**AINT308 Lab Practical Details**

All assessments are to be recorded and put on YouTube, with links to the videos in the report.

Assessment 1 – 40%

1. **Program the owl to move its neck in a sinusoidal manner at a reasonable rate** (you may find some rates produce less “jitter” than others). Use most of the neck servos range.
2. **Program 4 other behaviours that can be selected via a key press:**
3. **Move both eyes to scan the horizontal axis at a reasonable rate.** Both eyes are to be parallel, or verging on a virtual target.
4. **Show chameleon like eye motion for 10 seconds.** Watch videos of chameleon eye motion, and try your best to mimic its behaviour.
5. **The last two behaviours are up to you, but it must mimic a real eye motion with video evidence.** Such as some kind of human emotive eye motion.

Assessment 2 – 60%

1. **Verge onto a target using cross-correlation and servo control.** The code given to you already implements this on one eye, so modify it to work on both eyes. Once you have done this, the distance to the target can be approximated by assessing the angles of each eye. Produce a graph of the approximation against the ground truth distance, and try to make it more accurate by characterising your error.
2. **Calibrate your cameras/servos in order to produce a disparity map.** The code to make a disparity map has been given to you, but the owl must be properly set up for it to work.

Firstly, the cameras must be as parallel as possible. This can be achieved by moving the servos so that the centre of each camera verges on a distant target (e.g. a distant building). Save these values in the “owl.pwm” file under servo centre positions.

Then the cameras need calibrating for small misalignments the servos can’t correct for, and lens distortion. The owl-cv.h header file includes a function “OwlCalCapture” which can be used to take pictures by pressing “s”. Use this to take pictures of the chess board on your box at varying angles (though not too extreme). Examples of calibration images can be found in “Data/CalGood”. Modify “stereo\_calib.xml” to include the file paths of your calibration images, and run the software in “Tools/StereoCalibration”. This program will edit the “extrinsics.xml” and “intrinsics.xml” files in the data folder, which the disparity program uses for distortion correction. Calibration is very sensitive, and may fail a few times before you get a good correction, so try a few different image sets.

With the owl calibrated, modify the disparity code to produce a depth map (the brightness of each pixel is proportional to the distance). This can be done in a similar way to the cross-correlation task. Plot the disparity value against the ground truth distance experimentally, and characterise it. Have a distance approximation of the centre of the screen too, which will help gather this data.

1. **Program a Saccadic model to have the eyes saccade around a scene.** A saccadic model (as described in Itti & Koch’s bottom-up Saccadic model) adds together several feature maps into a “saliency” map. This map will highlight objects with a combination of interesting features, for example a brightly coloured and detailed shirt.

The code given to you already works as a saccadic model, however is only has a few feature maps. Your task is to add at least two extra feature maps, with justification of why they would be a good addition to the model.

Also add code to try and verge on salient targets, and give a distance approximation. This by no means is an easy task, and don’t worry too much about making it accurate.