MOTION DETECTION ALGORITHMS USING TEMPLATE MODEL

Hung X. Nguyen, Sreeja Rajesh and Derek Abbott

Centre for Biomedical Engineering (CBME) and
Department of Electrical and Electronic Engineering, Adelaide University, SA 5005, Australia

ABSTRACT

The Horridge Template model is an empirical motion detection model inspired by insect vision. This model has been successfully implemented in several micro-sensor VLSI chips using grayscale pixels. The template model is based on movement of detected edges rather than the whole object, which consequently facilitates simple tracking techniques. Simple tracking algorithms developed by Nguyen[1] have been successful in tracking coherent movement of objects in a simple environment. An extension of the template model using color templates developed by Chin[2] has also been successful in tracking objects moving in close proximity to each other.

This paper introduces a low-cost insect vision prototype based on the use of a color CMOS camera. We implement a further extension to the above algorithms with error checking. Several error-checking schemes are used during template formation and manipulation to reduce noise and randomness. This enables the detection of moving objects in noisy environments, which may be applied to many real life situations.

Keywords: Motion detectors, Collision avoidance sensors, Artificial insect vision.

1. INTRODUCTION

It is really amazing to see how insects navigate with seemingly great ease and how they are quite adept in maneuvering their way through this complex and unpredictable world. Despite possessing a tiny brain that contains far fewer nerve cells than our own, insects are capable of performing a variety of tasks ranging from collision avoidance to posture maintenance. Recent research, observing freely flying insects reveal a number of computational shortcuts that the insects use in perceiving their visual world and navigating successfully in it -- this has helped us to implement simpler systems to detect motion.[3, 4]. Many insects seem to rely mainly on information provided by an array of motion-sensitive detectors in order to navigate[5, 6].

The study of the insect visual system has offered a solution to the computational bottleneck faced by conventional video-based visual systems, The bottleneck arises because algorithms that are written to detect edges, estimate motion flow fields, find discontinuities in depth, motion, color and intensity have to be iterated several hundreds, if not thousands, of times before a solution is reached. Furthermore, the amount of computer memory it takes and the computational cost is high. So the use of simple low-level motion cues, used by insects, inspired the "template model" proposed by Horridge [7,8].

The Horridge template model is a simplified model of the insect visual system. This empirical model compares the contrast between two adjacent receptors, at two sampling instances, to form a template. Templates can give simple directional information. Operations are performed mainly between adjacent receptors, which make this model easy to implement in a parallel architecture. Several generations of insect vision chips based on the template model have been developed by Moini *et al.*[9, 10,11]

When two different objects, converge together, crossover and then diverge again or when they move together at the same speed for a short time while their paths are crossing over, it was found that the use of chrominance information can assist in tracking the object more accurately than with simple luminance information. Consequently, Chin *et al.* extended the Horridge template model to incorporate color information. The resulting color templates were found to detect more moving edges than the grayscale model[2.]

In this paper, we further extend the color template algorithm, and we perform various error-checking algorithms to improve its performance and reduce noise. The reason for implementing de-noising techniques to color template formation is due to the likely conjecture that there will be some components of noise in the signal that will be uncorrelated between the color bands and some noise that will be correlated. Thus the use of three-color chrominance levels offers more degrees of freedom for implementing various noise reduction techniques. The noise is filtered before and after template formation of the template. The pre-filtering techniques used in this paper are spatial averaging and MNC (multiplicative noise cancellation). The post-filtering technique utilises a windowing technique and a search for special template-pair signatures. After experimenting with the combination of various methods, we have determined an optimum procedure that reduces noise considerably. This will help to enhance the detection of moving objects in noisy environments.

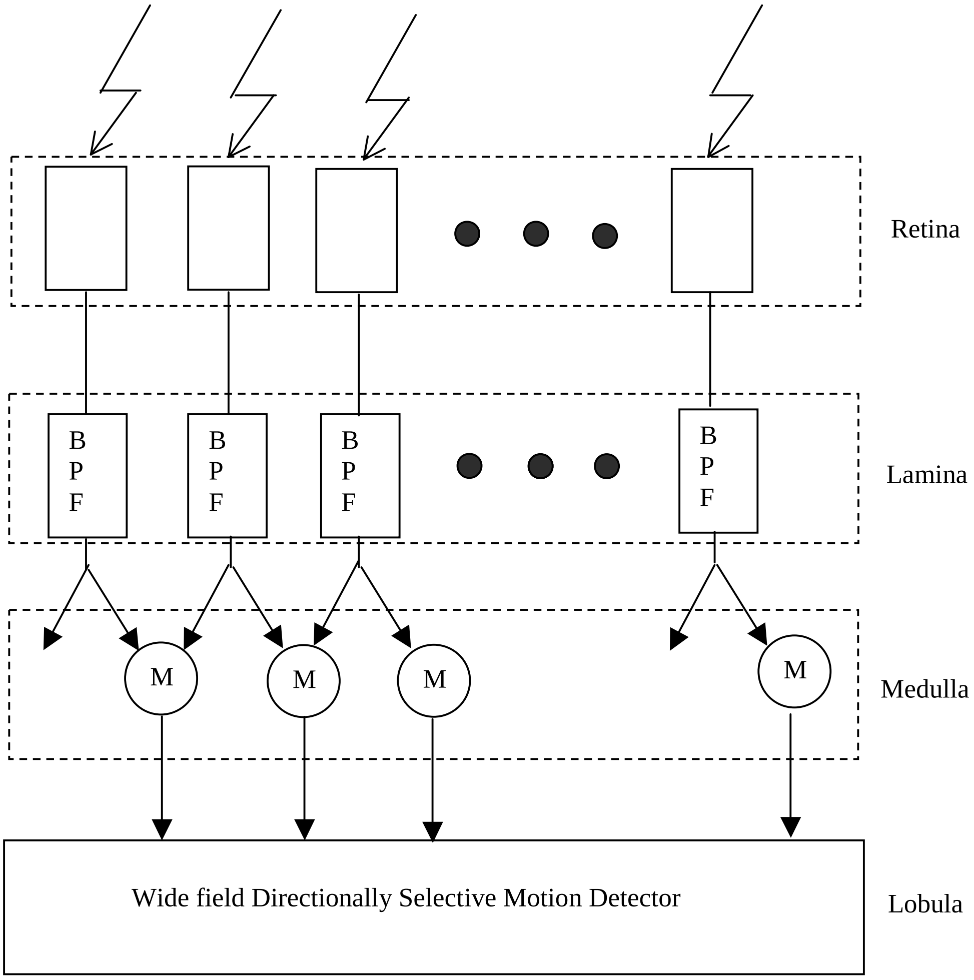


Figure 1. Simplified model of the insect visual system. The M blocks are small field motion detection elements.

The idea of using a commercial color CMOS camera, rather than a custom VLSI insect vision-based chip, is for fast turn-around prototyping and evaluation of the proposed algorithms. The drawback with the custom camera is it does not provide for parallel auto gain control (AGC) for every pixel, as exists for true insect vision. However, our preliminary results are valid for indoor applications under control lighting conditions, where parallelized AGC is obviously not necessary.

2. THE INSECT VISUAL SYSTEM

Insects compared to other living organisms, which range from human beings to semi-blind worms, have an intermediate visual system. The secret of the simplicity of the insect visual system is many tasks like perfect recognition of objects have either been eliminated or simplified. Moreover, in the visual system of insects, motion detection plays a predominant role.

The insect visual system can be classified into four layers, which include the *retina, lamina, medulla* and *lobula,* as shown in Figure 1. The retina of insects has a large array of photoreceptors that sample the visual field. The individual segments of the compound eye are called the *ommatidia*. The retina in our case is the CMOS camera.

The lamina consists of neurons, which control the dynamic range of the photoreceptors thus allowing the sensitivity of the retina to be adjusted for different conditions. Contrast enhancement is a vital function of the lamina and this is performed through biological band pass and high pass filters[12]. The lamina of the insects can be modeled as a temporal differentiator as was done our previous custom insect vision chips. But in our prototype, this was performed by simple frame differencing.

The medulla, which is one of the most complex layers of the insect visual system, contains many motion sensitive neurons. These neurons detect changes between the adjacent ommatidia and thus detect motion. These are the directional motion sensitive neurons, which perform small spatial field motion detection. The template model was inspired by the small field spatial detection neurons of the medulla.

The lobula receives input from the medulla to perform higher-level functions. The lobula also contains wide field directionally selective motion detection neurons, which are responsible for the detection of whole field or whole frame motion. However, at this point we adopt engineered techniques that are not directly inspired by the lobula.

3. THE TEMPLATE MODEL

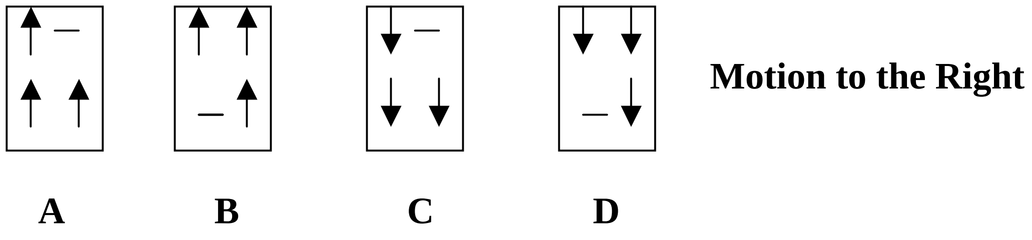
The template model proposed by Horridge, [7,8] models the function of the small field motion detection neurons in the medulla. In the template model the temporal contrast of the adjacent cells at consecutive instances are used to determine the direction of motion of an object. For simplicity, only temporal contrast at two instants and from two neighboring cells is considered.

The visual field is sampled spatially. Each sampling channel detects changes in light intensity at two consecutive sampling instances to show either an increase ↑, a decrease ↓ or a no change –. Adjacent photoreceptors carry out the detection of directional motion in insects and thus by copying this mechanism, the changed states are spatially combined between adjacent channels. Hence for a pair of adjacent sampling channels, there are 9 possible combinations:

(– –), (↑ –), (↑ ↑), (↑ ↓), (↓ –), (↓ ↓), (– ↓), (↓ ↑), (– ↑).

Since in any visual system directional motion is accomplished as a spatio-temporal operation, the temporal domain is included by associating the combinations obtained at two consecutive sampling times t_0 and t_1, thus yielding 81 spatio-temporal combinations or templates. Thus the temporal contrast of two neighboring cells, at two sampling instances, are combined to give simple motion measures called 'templates.'

Out of these 81 templates, there are only 8 templates, which indicate coherent motion. The templates sensitive to coherent motion are the ones in which one of the four entries is no change in intensity and the other three must identically either increase or decrease in intensity. These templates are known as *directionally motion sensitive templates* (DMSTs) [13]. These templates possess the 3:1 diagonally symmetric structure. The DMSTs are named using the first eight alphabetic characters, from 'A' to 'H'. The templates having one intensity change entry and three no change entry are purely generated by motion – they occur at the concave corners of the staircase motion trajectory[14]. In other words, they occur simultaneously in pairs with the DMSTs at each sampling instance. They indicate the current position of the moving object and hence are referred to as the *position conjugate templates (PCTs).* The PCTs indicate the location of their DMST counterparts, which occur at the next sampling instant. The pair of DMSTs and corresponding PCTs is called *conjugate pairs* of templates. The PCTs are named using the numeric character from '1' to '8'.



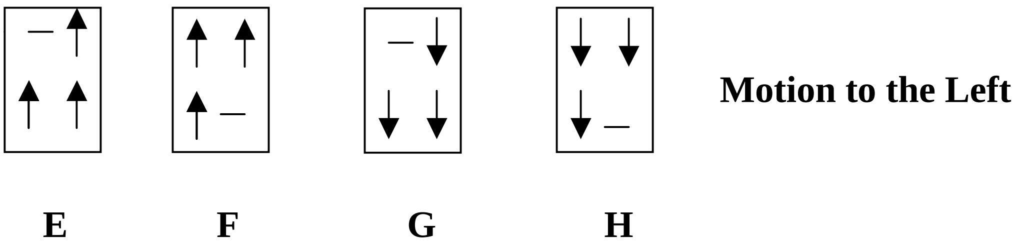


Figure 2. Directionally Motion Sensitive Templates

The template model is easily implemented in VLSI, because the thresholding and template formation operations are simple and can be done in parallel. The presence of only three states reduces the bandwidth requirements compared to other vision systems.

4. PROTOTYPING PLATFORM

A series of bugeye chips[9, 10,11] based on the template model has been implemented using grayscale photo detectors. In our previous custom insect vision chips, the retina was modeled using an array of on-chip photo detectors with dynamic range control circuitry. This is replaced by a CMOS camera in our prototype. It should be noted that the CMOS camera does not have Automatic Gain Control (AGC) circuitry at the pixel level, but this feature is not essential for indoor controlled lighting conditions.

The filtering operation in the lamina layer was implemented using temporal differentiation in the custom chips; whilst in our CMOS camera prototype this is performed using simple frame differencing. The CMOS camera is interfaced to a personal computer and the captured image is stored in the buffers. The difference between the current image in the current frame buffer and the previous image in the previous frame buffer will give changes in the motion, which are thresholded and stored as templates. Template generation here is performed in software whereas some custom chips have template generation elements in the hardware. Thus by comparing the templates obtained from frame differencing with directionally sensitive motion templates, the direction of motion of the moving object can be obtained. The time step size between the frames or the frame interval can be adjusted. But it is found that by keeping the frame interval low, more frames can be captured giving us more template information. When the frame interval is high, a lot of frames and information is lost. In our case, we have chosen a low frame interval of 60 ms as it was found to give the best results.

5. COLOUR TEMPLATES

The CMOS camera uses the RGB color scheme. The RGB color scheme works on the principle that each color is comprised of different combinations of the three primary colors red, green and blue. The RGB scheme can be converted into the luminance-chrominance model using the formulae below:

$$Y = 0.299R + 0.587G + 0.114B$$

$$Cr = R - Y$$

$$Cb = B - Y$$

$$Cg = G - Y$$

Where, Y is called the Luminance, Cr is the Red Chrominance, Cb is the Blue Chrominance and Cg is the Green Chrominance. R is the Red, G is the Green and B is the Blue value of an RGB pixel.

The luminance and the chrominance values of each pixel or channel are put through the same differentiation, thresholding and template generation stages as the original grayscale templates. Through the chrominance values, color templates can be formed. The same eight directionally motion sensitive elements can be detected for the luminance and each of the three-color chrominance. Since it is thought that some components of noise will be uncorrelated between color bands and some noise will be correlated, the use of the three chrominance levels offers a better platform for implementing various noise reduction techniques.

6. PRE-TEMPLATE FILTERING

The signal captured by the CMOS camera is very noisy. Noise does not just come from the environment but it also originates from the camera when the signal is converted from analog mode to digital mode. Here we implement two algorithms to pre-filter noise before the signal is passed to the template formation stage. The first algorithm, using the spatial averaging method, is a very simple algorithm that helps to smooth the response and increase the sensitivity of the program without compromising speed. The second algorithm is Multiplicative Noise Cancellation (MNC), which was introduced by Moini *et al.*[11].

6.1 Spatial Averaging

Spatial smoothing of the input signal is one of the filtering operations done by many front-end processing stages for reducing the input noise. Input signal captured by the CMOS camera is stored in a buffer. Before this data is passed to the differentiation stage the values stored in the buffer will then be averaged. Spatial averaging is done by taking the average over certain number of channels. The resulting values will then be stored in the original buffer replacing the previous values. Averaging helps smoothing the edges of the signal hence improving the template formation as shown in Figure 3. Further more averaging also helps in detecting over a given range of channel instead of just between two adjacent channels. In our software, we implement spatial average on 2 and 3 channels. Averaging on 4 channels or more is not efficient because of the blurring effect of the camera when the object is moving so fast.

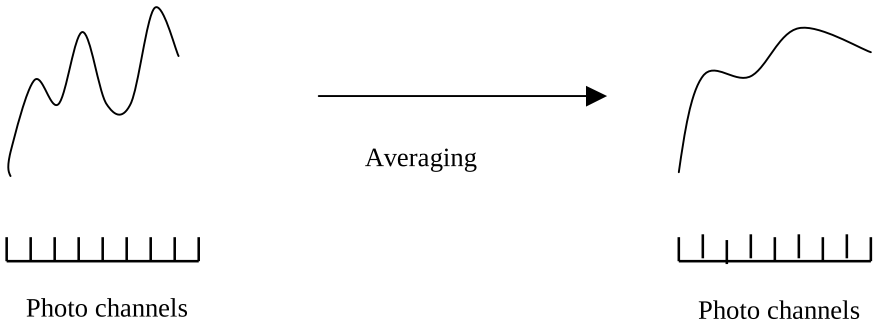


Figure 3. Averaging over 2 channels helps smoothing the response.

6.2 Multiplicative noise cancellation

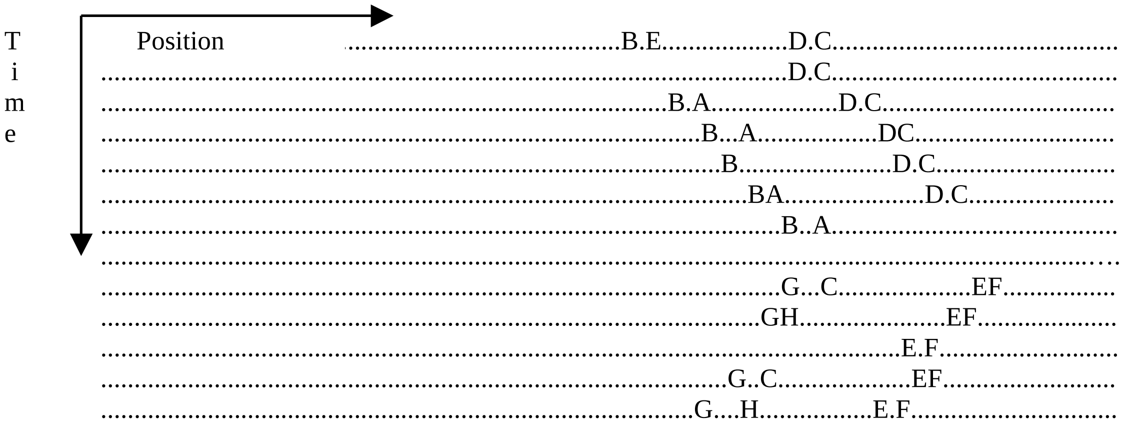
MNC is an extension of spatial averaging which also is applied to the input signal before the differentiation stage. Experimental results indicated that the 100 or 120 Hz frequency component of light sources, operating with 50 or 60 Hz mains power, severely affect the detection of motion because of the temporal differentiation stage. In this stage noise is sometimes amplified to a point where it dominates the signals induced by the motion of objects. As this noise is usually common among several neighboring photo detectors and has a multiplicative nature, spatial averaging is ineffective, but dividing the signal in one channel by the spatial average of the input signals effectively reduces the noise [15]. The MNC consists of spatial averaging, and division-by-average stages. MNC has been successfully implemented in our previous custom chips. In our experiment we attempted to replicate MNC using the CMOS camera to improve the noise performance of the system. Our software was designed such that the averages over 3, 5 or 7 channels can be selected.

6.3 Experiment results

In this experiment, we generated a test sequence by moving a dark object at a close range in front of the camera. Our program was then used to capture the motion of the object at frame interval of 60 ms. In Figure 4; we tried to replicate the result of the custom chips in the grayscale domain without applying any pre-filtering technique. We then applied spatial averaging and MNC algorithms to the same test sequence. Spatial averaging results are shown in Figure 5, and MNC results are shown in Figure 6.

The results show that the spatial averaging algorithm provides a better result than the one obtained without using any de-noising technique. The MNC technique is also found to be successful in removing the unwanted AC noise. However the MNC technique seemed to introduce some noise of its own into the system. This is due to the fact that division operation using digital computer is not optimized to have the level of precision needed. The errors that are created in this operation are carried on to the template formation stage and then later might be filtered out by the post-template filtering methods.

Figure 4. Grayscale templates for small object moving across out prototype camera at close range. Note that only 8 directionally motion sensitive templates ('A' to 'G') have been included in this result. The other 73 templates were represented by '.'.



```
..........................................B.E..................D.C.............................................
............................................................................................................D.C.............................................
.........................................................................................B.A....................D.C....................................
..............................................................................................B...A...................DC...............................
.................................................................................................B........................D.C............................
.....................................................................................................BA......................D.C.......................
..........................................................................................................B..A.............................................
................................................................................................................................................................
..........................................................................................................G...C.....................EF..................
.......................................................................................................GH.......................EF......................
..............................................................................................................................E.F............................
..................................................................................................G..C.....................EF...........................
............................................................................................G....H..................E.F................................
```

Figure 5. Templates of an object moving at close range applying spatial averaging to the signal. Here we chose to use average over 2 channels.

```
.........B.A...................................D................................................................................................
..........................B.A.............................................D.C..................................................................
..................................................B...A.............................................D..C.....................................
...............................................................................B.A..................................DC.........................
......................................................................................................B.A.............E.................D.C...
......................................................................................................................................................
......................................................................................................G.H..........................E.F...........
...............................................................................G.H............................E.F...............................
.............................................................G.H...................................................................................
.............................................G.H.....................................E.F...D.................................................
..........................G.H...............................................E.F..................................................................
........G.H......................................E.F...........................................................................................
```

Figure 6. Templates of an object moving at close range applying MNC to the signal. Here we chose to use average over 5 channels.

```
...........D.A...........................D.........................................................
..................A.................................................................................
...............................B......................C.............................................
..........................................A.E.....................C................................
.....................................................A...............C.............................
.................................................................B..E..............................
...................................................................................................
...................................GA...............................G.E.............................
...........................H.......................................................................
...................G...............................................F...............................
......................................E............................................................
............H........H..............................................................................
.....G..C.............F............................................................................
```

Though the pre-filtering technique filters much of the noise, it does not remove all the noise that is present. These remaining errors are carried over during the template formation and the post filtering methods aim at filtering them. We now introduce two algorithms for filtering out noise from the data, provided in real-time by the templates formation stage. Both algorithms work based on the principle that certain templates occur together in coherent motion, but differ in that the first is based on the continuity of the motion, while the second uses the association of certain template pairs to detect noise.

7.1 Windows Operation

In coherent motion, where the object is moving with constant speed, a moving edge causes directionally sensitive templates to occur close together in space and time. Experimental and theoretical results [14,16] have proven that motion sensitive templates occur in groups. Further experiments using our camera show that motion templates occur in a group of at least 2 templates. This result indicates that isolated templates are likely to be noise.

To utilize this observation, we apply a window operation to the templates in both the time and space dimensions. We examine each 3x3 window of templates (3 in the time dimension and 3 in the spatial dimension). If there are less than 2 templates indicating motion of one kind (either to the left or to the right) in that window then these templates are considered as noise and are thrown away.

The window size might vary depending on the speed of the object. Faster objects that can jump passed several pixels in one period of time might require a bigger window size. Our program was implemented such that different window sizes can be chosen externally.

7.2 Template pairs

Further results from Nguyen[1] demonstrated that motion templates occur in pairs. There are two kinds of template pairs. The first one is the “motion conjugate” pair, where DMSTs and PCTs occur together in any coherent motion. Furthermore, theoretical results from Nguyen[14] have shown that a moving edge always generates 2 associated DMSTs ((A, B), (C, D)…). These pairs are associated because of their nature of one corresponding to the edge entering a region and the other corresponding to the edge leaving that region. Our template-pair algorithm is based on these results. In this algorithm, we look at each row of templates and search for the templates pairs. If a template occurs at one instant but its partner template (DMST or PCT) does not occur at that same instant, then this template is treated as noise and is thrown away.

7.3 Experiment results

We carried out several experiments to evaluate the effectiveness of the two algorithms. The results from windowing algorithm and the template pair algorithm are shown in Figures 7 and 8 respectively.

Figure 7. Applying the windowing algorithm to the templates

```
.........B.A............................………...........................................................................
.........................B.A.......................…………D.C...........................................................…
...............................................B.A....................................D.C…...........................…......
....................................................C.....................B.A................................DC.....................
.......................................................................…………B.A............................D.C.....
.........................................................................................................................................
…………………………………………………………G.H.......................E.F.......…
..........................................................................G.H..........................E.F..............................
.........................................................G.H...........................................D.................................
............................................G.H...................................E.F.................................…………
.........................G.H...........................................E.F.............................................................
……G.H.....................................E.F…………………………………………………
```

Figure 8. Applying the template-pair algorithm to the output

```
.........B.A.................................D..........................................................................................
.........................B.A.......................…………D.C...........................................................…
...............................................B...A.........................................D.C...............................…....
..........................................................................B.A...............................DC.......................
.......................................................................…………B.A............................D.C.....
.........................................................................................................................................
…………………………………………………………G.H.........................E.F.......…
..........................................................................G.H..........................E.F..............................
.........................................................G.H....................................................................................
............................................G.H...................................E.F.................................…………
.........................G.H...........................................E.F..............................................................
……G.H.....................................E.F…………………………………………………
```

.

In terms of performance, the windowing algorithm produces less consistent results compared to the template-pair algorithm. The windowing algorithm seems to leave noise in the result wherever the noisy templates are surrounded by other templates. The conjugate-pair approach does not have this problem because it does not just look at the templates locally but looks at templates in a whole row. However, for fast moving objects, where flicker noise[18] is introduced, template pairs normally do not occur correctly. In these cases the windowing algorithm, with appropriate size windows, provides better performance.

8. DE-NOISING THE COLOR TEMPLATES

Color templates were introduced by Chin *et al.*[2].In this experiment, we applied the de-noising techniques to color templates. Here, color templates were created, by extracting color information from the CMOS camera. In order to reduce the processor load, here we only implement blue and red templates (we can get red and blue colors straight from the camera). However using the formula in Section 5 we can easily create green templates. Figure 9 shows color templates without any de-noising operations.

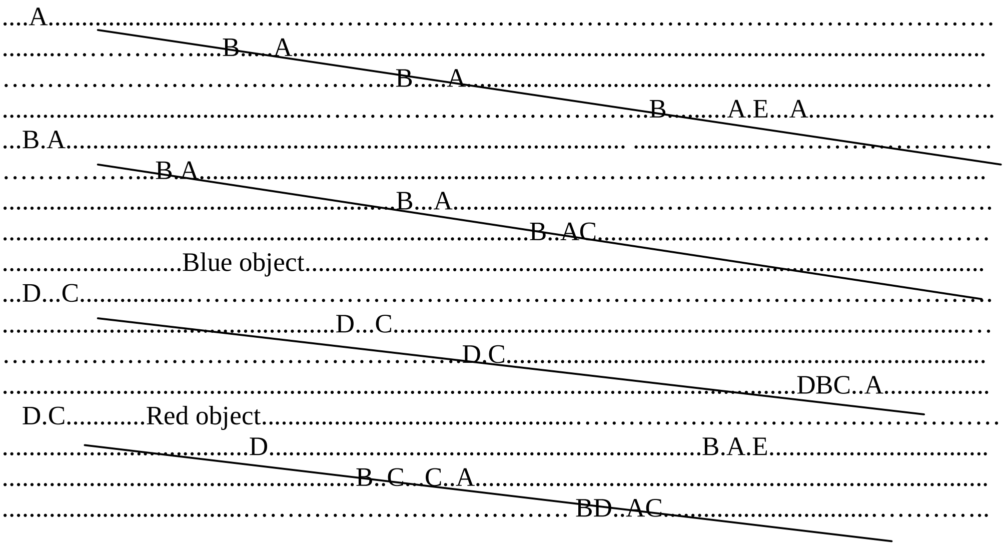


Figure 9. Color templates from the multi object test sequence from our prototype. The sequence contains 1 red and 1 blue object. The objects are moving across the camera next to each other with relatively the same speed.

Applying spatial averaging over 2 channels and conjugate pair algorithms to the above result helped get rid of most of the noise as shown in Figure 10.

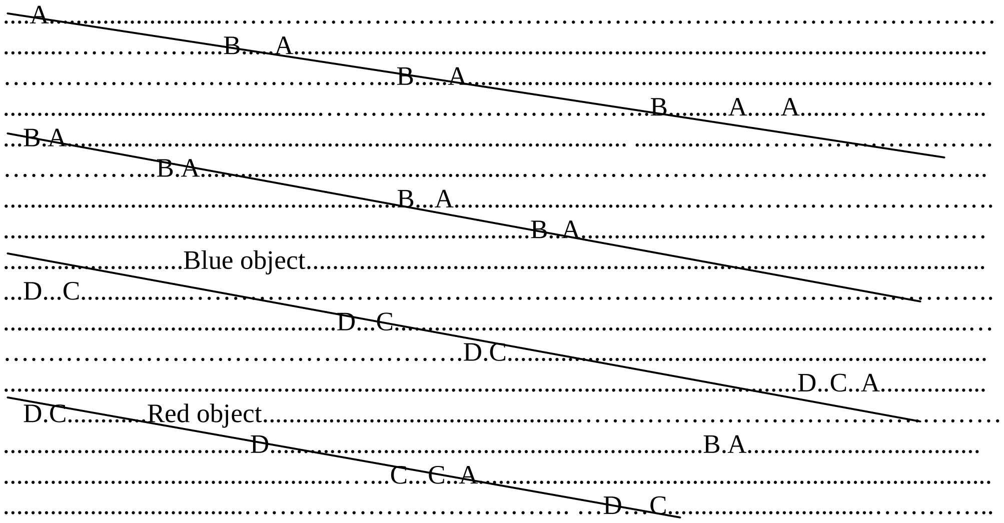


Figure 10. Colour templates after de-noising.

9. MEASURING ANGULAR VELOCITY

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 dark 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. This experiment is done using only the luminance templates.

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 [18]. 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 11, the angular velocity is angular displacement/ΔT).

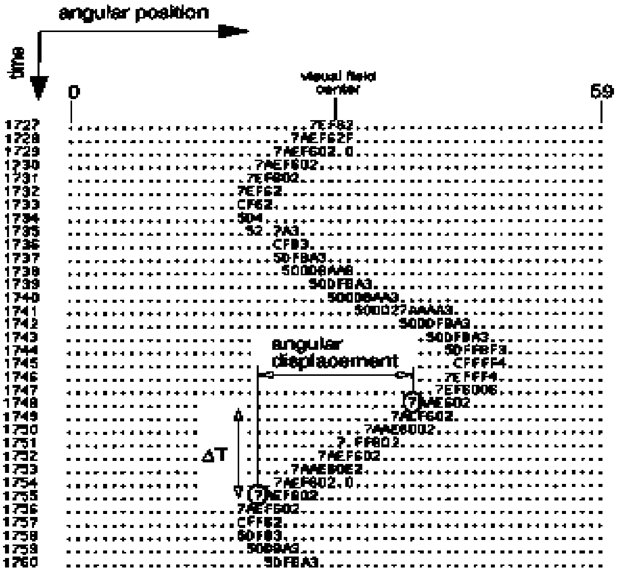


Figure 11.

Template response (dark object in front of light background). After Yakovleff et al [19]

There are two algorithms for estimating velocity in real-time that have been developed and tested. The first algorithm is forward tracking [19], and the second algorithm is stair-step tracking [14]. In our experiment, we used the forward tracking algorithm for velocity measurement. In the forward tracking algorithm, we kept track of certain motion templates 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.

Results from our experiments are shown in Figures 12.1 to 12.9. 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.

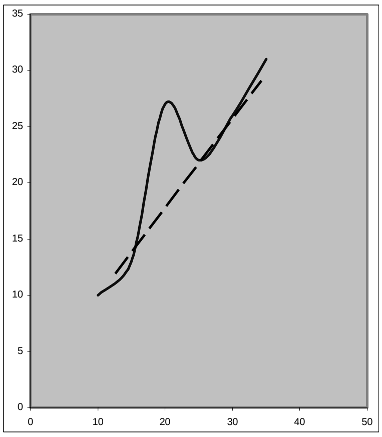


Figure 12.1. Without noise removal

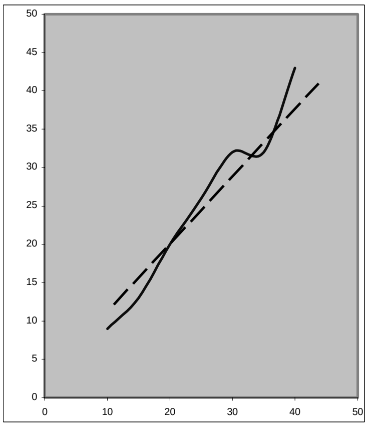


Figure 12.2. Using the spatial average method

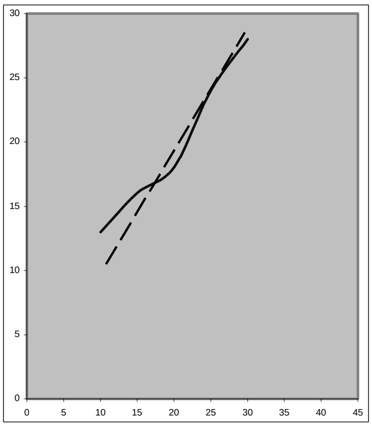


Figure 12.3. Using the MNC method

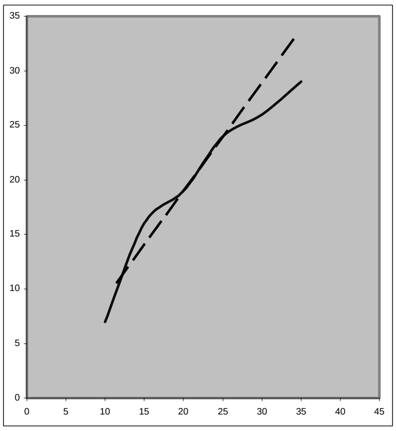


Figure 12.4. Using the windowing method

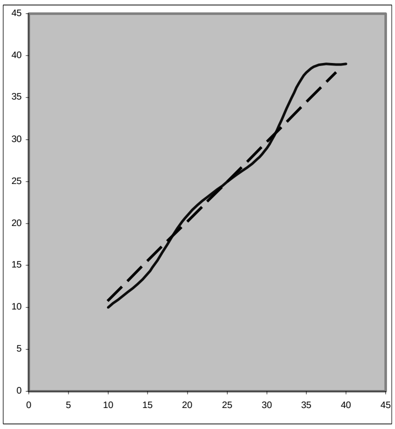


Figure 12.5. Using the template-pair method

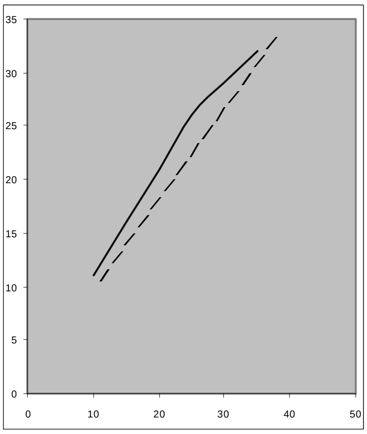
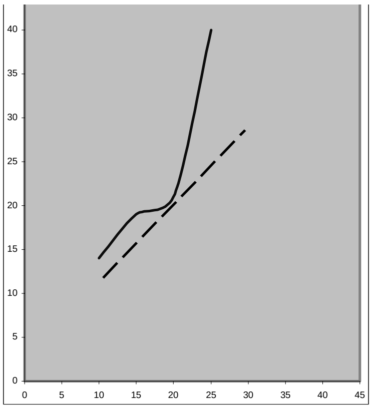
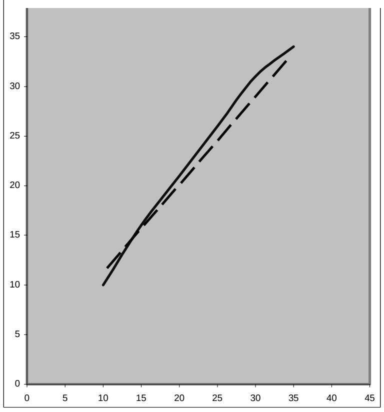


Figure 12.6. Using the averaging method with windows





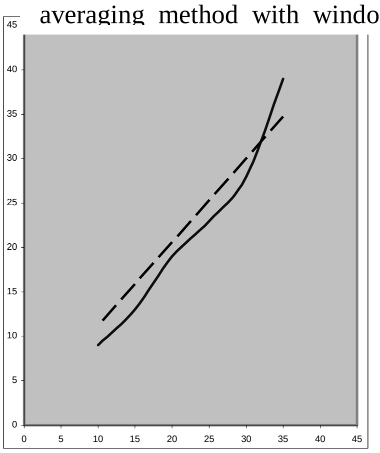


Figure 12.7. Using the MNC method with windowing

Figure 12.8. Using the spatial average method and the template -pair method

Figure 12.9. Using the MNC method and the template-pair

——— Real response.

— — — Ideal response.

Figure 12.1 shows that without applying any filtering process the response scatters randomly around the ideal response. While results from figure 12.2 show that the averaging process helps in smoothing the response. The response still scatters around the ideal response but is much smoother. Figure 12.3 indicates that the measured speeds are shifting up from the ideal response. This can be explained by the calculation errors (division) of our digital computer. Figure 12.4 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.

Figures 12.6, 12.7, 12.8 and 12.9 represent responses with different combinations of pre-template filtering and post-template filtering. These figures indicate big 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 our experiments we can see 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.

10. CONCLUSIONS

We have provided a brief introduction to the fundamental concepts of template model and how they apply to the real time motion detection. We also introduced some de-noising techniques before and after the template formation stage. Several experiments have been done and preliminary results from these experiments indicate certain improvement to the template model. Furthermore it also demonstrates that combining pre-template filter and post-template filter does provide better performance as one can expect.

The use of color templates is believed to have helped in these de-noising techniques as it provides a more independent platform for noise reduction. Further work needs to be done before any conclusions can be made about the color templates. The next challenge would be to consider extending the color template idea to colors other than the three primary colors, red, blue and green by extending the three dimensions field to n dimensions to get multicolor information, and to try applying other de-noising techniques.

ACKNOWLEDGEMENTS

Funding from the Sir Ross & Sir Keith Smith Fund and Australian Research Council is gratefully acknowledged. We would like to thank Leonard Hall, EEE Dept., Adelaide University, for his assistance.

REFERENCES

1. X.T.Nguyen, Smart *VLSI Micro-Sensors for Velocity Estimation Inspired by Insect Vision, Ph.D.* Thesis, EEE Dept., University of Adelaide, 1995.
2. K.Chin and D.Abbott, “Motion Detection Using Colour Templates,” *Proc. SPIE*, Vol**. 3893**, pp. 314-321, 1999.
3. A.Yakovleff, A.Moini, A.Bouzerdoum, X.T.Nguyen, R.E.Bogner, K.Eshraghian and D.Abbott, “A micro-sensor based on insect vision”, *Proc. Computer Architecture for Machine Perception Workshop (CAMP’93),* New Orleans, USA, pp.137-146, 1993.
4. D.Abbott, A.Moini, A.Yakovleff, X.T. Nguyen, A.Blanksby, G.Kim, R.E.Bogner and K.Eshraghian, “A new VLSI smart sensor for collision avoidance inspired by insect vision,” *Proc. SPIE, Intelligent Vehicle Highway Systems*, Boston, **Vol. 2344**, pp. 105-115, 1994.
5. M.Lehrer, M.V.Srinivasan, S.W.Zhang and G.A.Horridge, “Motion cues provide the bee’s visual world with a three dimension,” *Nature,* **Vol. 332**, pp.356-357, 1998.
6. R.Beare, A.Blanksby, A.Bouuzerdoum, “Low Level Visual Motion Processing Using Local Motion Detectors,*” Proc. IEEE International Conference on Neural Networks,* **Vol. 1**, pp.1-6, Perth, Australia, 1995
7. G.A. Horridge, “A template theory to relate visual processing,” *Proc. of the Royal Society of London B,* **Vol. 239**, pp. 17-33, 1990.
8. P.Sobey, “Determining range information from self motion – the template model,” *Proc. SPIE,* **Vol. 1382,** Intelligent Robots and Computer Vision, pp. 123-131, 1990.
9. A.Moini, A.Bouzerdoum, A.Yakovleff and K.Eshraghian, “A two dimensional motion detector based on insect vision,” *Advanced Focal Plane Arrays and Electronic Cameras*, Berlin, Germany, 1996.
10. A.Moini and A.Bouzerdoum, “A Biologically motivated Imager and Motion Detector with Pixel Level Image Processing,” Proc. Australian Microelectronics Conference, Melbourne, 1997.
11. A.Moini, A.Bouzerdoum, K.Eshraghian, A.Yakovleff, X.T.Nguyen, A.Blanksby, R.Beare, D.Abbott and R.E.Bogner, “An Insect Vision Based Motion Detection Chip,” IEEE *Journal of Solid State Circuits*, **Vol.32**, No.2, 1997.
12. R.Beare and A.Bouzerdoum, “A simple model of the sustained unit in the insect lamina,” *Australian and New Zealand Conference on Intelligent Information Systems*, Adelaide, 1996.
13. A.Yakovleff, A.Moini, “Motion perception using analog VLSI,” Journal of Analog Integrated Circuits and Signal Processing 2, pp. 1-22, 1997.
14. X.T.Nguyen, A.Bouzerdoum, R.E. Bogner, A.Moini, K.Eshraghian and D.Abbott, “The stair-step tracking algorithm for velocity estimation,” Proc. Australian New Zealand Conference on Intelligent Information Systems, pp. 412-416, Perth, Australia, December 1993.

15. A.Moini, A.Bouzerdoum, A.Yakovleff, D.Abbott, O.Kim, K.Eshraghian, and R.E.Bogner, “An analog implementation of early visual processing in insects,” *Proc. 1993 Int. Symp. VLSI Technology, Systems, and Applications*, Taipei, Taiwan, May 1993, pp. 283–287.
16. X.T.Nguyen, A.Bouzerdoum, R.E.Bogner, A.Moini and K.Eshraghian, “Feature Representation of motion Trajectories,” *Proc. IEEE Int. Conf. Neural Networks*, **Vol. 6**, pp. 2922-2927, Perth, Australia, 27 November-1 December.
17. G.A. Horridge, “A template theory to relate visual processing to digital to digital circuitry,” *Proc R. Soc. London. B*, **Vol. 230,** pp. 279-292, 1990.
18. A. Yakovleff, A. Moini, A. Bouzerdoum, X. T. Nguyen, R. E. Bogner and K. Eshraghian, “Dual purpose interpretation of sensory information,” *Proc. IEEE International Conference on Robotics and Automation*, San Diego, California, U.S.A., **Vol. 2,** pp 1635 –1640.
19. A. Yakovleff, D.Abbott, X. T. Nguyen and K. Eshraghian, “Obstacle avoidance and motion-induced navigation,” *Proc. Computer Architectures for Machine Perception Workshop* (CAMP’95), Como, Italy, 18-20 September 1995, pp 384-393.