>
<td>268</td>
</tr>
<tr>
<td>O</td>
<td>287</td>
<td>174</td>
<td>287</td>
</tr>
</tbody>
</table>
Accuracy testing requires, “an obvious distinction between the white of the eye and the iris to clearly define iris location” (Wild, 2004, p. 3). Without a clear location, testing would not be possible as the location of the iris boundary is highly subjective in low contrast images and it is not suitable to assume that two people would position the iris boundary at the same location (Wild, 2004).

The high contrast test images allowed the IHM to calculate iris position by scanning for the drop in pixel intensity from 255 to 0 present along the sclera/iris boundary (Wild, 2004). Such drops in intensity are not present elsewhere in the image. Subsequently, the IHM was configured to suit this observation. Such dramatic changes in pixel intensity are unlikely to occur during operational testing as the sclera/iris boundary appears to change from dark grey to light grey rather than from black to white (see Figure 5.3). As discussed in Section 3.2.1.4, the magnitude of the drop in pixel intensity along the sclera/iris boundary directly affects the accuracy of iris detection.
Figure 5.3: Test images supplied by SCIS Research Support experience a greater drop in pixel intensity along the sclera/iris boundary than images captured by the VBEGT system camera during operational system testing.
5.1.2 Determining the Speed of the Iris Hypothesis Method

Once the maximum accuracy of the IHM was determined, testing proceeded to determine the average execution time taken to produce these results. As discussed in Section 3.5.1.2.1, a customised profiling tool was developed to determine where the VBEGT application spends most of its time. Example output from this profiler appears in Figure 5.4.

<table>
<thead>
<tr>
<th>Function Name</th>
<th>Time Per Function Call</th>
<th>Execution Calls</th>
<th>Average Function Execution Time</th>
<th>Percentage Of Total Function Execution</th>
</tr>
</thead>
<tbody>
<tr>
<td>'Detect Fixation'</td>
<td>0.0001</td>
<td>15</td>
<td>0.0000</td>
<td>0.00%</td>
</tr>
<tr>
<td>'Process Iris Hypotheses'</td>
<td>0.0000</td>
<td>15</td>
<td>0.0000</td>
<td>0.00%</td>
</tr>
<tr>
<td>'Find Iris Hypothesis'</td>
<td>0.0011</td>
<td>29012</td>
<td>0.0000</td>
<td>1.18%</td>
</tr>
<tr>
<td>'Create Iris Hypotheses'</td>
<td>0.0000</td>
<td>5</td>
<td>0.0001</td>
<td>0.59%</td>
</tr>
<tr>
<td>'Process Line'</td>
<td>0.0000</td>
<td>6750</td>
<td>0.0000</td>
<td>2.36%</td>
</tr>
<tr>
<td>'Process Image'</td>
<td>0.0455</td>
<td>15</td>
<td>1.4180</td>
<td>73.47%</td>
</tr>
<tr>
<td>'Pre-Process Image'</td>
<td>0.4266</td>
<td>15</td>
<td>1.8901</td>
<td>97.93%</td>
</tr>
<tr>
<td>'Capture File Image'</td>
<td>0.0000</td>
<td>15</td>
<td>0.0000</td>
<td>0.00%</td>
</tr>
<tr>
<td>'Determine User Gaze'</td>
<td>1.8901</td>
<td>15</td>
<td>1.8901</td>
<td>97.93%</td>
</tr>
<tr>
<td>'Pre-Process Iris Hypothesis'</td>
<td>0.0000</td>
<td>15</td>
<td>0.0000</td>
<td>0.00%</td>
</tr>
</tbody>
</table>

Figure 5.4: Example output from the VBEGT showing the total number of calls and execution times for each step of the IHM.

From this output, it shows that the VBEGT system spends most of its time during its image pre-processing stage. This stage consumes over of 73% of total execution time. In comparison, accessing the test image stored on the test machine's hard drive consumes 22%, whilst the IHM consumed approximately 3% of total execution time. By implementing a more efficient pre-processing, this study would have been able to reduce the amount of time spent during this stage. However, due to given time constraints, such implementation was not possible and may be considered a direction for further study.
In an attempt to improve upon the execution times recorded, the IHM implements a method whereby a specified number of rows in the Eye Image are skipped during processing. For example, the IHM may process every other image row, or every third. To determine the effect of row skipping, the average execution speed of the IHM was evaluated when it was configured to skip 0, 1, 2, 3 or 4 rows of the Eye Image. Results from this evaluation are recorded in Table 5.2. As shown by the graph located in Figure 5.5, execution time reduces significantly as the number of rows skipped increased. This reduction follows the quartic regression function shown in Figure 5.6.

Table 5.2: Average execution times for the IHM based upon the number of rows skipped during image analysis.

<table>
<thead>
<tr>
<th>Number of Rows Skipped</th>
<th>Execution Time (sec)</th>
<th>Average Execution Time (sec)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Trial 1</td>
<td>Trial 2</td>
</tr>
<tr>
<td>0</td>
<td>0.68</td>
<td>0.66</td>
</tr>
<tr>
<td>1</td>
<td>0.31</td>
<td>0.30</td>
</tr>
<tr>
<td>2</td>
<td>0.19</td>
<td>0.19</td>
</tr>
<tr>
<td>3</td>
<td>0.14</td>
<td>0.13</td>
</tr>
<tr>
<td>4</td>
<td>0.10</td>
<td>0.08</td>
</tr>
</tbody>
</table>

Figure 5.5: As the number of rows skipped when processing a given Eye Image increases, the average execution time of the IHM decreases.
\[ y = ax^4 + bx^3 + cx^2 + dx + e \]

a = quartic regression coefficient
b = cubic regression coefficient
c = quadratic regression coefficient
d = linear regression coefficient
e = regression constant term (intercept)

<table>
<thead>
<tr>
<th>Regression coefficient</th>
<th>Intercept</th>
</tr>
</thead>
<tbody>
<tr>
<td>a = 0.541660000E-03</td>
<td></td>
</tr>
<tr>
<td>b = -0.641666000E-01</td>
<td></td>
</tr>
<tr>
<td>c = 0.279583330E+00</td>
<td></td>
</tr>
<tr>
<td>d = -0.580833300E+00</td>
<td></td>
</tr>
<tr>
<td>e = 0.660000000E+00</td>
<td></td>
</tr>
</tbody>
</table>

Figure 5.6: Quartic regression formula representing the best fit for the data shown in Table 5.2.

5.1.2.1 Trading Accuracy for Speed

By skipping rows, it is possible to decrease the average time taken to process a given Eye Image (see Table 5.2 and Figure 5.5). This decrease is accompanied by an increase in the average vertical error as shown in Table 5.3 and Figure 5.7. This error follows the regression formula shown in Figure 5.8 and is sourced from the deviation between the iris position recorded by SCIS Research Support and that calculated by the VBEGT system. The iris positions calculated by the IHM for each calibration point based upon the number of rows skipped during image analysis are located in Appendix J.

Table 5.3: The average error in pixels reported for the IHM based upon the number of rows skipped during image processing.

<table>
<thead>
<tr>
<th>Number of Rows Skipped</th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
</tr>
</thead>
<tbody>
<tr>
<td>Left (x-Axis)</td>
<td></td>
<td></td>
<td>-1.73</td>
<td>0</td>
<td>-8.53</td>
</tr>
<tr>
<td>Bottom (y-Axis)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Left (x-Axis)</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Bottom (y-Axis)</td>
<td>0</td>
<td>0</td>
<td>-4.66</td>
<td>0</td>
<td>-11.86</td>
</tr>
</tbody>
</table>
Figure 5.7: The average horizontal and vertical error reported for the IHM based upon the number of lines skipped during image processing.

\[ y = ax^4 + bx^3 + cx^2 + dx + e \]

- \( a \): quartic regression coefficient
- \( b \): cubic regression coefficient
- \( c \): quadratic regression coefficient
- \( d \): linear regression coefficient
- \( e \): regression constant term (intercept)

<table>
<thead>
<tr>
<th>Regression coefficient</th>
<th>Intercept</th>
</tr>
</thead>
<tbody>
<tr>
<td>( a )</td>
<td>( b )</td>
</tr>
<tr>
<td>-0.758333000E-01</td>
<td>0.461666660E+00</td>
</tr>
</tbody>
</table>

Figure 5.8: Quartic regression formula representing the best fit for the data shown in Table 5.3.
The discrepancy between the iris positions derived by the IHM and those recorded by SCIS Research Support is due in part to the implementation of the IHM. The number of rows skipped is configurable via the VBEGT configuration file discussed in Section 4.3.1. As shown by the results listed in Table 5.3 and Figure 5.7, row skipping does not affect the horizontal accuracy of the IHM. The average vertical error increases as the number of rows skipped increases. This error occurs due to the start and end rows of an iris being missed if they start on a row that is skipped (see Table 5.4 and Table 5.5).

Table 5.4: The effects of row skipping on sample iris images showing the Line Segments used to calculate the height of the iris for those images of even height.
By reducing the execution time required to process a given Eye Image, the VBEGT system is able to provide a real-time response. As discussed in Section 2.3.5, the success or failure of a given eye-gaze tracking system may depend upon the time taken to report the location of a user’s iris (Mulligan, 1997a; Quek, 1995). Despite the need for real-time response times, a VBEGT system must also be considered accurate. Therefore, a balance between accuracy and speed of response must be reached.

The relationship between the speed of execution and accuracy of the VBEGT system developed appears in Figure 5.9. As shown by this figure and Table 5.2, the rate of increase is most significant between zero and one row being skipped during image processing. The average vertical error appears to increase from 0 to \(-1.73\) pixels. At this point, the IHM experiences the lowest increase in vertical error and maintains an average horizontal error of zero pixels (see Table 5.3 and Figure 5.7).

<table>
<thead>
<tr>
<th>Number Of Rows Skipped</th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
</tr>
</thead>
</table>
| 0                      | ![Table 5.5](image)

The effects of row skipping on sample iris images showing the Line Segments used to calculate the height of the iris for those images of odd height.
Figure 5.9: Relationship between average vertical error and execution times for the IHM based upon the number of rows skipped per image.
As demonstrated by Table 5.4 and Table 5.5, the increase in vertical error is due to the start and end rows of an iris being missed during image processing, due to the row skipping algorithm implemented by the IHM. Shortening the vertical aspect of a user’s iris does not necessarily result in an erroneous estimation of the user’s visual line of gaze. Using Table 5.4 and Table 5.5 as examplea, the centre point for each sample iris depends upon whether the Line Segments comprising the iris starts on an odd or even row of the Eye Image as shown by Table 5.6. See Appendix K for centre point calculations.

Table 5.6: The effects of row skipping upon the calculation of the centre point for a given iris image, based upon the positions shown in Table 5.4 and Table 5.5.

<table>
<thead>
<tr>
<th>Starting Row</th>
<th>True Centre Point</th>
<th>Number Of Rows Skipped</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>0</td>
</tr>
<tr>
<td>Even</td>
<td>3.5</td>
<td>3.5</td>
</tr>
<tr>
<td>Odd</td>
<td>4.5</td>
<td>4.5</td>
</tr>
<tr>
<td>Even</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>Odd</td>
<td>4</td>
<td>4</td>
</tr>
</tbody>
</table>

As computers work with pixels, the IHM rounds the centre point values calculated up to the next whole number. This practice results in the values listed in Table 5.6 appearing as shown in Table 5.7. By doing so, the average probability of introducing an error due to skipping rows in the Eye Image is significantly reduced when one row of the Eye Image is skipped (see Table 5.8). If more than one row is skipped, the probability of introducing error ranges from 50% to 75% (see Table 5.8). Subsequently, it is believed that configuring the VBEGT system developed for this study to skip every other row of a given Eye Image, at the cost of introducing a 25% rate of error, is worth the reduction of execution time from 0.66 seconds to 0.30 seconds per Eye Image.
Table 5.7: The effects of row skipping upon the calculation of the centre point for a given iris image, based upon those shown in Table 5.4 and Table 5.5.

<table>
<thead>
<tr>
<th>Starting Row</th>
<th>True Centre Point</th>
<th>Number Of Rows Skipped</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>0</td>
</tr>
<tr>
<td>Even Image Height</td>
<td></td>
<td>4</td>
</tr>
<tr>
<td>Odd Image Height</td>
<td></td>
<td>5</td>
</tr>
</tbody>
</table>

| Even Image Height |       | 3 | 3 | 3 | 3 | 2 | 3     |
| Odd Image Height  |       | 4 | 4 | 4 | 5 | 4 | 5     |

Table 5.8: Average percentage probability of error caused by skipping rows of the Eye Image during processing.

<table>
<thead>
<tr>
<th>Percentage Probability Of Error</th>
<th>Number Of Rows Skipped</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>0</td>
</tr>
<tr>
<td>Rounded</td>
<td>0</td>
</tr>
<tr>
<td>Un Rounded</td>
<td>0</td>
</tr>
</tbody>
</table>
5.2 **Operational System Testing**

Operational system testing is concerned with determining the accuracy of the VBEGT System as it gazes upon a set of predefined points on the Test Grid. As discussed in the following section the test environment was configured such that it mimics a typical computer user’s environment. In order to determine what a typical user environment is, this study has adhered to the Australian Standards\(^{41}\) for safe computer operation.

Once the test environment is configured, the resolution of the VBEGT system may be determined. The maximum resolution of the system was evaluated in terms of defining the range of horizontal and vertical movement required by the MUM to move from the top left of the user’s screen to the bottom right.

5.2.1 **Configuration of the Experiment**

Operational system testing involved capturing the movements of the MUM’s replica eye as it gazed upon various points of the Test Grid. As shown in Figure 5.10, the MUM was oriented at a distance of 550mm from the computer monitor, which is the median distance between the maximum (750mm) and minimum (350mm) distances outlined by A.S. 3590 – 1990. This distance is within the range noted by Charles Darwin University (2004) to be “common amongst computer users” (Charles Darwin University, 2004, p. 1). The A.S. 3590 – 1990 standard also recommends that the screen should be below the user’s eye level as shown in Figure 5.11. The computer monitor chosen for this study was an ADI Micro Scan 5P+ 17” monitor (16” diagonal viewable image).

---

\(^{41}\) The Australian Standards used when configuring this experiment were A.S. 3590 – 1990: Screen-Based Workstations, Part Two: Workstation Furniture, AS 3590.3-1990 Screen based workstations: Part 3 Input Devices, and A.S. 1680-1990 Interior Lighting.
Figure 5.10: The MUM was positioned 550mm away from the monitor in accordance with Australian Standards.

Figure 5.11: The MUM was positioned such that the computer monitor is below EMM “eye-level” to conform with Australian Standards.
VBEGT Systems are sensitive to the lighting conditions of the surrounding environment in which they are operating. As per the recommendation for the Australian Council of Trade Unions (2002) the lighting chosen for this experiment mimicked that found in a clerical office environment. By using a white light, the VBEGT system is able to set a more accurate value for the white balance of a given image compared to those operating in environment that uses a yellow light source (Davis, 1997). In doing so, this aids feature extraction by making the boundaries between the iris and the sclera easier to distinguish than those captured using an incorrect white balance (Davis, 1997).

In order to obtain a clear image of the user's iris, this study places the camera at the bottom of the user's monitor facing upwards. This orientation differs from that used by standard video-conferencing, which positions the camera at the top of the user's monitor facing downwards. As shown by Figure 5.12, the image captured by the camera facing upwards shows more of the user's iris in comparison to the image captured by the camera facing downwards which shows the iris being obscured by the user's eyelids.

5.2.2 Determining the Maximum Resolution of the Video-Based Eye-Gaze Tracking System

Given the configuration of the experiment shown from Figure 5.10 through Figure 5.12, the MUM was able to display EMM movement in the range of 37 horizontal and 30 vertical steps during operational system testing. From this range, the maximum screen positional accuracy of the VBEGT System that may be identified using the EMM as a test subject would be 37 x 30 pixel regions.

42 "Generally, in a clerical office environment, a 36 watt white fluorescent tube is recommended, as it provides reasonable colour rendering properties with a very good efficiency." (Australian Council of Trade Unions, 2002, p. 1)
A) Eye Image captured by a camera located at the top of the user’s monitor facing downwards.

B) Eye Image captured by a camera located at the top of the user’s monitor facing upwards.

Figure 5.12: The camera was positioned at the bottom of the monitor to maximise the range of user eye movement that could be captured without obstruction by the user’s eyelids.

From a distance of 550mm the range of eye movement detected by the IHM was in the order of 42 horizontal pixels by 23 vertical pixels. Assuming no error, these values indicate the maximum resolution that may be supported by the VBEGT system at this distance. To increase this resolution the VBEGT system, the distance from the camera to the user would need to be decreased or a zoom lens attached to the camera.

During operation, the VBEGT system was able to record 94.6% of horizontal and 80% vertical EMM movement accurately based upon the average horizontal and vertical locations for the user’s iris recorded during operational system testing. The error rates observed, 5.4% and 20% respectively, are due in part to the range of movement detected in the Eye Image (see Table 5.9).
Table 5.9: Percentage error of the IHM when detecting iris location based upon the average locations for the iris recorded during operational system testing.

<table>
<thead>
<tr>
<th>Axis of Movement</th>
<th>Number of EMM Positions</th>
<th>Number of Duplicate Positions</th>
<th>Percentage Error</th>
</tr>
</thead>
<tbody>
<tr>
<td>Horizontal</td>
<td>37</td>
<td>2</td>
<td>5.4%</td>
</tr>
<tr>
<td>Vertical</td>
<td>30</td>
<td>6</td>
<td>20%</td>
</tr>
</tbody>
</table>

Each horizontal step of the EMM covered approximately one pixel of movement in the captured Eye Image\(^4^3\). Vertical movement of the EMM, however, is less than one pixel per step, i.e., 0.76 pixels per step\(^4^3\). Due to the number of pixels per step being less than one, vertical screen positional is reduced to 23 regions as displayed by the Eye Image (see Figure 5.13) instead of the 30 regions moved by the EMM.

---

\(^4^3\) The number of pixels of movement is the product of the number of steps made from one side of the screen to other divided by the total number of pixels shown to detect this movement in the eye image. Therefore, if the horizontal steps from one side of the screen to the other equal 37 and the number of pixels showing this movement is 42 (see Figure 5.13), then the number of pixels per step of the EMM is equal to 42 pixels divided by 37 steps, i.e., 1.135 pixels per step. Similarly, moving from top to bottom would equal 23 pixels of movement divided by 30 vertical steps, i.e., 0.76 pixels per step.
Figure 5.13: Range of iris movement detected by the VBEGT system during operational system testing.
Summary

Testing the VBEGT system developed for this study occurred in two phases. The first phase revolved around determining the speed an accuracy of the IHM. Testing occurred within a controlled environment, using test images provided by SCIS Research Support. Based upon these images the IHM was found to operate robustly and provide an accurate and repeatable representation of the user’s iris within a given Eye Image.

In addition to determine the accuracy and repeatability of the IHM, measurements were taken to determine whether a significant increase in the speed of response of the IHM may be afforded by skipping rows of the Eye Image during processing. It was found that the VBEGT system response speeds decreased from 0.66 seconds to 0.30 seconds per Eye Image.

The row skipping algorithm used by this study, was found to introduce an average error of -1.73 pixels for 20% of images to the vertical component of the calculated visual line of gaze. The horizontal component remains unchanged. The author believes that the significant increase in the system’s processed frame rate justifies the possible introduction of error.

The second phase of testing, involved using the VBEGT system within a controlled environment similar to a clerical office environment typical to mainstream computer users. Where possible the controlled environment adhered to A.S. 3590 – 1990: Screen-Based Workstations, Part Two: Workstation Furniture, AS 3590.3-1990 Screen based workstations: Part 3 Input Devices, and A.S. 1680-1990 Interior Lighting. Results from operational system testing indicated that the maximum resolution of the VBEGT system at a distance of 550mm was able to distinguish between a maximum of 42 horizontal regions and 23 vertical regions.
DISCUSSION

Obtaining an accurate, objective evaluation of whether a given VBEGT system is suitable for mainstream use is problematic. The main difficulty arises from determining the requirements of mainstream users. This study addresses the use of VBEGT as an alternative to low bandwidth input devices (the keyboard and mouse) used to interact with non-specialised computing applications.

Determining mainstream suitability for VBEGT requires the assumption that "mainstream", relates to those computer users who use a keyboard or mouse to action movements within non-specialised computing applications. Such movements include positioning a mouse cursor for object selection. Common examples of non-specialised computing applications include navigating the Internet via a web browser and writing a text document using a word processor. The development of eye-gaze enabled input devices assumes that these applications do not require fine cursor movement. Given this definition, the suitability of a VBEGT system for mainstream use is dependent upon being able to provide a means of input comparable to a mouse, in terms of performance and ease of use.

After defining the target population, VBEGT system evaluation is able to take a more systematic approach. This approach focuses upon a set of recognised critiques that define an "ideal" eye-gaze tracking system. Following this evaluation, the system assessment refers back to the target population to answer the research question. This section follows this approach to determining the mainstream suitability of the VBEGT system developed and technologies implemented.
6.1 Critique of Eye-Gaze Tracking Systems

As discussed in Section 2.2.3 and Appendix C, Hallet (1986) lists twelve criteria which are desirable in mainstream eye gaze tracking systems. Despite these requirements being desirable, they are not all prerequisites for an acceptable eye-gaze tracking interface (Glenstrup & Engell-Nielsen, 1995). What may be considered acceptable depends largely upon the desired application of the system (Glenstrup & Engell-Nielsen, 1995). This section discusses the performance of the VBEGT system developed for this study in relation to Hallet’s critiques. Where necessary the relevance of each critique is addressed in relation to their suitability for selective eye-gaze tracking as part of a system developed for mainstream computing use.

6.1.1 Unobstructed Field of View

In order to track user eye movements successfully, the camera attached to the VBEGT system needs to maintain an unobstructed view of the user. To obtain such a view, this study mounted its imaging device to the bottom of the computer monitor pointing up towards the user. This configuration differs from the typical video-conferencing position to reduce the area of the user’s iris obscured by the upper eyelid as discussed in Section 5.2.1.

By mounting the camera to the bottom of the user’s monitor, the VBEGT system does not interfere with the user’s field of view as compared to other systems with mount cameras in front of the user. For example Kim, Kim, Chung (2001), use a head mounted tracking system which requires a camera to be mounted onto a pair of spectacles worn by the user during system operation. Extended use of an intrusive method such as the system developed by Kim, Kim, Chung (2001) is likely to cause user discomfort to the level where it would render the system unusable by mainstream users due to a user’s vision.

---

44 A typical video-conference positions the camera at the top of the user’s monitor facing down towards the user (Capron, 1998).
being obscured by the camera (Como et al., 2002). Obscuring a user’s vision in such a
way may also cause visual fatigue, as the user has to concentrate to look past the camera
in order to concentrate upon the activity at hand. According to the National Occupational
Health And Safety Commission, visual fatigue should be reduced where possible to
maintain a healthy work environment (National Occupational Health And Safety

Head mounted equipment is commonly used to overcome the effect of head
movements made by the user whilst using the VBEGT system. This study assumes that
the user’s head remains in a constant position. As discussed in Section 6.1.11, VBEGT
systems such as the one developed for this study may be extended to use head and body
recordings that may be used to provide a more accurate estimation of user gaze than using
user eye movements alone. Such extension would allow the VBEGT system to overcome
the effect of user head movements and allow the user to move freely without restriction
during system use (Shih & Liu, 2002).

6.1.2 Makes No Contact with the Subject

As discussed in Section 1.1, a major factor in determining whether a given eye­
gaze tracking system is suitable for mainstream use depends upon the level of restriction
imposed upon the user. Previously, chin rests were required to restrict user head movent,
or required them to wear cumbersome equipment (Baluja & Pomerleau, 1994). Even if
made commercially available, it is unlikely that computer users would use these systems
for mainstream use (i.e., outside laboratory situations) due to their intrusive nature
(Glenstrup & Engell-Nielsen, 1995; Software Engineering Australia, 2003b).

By design, VBEGT systems utilise a camera to capture images of the user’s eyes.
As discussed in Section 2.2.1, eye-gaze tracking has evolved to use video cameras to
track user’s eye movements for the very fact that it does not require contact with the user
in order to track their eye movements. In order to create an eye-gaze tracking system that would be suitable for mainstream computer use, a non-contact monitoring method was deemed essential for use in this study.

The VBEGT system developed for this study positioned the camera at the bottom of the user’s monitor. By doing so, the system was able to maintain a comfortable distance from the user and still capture an unobstructed view of the user’s eye. By mounting the camera to the monitor, the user is still able to move freely without restriction in a manner common to mainstream computer use and reduce the risk of Occupational Overuse Syndrome (Worksafe Western Australia, 1996).

6.1.3 Artificial Image Stabilising

During operation, it is possible that an eye-gaze tracking system of sufficient resolution would be able to pick up the nystagmic and slow drift movements of a user’s eye as he/she views their computer monitor. In doing so, the practical accuracy of eye-gaze tracking system may be limited, as these movements may be misinterpreted and reported as intentional eye movement to the gaze-enabled application (Jacob & Karn, 2003). It is possible to improve the accuracy of the tracking system by averaging over a fixation (Hallet, 1986; Jacob & Karn, 2003). Such averaging, however, does limit the responsiveness of the eye-gaze tracking system, as it takes \( n \) frames before the system reports the start of fixation (Andiel et al., 2002).

This study used the public domain algorithm supplied by LC Technologies Inc. (2002) as listed in Figure F. 1 (see also Figure 4.12) to determine the current state of user eye movement. This algorithm utilises a moving average to determine whether the current gaze point is part of the current fixation. This algorithm has a pre-configured gaze deviation threshold which defines the distance from which a given gaze point may deviate from the previous gaze point and still be included in the current fixation. To
determine the start of fixation, the algorithm supplied by LC Technologies Inc. (2002) requires a minimum of three gaze points upon which to calculate its average (a form of moving average). Subsequently, fixation is determined 0.90 seconds after the start of fixation.45

If the gaze point calculated by the IHM is considered to be part of the current fixation, the algorithm provided by LC Technologies Inc. (2002) returns the average of the last three gaze points comprising the current fixation as the location of user gaze. If the calculated gaze point is not part of the current fixation, its value is returned unaltered.

6.1.4 Accuracy of At Least 1% or A Few Minutes Of Arc

Determining the accuracy of a given VBEGT system requires addressing all sources of error that may affect the calculated gaze point. As discussed in Section 2.2.2 a number of factors affect the accuracy of the gaze point determined. These errors may be divided into those originating from the VBEGT system and those from the user. Common sources of error originating from the VBEGT system are due to the resolution of the Eye Image as discussed in Section 6.1.5.

The VBEGT system implemented as part of this study utilises a row skipping algorithm to reduce the number of CPU cycles consumed by the IHM per processed image (see in Section 5.1.2.1). In doing so, it is possible that a -1.73 pixel error will be introduced for approximately 25% of images used to determine the vertical component of the user’s visual line of gaze. The row-skipping algorithm does not introduce any error for the horizontal component.

45 It takes an average of 0.30 seconds to process a given Eye Image when the VBEGT system is configured to process every other row. Given that LC Technologies Inc. (2002) requires a minimum of three gaze points upon which to calculate its average, fixation is determined after 3 * 0.30 seconds.
The most prevalent sources of error common to eye-gaze tracking systems is introduced by the user through head movement. By moving their head a user is able to redirect their gaze without displacing their iris as far as would otherwise be necessary to view a given object should their head remain in a fixed position. Therefore if a given VBE GT system is calibrated such that 10 pixels of iris displacement corresponds to calibration point A, if the user moves their head to view calibration point A, then the iris displacement may be reduced to 3 pixels and the eye-gaze tracking system may report an incorrect calculation of user gaze, based upon this reduced displacement.

The effects of user head movement have been negated by this study, through the use of a mechanical user model. This model is able to move its replica eye independently of the rest of the model as discussed in Section 4.2. Use of this mechanical model also benefits the calibration process as it offers the operator visual confirmation of the point at which the mechanical model gazes.

From a distance of 550mm, operational system testing found the VBE GT system developed for this study was able to determine 34 discrete horizontal locations for the mechanical model of the 37 locations possible. Vertically, 24 positions were possible given an error rate of 20% due in part to the row skipping algorithm used to improve the response of the VBE GT system.

6.1.5 Resolution of 1 Min or 1 Min Arc·sec-1

The resolution of a given eye-gaze tracking system differs from its accuracy as it does not include all sources of error, for example error introduced by user head movements. The resolution of an eye-gaze tracking system may be considered from two points of view: static resolution, determined by the resolution of the Eye Image captured (Mulligan, 1997b); and dynamic resolution, determined by the eye-gaze tracking system's processed frame rate (Mulligan, 1997b).
As discussed in Section 5.2.2, the static resolution of the VBEGT system developed for this study is dependent upon the range of displacement of the user’s iris detected in a given sequence of Eye Images. As shown in Figure 5.13, the maximum static resolution of the VBEGT system is 42 horizontal regions by 23 vertical regions for a user sitting 550mm from the VBEGT system camera. These measurements equate to regions approximately 24 pixels by 33 pixels in size based upon a screen resolution of 1024 x 768.

Hallet (1986) suggests that an ideal tracking device should be able to provide a resolution of at least 1 min or 1 min arc-sec. However, as noted by Jacob and Karn (2003) such a high level of resolution is not required for selective eye-gaze tracking systems:

“The necessary resolution of an eye tracker that is useful in a real-time interface (as opposed to the more stringent requirements for basic eye movement research) is limited, since a user generally need not position his or her eye more accurately than about one degree to see an object sharply” (Jacob & Karn, 2003, p. 591).

The dynamic resolution of the VBEGT system is dependent upon the processed frame rate. The VBEGT system implemented during this study is able to determine the user’s visual line of gaze at a rate of 0.30 seconds per frame.

---

46 The number of screen regions is equal to the screen resolution divided by the ratio of screen resolution to Eye Image pixels displaced during iris movement.
47 One minute, equates to 1/60th of a degree, such movement as discussed in Section 6.2.4 is beyond the capabilities of the mechanical user model used to test the VBEGT system.
48 As discussed in Section 5.1.2.1, skipping every other row of an Eye Image in order to reduce the processed frame rate to 0.30 seconds from 0.66 seconds is viable, as it does not introduce an inordinate percentage error to the visual line of gaze determined by the IHM.
Wide Dynamic Range for Eye Position and Velocity

The dynamic range of a given eye-gaze tracking system depends upon the range through which the system is able to measure the periodic position of the user’s iris. For example, systems that provide input to computer systems require a high degree of accuracy (number of pixels) across a small range (e.g., from one side of a computer monitor to another), whilst virtual reality applications require rapid tracking of the eye across a large range (e.g., horizontally from one side of a room to the other).

The VBEGT system developed for this study is a selective system, which requires a high degree of accuracy across a small range. The resolution of the system is 42 horizontal regions by 23 vertical regions given a user sitting 550mm from the VBEGT system camera. This equates to the system being able to distinguish 7.5mm of horizontal movement and 10.4mm of vertical across the user’s computer monitor.

Dynamic range for eye velocity is restricted by the rate at which the VBEGT system is able to process Eye Images. Due to the nature of human eye movements, a selective eye-gaze tracking system is required to track the user’s eye at speeds ranging between 0.03 and 0.12 seconds for saccades and 0.06 and 0.2 seconds for fixations. The VBEGT system developed for this study is able to track the eye at a rate of 0.30 seconds. Subsequently, this system is unable to detect the fixation and saccadic movements of the average user.

---

49 Based upon a 17” monitor sporting a screen size of 315mm by 240mm.
50 Saccades generally last for 30 to 120 milliseconds, whilst fixation generally lasts between 200 and 600 milliseconds depending upon the complexity of the scene being viewed (Cowen, 2001; Hyrskykari, 1997).
Temporal Dynamics and Speed of Response

Temporal dynamics are limited by the processed frame rate of a given VBEGT system. By having a limited frame rate some of the dynamics of user eye movement may be lost (Sibert & Jacob, 2000). For example, Sibert and Jacob (2000) recognised that saccades take between 30 and 120 milliseconds. Assuming a VBEGT system captures an image and calculates a user’s visual line of gaze every 1/60 second, the system will capture between 2 (1.8) and 7 (7.5) frames of saccadic movement.

By decreasing the time taken to capture images and calculate the visual line of gaze, one may increase the number of frames of saccadic movement captured. This increase would improve the system’s temporal dynamics by increasing response times and reduce the possibilities of incorrectly responding to user eye movement or capturing blurred images (see Figure 6.1).

As discussed in Section 6.2.1, the VBEGT system developed for this study processes Eye Images at a rate of 0.30 seconds per image. Such speeds were found to be longer than the duration of a typical user’s saccade or fixation, thus reducing the suitability of the system for mainstream computing use.

---

51 Blurred images typically result in tracking errors as the shape of the iris appears as an elongated ellipse rather than appearing circular. Such tracking errors would result in an invalid gaze point being returned from the eye-gaze tracking system.
Occasionally the camera captures the Eye Model mid movement. In these situations, the iris is blurred. Such blurring often results in detection errors as the geometric properties of the iris are changed.

6.1.8 Possess a Real-Time Response

As discussed in Section 6.1.8, eye gaze tracking systems are unable to provide a true real-time response to use eye movements. This is due to the effect of artificial image stabilizing discussed in Section 6.1.3. Eye-gaze tracking systems are, therefore, referred to as responding to user eye movements in pseudo real-time.

The VBEGT system developed for this study requires three Eye Images to be processed before the state of the user’s eye may be determined. The number of samples required to determine the state of user eye movements is equal to the minimum number accepted by the algorithm provided by LC Technologies Inc (i.e., three images). By using the minimum value supported by the algorithm provided, the VBEGT system developed

---

52 The number of images required before the state of a user's eye may be determined by the VBEGT system is configurable through the VBEGT system configuration script discussed in Section 4.3.1. The minimum number of samples required to determine the state user eye movement is defined by the public domain algorithm supplied by LC Technologies Inc (see Section 4.1.2.4 and Appendix F).
for this study is able to achieve as near a real-time response as possible given the rate at which it is able to process a given Eye Image. In an attempt to improve the processing times, the IHM is configured to implement a row skipping algorithm. By skipping every other row of a given Eye Image, the VBEGT system is able to achieve response rates of 0.30 seconds, instead of 0.66 seconds.

### 6.1.9 Measure Three Degrees of Angular Rotation

As discussed in Section 2.1, the human eye requires three axes of rotation to effect the viewing of a given object. To achieve the transfer of information between human and computer for visually mediated applications only two of these axes are required, i.e., horizontal (pan) and vertical (tilt) movement as computer input occurs in two dimensions (Florin, 2004; Jacob, 1991; Polpitiya et al., 2002). Therefore, the ability to track torsional rotation is not a requirement for this study as it addresses eye-gaze tracking as a form of HCI for selective systems.

### 6.1.10 Extendable to Binocular Recording

Binocular recording requires the simultaneous tracking of a user’s left and right eyes. In order to calculate a user’s visual line of gaze accurately within three-dimensional space, two reference points are required (Minagawa, Saito & Ozawa, 1997). These points determines the position of each of the user’s eyes relative to their computer monitor (Minagawa et al., 1997). Such gaze calculation is considered to be less susceptible to user head movements (Minagawa et al., 1997; Talmi & Liu, 1999). Compensating for user head movements and determining a user’s three-dimensional visual line of gaze is considered outside the scope of this study. Subsequently, extension of the VBEGT system developed for this study to support binocular reading is also considered outside the scope of this study.
6.1.11 Compatible with Head and Body Recordings

The VBEGT system developed for this study was tailored to suit a specific set of operational requirements that focused upon determining the user's visual line of gaze based exclusively upon iris location. The user's head was required to maintain a fixed location as the calculations required to compensate for user head movements was deemed to be outside the scope of this study due to time constraints. In order to be compatible with multimodal tracking systems, a given VBEGT system would need to be able to calculate the user's visual line of gaze using both the location of the user's head as well as iris position (Duchowski, 2002; Tanriverdi & Jacob, 2000).

6.1.12 Accommodates a Variety of Subjects

Like other forms of HCI, eye-gaze tracking systems must be able to accommodate a variety of subjects (Bates, 2002). A number of physical and physiological differences exist between different users. The most prevalent physical difference existing between users is the iris colour (Bates, 2002). Human eye colour ranges from blue, grey, green, green/blue and brown to other colours (Collins, 2004). In order to compensate for the various eye colours, the VBEGT system developed for this study converts the Eye Images to grey scale to remove a possible calibration dependency that would require the user to inform the system of their eye colour.

Other common differences existing between users includes whether they require the use of corrective lenses such as glasses or contact lenses (Bates, 2002). Both glasses and contact lenses are known to cause error within eye-gaze tracking systems due to their reflective surfaces, which are known to cause glints (Bates, 2002) (see Section 3.2.2.1).

---

53 Multimodal systems generally allow the user to have free movement to move their entire body whilst interacting with a given computer system (Duchowski, 2002; Tanriverdi & Jacob, 2000)
The physiological differences between users include the duration for which a user fixates upon a given object within a scene displayed on screen (see Section 2.2.2.2). For example, an image displayed within a web browser or icon within a word processing application. Although dependent upon the complexity of the scene being perceived (Cowen, 2001; Mackworth & Morandi, 1967), the physiological difference between users has the greatest impact upon applications that initiate object selection based upon the duration for which a user fixates upon a given target (Jacob, 1991). An incorrect duration for fixation may result in the system experiencing the Midas Touch Problem (see Appendix D). The duration for which fixation occurs when initiating object selection is considered outside of the scope of this study. Such configuration is the responsibility of the eye-gaze enabled application as it differs between individual users and applications (Cowen, 2001).
6.2 Evidence Found to Support the Research Sub-Questions

In order to answer the research question a number of sub-questions were outlined to highlight the main requirements of a VBEGT system intended for mainstream computing use. These requirements, as discussed in Section 1.1, were shown to govern the success of a given the VBEGT system in the mainstream computing market. This section discusses the performance of the VBEGT system developed for this study in terms of these sub-questions.

6.2.1 Provides a Real-Time Response to User Eye Movements

As discussed in Section 5.1.2 the VBEGT system developed for this study is able to process images at an average rate of one image every 0.66 seconds. These times may be reduced to an average of 0.30 seconds per image if the system is configured to skip every other row in the image being processed. As discussed in Section 5.1.2.1, row skipping may be implemented such that the increase in system response times outweighs any loss in system accuracy.

Eye-gaze tracking systems monitor the two main forms of eye movement that enable the perception of an object, i.e., saccadic movements and fixation (Hyrskykari, 1997). Saccades generally last for 30 to 120 millisecond, i.e., 0.03 and 0.12 seconds respectively (Hyrskykari, 1997). Such movement would be undetectable using the VBEGT system developed for this study, as the processed frame rates supported are longer than the duration of a typical saccade.

Fixations are known to last between 200 and 600 milliseconds depending upon the complexity of the scene viewed (Cowen, 2001; Hyrskykari, 1997). Using the definition for the start of fixation used by LC Technologies Inc, wherein a user’s gaze must remain within a given area for more than three consecutive processed frames, the eye-gaze tracking system would be required to achieve processing speeds between 0.06
and 0.20 seconds per image. Again such movement would be undetectable using the VBEGT system developed for this study, as the processed frame rates supported are longer than the duration of a typical fixation.

6.2.2 Does Not Interfere With the User

As discussed in Section 6.1.2, the VBEGT system developed for this study does not make any form of physical contact with its subject. Interference with the user extends beyond simple physical contact, a user often requires the need to get up and move around. By design, the VBEGT system developed for this study makes no contact with the user. It involves the use of a camera mounted to the bottom of the user’s computer monitor (see Figure 5.10). Having a camera mounted to a computer monitor has become commonplace since the advent of videoconferencing, therefore such positioning is not considered psychologically intrusive (Jacob & Karn, 2003).

By mounting the camera to the computer monitor, the user is offered an unobstructed view of the monitor and its surrounds compared to other systems which mount a camera to a specialised pair of spectacles (Kim, Kim & Chung, 2001) (see Section 6.1.1). In addition, by having the camera mounted to the computer monitor the user is able to move freely without restriction\(^{54}\) in a manner common to mainstream computer use and reduce the risk of Occupational Overuse Syndrome (Worksafe Western Australia, 1996).

\(^{54}\) It is expected that the user will remain within the field of view of the camera whilst operating the computer system using the eye-tracking device (Xu et al., 1998).
6.2.3 Enables the Use of Eye Movements as Computer Input

Despite the limited dynamic resolution and speed of response (see Sections 6.1.5 and 6.2.1), the VBEGT system developed for this study exhibits the core functionality required to enable the use of eye movements as computer input. The dynamic resolution of the VBEGT system is 0.30 seconds per Eye Image. As discussed in Section 6.2.1, fixations and saccades generally last for 200 and 600 milliseconds and 30 to 120 milliseconds respectively. In order to detect these movements the VBEGT system would need to operate at a rate of 0.03 to 0.12 seconds and 0.06 to 0.20 seconds per image respectively.

As discussed in Section 6.1.5, the maximum static resolution of the VBEGT system is 42 horizontal pixels by 23 vertical pixels for a user sitting 550mm from the VBEGT system camera. This equates to approximately 42 by 23 individually distinguishable regions based upon a screen resolution of 1024 x 768. This provides an improvement over those systems discussed in Section 2.3.4 that were unable to provide a resolution exceeding 26 by 20 individually distinguishable segments on a user’s monitor. Static resolution is dependent upon the range of displacement of the user’s iris detected in a given sequence of Eye Images. To improve the static resolution the VBEGT system camera may be fitted with a zoom lens, positioned closer to the user, or fitted with a higher resolution-imaging element. The static resolutions exhibited by the VBEGT system developed for this study indicates that there is the potential for the development of a commercially viable VBEGT system that would be considered acceptable for mainstream computing use.

---

55 The number of screen regions is equal to the screen resolution divided by the ratio of screen resolution to Eye Image pixels displaced during iris movement.
56 Xu, Machin and Sheppard (1998) reported their system as being “able to operate at an average accuracy of 1.5 degrees or around 12mm apart on a computer screen” (Xu et al., 1998, p. 436). Given that the monitor used during this study measures 315 x 240mm the system developed by Xu, Machin and Sheppard (1998) would result in an average of 26 x 20 distinguishable regions.
57 Zoom lenses are known to restrict the viewing angle (field of view) of a given camera. This reduction has the potential to reduce the amount to which a user may move and still be remain within the camera’s field of view.
6.2.4 Considered Accurate within the Limitations of the Equipment Chosen

As discussed in Section 4.2.1, the MUM is able to support three degrees of head movement (not required by this study) and two degrees of eye movement in order to simulate a human user. The EMM is able to produce eye movement in the order of 0.9 degrees per step as shown in Table 6.1. From a distance of 550mm\(^{58}\), each step of the EMM would equal 9mm of movement both horizontally and vertically across a computer monitor (see Figure 6.2 and Figure 6.3).

Table 6.1: Physical characteristics of the EMM (adapted from Holme (2003)).

<table>
<thead>
<tr>
<th>Degree of Freedom</th>
<th>Range in Steps</th>
<th>Displacement per Step</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pan Stepper</td>
<td>131</td>
<td>0.9 degrees</td>
</tr>
<tr>
<td>Tilt Stepper</td>
<td>78</td>
<td>0.9 degrees</td>
</tr>
</tbody>
</table>

Figure 6.2: At a distance of 550mm each vertical step made by the EMM covers 9mm of the computer monitor.

---

\(^{58}\) 550mm is the median distance between the between the maximum (750mm) and minimum (350mm) distances outlined by A.S. 3590 – 1990 (see Section 5.2.1).
6.3 **Evidence Found to Support the Research Question**

This study was designed to address the fundamental research question: “will current technological advances in image capture technology allow the implementation of a rapid and accurate real-time non-intrusive VBEGT system that may be considered suitable for mainstream computing use.” Inherent in this main question are a number of sub-questions. These sub-questions were addressed in Section 6.2, where it was found that the VBEGT system developed for this study demonstrated the core functionality required for an eye-gaze tracking system, despite having a limited dynamic resolution and speed of response.

These limitations result from the pre-processing algorithm chosen to aid in distinguishing the iris from the rest of the Eye Image. As discussed in Section 2.3.5, it is only a matter of time before image-processing times are negligible due to the increase in the number of transistors per integrated circuit (i.e., increased processor speeds). Once processor speeds increase to the point at which they are able to process images at speeds comparable to the throughput of the image transfer mechanism (in excess of 60 frames per second\(^{59}\)) VBEGT systems will be able to respond to user eye movements in a more timely fashion.

In addition to the sub-questions originating from the main research question, the VBEGT system was evaluated in terms of the critique outlined by Hallet (1986) for an ideal system. A number of these critiques were beyond the scope of selective eye-gaze tracking systems such as the one developed for this study. For those critiques deemed applicable to this study, the VBEGT system developed was able to satisfy the majority of those critiques. Despite the limited dynamic resolution and speed of response, the prototype VBEGT system developed for this study highlights the potential for the development of a commercially viable VBEGT system that would be considered acceptable for mainstream computing use.

\(^{59}\) Based upon the PixeLINK Colour MegaPixel FireWire Camera model number PL-A642 produced by the Vitana Corporation operating at a resolution or 320 pixels by 240 pixels.
6.4 Summary

In order to determine the mainstream applicability of the VBEGT system developed during this study, it was evaluated against the critiques outlined by Hallet (1986) for an ideal tracking system, in addition to the sub-questions originating from the main research question. The VBEGT system was found to be limited in terms of dynamic resolution and speed of response. However, as the speed of computers increase and improved pre-processing algorithms developed, the speed of execution for the VBEGT system will increase and approach a real-time response, limited only by the speed of the image transfer mechanism.

The dynamic resolution of the system averaged 0.66 seconds per Eye Image, depending upon the number or rows skipped by the IHM during image processing. Row skipping involves skipping rows of the Eye Image during image processing. By doing so, a small percentage of error may be introduced to the vertical component of the user’s visual line of gaze. This study has found that skipping every other row of an Eye Image reduces processing time by 0.36 seconds per image, whilst introducing a -1.73 pixel error for 20% of images. As the speed of execution is important to the acceptance of an eye-gaze tracking system by mainstream user, the researcher felt that this small degree of error was acceptable.

The maximum static resolution of the system is 42 horizontal regions by 23 vertical regions for a user sitting 550mm from the VBEGT system camera. This provides an improvement over those systems discussed in the literature review that were unable to provide a resolution exceeding 26 by 20 individually distinguishable regions on a user’s monitor. The static resolution exhibited by the VBEGT system developed for this study indicates the potential for the development of a commercially viable VBEGT system that would be considered acceptable for mainstream computing use. In order to improve the static resolution of the developed system, a zoom lens would need to be attached to the camera, user moved closer or a higher resolution imaging element installed in the camera.
By design the VBEGT system developed for this study satisfied a number of mainstream computing requirements: it does not interfere with the user on either a physical or psychological level; nor does it obstruct a user's field of view; it accommodates a wide variety of users; and, enables the use of eye movements as computer input.

Typical user eye movements used by selective gaze enabled systems include fixations, which average between 200 and 600 milliseconds depending upon the complexity of the scene being viewed, and saccades that last for 30 to 120 milliseconds. In order to detect these movements, a given eye-gaze tracking system would need to operate at a rate of 0.03 to 0.12 seconds and 0.06 to 0.20 seconds per image respectively.

Despite the limited dynamic resolution and speed of response, the VBEGT system developed for this study exhibits the core functionality required to enable the use of eye movements as computer input and highlights the potential for the development of a commercially viable VBEGT system designed for mainstream computer users.
Despite their abilities to process and convey vast amounts of data, humans and computers communicate using narrow-bandwidth interaction techniques (e.g., mouse), which have remained constant since the 1970s. The literature reviewed, agreed that the future of HCI centres upon the creation of faster, natural and convenient methods of transmitting information from the user to a given computer work station. Eye-gaze tracking was identified by the literature as a promising high-bandwidth interaction technique.

Commercial eye-gaze tracking systems range from intrusive systems that use specialised contact lenses placed over the user’s iris, or infra red light projected into the user’s eye. The literature reviewed, showed that in order to be considered practical for mainstream use a given eye-gaze tracking system must not only operate in an accurate and timely fashion, it must not impose any restrictions upon the user or cause either physical or psychological interference, for example, contact lenses. The most promising eye-tracking technique for mainstream computing was identified as video-based techniques due to its non-intrusive nature.

This study revolved around the creation of a video-based technique that utilised recent advances in digital image capture technology and an increase in computer processor speeds, in order to improve upon existing systems. The most notable advancement in the technology utilised by this study, was the significant increase in image resolution from 640 x 480 pixels to 1280 x 1024 pixels. Previous studies identified by the literature, were able to achieve a static resolution of approximately 20 x 20 regions using a resolution of 640 x 480 pixels. At a distance of 550mm, this study demonstrated an increased static resolution of 42 x 23 regions was possible by using the increased image resolution of 1280 x 1024 pixels - essentially double the horizontal accuracy of previous studies.
In addition to an increase in resolution, contemporary image capture technology has improved the speed at which the image is transferred from the camera to computer. The cameras used by the significant studies included USB 1.1 compliant cameras. The USB 1.1 interface supports transfer speeds up to 12 Mbytes per second. Contemporary cameras support the FireWire400 interface. FireWire 400 is capable of transfer rates up to 50Mbytes per second. Assuming a negligible overhead for image processing and gaze determination, a VBEGT system supporting USB 1.1 would achieve a maximum response time of approximately 76.80 ms as opposed to 18.43 ms made possible by the FireWire 400 interface. In reality however, such frame rates are impossible to achieve, due to the overhead caused by image processing. As computer processors advance, image-processing times will decrease until VBEGT system response times approach the theoretical limit of image transfer medium.

System testing showed that a dynamic resolution of 0.30 seconds per image was achieved by the study’s VBEGT system due to the use a row skipping algorithm. This algorithm reduced execution times by 0.36 seconds per image by skipping every other row in the Eye Image during processing. Previous studies identified by the literature were achieving speeds between 0.627 seconds and 5 seconds depending on whether their image-processing algorithm was optimised.

The row skipping algorithm used by this study, was found to introduce an average error of -1.73 pixels for 20% of images for the vertical component of the calculated visual line of gaze. The horizontal component remains unchanged. This study believes that the significant increase in the system’s processed frame rate (i.e., dynamic resolution) justifies the introduction of this small error for an average of one in five frames.

Determining the suitability of the VBEGT system developed by this study in terms of its suitability for mainstream computing use required evaluation in terms of the critiques outlined by Hallet (1986). Hallet’s critiques were devised with all forms of eye-
gaze tracking in mind, for example, psychological research, gaze contingent systems and selective systems. Therefore, a number of these critiques are not applicable to the eye-gaze tracking system developed for this study, as it was designed for use in selective systems.

In order to track user eye movements successfully and still satisfy the applicable critiques outlined by Hallet (1986), the VBEGT developed for this study used a one megapixel WebCam. It was shown that by mounting the WebCam at the bottom of the monitor facing upwards, as compared to placing it at the top of the monitor facing downwards as per a typical video-conferencing position, the system was able to detect a larger proportion of user eye movements. Such a configuration resulted in both the user and the VBEGT system having an unobstructed view of each other during operation. This configuration results in the system being termed non-intrusive as no physical contact is made with the user, or restrictions imposed due to the nature of the equipment. The literature examined by this study concluded that a non-contact passive monitoring method was essential for a given eye-gaze tracking system to be classified as being suitable for mainstream computing use.

In addition to the passive monitoring, mainstream computing users require a given eye-gaze tracking system to be able to respond to eye movements in real-time. As discussed by this study, eye-gaze tracking systems are unable to provide a true real-time response to user eye movements due to the effects of artificial image stabilizing. The VBEGT system developed during this study used a public domain algorithm for detecting the state of user eye movement. This algorithm required a minimum a three gaze points in order to determine whether the eye was currently fixating or performing saccadic movement. If three consecutive gaze points were detected, the algorithm returns the average of these gaze points. By returning an average visual line of gaze, the system was able to provide a stabilized representation that was unaffected by the nystagmic movements of the user’s eye during fixation.
Eye-gaze tracking systems monitor the two main forms of eye movement that enable the perception of an object, i.e., saccadic movements and fixation. Saccades generally last for 30 to 120 millisecond, i.e., 0.03 and 0.12 seconds respectively. Such movement would be undetectable using the VBEGT system developed for this study, as the processed frame rates supported are longer than the duration of a typical saccade.

Fixations are known to last between 200 and 600 milliseconds depending upon the complexity of the scene viewed. Using the definition for the start of fixation used by LC Technologies Inc, wherein a user’s gaze must remain within a given area for more than three consecutive processed frames, the eye-gaze tracking system would be required to achieve processing speeds between 0.06 and 0.20 seconds per image. Again such movement would be undetectable using the VBEGT system developed for this study, as the processed frame rates supported are substantially longer than the duration of a typical fixation.

This study showed that technological improvements have permitted an improvement in the field of eye-gaze tracking, in terms of eye movement detection and increased image-processing speeds. However, the author feels that technology has not advanced to the point where a fine-grained real-time eye-gaze tracking system for suitable for mainstream computing use may be developed, based upon the technology made available to the study. Once processor speeds have increased to the point where they are able to process images at speeds comparable to the throughput of the image transfer mechanism, VBEGT systems will be able to respond to user eye movements real-time. Increases in image capture technology, in terms of a higher resolution image sensor, will allow for a finer grade of eye movement to be detected. This increase would result in the user’s monitor being divided into a greater number of regions for input.

Suggested areas for future research includes: the accommodation of user head movement; the development of alternate methods for image pre-processing; and, faster algorithms for both image processing and pre-processing.
Presently, eye-gaze tracking systems are limited, both in practice and in the market place. The is due to the current systems being intrusive or having not reached a stage where they are of sufficient and accuracy to be used by mainstream computer users. Development of a VBEGT system to fill this commercial void has been hindered by image capture technology and computer processor speeds. Despite the limited dynamic resolution and speed of response, the VBEGT system developed for this study exhibits the core functionality required to enable the use of eye movements as computer input. This study also highlights the potential for the development of a commercially viable VBEGT system designed for mainstream computer use.
## APPENDIX A  DEFINITION OF TERMS

<table>
<thead>
<tr>
<th>Term</th>
<th>Definition</th>
<th>Source</th>
</tr>
</thead>
<tbody>
<tr>
<td>AGC</td>
<td>Automatic Gain Control</td>
<td>Acronym</td>
</tr>
<tr>
<td>Ambient Light</td>
<td>The natural light present in a scene.</td>
<td>(Microsoft Corporation 2002)</td>
</tr>
<tr>
<td>Automatic Gain Control</td>
<td>A process or means by which gain is automatically adjusted to improve the quality of a given specimen. This process is commonly used in image and audio processing.</td>
<td>(Telecom Glossary 2K 2001)</td>
</tr>
<tr>
<td>Automatic White Balance</td>
<td>The automatic calculation of the white balance for a given image as calculated by the image capture device.</td>
<td>See Section 3.2.1.2 (Bockaert, 2004)</td>
</tr>
<tr>
<td>Bitmap Image</td>
<td>An uncompressed image file format.</td>
<td>(Howe, 1999)</td>
</tr>
<tr>
<td>CI</td>
<td>Console Interface</td>
<td>Acronym</td>
</tr>
<tr>
<td>CMOS</td>
<td>Complementary Metal-Oxide Semiconductor</td>
<td>Acronym</td>
</tr>
<tr>
<td>Colour Temperature</td>
<td>The temperature, measured in degrees Kelvin, to which an object would have to be heated before it would radiate a given colour.</td>
<td>See Section 3.2.1.2 (Bockaert, 2004)</td>
</tr>
<tr>
<td>Complementary Metal-Oxide Semiconductor</td>
<td>One of the two major types of image sensors used in digital cameras and web cams.</td>
<td>(Microsoft Corporation 2002)</td>
</tr>
<tr>
<td>Console Interface</td>
<td>An application developed for this study to provide an interface to the MUM. This interface allows the positioning of the MUM during system calibration and testing.</td>
<td>See Section 4.2.2.2.1</td>
</tr>
<tr>
<td>Term</td>
<td>Definition</td>
<td>Source</td>
</tr>
<tr>
<td>------------------------------------------</td>
<td>----------------------------------------------------------------------------------------------------------------------------------------------</td>
<td>---------------------------------</td>
</tr>
<tr>
<td>Corneal Reflection</td>
<td>A method of eye-gaze tracking that measures the reflection of infrared light from the user's cornea.</td>
<td>See Section 2.2.1</td>
</tr>
<tr>
<td>Device Frame Rate</td>
<td>A measure of the time taken to capture images and transfer them to a host computer.</td>
<td>See Section 2.2.2.1.2</td>
</tr>
<tr>
<td>Dwell Time</td>
<td>Marks the time for which a user is required to gaze (fixate) upon an object to initiate object selection.</td>
<td>(Backer &amp; Peral, 1997)</td>
</tr>
<tr>
<td>Edge Detection</td>
<td>The process whereby values are assigned pixels in proportion to the likelihood that they are part of an image edge (i.e., a pixel that is on the boundary between two regions of different intensity values). This is an identification process only, to extract the edges requires them to be grouped and verified using a different process.</td>
<td>(Fisher, Perkins, Walker &amp; Wolfart, 2000b)</td>
</tr>
<tr>
<td>EIHEA</td>
<td><strong>Existing Iris Hypothesis Extension Algorithm</strong></td>
<td>Acronym</td>
</tr>
<tr>
<td>EMM</td>
<td><strong>Eye Movement Model</strong></td>
<td>Acronym</td>
</tr>
<tr>
<td>Existing Iris Hypothesis Extension Algorithm</td>
<td>An algorithm developed for this study, which allows Iris Hypotheses to grow line by line as matching Line Segments are extracted from the Eye Image.</td>
<td>See Section 4.1.2.3.4 Error! Reference source not found.</td>
</tr>
<tr>
<td>Eye Image</td>
<td>An image captured by the VBEGT system, which contains the user's eyes.</td>
<td></td>
</tr>
<tr>
<td>Eye Movement Model</td>
<td>A mechanical model eye that is capable of moving about two degrees of freedom to mimic user eye movements.</td>
<td>See Section 4.2.1.1</td>
</tr>
<tr>
<td>Eye-Gaze Tracking</td>
<td>The process of converting user eye movements and the visual line of gaze to computer input.</td>
<td>(Backer &amp; Peral, 1997)</td>
</tr>
<tr>
<td>Term</td>
<td>Definition</td>
<td>Source</td>
</tr>
<tr>
<td>----------------------------------</td>
<td>-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
<td>---------------------------------------------</td>
</tr>
<tr>
<td>Eye-Typing</td>
<td>A HCI technique primarily for people with severe disabilities where eye movements directed towards an on-screen keyboard represent the typing of a letter or execution of a keyboard command.</td>
<td>(White &amp; Hutchinson, 1990)</td>
</tr>
<tr>
<td>Feature Extraction</td>
<td>The process of locating and extracting specific areas or features from an image.</td>
<td>(Fisher et al., 2000b)</td>
</tr>
<tr>
<td>FireWire 400</td>
<td>A cross-platform implementation of the IEEE 1394a standard for high-speed serial data communication. It is capable of moving large amounts of data between computers and peripheral devices.</td>
<td>(Apple Computer Inc., 2002)</td>
</tr>
<tr>
<td>Fitts' Law</td>
<td>A quantitative model that predicts the time taken to move the hand or other limb to a target, based upon the distance to move and the size of the target.</td>
<td>(I. S MacKenzie, 1995)</td>
</tr>
<tr>
<td>Fixation</td>
<td>The stable (or relatively stable) state when a saccade reaches its destination. At this point a person is said to be viewing an object.</td>
<td>(Hyrskykari, 1997)</td>
</tr>
<tr>
<td>Fixation Hypothesis</td>
<td>A term used by LC Technologies Inc. to represent the presumed position of user gaze.</td>
<td>(Logitech Inc., 2002)</td>
</tr>
<tr>
<td>Frame Rate</td>
<td>The number of image frames displayed/captured every second measured in frames per second (fps).</td>
<td>See Section 3.2.4</td>
</tr>
<tr>
<td>Gaze</td>
<td>A sequence of one or more consecutive fixations made by a user on a given target area.</td>
<td>(Salvucci &amp; Anderson, 2001)</td>
</tr>
<tr>
<td>Gaze-Contingent Eye-Gaze Tracking System</td>
<td>A system that uses the user’s visual line of gaze to facilitate the rendering of complex displays.</td>
<td>(Duchowski, 2002)</td>
</tr>
<tr>
<td>Term</td>
<td>Definition</td>
<td>Source</td>
</tr>
<tr>
<td>-------------------------------</td>
<td>-----------------------------------------------------------------------------</td>
<td>-------------------------</td>
</tr>
<tr>
<td>Glint</td>
<td>A reflection occurring on the user's iris</td>
<td></td>
</tr>
<tr>
<td>HCI</td>
<td>Human-Computer Interaction</td>
<td>Acronym</td>
</tr>
<tr>
<td>Head Movement Model</td>
<td>A mechanical model that is capable of moving about three degrees of freedom to mimic a user's head movements.</td>
<td>See Section 4.2.1.1</td>
</tr>
<tr>
<td>Histogram Equalisation</td>
<td>An image pre-processing technique designed to improve the contrast and dynamic range of captured images before feature extraction.</td>
<td>(Fisher et al., 2000c)</td>
</tr>
<tr>
<td>HMM</td>
<td>Head Movement Model</td>
<td>Acronym</td>
</tr>
<tr>
<td>Human-Computer Interaction</td>
<td>The study of how humans interact with computers, and how to design computer systems that are easy, quick and productive for humans to use.</td>
<td>(Howe, 1999)</td>
</tr>
<tr>
<td>Human-Computer Interface</td>
<td>Any piece of software or hardware that facilitates interaction between humans and computers.</td>
<td>(Howe, 1999)</td>
</tr>
<tr>
<td>IEEE-1394a</td>
<td>The IEEE specification for FireWire 400.</td>
<td>See FireWire 400</td>
</tr>
<tr>
<td>IHCA</td>
<td>Iris Hypothesis Creation Algorithm</td>
<td>Acronym</td>
</tr>
<tr>
<td>IHM</td>
<td>Iris Hypothesis Method</td>
<td>Acronym</td>
</tr>
<tr>
<td>Image Frame Rate</td>
<td>Time taken to capture an image – does not include the time taken to transfer that image to a host computer.</td>
<td>See Section 2.2.2.1.2</td>
</tr>
<tr>
<td>Term</td>
<td>Definition</td>
<td>Source</td>
</tr>
<tr>
<td>----------------------------------------------------</td>
<td>-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
<td>--------------------------------</td>
</tr>
<tr>
<td>Interaction Technique</td>
<td>The use of a physical input device (e.g., a mouse) to perform generic interaction tasks within human-computer dialogues. It represents an abstraction of some common class of interactive task, for example, choosing one of several objects shown on a display screen.</td>
<td>(Jacob, 1991)</td>
</tr>
<tr>
<td>Interactive Gaze Application</td>
<td>An application that responds to, or interacts with, observed user eye movements.</td>
<td>(Duchowski, 2002)</td>
</tr>
<tr>
<td>Intrusive Eye-Gaze Tracking Techniques</td>
<td>Eye-gaze tracking whereby the tracking device makes direct contact with the user and/or involves the use of equipment designed to restrict user movement. Notably, intrusive techniques are highly accurate.</td>
<td></td>
</tr>
<tr>
<td>Iris Hypothesis Creation Algorithm</td>
<td>An algorithm developed for this study, which allows Iris Hypotheses to be created using Line Segments extracted from a given Eye Image.</td>
<td>See Section 4.1.2.3.2</td>
</tr>
<tr>
<td>Iris Hypothesis Method</td>
<td>The custom feature extraction algorithm implemented by this study to locate the user’s iris within a given Eye Image.</td>
<td>See Section 3.2.2</td>
</tr>
<tr>
<td>ITU-T</td>
<td>International Telecommunication Union - Telecommunication Standardization Sector</td>
<td></td>
</tr>
<tr>
<td>Joint Photographic Experts Group</td>
<td>The committee, which created the JPEG image compression algorithm. The standardisation bodies involved in developing JPEG were ISQ and ITU-T.</td>
<td>(JPEG, 2004)</td>
</tr>
<tr>
<td>JPEG</td>
<td>Joint Photographic Experts Group Acronym</td>
<td></td>
</tr>
<tr>
<td>Term</td>
<td>Definition</td>
<td>Source</td>
</tr>
<tr>
<td>----------------------------------------------</td>
<td>-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
<td>-----------------------------</td>
</tr>
<tr>
<td>JPEG Image</td>
<td>An image stored using the lossy compression technique developed by the Joint Photographic Experts Group. Although it can reduce file sizes by approximately 5% or their normal size, some detail is lost in the compression.</td>
<td>(Webopedia 2004)</td>
</tr>
<tr>
<td>Last Line Segment Added Threshold</td>
<td>A threshold value used by the IHCA to determine whether a given hypothesis is still valid based upon the number of rows that have passed since the addition of the last Line Segment.</td>
<td>See Section 4.1.2.3.2</td>
</tr>
<tr>
<td>Line Segment Combination Algorithm</td>
<td>An algorithm developed for the study, which combines Line Segments under the assumption that a glint is present in the image. When glints are present in the image, this causes a Line Segment to be extracted incorrectly as two separate segments.</td>
<td>See Section 4.1.2.3.5</td>
</tr>
<tr>
<td>Line Segment Extraction Algorithm</td>
<td>An algorithm developed for this study, which extracts Line Segments from a given Eye Image based upon the intensity of the pixels present in each row of the image.</td>
<td>See Section 4.1.2.3.1</td>
</tr>
<tr>
<td>Line Segment Matching Algorithm</td>
<td>An algorithm developed for this study, which matches Line Segments to Iris Hypotheses.</td>
<td>See Section 4.1.2.3.3</td>
</tr>
<tr>
<td>LLSAT</td>
<td>Last Line Segment Added Threshold</td>
<td>Acronym</td>
</tr>
<tr>
<td>LSCA</td>
<td>Line Segment Combination Algorithm</td>
<td>Acronym</td>
</tr>
<tr>
<td>LSEA</td>
<td>Line Segment Extraction Algorithm</td>
<td>Acronym</td>
</tr>
<tr>
<td>LSMA</td>
<td>Line Segment Matching Algorithm</td>
<td>Acronym</td>
</tr>
<tr>
<td>Term</td>
<td>Definition</td>
<td>Source</td>
</tr>
<tr>
<td>----------------------------------------------------</td>
<td>----------------------------------------------------------------------------------------------</td>
<td>-------------------------------</td>
</tr>
<tr>
<td>Mechanical User Model</td>
<td>A mechanical model developed for this study</td>
<td>See Section 4.2</td>
</tr>
<tr>
<td>Mechanical User Model Control Unit</td>
<td>A software component developed for this study, which allows the EMM and HMM to be moved independently to a given position on screen.</td>
<td>See Section 4.2.2</td>
</tr>
<tr>
<td>Movement Script</td>
<td>A precompiled script containing commands sent to the MUMCU to move the EMM and/or HMM to given screen positions.</td>
<td>See Section 4.2.2.2.2</td>
</tr>
<tr>
<td>Movement Script Generator</td>
<td>A software tool developed for this study, which allows Movement Scripts to be generated automatically based upon a set of variables provided by the user.</td>
<td>See Section 4.2.2.2.2</td>
</tr>
<tr>
<td>Moving Average</td>
<td>A digital filtering technique used by modern video cameras to correct jittery movements made by the operator, or other forms of random noise.</td>
<td>(Smith, 1999)</td>
</tr>
<tr>
<td>MSG</td>
<td><strong>Movement Script Generator</strong> Acronym</td>
<td></td>
</tr>
<tr>
<td>MUM</td>
<td><strong>Mechanical User Model</strong> Acronym</td>
<td></td>
</tr>
<tr>
<td>MUMCU</td>
<td><strong>Mechanical User Model Control Unit</strong></td>
<td></td>
</tr>
<tr>
<td>Non-Intrusive</td>
<td>Any input technique that does not require physical contact be made with or restricts user movement.</td>
<td></td>
</tr>
<tr>
<td>Non-Specialised Computing Application</td>
<td>A mainstream computing application, for example, an Internet web browser or word processor.</td>
<td></td>
</tr>
<tr>
<td>Nystagmus</td>
<td>Small eye movements made involuntarily during fixation to stimulate the retina.</td>
<td>(All About Vision 2004)</td>
</tr>
<tr>
<td>Term</td>
<td>Definition</td>
<td>Source</td>
</tr>
<tr>
<td>------</td>
<td>------------</td>
<td>--------</td>
</tr>
<tr>
<td>Occupational Overuse Syndrome</td>
<td>Occupational Overuse Syndrome is also known as Repetition Strain Injury.</td>
<td>(National Occupational Health And Safety Commission 1998) See Repetition Strain Injury</td>
</tr>
<tr>
<td>OOS</td>
<td>Occupational Overuse Syndrome</td>
<td>Acronym</td>
</tr>
<tr>
<td>Pinpoint Accuracy</td>
<td>A measurement of accuracy in which a system calculates Cartesian pixel coordinates rather than a rectangular range of pixels.</td>
<td></td>
</tr>
<tr>
<td>Processed Frame Rate</td>
<td>A measure of the time taken to process a given image i.e., perform pre-processing; determine eye state; and, visual line of gaze. When combined with device frame rate, this dictates the responsiveness of a given VBEGT system.</td>
<td>See Section 2.2.2.1.4</td>
</tr>
<tr>
<td>Real-Time</td>
<td>Describes an application, which is required to respond to a given stimuli within some small upper limit of response time (typically milliseconds or microseconds).</td>
<td>(Howe, 1999)</td>
</tr>
<tr>
<td>Real-Time Video-Based Eye-Gaze Tracking System</td>
<td>A real-time system implemented for video-based eye-gaze tracking.</td>
<td></td>
</tr>
<tr>
<td>REB</td>
<td>Replica Eye Base</td>
<td>Acronym</td>
</tr>
<tr>
<td>REC</td>
<td>Replica Eye Cap</td>
<td>Acronym</td>
</tr>
<tr>
<td>Region Of Interest</td>
<td>The ability to specify the region of interest allows a developer to utilise the available bandwidth of the image transfer medium by only sending a specific part of a captured image, e.g., the region containing the iris. In doing so, the available bandwidth of the transfer medium is not consumed by extraneous data.</td>
<td></td>
</tr>
<tr>
<td>Term</td>
<td>Definition</td>
<td>Source</td>
</tr>
<tr>
<td>-------------------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
<td>-----------------------------------------------------------------------</td>
</tr>
<tr>
<td>Repetition Strain Injury</td>
<td>It is a collective term for a range of conditions characterised by discomfort of persistent pain in muscles, tendons and other soft tissues in the back, neck, shoulder, elbows, wrists, hands, and fingers.</td>
<td>(National Occupational Health And Safety Commission 1998)</td>
</tr>
<tr>
<td>Replica Eye</td>
<td>The eye shaped component of the EMM. It is comprised of the REC and REB.</td>
<td>See Section 4.2.1.2</td>
</tr>
<tr>
<td>Replica Eye Base</td>
<td>The main structural component of the Replica Eye, which allows the REC to be attached.</td>
<td>See Section 4.2.1.2</td>
</tr>
<tr>
<td>Replica Eye Cap</td>
<td>A covering for the REB that is modelled upon the characteristics of a human eye, including a coloured iris, pupil and blood vessels.</td>
<td>See Section 4.2.1.2</td>
</tr>
<tr>
<td>Roberts Cross Operator</td>
<td>This operator performs a 2-D spatial gradient measurement on an image to determine the location of the edges of a given feature within that image.</td>
<td>(Fisher, Perkins, Walker &amp; Wolfart, 2000e)</td>
</tr>
<tr>
<td>Rolling Shutter</td>
<td>A rolling shutter works similar to a focal plane shutter in a film camera. Typically, rows of pixels in the image sensor are read out in sequence starting from the top of the image and proceeding row by row to the bottom of the image.</td>
<td>(Kodak Inc 2003)</td>
</tr>
<tr>
<td>RS232</td>
<td>RS232, now known as EIA232, is a common interface standard for data communications equipment, developed by the Electronic Industries Association.</td>
<td>(Strangio, 2003)</td>
</tr>
<tr>
<td>RSI</td>
<td>Repetition Strain Injury Acronym</td>
<td>Acronym</td>
</tr>
<tr>
<td>Saccade</td>
<td>Sudden rapid eye movements used to move the eye-gaze to a new position in space.</td>
<td>(Hyrskykari, 1997)</td>
</tr>
<tr>
<td>Term</td>
<td>Definition</td>
<td>Source</td>
</tr>
<tr>
<td>-------------------------------------</td>
<td>----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
<td>-------------------------------</td>
</tr>
<tr>
<td>Screen Positional Accuracy</td>
<td>A function of camera resolution where the relationship between the numbers of pixels in a captured image comprising the maximum displacement of the pupil from a given point determines the maximum number of distinguishable input regions on a computer screen.</td>
<td>See Section 2.2.2.1.1</td>
</tr>
<tr>
<td>Seed Line Segment</td>
<td>A Seed Line Segment is the most recent Line Segment to be included in a given hypothesis. This Line Segment is the basis by which additional Line Segments are examined to determine whether they belong to the current Iris Hypothesis.</td>
<td>See Section 3.2.2.1.2.1</td>
</tr>
<tr>
<td>Selective Eye-Gaze Tracking System</td>
<td>Selective eye-gaze tracking systems use the visual line of gaze as a pointing device.</td>
<td>(Duchowski, 2002)</td>
</tr>
<tr>
<td>Sobel Operator</td>
<td>A method of edge detection, which performs a 2-D spatial gradient measurement on an image to emphasize regions of high spatial frequency. These areas typically correspond to the edges of a given feature within an image. This operator is similar to the Roberts Cross operator.</td>
<td>(Fisher, Perkins, Walker &amp; Wolfart, 2000f)</td>
</tr>
<tr>
<td>Systematic Error</td>
<td>The disparity (or drift) between the average location of the gaze points recorded and the actual fixation maintained by the user.</td>
<td>(Hornof &amp; Halverson, 2002)</td>
</tr>
<tr>
<td>Test Grid</td>
<td>A grid contains a series of boxes, which denote the maximum number of logical screen positions that a user's computer monitor may be divided into during operational system testing.</td>
<td>See Section 3.4</td>
</tr>
<tr>
<td>Term</td>
<td>Definition</td>
<td>Source</td>
</tr>
<tr>
<td>-------------------------</td>
<td>-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
<td>-----------------------------------------------------------------------</td>
</tr>
<tr>
<td>Thresholding</td>
<td>A method used to separate the regions of an image, which correspond to the objects present of interest, from those regions that correspond to background.</td>
<td>(Fisher, Perkins, Walker &amp; Wolfart, 2000g)</td>
</tr>
<tr>
<td>Torsional Rotation</td>
<td>The process of keeping the eye orientated straight up and down when the head is moved from side to side.</td>
<td>(All About Vision 2004)</td>
</tr>
<tr>
<td>Uni-ocular Tracking</td>
<td>Eye-gaze tracking which uses the movement of only one of the user’s eyes to determine the location and state of gaze.</td>
<td></td>
</tr>
<tr>
<td>Universal Serial Bus</td>
<td>A fast, bi-directional, isochronous, low-cost, dynamically attachable serial interface that is consistent with the requirements of the PC platform of today and tomorrow.</td>
<td>(Compaq Computer Corporation, Intel Corporation, Microsoft Corporation &amp; NEC Corporation, 1998)</td>
</tr>
<tr>
<td>USB</td>
<td>Universal Serial Bus</td>
<td>Acronym</td>
</tr>
<tr>
<td>USB 1.1</td>
<td>The original specification for the Universal Serial Bus developed in 1995. The major goal of USB was to define an external expansion bus, which makes adding peripherals to PCs easy as hooking up a telephone to a wall-jack.</td>
<td>(USB Implementers Forum Inc, 2004)</td>
</tr>
<tr>
<td>USB 2.0</td>
<td>An evolution of the USB 1.1 specification, which provides a higher performance interface. USB2.0 is able to achieve data transfer rates of up to 480Mb/s as opposed to the 12Mb/s supported by USB 1.1.</td>
<td>(USB Implementers Forum Inc, 2004)</td>
</tr>
<tr>
<td>Valid Line Segment Array</td>
<td>An array storing valid Line Segments that have been extracted from a given Eye Image. These Line Segments are then formed into Iris Hypotheses by the IHM,</td>
<td>See Section 4.1.2.3.1</td>
</tr>
<tr>
<td>Term</td>
<td>Definition</td>
<td>Source</td>
</tr>
<tr>
<td>-------------------------------------</td>
<td>----------------------------------------------------------------------------</td>
<td>---------------------------------------------</td>
</tr>
<tr>
<td>Variable Error</td>
<td>The spread (or dispersion) of the individual gaze points recorded for a given fixation.</td>
<td>(Hornof &amp; Halverson, 2002) See Section 2.2.2.2</td>
</tr>
<tr>
<td>VBEGT</td>
<td>Video-Based Eye-Gaze Tracking</td>
<td>Acronym</td>
</tr>
<tr>
<td>Vergence</td>
<td>Refers to the eye’s ability to turn either inward (convergence) or outward (divergence).</td>
<td>(All About Vision 2004)</td>
</tr>
<tr>
<td>Video-Based Eye-Gaze Tracking</td>
<td>A method for eye-gaze tracking that involves the processing of images of user eye movements in order to determine the user’s visual line of gaze.</td>
<td></td>
</tr>
<tr>
<td>Videoconferencing</td>
<td>A videoconference is a live connection between people in separate locations for the purpose of communication, usually involving audio and often text as well as video. Videoconferencing generally takes place via the use of webcams.</td>
<td>(Wikipedia, 2004b)</td>
</tr>
<tr>
<td>Visual Line Of Gaze</td>
<td>A line projecting forward in space from the eye indicating that the user is looking at something along that line.</td>
<td>(Jacob, 1993)</td>
</tr>
<tr>
<td>Visually Mediated Application</td>
<td>An eye-gaze enabled human-computer interface.</td>
<td>(Duchowskii, 2002)</td>
</tr>
<tr>
<td>Visually Mediated Interaction</td>
<td>Interaction between two people where in visual clues are given to indicate objects of interest.</td>
<td>(Howell &amp; Buxton, 2002)</td>
</tr>
<tr>
<td>Webcam</td>
<td>A real-time camera whose images may be accessed by the World Wide Web. It is commonly used for personal videoconferencing.</td>
<td>(Wikipedia, 2004c)</td>
</tr>
<tr>
<td>Term</td>
<td>Definition</td>
<td>Source</td>
</tr>
<tr>
<td>------------------</td>
<td>-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
<td>------------------------------</td>
</tr>
<tr>
<td>White Balance</td>
<td>Most cameras can adjust for color temperature by zooming into a white object and setting the white balance (telling the camera that a given object is white); the camera then shows true white as white and adjusts all the other colors accordingly. White-balancing is necessary especially indoors under fluorescent lighting and when moving the camera from one lighting situation to another.</td>
<td>(Wikipedia, 2004d)</td>
</tr>
</tbody>
</table>
APPENDIX B  
BENEFITS OF EYE-GAZE TRACKING

Eye-gaze tracking is described by Jacob (1993) as "a possible means of high-bandwidth communication between human and computer" (p. 151). This description leads to eye-gaze tracking being regarded as a possible future input technique (Jacob et al., 1993). Beyond providing high-bandwidth communications, eye-gaze tracking has a number of other virtues, which continue interest in the development of this technique. These virtues are discussed in the following:

Natural Mode of Input: Typically, when operating a mechanical pointing device, the user tends to focus (or fixate) upon the destination to which they want to move the cursor before actuating its movement. In this way, fixation provides an indication of the user’s goal making it is possible to bypass the need for a mechanical input device (Jacob, 1994a).

The naturalness afforded by eye movements as input allows the creation of human-computer interfaces that are easy to operate. As noted by Jacob (1994a, p. 260):

“No training or particular coordination is required of normal users for them to be able to cause their eyes to look at an object; and the control-to-display relationship for this device is already established in the brain.”

Consequently, eye-gaze may be used as an implicit form of user input as opposed to mechanical input devices, for example the mouse, where users are required to have mastered the hand-eye co-ordination necessary to effect operations such as object selection (Jacob, 1993).
Information From The User: Mechanical input devices are known to provide information concerning where the user happens to point (intentionally or unintentionally), whereas eye-gaze gives additional information concerning user attention (Hyrskykari, 1997). As summarised by Jacob (1994a, p. 260):

“[whenever] the user changes his or her focus or attention, [this] change of focus is available as a pointing command to the computer. Mouse input tells the computer system simply that the user has intentionally picked up the mouse and pointed it at something. Eye tracker input may be interpreted in the same way (the user intentionally pointed his or her eye at something). But it may also be interpreted as an indication of what the user is currently paying attention to, without any explicit input action on his or her part.”

Such additional information extends the application of eye-gaze tracking systems beyond the simplistic control of computer systems where the intentional movement of a user’s eyes is used to perform pre-programmed tasks, for example, selecting and clicking a button. Being able to determine the object of a user’s attention is useful in a number of ways, for example:

“as a methodology, eye-tracking has the potential to aid traditional software usability evaluation by recording user’s flow of visual attention on displays” (Goldberg, Stimson, Lewenstein, Scott & Wichansky, 2002, p. 1).

“The data [collected] is objective [thus] eliminating much of the subjectivity found in traditional forms of user testing [and] provides circumstances whereby it is possible to benchmark improvements or changes over time and undertake highly comparative test[ing]” (Software Engineering Australia, 2003a, p. 60).
Allows Interaction Where Users Are Unable To Use Their Hands: The ability to “express oneself in a fast and effect manner is fundamental to one’s quality of life” (Hansen, Hansen & Johansen, 2001, p.330). A number of eye-tracking studies have developed systems designed specifically for disabled use that allow a user to communicate via computer in cases where communication may not have previously been possible (Hansen et al., 2001). Notably these include Camera Mouse (Betke, Gips & Fleming, 2001), ERICA (White & Hutchinson, 1990) and EagleEyes (Gips, DiMattia, Curran & Olivieri, 1996). These systems highlight, by case study, the success of eye-gaze tracking in cases where users do not possess the fine motor skills necessary to effect human-computer interaction using traditional means. Betke, Gips and Fleming (2001) suggest that:

“in the near future, standard desktop computers will be equipped with cameras. This will give rise to a new generation of assistive technologies that do not involve customised, expensive electro-mechanical devices to accommodate special needs, but instead are software based” (p. 5).

In this way, as eye-gaze tracking systems are expected to become more user-friendly (non-intrusive, accurate and responsive). Owing to this, their incorporation into mainstream computer systems is expected to increase, particularly in disability services as “handicapped people (i.e., those who may not be able to perform hand/body movements), are usually able to move their direction of gaze freely” (Backer & Peral, 1997, p. 1). “The physically disabled have the most to gain and the greatest dependence on computer and electronic aids for work, recreation environmental control and even for the most basic communication needs” (Kaufman, Bandopadhay & Shaviv, n.d, 1.).

VBEGT systems have applications in fields other than disability services. For example, virtual reality interfaces, exploit the user’s natural navigational commands, such as head position and eye-gaze (Jacob, 2002), whilst leaving the user’s hands free for other tasks within the system including object manipulation (Tanriverdi & Jacob, 2000).
Reduction Of Workload: The operation of a mechanical pointing device requires the user to look at the destination to which they wish to move on the computer display before moving the cursor (Hornof & Halverson, 2002). Considering that eye-gaze tracking involves the monitoring of user eye movements, this initial step is all that is required to move the cursor (Hyrskykari, 1997). Subsequently the need for the time-consuming operations of locating, grasping and moving manual input devices is eliminated (Hyrskykari, 1997). Additionally, by removing the need for the repeated movement of a mechanical device it is “possible to remove muscle strain caused by mouse operations” (Engell-Nielsen et al., 1995, p. 2), which may “cause a user to eventually give up their preferred communication tool due to repetitive strain injury” (Hansen et al., 2001, p 5).

Eye-Gaze Interfaces Are Fast: The subsequent reduction of workload, as discussed previously, causes a resultant speed increase that is “due to extremely rapid eye movements and that using gaze as a pointing device leaves out the actual moving of the pointer to the desired location. With other pointing devices the user must first search the goal with their eyes and after that transfer the pointer to the selected goal” (Hyrskykari, 1997, p. 5). These claims are further substantiated by the remarks made by Hornof & Halverson (2002) wherein it is explained that eye movements are representative of and faster than the movement of a mechanical pointing device:

“Eye-hand coordination research has shown that the eyes are used in the closed loop “current control” phase of rapid eye movement (Woodworth, 1899). Keeping the eyes on the target facilitates getting the hand or the manipulated display item (i.e., the [mouse] cursor) to the target as quickly as possible. As the cursor approaches the target, the remaining distance to the target is monitored and corrective adjustments are made until the target is reached (Rosenbaum, 1991; Wickens, 1984). Eye movement studies confirm that, even when the hands start moving before the eyes, the eyes arrive at the target before the hands and stay on the target for the completion of the aimed movement (Abrams, Meyer & Kornblum, 1990; Helsen, Elliott, Starkes & Ricker, 1998; B. A. Smith, Ho, Ark & Zhai, 2000)” (p. 2).

Therefore, eye movements are indicative of the path travelled by the mouse cursor and thereby cause the physical movements of the hand to be redundant (Hornof & Halverson, 2002; Hyrskykari, 1997). Eye-movement based interfaces are subsequently
able to achieve higher response speeds as the time taken for redundant mouse movements are eliminated. According to Hyrskykari (1997, p. 2):

“There is evidence that interfaces that use gaze control as input are faster than interfaces that use conventional techniques. For example, Ware & Mikaelian (1987) reported the object selection was performed about twice as fast with an eye tracker than with other pointing devices”.

Eye-gaze tracking, in common with traditional input methods (e.g., the mouse), is subject to quantitative evaluation (Accot & Zhai, 1997). The most prevalent technique is Fitts’ Law, one of the few quantitative evaluation tools available to the study of human-computer interaction for determining the time consumed to perform object selection (Card, English & Burr, 1978; I. S. MacKenzie & Buxton, 1992). This law provides:

“a sound measure of aggregate performance and a valuable engineering model for understanding the movement toward a target for human-computer interaction problems. Fitts’ Law may be used to compare input devices performance on tasks that require absolute accuracy with unconstrained movement” (Sibert & Jacob, 2000, p. 285).

Application of Fitts’ Law has supported the works of Sibert, Jacob and Templeman (2001), Istance and Howarth (1994), Hyrskykari (1997), and Stiefelhagen, Yang, and Waibel (1996) in concluding that:

“one may expect performance benefits, particularly in terms of speed of selection, if a user need only look at an object to acquire it, rather than having, in addition, to control and position a cursor by hand” (Istance & Howarth, 1994, p. 2).

“The further you need to move, the greater the advantage of the eye because its cost is nearly constant ... it is excellent for jumping to distinct regions of the screen quickly” (Sibert et al., 2001, p. 3).

Such increases in performance lead to a consideration of eye-gaze tracking as an appealing high-bandwidth mode of input superior to current narrow-bandwidth techniques.
APPENDIX C    CRITIQUE OF EYE-GAZE TRACKING SYSTEMS

Hallet (1986) suggests a list of twelve critical items that may be used to evaluate
the design and operation of an eye-gaze tracking system. These items are based upon the
concept of an "ideal" tracking device. To date "no single technique fully satisfies all the
useability requirements" (Glenstrup & Engell-Nielsen, 1995, p. 10) listed in Section 2.2.3
and discussed as follows.

Unobstructed Field of View: The camera needs to maintain an unobstructed view of the
user's eyes in order to capture the necessary images for tracking. The positions of other
facial features may be required to provide perspective to aid in the determination of the
user's visual line of gaze and head movement.

Some techniques, for example those used by Kim, Kim and Chung (2001), use
head mounted tracking systems which mount a camera onto spectacles which take close­
up images of the user's eye (Kim et al., 2001). These systems require an alternate method
for determining head movement, for example magnetic position sensors (Kim et al.,
2001). Incorporation of additional recording devices require extra processing, thereby
reducing system responsiveness. This technique is intrusive as it requires physical
contact with the user and may obscure user vision as the spectacle-mounted camera
directly faces the user's eye.

Makes No Contact With The Subject: A basic requirement of eye-gaze tracking
systems is that no physical contact is made with the user (Glenstrup & Engell-Nielsen,
1995). VBEGT is implicitly non-intrusive as a majority of systems mount the camera on
or near the monitor to record user eye movements from a distance (Glenstrup & Engell-
Nielsen, 1995). Such monitoring is the basis of the two main advantages of VBEGT
systems described by Mulligan (1997a):
• no physical contact is made with the user’s eye

• user movement is not restricted as long as the user remains in the camera’s field of view.

These advantages form the foundation of the underlying assumption that video-based methods provides are able to monitor user eye movements in a way considered acceptable by mainstream computer users (Quek, 1995). Contact made with the user, for example, in those systems requiring head-mounted equipment, may also restrict their user’s view of the computer and ability to perform a given task (Corno et al., 2002). By any means, physical intrusion upon the user reduces their acceptance of the system and thereby limits its suitability for mainstream use (Abd-Almageed et al., 2002).

Artificial Image Stabilising: As noted by Jacob (1991, p. 161) “during a fixation, a user generally thinks [they are] looking steadily at a single object. [They are] not consciously aware of the small, jittery motions [(nystagmus)].” The eye-tracking device picks up these small movements and may incorrectly interpret these as altering the determined line of gaze. As the user is unaware of these movements Jacob (1991) suggests that “the human-computer dialog should be constructed so that it ignores these motions since, ultimately, it should correspond with what the user thinks [they are] doing, rather that what [their] eye muscles are actually doing.”

To overcome the influence of these movements Hallet (1986) suggests that video-based techniques may apply some form of averaging across captured images. This method is a common digital filtering technique referred to as a moving average. Modern video cameras apply a moving average60 to correct jitter when capturing streaming video without reducing quality (Smith, 1999).

---

60 A digital filtering technique used by modern video cameras to correct jittery movements made by the operator, or other forms of random noise (S. W. Smith, 1999).
Accuracy Of At Least 1% Or A Few Minutes Of Arc: The accuracy of an eye-gaze tracking system is an important feature in the evolution of this input technique (Glenstrup & Engell-Nielsen, 1995). Accuracy “includes all sources of error and is hard to estimate without comparison against an ideal (and non-existent) instrument” (Hallet, 1986, p. 10.27). A usual measurement is the linearity of the plot of actual eye position (i.e., where the user is actually looking), versus assumed eye position calculated by the system.

Due to the structure of the human eye, the maximum possible accuracy of the eye-gaze tracking is limited. This limitation is imposed by the width of the fovea - the region of the eye that gives clear vision (Jacob, 1991). Therefore, the accuracy of an implemented eye-gaze tracking system is approximately 1° of visual arc, hence the eye may be gazing and seeing an object clearly but the eye-tracking angular tolerance may be up to ±1° (Jacob, 1991).

Resolution Of 1 Min Or 1 Min Arc·sec⁻¹: The resolution of the system is important to the development of eye-gaze tracking systems (Mulligan, 1997b). It represents the smallest eye movement that may readily be detected and is established from the noise level of the instrument (Hallet, 1986). Resolution differs from accuracy in that it does not includes all sources of error, for example uncontrolled head movement (Hallet, 1986). VBEGT system resolution is determined by the resolution of the image captured and the frame rate of the capture device (Mulligan, 1997b).

Resolution, in the context of eye-tracking, is considered in terms of static resolution (suggested resolution of 1 Min) and dynamic resolution (suggested resolution of 1 Min Arc·sec⁻¹). When tracking eye movement the static resolution of the system is based upon the system’s ability to determine the point about which eye is fixated i.e., when the eye is “stationary” and the user views an object on the scene (Hyrskykari, 1997). Dynamic resolution is considered in terms of detecting the position of user gaze at predefined time intervals during the movement of the eye, i.e., saccadic movement – the movement of a user’s eye between one fixation point and another (Hyrskykari, 1997).
Wide Dynamic Range For Eye Position And Velocity: The dynamic range of eye-gaze tracking systems depends upon the range through which the system is able to measure the periodic position of the eye. For example, systems that provide input to computer systems require a high degree of accuracy (number of pixels) across a small range (e.g., from one side of the computer monitor to another) whilst virtual reality applications require rapid tracking of the eye across a large range (e.g., from one side of a room to another.)

Tracking of the velocity of the eye is also important and subject to the system’s ability to measure eye position at given time intervals. For example, cognitive studies require constant sampling of user eye movements, whilst interactive gaze applications may not require such rapid sampling of eye movement (Glenstrup & Engell-Nielsen, 1995).

Temporal Dynamics And Speed Of Response: Temporal dynamics are limited by processed frame rate (see Section 2.2.2.1.4) through which some dynamics of a saccade may be lost (Sibert & Jacob, 2000). For example, Sibert and Jacob’s (2000) study recognised that saccades take between 30 and 120 msec. Assuming a VBEGT system captures an image and calculates a user’s visual line of gaze every 1/60 second. The systems will capture between 2 (1.8) and 7 (7.5) frames of saccadic movement.

By decreasing the time taken to capture images and calculate the visual line of gaze, one may increase the number of frames of saccadic movement captured. This increase would improve temporal dynamics by increasing the system’s response times (e.g., detecting saccades and the end of fixations) and reduce the possibilities of the system incorrectly responding to user eye movement, for example overestimating (overshooting) or underestimating (undershooting) fixation duration.
Possess A Real-Time Response: Jacob (1992) suggests that eye-gaze tracking, as a future input technique, permits the creation of a faster, more convenient means to transfer information from human to computer. The speed with which such systems determine the user’s visual line of gaze and provides this as input to a computer is important. VBEGT systems are required to process user input in real-time in order to provide an interface that responds to the user’s eye movements directly and not some time later (Glenstrup & Engell-Nielsen, 1995).

Measure Three Degrees Of Angular Rotation: Polpitiya, Ghosh, Martin, and Schovane (2002) suggests that “if the eye is moved from one fixation to another, in theory, there are unlimited ways to orient the axis about which the eye rotates in three-dimensional space.” Intrusive eye-tracking methods, such as those using mechanical contact lenses, by design, are able to provide these readings. However, when using video-based methods extra processing is required, for example, when measuring the positions of scleral blood vessels, the corneal reflections, or the pupillary centre (Hallet, 1986). This study involves the tracking of the iris. The iris shows a distinct radial structure, the gains of which are explained by Hallet (1986, p. 10.25):

“The iris shows a distinct radial structure. Consequently, if two consecutive video images are digitized and scanned along a circular path centred on the pupil, then cross correlation will give the roll of the eye in the interval between the two video frames.”

The ability to track the three degrees of angular rotation is not essential for selective eye-gaze tracking systems61, as stated by Polpitiya et al (2002, p. 5):

“For steady fixation with the head upright, the actual positions of the eye are restricted in such a way that there is only one eye position for every gaze direction. This restricts the three-dimensional space of all possible orientations to a two-dimensional subspace.”

---

61 Selective eye-gaze tracking systems use the visual line of gaze as a pointing device (Duchowski, 2002).
**Extendable To Binocular Recording:** Binocular recording requires the simultaneous recording of both eyes to be able to determine accurately the position of the user relative to the computer monitor and VBEGT system (Minagawa et al., 1997). These systems are described by Shih and Liu (2002) as being capable of determining a true 3-D visual line of gaze. 2-D techniques, in contrast, are unable to provide full 3-D information concerning the visual line of gaze and the exact position of the eye in 3-D space – these techniques use a relative orientation of a user's eye with respect to the user head movement. The ability to determine a 3-D visual line of gaze may be considered more accurate as it reduces the influence of a user's head movement (Minagawa et al., 1997). Determining the 3-D visual line of gaze is outside the scope of this study.

**Compatible with Head and Body Recordings:** It is uncommon for high-bandwidth techniques to implement eye-gaze tracking exclusively. Normally, other features, including head position, are also tracked to increase system accuracy or provide a 3-D model of the user in space (Perrone, 1995). Multimodal systems combine eye-gaze with other input methods, for example hand movements used in virtual reality (Tanriverdi & Jacob, 2000) to provide an interactive interface for the user (Duchowski, 2002).

**Accommodates A Variety Of Subjects:** Eye-gaze tracking is proposed, by Bates (2002), as an alternative input device to current mechanical methods (e.g., the mouse) and must, therefore, be able to be used by a variety of users. This mode of input centres upon the monitoring of users' eyes, which may differ in iris colour and size, and the user may wear glasses or contact lenses. It is important that eye-gaze tracking systems accommodate these features, along with physiological gaze differences (see Section 2.2.2.2). In doing so, they may provide an input means acceptable to mainstream computer users (Glenstrup & Engell-Nielsen, 1995). An effective VBEGT system, therefore, is required to react predictably to different users.
APPENDIX D  CONSIDERATIONS FOR EYE-MOVEMENT BASED USER INTERFACES

Advancements in computer technologies allow the creation of interactive applications for eye-gaze tracking. These applications allow the user to interact with a computer system by moving their eyes (Jacob, 1991). When designing eye-gaze enabled human-computer interfaces there are a number of concerns that need consideration when creating an interface suitable for use with mainstream software: the "Midas Touch Problem"; multiple fixations in a single gaze; and, visual feedback. Each of these is discussed as follows.

The "Midas Touch Problem": As described by Jacob (1991, p. 168) "the most naïve approach to using eye position as input might be to use it as a direct substitute for a mouse: changes in the user's line of gaze would cause the mouse [pointer] to move." Closer inspection of this concept suggests that it would become annoying as the system interprets every eye movement as a command to the system, resulting in the user attempting to control their eye movements so as not to trigger accidental system commands. This problem is known as the "Midas Touch Problem" wherein the user may not look anywhere without issuing a command. "Normal visual perception requires that the eyes move about, scanning the scene before them." (Jacob, 1991, p. 168) In conclusion, it is infeasible to construct a system that interprets every eye movement of the user as a command directed to the system.

Multiple Fixations In A Single Gaze: As observed by Jacob (1993, p. 155) "a user may view a single object with a sequence of several fixations, all in the general area of the object. They are distinct fixations sets, separated by measurable saccades larger than the jitter [those movements caused by the eye as it moves to stimulate the retina during the perception of an image], they would be reported as multiple fixations." Therefore, it is important that the interface treat data from the eye-tracking device as a single fixation upon a given object, and not multiple fixations upon a single object.
**Visual Feedback:** Jacob (1993) describes a situation in which the visual feedback from the computer interface, controlled by data received from the eye-tracking system, may interfere with the user.

"An eye following cursor will tend to move around and thus attract the user’s attention. If there is any systematic calibration error [in the system], the cursor will be slightly offset from where the user is actually looking, causing the user’s eye to be drawn to the cursor, which will further displace the cursor, creating a positive feedback loop." (Jacob, 1993, p. 156)

Jacob (1993) suggests that one needs to be careful when designing eye movement based interfaces so as not to distract the user. Although, Jacob (1993) recommends that the use of eye-gaze tracking as a direct replacement for a mouse may be dangerous and is best avoided. However, Bates (2002, p. 4) concludes that:

"problems associated with eye-gaze interaction are not permanently inherent within the human-computer system as a whole. These problems may be reduced with experience and understanding of the flaws of the system."

Therefore, the influence of the considerations outlined by Jacob (1991; 1993) are largely dependent upon the design of the eye movement enabled interface and the interpretation of the information supplied by the eye-gaze tracking device rather the nature of human eye movements and current eye-gaze tracking techniques.
APPENDIX E  DETERMINING IMAGE TRANSFER SPEEDS

Assumptions:

1 byte = 8 bits

USB 1.1 = 12 Mbytes/s
          = 12 Mbytes/s * 8 bits
          = 96 Mbits/s

FireWire 400 = 50 Mbytes/s
               = 50 Mbytes/s * 8 bits
               = 400 Mbits/s

Image resolution = 640 x 480 pixels

Colour depth = 8 bit per primary colour (red, green and blue)

Image size = (640 * 480) * 8 bit * 3
            = 7,372,800 bits per 640 x 480 pixel image

Rate of Image Transfer:

For the above image definition, the two transfer technologies give the following transfer rates. These rates are effectively raw transfer rates and do not take into account general processing or protocol overheads.

USB 1.1:
Assuming for the moment the raw transfer speed of USB 1.1

\[ T_{\text{transfer}} = \frac{7,372,800 \text{ bits-per-image}}{96,000,000 \text{ bits-per-second}} \]

\[ = 76.80 \text{ milliseconds-per-image} \]

This equates to approximately 13 images per second. Less with protocol overheads.

FireWire 400:

\[ T_{\text{transfer}} = \frac{7,372,800 \text{ bits-per-image}}{400,000,000 \text{ bits-per-second}} \]

\[ = 18.43 \text{ milliseconds-per-image} \]

This equates to approximately 54 images per second. Less with protocol overheads.
APPENDIX F  ORIGINAL CODE LISTING FOR DETERMINING
THE CURRENT STATE OF USER GAZE

FILE:  FIXFUNC.C
Program Name: Eye Fixation Analysis Functions

Company: LC Technologies, Inc.
9455 Silver King Court
Fairfax, VA 22031
(703) 385-7133

Makers of the Eyegaze System, additional information about LC
Technologies and its products may be found at http://www.eyegaze.com

/*================================================================*/
/* FUNCTION, VARIABLE AND CONSTANT DEFINITIONS: */
/*================================================================*/
#include
#include
#include
#include
#include
#include
#define RING_SIZE 31 /* length of the delay line in
/* DetectFixation() -- */
/* should be greater than */
/* minimum_fix_samples */
/*================================================================*/
/* FUNCTION PROTOTYPES */

SVOID ResetPresFixation(void);
SVOID ResetNewFixation(void);
SVOID StartPresFixAtGazepoint(float x_gaze, float y_gaze);
SVOID StartNewFixAtGazepoint(float x_gaze, float y_gaze);
SVOID UpdatePresFixation(float x_gaze, float y_gaze, int minimum_fix_samples);
SVOID UpdateNewFixation(float x_gaze, float y_gaze);
SVOID CalcGazeDeviationFromPresFix(float x_gaze, float y_gaze);
SVOID CalcGazeDeviationFromNewFix(float x_gaze, float y_gaze);
SVOID CheckIfFixating(int minimum_fix_samples);
SVOID MoveNewFixToPresFix(int minimum_fix_samples);
SVOID DeclareCompletedFixation(int minimum_fix_samples);
SVOID RestoreOutPoints(void);
/*================================================================*/
/* GLOBAL FIXFUNC VARIABLES */

SLONG lCallCount; /* number of times this function has been */
/* called since it was initialized */
/* 30ths or 60ths of a second, */
/* (depending on eyetracking sample rate) */

SINT iNoEyeFound; /* number of successive samples with no */
/* eye found */

/*================================================================*/
/* DATA ON PREVIOUS FIXATION */

SLONG lPrevFixEndCount; /* count that the previous fixation ended */

/*================================================================*/
/* DATA ON PRESENT FIXATION */

SLONG lPresFixStartCount; /* call count that the fixation starts */
SLONG lPresFixEndCount; /* call count that the fixation ends */
SINT nPresFixSamples; /* number of samples in the present fix */
SFLOAT fXPresFixSum; /* summations for calculation of average */
SFLOAT fYPresFixSum; /* fixation position */
SFLOAT fXPresFix; /* average coordinate of the eye fixation */
SFLOAT fYPresFix; /* point (user selected units) */
SINT nPresOut; /* number of samples outside the fixation */
SFLOAT fPresDr; /* difference between gazepoint and */
/* fixation (x, y, and radius) */
/* DATA ON NEW FIXATION */
SLONG new_fix_start_count;  /* call count that the new fixation starts */
SLONG new_fix_end_count;   /* call count that the new fixation ends */
SINT n_new_fix_samples;   /* number of samples in the fixation */
SFLOAT x_new_fix_sum;   /* summations for the FIR filter */
SFLOAT y_new_fix_sum;   /* calculations of the eye motion */
SFLOAT x_new_fix;      /* average coordinate of the eye fixation */
SFLOAT y_new_fix;      /* point (user selected units) */
SFLOAT new_dr;        /* difference between gazepoint and */
                        /* fixation (x, y, and radius) */

/* call count that the new fixation starts */
/* call count that the new fixation ends */
/* number of samples in the fixation */
/* summations for the FIR filter */
/* calculations of the eye motion */
/* average coordinate of the eye fixation */
/* point (user selected units) */
/* difference between gazepoint and */
/* fixation (x, y, and radius) */

/* RING BUFFERS STORING PAST VALUES */
SFLOAT x_gaze_ring[RING_SIZE];
SFLOAT y_gaze_ring[RING_SIZE];
SBYTE gaze_found_ring[RING_SIZE];
SINT eye_motion_state[RING_SIZE];
                        /* state of the eye motion: */
                        /* MOVING */
                        /* FIXATING */
                        /* FIXATION.Completed */
SFLOAT x_fix_ring[RING_SIZE];
SFLOAT y_fix_ring[RING_SIZE];
SFLOAT gaze_deviation_ring[RING_SIZE];
SINT sac_duration_ring[RING_SIZE];
SINT fix_duration_ring[RING_SIZE];
SINT iRingIndex;    /* ring index of the present gaze sample */
SINT iRingIndexDelay; /* ring index of the gaze sample taken */
                        /* minimum_fix_samples ago */

****************************************************************************/
void InitFixation(int minimum_fix_samples)
                        /* minimum number of gaze samples */
                        /* that can be considered a */
                        /* fixation */
                        /* Note: if the input value is */
                        /* is less than 3, the function */
                        /* sets it to 3 */
                        /* This function clears any previous, present and new fixations, and it */
                        /* initializes DetectFixation()'s internal ring buffers of prior */
                        /* gazepoint data. InitFixation() should be called prior to a sequence */
                        /* of calls to DetectFixation(). */
                        /* Initialize the internal ring buffer. */
                        /* Set the call count to zero, and initialize the previous fixation end */
                        /* count so the first saccade duration is a legitimate count. */
                        /* lCallCount = 0; */
                        /* lPrevFixEndCount = 0; */
                        /* Reset the present fixation data. */
                        /* ResetPresFixation(); */
                        /* Reset the new fixation data. */
                        /* ResetNewFixation(); */
                        /* Initialize the number of successive samples with no eye found. */
                        /* iNoEyeFound = 0; */
                        /* Initialize the number of successive samples with no eye found. */
int DetectFixation(
    BYTE gazepoint_found, /* flag indicating whether or not */
    /* the image processing algo */
    /* detected the eye and computed */
    /* a valid gazepoint (TRUE/FALSE) */
    float x_gaze, /* present gazepoint */
    float y_gaze, /* (user specified units) */
    float gaze_deviation_threshold, /* distance that a gazepoint may */
    /* vary from the average fixation */
    /* point and still be considered */
    /* part of the fixation */
    /* (user specified units) */
    int minimum_fix_samples, /* minimum number of gaze samples */
    /* that can be considered a */
    /* fixation */
    /* Note: if the input value is */
    /* is less than 3, the function */
    /* sets it to 3 */
    /* OUTPUT PARAMETERS: */
    /* Delayed Gazepoint data with */
    /* fixation annotations: */
    BYTE *ptr_gazepoint_found_delayed,
    /* sample gazepoint-found flag, */
    /* min_fix_samples ago */
    float *ptr_x_gaze_delayed, /* sample gazepoint coordinates, */
    float *ptr_y_gaze_delayed, /* min_fix_samples ago */
    float *ptr_gaze_deviation_delayed,
    /* deviation of the gaze from the */
    /* present fixation, */
    /* min_fix_samples ago */
    float *ptr_x_fix_delayed, /* fixation point as estimated */
    float *ptr_y_fix_delayed, /* min_fix_samples ago */
    int *ptr_saccade_duration_delayed,
    /* duration of the saccade */
    /* preceeding the preset fixation */
    /* (samples) */
    int *ptr_fix_duration_delayed) /* duration of the present fixation */
) /* (samples) */

/* RETURN VALUES - Eye Motion State: */
/* MOVING 0 The eye was in motion min_fix_samples ago. */
/* FIXATING 1 The eye was fixating min_fix_samples ago. */
/* FIXATION_COMPLETED 2 A completed fixation has just been detected; */
/* the fixation ended min_fix_samples ago. */

/* Include FIXFUNC.H for function prototype and above constant definitions. */

/* SUMMARY */
/* This function converts a series of uniformly-sampled (raw) gaze */
/* points into a series of variable-duration saccades and fixations. */
/* Fixation analysis may be performed in real time or after the fact. */
/* allow eye fixation analysis during real-time eyegaze data collection, */
/* the function is designed to be called once per sample. When the eye */
/* is in motion, ie during saccades, the function returns 0 (MOVING). */
/* When the eye is still, ie during fixations, the function returns 1 */
/* (FIXATING). Upon the detected completion of a fixation, the function */
/* returns 2 (FIXATION_COMPLETED) and produces: */
/* a) the time duration of the saccade between the last and present */
/* eye fixation (eyegaze samples) */
/* b) the time duration of the present, just completed fixation */
/* (eyegaze samples) */
/* c) the average x and y coordinates of the eye fixation */
/* (in user defined units of x_gaze and y_gaze) */
/* Note: Although this function is intended to work in "real time", there */
/* is a delay of minimum_fix_samples in the filter which detects the */
/* motion/fixation condition of the eye. */
/* PRINCIPLE OF OPERATION */

This function detects fixations by looking for sequences of gaze-point measurements that remain relatively constant. If a new gazepoint lies within a circular region around the running average of an on-going fixation, the fixation is extended to include the new gazepoint. (The radius of the acceptance circle is user specified by setting the value of the function argument `gaze_deviation_threshold`.)

To accommodate noisy eyegaze measurements, a gazepoint that exceeds the deviation threshold is included in an on-going fixation if the subsequent gazepoint returns to a position within the threshold.

If a gazepoint is not found, during a blink for example, a fixation is extended if a) the next legitimate gazepoint measurement falls within the acceptance circle, and b) there are less than `minimum_fix_samples` of successive missed gazepoints. Otherwise, the previous fixation is considered to end at the last good gazepoint measurement.

/* UNITS OF MEASURE */

The gaze position/direction may be expressed in any units (e.g., millimeters, pixels, or radians), but the filter threshold must be expressed in the same units.

/* INITIALIZING THE FUNCTION */

Prior to analyzing a sequence of gazepoint data, the InitFixation function should be called to clear any previous, present and new fixations and to initialize the ring buffers of prior gazepoint data.

/* PROGRAM NOTES */

For purposes of describing an ongoing sequence of fixations, fixations in this program are referred to as "previous", "present", and "new". The present fixation is the one that is going on right now, or, if a new fixation has just started, the present fixation is the one that just finished. The previous fixation is the one immediately preceding the present one, and a new fixation is the one immediately following the present one. Once the present fixation is declared to be completed, the present fixation becomes the previous one, the new fixation becomes the present one, and there is not yet a new fixation.

/* Make sure the minimum fix time is at least 3 samples. */
if (minimum_fix_samples < 3)
    minimum_fix_samples = 3;

/* Make sure the ring size is large enough to handle the delay. */
if (minimum_fix_samples >= RING_SIZE)
{
    lct_settextmode();
    printf("minimum_fix_samples %i >= RING_SIZE %i\n",
           minimum_fix_samples, RING_SIZE);
    printf("Press any key to terminate...");
    getch();
    exit(99);
}

/* Increment the call count, the ring index, and the delayed ring index. */
CallCount++;
iRingIndex++;

if (iRingIndex >= RING_SIZE)
    iRingIndex = 0;

iRingIndexDelay = iRingIndex - minimum_fix_samples;
if (iRingIndexDelay < 0)
    iRingIndexDelay += RING_SIZE;

/* Update the storage rings. */
x_gaze_ring[iRingIndex] = x_gaze;
y_gaze_ring[iRingIndex] = y_gaze;
gaze_found_ring[iRingIndex] = gazepoint_found;
/* Initially assume the eye is moving. */
/* Note: These values are updated during the processing of this and */
/* subsequent gazepoints. */
eye_motion_state[iRingIndex] = MOVING;
x_fix_ring[iRingIndex] = -0.0F;
y_fix_ring[iRingIndex] = -0.0F;
gaze_deviation_ring[iRingIndex] = -0.1F;
sac_duration_ring[iRingIndex] = 0;
fix_duration_ring[iRingIndex] = 0;

/* - - - - - - - - - - - - - Process Tracked Eye - - - - - - - - - - - - - */

/* A1 If the eye's gazepoint was successfully measured this sample, */
if (gazepoint_found == TRUE)
/* The number of successive no-tracks is zero. */
   iNOEyeFound = 0;
/* B1 If there is a present fixation, */
if (INPresFixSamples > 0)
   /* Compute the deviation of the gazepoint from the present fixation. */
   CalcGazeDeviationFromPresFix(x_gaze, y_gaze);
/* C1 If the gazepoint is within the present fixation region, */
   if (fPresDr <= gaze_deviation_threshold)
      /* Restore any previous gazepoints that were temporarily left */
      /* out of the fixation. */
      RestoreOutPoints();
      /* Update the present fixation hypothesis, and check if there */
      /* are enough samples to declare that the eye is fixating. */
      UpdatePresFixation(x_gaze, y_gaze, minimum_fix_samples);
/* C2 Otherwise, if the point is outside the present fixation region, */
else /* if (fPresDr > gaze_deviation_threshold) */
   /* Increment the number of gazepoint samples outside the */
   /* present fix. */
   INPresOut++;
/* D1 If the present fixation is finished, i.e., if there have */
/* been minimum_fix_samples since the gazepoint last matched */
/* the present fixation, and the present fixation is long */
/* enough to count as a real fixation */
/* if (((int)(lCallCount - lPresFixEndCount) >= */
/* minimum_fix_samples) && */
/* (INPresFixSamples >= minimum_fix_samples)) */
   /* Declare the present fixation to be completed, move the */
   /* present fixation to the prior, move the new fixation to */
   /* the present, and check if the new (now present) fixation */
   /* has enough points for the eye to be declared to be fixating. */
   DeclareCompletedFixation(minimum_fix_samples);
   /* Compute the deviation of the gazepoint from the now */
   /* present fixation. */
   CalcGazeDeviationFromPresFix(x_gaze, y_gaze);
/* E1 If the gazepoint is within the now present fixation region, */
if (fPresDr <= gaze_deviation_threshold)
   /* Update the present fixation data, and check if there */
   /* are enough samples to declare that the eye is fixating. */
   UpdatePresFixation(x_gaze, y_gaze, minimum_fix_samples);
E2 Otherwise, if the gazepoint is outside the now present fixation,
else /* if (fPresDr > gaze_deviation_threshold) */
{
  /*
   Start a new fixation at the gazepoint.
   StartNewFixAtGazepoint(x_gaze, y_gaze);
  */
}

D2 Otherwise, if the present fixation is not finished,
else
{
  F1 If there is a new fixation hypothesis,
  if (n_new_fix_samples > 0)
  {
    /* Compute the deviation of the gazepoint from the new fixation.
    CalcGazeDeviationFromNewFix(x_gaze, y_gaze);
    */
    G1 If the new point falls within the new fix,
    if (new_dr <= gaze_deviation_threshold)
    {
      /* Update the new fixation hypothesis.
      UpdateNewFixation(x_gaze, y_gaze);
      */
    } /*
  } /*
  H. If there are now enough points in the new fix to declare it a real fix,
  if (n_new_fix_samples == minimum_fix_samples)
  {
    /* Drop the present fixation data, move the new new fixation into the present fixation, and see if the new (now present) fixation has enough points to declare the eye to be fixating.
    MoveNewFixToPresFix(minimum_fix_samples);
    */
  } /*
  G2 Otherwise, if the point is outside the new fix,
  else /* if (new_dr <= gaze_deviation_threshold) */
  {
    /* Start the new fixation at the new gazepoint.
    StartNewFixAtGazepoint(x_gaze, y_gaze);
    */
  } /*
  F2 Otherwise, If there is not a new fix,
  else /* if (n_new_fix_counts == 0) */
  {
    /* Start the new fixation at the gazepoint.
    StartNewFixAtGazepoint(x_gaze, y_gaze);
    */
  } /*
  } /*
  B2 Otherwise, if there is not a present fixation,
  else /* if (INPresFixSamples = 0) */
  {
    /* Start the present fixation at the gazepoint and reset the new fixation.
    StartPresFixAtGazepoint(x_gaze, y_gaze);
    */
  } /*
} /*

/* A2 Otherwise, if the eye's gazepoint was not successfully measured */
/* this sample, */
else /* if (gazepoint_found == FALSE) */
{
/* Increment the number of successive samples with no eye found. */
inNoEyeFound++;

/* If it has been min-fix-samples since the last sample in the */
/* present fixation, */
if ((int) (lCallCount - lPresFixEndCount) >= minimum_fix_samples) {
/* If there had been a fixation prior to loosing track of the eye, */
if (INPresFixSamples >= minimum_fix_samples) {
/* Declare the present fixation to be completed, move the */
/* present fixation to the prior, move the new fixation to */
/* the present, and check if the new (now present) fixation */
/* has enough points for the eye to be declared to be fixating. */
DeclareCompletedFixation(minimum_fix_samples);
}
/* Reset the present fixation data. */
ResetPresFixation();
}

/-----------------------------------------------------------------------
Pass Data Back -----------------------------------------------------------*/

/* Pass the delayed gazepoint data, with the relevant saccade/fixation */
/* data, back to the calling function. */
*ptr_x_gaze_delayed = x_gaze_ring[iRingIndexDelay];
*ptr_y_gaze_delayed = y_gaze_ring[iRingIndexDelay];
*ptr_gazepoint_found_delayed = gaze_found_ring[iRingIndexDelay];
*ptr_x_fix_delayed = x_fix_ring[iRingIndexDelay];
*ptr_y_fix_delayed = y_fix_ring[iRingIndexDelay];
*ptr_gaze_deviation_delayed = gaze_deviation_ring[iRingIndexDelay];
*ptr_saccade_duration_delayed = sac_duration_ring[iRingIndexDelay];
*ptr_fix_duration_delayed = fix_duration_ring[iRingIndexDelay];

/* Return the eye motion/fixation state for the delayed point. */
return(eye_motion_state[iRingIndexDelay]);

**************************************************************************/
SVOID ResetPresFixation(void)
/* This function resets the present fixation, i.e., declares it nonexistent. */
{
    lPresFixStartCount = 0;
    lPresFixEndCount = 0;
    INPresFixSamples = 0;
    fXPResFixSum = 0.0F;
    fYPResFixSum = 0.0F;
    fXPResFix = 0.0F;
    fYPResFix = 0.0F;
    INPresOut = 0;
}

**************************************************************************/
SVOID ResetNewFixation(void)
/* This function resets the new fixation, i.e., declares it nonexistent. */
{
    new_fix_start_count = 0;
    new_fix_end_count = 0;
    n_new_fix_samples = 0;
    x_new_fix_sum = 0.0F;
    y_new_fix_sum = 0.0F;
    x_new_fix = 0.0F;
    y_new_fix = 0.0F;
}
**VOID StartPresFixAtGazepoint(float x_gaze, float y_gaze)

/* This function starts the present fixation at the argument gazepoint */
/* and makes sure there is no new fixation hypothesis. */
{

/* Start the present fixation at the argument gazepoint. */
INPresFixSamples = 1;
fXPresFixSum = x_gaze;
fYPresFixSum = y_gaze;
fXPresFix = fXPresFixSum / INPresFixSamples;
fYPresFix = fYPresFixSum / INPresFixSamples;
1PresFixStartCount = lCallCount;
1PresFixEndCount = lCallCount;
inPresOut = 0;

/* Make sure there is no new fixation. */
ResetNewFixation();
}

**************************************************************************/
**VOID StartNewFixAtGazepoint(float x_gaze, float y_gaze)

/* This function starts the new fixation at the argument gazepoint. */
{

n_new_fix_samples = 1;
x_new_fix_sum = x_gaze;
y_new_fix_sum = y_gaze;
x_new_fix = x_gaze;
y_new_fix = y_gaze;
new_fix_start_count = lCallCount;
new_fix_end_count = lCallCount;
}

**************************************************************************/
**VOID UpdatePresFixation(float x_gaze, float y_gaze, int minimum_fix_samples)

/* This function updates the present fixation with the argument gazepoint, */
/* checks if there are enough samples to declare that the eye is now */
/* fixating, and makes sure there is no hypothesis for a new fixation. */
{

/* Update the present fixation with the argument gazepoint. */
inPresFixSamples++;
fXPresFixSum += x_gaze;
fYPresFixSum += y_gaze;
fXPresFix = fXPresFixSum / inPresFixSamples;
fYPresFix = fYPresFixSum / inPresFixSamples;
inPresFixedCount = lCallCount;
inPresOut = 0;

/* Check if there are enough samples in the present fixation hypothesis */
/* to declare that the eye is fixating. */
CheckIfFixating(minimum_fix_samples);

/* There is no hypothesis for a new fixation. */
ResetNewFixation();
}

**************************************************************************/
**VOID UpdateNewFixation(float x_gaze, float y_gaze)

/* This function updates the new fixation with the argument gazepoint. */
{

/* Update the new fixation with the argument gazepoint. */
n_new_fix_samples++;
x_new_fix_sum += x_gaze;
y_new_fix_sum += y_gaze;
x_new_fix = x_new_fix_sum / n_new_fix_samples;
y_new_fix = y_new_fix_sum / n_new_fix_samples;
new_fix_end_count = lCallCount;
}
SVOID CalcGazeDeviationFromPresFix(float x_gaze, float y_gaze)
/* This function calculates the deviation of the gazepoint from the */
/* present fixation location. */
{
    float dx, dy;        /* horizontal and vertical deviations */
    dx = x_gaze - fXPresFix;
    dy = y_gaze - fYPresFix;
    fPresDr = (float)sqrt(dx * dx + dy * dy);
    /* Put the deviation in the ring buffer for future reference. */
    gaze_deviation_ring[iRingIndex] = fPresDr;
}

SVOID CalcGazeDeviationFromNewFix(float x_gaze, float y_gaze)
/* This function calculate the deviation of the gazepoint from the new */
/* fixation location. */
{
    float dx, dy;        /* horizontal and vertical deviations */
    dx = x_gaze - x_new_fix;
    dy = y_gaze - y_new_fix;
    new_dr = (float)sqrt(dx * dx + dy * dy);
}

SVOID CheckIfFixating(int minimum_fix_samples)
/* This function checks to see whether there are enough samples in the */
/* presently hypothesized fixation to declare that the eye is fixating */
/* yet, and if there is a true fixation going on, it updates the ring */
/* buffers to reflect the fixation. */
{
    int i, ii;            /* dummy ring indices */
    /* If there are enough samples for a fixation, */
    if (iNPresFixSamples >= minimum_fix_samples)
    {
        /* Declare the eye to be fixating. Go back through the last */
        /* minimum_fix_samples entries of the ring buffer making sure that all */
        /* samples from the present fixation are marked as fixating, and set */
        /* the entries with the newest estimate of the fixation location. */
        for (i = 0; i < minimum_fix_samples; i++)
        {
            ii = iRingIndex - i;
            if (ii < 0)
                ii += RING_SIZE;
            eye_motion_state[ii] = FIXATING;
            x_fix_ring[ii] = fXPresFix;
            y_fix_ring[ii] = fYPresFix;
            sec_duration_ring[ii] = (int)(lPresFixStartCount - lPrevFixEndCount - 1);
            fix_duration_ring[ii] = (int)(lPresFixEndCount - lPresFixStartCount + 1 - i);
        }
    }
}
/* Move the new fixation to the present fixation. */
inPresFixSamples = n_new_fix_samples;
fXPresFixSum = x_new_fix_sum;
fYPresFixSum = y_new_fix_sum;
fXPresFix = x_new_fix;
fYPresFix = y_new_fix;
lPresFixStartCount = new_fix_start_count;
lPresFixEndCount = new_fix_end_count;
inPresOut = 0;

/* Reset the new fixation. */
ResetNewFixation();

/* Check if there are enough samples in the new (now present) fixation to */
/* declare that the eye is fixating. */
CheckIfFixating(minimum_fix_samples);
}

/*******************************************************************************/
SVOID DeclareCompletedFixation(int minimum_fix_samples)

/* This function: */
/* a) declares the present fixation to be completed, */
/* b) moves the present fixation to the prior fixation, */
/* c) moves the new fixation, if any, to the present fixation, and */

{ /* Declare the present fixation to be completed. */
  eye_motion_state[IRingIndexDelay] = FIXATION_COMPLETED;

  /* Move the present fixation to the previous fixation. This saves the */
  /* end time of the present fixation for later computation of the saccade */
  /* period between this and the next fixation. */
  lPrevFixEndCount = lPresFixEndCount;

  /* Move the new fixation data, if any, to the present fixation, reset */
  /* the new fixation, and check if there are enough samples in the new */
  /* (now present) fixation to declare that the eye is fixating. */
  MoveNewFixToPresFix(minimum_fix_samples);
}

/*******************************************************************************/
SVOID RestoreOutPoints(void)

/* This function restores any previous gazepoints that were left out of */
/* the fixation and are now known to be part of the present fixation. */

{ /* dummy ring indices */
  int i, ii;

  /* If there were some previous points that temporarily went out of the */
  /* fixation region, */
  if (inPresOut > 0)
  { /* Undo the hypothesis that they were outside the fixation and declare */
    /* them now to be part of the fix. */
    for (i = 1; i <= inPresOut; i++)
    { ii = iRingIndex - i;
      if (ii < 0)
        ii += RING_SIZE;

        /* Declare the new fixation. */
        ResetNewFixation();

        /* Check if there are enough samples in the new (now present) fixation to */
        /* declare that the eye is fixating. */
        CheckIfFixating(minimum_fix_samples);
    } /* Declare the present fixation to be completed. */
    eye_motion_state[IRingIndexDelay] = FIXATION_COMPLETED;

    /* Move the present fixation to the previous fixation. This saves the */
    /* end time of the present fixation for later computation of the saccade */
    /* period between this and the next fixation. */
    lPrevFixEndCount = lPresFixEndCount;

    /* Move the new fixation data, if any, to the present fixation, reset */
    /* the new fixation, and check if there are enough samples in the new */
    /* (now present) fixation to declare that the eye is fixating. */
    MoveNewFixToPresFix(minimum_fix_samples);
}

/*******************************************************************************/
SVOID DeclareCompletedFixation(int minimum_fix_samples)

/* This function: */
/* a) declares the present fixation to be completed, */
/* b) moves the present fixation to the prior fixation, */
/* c) moves the new fixation, if any, to the present fixation, and */

{ /* Declare the present fixation to be completed. */
  eye_motion_state[IRingIndexDelay] = FIXATION_COMPLETED;

  /* Move the present fixation to the previous fixation. This saves the */
  /* end time of the present fixation for later computation of the saccade */
  /* period between this and the next fixation. */
  lPrevFixEndCount = lPresFixEndCount;

  /* Move the new fixation data, if any, to the present fixation, reset */
  /* the new fixation, and check if there are enough samples in the new */
  /* (now present) fixation to declare that the eye is fixating. */
  MoveNewFixToPresFix(minimum_fix_samples);
}

/*******************************************************************************/
SVOID RestoreOutPoints(void)

/* This function restores any previous gazepoints that were left out of */
/* the fixation and are now known to be part of the present fixation. */

{ /* dummy ring indices */
  int i, ii;

  /* If there were some previous points that temporarily went out of the */
  /* fixation region, */
  if (inPresOut > 0)
  { /* Undo the hypothesis that they were outside the fixation and declare */
    /* them now to be part of the fix. */
    for (i = 1; i <= inPresOut; i++)
    { ii = iRingIndex - i;
      if (ii < 0)
        ii += RING_SIZE;

        /* Declare the new fixation. */
        ResetNewFixation();

        /* Check if there are enough samples in the new (now present) fixation to */
        /* declare that the eye is fixating. */
        CheckIfFixating(minimum_fix_samples);
    } /* Declare the present fixation to be completed. */
    eye_motion_state[IRingIndexDelay] = FIXATION_COMPLETED;

    /* Move the present fixation to the previous fixation. This saves the */
    /* end time of the present fixation for later computation of the saccade */
    /* period between this and the next fixation. */
    lPrevFixEndCount = lPresFixEndCount;

    /* Move the new fixation data, if any, to the present fixation, reset */
    /* the new fixation, and check if there are enough samples in the new */
    /* (now present) fixation to declare that the eye is fixating. */
    MoveNewFixToPresFix(minimum_fix_samples);
}
```c
if (gaze_found_ring[ii] == TRUE)
{
    INPresFixSamples++;
    fXPresFixSum += x_gaze_ring[ii];
    fYPresFixSum += y_gaze_ring[ii];
    eye_motion_state[ii] = FIXATING;
}

/* Set the number of "out" points to be zero. */
INPresOut = 0;
```

Figure F.1: Public domain source code to detect the current state of user gaze as provided by LC Technologies Inc (2002).
APPENDIX G  VIDEO-BASED EYE-GAZE TRACKING AND TEST SYSTEM CONFIGURATION SCRIPTS

/*~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~**
** REAL-TIME NON-INTRUSIVE EYE-GAZE TRACKER CONFIGURATION FILE **
** ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~**
** AUTHOR: Joanne Church **
** DATE: 20/02/2004 **
** VERSION: 1.0 **
** PURPOSE: To configure the VBEGT system for operational testing **
** DESCRIPTION: This file stores the values passed to the VBEGT system upon **
** initialisation. **
** NOTES: None **
** ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~*/

/*~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~**
** EYE TRACKER CONTROL VALUES **
** ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~**
** FORMAT: The values inserted between the [EYE_TRACKER_START] and [EYE_TRACKER_END] **
** markers are used to configure the following attributes of the video-based eye-gaze **
** tracking system. These attributes must be assigned a value and appear in the order **
** specified herein. **
** ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~**
** Image Threshold Value: **
** Fixation Number: **
** Fixation Resolution: **
** Line Segment Deviation: **
** Line Segment Pixel Threshold: **
** Minimum Line Segment Length: **
** Lines To Skip: **
** Eye Data Array Size: **
** Left Screen Position: **
** Right Screen Position: **
** ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~**

Used to determine whether a given pixel is termed "black" (part of the current line segment) or "white" (not a part of the current Line Segment).

The number of gaze points within a specified vicinity that denote whether a user is fixating upon a given point or the eye is still in transit from one point of fixation to another.

The margin that determines whether consecutive gazes are centred about the same point i.e., used to determine whether the user is fixating upon a given point.

The deviation between two Line Segment's start and end positions. Used when matching line segments to hypotheses.

Represents the maximum number of pixels that may be missed before a break in that Line Segment occurs.

The minimum size of a line segment retrieved from a given image row.

The number of rows in the image to skip during processing. For example, setting this value to 2 would process every second line of the image.

The size of the array storing recent eye movement. This array is used when determining the current state of the eye i.e., moving (saccade) or focused (fixation)

The left most screen position, in pixels, marking the edge of the user's screen.

The right most screen position, in pixels, marking the edge of the user's screen.
** Top Screen Position: ** The top most screen position, in pixels, marking the edge of the user's screen.

** Bottom Screen Position: ** The bottom most screen position, in pixels, marking the edge of the user's screen.

** NOTES: **

** None **

**=======================================================================================================*/

[EYE_TRACKER_START]
40 // Image threshold value
3 // Fixation number
5 // Fixation resolution
18 // Line Segment deviation
8 // Line Segment pixel threshold
4 // Minimum Line Segment length
2 // Number of lines to skip
10 // Eye data array size
0 // Left-most position of the active user's screen
0 // Top-most position of the active user's screen
640 // Right-most position of the active user's screen
480 // Bottom-most position of the active user's screen

[EYE_TRACKER_END]

Figure G.1: VBEGT Configuration Script showing example configuration and descriptions for each configurable item.
**REAL-TIME NON-INTRUSIVE EYE-GAZE TRACKER CAMERA CONFIGURATION FILE**

**AUTHOR:** Joanne Church

**DATE:** 20/02/2004

**VERSION:** 1.0

**PURPOSE:** To configure the camera used by the VBEGT system for operational testing

**DESCRIPTION:** This file stores the values passed to the VBEGT system camera upon initialisation. These values are specific to the PixeLINK Colour Mega Pixel FireWire 400 Camera model number PL-A642.

**NOTES:** None

**====================================================================================**

**CAMERA CONTROL VALUES**

**FORMAT:** The values inserted between the [CAMERA_START] and [CAMERA_END] markers are used to configure the environment variables used be the camera chosen for VBEGT. These variables are described and ordered as follows. Values must be provided for each environment variable.

**Maximum Exposure Time:**

The maximum exposure time that may be applied to an image before reducing the camera's frame rate to a value below the minimum acceptable frame rate specified for this camera.

**Decimation:**

The decimation value (1, 2, or 4) used by the camera during image capture. This value affects the size of the final image captured (see description for sub-window width and height).

**Data Transfer Size:**

The size of the pixel information returned by the camera. The data transfer size selected (8 or 16 bit) affects the clock rate selected (see description for clock rate). 8 bpp (256 shades of red, green and blue) is the default for this project. 16 bpp is not supported.

**Clock Rate:**

The clock frequency set for the imaging device (24, 16, 12, 8, 6 or 4 MHz). When the image transfer size is set to 16 bits a clock frequency of 24 MHz will crash the system.

**Sub-window Width:**

The width of the sub-window in pixels for the captured image. Note this value does not denote the width of the final image as image width is defined as: image width = sub-window width / decimation value.

**Sub-window Height:**

The height of the sub-window in pixels used for image capture. Note this value does not denote the final height of the captured image, as: image height = sub-window height / decimation value.

**Sub-window Left Position:**

The position of the sub-window within the camera’s field of view (1280 x 1024 pixels) in pixels from the left side of the imager (camera) window.

**Sub-window Top Position:**

The position of the sub window, in pixels, from the top of the imager (camera) window.

**Capture Mode:**

Defines the camera mode used to capture images. This is either video (CAPTURE_MODE_VIDEO_MODE) or still mode (CAPTURE_MODE_STILL_MODE).
**Colour Mode:**
The colour conversion applied to the image before processing i.e., the image is converted to an RGB24 colour image (IMAGE_COLOUR), RGB24 greyscale image (IMAGE_GREYSCALE) or RGB24 black and white image (IMAGE_BLACK_AND_WHITE).

**NOTES:**

**The values for the sub-window width, height, left position and top position must abide by the following:**

1. The left position of the sub-window + the width of the sub-window must be less than the maximum horizontal resolution of the camera i.e., 1280 pixels.
2. The top position of the sub-window + the height of the sub-window must be less than the maximum vertical resolution of the camera i.e., 1024 pixels.

When entering clock rates use the following values for the corresponding clock rate as defined in the PixelLINK camera API:

- 24 MHz = 0x80 = 128
- 16 MHz = 0x00 = 0
- 12 MHz = 0x81 = 129
- 8 MHz = 0x01 = 1
- 6 MHz = 0x82 = 130
- 4 MHz = 0x02 = 2

```
[CAMERA_START]
50 // Maximum Exposure Time
1 // Decimation (1, 2, or 4)
8 // Data Transfer Size (8 or 16)
128 // Clock Rate (24, 16, 12, 8, 6 or 4 MHz)
136 // Sub-window Width (columns)
450 // Sub-window Height (rows)
400 // Sub-window Left Position
640 // Sub-window Top Position
CAPTURE_MODE_STILL_MODE // Capture Mode (CAPTURE_MODE_[STILL|VIDEO]_MODE)
IMAGE_GREYSCALE // Colour Mode (IMAGE_[COLOUR|GREYSCALE])
BLACK_AND_WHITE]
[CAMERA_END]
```

Figure G.2: Camera Configuration Script showing example configuration and descriptions for each configurable item.
REAL-TIME NON-INTRUSIVE EYE-GAZE TRACKER USER CONFIGURATION FILE

AUTHOR: Joanne Church
DATE: 20/02/2004
VERSION: 1.0
PURPOSE: To configure the VBEGT system for particular user.
DESCRIPTION: This file stores the values passed to the VBEGT system upon initialisation. These values relate to the characteristics of the user.
NOTES: None

USER CONTROL VALUES

FORMAT: The values inserted between [USER_DETAILS_START] and [USER_DETAILS_END] are used to configure the environment variables used for the current VBEGT system user. These variables are described and ordered as follows. Values must be provided for each environment variable.

Maximum Iris Radius: The maximum size for a circle (in pixels) that may be recommended to the VBEGT system as the user’s iris.
Minimum Iris Radius: The minimum size (in pixels) for a circle that may be recommended to the VBEGT system as the user’s iris.
Top Left Calibration Top: The top most point of a rectangle encasing the user’s iris. The coordinates of this position are relative to the position of the iris in a given eye image when the user is looking at the top left position on a given calibration grid.
Top Left Calibration Bottom: The bottom most point of a rectangle encasing the user’s iris. The coordinates of this position are relative to the position of the iris in a given eye image when the user is looking at the top left position on a given calibration grid.
Top Left Calibration Left: The left most point of a rectangle encasing the user’s iris. The coordinates of this position are relative to the position of the iris in a given eye image when the user is looking at the top left position on a given calibration grid.
Top Left Calibration Right: The right most point of a rectangle encasing the user’s iris. The coordinates of this position are relative to the position of the iris in a given eye image when the user is looking at the top left position on a given calibration grid.
Bottom Right Calibration Top: The top most point of a rectangle encasing the user’s iris. The coordinates of this position are relative to the position of the iris in a given eye image when the user is looking at the bottom right position on a given calibration grid.
Bottom Right Calibration Bottom: The bottom most point of a rectangle encasing the user’s iris. The coordinates of this position are relative to the position of the iris in a given eye image when the user is looking at the bottom right position on a given calibration grid.
** Bottom Right Calibration Left:** The left most point of a rectangle encasing the user's iris. The coordinates of this position are relative to the position of the iris in a given eye image when the user is looking at the bottom right position on a given calibration grid.

** Bottom Right Calibration Right:** The right most point of a rectangle encasing the user's iris. The coordinates of this position are relative to the position of the iris in a given eye image when the user is looking at the bottom right position on a given calibration grid.

** Centre Calibration Top:** The top most point of a rectangle encasing the user's iris. The coordinates of this position are relative to the position of the iris in a given eye image when the user is looking at the centre position on a given calibration grid.

** Centre Calibration Bottom:** The bottom most point of a rectangle encasing the user's iris. The coordinates of this position are relative to the position of the iris in a given eye image when the user is looking at the centre position on a given calibration grid.

** Centre Calibration Left:** The left most point of a rectangle encasing the user's iris. The coordinates of this position are relative to the position of the iris in a given eye image when the user is looking at the centre position on a given calibration grid.

** Centre Calibration Right:** The right most point of a rectangle encasing the user's iris. The coordinates of this position are relative to the position of the iris in a given eye image when the user is looking at the centre position on a given calibration grid.

** NOTES:**

** None

```
[USER_DETAILS_START]
126  // Max Pupil Radius (Pixels)
114  // Min Pupil Radius (Pixels)
66   // Top Left Calibration Top
98   // Top Left Calibration Bottom
46   // Top Left Calibration Left
80   // Top Left Calibration Right
26   // Bottom Right Calibration Top
52   // Bottom Right Calibration Bottom
65   // Bottom Right Calibration Left
100  // Bottom Right Calibration Right
44   // Centre Calibration Top
74   // Centre Calibration Bottom
55   // Centre Calibration Left
89   // Centre Calibration Right
[USERDETAILS_END]
```
USER CONTROL VALUES

** FORMAT:** The values inserted between [CYCLOPS_CONFIG_START] and [CYCLOPS_CONFIG_END] are used to configure the environment variables used for the Mechanical User Model. These variables are described and ordered as follows.

** Values must be provided for each environment variable

**

- **Active Screen Width:** The width of the user's screen in pixels. This is equal to the screen resolution. For example, if the screen resolution is 640 x 480, the active screen width is 640.

- **Active Screen Height:** The height of the user's screen in pixels. This is equal to the screen resolution. For example, if the screen resolution is 640 x 480, the active screen height is 480.

- **X Screen Divisions:** The number of horizontal divisions a screen may be divided into for user input.

- **Y Screen Divisions:** The number of vertical divisions a screen may be divided into for user input.

- **Distance From The Screen:** The distance between the Eye Model and the computer monitor in millimetres.

- **Physical Screen Width:** The width of the computer screen in millimetres.

- **Physical Screen Height:** The height of the computer screen in millimetres.

- **Starting Division:** The division number upon which the Eye Model gazes at the beginning of each test.

- **Eye X Step Range:** The maximum number of horizontal steps that the Eye Model may travel from left to right.

- **Eye Y Step Range:** The maximum number of vertical steps that the Eye Model may travel from top to bottom.

- **Eye X Degrees Per Step:** The number of horizontal degrees travelled per step.

- **Eye Y Degrees Per Step:** The number of vertical degrees travelled per step.

- **Head X Step Range:** The maximum number of horizontal steps that the Head Model may travel from left to right.

- **Head Y Step Range:** The maximum number of vertical steps that the Head Model may travel from top to bottom.

- **Head Z Step Range:** The maximum number of steps that the Head Model may travel back and forth.

- **Head X Distance Per Step:** The number of horizontal degrees travelled per step.
** Head Y Distance Per Step: The number of vertical degrees travelled per step.
**
** Head Z Distance Per Step: The number of degrees travelled per step.
**
** NOTES:
**
** None
**
====================================================================================

[CYCLOPS_CONFIG_START]
800 // Active Screen Width
500 // Active Screen Height
1 // X Screen Divisions
2 // Y Screen Divisions
550 // Distance From The Screen
315 // Physical Screen Width
240 // Physical Screen Height
0 // Starting Division
131 // Eye X Step Range
78 // Eye Y Step Range
0.9 // Eye X Degrees Per Step
0.9 // Eye Y Degrees Per Step
23510 // Head X Step Range
13367 // Head Y Step Range
14557 // Head Z Step Range
0.01 // Head X Distance Per Step
0.005 // Head Y Distance Per Step
0.01 // Head Z Distance Per Step
[CYCLOPS_CONFIG_END]

Figure G.4: Mechanical User Model Configuration Script showing example configuration and descriptions for each configurable item.
Figure G.5: Movement Script Generator Configuration Script showing example configuration and descriptions for each configurable item.
APPENDIX H  SAMPLE IMAGES USED FOR ALGORITHM TESTING

For the purpose of initial VBEGT algorithm testing, the SCIS Research Support department of Edith Cowan University supplied a series of test images to use when determining the accuracy and repeatability of the IHM algorithm. These images are based upon a series of digital images of a user (the researcher) looking at the calibration grid pictured in Figure H. 1, supplied during the conception of this study (pre-research proposal).

![Figure H. 1: Calibration grid used to direct user eye movements for IHM algorithm testing.](image-url)

Test images were generated from the initial images supplied using the following procedure provided by Wild (2004). Images resulting from this process are located in Table H. 1

1. measure the diameter of the iris;

2. using MS Paint, draw a circle of that diameter over the iris of each image (see Figure H. 2). The circle attempts to ‘contain’ the majority
Table H.1: Test images supplied by SCIS Research Support to facilitate IHM algorithm testing (Wild, 2004).

<table>
<thead>
<tr>
<th>Calibration Point</th>
<th>Eye Image</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td><img src="image" alt="Calibration Point A" /></td>
</tr>
<tr>
<td>B</td>
<td><img src="image" alt="Calibration Point B" /></td>
</tr>
<tr>
<td>C</td>
<td><img src="image" alt="Calibration Point C" /></td>
</tr>
<tr>
<td>D</td>
<td><img src="image" alt="Calibration Point D" /></td>
</tr>
<tr>
<td>E</td>
<td><img src="image" alt="Calibration Point E" /></td>
</tr>
<tr>
<td>Calibration Point</td>
<td>Eye Image</td>
</tr>
<tr>
<td>-------------------</td>
<td>-----------</td>
</tr>
<tr>
<td>F</td>
<td><img src="image" alt="Eye Image F" /></td>
</tr>
<tr>
<td>G</td>
<td><img src="image" alt="Eye Image G" /></td>
</tr>
<tr>
<td>H</td>
<td><img src="image" alt="Eye Image H" /></td>
</tr>
<tr>
<td>I</td>
<td><img src="image" alt="Eye Image I" /></td>
</tr>
<tr>
<td>J</td>
<td><img src="image" alt="Eye Image J" /></td>
</tr>
<tr>
<td>K</td>
<td><img src="image" alt="Eye Image K" /></td>
</tr>
<tr>
<td>Calibration Point</td>
<td>Eye Image</td>
</tr>
<tr>
<td>-------------------</td>
<td>----------</td>
</tr>
<tr>
<td>L</td>
<td><img src="image" alt="Eye Image L" /></td>
</tr>
<tr>
<td>M</td>
<td><img src="image" alt="Eye Image M" /></td>
</tr>
<tr>
<td>N</td>
<td><img src="image" alt="Eye Image N" /></td>
</tr>
<tr>
<td>O</td>
<td><img src="image" alt="Eye Image O" /></td>
</tr>
</tbody>
</table>
Table H. 2: X, Y coordinates marking the position of the user’s iris within the test images supplied by SCIS Research Support (Wild, 2004).

<table>
<thead>
<tr>
<th>Calibration Point</th>
<th>Left Circle X-Axis (pixels)</th>
<th>Bottom Circle Y-Axis (pixels)</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>225</td>
<td>396</td>
</tr>
<tr>
<td>B</td>
<td>162</td>
<td>374</td>
</tr>
<tr>
<td>C</td>
<td>170</td>
<td>404</td>
</tr>
<tr>
<td>D</td>
<td>255</td>
<td>387</td>
</tr>
<tr>
<td>E</td>
<td>250</td>
<td>327</td>
</tr>
<tr>
<td>F</td>
<td>226</td>
<td>254</td>
</tr>
<tr>
<td>G</td>
<td>235</td>
<td>247</td>
</tr>
<tr>
<td>H</td>
<td>312</td>
<td>242</td>
</tr>
<tr>
<td>I</td>
<td>310</td>
<td>205</td>
</tr>
<tr>
<td>J</td>
<td>270</td>
<td>190</td>
</tr>
<tr>
<td>K</td>
<td>275</td>
<td>200</td>
</tr>
<tr>
<td>L</td>
<td>319</td>
<td>190</td>
</tr>
<tr>
<td>M</td>
<td>314</td>
<td>169</td>
</tr>
<tr>
<td>N</td>
<td>268</td>
<td>185</td>
</tr>
<tr>
<td>O</td>
<td>287</td>
<td>175</td>
</tr>
</tbody>
</table>

The locations for the iris positions recorded in Table H. 2 required adjustment to compensate for the values recorded using MS Paint. MS Paint considers the first row of a given image to be row number one. This study, considers the first row of a given image to be row number zero, in keeping with the row numbering supported by the web camera selected for this study. Therefore, the values stored in Table H. 2 needs to be adjusted so they appear as shown in Table H. 3.

---

Table H.3: Adjusted X, Y coordinates marking the position of the user’s iris within the test images supplied by SCIS Research Support (Wild, 2004).

<table>
<thead>
<tr>
<th>Calibration Point</th>
<th>Left Circle X-Axis (pixels)</th>
<th>Bottom Circle Y-Axis (pixels)</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>225</td>
<td>395</td>
</tr>
<tr>
<td>B</td>
<td>162</td>
<td>373</td>
</tr>
<tr>
<td>C</td>
<td>170</td>
<td>403</td>
</tr>
<tr>
<td>D</td>
<td>255</td>
<td>386</td>
</tr>
<tr>
<td>E</td>
<td>250</td>
<td>326</td>
</tr>
<tr>
<td>F</td>
<td>226</td>
<td>253</td>
</tr>
<tr>
<td>G</td>
<td>235</td>
<td>246</td>
</tr>
<tr>
<td>H</td>
<td>312</td>
<td>241</td>
</tr>
<tr>
<td>I</td>
<td>310</td>
<td>204</td>
</tr>
<tr>
<td>J</td>
<td>270</td>
<td>187</td>
</tr>
<tr>
<td>K</td>
<td>275</td>
<td>199</td>
</tr>
<tr>
<td>L</td>
<td>319</td>
<td>189</td>
</tr>
<tr>
<td>M</td>
<td>314</td>
<td>168</td>
</tr>
<tr>
<td>N</td>
<td>268</td>
<td>184</td>
</tr>
<tr>
<td>O</td>
<td>287</td>
<td>174</td>
</tr>
</tbody>
</table>
APPENDIX I PROFILING VIDEO-BASED EYE-GAZE TRACKING SYSTEMS

VBEGT, like other devices used for HCI, are response-time critical (Quek, 1995). Time lags and processing latency may render the system frustratingly clumsy and seemingly unusable to mainstream computer users (Quek, 1995). Therefore, it is essential that the system is able to compute the user’s eye position in real-time (Mulligan, 1997a). In order to provide the necessary information regarding the VBEGT system’s performance, this study required the development of a customised profiling tool.

The profiling tool developed was based upon the instrumentation library supplied by Spuler (1992). This library keeps track of the processing time taken by code blocks using “clocks”. These clocks are referenced by a character string name and are started and stopped by the functions start_clock() and stop_clock(). At the end of the program the clock_report() function may be used to generate a summary of the times used by each clock and the percentage of total run time consumed by the clock. Figure I.1 and Figure I.2 show the code listing for the header and source files provided by Spuler (1992).

```c
/*----------------------------------------~--------------------------------*/
/* INSTRUMENT.H: Header file for instrumentation library */
/*----------------------------------------~--------------------------------*/
#ifndef _INSTRUMENTHINCLUDED
#define _INSTRUMENTHINCLUDED

void start_clock ( char * name ); /* Start the clock */
void stop_clock ( char * name ); /* Stop the clock*/
void clock_report ( void ); /* Display the times for each clock */

#endif /* _INSTRUMENTHINCLUDED */
```

Figure I.1: Header file showing the public functions made available by the instrumentation library provided by Spuler (1992).

63 The process of adding timings instructions to a program to examine its efficiency is called instrumenting the program (Spuler, 1992).
/* INSTRUMENT.C: Source file for instrumentation library */
#include <stdio.h>
#include <time.h>
#include <string.h>
#include "instrument.h" /* include the interface header file */

typedef int bool
#define TRUE 1
#define FALSE 0

/* Structure representing a clock */
typedef struct clock_node {
    clock_t ticks;    /* number of clock ticks recorded */
    clock_t last_time; /* time when clock was last switched on */
    char * name;    /* name of clock */
    struct clock_node * next; /* next pointer for linked list */
    bool clock_on; /* flag if clock started or stopped */
} clock_type, * clock_ptr;

clock_type * clock_head = NULL; /* head of linked list of clocks */

/* Set a clock off recording time */
void start_clock (char * name )
{
    clock_ptr p;

    for ( p = clock_head; p != NULL; p = p->next )
    {
        if ( strcmp ( p->name, name ) == 0 )
        {
            break;
        }
    }

    if ( p == NULL ) /* not found - create a new clock */
    {
        p = malloc ( sizeof ( struct clock_node ) );
        p->name = malloc ( strlen ( name ) + 1 );
        strcpy( p->name, name ); /* store the clock name */
        p->ticks = 0; /* no time on the clock yet */
        p->next = clock_head; /* add to the front of the linked list */
        p->clock_on = TRUE;
        clock_head = p;
    }
    else if ( p->clock_on )
    {
        fprintf( stderr, "Error: clock '%s' already on\n", name );
        return; /* no need to set last_time */
    }
p->clock_on = TRUE;       /* start the clock */
p->last_time = clock();    /* store the current time */
}

/*-------------------------------------------*/
/* Stop a running clock and update its count of elapsed time */
/*-------------------------------------------*/
void stop_clock( char * name )
{
    clock_t ticks = clock();        /* record time first */
clock_ptr p;

    for ( p = clock_head; p != NULL; p = p->next )
    {
        if ( strcmp( p->name, name ) == 0 )
            break;
    }

    if ( p == NULL )                /* Error - clock name not found */
        fprintf( stderr, "Error: clock '%s' not found\n", name );
    else if ( !p->clock_on )        /* Error - clock not started */
        fprintf( stderr, "Error: clock '%s' not started\n", name );

    p->clock_on = FALSE;            /* stop the clock */
    p->ticks += ticks - p_last_time; /* record elapsed time */
}

/*-------------------------------------------*/
/* Print out the profiling report based on all clocks */
/*-------------------------------------------*/
void clock_report( void )
{
    clock_ptr p;
    clock_t total = clock();       /* total time for the entire program */

    fprintf( stderr, "-------- CLOCK PROFILE --------\n" );

    for ( p = clock_head; p != NULL; p = p->next )
    {
        if ( p->clock_on )
            fprintf( stderr, "Error: clock '%s' not stopped\n", p->name );

        fprintf( stderr, "Clock '%s' :	%5.2f secs, %5.2f\%
", p->name,
            ( p->ticks / (double) CLOCKS_PER_SEC ),
            ( p->ticks / (double) total * 100.0 ) );
    }
}

Figure 1.2: Source file showing the implementation of the public functions made available by the instrumentation library provided by Spuler (1992).
The method supplied by Spuler (1992) was chosen for this study as it allows entire functions or small blocks of code to be profiled. Other methods such as the UNIX tools "prof" and "pixie" profile code based upon single functions or basic code block. In order to profile complex code, such as feature extraction, adding instrumentation to significant code segments allows the developer to know for certain which functions or groups of statements consume the most time Spuler (1992). Figure I.3 shows how the instrumentation library supplied by Spuler (1992) may be used to profile a given code segment. Example output from the instrumentation library appears in Figure I.4.

```c
/*-------------------------------------------------------------------------*/
/* EXAMPLE.C: Example source showing the use of the instrumentation library*/
/*-------------------------------------------------------------------------*/
#include <stdio.h>
#include "instrument.h" /* Include the instrumentation header file */

long sum( long n )
{
    long i, total = 0L;
    start_clock( "sum" ); /* start clock for sum */
    for ( i = 1; i <= n; i ++ )
    {
        total += i;
    }
    stop_clock( "sum" ); /* stop clock for sum */
    return total;
}

void main ( void )
{
    long i, total = 0L;
    start_clock( "main" ); /* start clock for main */
    for ( i = 1; i <= 1000; i ++ )
    {
        total += sum( i ); /* sum of sums */
    }
    printf( "%s of sums of 1..1000 = %ld\n", total);
    stop_clock( "main" ); /* stop clock for main */
    clock_report(); /* print out the profile */
}
```

Figure I. 3: Example source showing the usage of the instrumentation library provided by Spuler (1992).

---

64 A basic code block is a sequence of code that contains no branches Spuler (1992).
Figure I. 4: Example output from the instrumentation library when used to profile the code shown in Figure I. 3 (Spuler, 1992)
APPENDIX J  ACCURACY MEASUREMENTS FOR THE IRIS HYPOTHESIS METHOD

Table J.1: Combined deviations for the iris locations calculated by the IHM as opposed to the locations recorded by SCIS Research Support based upon the number of rows skipped during image processing.

<table>
<thead>
<tr>
<th>Calibration Point</th>
<th>Number Of Rows Skipped</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>Left (x-Axis)</td>
</tr>
<tr>
<td>A</td>
<td>0</td>
</tr>
<tr>
<td>B</td>
<td>0</td>
</tr>
<tr>
<td>C</td>
<td>0</td>
</tr>
<tr>
<td>D</td>
<td>0</td>
</tr>
<tr>
<td>E</td>
<td>0</td>
</tr>
<tr>
<td>F</td>
<td>0</td>
</tr>
<tr>
<td>G</td>
<td>0</td>
</tr>
<tr>
<td>H</td>
<td>0</td>
</tr>
<tr>
<td>I</td>
<td>0</td>
</tr>
<tr>
<td>J</td>
<td>0</td>
</tr>
<tr>
<td>K</td>
<td>0</td>
</tr>
<tr>
<td>L</td>
<td>0</td>
</tr>
<tr>
<td>M</td>
<td>0</td>
</tr>
<tr>
<td>N</td>
<td>0</td>
</tr>
<tr>
<td>O</td>
<td>0</td>
</tr>
</tbody>
</table>
Table J. 2: Comparison between the iris locations recorded by SCIS Research Support and those determined using the IHM without skipping rows of the Eye Image.

<table>
<thead>
<tr>
<th>Calibration Point</th>
<th>Iris Location Recorded By SCIS Research Support</th>
<th>Iris Location Determined By The IHM Algorithm</th>
<th>Accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Left (x-Axis)</td>
<td>Bottom (y-Axis)</td>
<td>Left (x-Axis)</td>
</tr>
<tr>
<td>A</td>
<td>225</td>
<td>395</td>
<td>225</td>
</tr>
<tr>
<td>B</td>
<td>162</td>
<td>373</td>
<td>162</td>
</tr>
<tr>
<td>C</td>
<td>170</td>
<td>403</td>
<td>170</td>
</tr>
<tr>
<td>D</td>
<td>255</td>
<td>386</td>
<td>255</td>
</tr>
<tr>
<td>E</td>
<td>250</td>
<td>326</td>
<td>250</td>
</tr>
<tr>
<td>F</td>
<td>226</td>
<td>253</td>
<td>226</td>
</tr>
<tr>
<td>G</td>
<td>235</td>
<td>246</td>
<td>235</td>
</tr>
<tr>
<td>H</td>
<td>312</td>
<td>241</td>
<td>312</td>
</tr>
<tr>
<td>I</td>
<td>310</td>
<td>204</td>
<td>310</td>
</tr>
<tr>
<td>J</td>
<td>270</td>
<td>187</td>
<td>270</td>
</tr>
<tr>
<td>K</td>
<td>275</td>
<td>199</td>
<td>275</td>
</tr>
<tr>
<td>L</td>
<td>319</td>
<td>189</td>
<td>319</td>
</tr>
<tr>
<td>M</td>
<td>314</td>
<td>168</td>
<td>314</td>
</tr>
<tr>
<td>N</td>
<td>268</td>
<td>184</td>
<td>268</td>
</tr>
<tr>
<td>O</td>
<td>287</td>
<td>174</td>
<td>287</td>
</tr>
</tbody>
</table>
Table J.3: Comparison between the iris locations recorded by SCIS Research Support and those determined using the IHM when skipping one row of the Eye Image.

<table>
<thead>
<tr>
<th>Calibration Point</th>
<th>Iris Location Recorded By SCIS Research Support</th>
<th>Iris Location Determined By The IHM Algorithm</th>
<th>Accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Left (x-Axis)</td>
<td>Bottom (y-Axis)</td>
<td>Left (x-Axis)</td>
</tr>
<tr>
<td>A</td>
<td>225</td>
<td>395</td>
<td>225</td>
</tr>
<tr>
<td>B</td>
<td>162</td>
<td>373</td>
<td>162</td>
</tr>
<tr>
<td>C</td>
<td>170</td>
<td>403</td>
<td>170</td>
</tr>
<tr>
<td>D</td>
<td>255</td>
<td>386</td>
<td>255</td>
</tr>
<tr>
<td>E</td>
<td>250</td>
<td>326</td>
<td>250</td>
</tr>
<tr>
<td>F</td>
<td>226</td>
<td>253</td>
<td>226</td>
</tr>
<tr>
<td>G</td>
<td>235</td>
<td>246</td>
<td>235</td>
</tr>
<tr>
<td>H</td>
<td>312</td>
<td>241</td>
<td>312</td>
</tr>
<tr>
<td>I</td>
<td>310</td>
<td>204</td>
<td>310</td>
</tr>
<tr>
<td>J</td>
<td>270</td>
<td>187</td>
<td>270</td>
</tr>
<tr>
<td>K</td>
<td>275</td>
<td>199</td>
<td>275</td>
</tr>
<tr>
<td>L</td>
<td>319</td>
<td>189</td>
<td>319</td>
</tr>
<tr>
<td>M</td>
<td>314</td>
<td>168</td>
<td>314</td>
</tr>
<tr>
<td>N</td>
<td>268</td>
<td>184</td>
<td>268</td>
</tr>
<tr>
<td>O</td>
<td>287</td>
<td>174</td>
<td>287</td>
</tr>
</tbody>
</table>
Table J.4: Comparison between the iris locations recorded by SCIS Research Support and those determined using the IHM when skipping two rows of the Eye Image.

<table>
<thead>
<tr>
<th>Calibration Point</th>
<th>Iris Location Recorded By SCIS Research Support</th>
<th>Iris Location Determined By The IHM Algorithm</th>
<th>Accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Left (x-Axis)</td>
<td>Bottom (y-Axis)</td>
<td>Left (x-Axis)</td>
</tr>
<tr>
<td>A</td>
<td>225</td>
<td>395</td>
<td>225</td>
</tr>
<tr>
<td>B</td>
<td>162</td>
<td>373</td>
<td>162</td>
</tr>
<tr>
<td>C</td>
<td>170</td>
<td>403</td>
<td>170</td>
</tr>
<tr>
<td>D</td>
<td>255</td>
<td>386</td>
<td>255</td>
</tr>
<tr>
<td>E</td>
<td>250</td>
<td>326</td>
<td>250</td>
</tr>
<tr>
<td>F</td>
<td>226</td>
<td>253</td>
<td>226</td>
</tr>
<tr>
<td>G</td>
<td>235</td>
<td>246</td>
<td>235</td>
</tr>
<tr>
<td>H</td>
<td>312</td>
<td>241</td>
<td>312</td>
</tr>
<tr>
<td>I</td>
<td>310</td>
<td>204</td>
<td>310</td>
</tr>
<tr>
<td>J</td>
<td>270</td>
<td>187</td>
<td>270</td>
</tr>
<tr>
<td>K</td>
<td>275</td>
<td>199</td>
<td>275</td>
</tr>
<tr>
<td>L</td>
<td>319</td>
<td>189</td>
<td>319</td>
</tr>
<tr>
<td>M</td>
<td>314</td>
<td>168</td>
<td>314</td>
</tr>
<tr>
<td>N</td>
<td>268</td>
<td>184</td>
<td>268</td>
</tr>
<tr>
<td>O</td>
<td>287</td>
<td>174</td>
<td>287</td>
</tr>
</tbody>
</table>
Table J.5: Comparison between the iris locations recorded by SCIS Research Support and those determined using the IHM when skipping three rows of the Eye Image.

<table>
<thead>
<tr>
<th>Calibration Point</th>
<th>Iris Location Recorded By SCIS Research Support</th>
<th>Iris Location Determined By The IHM Algorithm</th>
<th>Accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Left (x-Axis)</td>
<td>Bottom (y-Axis)</td>
<td>Left (x-Axis)</td>
</tr>
<tr>
<td>A</td>
<td>225</td>
<td>395</td>
<td>225</td>
</tr>
<tr>
<td>B</td>
<td>162</td>
<td>373</td>
<td>162</td>
</tr>
<tr>
<td>C</td>
<td>170</td>
<td>403</td>
<td>170</td>
</tr>
<tr>
<td>D</td>
<td>255</td>
<td>386</td>
<td>255</td>
</tr>
<tr>
<td>E</td>
<td>250</td>
<td>326</td>
<td>250</td>
</tr>
<tr>
<td>F</td>
<td>226</td>
<td>253</td>
<td>226</td>
</tr>
<tr>
<td>G</td>
<td>235</td>
<td>246</td>
<td>235</td>
</tr>
<tr>
<td>H</td>
<td>312</td>
<td>241</td>
<td>312</td>
</tr>
<tr>
<td>I</td>
<td>310</td>
<td>204</td>
<td>310</td>
</tr>
<tr>
<td>J</td>
<td>270</td>
<td>187</td>
<td>270</td>
</tr>
<tr>
<td>K</td>
<td>275</td>
<td>199</td>
<td>275</td>
</tr>
<tr>
<td>L</td>
<td>319</td>
<td>189</td>
<td>319</td>
</tr>
<tr>
<td>M</td>
<td>314</td>
<td>168</td>
<td>314</td>
</tr>
<tr>
<td>N</td>
<td>268</td>
<td>184</td>
<td>268</td>
</tr>
<tr>
<td>O</td>
<td>287</td>
<td>174</td>
<td>287</td>
</tr>
</tbody>
</table>
Table J. 6: Comparison between the iris locations recorded by SCIS Research Support and those determined using the IHM when skipping four rows of the Eye Image.

<table>
<thead>
<tr>
<th>Calibration Point</th>
<th>Iris Location Recorded By SCIS Research Support</th>
<th>Iris Location Determined By The IHM Algorithm</th>
<th>Accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Left (x-Axis)</td>
<td>Bottom (y-Axis)</td>
<td>Left (x-Axis)</td>
</tr>
<tr>
<td>A</td>
<td>225</td>
<td>395</td>
<td>225</td>
</tr>
<tr>
<td>B</td>
<td>162</td>
<td>373</td>
<td>162</td>
</tr>
<tr>
<td>C</td>
<td>170</td>
<td>403</td>
<td>170</td>
</tr>
<tr>
<td>D</td>
<td>255</td>
<td>386</td>
<td>255</td>
</tr>
<tr>
<td>E</td>
<td>250</td>
<td>326</td>
<td>250</td>
</tr>
<tr>
<td>F</td>
<td>226</td>
<td>253</td>
<td>226</td>
</tr>
<tr>
<td>G</td>
<td>235</td>
<td>246</td>
<td>235</td>
</tr>
<tr>
<td>H</td>
<td>312</td>
<td>241</td>
<td>312</td>
</tr>
<tr>
<td>I</td>
<td>310</td>
<td>204</td>
<td>310</td>
</tr>
<tr>
<td>J</td>
<td>270</td>
<td>187</td>
<td>270</td>
</tr>
<tr>
<td>K</td>
<td>275</td>
<td>199</td>
<td>275</td>
</tr>
<tr>
<td>L</td>
<td>319</td>
<td>189</td>
<td>319</td>
</tr>
<tr>
<td>M</td>
<td>314</td>
<td>168</td>
<td>314</td>
</tr>
<tr>
<td>N</td>
<td>268</td>
<td>184</td>
<td>268</td>
</tr>
<tr>
<td>O</td>
<td>287</td>
<td>174</td>
<td>287</td>
</tr>
</tbody>
</table>
APPENDIX K  CENTRE POINT CALCULATION EXAMPLE

Table K.1: Centre point calculation information for a sample iris image of even image height starting on an even row of the Eye Image.

<table>
<thead>
<tr>
<th>Number of Rows Skipped</th>
<th>Row Number</th>
<th>Iris Image</th>
<th>Start Row</th>
<th>End Row</th>
<th>Height</th>
<th>Centre Point</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td><img src="image1" alt="Iris Image" /></td>
<td>0</td>
<td>7</td>
<td>8</td>
<td>3.5</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td><img src="image2" alt="Iris Image" /></td>
<td>0</td>
<td>6</td>
<td>7</td>
<td>3</td>
</tr>
<tr>
<td>2</td>
<td>0</td>
<td><img src="image3" alt="Iris Image" /></td>
<td>0</td>
<td>6</td>
<td>7</td>
<td>3</td>
</tr>
<tr>
<td>3</td>
<td>0</td>
<td><img src="image4" alt="Iris Image" /></td>
<td>0</td>
<td>4</td>
<td>5</td>
<td>2</td>
</tr>
<tr>
<td>4</td>
<td>0</td>
<td><img src="image5" alt="Iris Image" /></td>
<td>0</td>
<td>5</td>
<td>6</td>
<td>2.5</td>
</tr>
</tbody>
</table>
Table K.2: Centre point calculation information for a sample iris image of even image height starting on an odd row of the Eye Image.

<table>
<thead>
<tr>
<th>Number of Rows Skipped</th>
<th>Row Number</th>
<th>Iris Image</th>
<th>Start Row</th>
<th>End Row</th>
<th>Height</th>
<th>Centre Point</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td><img src="image" alt="Iris Image" /></td>
<td>1</td>
<td>8</td>
<td>8</td>
<td>4.5</td>
</tr>
<tr>
<td></td>
<td>1</td>
<td><img src="image" alt="Iris Image" /></td>
<td>2</td>
<td>8</td>
<td>7</td>
<td>5</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td><img src="image" alt="Iris Image" /></td>
<td>3</td>
<td>6</td>
<td>4</td>
<td>4.5</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td><img src="image" alt="Iris Image" /></td>
<td>4</td>
<td>8</td>
<td>5</td>
<td>6</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td><img src="image" alt="Iris Image" /></td>
<td>5</td>
<td>5</td>
<td>1</td>
<td>5</td>
</tr>
</tbody>
</table>
Table K.3: Centre point calculation information for a sample iris image of odd image height starting on an even row of the Eye Image.

<table>
<thead>
<tr>
<th>Number of Rows Skipped</th>
<th>Row Number</th>
<th>Iris Image</th>
<th>Start Row</th>
<th>End Row</th>
<th>Height</th>
<th>Centre Point</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td><img src="image1" alt="Iris Image" /></td>
<td>0</td>
<td>6</td>
<td>7</td>
<td>3</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td><img src="image2" alt="Iris Image" /></td>
<td>0</td>
<td>6</td>
<td>7</td>
<td>3</td>
</tr>
<tr>
<td>0</td>
<td>2</td>
<td><img src="image3" alt="Iris Image" /></td>
<td>0</td>
<td>6</td>
<td>7</td>
<td>3</td>
</tr>
<tr>
<td>0</td>
<td>3</td>
<td><img src="image4" alt="Iris Image" /></td>
<td>0</td>
<td>4</td>
<td>5</td>
<td>2</td>
</tr>
<tr>
<td>0</td>
<td>4</td>
<td><img src="image5" alt="Iris Image" /></td>
<td>0</td>
<td>5</td>
<td>6</td>
<td>2.5</td>
</tr>
</tbody>
</table>
Table K.4: Centre point calculation information for a sample iris image of odd image height starting on an odd row of the Eye Image.

<table>
<thead>
<tr>
<th>Number of Rows Skipped</th>
<th>Row Number</th>
<th>Iris Image</th>
<th>Start Row</th>
<th>End Row</th>
<th>Height</th>
<th>Centre Point</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0 1 2 3 4 5 6 7 8</td>
<td></td>
<td>1 7 7 4</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>0 1 2 3 4 5 6 7 8</td>
<td></td>
<td>2 6 5 4</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>0 1 2 3 4 5 6 7 8</td>
<td></td>
<td>3 6 4 4.5</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>0 1 2 3 4 5 6 7 8</td>
<td></td>
<td>4 4 1 4</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>0 1 2 3 4 5 6 7 8</td>
<td></td>
<td>5 5 1 5</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
REFERENCES


