Characterisation of Insect Vision Based Collision Avoidance Models Using a Video Camera

R. Guzinski [a], K. Nguyen [a], Z.H. Yong[a], S. Rajesh [a, b, c], D.C. O'Carroll [b, c], D. Abbott [a, b]

[a] School of Electrical and Electronic Engineering, The University of Adelaide, SA 5005, Australia
[b] Centre for Biomedical Engineering, The University of Adelaide, SA 5005, Australia
[c] School of Molecular and Biomedical Science, The University of Adelaide, SA 5005, Australia

ABSTRACT

Insects have very efficient vision algorithms that allow them to perform complex manoeuvres in real time, while using a very limited processing power. In this paper we study some of the properties of these algorithms with the aim of implementing them in microchip devices. To achieve this we simulate insect vision using our software, which utilises the Horridge Template Model, to detect the angular velocity of a moving object. The motion is simulated using a number of rotating images showing both artificial constructs and real life scenes and is captured with a CMOS camera. We investigate the effects of texel density, contrast, luminance and chrominance properties of the moving images. Pre and post template filtering and different threshold settings are used to improve the accuracy of the estimated angular velocity. We then further analyse and compare the results obtained. We will then implement an efficient velocity estimation algorithm that produces reliable results. Lastly, we will also look into developing the estimation of time to impact algorithm.

Keywords: artificial insect vision, Horridge Template Model, motion detection, collision avoidance sensors

1. INTRODUCTION

Insects have highly effective visual systems which enable them to perform intricate flight maneuvers, avoiding collisions with moving objects in crowded environments (e.g. flying amongst tree branches), and reacting to visual stimuli with high speed and precision, e.g. a male fly following after a female fly can adjust his course within 30 ms of the female changing her course[1]. In natural environments, they display an airborne agility which outperforms any man-made flying structure, and thus have captured the attention of NASA, the Australian and US defence forces. Making their feats even more impressive is the fact that flies have a brain which is only the two hair's breadths long and five across, more than one million times smaller than the human brain[2].

Thus insects are the primary motivator of this study, which attempts to simulate their motion detection ability using the Horridge Template Model. Implementing this model in software, we capture and analyse visual data which is in the form of a pattern or natural scene moving at constant rate, and analyse it to determine different properties of Template Model.

The underlying goal is to measure the speed of a moving object as accurately as possible. This is achieved by software implementation of the Horridge Template model, which attempts to simulate the visual system of insects, particularly in the way they detect moving edges.

The aim is to investigate the effect of the texel density of moving pattern, experimentation with natural images and environment factors on the response and accuracy of Horridge Template Model, building upon previous work[3, 4].

2. BACKGROUND

Experimentation and study into the insect visual system over many years has resulted in a better understanding of how insects so masterfully detect motion and thus avoid collisions. Several models have been developed that allow both

BioMEMS and Nanotechnology II, edited by Dan V. Nicolau, Proc. of SPIE Vol. 6036, 60361D, (2006)
1605-7422/06/$15 · doi: 10.1117/12.638897

software and hardware implementation, in an attempt to simulate the process, by which insects track moving object. In this study, the Horridge Template Model has been implemented.

2.1. Template Model

The Template Model was developed by G. A. Horridge in 1990[5]. The principle of the Template Model is based on a certain number of motion *templates* indicating direction of motion along a single dimensional path (e.g. leftwards or rightwards direction). These templates are derived from comparing the intensities of a pixel sampled at consecutive instances in time. There are three possible states: increase, decrease, or no change, which are represented by the following symbols:

Increase: ↑

Decrease: ↓

No Change: –

The pixels' intensities are compared over an entire frame at consecutive time instances; which will result in several combinations of states. These states indicate leftwards or rightwards movement.

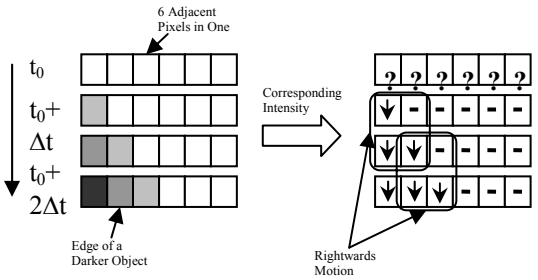

Figure 1: Dark object edge moving right to generate templates.

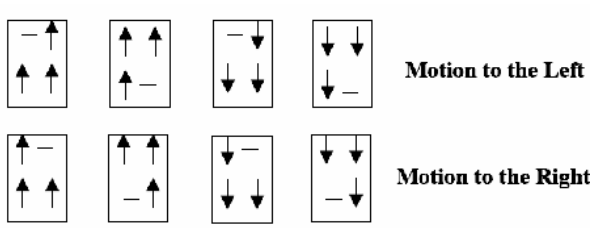

Figure 2: Two sets of templates which indicate leftwards and rightwards motion.

Figure 1 shows a particular set of six adjacent pixels is sampled at intervals of Δt. At time = t_0, the pixels are all white, an edge of a darker object progressively moves into the region and is sampled at t_0+Δt, t_0+2Δt and so on. As a moving edge enters the sampled region, the intensities at consecutive time instances for two adjacent pixels are compared to obtain a particular combination of these intensities which is referred to as a *template*.

When a dark object when moves into a light region, it will decrease the intensity in light region and a light object moving into a dark region will increase the intensity in dark region.

The top four templates in Figure 2 are for an object moving leftwards. First template corresponds to a light object moving into a dark region, second template is a light object leaving a dark region, third template is when a dark object moves into a light region, fourth template is when a dark object leaves a light region. The bottom four templates in Figure 2 are for an object moving rightwards.

Thus there are a total of eight templates, which essentially give an indication of moving edges. These templates are called Directionally Motion Sensitive Templates (DMSTs) Therefore, by generating arrays of these templates from each set of frames obtained at two consecutive time intervals, the moving edges can be matched in any two consecutive template arrays. Thus the offset between matched moving edges will be proportional to its speed.

Originally, the Template Model is based on luminance (or greyscale) information, it has been extended by Chew[4] to include chrominance (colour) information as well. By detecting changes in chrominance as well as changes in luminance, allow us to be able to obtain more accurate templates, more reliable and precise results. Due to the simplicity of the Template Model, it can be easily implemented in either hardware (VLSI chips) or software or both[6].

3. COMPARISON OF RESPONSES BASED ON TEXTURE DENSITY

The responses for the Reichardt correlator and the wide-field neuron of the hoverfly were compared when Dror and O'Carroll carried out a set of experiments to verify the relationship between image power spectra and velocity response curves[7]. The steady state response of wide-field neurons of the hoverfly with respect to the motion of textures of different densities was recorded over a range of velocities. The mean response level of the wide-field neuron was then plotted against the corresponding velocities for different texture densities to obtain the velocity response curves. The velocity response curves for the model correlator are then compared with those of the wide-field neuron. The shapes of the curves suggest that image power spectra and the velocity response curves are indeed correlated. The response of the model correlator and the wide-field neuron was then compared with that of the Horridge template model, which was implemented in software by Rajesh et al.[8] The same stimulus was used to obtain the velocity response curves. Although the shape of the curves differed with those of the model correlator and the wide-field neuron, there were still important characteristics that were similar when compared. As the results had variations and were not particularly conclusive, we implemented another version of the template model and obtained the response for it. Like the Reichardt correlator model, the template model is also developed from an elementary motion detector (EMD), which is the minimum prerequisite for directionally selective motion detection in a visual system, thus we would expect some similarity in the curves for the template model and the model correlator[8]. In the subsections below, the responses of the model correlator, wide-field neuron, Rajesh et al.'s implementation of the Horridge template model and our implementation of the template model will be clearly explained and compared.

3.1. Predicted response of the Reichardt correlator model

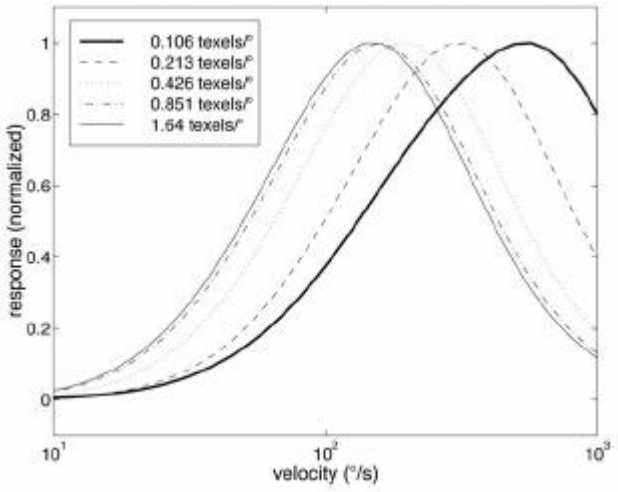

Figure 3: Velocity response curves predicted from the power spectra of random textures at five different densities for a model correlator. After [7].

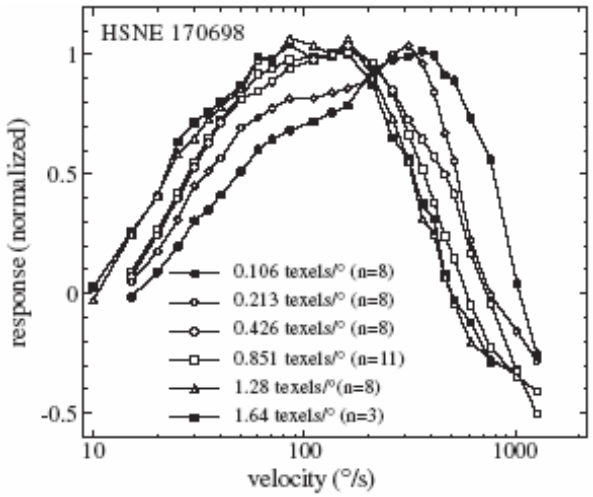

Figure 4: Velocity response curves measured at six different texture densities for a single HSNE neuron. The n value gives the number of measurements done and averaged for each texture density. After [7].

Figure 3 shows the velocity response curves for a model correlator that are predicted analytically using the power spectra of random textures consisting of horizontal rectangular texture elements ('texels') at five different densities. For the curves in Figure 3, the response levels for each curve vary significantly with density, thus the curves are normalized to a maximum value of 1.0 to facilitate comparisons. From Figure 3, we can see that all the curves rise up to a peak response velocity before falling off. The curves also shift to the left as texture density increases. These two characteristics of the curves should also be present in the velocity response curves for the wide-field neuron according to theoretical predictions.

3.2. Response of the wide-field neuron of the hoverfly

For the experiment, male specimens of the hoverfly *Volucella* were used. Horizontal system (HS) cells usually consist of three neurons which are termed north (HSN), equatorial (HSE) and south (HSS) horizontal cells. But in syrphids like *volucella*, there is an extra neuron named HSNE present. The HSNE neuron is used for the recordings to obtain the velocity response curves shown in Figure 4.

The hoverfly's actual perception of velocity might not be necessarily measured due to the additional processing that may occur subsequent to or in parallel with the HSNE neuron. But because the wide-field neurons carry out extensive spatial integration, plus the fact that the output data is averaged over time, thus the results obtained effectively become velocity response curves for the wide-field neurons[7]. Figure 4 shows the velocity response curves measured at six different texture densities for one HSNE neuron. The curves follow the prediction mentioned in the previous section and they rise up to a peak response velocity before falling off. With increasing texture density, the image power spectrum becomes flatter, thus resulting in the curves shifting to the left[7]. We can also see that the curves stop shifting to the left at the higher densities due to the mean horizontal power spectrum of the texture field becoming almost completely flat.

3.3. Response for the Horridge template model

In the subsections below, Rajesh et al's software implementation of the template model will be clearly explained and compared with our implementation of the Horridge template model.

3.3.1. Rajesh et al's software implementation

For the experiment, random horizontal texture elements ('texels') were printed on a horizontal roll of paper. The paper was then stuck on the inside of a hollow cylinder, which had a camera placed in the centre of it. A motor made the cylinder rotate at different angular speeds depending on the voltage supply to the motor. The software implemented template model would count the number of templates that are produced by detecting the movement of the texels. Texels of different sizes can be produced by varying the texture density parameter. Figure 5 shows velocity response curves for the Horridge template model.

The curves in Figure 5 show similar trends to the curves of the model correlator and the wide-field neuron of the hoverfly. Most of the curves rise up to a peak response at some optimal velocity and then fall off. The curves also shift to the left as texture density increases, emulating the response of the model correlator and the wide-field neuron. However, the results are not consistent and they show variations with the curves for the Reichardt model and the experiment done on the hoverfly. A reason for this could be the fact that although the Horridge template model is essentially a discrete–time model, the front end of the true Horridge template model carries out a continuous-time differentiation before digitizing the signal[8]. However, for the experiment, the Horridge template model implemented is fully in discrete-time. Thus, resulting in variations in curves obtained. Another reason for this could be the noise introduced by the digital-to-analogue conversion of the signal[8]. The third reason is the motion blur caused by the camera. The fourth reason for the inconsistency and variations of results could be the inconsistent angular speed of the stimulus. The angular speed of the stimulus is controlled by the voltage supply to the motor, thus the speed of the texels are subject to variations in motor speed.

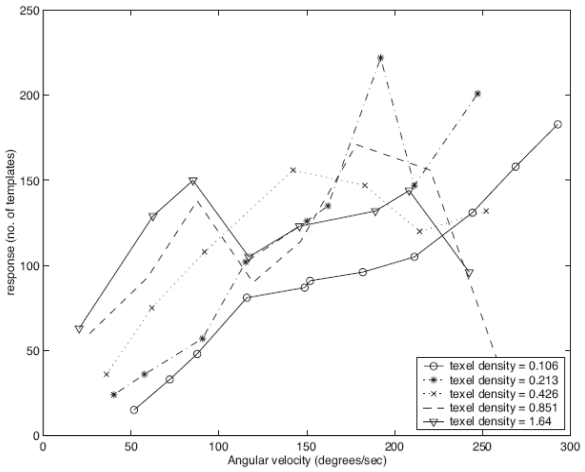

Figure 5: Velocity response curves measured at five different texture densities for the template model. After [8].

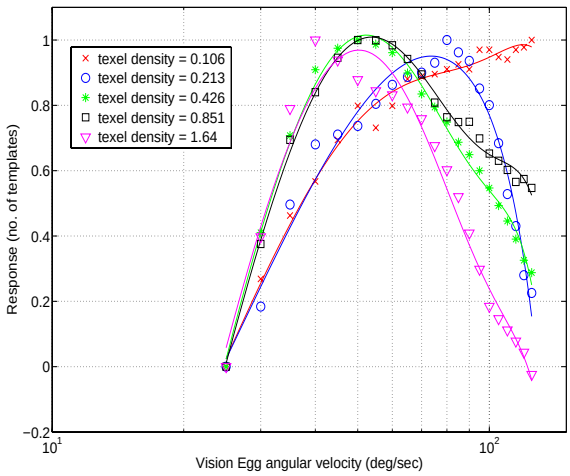

Figure 6: Normalized velocity response curves measured at five different texture densities using the template model (threshold = 30).

3.3.2. Our implementation of the template model

To compare the velocity response curves of the model correlator, wide-field neuron and Rajesh's implementation of the template model with our implementation of the Horridge template model, we conducted the experiment using our software. For the experiment, we generated pictures of random horizontal texture elements ('texels') using MATLAB[22]. Textures of different densities can be produced by changing parameters 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[23]. Using Vision Egg, we could increase or decrease the speeds at which the pictures revolve around the CMOS camera. Our software 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 are found in Figure 6.

The templates are counted with better accuracy at lower velocities than at higher velocities using our software. This ties in well within our expectations because the forward tracking method[9] is used in our software to track the templates and to determine their velocity. Using this method, the velocity of a slow moving object is determined and updated at each sampling instant[10]. In theory, this makes it more useful in counting templates for objects moving at low velocities. Thus, by using the software to obtain the template counts, we can expect the response of all texture densities to decrease at high velocities. This is due to the higher risk of having lost or missing templates when the texels move at higher speeds. Theoretically, we would expect the curves to be similar in shape to the responses of the model correlator and the wide-field neuron because the template model is essentially developed from an EMD like the correlator model[8].

From Figure 6, 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 response starts to decrease after the peak response velocity. This occurs because of the blurring effect caused by the fast motion of texels[8]. 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. We can also see that the curves cease to shift to the left at the very highest densities, a similar feature found in the velocity response curves of the wide-field neuron of the hoverfly. Thus, it compares favourably with the response of both the HSNE

neuron and the Reichardt correlator. The results are also more consistent and more conclusive than the velocity response curves done by Rajesh. A reason for the better results obtained could be attributed to the use of Vision Egg stimulus instead of stimulus 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.

3.4. Estimating angular velocities using the template model

By using the template model, the angular velocity of an object can be estimated by evaluating the ratio of displacement of a motion sensitive template to the time between the template's occurrences[8]. This estimation can be made because, through experiments, it has been shown that a moving object consistently causes the same template to occur at subsequent time steps and at positions that correspond to the displacement of the object relative to the detector[9]. Thus, if we make a plot of actual angular velocity versus measured angular velocity, we can conclude whether the Horridge template model is effective in estimating the angular velocity of an object. In the following subsections, we will be comparing the graphs done by Rajesh using the rotating cylinder as stimuli and our graphs that are obtained using the Vision Egg as stimuli.

3.4.1. Rajesh et al's software implementation

The software implemented Horridge template model estimates the angular velocity of the texels that were printed on a horizontal roll of paper and stuck on the inside of a rotating cylinder. Five different texture densities were measured from speeds of 25 deg/sec to 275 deg/sec. The results are shown in Figure 7. The horizontal axis represents the actual angular velocity measured using a tachometer while the vertical axis represents the measured angular velocity estimated by the template model algorithm.

From Figure 7, we can see that at lower velocities, the curves are very close to the identical line meaning that the template model algorithm gives a very accurate estimate of angular velocity. But as the angular velocity increases, the curves deviate from the ideal line as motion blurring of the texels occur.

3.4.2. Our implementation of the template model

To compare the graphs, we used the Vision Egg to display the textures of different densities instead of a rotating cylinder. Our software estimates the angular velocity of the moving texels using the template model algorithm and the results are shown in Figure 8. The horizontal axis represents the actual angular velocity of the texels obtained from Vision Egg and the vertical axis represents the estimated angular velocities from our software.

From Figure 8, we can see that at velocities below 125 deg/sec, 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. Thus the response deviates more from the ideal line at higher velocities. It can also be seen that as the texture density increases, the velocity from which the deviation from the ideal line starts to occur decreases. The lower the texture density, the more accurate the estimation will be at higher speeds. While at high texture densities, the estimation gets worse. This is due to the fact that at higher texture densities, there are more edges to detect and at higher velocities, the texels pass the CMOS camera at a faster rate, resulting in the loss of templates. Motion blurring is also added to this effect and the response gets worse at higher densities.

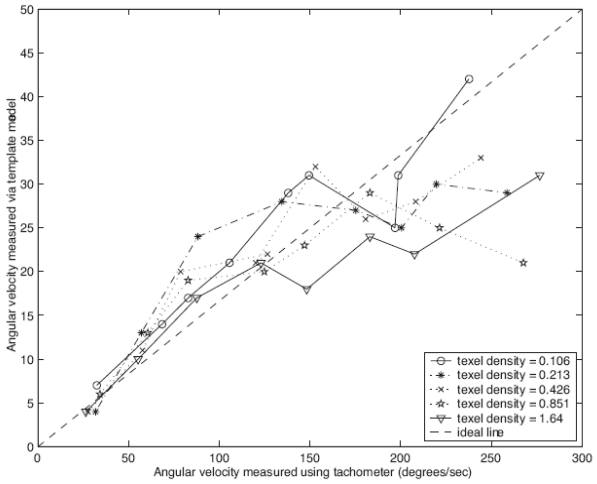

Figure 7: Angular velocity measured by the tachometer versus angular velocity obtained using the Horridge template model for five different texture densities. After [8].

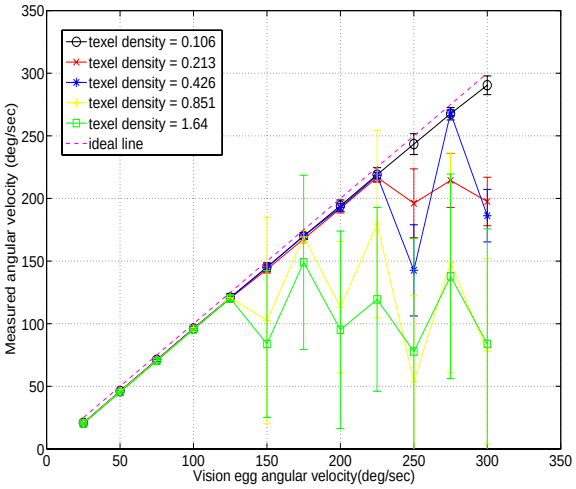

Figure 8: Vision Egg angular velocity versus angular velocity estimated using the Horridge template model for five different texture densities.

3.5. Discussion of results

The comparisons made above focus 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 our implementation of the template model as several important characteristics present in the curves of the wide-field neuron have been matched. Our results are also more obvious and more conclusive than the velocity response curves from Rajesh's implementation. 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.

For the estimation of velocities using the template model, our implementation also fares better in many aspects when compared to Rajesh's implementation. At lower velocities, our implementation 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. Overall, our implementation of the template model produces a reliable response and estimates velocities considerably well within certain limits.

4. VELOCITY RESPONSE CURVES FOR NATURAL IMAGES

4.1. Description and Setup

Although most natural scenes have similar spatial structure (texel density) [11] their contrast varies quite substantially. Studies of insect eye neurons have shown that while parameters like contrast are fundamental in early visual processing, they do not affect the final velocity response curves[12]. This property is referred to as velocity constancy and it suggests that there must be some mechanisms in the visual motion processing in insect's eyes which allow them to reject the effects of factors such as contrast.

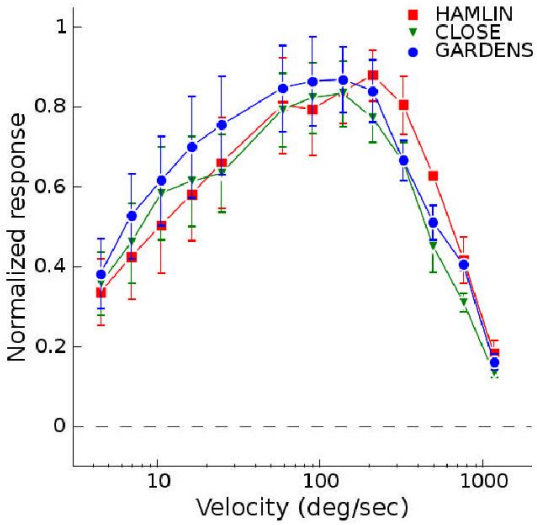

Figure 9: Velocity tuning curves obtained from eye cells in the hoverfly Eristalis tenax. The three images used as stimuli, identified by the labels at top, vary significantly in global contrast. The normalisation eliminates variability due to differences in scale of the curves 'quality of recording' between animals. Error bars indicate ± 1 standard error. Experimental preparation and stimulus display are as described in [13]. After [13].

Finding out what those mechanisms are and how to simulate them using the Template model is important because it would allow us achieve velocity constancy. If that were the case the speed of a moving image could be determined just by looking at the magnitude of the response. Up to the optimal speed there is a one-to-one relationship between the magnitude of the response and the magnitude of the velocity. That is in fact the way the insects determine the velocity of the objects around them and is called the optical flow. The larger the optical flow (response) the faster the apparent motion of the object relative to the insect's eye.

Figure 10 shows the results of previous study carried out using the Reichardt Correlator model[14]. They indicate that the system came the closest to velocity constancy when spatial high pass filtering and motion adaptation were used. In this experiment we investigate if these two features implemented using the Template model would also produce a similar outcome. The features' models are designed to capture the general characteristics of their biological equivalents.

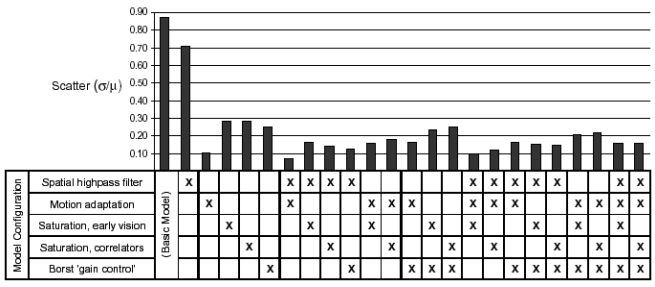

Figure 10: Summary of results from all simulations showing inter-scene scatter in the Correlator model response to five natural test images with different contrasts used in simulations. Scatter is measured as standard deviation divided by mean over the responses to the five-image set, where each response is the average output over one image rotation. After [14].

Figure 11: The top image shows the result of inverting the colours of a high pass spatially filtered version of the bottom natural panorama. The bottom is the original image.

The motion adaptation in insect eye cells consists at least of a contrast gain reduction[13]. In the Template model the gain reduction was implemented as automatic threshold control. The threshold for each frame was set according to the contrast of that frame as the mean of the absolute deviation of each pixel's luminance values[12]. Therefore the higher the contrast of the frame the higher the threshold and the less sensitive the model is to luminance intensity changes.

Spatial high- pass filtering is frequently encountered in the field of neurobiology and it has been shown that it also exists in the biological visual systems[15]. It has also been observed that most of differences in spatial structure of natural scenes exist in the low frequency components[14]. In this experiment the colours of the filtered images where inverted so that the background is light while the edges are darker. This was carried out because we observed that the camera was more sensitive to dark edges on light background than to light edges on dark background.

4.2. Results and Discussion

Figure 12 shows the response curve for two natural images, both when setting the threshold to a constant value and when using the auto threshold.

Looking just at the response curves produced when the threshold was set to a constant value it can be seen that natural images produce similar general shape of the graph. Although the magnitude of the response varies significantly between the two scenes at lower speeds, the optimal velocity seems to be very similar at around 90 deg/sec. This suggests that as predicted the texel densities of the two natural panoramas are very similar. Also since the graph has the same shape as seen when looking at the response curves of the black and white patterns it seems that the template model works correctly when presented with more complex natural scenes.

The response curves produced when auto threshold is used look less promising. They get further and further apart as the angular speed increases. However a different picture emerges when the curves are normalised as shown in Figure 13.

After normalising the data the response curves produced when using the auto threshold are almost identical. The two curves obtained when the threshold was set still have an amplitude difference between them. This shows that the auto threshold has a positive effect on the velocity response of the Template model. When the auto threshold is used the response curves have different slopes in their linear section but otherwise their shapes are identical up to the optimal velocity.

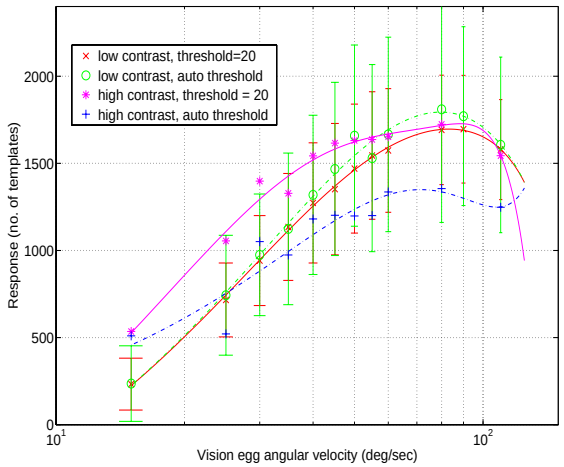

Figure 12: The response curves for two natural scene panoramas, one with low contrast and the other with high contrast. The templates were counted only in the luminance plane, and the total consists of both the templates indicating the correct direction of motion and the incorrect one. Error bars indicate standard deviation.

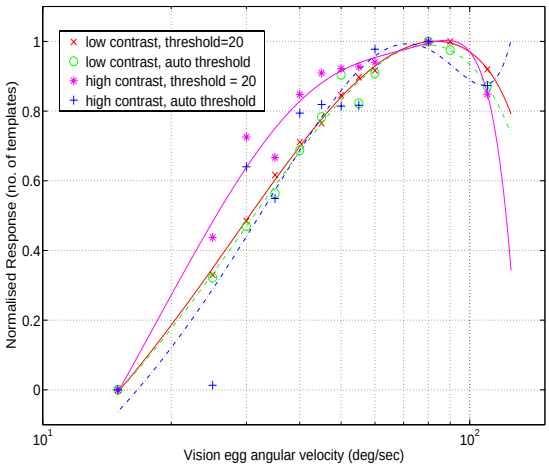

Figure 13: Normalised response curves produced by the two natural images used in Figure 12.

Figure 14 shows the velocity response curves obtained by the template model when looking at the moving high- pass spatially filtered panoramas. Although the curves still have the correct shape up to the optimal velocity, the optimal velocity has been substantially reduced. The optimal velocity of non – filtered images was around 90 deg/sec but for the filtered ones it dropped to only around 17 deg/sec. The curves also have very different magnitudes and they never approach each other.

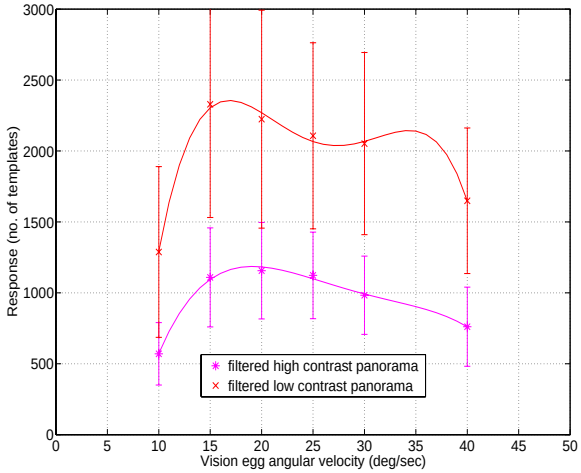

Figure 14: The velocity response curves for two spatially high-pass filtered natural panoramas. Only the luminance templates were counted and the both templates showing the correct and incorrect direction of motion were included. The error bars show standard deviation.

It looks like the high- pass filtering has no positive effect on the velocity response curves. However we think that it was most likely caused by the inadequacy of the equipment, especially the CMOS camera, and not by the template model. The camera had problem recognising the edges in the filtered images. This might be due to its automatic brightness control which was too sensitive and bleached out the received picture. At higher speeds the already faint edges blurred even more and became even harder to detect thus reducing the number of templates produced.

Because of the low optimal velocity obtained while looking at the response curves of the filtered images we decided not to perform the experiment where both the automatic threshold control and spatial high- pass filtering are used. At best the result would have the same, low optimal velocity. However it is recommended to perform this experiment in the future, using a new camera.

5. PRE AND POST TEMPLATE FILTERS

5.1. Description and Setup

In this experiment we set out to check if using pre and post template filtering would improve the accuracy of the motion detection algorithm by removing the signal noise. Two pre- template filters, spatial averaging and multiplicative noise cancellation, and one post- template filter, template pairs, were tested. Those filters were chosen to enable a comparison with results obtained by Rajesh and Abbott[16] since the experiment described in that paper gave inconclusive results.

We performed experiments with each filter individually as well as with combinations of one pre- template filter and the post- template one. Each experiment was repeated for luminance and red and blue chrominances with the squares coloured black, red and blue respectively.

The pre- template spatial averaging filter performs spatial smoothing of the input signal and is basically a low pass filter. It works by replacing the intensity value of each receptor's (pixel's) luminance or chrominance by the average value of the intensities from the surrounding receptors. In this experiment we took the average over three adjacent receptors in the

horizontal plane. This is a very well known filtering algorithm used in many applications to reduce the noise of the input signal. It is also present in the eye optics[14].

Figure 15: Spatial averaging smoothes out the input signal.

Multiplicative Noise Cancellation (MNC) was developed by Moini et al.[17] It is most useful when there is noise that is common among few neighbouring detectors. In that case spatial averaging does not remove the noise. However dividing the magnitude of one channel's luminance or chrominance by the average of the values from neighbouring receptors is more effective[17, 18]. Therefore the MNC consists of two stages: spatial averaging and division by averages.

This kind of noise usually comes from artificial room lighting. Experimental results indicated that the 100 or 120 Hz frequency component of light sources, operating with 50 or 60 Hz mains power, severely affect the detection of motion because of the temporal differentiation stage[16]. 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 centred 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 recognised. 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.

The last implemented filter was the Template Pairs filter. This filter is used after the formation of templates to remove the ones which were created by noise. Experimental results show that templates indicating motion always occur in pairs[19]. There are two types of template pairs. The first one is called a *motion conjugate* pair and it exists in any coherent motion where each DMST should be adjacent to a corresponding PCT. The second type comes from theoretical results produced by Nguyen et al.[20] which show that a moving edge always produces 2 associated DMSTs. One corresponds to an edge entering a region while the second one to an edge leaving that region.

The Template Pairs filter was implementation by looking at the horizontal neighbourhood of each receptor which detected a motion template at a given instant. If a corresponding DMST or PCT did not exist in that neighbourhood the template was treated as noise and removed. The speed estimation algorithm was called only after all the noise induced templates were removed.

5.2. Results and Discussion

As a basis for comparison of the results we first run the Insect Vision with none of the filters turned on. The results are shown in Figure 16.

The graph shows that the speed estimation algorithm works perfectly up to around 250 deg/sec. After that the detected speed drops slightly below the actual speed but is still quite accurate up to around 350 deg/sec. 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.

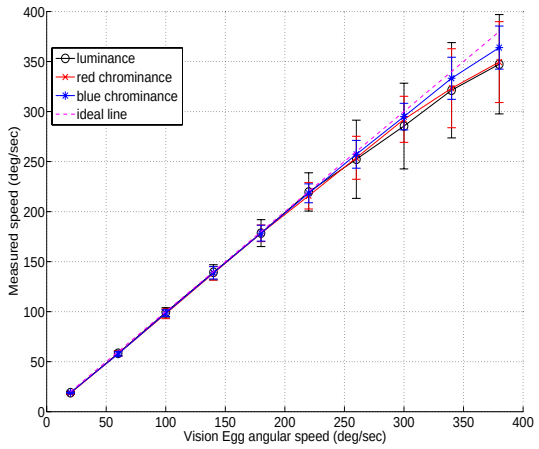

Figure 16: The graph of actual speed vs the estimated speed without using any filters. The error bars show standard deviation.

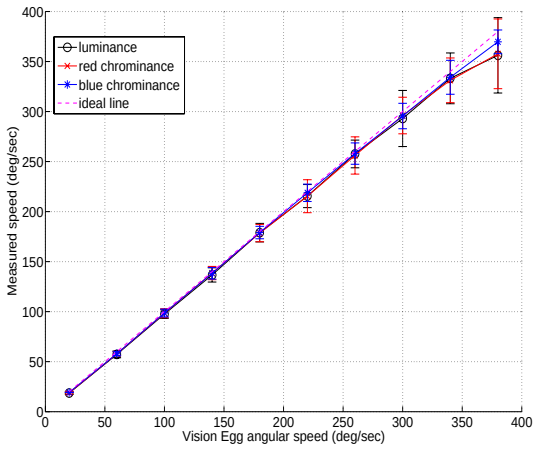

Figure 17: The graph of actual speed vs the estimated speed using spatial averaging filter. The error bars show standard deviation.

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 our application we had to multiply the values after filtering by a 100 and to set the threshold 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.

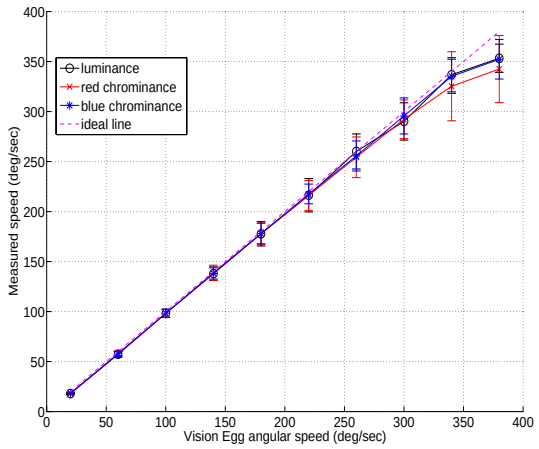

Figure 18: The graph of actual speed vs the estimated speed using spatial averaging and template pairs filters. The error bars show standard deviation.

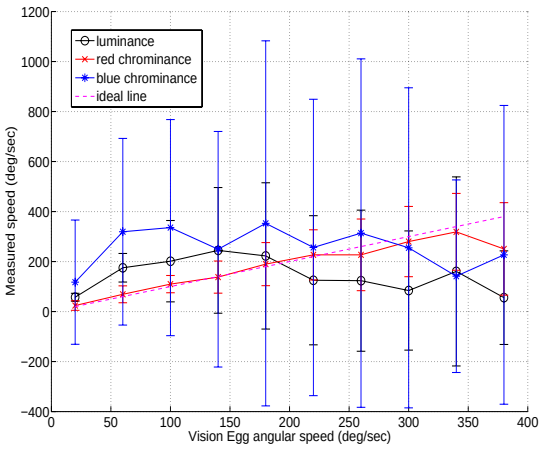

Figure 19: The graph of actual speed vs the estimated speed using mnc filters. The error bars show standard deviation.

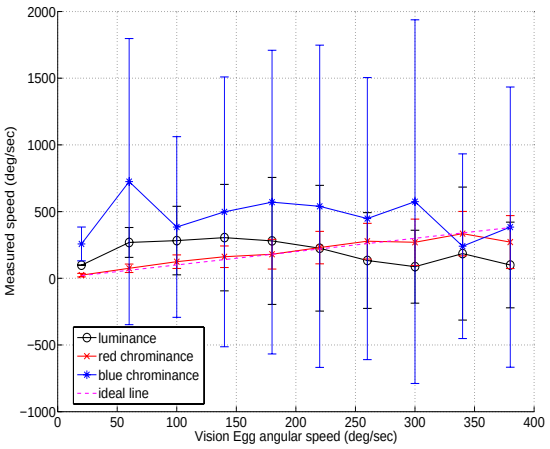

Figure 20: The graph of actual speed vs the estimated speed using mnc and template pairs filters. The error bars show standard deviation.

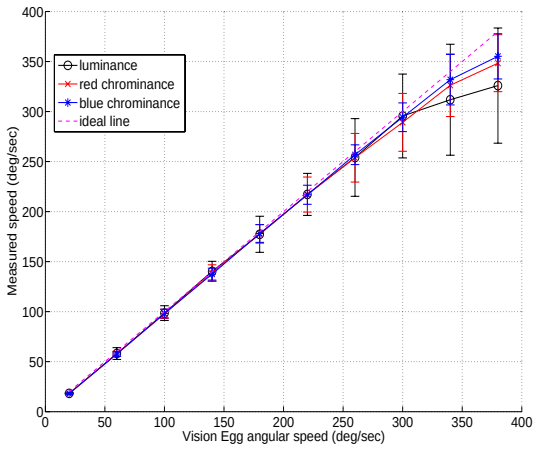

Figure 21: The graph of actual speed vs the estimated speed using template pairs filter. The error bars show standard deviation.

6. ESTIMATION OF TIME TO IMPACT

The estimation of time to impact for a looming object is an important aspect of collision avoidance. Many applications can be developed using the ability to estimate the time taken by an object to reach or hit the camera, including the use of detectors attached to the side mirror of cars to avoid and detect collisions. We decided to implement this using the template model and have started working on the algorithm. Although this is not yet complete, we describe the concept here to show the direction of future work. To calculate the estimated time to impact the following formula[21] is used,

$$t = \frac{\sin 2\theta}{2\dot{\theta}}$$

where t (seconds) represents the estimated time to impact of an object whereby θ (deg) is the angle at which the camera captures the object and $\dot{\theta}$ (deg/sec) is the rate of change of θ over time. From the software, we will be able to measure the change in number of pixels over time. Thus to find $\dot{\theta}$, we would need to know how many degrees a pixel captured by the camera represents. To find this conversion factor, we decided to perform calibration to measure this conversion factor. The camera was placed at a fixed distance of 16.7 cm away from the screen and it was exposed to a looming object on Vision Egg. The looming object increases it size at a certain speed in pixels/second. As the object increases in size, the camera effectively "sees" the object looming towards it. Then by simulating the object looming at different speeds, we measure the changes in width of the object captured by the camera over a time period of 30 seconds. With the knowledge that the camera captures images at 320 pixels wide, we can calculate the rate of change of pixels.

We then plot the rate of change of pixels (pixels/sec) against rate of change of angle (deg/sec) obtained by doing some simple geometry to find the change in viewing angle of the object over time. Figure 22 shows the calibration done to find the conversion factor.

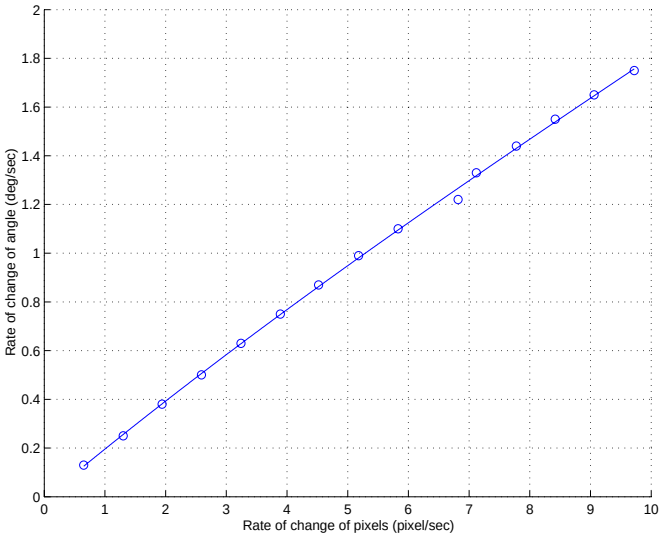

Figure 22: Calibration to find how many degrees each pixel captured by the camera represents.

From Figure 22, we can find the conversion factor by calculating the gradient of the line. The conversion has been found to be 1 pixel $\approx$ 0.18 degrees. Using this conversion factor, we can find $\dot{\theta}$ from the rate of change of pixels measured using the software. θ can be found from the number of pixels captured by the camera that represents the looming object. After finding both θ and $\dot{\theta}$, we can use the formula to find the time to impact.

7. CONCLUSIONS

Velocity response curves of the model correlator, the wide-field neuron of the hoverfly and the template model have been compared. Our implementation of the template model has produced velocity response curves that closely match the characteristics of the curves from the correlator model and this has been verified by experiments carried out on wide-field neurons of the hoverfly. Our results are also more conclusive than previous work. We can conclude that our implementation of the template model produces a reliable response and estimates velocities considerably well within certain limits.

We have also determined that automatically controlling the threshold for each frame brings the Template model closer to velocity constancy when looking at its performance for natural scenes with different contrasts. The high- pass spatial filtering did not bring the desired results although it could be considered for further investigation in the future with a more precise camera.

We also investigated the effect of different pre and post template filters on the accuracy of the speed estimation algorithm. We came to the conclusion that although the algorithm is already very accurate at lower velocities the spatial averaging could improve its performance slightly at higher velocities. MNC filter introduces too much noise and is not very useful in this setting, while the template pairs filter also does not improve the precision of the algorithm.

ACKNOWLEDGMENT

Funding from the School of Electrical and Electronic Engineering, The University of Adelaide, is gratefully acknowledged.

REFERENCES

1. "Electric Eye (research on honey bees)," *New Scientist* **171**, pp. 38, 2001.
2. http://www.innovations-report.de/html/berichte/biowissenschaften_chemie/bericht-7677.html
3. A. Budimir, S. Correll, S. Rajesh, D. Abbott, "Characterization of Insect Based Collision Avoidance Models Using a Video Camera," *Proc. SPIE, Smart Structures, Devices and Systems* **5649**, pp. 595-606, 2004.
4. B. Keipert, "Characterization of Insect Based Collision Avoidance Models Using a Video Camera," internal report, The University of Adelaide, 2003.
5. G. A. Horridge, "A Template Theory to Relate Visual Processing to Digital Circuitry," *Proceedings Royal Society of London B* **230**, pp. 279-292, 1990.
6. A. Moini, A. Bouzerdoum, A. Yakovlev, K. Eshraghian, "A two dimensional motion detector based on insect vision," *Advanced Focal Plane Arrays and Electronic Cameras*, Berlin, Germany, pp. 146-157, October 1996.
7. R. O. Dror, "Accuracy of visual velocity estimation by Reichardt correlators," Master's Thesis, University of Cambridge, Cambridge, UK, 1998.
8. S. Rajesh, D. C. O'Carroll and D. Abbott, "Man-made velocity estimators based on insect vision," *Smart Materials and Structures* **14**, pp. 413-424, 2004.
9. A. Yakovleff, X. T. Nguyen, A. Bouzerdou, A. Moini, R. E. Bogner and K. Eshraghian, "Dual-purpose interpretation of sensory information," *Proc. IEEE Int. Conf. on Robotics and Automation* **2**, , (Piscataway, NJ: IEEE) pp. 1635–40, San Diego, CA, 1994.
10. X. T. Nguyen, A. Bouzerdoum and A. Moini, "Velocity measurement using a smart micro-sensor," *Int. Symp. on Robotics and Cybernetics* **943**, pp 937–42, Lille, France, 1996.
11. D. L. Ruderman, "The statistics of natural images," *Network: Computation in Neural Systems* **5**, pp. 517-548, 1994
12. A. Straw, T. Rainsford, D. O'Carroll, "Estimates of natural scene velocity in fly motion detectors are contrast independent," *Current Biology*, submitted, 2005.
13. R. A. Harris, D. C. O'Carroll, S.B. Laughlin "Contrast gain reduction in fly motion adaptation," *Neuron* **28**, pp. 595-606, 2000.
14. P. A Shoemaker, D. C. O'Carroll, A. D. Straw, "Velocity constancy and models for wide-field visual motion detection in insects," in submission, 2005.
15. M. V. Srinivasan, S. B. Laughlin, A. Dubs, "Predictive coding: a fresh view of inhibition in the retina," *Proceedings Royal Society of London B* **216**, pp. 427-459, 1982.
16. H. Nguyen, S. Rajesh, D. Abbott, "Motion detection algorithms using template model," *Proc. SPIE, Electronics and Structures for MEMS II* **4591**, pp. 78-90, 2001.
17. A. Moini, A. Blanksby, A. Bouzerdoum, K. Eshraghian, R. Beare, "Multiplicative noise cancellation (MNC) for analog VLSI sensors," *ETD 2000, Electronic Technology Direction of the Year 2000,* pp. 253-257, 1995.
18. A. Moini, A. Bouzerdoum, A. Yakovleff, D. Abbot, O. Kim, K. Eshraghian, R. E. Bogner, "An analog implementation of early visual processing in insects," *Proceeding of the International Symposium of VLSI Technology, Systems and Applications,* pp 283-287*,* Taipei, Taiwan, 1993.
19. X. T. Nguyen, "Smart VLSI Micro-Sensors for Velocity Estimation inspired by Insect Vision," PhD Thesis*,* EEE Department, The University of Adelaide, 1995.
20. X. T. Nguyen, A. Bouzerdoum, R. E. Bogner, A. Moini, K. Eshraghian, D. Abbot, "The stair-step tracking algorithm for velocity estimation," *Proceedings of the Australian New Zealand Conference on Intelligent Information Systems*, Perth, Australia, 1993.
21. D. Abbott, A. Moini, A. Yakovleff, X. T. Nguyen, A. Blanksby, G. Kim, A. Bouzerdoum, R. E. Bogner and K. Eshraghian, "A new VLSI smart sensor for collision avoidance inspired by insect vision," *Proc. SPIE, Intelligent Vehicle Highway System* **2344**, pp. 105-115, Boston, 1994.
22. http://www.mathworks.com/
23. http://www.visionegg.org/