

Visual Motion Sensors and Neuromorphic Obstacle Avoidance System

¹S. Pavani, ²K.Sri sushma, ³K. Chaitanya, ⁴V. Sreedhar,

¹m.Tech, Dsce, Snist, Hyderabad, India

²M.tech(IT&ESD), Hyderabad

³M.S Vlsi&esd, Jntu-Ceh, , Hyderabad

⁴M.Tech, Dsce, Snist, Hyderabad, India

Abstract: In this work, we present motion detection algorithms that are based not only on biological models but also on the computational properties of motion perception. First, we describe monolithic implementations of hysteretic winner-take-all and nonlinear differentiator based algorithms. Second, we explain multi-chip implementations of biomimetic intensity-based models, namely Adelson-Bergen model. In addition, we describe an obstacle avoidance algorithm that is realized by incorporating a multi-chip version of the Adelson-Bergen algorithm with centering behavior and time-to-collision computation. In this way, the overall system can successfully acquire clues about the structure of its environment so that collisions can be effectively avoided. This system might be employed in building a robot that can navigate in complex cluttered environments.

I. Introduction

In this we describe an obstacle avoidance algorithm based on the system level implementation of the Adelson-Bergen multi-chip sensor integrated with centering and escape behaviors. As with biological systems, perception of motion information occupies a vital role in behavioral tasks achieved by artificial systems. It is utilized for tracking, collision avoidance, object recognition, time-to-flight computation, guidance, balance, and postural control. In building biologically inspired architectures, it always has to be taken into consideration that biological models are by their nature continuous-time systems and they operate by employing a massively parallel processing strategy. These biological principles can be efficiently utilized in visual computations. In contrast to other sensory computations, optic flow computation is a very intensive process that is constrained by power consumption, and employing such technology definitely improves the performance of built systems.

In addition, conventional design approaches in image processing that are implemented by employing a CCD camera together with a DSP processor cause problems such as an image transfer bottleneck, temporal aliasing and high power consumption. These systems work at high frequencies to deliver image information in a timely way and in achieving this they have to consume considerable amounts of power. Whereas, an integrative approach by analog VLSI technology and neuromorphic design principles makes intensive visual computations possible to be realized in power and space efficient systems. In contrast to conventional designs, these systems are data driven, that is, the output is sampled when there is a demand. Hence they consume little power and yield temporal aliasing-free computation.

Inspired by biological models in the visual pathways of organisms and by computational properties of visual motion perception, a variety of visual motion sensors, which consume little power and work in real time, have been developed to solve problems faced in optical flow computation.

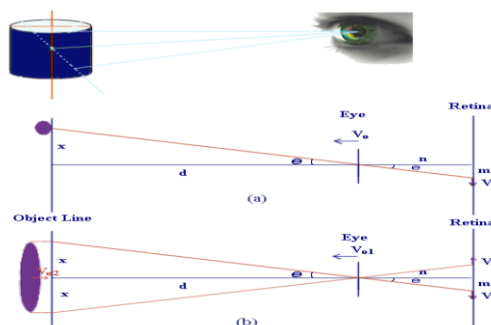


Figure 5.1: Motion parallax. (a) The effect on the retina of egomotion towards an object is illustrated. (b) The effect of egomotion and/or object motion on a collision course is depicted.

Motion Parallax.

In Figure 5.1a, the effect of an object’s motion on the retina while the observer moves is illustrated. In this case, the object is not in the heading direction. Hence in order to avoid collision with such an object we have to relate the relative distance and the speed of the observer to those of the object. In this figure μ is defined as the angle subtended by the object from the line of motion, x represents the distance of the object perpendicular to the line of the heading direction and d is used to show the distance of the observer perpendicular to the trajectory of the object. Also V_o represents the speed of the observer and V_r is used to show the speed of the object’s image on the retina. In addition, m represents the distance of the object’s image on the retina perpendicular to the line of the heading direction and n is used to show the length of the retinal focus. In Figure 5.1a, the relation between the distance parameters and the angle can be found as follows:

$$\theta = \tan^{-1} \frac{x}{d} = \tan^{-1} \frac{m}{n} \tag{5.1}$$

In this study, we assume that the movement of the observer is a pure translation in the heading direction at a constant speed and therefore V_o can be set equal to the time derivative of the distance d and V_r can be set equal to the time derivative of the distance m .

$$\frac{\partial \theta}{\partial t} = \frac{\frac{-x}{d^2}}{1 + (\frac{x}{d})^2} \frac{\partial d}{\partial t} = -\frac{1}{d} \tan \theta \frac{1}{1 + \tan^2 \theta} V_o \tag{5.2}$$

$$\frac{\partial \theta}{\partial t} = \frac{\frac{1}{n}}{1 + (\frac{m}{n})^2} \frac{\partial m}{\partial t} = \frac{1}{n} \frac{1}{1 + \tan^2 \theta} V_r \tag{5.3}$$

where x and n are assumed to be constant and, By equating these two functions and ignoring the minus sign in equation 5.2, we can find the following expression:

$$V_r = \frac{m}{d} V_o = \frac{nx}{d^2} V_o \tag{5.4}$$

From these relations, we can easily derive the distances x , d and z (the distance of the object from the observer) as follows:

$$x = \frac{m^2 V_o}{n V_r} \tag{5.5}$$

$$d = m \frac{V_o}{V_r} \tag{5.6}$$

$$z = \sqrt{x^2 + d^2} = \frac{V_o m}{V_r n} \sqrt{m^2 + n^2} \tag{5.7}$$

From these equations, the relative distance parameters can easily be calculated without knowing the external distance parameters such as d and x . By using the relations between the speed and the distance parameters, we can calculate the distance of the observer from the center of the object as

$$d = m \frac{V_o}{V_r} \tag{5.8}$$

follows

The time-to-collision with the object can be found as follows:

$$t_{collision} = \frac{d}{V_o} = \frac{m}{V_r} \tag{5.9}$$

In this equation, m and V_r can be extracted from the optical motion fields on the retina. By computing these parameters, the time-to-collision can be easily obtained.

The Centering Behavior

The underlying mechanism for the centering behavior is a non-direction selective movement detector, output of which encodes retinal velocity information. It is suggested that the centering mechanism may also play a role in the movement avoidance response. Thus the centering response can be used to explain movement avoidance as well as obstacle avoidance. In implementations on robots or different agents, the main aim is to maximize the robustness and effectiveness of the obstacle avoidance system. It can be successfully incorporated into systems to minimize the possibility of collisions. CCThis information is used as a gain in the computation of the next rotatory movement of the system. In this way, the effect of a larger motion on one of the eyes can be suppressed with a motion in opposite direction. Assuming that I_{right} and I_{left} are the angular speed of image perceived by the right and left eye respectively, then motion contrast can be expressed as follows:

$$I_{contrast} = \frac{I_{right} - I_{left}}{I_{right} + I_{left}} \tag{5.10}$$

Its response can be saturated for a range of values and the linearity range can be adjusted accordingly if it is used with a nonlinear function as shown below:

$$I_{gain} = I_{bias} \tanh(C_{linearity} \cdot I_{contrast}) \quad (5.11)$$

where $C_{linearity}$ is a coefficient used to adjust the linear region of the \tanh function. The contrast and gain functions are illustrated in Figure 5.2. Firstly, an obstacle avoidance system can be developed by utilizing the fact that the maximum velocity signals the closest object to each eye. In this way the system is made to focus on imminent collisions. In this thesis, since we use Adelson-Bergen sensor instead of a velocity sensor we utilize the spatiotemporal response of the sensor to employ this obstacle avoidance method. By using EMD responses and the maximum operation, we can express I_{right} and I_{left} in Equation 5.10 as follows:

$$I_{right} = \max(I_{EMDr1}, I_{EMDr2}, \dots, I_{EMDrn}) \quad (5.12)$$

$$I_{left} = \max(I_{EMDl1}, I_{EMDl2}, \dots, I_{EMDln}) \quad (5.13)$$

where I_{EMDri} and I_{EMDli} represent the motion outputs of the EMDs on the right and left eyes, respectively, and n is the number of EMDs on each eye. In order to obtain the average of the motion outputs on both eyes, we use a predetermined threshold. In this way, we can determine the number of EMDs that have motion responses exceeding this threshold. At the same time, we sum the motion responses that exceed the threshold. By using these parameters we can determine I_{right} and I_{left} as follows:

$$I_{right} = \frac{(\dots I_{EMDr_{k-1}} + I_{EMDr_k} + I_{EMDr_{k+1}} + \dots)}{k} \quad (5.14)$$

$$I_{left} = \frac{(\dots I_{EMDl_{j-1}} + I_{EMDl_j} + I_{EMDl_{j+1}} + \dots)}{m} \quad (5.15)$$

where I_{EMDri} is the i th right EMD response that exceeds the threshold and k represents the number of responses that exceed the threshold. Similarly, I_{EMDlj} is the j th left EMD response that is larger than the threshold and m represents the number of responses that have values larger than the threshold. Lastly, the sum of the right and left motions can be used as a reference for the rotatory movements. In this case, the agent tries to avoid the region that causes larger spatiotemporal motion output. In this way, it minimizes the probability of collision since a larger motion response may also mean more objects or closer objects. By using the sum operation, we can express I_{right} and I_{left} as follows:

$$I_{right} = I_{EMDr1} + I_{EMDr2} + \dots + I_{EMDrn} \quad (5.16)$$

$$I_{left} = I_{EMDl1} + I_{EMDl2} + \dots + I_{EMDln} \quad (5.17)$$

In order to obtain quantitative results to compare the effectiveness of these algorithms, we simulated them in 2-D environments. From these simulations, we can safely conclude that taking the sum of velocity vectors on both eyes is the best way of implementing the centering behavior.

Looming Sensitive Neurons and Time-To-Collision Computation

Wang and Frost found that in the pigeon brain, the nucleus rotundus responds maximally to objects on a collision course. In this respect, the anatomy of locusts and pigeons provides great deal of information to model neurons specialized in object avoidance and escape mechanisms. The locust obtains its sensitivity to approaching objects by the neural computations performed by wide-field visual neurons, namely the lobula giant movement detector (LGMD) and the descending contralateral motion detector (DCMD). The DCMD neuron fires when there is a movement anywhere within the visual field, but responds strongly to approaching objects on an impending collision course. Rind et al. proposed that the DCMD tracks edge motion throughout the object approach and produces strongly maintained response when the object moves close to the eye.

Hardware Architecture of the Centering Behaviour

In order to build a robust and reliable system based on this behavior, nondirectional velocity information should be obtained from the optical flow field and accordingly the sums of the left and right eye motion fields should be calculated. For our hardware implementation, we utilize the Adelson-Bergen correlation-based visual motion sensor to obtain small field motion output. However, the Adelson-Bergen motion detector is not exactly a velocity sensor, rather it is a spatiotemporal frequency tuned visual motion sensor since its velocity response confounds velocity with the spatial structure of the object it is exposed to. In this context, we refer to the velocity as the ratio of the temporal and spatial frequencies, mathematically shown as follows:

$$V = \frac{w_t}{w_x} \quad (5.18)$$

where V represents velocity, w_t is the temporal frequency and w_x is the spatial frequency of the image motion. A solution to the problem of velocity estimation is proposed by Srinivasan et al. They suggested that a

velocity sensitive model can be constructed by using multiple correlators. This model can be achieved by tuning each correlator to a different spatiotemporal frequency as shown in Figure 5.4.

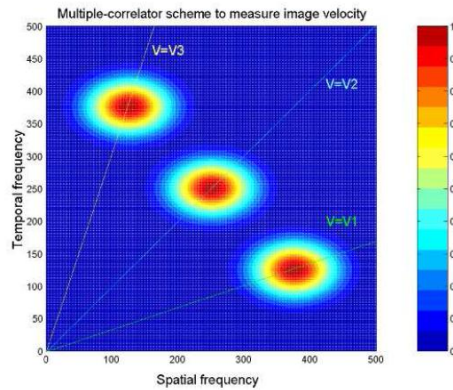


Figure 5.4: Multiple-correlator scheme realized by correlation-based motion detectors. The responses of a correlation based motion detector are shown for three different spatiotemporal frequency tuning. Each line that crosses the maximum of a correlator represents the angular velocity of the image detected by that correlator.

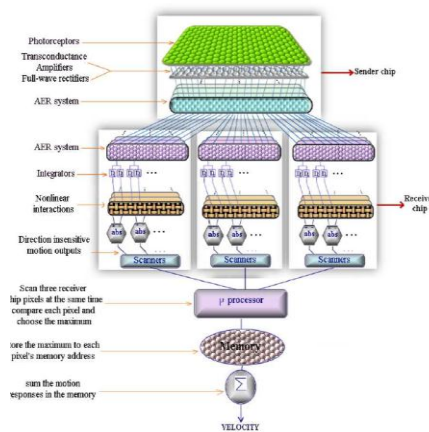


Figure 5.6: An implementation of multi-correlator based velocity sensor realized with three correlators to be used in wide field velocity detection of the retinal image. Response of the sender pixels are transmitted to three different receiver chips that are tuned to different spatiotemporal frequencies. The motion outputs of same pixels in each receiver chip are scanned by the micro processor at the same time to be able compare them. The maximum output is stored in the memory and once processor scans the last pixel, all stored motion outputs are summed.

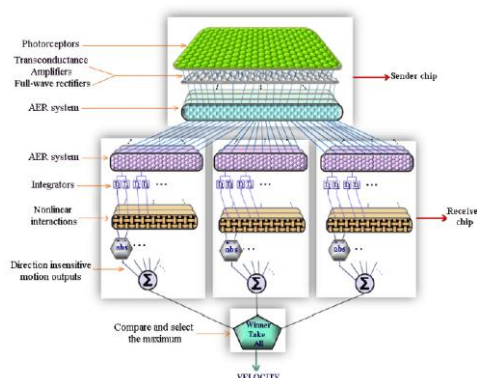


Figure 5.7: The second implementation of multi-correlator based velocity sensor. One sender and three receiver chips are employed to detect the small field motion output for three different spatiotemporal frequency tunings. The pixel responses of each receiver chip are summed individually and then compared by winner take all circuit to estimate the velocity of optical image.

The sum of the maximum small field motions can be mathematically represented as follows:

$$I_{sum} = \max_1(I_{EMD_1}, I_{EMD_2}, I_{EMD_n}) + \max_2(I_{EMD_1}, I_{EMD_2}, I_{EMD_n}) \dots + \max_n(I_{EMD_1}, I_{EMD_2}, I_{EMD_n}) \quad (5.19)$$

In order to implement such a computational scheme as shown in Figure 5.7, the same signal the sender chip produces can be transferred to multiple receiver chips that are tuned to different temporal frequencies. After the small field motion computation in these receiver chips, the pixel outputs are absolute valued and then summed. Accordingly, each receiver will produce a sum and this sum will be compared with other receiver outputs to decide for the maximum wide-field velocity. The computation can be formulated as follows:

$$I_{sum} = \max(\text{sum}_1(EMD_1, \dots, EMD_n), \text{sum}_2(EMD_1, \dots, EMD_n), \text{sum}_3(EMD_1, \dots, EMD_n)) \quad (5.20)$$

However, in this way, we can decrease the complexity and the computational load of the overall system.

Circuit implementation of the direction insensitive movement detector neuron

A multi-chip system can be implemented by incorporating a sender chip with one or multiple receiver chips. Each pixel in the sender has an AER circuit so that immediate changes can be reliably sent to receiver chips. A receiver chip is utilized to perform small- and wide-field motion computations. It employs the elementary motion detectors (EMD) to obtain small field motion vectors. For that purpose Adelson- Bergen motion detectors are used and the outputs of these detectors are absolute valued to realize the direction-insensitive visual motion sensor. Each detector in the receiver chip integrates the signals sent by the sender and performs the small-field motion computation. These pixels integrate the signals from the sender so that the amplitude of the changes in the intensity level of the image sensed by each photoreceptor can be reconstructed and necessary delays to realize the correlation scheme can be obtained. The P and N pixels that integrate the positive and negative parts of the signal sent by the sender are illustrated in Figure 5.8a and Figure 5.8b, respectively. The P pixel is designed to contain the motion computation circuitry, whereas the N pixel includes only AER interface circuit and integrators. After the integration stage, the motion computation is performed by using subtraction and absolute value circuits. The current output of the Adelson-Bergen sensor can be formulated as follows:

$$I_{motion} = |I_{2negD} + I_{1neg} - I_{2posD} - I_{1pos}| + |I_{2neg} + I_{1posD} - I_{2pos} - I_{1negD}| - |I_{2posD} + I_{1neg} - I_{2negD} - I_{1pos}| - |I_{2neg} + I_{1negD} - I_{2pos} - I_{1posD}| \quad (5.21)$$

After this stage, an absolute value circuit is added to obtain the final direction-insensitive motion output. The final motion output of each EMD can be formulated as follows:

The wide-field motion output is obtained by making use of the capabilities of the scanners that help us sum the motion outputs of all EMDs.

This system can be used to detect direction-insensitive motion sensed by the right and left eyes. The motion output does not represent the motion velocity but an implementation of the multicorrelator based velocity sensor is straightforward with a modular design strategy. Firstly, the temporal frequency tuning of each chip can be adjusted by changing the biases of the integrator circuit. Secondly, the spatial tuning can be changed by interchanging the columns of the AER system in such a way that instead of computing the motion response of neighboring pixels, interaction between odd-numbered pixels or even-numbered pixels is used. This strategy can be achieved by changing the column numbers of the receiver chip as illustrated below:

Sender chip	Receiver chip
1	1
2	3
3	5
4	2
5	4
6	6

The unwanted interaction between the second and fifth columns can be avoided by implementing the circuit in such a way that there will not be an interaction between these columns.

PIC can be employed to map the addresses sent from the sender to the receiver chip. Besides, since in this case we do not have any controlover the internal structure of the chip, we need to use a new mapping strategy to avoid unrelatedinteraction between the second and fifth columns. An alternative mapping can be realized as follows:

Sender chip	Receiver chip
1	1
2	3
3	5
4	6
5	4
6	2

In this case we will have motion output obtained from the interaction between the fifth and sixth.

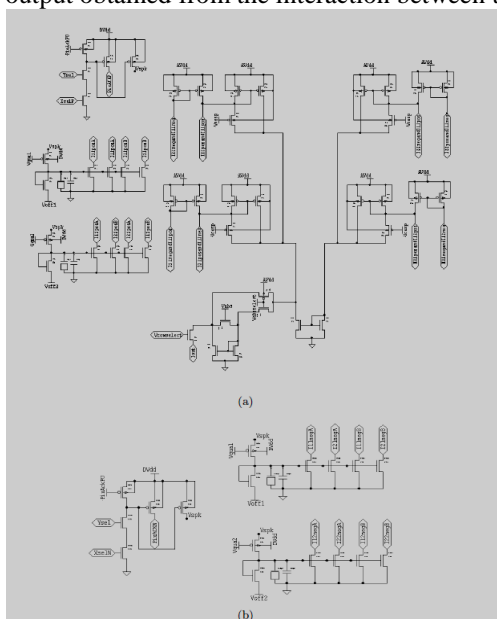


Figure 5.8: The implementation of the circuit that integrates the positive and negative parts of the signal sent from the sender and computes the final motion output. (a) The P pixel that consist of AER interface circuit, integrators, absolute value circuits and subtractors. (b) The N pixel that only integrates the negative part of the signals.

Hardware Architecture of the Escape Behavior

Although the centering behavior is a very efficient way of dealing with navigation in cluttered environments, it becomes ineffective when it is challenged by objects in the heading direction. Particularly, the symmetric expansion of an object in the heading direction will not cause a robot driven by the centering response to turn away. For this reason, a looming sensitive sensor should be implemented and placed in the heading direction of the system. In this way, during the pure translator movements of the robot, the looming sensitive sensor can signal the objects that are on a collision course or can compute the time to collision parameter so that the angle of rotation to escape from the object can be calculated without initiating a sudden turn. Firstly, we assume a symmetric or quasi-symmetric expansion of an object in the heading direction, because in other cases, the centering response can initiate enough turn to escape from the object. Therefore, the real problem is to be able to signal an escape response when the object is looming symmetrically on a collision course. Looming is a real time response to movements on a collision course and therefore it has to be tracked and integrated in time. For this reason, after finding the maximum velocity on each eye, they need to be tracked in time and checked if the rightward maximum velocity in the right eye is moving to the right and if the leftward maximum velocity in the left eye is moving to the left.

Assuming that the center of the visual field is the origin of the position vector, then we can relate the positions of the maximum velocities to find the symmetry parameter. For instance, if the magnitude of the maximum velocity positions on each eye are very different, then this information gives us a clue that the expansion is not symmetric and the centering behavior has to be employed to avoid collisions in such cases.

An alternative strategy can be employed by changing the maximum operation with a group of velocities limited by the maximum velocity operation. The main purpose of this operation is to track the closest part of the object and from this information to find the size and expansion rate. As a result of these observations, we can find the degree of looming by utilizing the formula found in Equation 5.9 and formulate the escape parameters that initiate the escape behavior as follows:

$$I_{timeleft} = \frac{\theta}{\delta\theta} \tag{5.23}$$

$$I_{symmetry} = \frac{x_{mazright} - x_{mazleft}}{x_{mazright} + x_{mazleft}} \tag{5.24}$$

$$looming = \frac{1}{I_{timeleft}} = \frac{\delta\theta}{\theta} \tag{5.25}$$

Algorithm Implemented on an Agent

In order to implement the centering behavior, the responses of the comparison circuits are summed by using a sum circuit. As a result of this operation, we can get an estimate of the rightward and leftward motion. The responses are sent to the micro-processor which processes this information and mediates the movement of the agent. The process includes finding the contrast of the right motion relative to the left motion. Accordingly, contrast levels that are close to zero inhibit rotation and allow only translatory movement of the agent. Whereas, motion contrasts that are close to plus or minus one create the largest rotator movements.

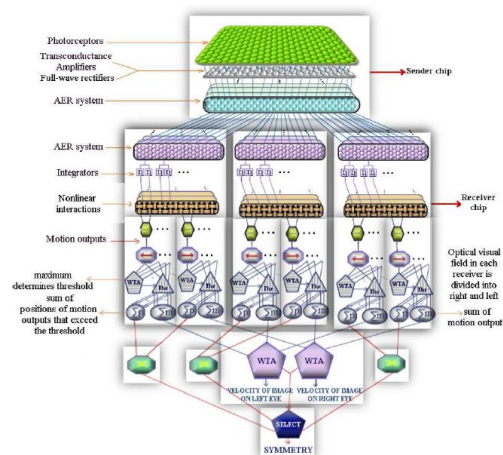


Figure 5.9: The final implementation of the escape response. One sender and three receiver chips are employed to extract the small field motion output. Each receiver chip is tuned to a different spatiotemporal frequency and its optical field is divided into two regions to represent the right and left eyes. The motion outputs on each eye are rectified and then the maximum rectified output on each eye becomes the threshold of motion responses. After the threshold stage, the sum of the positions and magnitudes of motion signals are calculated to calculate the symmetry, and the right and left eyes motions.

This agent with its escape and centering behavior can allow successful navigation in cluttered and hazardous environments and is therefore well-suited for robotics applications.

II. Conclusion

In this thesis, monolithic and multi-chip implementations of neuromorphic visual motion models were explained and a biologically inspired obstacle avoidance system based on the anatomy and physiological behaviors of flying insects was described. we designed a biomimetic obstacle avoidance system by exploiting the centering and escape behaviors observed in flying insects. In the motion computation, we utilized motion parallax to obtain information on the obstacles and structure of the environment. In order to achieve navigation in cluttered environments, we implemented the centering behavior which is mediated by a direction insensitive movement detecting system. By taking the absolute value of the Adelson-Bergen algorithm output, we suggested an implementation of a multi-chip nondirection selective motion sensor to realize the centering behavior. In this thesis, we proposed that multi-correlator scheme can be used to improve the reliability of the velocity output obtained from the implemented sensor. The main problem with the centering behavior is that it is not that effective in avoiding collisions with objects that are in the heading direction. We tried to solve this problem by utilizing the escape response which is initiated by the lobula giant movement detector in locusts to avoid predators. Together with the centering behavior we obtained profoundly promising results from the simulations. However, the proposed system has to be improved in terms of the number of motion chips used in order to make the system more practical in robotics applications. Furthermore, the optimization has to be done to decrease the number of sender chips and to increase the angle of the visual field.

Bibliography

- [1]. M.V. Srinivasan and M. Lehrer, Temporal acuity of honeybee vision: behavioral studies using moving stimuli.
- [2]. M.V. Srinivasan, M. Lehrer, W.H. Kirchner, and S.W. Zhang, Range perception through apparent image speed in freely-flying honeybees.
- [3]. M.V. Srinivasan and S.W. Zhang, Evidence for two distinct movement-detecting mechanisms in insect vision, 80 (1993), 38–41. Visual control of honeybee flight, *Orientation and communication in arthropods* (M Lehrer, ed.), 1997, pp. 95–113.
- [4]. M.V. Srinivasan, S.W. Zhang, and N. Bidwell, Visually mediated odometry in honeybees.
- [5]. A. Starzyk, J. and X. Fang, CMOS current mode winner-take-all circuit with both excitatory and inhibitory feedback, *Electronic Letters* 29 (1993), no. 10, 908–910.
- [6]. H. Sun and B. J. Frost, Computation of different optic variables of looming objects in pigeon nucleus rotundus neurons, *Nat Neurosci* 1 (1998), 296–303.
- [7]. J. Tanner and C. Mead, An integrated analog optical motion sensor, *VLSI Signal Processing, II* (New York, ed.), IEEE Press, 1986, pp.
- [8]. Y. Taur and T.H. Ning, *Fundamentals of modern VLSI devices*, ch. 3, Cambridge University Press, 1998.
- [9]. C. Toumazou, F.J. Lidgley, D.G. Haigh, and P. Peregrinus, *Analogue IC design: the current mode approach*, ch. 2, Peter Peregrinus Ltd., 1990.
- [10]. P. Venier, A. Mortara, X. Arreguit, and E. Vittoz, An integrated cortical layer for orientation enhancement, *IEEE Journal of Solid State Circuits* 32 (1997), no. 2, 177–186.
- [11]. G.K. Wallace, Visual scanning in the desert locust *Schistocerca Gregaria* Forskal, *J. Exp. iol.* 36 (1959), 512–525.