Eye Tracking using FLIR Blackfly® S

Kaushik Raj Nadar

January 2023

Contents

1 Introduction

Eye tracking is a valuable tool for understanding visual attention and visuo-motor control in cognitive psychology research. One cost-effective method of eye tracking is using a low-cost infrared camera. This method utilizes infrared light emitted by the eyes to determine gaze direction and is captured by the camera. The captured images are then processed using specialized software to provide accurate gaze data. This technique is a cost-effective alternative to traditional eye-tracking methods, making it accessible for researchers in cognitive psychology and visuo-motor control fields. Using this technique, researchers can perform experiments to study visual attention and visuo-motor control in tasks such as reading, scene perception, and object tracking. The results of these experiments can provide insights into how the visual system processes and interprets visual information, as well as inform the development of technologies that rely on eye-tracking such as assistive devices and human-computer interaction. However, it has some limitations such as lighting conditions and accuracy in certain head poses.

2 Installation

This guide is for using the BlackFly IR camera with a system running the Windows 10 OS.

1. Ensure Python 3.8.x is installed in the system.
   Link to python installation:
   `https://www.python.org/downloads/release/python-3810/`

2. Using the following command, you verify the version of Python installed:

   ```
   python3 --version
   ```

3. Download and install the Spinnaker SDK from here
   `https://flir.app.boxcn.net/v/SpinnakerSDK/folder/73503062578`
   (use Linux/Mac-specific drivers if you're not using Windows)

4. Download spinnaker python-3.x.x.x-cp38-cp38-win amd64.zip from the following website:
   `https://flir.app.boxcn.net/v/SpinnakerSDK/folder/73501875299`
   The 3.x.x.x may vary based on the downloaded file. (The last tested version when authoring this tutorial was 2.7.0.128)
   After the file is downloaded, extract the zip and place the whl file in the project folder.

5. Start PowerShell and go to the project folder using the cd command.

6. Create a python virtual environment. A virtual environment is a tool that helps to keep dependencies required by different projects separate by creating isolated python virtual environments for them. You can create a python virtual environment using the following command:

   ```
   python3.8 -m venv camenv/
   ```

7. Activate the virtual environment using this command:

   ```
   camenv/Scripts/activate
   ```

8. Now install the dependencies:

   ```
   pip install -r requirements.txt
   ```

9. Install the PySpin module:

   ```
   pip install spinnaker_python-3.0.0.118-cp38-cp38-
   win_amd64.whl
   ```

10. Now, with the virtual environment active, the application should function using the command:

    ```
    python AcquireAndDisplay.py
    ```

3 Usage

3.1 Starting the Program

This section demonstrates how to start the eye-tracking software.

1. Before running the program, connect the IR camera to the USB port of your computer.
2. Start PowerShell and go to the project folder using the cd command.
3. Activate the virtual environment using this command:

   ```
   camenv/Scripts/activate
   ```

4. To run the program simply use the command:

   ```
   python AcquireAndDisplay.py
   ```

5. Once the program starts, it will wait for a client to connect to the system via TCP/IP.
6. Start Monkey Logic App in MATLAB.
7. Open I/O settings.
8. Open USB settings.
9. Select My Eye tracker as the eye-tracker and click on the three dots button besides it.
10. Here, select TCP and configure the IP address and port used by the IR camera system. If you are unsure, then set the IP address to 127.0.0.1 and port number to 10001.
11. Now, you can successfully run tasks requiring eye-tracking in Monkey Logic

Note: If you hit the check connection button after establishing a TCP connection, you will need to relaunch the program and repeat the steps from the start before doing trials. Therefore, avoid tapping the check connection button.

3.2 Selecting the Region of Interest (ROI)

After starting the program, a viewfinder window appears on the screen along with the Monkey Logic windows. The program offers a useful feature to select the ROI by cropping a section of the screen window.

- Pressing ’s’ key, lets you select the ROI by freezing a frame at the instance when the key is pressed.
- Using the ’r’ key, you can reset the ROI to its default value.
- ’q’ key stops the program from acquiring images from camera and ends the program.

Every time the ’r’ key is pressed, the new ROI rectangle coordinates are stored in a text file named ’roi_coords.txt’. This file is located in the same folder as the program and can also be manually edited to set the desired ROI.

The main program AcquisitionAndDisplay.py contains the above features but alternatively, the SelectROI.py program can be used to set the ROI coordinates separately.

4 How it Works ?

4.1 Overview

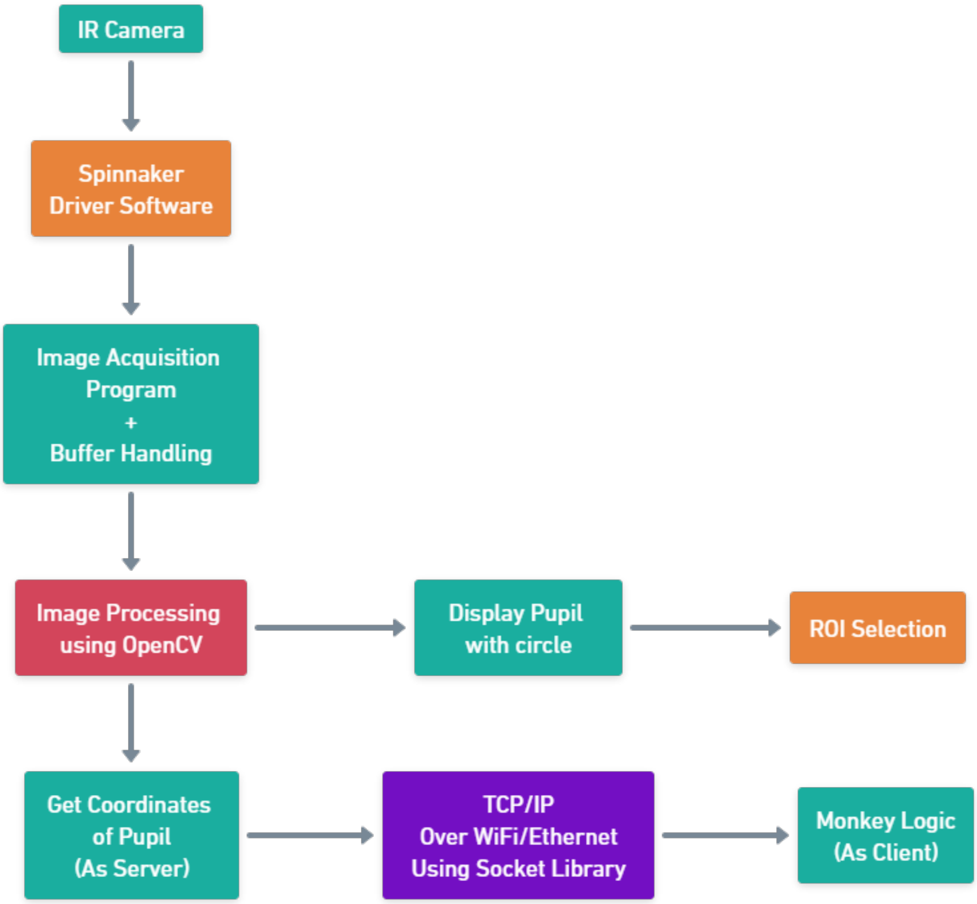

Figure 1: Flowchart depicting the flow of control of the eye tracking software

Using the camera's own proprietary driver software, the hardware is connected to the programme to create an interface. Image capture and buffer management are both tasks that may be carried out in an automated fashion when utilising this programme. A bounding circle or rectangle is created across the pupil, and the results are shown to the user via a graphical user interface (GUI). The pictures are processed in real time in order to find the pupils of the eyes, and the pupil coordinates are taken from each frame. After that, the coordinates of the pupil are sent to the Monkey Logic programme, which has a sampling rate of 200 Hz and may be utilised for real-time eye-tracking.

4.2 Buffer Handling

4.2.1 What are buffers?

When an image is captured by a camera it is automatically transferred to the PC and stored in the PC's RAM. The space assigned to store that image in memory is called a buffer. By default, the number of buffers allocated in memory is automatically assigned, based on the maximum frame rate of the camera. The number of buffers associated with a camera can be changed by switching the stream buffer count mode from auto to manual and assigning a value to the manual stream buffer count. The maximum number of buffers is determined by the available memory in the RAM and the size of each buffer is dependent on the image size. The minimum number of buffers you can set is 3.

4.2.2 Buffer Handling in FLIR Camera

Buffer handling determines the number of buffers available for each camera, if they can be overwritten, and the order in which images are retrieved.
In Spinnaker, there are four types of buffer handling:

1. **Newest First**—The application gets the image from the tail of the output buffer queue (the newest available one). This is useful for applications like motion prediction where it is critical to always get the newest frame but older frames can also add valuable information.

2. **Newest Only**—The application always gets the latest completed image (the newest one). Older images are discarded. Stream buffer count is ignored. This mode is typically used in a live display GUI where it is important that there is no lag between camera and display.

3. **Oldest First**—The application always gets the image from the head of the output buffer queue (the oldest available one).

4. **Oldest First Overwrite**—The application always gets the image from the head of the output buffer queue (the oldest available). If a new image arrives it overwrites the existing buffer from the head of the queue. This behaves like a circular buffer.

You can read more about buffer handling at the following link:
`https://www.flir.in/support-center/iis/machine-vision/application-note/understanding-buffer-handling/`

4.3 Image Processing

We have defined two functions to perform extraction of pupil coordinates:

1. First function is named detect_pupil (Circle).
2. Second function is named detect_pupil2 (Rectangle).

In both of these functions, there are some common operations performed[1]. First, some filters are applied to the image before finding the possible contours and then finding the minimum enclosing shape in both the cases. Crosshairs are also drawn at the center of the pupil as well as a bounding box/circle is drawn to enclose the pupil. Finally, the coordinates are normalized as required by the monkey logic and hence scaled down to a value range between -10 to 10.

4.3.1 detect_pupil function (Circle)

Steps involved for detecting pupil from images:

1. Load the image
2. Convert it into RGB colour image
3. Find its negative or inversion
4. Convert back to grey
5. Erosion of image to give contour a proper boundary
6. Threshold the image to separate the darker parts
7. Find the contours of the image with respect to area
8. Find the maximum area
9. Based on the maximum area, a circle is fit into the contour, thereby finding its center and radius
10. Return the normalized coordinates

[1]Special Thanks to Bharath for the image processing algorithm

4.3.2 detect_pupil2 function (Rectangle)

This function works in the following way:

1. Load the image
2. Convert the image to grey
3. Apply gaussian filter to remove gaps in the eyes
4. Threshold the image and obtain only the required pixel values which are related to eye
5. Find the contours or region with highest pixel density and sort it based on the area
6. Put a rectangle enclosing the area
7. Draw two perpendicular lines at the center
8. Return the normalized coordinates of the center

4.4 Socket Programming

Socket programming is a method of allowing two network nodes to interact with one another. While one socket (node) waits on a certain port at an IP address, another socket reaches out to the first socket in order to establish a connection between them. The listener socket is created on the server, while the client initiates communication with the server.
They are the true pillars upon which web surfing is supported. To break it down even further, there is a "client" and a "server."
Importing the socket library and constructing a basic socket are the first steps in the socket programming process.

```
import socket
tcp_socket = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
```

Here we made a socket instance and passed it two parameters. The first parameter is AF_INET and the second one is SOCK_STREAM. AF_INET refers to the address-family ipv4. The SOCK_STREAM means connection-oriented TCP protocol. Now we can connect to a client using this socket.

```
    server_address = ('127.0.0.1', 10001)
    tcp_socket.bind(server_address)
```

We are defining the IP address and port number to which the server should bind. In this case, the server is bound to IP address 127.0.0.1:10001. If monkey logic is used on the same system/pc, then the server address may be 127.0.0.1 or localhost. Leaving it blank means the server can be connected using any of its public IP.

```
tcp_socket.listen(1)
print("Waiting for connection: Kindly connect using Monkey Logic")
connection, client = tcp_socket.accept()
print("Connected to client IP: {}".format(client))
```

Here, we await a connection from the client side so that we can proceed to send the coordinates data over the network to Monkey Logic. On successful connection, the server should recieve an 'OK' message from the client (Monkey Logic).

```
    connection.send(csv_str.encode())
    connection.close()
```

Send the coordinates to the client in a CSV format using the send() method. Also, do not forget to write the close() method at the end of the program to close the TCP/IP connection.

4.5 Region Of Interest Selector and GUI

4.5.1 Region Of Interest (ROI) Selector

The Region of Interest (ROI) Selector is a script that crops the picture from the device before continuing to analyse it further. This ensures that the attention is kept solely on the eyes and that no other part of the image is taken into consideration. By doing things in this manner, we are able to obtain a more reliable and consistent influx of the pupil coordinates. This is particularly beneficial when the camera is kept a considerable distance away from the subject's eyes in order to prevent the subject from being distracted while seeing the experiment screen.

```
    ro = cv2.selectROI(image_data)
```

OpenCV's inbuilt function selectROI provides a simple GUI to select the ROI at a particular frame of the acquisition. When a ROI is selected successfully, the "roi_coords.txt" file in the project folder will be updated and the old coordinates will be replaced with the new ones.

```
image_data = image_data[int(ro[1]):int(ro[1]+ro[3]),
                        int(ro[0]):int(ro[0]+ro[2])]
```

The image is then cropped according to the coordinates of the ROI rectangle.
Note: The ROI can be modified manually using the "roi_coords.txt" file in the project folder.

4.5.2 Graphical User Interface (GUI)

The user interface (GUI) is displayed to the user by utilising the imshow function in OpenCV. It was discovered through experimentation that the imshow function of Matplotlib worked badly for high frame rate realtime video presentation. Because of this, OpenCV is the better option in this circumstance. On the other hand, the proprietary software developed by Spinnaker known as SpinView may also be utilised for the purpose of calibrating the experimental setup.
Limitation: The framerate of realtime streaming while using a GUI is capped at 66 Hz. This may be fixed by decreasing the exposure time (4500) in the SpinView Software.