The Watching Window Project
10 Week Summer Bursary 2002

Name: KeeKim Heng
Supervisor: Geoff Wyvill

During my ten week period of work with the Computer Science Graphics and Vision Research group, I engaged research in several areas related to the watching window project. My initial task was to tidy up the code that had been developed last summer. I removed all open GL code as this was no longer necessary on the tracking computer, and added comments to the code. I also attempted to further abstract all image related functions into a separate file.

My first main goal was to improve the Brightness Correction routine that had been implemented by Kevin Novins last year. It worked by taking a sample from the four corners of each frame, and using these by comparing them to the original background frame to derive the linear function that transformed the brightness for each channel. The problem with this technique occurred when part of one of the sampled corners was blocked by the user. This introduced incorrect pixels and therefore produced incorrect linear functions. I implemented a technique that derived the average change for each corner and then removed the corner that had the greatest distance from this average. I then repeated this process with the three remaining corners, thus producing a linear equation from the two corners that were the least likely to be blocked by the user. This solved the problem effectively.

My second goal was to implement a technique that removed all blobs of foreground pixels except the largest. I achieved this by counting through the total number of blobs in the frame and looking for the blob with the most pixels. I sped up the process of finding the largest blob by reducing the resolution to a fraction of the original, to find the largest blob, and then blowing up the resolution back to it's original size. This also sped up the background subtraction process because it reduced the number of pixels that needed to be compared.

Thirdly, I wanted to improve the background subtraction method that was currently being used, which was comparing the current pixel to the original pixel by using the Manhattan distance between them and a global threshold. I researched many alternative techniques and decided that some sort of statistical background model would be the most suitable given the necessity to implement a fast technique (to handle the fast flow of incoming frames). I implemented the models defined in two papers, but found that even though both produced reasonable results, they involved to many computations and slowed the frame rate to an undesirable speed. In the end, I simplified the models described in the papers, so that a statistical model could still be implemented. This produced adequate results in a stable environment.

Although I had improved the background subtraction processes used by the Watching Window, the solution did not work as well as I had hoped in variable envirnoments. I had not solved the problem of shadows and changing light conditions. This lead me to implement a system that could be used to understand the image on a lower level. I developed a tool (the Visual Analysis Toolkit) that would allow me to understand the images captured by the camera. With the tool, I was able to capture short pieces of footage, and then analyse the image on a pixel level by viewing their distribution in a 3D colour space.

At this point, the Watching Window Project changed direction, as the project was to be presented at the Innovate Conference. This turned the focus to the application area of the project. For the conference, we needed to demonstrate the Project through a series of game-like applications.

My first step was to find out how to produce visually pleasing anaglyphs (stereoscopic images produced through the use of special glasses) that could be rendered at a fast enough rate. The computer I was using did not have special hardware for page flipping, or left and right buffering (used to contain the image each eye should see). I found an excellent method for drawing anaglyphs using openGL on the Internet, and incorporated this into a Watching Window application.

I produced three simple demonstrations that incorporated head and hand tracking. The first application displayed a window open to a scene of a desert. The user could move around the room to see different parts of the scene outside the window. The second application displayed three bells to the user. The user could then reach out and touch each bell, causing them to swing and make a sound. The final application was a game where the user had to dodge asteroids and collect mines with their head. I presented these three applications at the Innovate Conference.