Obstacle Avoidance and Motion-Induced Navigation

A. Yakovleff*, D. Abbott, X.T. Nguyen & K. Eshraghian

Centre for GaAs VLSI Technology and Department of Electrical & Electronic Engineering,
the University of Adelaide, SA 5005, Australia
(email: andre@itd.dsto.gov.au)

Abstract

In nature, the visual detection of motion appears to be used in a variety of tasks, ranging from collision avoidance to posture maintenance. Many insects seem to rely primarily on information provided by an array of elementary movement detectors in order to navigate. Moreover, experimental evidence suggests that motion information is interpreted at an early stage of the insect visual system, and may be closely linked to motor control.

A motion detector, whose design is based on some of the characteristics of the insect visual system, has been implemented on a single VLSI chip. This paper shows the manner in which motion information, provided by the chip in real-time, may be utilised by the control system of an autonomous vehicle in low-level perceptual tasks.

Key words - Motion perception, navigation, insect vision, VLSI, sensing, micro-sensor.

1: Introduction

Motion perception is of critical importance for a wide variety of tasks involving quite different behavioural aspects, such as collision avoidance and posture maintenance. The detection of motion thus appears to play a fundamental role, firstly as a provider of cues from which structural information about the environment is obtained, and secondly as a stimulus affecting behaviour (see [1] for a comprehensive review).

While some species benefit from stereopsis to perceive depth, other species who do not have binocular vision are nonetheless able to navigate efficiently in a three-dimensional environment. A monocular cue, motion parallax, provides information about the position of a moving observer relative to objects or surfaces, and is most notably used by some insects [2]. Also, in order to distinguish objects from the background, insects appear to use the velocity differences on the retinal image induced by peering motion [3].

In spite of being equipped with a relatively limited visual system, insects such as bees and flies are remarkably adept at controlling their flight paths [4], thus lending support to the feasibility of linking the motor control of an autonomous vehicle to the interpretation of visual motion information.

A VLSI motion detector based on the insect visual system is briefly described and experimental results are presented. While other workers have considered various schemes based on insect vision (see [5][6][7][8] for instance), this work represents a world first VLSI single-chip smart sensor solution. Two algorithms for estimating relative angular velocity from the chip response are introduced, and possible ways in which velocity information may be further interpreted are shown. Finally, the manner in which motion information may be used by the control of an autonomous vehicle for navigational purposes are discussed.

2: System overview

We have developed a VLSI chip, dubbed the "bugeye", which detects and interprets changes in contrast in order to provide directional motion information in real-time. The design is based on the insect visual system, and implements the "template model", a motion detection scheme proposed in [9].

As illustrated in Figure 1, the scheme consists of combining spatially and temporally the responses of light detectors to changes in contrast. In the presence of motion, which is detected in a single spatial dimension, some of

the combinations, hereafter referred to as "templates", indicate both the direction of motion and the polarity, i.e., the sign of the change in contrast, which is bright-to-dark in Figure 1.
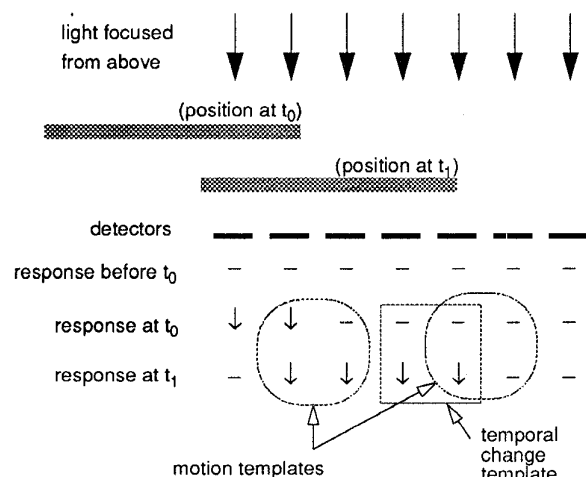

Figure 1 Example of template formation

The architecture of the bugeye has been described previously in [10]. Briefly, the chip comprises a linear array of 61 photo-receptors, onto which light is focused through a cylindrical gradient index lens whose focal plane coincides with the chip surface (Figure 2). As the separation between adjacent photo-receptors corresponds to one degree of visual angle, the total aperture of the bugeye is 60 degrees, although the lens aperture itself is 72 degrees.
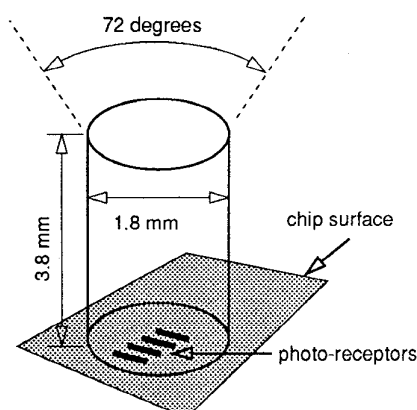

Figure 2 Gradient index lens characteristics

The photo-receptor output voltages are sampled at a rate of approximately 10 milliseconds, and fed into an array of analog contrast change detectors which compare the present values to the previous. The change detector outputs are then thresholded, providing digital signals which indicate whether the corresponding contrast has increased, decreased, or remained constant (see [11] for a description of the analog circuitry). The digital signals are stored in order to form the templates described previously, by combining adjacent responses at consecutive sampling instants.

At each sampling time, 60 templates are thus formed in parallel. The templates are then multiplexed and encoded serially by the template memory into 4-bit quantities, and stored into the save memory. Concurrently, an on-chip processor detects the occurrences of pre-defined target templates and tracks their displacements from one sampling instant to the next. The design of the original processor, however, has been superseded by new tracking algorithms based on experimental results (see Section 4), and hence, the on-chip tracking processor will not be used subsequently. Figure 3 depicts the internal architecture of the chip, which was fabricated using a standard double polysilicon, double metal, p-well 2μ CMOS process.

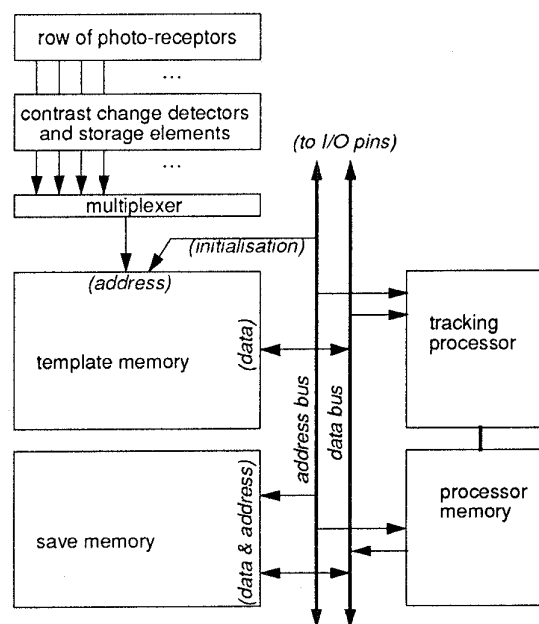**Figure 3 Internal architecture of the bugeye**

The templates, which are stored sequentially in the save memory, can be accessed externally in between sampling instants. A memory address thus corresponds to the angular position of a template, i.e., the contents of the first address pertain to the template formed from the responses of the first two detectors, the second address pertains to the

responses of the second and third detectors, and so forth. Note that the 4-bit template encoding is defined by the contents of the template memory, and may be changed at will.

3: Experimental results

For experimental purposes, the bugeye is interfaced to a computer which controls the sampling signal as well as memory accesses. Since the computer is operating at several mega-Hertz, reading and displaying or processing data obtained from the bugeye can be accomplished in between sampling instants, thus in real-time.

The response of the bugeye to the motion of a dark object on a lighter background is illustrated in Figure 4. In between sampling instants, templates are read sequentially from the save memory and printed on a single line as hexadecimal numbers (except for the "no motion" template, which is printed as a dot). The vertical and horizontal axes therefore represent sampling instants and angular position respectively, and hence the response is shown as a spatio-temporal image. The template encoding is arbitrary. Thus, templates 7 and E, for instance, only occur when the motion is to the left and there is a bright-to-dark contrast change, while templates 3 and B indicate motion to the right.
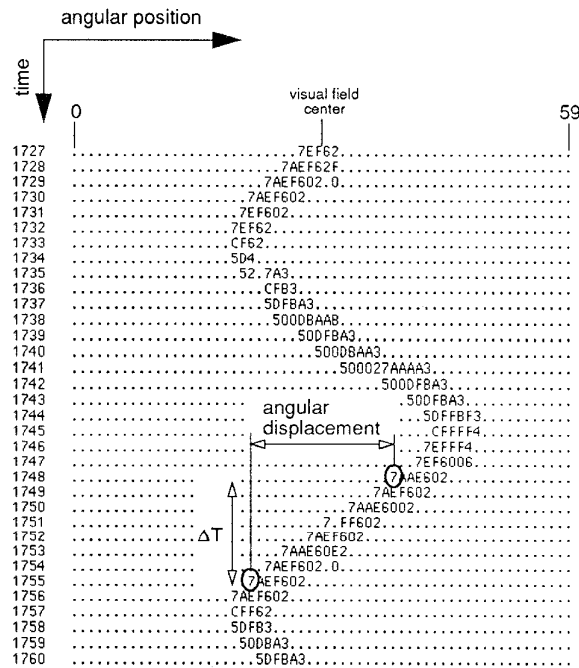
Experimentation has shown that the bugeye can detect motion up to approximately 8 metres away, and under lighting conditions ranging from dimly lit to normal daylight.

4: Velocity estimation

The experimental results show that a moving object (or edge) consistently causes the same motion sensitive templates to occur at subsequent time steps, and at positions corresponding to the displacement of the edge relative to the detector. Notwithstanding a correction factor due to the optical aberrations of the lens, 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 4, the angular velocity is angular displacement/ΔT).

Two algorithms for estimating velocity in real-time have been developed and tested. The first algorithm, forward tracking [12], is based on the premise that if a moving edge causes a motion sensitive template to occur at a certain position, then the same edge will cause the same template to eventually occur nearby, i.e., a few positions further within a maximum range, and in the direction indicated by the template. Therefore, by keeping track 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.

Figure 5 depicts an example of the template response to an object moving to the right. Two motion sensitive template targets, 3 and 5, are detected first at time steps 153 and 156, respectively. The current position of each target, and the sum of that target's previous displacements within the tracking window, are shown at each time step.
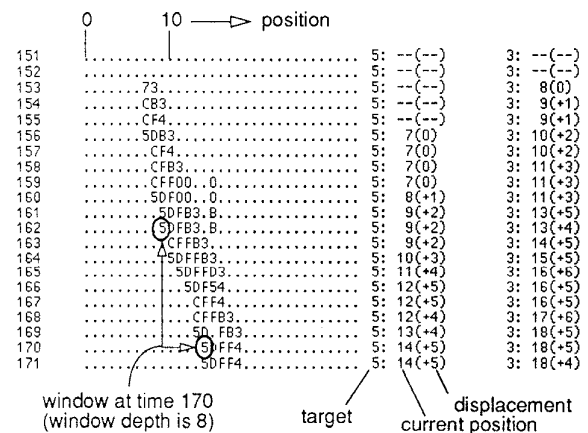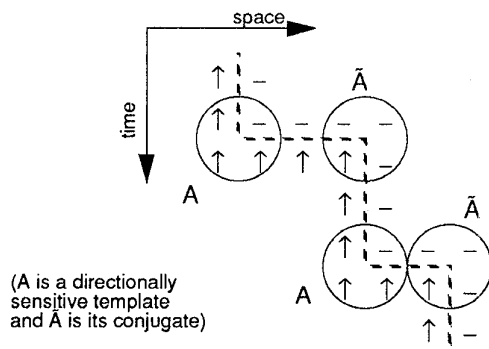

Figure 4 Template response (dark object in front of light background)

Figure 5 Forward tracking example

This scheme implies that a track may be lost due to the velocity of a moving edge being greater than the limit given by the maximum range, divided by the sampling period. It should be pointed out, however, that biological visual systems appear to make a similar assumption [13]. In other words, motion is not considered to be coherent if the "interframe" displacement of a moving object is beyond a certain limit.

The second algorithm, proposed in [14], is based on the observation that motion sensitive templates occur in pairs which can be classified as "motion conjugate". These templates should, in principle, occur simultaneously, as illustrated in Figure 6a, where a bright edge is moving to the right. In the example of Figure 6b, the conjugate templates of motion sensitive templates E and B are 7 and 3, respectively.

(a) Formation of motion conjugate pairs

(A is a directionally
sensitive template
and Ā is its conjugate)

(b) Example

Figure 6 Motion conjugate templates

The search algorithm thus consists of analysing the previous history of template responses contained in a "first-in, first-out" (FIFO) memory of fixed depth. After each sampling pulse, the latest template responses are stored in a row containing sixty 4-bit locations, and the earliest row is shifted out of the memory. The latest row is then examined sequentially, and each time a motion sensitive template is found, the alternating sequence of

conjugate templates is back-tracked in the previous rows of the memory. The angular velocity of a detected edge is then estimated from the beginning and end points of the corresponding track.

Both algorithms have been devised with a view to being implemented in hardware, but employ somewhat different resources. For instance, the back-tracking algorithm requires a fairly large memory which is controlled centrally, while the forward tracking algorithm can be implemented with a few registers and some simple control logic. However, the forward algorithm can only track one edge at a time, and hence, in practice, several tracking "engines" would be required. Therefore, it is debatable whether the overall hardware requirements would vary widely.

In terms of performance, the forward tracking algorithm produces more consistent velocity measurements than the other algorithm, due mainly to the fact that the displacement from one step to the next is at most equal to the maximum range. The back-tracking algorithm, on the other hand, is able to track any number of edges, and is better suited to tracking edges close to the sensor (i.e., fast relative motion).

5: Range estimation and time to impact

As suggested in the introduction, motion parallax may be used to evaluate distances between a moving observer and objects in its environment.

$$R = \frac{v \times \sin\varphi}{\dfrac{d\varphi}{dt}}$$
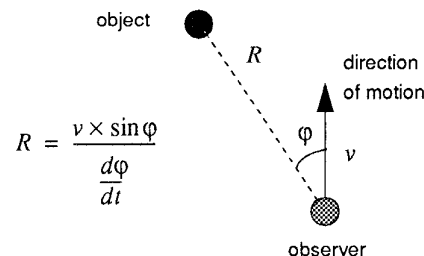
Figure 7 Range from self-motion

In the example of Figure 7, the distance R between the sensor and an object may be obtained from the angular position (φ) of the object with respect to the sensor's direction of motion, the relative angular velocity ($d\varphi/dt$), and the sensor's own velocity (v).

Relative velocity measurements may also be utilised to provide information concerning the presence of objects in the environment. In particular, when the sensor is on a collision course with an object, the edges on either side of it should appear to move away from each other. The time to impact(T) may then be estimated directly from the angle subtended by the object and the rate of angular expansion.

The "ideal" case is depicted in Figure 8, where the heading direction of the sensor is towards the centre of the object, while Figure 9 shows a more general case. For small angles, the case of Figure 8 simply reduces to Hoyle's formula [15].

$$T = \frac{1}{2} \times \frac{\sin 2\varphi}{\dfrac{d\varphi}{dt}}$$

Figure 8 Time to collision (simple case)

$$R \sin (\varphi_c - \varphi_p) = d \sin \varphi_p$$

if v is the velocity of
the sensor relative
to the object, then: $T_A = \dfrac{R \cos \varphi_c}{v}$ and $t_c - t_p$

and therefore: $T_A = (t_c - t_p) \times \dfrac{\sin \varphi_p \times \cos \varphi_c}{\sin (\varphi_c - \varphi_p)}$

Figure 9 Time to impact (general case)

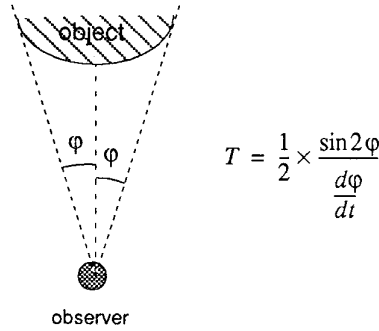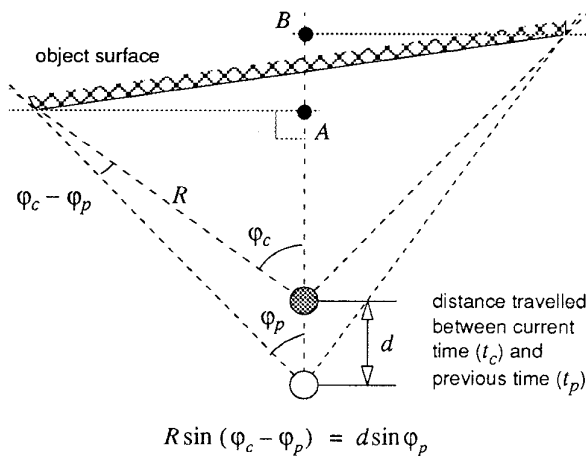It should be pointed out that, strictly speaking, T_A in Figure 9 is not the time to impact but instead the time the observer would take to reach point A which is in front of the object. Nonetheless, this result is quite remarkable considering that neither the sensor's motion, nor the

distance between the sensor and the object are known. It is worth pointing out that sensitivity to this "looming" effect is biologically plausible (see [16] for instance).

6: Looming and track association

The looming phenomenon may be detected with the bugeye by noticing that the moving edges are of identical polarity. For instance, if the object is dark with respect to the background, the motion templates detected to the left and right would indicate bright-to-dark changes in contrast. Figure 10 shows an example of a looming picture, which also corresponds to the case of Figure 9, as the angular velocity of the edge on the right-hand-side is, on average, slower than its counterpart to the left.

Figure 10 Template response to looming object (the relative heading direction is in the centre of the visual field)

The detection of looming illustrates one example of interpreting velocity measurements by taking into account the polarity information conveyed by the templates. In general, if an object's contrast with respect to the background is reasonably constant, and if the object is wide enough or close enough to the sensor, relative motion should elicit the detection of at least two edges.

The simple cases shown in Figure 11 differ only in terms of polarity (i.e., changes in contrast) and directions. Therefore, detecting such conditions consists of associating pairs of tracks on the basis of the absolute velocities being similar, and on identifying the combinations of changes in contrast.

It should be pointed out, however, that associated tracks do not necessarily correspond to edges belonging to the same object, particularly in the case where the sensor is rotated about its central axis ("panning" motion), and all the objects in the environment are static (see further).
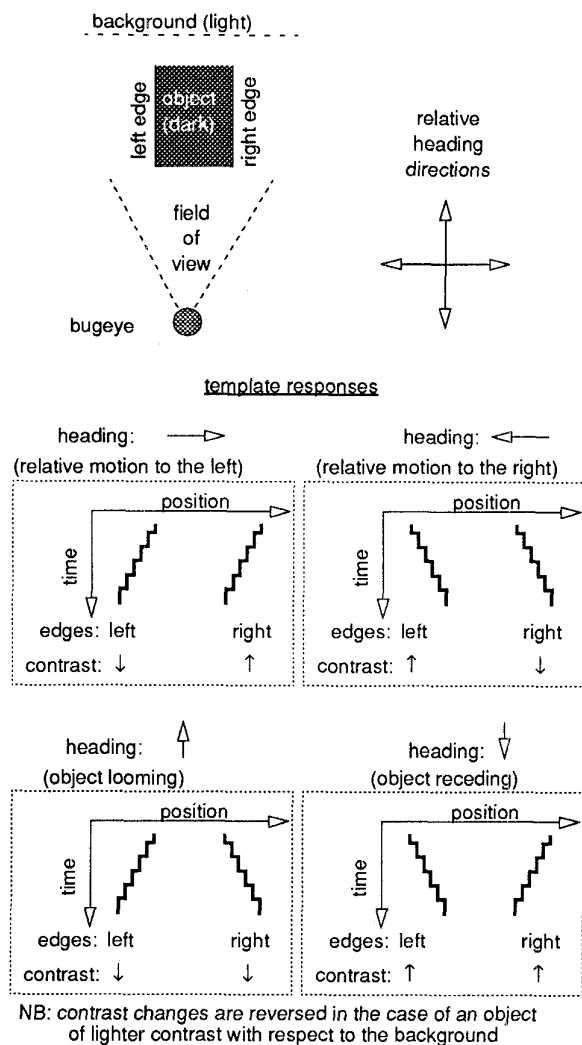
Figure 11 Tracks produced for different heading directions

Figure 12 Charge build-up model

The track association mechanism can be implemented with a matching comparator, a storage element, which could be a simple capacitor, and a thresholding device, or amplifier (Figure 12). At each time step, closely matched velocities cause "charges" to accumulate in the storage element. Conversely, if matching is insufficient, charges "leak" from the storage element. The latter thus reflects the past history of velocity matching, and controls the input to the thresholding device. Therefore, if velocities have been similar for a reasonable number of time steps, the accumulated charges reach the device's threshold value. The output state of the device then changes (or "fires"), thus indicating that velocities are matched.

This mechanism may be viewed as a crude "charge build-up" neuronal mechanism, whose dynamics are inspired from a biologically plausible model (see [17]). In Figure 12, the comparator output (q) is maximum when the velocities v_a and v_b are equal, decreases as the difference between the velocities increases, and is zero when the difference is greater than a "matching limit" parameter. Notice that charges leak from the capacitor when q is zero as well as when the amplifier is firing. This is to ensure that similar velocities cause charges to hover around the threshold, without accumulating far beyond the threshold.

The charge build-up mechanism can be implemented easily in software, using the velocity estimates produced by the forward tracking scheme of Section 4. The capacitor of Figure 12 can be modelled as a register which accumulates the charges (expressed in arbitrary units) produced by the matching comparator. Velocities are considered to be matched if the contents of the register reach a pre-set threshold, while the leakage resistor is implemented by decreasing the register value by a fixed number of units.

Notice that once the target templates have been detected for a number of time steps at least equal to the tracking depth (i.e., the windows are established), the sums of displacements are identically proportional to the velocities, and hence the displacements may be compared directly.

For the example of Figure 13, which uses the forward tracking data of Figure 5, the parameters are the following. The matching limit of the comparator is set to 3 template positions, or degrees. The comparator has a linear characteristic and produces 3 units when the velocities are equal, 2 units when the absolute difference between the sums of displacements is 1, 1 unit if the difference is 2, and zero otherwise. Finally, the threshold is set to 8 units, and the leakage is fixed at 2 units. As can be seen in Figure 13, the velocities are found to be matched at time step 167, and from time step 169 onwards.

that their velocities are similar, but of opposite polarities (i.e., the left edge is bright-to-dark, and the right edge is dark-to-bright).

forward tracking data

```
151    5:  --(--)      3:  --(--)
152    5:  --(--)      3:  --(--)
153    5:  --(--)      3:  8(0)
154    5:  --(--)      3:  9(+1)
155    5:  --(--)      3:  9(+1)
156    5:  7(0)        3:  10(+2)
157    5:  7(0)        3:  10(+2)
158    5:  7(0)        3:  11(+3)
159    5:  7(0)        3:  11(+3)
160    5:  8(+1)       3:  11(+3)
161    5:  9(+2)       3:  13(+5)
162    5:  9(+2)       3:  13(+4)
163    5:  9(+2)       3:  14(+5)
164    5:  10(+3)      3:  15(+5)
165    5:  11(+4)      3:  16(+6)
166    5:  12(+5)      3:  16(+5)
167    5:  12(+5)      3:  16(+5)
168    5:  12(+4)      3:  17(+6)
169    5:  13(+4)      3:  18(+5)
170    5:  14(+5)      3:  18(+5)
171    5:  14(+5)      3:  18(+4)
```
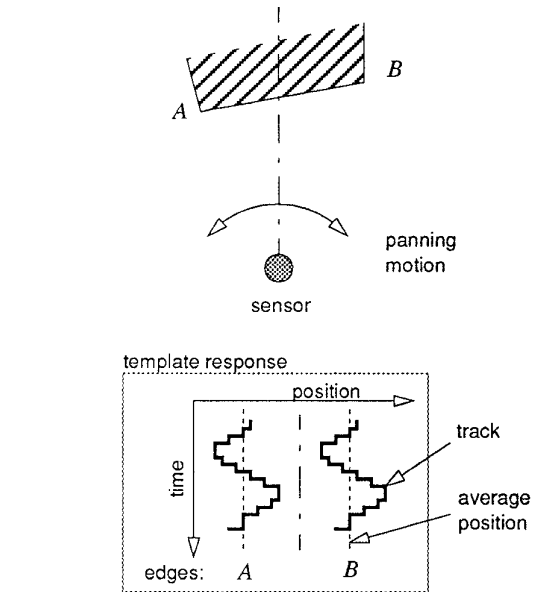
track association

```
5 /3
5 /3
5 /3
5 /3
5 /3          window
5 /3          established
5 /3
5 /3
5 /3
5 /3#
5 /3#
5 /3#
5#/3# 3 0     device
5#/3# 2 1     is firing
5#/3# 2 2
5#/3# 0 5
5#/3# 0 8*
5#/3# 2 7
5#/3# 1 9*
5#/3# 0 10*
5#/3# 1 10*
```

target   sum of displacements   absolute difference between sums of displacements   accumulated charges

Figure 13 Example of track association

7: Motion-induced navigation

The preceding sections indicate the manner in which motion information provided by the bugeye can be interpreted. In particular, it is possible to infer the presence of objects in the environment by associating motion tracks, and to estimate the distances between the sensor and the edges of objects.

Motion-induced navigation consists of combining the information provided by the sensor with knowledge of the sensor's own motion.

If the total absence of motion on the part of both the sensor and objects in its environment are chosen as initial conditions, an "angular position map" of the immediate surroundings can be constructed by panning, which consists of rotating the sensor about its central axis. The angular positions of edges are then estimated by compensating for the panning motion. In the example of Figure 14, the angular positions of edges A and B can be evaluated by averaging the maximum and minimum angular positions of the detected tracks. The motions of edges A and B may then be associated in order to infer the possibility of the presence of a solid object, by noticing

Figure 14 Estimation of angular edge positions by panning

The control system should be very circumspect in its decisions concerning track association. For instance, the association of edges B and C in Figure 15 could be interpreted as indicating either a gap, or the presence of a solid object whose contrast is lighter than that of the background (i.e., dark-to-bright change in contrast on the left and bright-to-dark on the right). Therefore, further examination of the sequence of possible "objects" is needed in order to resolve such ambiguities. In general, the control system's perception of the environment would require *a priori* knowledge, such as "objects usually present a uniform contrast", and so forth.

Figure 15 Track association ambiguities

390

Based on the angular positions of objects and empty spaces, a heading direction can be selected, and the motion of the vehicle may be utilised to estimate the distances separating it from potential obstacles (as in Figure 7), thus yielding a dynamic "relative position map". Navigation is then controlled incrementally through an appropriate choice of fixation points (see [18] for instance). In effect, the vehicle could navigate from one point to another by selecting an edge, and by controlling its motion as a function of the relative angular velocity and the changing position of the edge. This use of relative motion to continuously adjust the heading direction is reminiscent of the manner in which insects carry out depth perception and navigational tasks [19][20].

8: Control structure

The sensor provides the first measurement, or "percept", namely the indication of the presence of motion, from which more elaborate measurements are progressively derived and interpreted. Therefore, as the "level" of measurement increases, the corresponding percepts become more sophisticated. Thus the presence of motion percept is followed by a directional percept, then by some measure of velocity, and so on.

The percepts are in turn utilised by the control system at the appropriate levels of competence with a view to determining the adequate behaviour, which may imply inducing motion in order to enhance perception. A level of competence may correspond to a control layer, and in general, the principle whereby a control layer "subsumes" the functionality of the next lower level could be applied throughout the control architecture (see [21]).

The control system should be built in such a way that the lowest, or reactive, control layer is able to gain control of the vehicle in the case of looming, as the required time-critical reaction is more a reflex than a carefully thought out course of action. The control structure is depicted in Figure 16.

Ideally, the vehicle should be steered around obstacles while avoiding the looming conditions described previously. It is clear, though, that due to unforeseen circumstances the vehicle may yet find itself on a collision course with an obstacle. If the looming velocities are high, thus indicating that impact may be imminent, stopping abruptly would constitute the safest option. However, if the lowest estimated time to collision is deemed to be large enough, taking into account the vehicle's mechanical inertia, the alternative would be to steer the vehicle in the direction of the edge whose relative velocity is lowest. Referring to the example of Figure 10, the vehicle would thus veer (sharply) to the right.
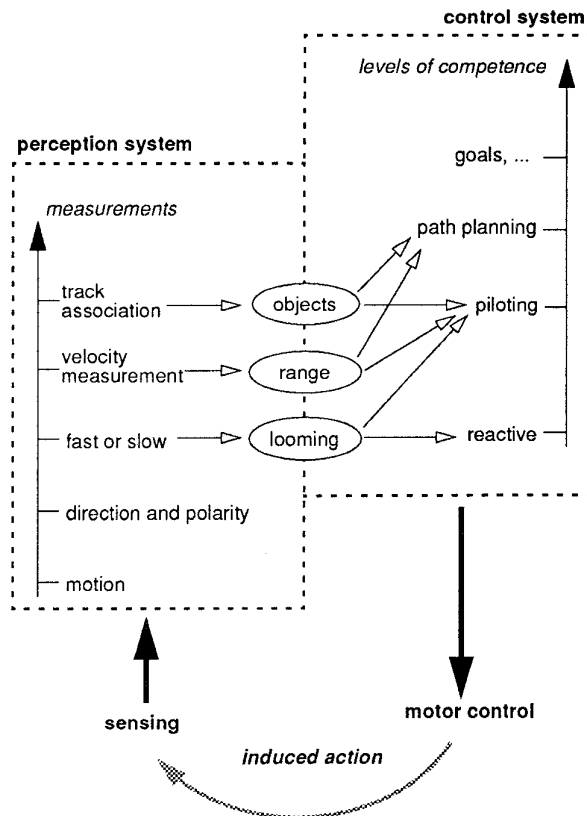

Figure 16 Relationships between sensing, interpretation, and control

While it may seem reasonable for the control system to trigger an evasive manoeuvre as soon as the computed time to impact has decreased to below some threshold value, it should be pointed out that looming conditions may be detected only when collision is imminent. Moreover, the accuracy of the time to impact computation is directly proportional to the accuracy of the velocity measurements.

This implies that the control system should look out for looming conditions as soon as possible. For instance, it may be desirable to trigger a "looming alarm" if two edges are seen to be moving away from each other at a fast rate, irrespective of the precise velocities. As suggested earlier, the reaction consists of steering the vehicle in the direction of the slowest moving edge, implying that the evasive manoeuvre is inferred solely from the difference between the velocities pertaining to the fast moving edges, and is obtained directly from the tracking data, without computing the time to impact.

An even simpler mechanism, which appears to exist in locusts [22], would consist of triggering an evasive manoeuvre when an object subtends more than some portion of the visual field, irrespective of the rate of expansion.

9: Discussion

The fact that nothing is detected in the absence of relative motion may be viewed either as a restriction, or, *a contrario*, as an advantage. In effect, the preliminary processing step of determining whether objects in the environment are in motion when the sensor (and therefore the vehicle) is static, is a straightforward consequence of the nature of the sensing. Hence, provided that objects are not extremely thin and on a collision course with the sensor, the absence of motion implies "no immediate danger".

An important feature of motion-induced navigation is that heading direction is not evaluated as such, but is continuously adjusted by reacting to stimuli, as distinct from estimating heading direction quantitatively before deciding whether it should be altered. In other words, control utilises robust, albeit crude, perception, and does not rely on sensing being accurate in the metric sense. The control is therefore somewhat different from that employed in conventional robotics in that it is based on mechanisms which closely link perception systems to motor control.

Early studies of the fly's "optomotor" response [23] strongly suggest the existence of a direct linkage between visual sensors and motor systems. It is also worth pointing out that psychophysical evidence indicate that even primates, not only insects, may extract heading direction and depth information simultaneously [24], thus lending support to the biological plausibility of linking motion perception and motor control.

Another aspect concerns the possibility of learning distinctive motion patterns, which could be environment-specific, in order to facilitate navigational tasks, in much the same way as static object recognition is used in other visual tasks. Interestingly, it appears that insects are indeed able to recognise the correct path towards a "reward", based solely on relative pattern motion [25], and are able to "store" task-specific cues [26].

10: Conclusion

The schemes described in this paper for steering a vehicle around obstacles rely purely on motion information obtained in real-time from a small VLSI sensor. Moreover, the bandwidth required to process the information to the adequate level of perception is quite small compared to camera-based systems, for instance. Thus the biologically-inspired system presented here is eminently suitable to light-weight, low-power applications.

References

[1] Nakayama K. (1985) "Biological Image Motion Processing: A Review", *Vision Research*, Vol. 25, No. 5, pp 625-660.

[2] Sobel E.C. (1990) "The locust's use of motion parallax to measure distance", *Journal of Comparative Physiology A*, Vol. 167, pp 579-588.

[3] Horridge G.A. (1986) "A theory of insect vision: velocity parallax", *Proceedings of the Royal Society of London B*, Vol. 229, pp 13-27.

[4] Egelhaaf M., Hausen K., Reichardt W. & Wehrhahn C (1988) "Visual course control in flies relies on neuronal computation of object and background motion", *Trends in Neurosciences*, Vol. 11, No. 8, pp 351-358.

[5] Campani M. (1995) "Optic flow and autonomous navigation", *Perception*, Vol. 24, pp 253-267.

[6] Cliff D. (1991) "A computational hoverfly; a study in computational neuroethology", *Proceedings, International Conference on Simulation and Adaptive Behaviour*, Paris, France, pp 87-96.

[7] Franceschini N., Pichon J-M. & Blanes C. (1991) "Real time visuomotor control: from flies to robots", *Proceedings, International Conference on Advanced Robotics*, Pisa, Italy, pp 931-935.

[8] Lopez L.R. (1994) "Neural Processing and Control for Artificial Compound Eyes", *Proceedings, IEEE International Conference on Neural Networks*, Walt Disney Dolphin Hotel, Orlando, Florida, U.S.A., Vol. 5, pp 2749-2753.

[9] Horridge G.A. (1990) "A template theory to relate visual processing to digital circuitry", *Proceedings of the Royal Society of London B*, Vol. 239, pp 17-33.

[10] Yakovleff A., Moini A., Bouzerdoum A., Nguyen X.T. Bogner R.E., Eshraghian K. & Abbott D. (1993) "A micro-sensor based on insect vision", *Proceedings, Computer Architectures for Machine Perception Workshop*, New Orleans, Louisiana, U.S.A., pp 137-146.

[11] Moini A., Bouzerdoum A., Yakovleff A., Abbott D., Kim O., Eshraghian K. & Bogner R.E. (1993) "An Analog Implementation of a Smart Visual Micro-Sensor Based on Insect Vision", *Proceedings, International Symposium on VLSI Technology, Systems, and Applications*, Taipei, Taiwan, pp 283-287.

[12] Yakovleff A., Nguyen X.T., Bouzerdoum A., Moini A., Bogner R.E. & Eshraghian K. (1994) "Dual-purpose interpretation of sensory information", *Proceedings, IEEE International Conference on Robotics and Automation*, San Diego, California, U.S.A., Vol. 2, pp 1635-1640.

[13] Braddick O. (1974) "A Short-Range Process in Apparent Motion", *Vision Research*, Vol. 14, pp 519-527.

[14] Nguyen X.T., Bouzerdoum A., Bogner R.E., Eshraghian K., Abbott D. & Moini A. (1993) "The Stair-Step Tracking Algorithm for Velocity Estimation", *Proceedings, First Australian and New Zealand Conference on Intelligent Information Systems*, Perth, Australia, pp 412-416.

[15] Abbott D., Moini A., Yakovleff A., Nguyen X.T., Blanksby A., Kim G., Bouzerdoum A., Bogner R.E. & Eshraghian K. (1994) "A new VLSI smart sensor for collision avoidance inspired by insect vision", *SPIE Proceedings, Intelligent Vehicle Highway Systems*, Boston, Massachusetts, U.S.A., Vol. 2344, pp 105-115.

[16] Regan D. & Hamstra S.J. (1993) "Dissociation of Discrimination Thresholds for Time to Contact and for Rate of Angular Expansion", *Vision Research*, Vol. 33, No. 4, pp 447-462.

[17] Abbott L.F. & Kepler T.B. (1990) "Model Neurons: From Hodgkin-Huxley to Hopfield", in *Statistical Mechanics of Neural Networks (Lecture Notes in Physics, 368)*, Luis Garrido ed., Springer-Verlag, pp 5-18.

[18] Ishiguro H., Stelmaszyck P. & Tsuji S. (1990) "Acquiring 3-D Structure by Controlling Visual Attention of a Mobile Robot", *Proceedings, IEEE International Conference on Robotics and Automation*, Cincinnati, Ohio, U.S.A., Vol. 2, pp 755-760.

[19] Lazzari C. & Varjù D. (1990) "Visual lateral fixation and tracking in the haematophagous bug *Triatoma infestans*", *Journal of Comparative Physiology A*, Vol. 167, pp 527-531.

[20] Srinivasan M.V. (1992) "How bees exploit optic flow: behavioural experiments and neural models", *Philosophical Transactions of the Royal Society of London B*, Vol. 337, pp 253-259.

[21] Brooks R.A. (1986) "A Robust Layered Control System for a Mobile Robot", *IEEE Journal of Robotics and Automation*, Vol. RA-2, No. 1, pp 14-23.

[22] Robertson R.M. & Johnson A.G. (1993) "Collision Avoidance of Flying Locusts: Steering Torques and Behaviour", *Journal of Experimental Biology*, Vol. 183, pp 35-60.

[23] Reichardt W. (1961) "Autocorrelation, a Principle for the Evaluation of Sensory Information by the Central Nervous System", in *Sensory Communication*, Walter A Rosenblith Ed., MIT Press and John Wiley & Sons (New York, London), pp 303-317.

[24] Perrone J.A. & Stone L.S. (1994) "A Model of Self-Motion Estimation Within Primate Extrastriate Visual Cortex", *Vision Research*, Vol. 34, No. 21, pp 2917-2938.

[25] Lehrer M. (1990) "How bees use peripheral eye regions to localize a frontally positioned target", *Journal of Comparative Physiology A*, Vol. 167, pp 173-185.

[26] Lehrer M. (1994) "Spatial Vision in the Honeybee: the Use of Different Cues in Different Tasks", *Vision Research*, Vol. 34, No. 18, pp 2363-2385.