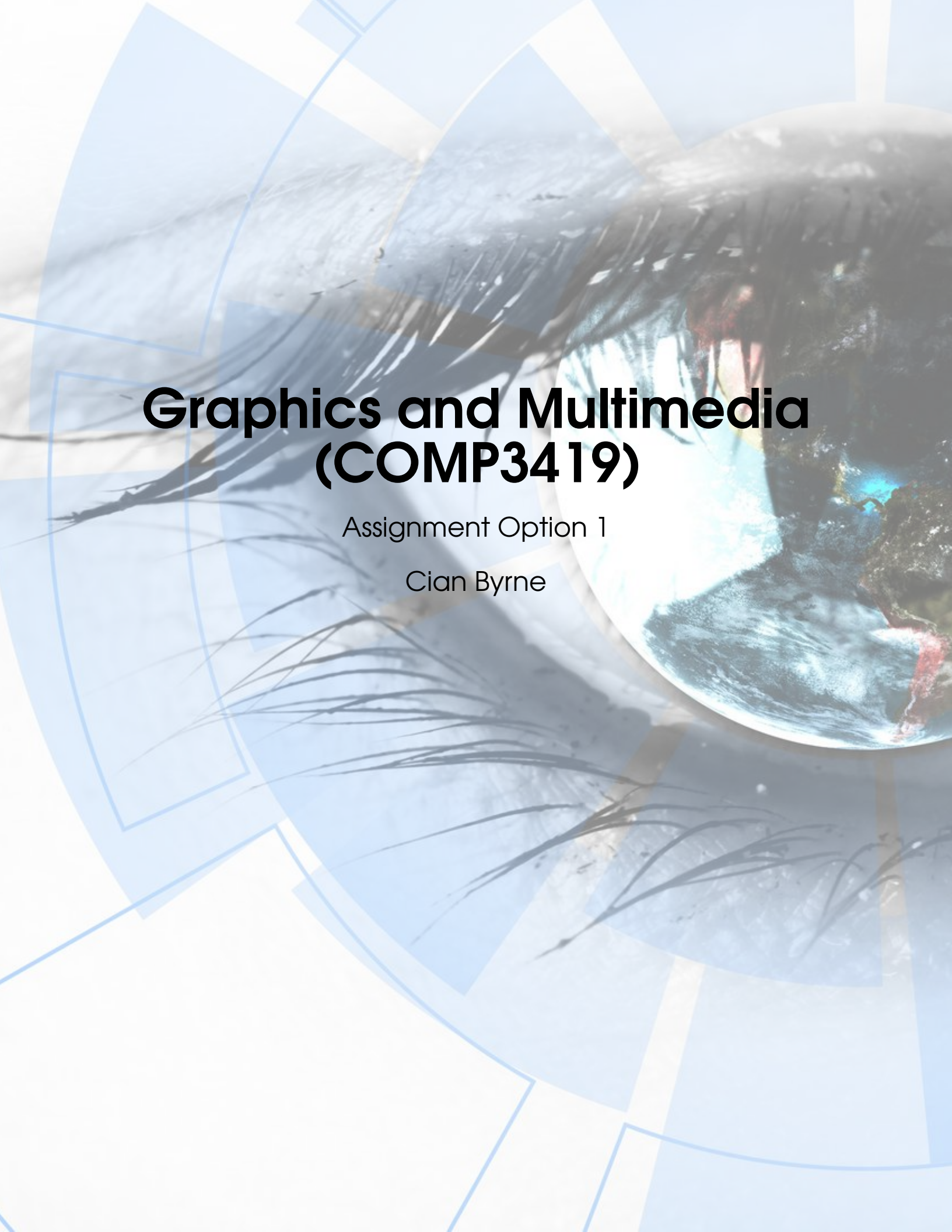
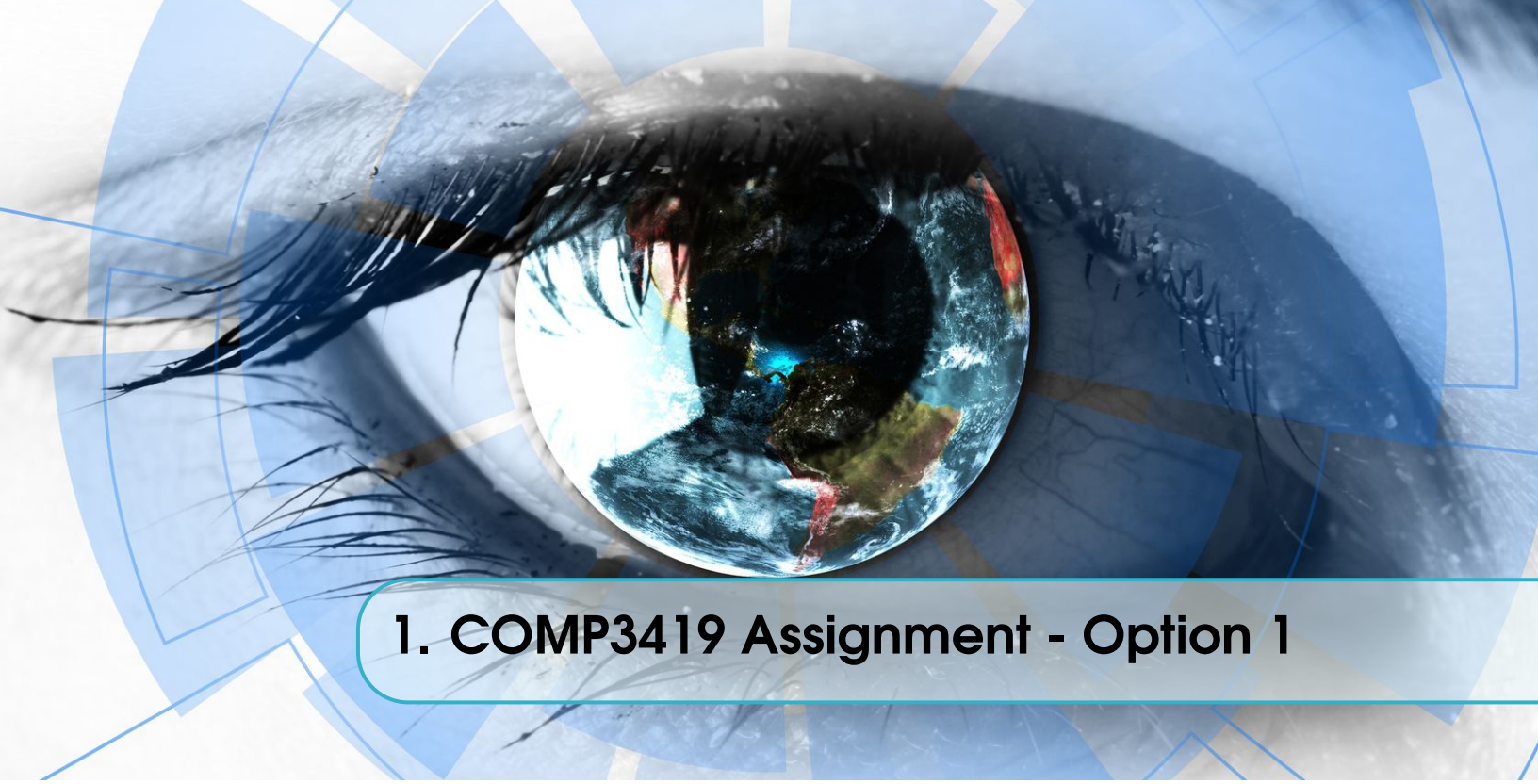


Graphics and Multimedia (COMP3419)

Assignment Option 1

Cian Byrne



1. COMP3419 Assignment - Option 1

1.1 Introduction

This report explores the techniques used to program a short video that incorporates several digital video processing, compositing and 2D animation techniques. These techniques were used to replicate motion of a moving object, simulate collision between objects, and produce special effects for demonstrating knowledge.

1.2 Implementation

To produce the final produce, there were several distinct stages. In this section, the process and algorithms that were used are explained, along with any experimental results or other techniques that were not used in the final product.

1.2.1 Motion Capture

This phase involved capturing the movements of a monkey that had red coloured markers attached to it. The challenge was that the markers were not always visible, were not completely red and at other times the same colour as other parts of the monkey. This made it extremely difficult to correctly capture all the movements of the monkey. Two techniques were experimented with.

The first part of this step was to remove the colours that were not part of the monkey. The RGB colour scale was used to achieve this.

Pre-Processing – Colour Removal

Case 1 – Basic Filter

The first colour filter that was used was very basic and only got rid of most the 'unwanted' pixels. It just used a single value for the red and green to filter out these pixels.

```
Red > 160 and Green < 110
```

This provided just enough resolution in the image to be able to do some average segmentation of the monkey but experienced issues as there was still some colour left in the face. This extra colour cased issues for later steps. So, a better filter was developed.

Case 2 – Advanced Filter

This technique works by applying three separate filters to the monkey video, each filter removing an individual colour set.

To develop this new filtering method, each pixel to be removed was written down and then individually removed. This was a very time consuming process and involved many updates and refinements. The result was very good, as only the pixels that we wanted were left. The values were found by experimentation, change a value, inspect image, note down values to remove, update the filters and repeat. Please see below for the final filter.

```
// raw filter
takeColor = red(c) > 149
          && green(c) > 37
          && green(c) < 199
          && blue(c) > 39
          && blue(c) < 125;

// face chromes
ignoreFilter1 = red(c) > 190
          && red(c) < 254
          && green(c) > 132
          && green(c) < 199
          && blue(c) > 44
          && blue(c) < 125;
// browns
ignoreFilter2 = red(c) > 148
          && red(c) < 201
          && green(c) > 83
          && green(c) < 166
          && blue(c) > 39
          && blue(c) < 116;

// if the pixel correct has color, calculate the new location
if( takeColor && !ignoreFilter1 && !ignoreFilter2)
```

Pre-Processing – Binary Image and Improvement

Binary Image

After the colours were separated and only the parts of the image that we wanted were present, a binary image was produced. Working with the binary image was far easier than using the coloured image. The binary image also allowed the use of metaphorical operations such as erosion and dilation to be used to improve the image.

Image Improvement

The trial and error testing method was used to find a nice mix of metaphorical operations. The order of operations was also tested. It was found that all erosion operations should be conducted at the start of the image improvement algorithm. If dilation was performed first, the image would become very distorted and extra pieces of the monkey that were not needed, would start to appear. This lead to the result of performing erosions first. Dilation was also needed after this, to make the parts of the monkey more visible, as the segmentation was not perfect.

The number of each operations were tested. After thorough investigation, the best number of erosions was discovered to be two; and the best number of dilations to be seven. After the improved image was obtained, the different methods for discovering the body parts were explored. Two methods were developed and tested.

Segmenting Motion – Finding Limbs

Method 1 - Blob Detection

This method relays heavily on the quality of the improved binary image given to the algorithm. If the image quality is poor, it is unlikely that the algorithm will be successful in finding the correct blob corresponding to the body part.

This algorithm works by forming clusters of points that are close to each other within a certain threshold. If a pixel is too far away from a cluster to be included, then a new cluster is formed. This is relatively accurate but can form blobs that are too large when parts of the image cross over. This inconsistency leads to some points missing because either the clusters didn't form correctly, or were missing because the hand or foot was not present on the current frame.

Some experimentation with the threshold was conducted. A threshold above nine resulted in the feet clusters merging. A threshold below give resulted in too any blobs being created – due to the image quality being poor. The golden threshold was discovered to be between five and seven. This produced the correct number of blobs most the time, but occasionally produced up to seven and as low as two blobs. This was not very helpful when we needed five all the time. A new method was chosen which yielded more reliable and consistent results.

Method 2 - Quadrant Search

This method is based on using a binary image that only contains the points that concern the monkey. The idea behind it is that if each of the four mittens are at the extremes of a rectangle, you can use the corners of the rectangle to model the motion.

In the above image, the complete process is shown for obtaining the five required points for mapping the motion of the monkey.

1. Original Monkey Video
2. Segmented Red Markers
3. Produced Binary Image
4. Improved Binary Image (after 2 x erode and 7 x dilate)
5. The basic points detection algorithm
6. The "blob" detection algorithm.

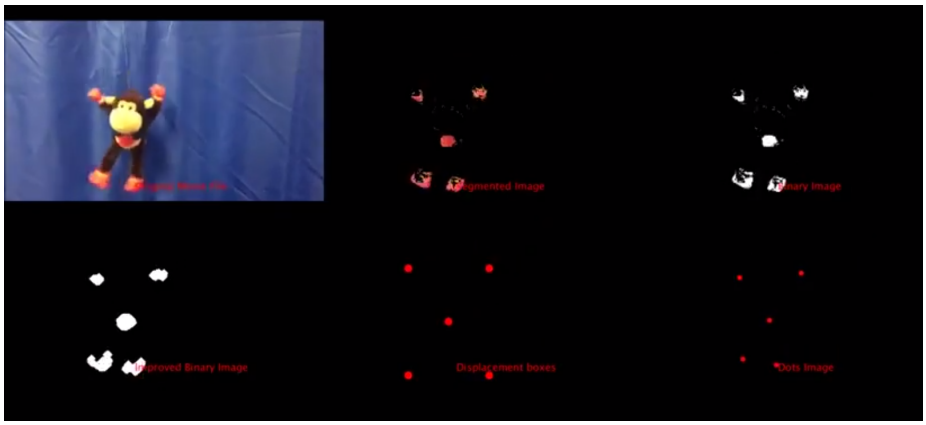
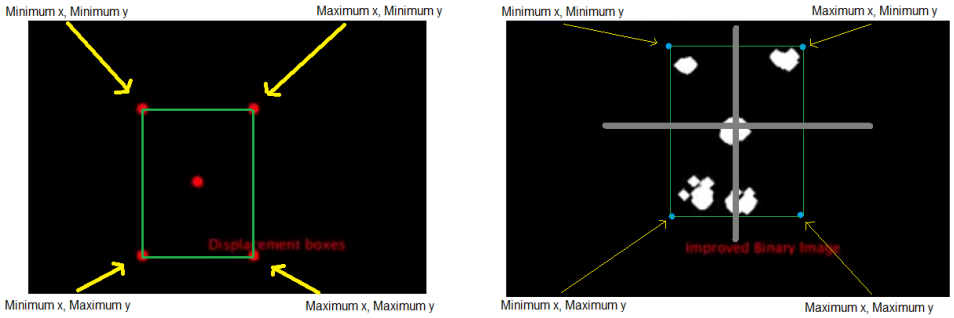


Figure 1.1: Demonstration of the different methods for separating the image.

Focusing on the (5) image. In the basic algorithm, the maximum and minimum points for x and y are found using a single iteration of the image. This is by detecting ONLY WHITE pixels on the image. It is not perfectly accurate but is still able to find a rough estimate for the points we are looking for. Using the maximum and minimum values, we can roughly calculate where the hands and feet of the monkey are. This produces a rectangle simulating very rough motion of the monkey. The centre point is calculated based on the maximum and minimum values. See below image:



(a) Basic process of getting points A

(b) Advanced method for getting points B

Figure 1.2: The Quadrant Search Method.

The search pattern is different for each of the points. For Example, the top left will start from the minimum x and minimum y values and work back towards the middle of the image (terminate in the middle). The bottom right will start from the maximum x and maximum y and work towards the middle in a kind of reversing fashion.

The idea being that, once the first white pixel is found in the search grid, the coordinates of that pixel can be used to as the location of the hand or foot. The centre point is still calculated using the minimum / maximum x and y coordinates (so when it disappears, it doesn't matter).

This method is slightly more accurate than the blob detection algorithm but does sometimes fail to find the hand or foot. This can be easily fixed by just using the maximum / minimum x and y coordinates depending on the grid being searched.

At this point the motion of the monkey has been successfully captured. The next stage is to

replace the background and replicate the motion that we have captured in previous steps.

1.2.2 Replace Background and Marionette

To replace the background is relatively trivial, considering that you can just start with a blank canvas and then insert and image as the background. This does not require further discussion.

The replacement of the monkey was also relatively simple, as explained in previous sections where the motion of the monkey was captured. It was just a matter of using the stored points and inserting the required body part at the correct location. The only other thing to note is that this must be drawn after the background, otherwise it will not appear on the screen. No further discussion is required.

1.2.3 Intelligent Objects

The intelligent objects were completed rendered in Processing and resemble different forms of public transport. Such as a bus, train and light rail transport. These objects had two different roles in the video. To be able to detect collisions with themselves and to interact with the replacement marionette (monkey). The drawing of the objects was trivial, it was just a matter of ensuring they were drawn on after the background, and replacement monkey character. They are drawn at randomly determined intervals during the execution of the program.

To detect the collision between similar objects, coordinate geometry was used. If you think of each of the objects as rectangles, then when one of the points from the object crosses over (or collides) with another, the x-values will overlap. For the purposes of this animation, this technique was used to determine collisions.

A similar technique was used to determine the collisions with the marionette. If any of the x-values overlapped either object, then a collision had occurred.

On a collision, a sound effect is triggered along with a flash of a flame image. Two different sound tracks were used to differentiate the collisions between the similar objects and the marionette.

At this stage, all the requirements of the assessment were fulfilled.

1.2.4 Conclusion

Throughout this project, there were several different techniques experimented with to find the best method to segment and capture the motion of the moving monkey. As evidenced by this report and the product, this has been successfully achieved. Each technique had both its positives and negatives, however, there could be better techniques for completing this exact same task which could have been further investigated if time permitted.

For more details on the implementation of the program, please review the code.