Appendix A

Testing of the template model using Vision Egg software

A.1 Vision Egg

Recent advances in computer technology make it possible to produce visual stimuli not previously possible. Arbitrary scenes, from traditional sinusoidal gratings to naturalistic 3D scenes can now be specified on a frame-by-frame basis in realtime with modern graphics hardware. Andrew Straw developed a programming library called the Vision Egg that makes it easy to take advantage of these innovations. The Vision Egg is free, open-source software making use of OpenGL and written in the high-level language Python with extensions in C code. Careful attention has been paid to the issues of calibration and hardware interfacing, making the Vision Egg suitable for psychophysical, electrophysiological, and behavioral experiments [Straw, 2004].

A.2 Measuring angular velocity

Recall that in Chapter 2, we performed an experiment using a rotating drum to measure angular velocity using the template model. This had difficulty in recording the speed control uniformly. Therefore the experiment is repeated using the Vision Egg software to demonstrate improved results. The results are repeated from Chapter 2 to compare it with the Vision Egg results.
A.2 Measuring angular velocity

A.2.1 Using a rotating drum

We have set up an experiment to test the effectiveness of different de-noising algorithms. In this experiment, the camera was placed in the center of a white hollow cylinder with a vertical black paper bar inside. The cylinder is motor controlled and the angular speed of the cylinder can be adjusted by changing the voltage supply to the motor. Our program then tried to measure the angular speed of the cylinder by detecting the motion of the dark paper. The black paper is used to measure the angular velocity of the luminance templates. The experiment is repeated with red and blue paper stripes to measure the angular velocities of the red and blue chrominance templates respectively.

Experimental results show that a moving object (or edge) consistently causes the same motion sensitive template to occur at subsequent time steps, and at positions corresponding to the displacement of the edge relative to the detector [Yakovlev et al., 1994]. The angular velocity may be estimated by evaluating the ratio of the displacement of a motion sensitive template, to the time between the template’s occurrences (i.e., in Figure 2.11, the angular velocity is angular displacement/ΔT).

The rotating speed was increased from 10 rpm to 45 rpm in steps of 5 rpm. The horizontal axis represents real speed in rpm (varying from 10 rpm to 45 rpm), which was measured by using a tachometer. The vertical axis represents speed in rpm (varying from 10rpm to 45 rpm) measured using our program.

A.2.2 Using Vision Egg software

In this method, we used two computers, one which is the video capture computer that hosts our insect vision software (template model) and the other computer which we call the Vision Egg stimuli computer which generates the rotating stimuli which is used instead of the drum and which runs images on Vision Egg. It uses the python programming language to create different kinds of moving visual stimuli that can be very useful for the vision community. Many of the physiological experimental results obtained in this Thesis used this software for generating stimuli for the electro-physiological experiments. The Vision Egg software simulates a rotating drum and the images are rotated across the LCD screen. The CMOS camera mounted on a retort stand captures the moving images and sends the captured frames to the video capture computer for the insect vision software to perform analysis and motion detection. The video capture computer also has an EPIX SV4 image capture card installed in it and it acts as an interface for the CMOS camera to
Appendix A  Testing of the template model using Vision Egg software

send captured images to the computer. The CMOS camera is placed at about 17 cm away from the LCD screen so that it could capture a larger area of the rotating drum without capturing anything outside of it. To reduce the effect of ceiling lights on our experiments, the CMOS camera and LCD screen are covered using a book or box [Budimir et al., 2004; Guzinski et al., 2006].

The experiments performed with the rotating drum are repeated by placing the camera in front of a computer with a Vision Egg software running on it (See Chapter 2).

There are two algorithms for estimating velocity in real-time that have been developed and tested. The first algorithm is forward tracking [Yakovleff et al., 1995], and the second algorithm is stair-step tracking [Nguyen et al., 1993]. In this experiment, the forward tracking algorithm for velocity measurement is used. In the forward tracking algorithm, certain motion templates are tracked within a fixed time “window” of previous (small) displacements, and of the time steps at which they occurred, the velocity is provided by the ratio of the sum of the displacements, to the size of the window. Using this method, the velocity of a slow moving object is determined and updated at each sampling instant. In theory, this makes it more useful in counting templates for objects moving at low velocities [Nguyen et al., 1996].

![Graph showing comparison of angular velocities](image)

**Fig. A.1. Comparison of angular velocity curves with no pre-filtering using the rotating drum.**

Benchmark angular velocity measured by a tachometer versus angular velocity determined by insect vision system. Case 1: Without using any noise removal algorithms. Frame rate was 60 ms.
A.2 Measuring angular velocity

Fig. A.2. Comparison of angular velocity curves with no pre-filtering using the Vision Egg software. Benchmark angular velocity measured by the Vision Egg algorithm versus angular velocity determined by insect vision system. Case 1: Without using any noise removal algorithms.

Fig. A.3. Comparison of angular velocity curves with spatial averaging using the rotating drum. Benchmark angular velocity measured by a tachometer versus angular velocity determined by insect vision system. Case 2: Using the spatial averaging algorithm. Frame rate was 60 ms.
Appendix A  
Testing of the template model using Vision Egg software

![Graph showing angular velocity curves comparison](image)

**Fig. A.4.** Comparison of angular velocity curves with spatial averaging using the Vision Egg software. Benchmark angular velocity measured by the Vision Egg software versus angular velocity determined by insect vision system. Case 2: Using the spatial averaging algorithm. Frame rate was 60 ms.

![Graph showing angular velocity curves with MNC filtering](image)

**Fig. A.5.** Comparison of angular velocity curves with MNC filtering using the rotating drum. Benchmark angular velocity measured by a tachometer versus angular velocity determined by insect vision system. Case 3: Using the MNC method. Frame rate was 60 ms.
A.2 Measuring angular velocity

Fig. A.6. Comparison of angular velocity curves with MNC filtering using the Vision Egg software. Benchmark angular velocity measured by the Vision Egg software versus angular velocity determined by insect vision system. Case 3: Using the MNC method.

Fig. A.7. Comparison of angular velocity curves with template pair filtering using the rotating drum. Benchmark angular velocity measured by a tachometer versus angular velocity determined by insect vision system. Case 4: Using the template pair method. Frame rate was 60 ms.
Fig. A.8. Comparison of angular velocity curves with template pair filtering using the Vision Egg software. Benchmark angular velocity measured by the Vision Egg algorithm versus angular velocity determined by insect vision system. Case 4: Using the template pair method. Frame rate was 60 ms.

Fig. A.9. Comparison of angular velocity curves with averaging and template pair method. Benchmark angular velocity measured by a tachometer versus angular velocity determined by insect vision system. Case 6: Using the averaging and template pair technique. Frame rate was 60 ms.
A.2 Measuring angular velocity

Fig. A.10. Comparison of angular velocity curves with the averaging and template pair method using the Vision Egg software. Benchmark angular velocity measured by the Vision Egg software versus angular velocity determined by insect vision system. Case 5: Using the averaging and template pair technique.

Fig. A.11. Comparison of angular velocity curves with MNC and template pair technique. Benchmark angular velocity measured by a tachometer versus angular velocity determined by insect vision system. Case 8: Using the MNC and template pair technique. Frame rate was 60 ms.
Appendix A  Testing of the template model using Vision Egg software

Fig. A.12. Comparison of angular velocity curves with MNC and template pair technique. Benchmark angular velocity measured by a tachometer versus angular velocity determined by insect vision system. Case 8: Using the MNC and template pair technique. Frame rate was 60 ms.

Fig. A.13. Comparison of angular velocity curves with windowing method. Benchmark angular velocity measured by a tachometer versus angular velocity determined by insect vision system. Case 5: Using the windowing method. Frame rate was 60 ms. This experiment was done using the Vision Egg stimuli as we found that the template pair algorithm gives better results than the windowing method.
A.2 Measuring angular velocity

A.2.3 Analysis of Results

Results from the experiments are shown in Figures A.1 to A.13.

Figure A.1 shows that without applying any filtering process the response scatters randomly around the ideal response. Figure A.2 gives a much better result compared that obtained using a drum and this could be due to the use of the rotating cylinder which might not have rotated at consistent speeds and thus have affected the results. Results from Figure A.3 and Figure A.4 show that the averaging process helps in smoothing the response in both the cases. Figure 2.3 and Figure A.6 indicates that the measured speeds are shifting up from the ideal response. This can be explained by the calculation errors (division) of the digital computer. In this stage noise is sometimes amplified to a point where it dominates the signals induced by the motion of objects. In this filter we again averaged the luminance (chrominance) values over three receptors adjacent in the horizontal plane. Dividing a value by the average gives a result centered on 1. In the Template model implementation this result is not very useful as the change in intensity detected by the receptor over one frame must be larger then the threshold before that change is recognized. The minimum value of the threshold is 1 so no change in intensity and templates would ever be detected. To overcome this, the result of MNC filtering was scaled up by multiplying it by a constant value of a 100.

Figure A.13 indicates that applying the windowing operation to the templates helps in eliminating the constantly varying nature of the response. This is due to the local nature of the windowing algorithm. The template-pairs algorithm produces similar performance to the windowing algorithm, but the result is much more consistent. Since the windowing algorithm seems to leave noise in the result wherever the noisy templates are surrounded by other templates, this method is not included in the Vision Egg experiments.

Figure A.9, Figure A.10, Figure A.11 and Figure A.12 are combinations of pre-template filtering and post-template filtering. These figures indicate significant improvement in performance compared to using each individual technique or not using any technique at all. This result demonstrates that the pre-filtering techniques and the post filtering techniques do actually combine well in eliminating noise. From these experiments it can be seen that combining averaging, during pre-template filtering, and conjugate-pair techniques, during post-template filtering give the best overall response. However many more experiments with different conditions need to be carried out before such general conclusion can be reached.
Appendix A  Testing of the template model using Vision Egg software

In all the cases it is seen that the results obtained with the Vision Egg software gave better results than that using the drum and the reason for the inconsistency and variations of results could be the inconsistent angular speed of the stimuli. The angular speed of the stimuli is controlled by the voltage supply to the motor, thus the speed of the stimuli might not be as consistent or accurate.

The graph shows that the speed estimation algorithm works perfectly up to around 250°/s with the Vision Egg software. After that the detected speed drops slightly below the actual speed but is still quite accurate up to around 350°/s. It is especially good when looking at blue chrominance. It could be argued that in this case the filters were not needed at all. However it is still important to see if they improve the accuracy at higher speeds and check that they do not deteriorate the accuracy at lower ones.

The results show that the speed estimation algorithm is most accurate when the spatial averaging filter is used. This is not a surprising result as that filter is commonly used in other applications. Instead of implementing the filter in software, and thus using up CPU time, it is also possible to implement it in hardware by changing the focus of the camera lens so that the picture recorded is slightly blurred.

The MNC filter is not suitable for this application, as it introduces a lot of noise and destroys the precision of the system. This is caused by the fact that to use this filter in this application the values have to be multiplied after filtering by a 100 and the threshold set to 1. This first operation amplifies any remaining noise and the second one lets it through into the template detection stage.

The template pairs filter does not seem to have much effect on the accuracy of the speed estimation algorithm. If anything it probably makes it a bit less accurate. This is also the case when it is used in conjunction with the pre-template filters.

A.3  Experiment to compare Template model response with Reichardt correlator response using Vision Egg

The experiments conducted in Chapter 4 to compare the template model response to the Dror’s elaborated Reichardt model and the response of the HSNE neuron with the rotating drum has been repeated here using the vision egg software. For this experiment, we generated pictures with square black shapes on a white background using MATLAB.
A.3 Experiment to compare Template model response with Reichardt correlator response using Vision Egg

These square black shapes are random horizontal texture elements ('texels') and different sizes can be produced by changing the texture density parameter in the MATLAB code.

The CMOS camera was then subjected to these pictures with different texture density at a fixed distance of 17 cm on Vision Egg. Using Vision Egg, we could increase or decrease the speeds at which the pictures revolve around the CMOS camera. Our program then counts the average number of templates which are produced by detecting the movement of the texels. The average template count is obtained by averaging the total number of templates over 1000 frames. The procedure is repeated for five different texture densities over a range of speeds. The response of the Horridge template model is measured in the form of templates, so the velocity response curve to be plotted will be the velocity obtained from Vision Egg versus the template counts obtained from the software. A velocity response curve is plotted for each texture density and they can be found in Figure A.14.

![Effect of texel density and angular velocity on Template count using Template Model](image)

**Fig. A.14. Velocity response curves measured at five different texture densities with the Horridge template model using Vision Egg stimuli.** For all the curves, it can be seen that the response increases at lower velocities. This is due to the fact that as velocity is increased, more texels pass in front of the CMOS camera in a shorter time interval, resulting in the increase in response. However, the most of the curves level off and reach their optimum template count at around 50°/s. From this point, the response starts to decrease rapidly as velocity increases. This occurs because of the blurring effect caused by the fast motion of texels.
Appendix A Testing of the template model using Vision Egg software

From Figure A.14, we can see that for all the curves, the response increases at lower velocities. This is due to the fact that as velocity is increased, more texels pass in front of the CMOS camera in a shorter time interval, resulting in the increase in response. However, the most of the curves level off and reach their optimum template count at around 50°/s. From this point, the response starts to decrease rapidly as velocity increases. This occurs because of the blurring effect caused by the fast motion of texels [Rajesh et al., 2004]. The higher the texture density, the lower the velocity at which the blur occurs. As the template model is used for motion detection, the motion of the texels is detected as edges. At low texture densities, the texels are bigger and fewer in number. There are also a smaller number of edges and thus fewer templates at low velocities. However, as the texture density increases, the number of edges detected and the number of templates counted increases. This results in the curve shifting to the left with increasing texture density. This shows that it has a similar response to the velocity response curves of both the Reichardt correlator and the HSNE neuron presented in Chapter 3. We can also see that the results are more consistent and with less variation than the velocity response curves in Figure 4.1, shown in Chapter 4.

A reason for the better results obtained could be attributed to the use of Vision Egg stimuli instead of stimuli stuck on the inside of a rotating cylinder. By using Vision Egg, the angular speeds of the stimuli were always consistent and very accurate. But for the rotating cylinder, the angular speeds were controlled by the voltage supply to the motor, thus the speeds were not as consistent and this has resulted in the variations found in Figure 4.1.

To go a further step in comparing the velocity response curves, we have normalized the curves in Figure A.14 to a maximum value of 1.0. This is shown in Figure A.15.

From Figure A.15, we can see the same characteristics like in Figure A.14, with the response rising to a peak response at an optimal velocity and then falling off. For Figure A.15, the shifting of curves to the left as texture density increases is also more obvious than that of Figure A.14. This compares very favourably with the velocity response curves of the model correlator and the wide-field neuron of the hoverfly presented in Chapter 3. With the normalization, we can also see that the curves cease to shift to the left at the very highest densities. This is also evident in the velocity response curves of the model correlator and the wide-field neuron of the hoverfly found in Figure 3.4 and Figure 3.5 respectively shown in Chapter 3.
A.3 Experiment to compare Template model response with Reichardt correlator response using Vision Egg

Fig. A.15. Normalized velocity response curves measured at five different texture densities using the Horridge template model. To go a further step in comparing the velocity response curves, we have normalized the curves in Figure A.14 to a maximum value of 1.0.

In order to understand the velocity performance of the template model using the the texel images, we used the Vision Egg to display the textures of different densities at different speeds instead of using a rotating cylinder stuck with the random texture elements as stimuli. Our vision software estimates the angular velocity of the moving texels using the template model algorithm and the results are shown in Figure A.16. The horizontal axis represents the actual angular velocity of the texels obtained from Vision Egg and the vertical axis represents the measured angular velocities estimated by our software.

From Figure A.16, we can see that at velocities below $125^\circ/s$, all the lines lie almost exactly on the ideal line. It has been expected that there will be a slight deviation of lines from the ideal line at lower velocities due to the presence of noise in the system. At higher velocities, the lines deviate away from the ideal line and this is particularly true as texture density increases. This occurs because at higher velocities, the edges are not clearly identified due to the blurring effect caused by the fast motion of texels.
Fig. A.16. Benchmark angular velocity presented by the vision egg experiment versus angular velocity determined by insect vision system. It can be seen that at velocities below 125°/s, all the lines lie almost exactly on the ideal line. It has been expected that there will be a slight deviation of lines from the ideal line at lower velocities due to the presence of noise in the system. At higher velocities, the lines deviate away from the ideal line and this is particularly true as texture density increases and this is due to the blurring effect caused by the fast motion of texels.

A.4 Conclusion

Velocity estimation forms an important part of our insect Vision research and the comparisons made above focuses mainly on the reliability of the template model to accurately estimate velocity. The velocity response curves of the template model should be similar to those of the model correlator as both models are developed from an EMD. This has been proven to be true as both models have very similar responses. Experiments done on the wide-field neuron of the hoverfly have also verified the results obtained for the model correlator and our implementation of the template model as several important characteristics present in the curves of the wide-field neuron have been matched. The results obtained from Vision egg experiments are found to be more obvious and more conclusive than the velocity response curves using the drum. This is mainly due to the use of Vision Egg as stimulus compared to the rotating cylinder that did not always rotate at consistent speeds.
A.4 Conclusion

For the estimation of velocities using the template model, vision egg implementation fares better in many aspects when compared to the drum method. At lower velocities, vision egg experiment gives a more accurate estimation of velocity. However at higher velocities, only those with lower texture densities continue to provide accurate estimations. As the texture density increases, the estimations become less accurate at high velocities.
Bibliography


Bibliography


Bibliography


Bibliography


Bibliography


BIBLIOGRAPHY


Glossary

Acronyms that follow are in the order that they appear in the Thesis.

2D – Two Dimensional
HS – Horizontal System
VS – Vertical System
HSN – Horizontal System North
HSE – Horizontal System Equatorial
HSS – Horizontal System South
FD – Feature Detecting
EMD – Elementary Motion Detectors
MLG – Male Specific Lobula Giant
CMOS – Complementary Metal Oxide Semiconductor
DMST – Directionally Motion Sensitive Template
PCT – Position Conjugate Template
VLSI – Very Large Scale Integration
AGC – Automatic Gain Control
RGB – Red Green Blue
MNC – Multiplicative Noise Cancellation
HSNE – Horizontal System North Equatorial
LMC – Lamina Monopolar cells
LPF – Low Pass Filter
TF – Temporal Filter
DF – Delay filter
CRT – Cathode Ray Tube
Glossary

RAM – Random Access Memory
DDR – Double Data RAM
MB – Mega Byte
UAV – Unmanned Aerial Vehicle
3D – Three Dimensional
DC – Direct Current
LPTCs – Lobula Plate Tangential cells
LCD – Liquid Crystal Display
Resume

Sreeja Rajesh graduated from Bharathiyar University, India, with a First Class Bachelors in Engineering (Electrical and Electronic Engineering) in 1999. In 2000, Sreeja worked for Beonic corporation Pty. Ltd. (formerly called Traffic Pro Pty. Ltd.) with a team of research scientists to develop customer surveillance and people counting software using Insect vision based and Computer vision based algorithms. She commenced her PhD under the supervision of Prof Derek Abbott (School of Electrical and Electronic Engineering) and Prof David O’ Carroll (School of Physiology) at the University of Adelaide. She was awarded a US Airforce and Sir Ross and Sir Keith Smith funded scholarship.

Sreeja Rajesh is a member of SPIE and OSA and has authored and coauthored more than 10 publications and has given more than 10 presentations at conferences including an invited talk at the ISSS Conference on Smart Materials and Structures, Bangalore, India. She is currently employed by the Defence Science and Technology Organization (DSTO) in Edinburgh, South Australia.
My scientific genealogy via one of my supervisors is as follows:

<table>
<thead>
<tr>
<th>Year</th>
<th>Degree</th>
<th>University</th>
<th>Supervisor</th>
</tr>
</thead>
<tbody>
<tr>
<td>1774</td>
<td>MA</td>
<td>University of Cambridge</td>
<td>John Cranke</td>
</tr>
<tr>
<td>1782</td>
<td>MA</td>
<td>University of Cambridge</td>
<td>Thomas Jones</td>
</tr>
<tr>
<td>1811</td>
<td>MA</td>
<td>University of Cambridge</td>
<td>Adam Sedgwick</td>
</tr>
<tr>
<td>1830</td>
<td>MA</td>
<td>University of Cambridge</td>
<td>William Hopkins</td>
</tr>
<tr>
<td>1857</td>
<td>MA</td>
<td>University of Cambridge</td>
<td>Edward John Routh</td>
</tr>
<tr>
<td>1868</td>
<td>MA</td>
<td>University of Cambridge</td>
<td>John William Strutt (Lord Rayleigh)</td>
</tr>
<tr>
<td>1883</td>
<td>MA</td>
<td>University of Cambridge</td>
<td>Joseph John Thomson</td>
</tr>
<tr>
<td>1903</td>
<td>MA</td>
<td>University of Cambridge</td>
<td>John Sealy Townsend</td>
</tr>
<tr>
<td>1923</td>
<td>DPhil</td>
<td>University of Oxford</td>
<td>Victor Albert Bailey</td>
</tr>
<tr>
<td>1948</td>
<td>MSc</td>
<td>University of Sydney</td>
<td>Ronald Ernest Aitchison</td>
</tr>
<tr>
<td>1964</td>
<td>PhD</td>
<td>University of Sydney</td>
<td>Peter Harold Cole</td>
</tr>
<tr>
<td>1980</td>
<td>PhD</td>
<td>University of Adelaide</td>
<td>Kamran Eshraghian</td>
</tr>
<tr>
<td>1995</td>
<td>PhD</td>
<td>University of Adelaide</td>
<td>Derek Abbott</td>
</tr>
<tr>
<td>2007</td>
<td>PhD submitted</td>
<td>University of Adelaide</td>
<td>Sreeja Rajesh</td>
</tr>
</tbody>
</table>