

Tracking improves performance of biological collision avoidance models

Vivek Pant^{1,*} and Charles M. Higgins^{1,2}

Accepted: May 31, 2012

Abstract Collision avoidance models derived from the study of insect brains do not perform universally well in practical collision scenarios, despite the fact that the insects themselves may perform well in similar situations. In this paper, we present a detailed simulation analysis of two well known collision avoidance models and illustrate their limitations. In doing so, we present a novel continuous-time implementation of a neuronally-based collision avoidance model. We then show that visual tracking can improve performance of these models by allowing an relative computation of the distance between the obstacle and the observer. We compare the results of simulations of the two models with and without tracking to show how tracking improves the ability of the model to detect an imminent collision. We present an implementation of one of these models processing imagery from a camera to show how it performs in real-world scenarios. These results suggest that insects may track looming objects with their gaze.

Keywords Insect vision, collision avoidance, computational modeling, computer vision

1 Introduction

Collision avoidance is a behavior elicited in many organisms when an object looms in the visual field. In insects, this phenomenon has been studied extensively at both the neuronal and the behavioral levels (Rind, 1984; Borst and Bahde, 1986; Hatsopoulos *et al.*, 1995; Rind and Bramwell, 1996; Sun and Frost, 1998; Laurent

and Gabbiani, 1998; Tammero and Dickinson, 2002). Insects are able to turn their head relative to the thorax in flight, and considerable evidence suggests that they use head movements to enhance the information they get from the visual scene (Wertz *et al.*, 2009; Boeddeker *et al.*, 2010), although it is not theoretically necessary (Hyslop and Humbert, 2010). It has long been known (Land, 1973) that flies move their head to stabilize the visual scene, and can make rapid saccadic changes in fixation. This visual stabilization has more recently been shown to include both roll and yaw movements (Schilstra and van Hateren, 1998; van Hateren and Schilstra, 1999). van Hateren and Schilstra (1999) also showed that saccadic turns of the head lag slightly behind turns of the thorax in flight, thus minimizing the time of the saccade. (van Hateren *et al.*, 2005) have suggested that responses of the H1 visual motion sensitive interneuron are enhanced by this visual stabilization between saccades. Similar stabilization has been demonstrated in honeybees (Boeddeker and Hemmi, 2010). None of these experiments have observed insects in flight actively directing their gaze to specific objects. However, biological models of collision avoidance are often evaluated without any accompanying head movements. Further, since none of these experiments placed insects in looming scenarios, it is unknown whether insects track looming objects with their gaze in such scenarios.

Computational models of insect collision avoidance can be broken into two broad classes: those based on a matched filter for optical flow, and those not employing direction selective elements. In this paper, we evaluate representatives from both classes of collision avoidance models while tracking the looming object.

Collision avoidance has also, of course, been studied outside the context of biological models. Optical

¹Department of Electrical and Computer Engineering,

²Department of Neuroscience, The University of Arizona, Tucson, Arizona 85721

*Current address: Atmel Corporation, San Jose, CA

flow, which is defined as the local velocity field of point objects in a scene, has been argued to be a good mechanism to determine the motion of rigid objects (Horn, 1986; Fermuller and Aloimonos, 1992). However, the computation of optical flow is by itself an ill-posed problem and simplifying assumptions, such as smoothness of the intensity pattern, are required to solve the mathematical equations (Horn and Schunck, 1981). Several researchers have shown that it is possible to navigate without collision by using optical flow with visual tracking for a binocular observer (Bandopadhyay and Ballard, 1991; Aloimonos *et al.*, 1988). Fermuller and Aloimonos (1992) have developed a collision avoidance algorithm for an active monocular observer using normal flow (optical flow normal to the boundaries of an object). However, for this algorithm to work the normal flow must be computed at each instant of time, which is computationally intensive for most realistic scenarios. Spatio-temporal frequency based algorithms, on the other hand, are mathematically stable computationally efficient methods that can be used to compute local motion at each point in a scene (Verri and Poggio, 1989; Lindemann *et al.*, 2005). It has been shown that spatio-temporal frequency based methods such as the Hassenstein-Reichardt (HR) model (Hassenstein and Reichardt, 1956; Reichardt, 1961) may also be used for estimating the qualitative properties of motion from the real world, such as the focus of expansion, or discontinuities due to relative motion of an object with respect to its background (Verri and Poggio, 1989). In some cases, including collision detection, these qualitative properties may be more effective than quantitative optical flow methods to determine the trajectory of an approaching obstacle.

The first model of collision avoidance that we have considered is based on the response of large visual interneurons in the optic lobes of the fly. These neurons have been suggested to supply visual cues for the control of head orientation, body posture, and flight steering (Borst, 1990; Krapp *et al.*, 1998). The model is based on a matched filter approach (Franz and Krapp, 1998), whereby motion information is filtered by a set of neurons to extract information for specific tasks, including collision avoidance. This model is similar to optical-flow based models that compute expansion by summing the outward components of the flow vectors (Borst and Bahde, 1986). A matched filter based collision avoidance model was proposed by Tammero and Dickinson (2002) based on their behavioral studies of fruit flies. This hypothetical model of behavioral data does not incorporate specific visual interneurons. In the case of this model, the expansion is computed by summing the outward components of motion computed by

HR type motion detectors. We refer to this model as the Spatio-Temporal Integration (STI) model and describe it in Section 2.1.

The second model that we have considered is based on two of the most studied neurons mediating collision avoidance in insects. These are the Lobula Giant Movement Detector (LGMD) and the Descending Contralateral Movement Detector (DCMD) (Rind, 1984; Milde and Strausfeld, 1990; Rind and Simmons, 1992; Hatsopoulos *et al.*, 1995). Interestingly, neurons with very similar responses have recently been recorded in crabs (Hemmi and Tomsic, 2011). Rind and Bramwell (1996) have modeled the response of the LGMD neuron using a neural network. We refer to this model as the Rind model and describe it in Section 2.2. A competing mathematical model based on the response of the same two neurons has been described by Laurent and Gabbiani (1998). We refer to this model as the η -function model, and compare it to the Rind model in Section 2.2.

The STI and the Rind models are based on data from a limited set of biological experiments that do not account for some critical scenarios that expose their limitations. In this paper, we present simulation analyses of the STI and the Rind models for a varied set of collision and non-collision scenarios to show their limitations and propose visual tracking as a solution to these limitations. We also present a camera-based physical implementation of the STI model, and compare the performance of the model with and without tracking.

2 Models of Collision Avoidance

2.1 The Spatio-Temporal Integration model

A model for collision avoidance and the landing response in the fruit fly *Drosophila melanogaster* was proposed by Tammero and Dickinson (2002). This model is an elaboration of a scheme based on the spatial and temporal integration of small-field motion units, proposed by Borst and Bahde (1986), to explain the landing response of the housefly. Borst and Bahde studied the stereotypical leg extension response of the fly while presenting an expanding visual stimulus. They concluded that when the expansion output exceeds a threshold, the fly extends its legs to land. Tammero and Dickinson have elaborated this model to account for both landing and collision avoidance. Tammero and Dickinson's model consists of an array of HR correlation based motion detectors that compute a qualitative representation of the optical flow field (Verri and Poggio, 1989). In a two dimensional (2D) version of this model,

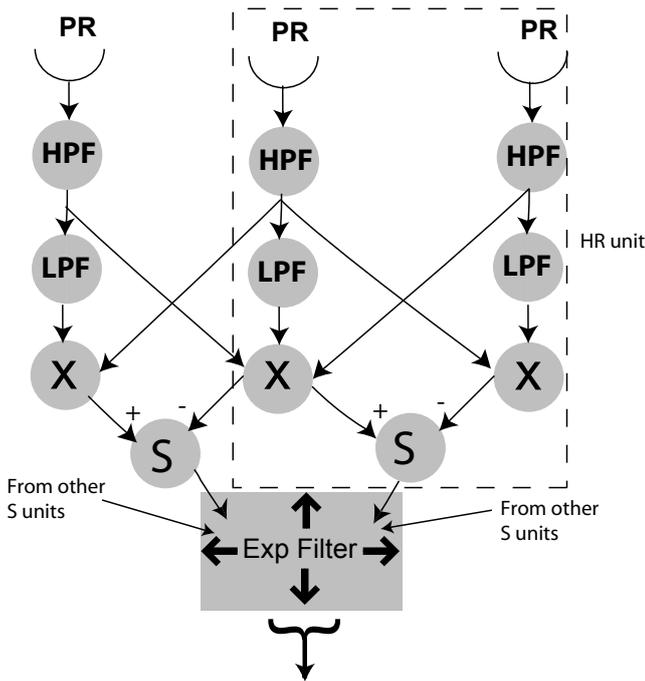


Fig. 1 Spatio-Temporal Integration model. The input layers implement an array of HR motion detector units. PR indicates a photoreceptor, LPF a temporal low-pass filter, and HPF a temporal high-pass filter. The motion output from the HR units is processed by an expansion filter which combines motion outputs along the horizontal and vertical axes to generate motion sensitivity in a radially outward direction.

the outputs from individual HR detectors in each sub-field are filtered into rightward, leftward, upward, and downward motion components. These motion components are then spatially combined such that the output in each of the four quadrants represents a net outward motion or expansion. A hardware implementation of this model has been shown to avoid collisions in many cases (Harrison, 2005); however, the results have not been analyzed to determine what type of collision scenario leads to their failure. Reiser and Dickinson (2003) have also implemented this model in a robotic fruit fly.

For this study, the STI model was implemented for a simulated insect with a 2D sensor of angular extent 180° in both azimuth and elevation (see Figure 1). The sensor plane was divided into four quadrants and the motion vectors pointing in the outward directions in each quadrant were summed together (for example, up and right motion components in the top-right quadrant). The total expansion output (E , “Exp Filter” in Figure 1) was computed as follows:

$$E = \sum_{top-left} \sqrt{neg(M_x)^2 + pos(M_y)^2}$$

$$\begin{aligned} & + \sum_{top-right} \sqrt{pos(M_x)^2 + pos(M_y)^2} \\ & + \sum_{bottom-right} \sqrt{pos(M_x)^2 + neg(M_y)^2} \\ & + \sum_{bottom-left} \sqrt{neg(M_x)^2 + neg(M_y)^2} \end{aligned} \quad (1)$$

where $pos()$ and $neg()$ are functions that respectively pass the positive and negative values of their inputs unchanged and return zero for inputs of the opposite sign. M_x and M_y are 2D matrices of motion outputs along the horizontal (x) and vertical (y) directions of the sensor plane, respectively. Components of M_x were positive when the local motion was left to right and negative otherwise. Components of M_y were positive when the local motion was bottom to top and negative otherwise.

2.2 A continuous-time implementation of the Rind model

Rind and Bramwell (1996) proposed a neural network model of the LGMD and DCMD neurons in a locust in response to looming visual stimuli. The DCMD neuron has a one-to-one synaptic connection with the LGMD neuron such that a spike in LGMD elicits a spike in DCMD (Rind, 1984). The responses of these neurons have been well characterized (Simmons and Rind, 1992; Hatsopoulos *et al.*, 1995). Intracellular electrophysiological data recorded from the DCMD neuron show that it responds strongly to expanding stimuli, but not to sustained motion. Rind and Simmons (1992) recorded extracellularly from the DCMD neuron and reported that it also responds to novel initiation of motion, but not to slow lateral motion. Based on these recordings, Rind and Bramwell (1996) proposed a neural network that models the activity of the LGMD neuron. This model has discrete-time components with ON/OFF type photodetector units at the input level. The output of the photoreceptor units interact with a delayed output of their lateral neighbors that have an exponentially decaying persistence response. Fixed-time delays and hand-tuned persistence parameters are key components in the original model.

The η -function model (Laurent and Gabbiani, 1998) differs from the Rind model in many ways, even though both are based on the response of the same LGMD-DCMD neurons. Most crucially, the η -function model requires computing the size of the obstacle and its rate of change, which requires a visual subsystem capable of extracting this information from the visual scene. The size information can be accurately estimated only by

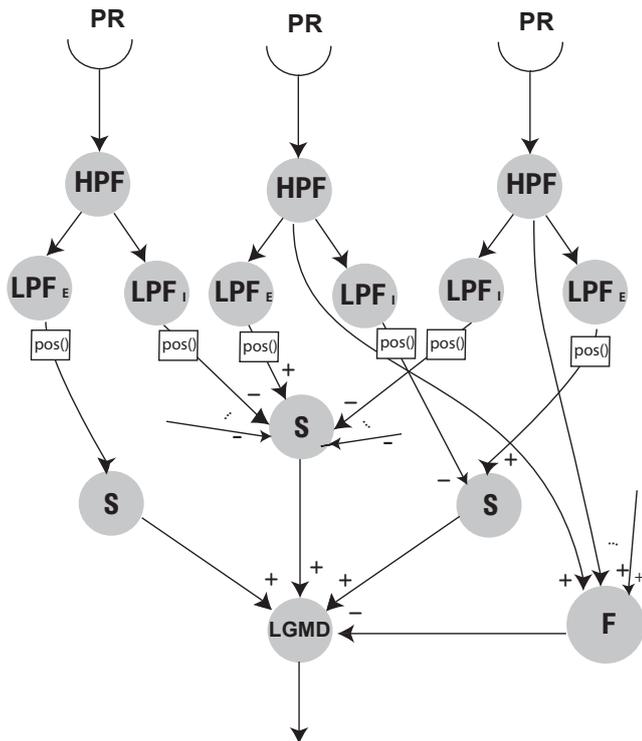


Fig. 2 Continuous-time implementation of the Rind model. The photoreceptor layer is a combination of phototransducing units (PR) and a high-pass filter stage (HPF) which replaces the edge detector in the original model. The second layer has two low-pass filtering units for excitatory (LPF_E) and inhibitory (LPF_I) units, each output of which is rectified. The third layer is a summation layer (S) that receives positive input from the LPF_E unit and negative input from 8 neighboring LPF_I units (not all connections shown). A feed-forward inhibition unit (F) aggregates the high-pass filter output from all the photoreceptor units. The fourth layer has an LGMD unit which receives positive input from all the S units and a negative input from the F unit.

distinguishing the object from the background which leads us to the very difficult problem of visual segmentation. Biological visual segmentation algorithms have been proposed (Koch *et al.*, 1986) and their electronic implementations have also been fabricated (Stocker, 2004). However, these are complex solutions and require significant processing hardware and computational time. Due to the complexity of implementing this visual subsystem, we have therefore not pursued the η -function model further.

We have implemented a novel continuous-time version of the Rind model by replacing the persistence parameters and fixed-time delays with first order low-pass filters (see Figure 2). To make the model more biologically realistic, the original model’s ON/OFF type photodetectors were replaced by a photoreceptor with a continuous response to light intensity cascaded with a temporal high-pass filter (PR and HPF in the fig-

ure). This edge information is passed into the excitatory and inhibitory units. The time-constant of the excitatory units (LPF_E, 12.3 ms) is smaller than that of the lateral inhibitory units (LPF_I, 55 ms), with these values being taken from the original Rind model. The low-pass filter units used in place of fixed delays have an added advantage. The response of the low-pass filter persists for some time based on its time-constant. This eliminates the need for artificially including persistence in the excitatory and inhibitory nodes as in the case of the original Rind model. We introduced a rectification stage after the excitatory (LPF_E) and inhibitory (LPF_I) units, thus making the output of the second layer strictly positive, a requirement in the original Rind model. A summation stage (S) receives inputs from one excitatory and eight adjacent inhibitory units, all nearest neighbors in the rectangular lattice. A feed-forward inhibition unit (F) was implemented as the sum of all the HPF units and was always active. The output from all the summation units and the feed-forward inhibition unit was pooled by the LGMD unit.

The responses of our continuous-time model are virtually identical to the original Rind model (data not shown), although our version is much simpler to implement. Like the original model, its response is dependent on the size of the approaching object and it responds selectively to approaching obstacles over receding ones. We have not included next-nearest neighbor induced lateral inhibition in our model, a feature in some implementations of the Rind model, because in our simulations this inhibition did not seem to have any significant affect on the response of the model. A detailed comparison of our implementation versus the original model can be found elsewhere (Pant, 2007).

3 Collision avoidance in a mathematical framework

In this section, we consider the collision avoidance problem from a mathematical perspective. In the real world, the trajectory charted by an obstacle on a collision course is three-dimensional (3D). However, the image that is processed by the insect visual system is only a two-dimensional (2D) projection of the real world. While the motion of the obstacle in the real world is described by three cartesian directions x , y , and z , the motion projected on the photoreceptors is possible only along two of the three directions (say, x and y). The motion along the third direction (the z axis) is not explicitly resolvable by processing a sequence of two dimensional images (Duric *et al.*, 1999). However, the *relative* distance of the objects in a scene may be computed directly by using non-directional speed and motion par-

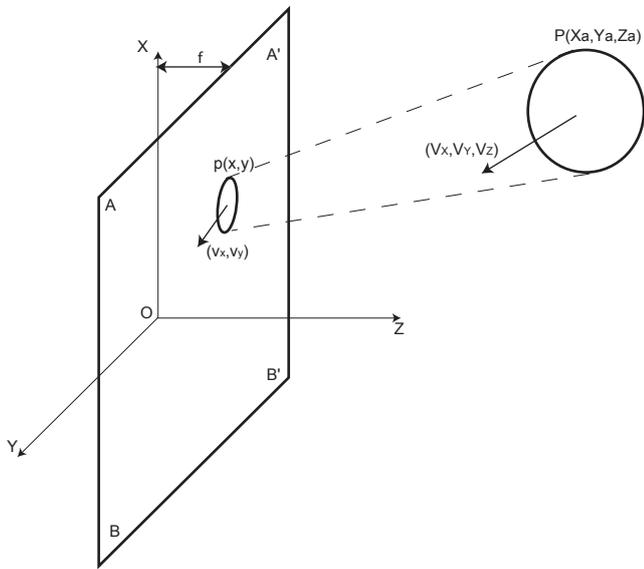


Fig. 3 2D projection of a 3D object. The point P in the real environment is projected as p on the sensor array.

allax (Bruckstein *et al.*, 2005). In this section, we show how to relate the response of the collision avoidance models to the distance from the obstacle through tracking. This is inspired by the active vision approach used in computer vision (Bajcsy, 1985).

3.1 Motion under 2D perspective projection

In a biological system, the visual input is usually modeled as a 2D projection of the real world onto an array of photoreceptors. The intensity map is then processed by successive stages of the visual system to interpret the scene (Rind and Bramwell, 1996; Franz and Krapp, 2000). The same is true for most optical-flow/motion-flow based computer vision algorithms (Horn, 1986; Ferrell and Aloimonos, 1992). In this section, we present a mathematical framework to analyze how a model that only has access to the intensity information extracts parameters of object motion like speed and distance. As shown in Figure 3, the sensors are arranged in a 2D plane $AA'BB'$. Under perspective projection (Foley *et al.*, 1995) on a plane at a distance f from the origin, the motion at a point $P(X_a, Y_a, Z_a)$ with translational speed $S(V_X, V_Y, V_Z)$, and no rotational motion component, is represented by the point $p(x, y)$ with speed $s(v_x, v_y)$, where

$$x = \frac{X_a f}{Z_a} \quad (2)$$

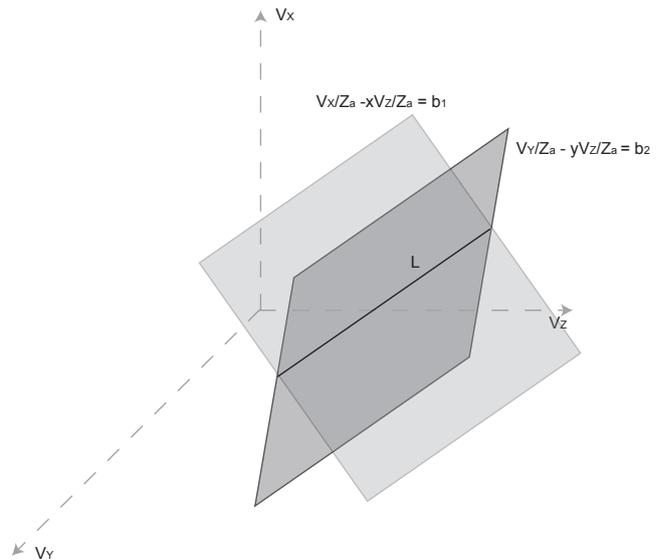


Fig. 4 Geometrical interpretation of an underdetermined system of equations. The planes parallel to the x - z and the y - z surfaces intersect each other at a line L .

$$y = \frac{Y_a f}{Z_a} \quad (3)$$

Without loss of generality, we can choose $f = 1$ as the focal length of our imaging system. Then, the velocity at point $p(x, y)$ as computed by differentiating x and y and substituting X_a and Y_a according to Equations 2 and 3 is:

$$v_x = \frac{dx}{dt} = \frac{d}{dt} \left(\frac{X_a}{Z_a} \right) = \frac{V_X}{Z_a} - \frac{xV_Z}{Z_a} \quad (4)$$

$$v_y = \frac{dy}{dt} = \frac{d}{dt} \left(\frac{Y_a}{Z_a} \right) = \frac{V_Y}{Z_a} - \frac{yV_Z}{Z_a} \quad (5)$$

Thus, we find that the speed of the projected point p on the sensor array is a combination of the speed in the respective cartesian (x or y) direction and the distance of the object normal to the sensor plane (z direction). These equations represent an underdetermined system with an infinite number of possible solutions. This becomes clear by looking at the geometrical interpretation of Equations 4 and 5. These equations represent planes in the (V_X, V_Y, V_Z) coordinate system (see Figure 4). The solution to the system of equations is represented by the intersection of the two planes: a line L . Thus, motion along the x and y directions (parallel to the sensor plane) is indistinguishable from the motion along the z direction (approach).

3.2 Tracking and collision avoidance

Many researchers in the field of computer vision have argued how an ‘active’ observer may be able to tackle

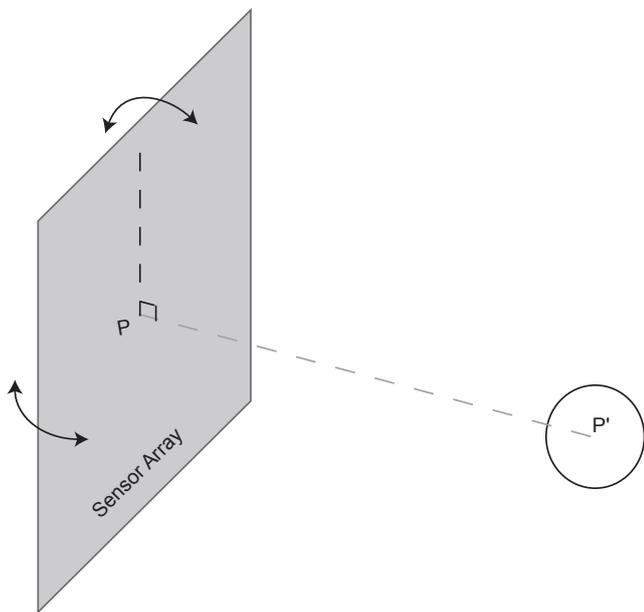


Fig. 5 Active Observer with two degrees of freedom, shown by the arrows along the horizontal and vertical axis. The visual sensor may be rotated about either axis to actively track the object in its visual field.

some of the problems we have discussed in the previous section (Bajcsy, 1985; Aloimonos *et al.*, 1988). An active observer is an observer that is free to move its ‘gaze’ in order to track (or fixate on) some feature or object in its visual field. In closely related work, Ferrell and Aloimonos (1992) showed how fixating an object may be used to simplify the computation of time to collision, a parameter which can also be used for obstacle avoidance. Figure 5 shows an active observer that has two degrees of freedom about its vertical and horizontal axes. Such an observer is able to rotate its sensor plane such that it may keep an object directly in front of it. As shown in Figure 5, this may be accomplished by keeping the line passing through the center of the sensor plane (P) and the center of mass of the object (P') normal to the sensor plane.

Tracking an object with the gaze significantly affects the pattern of optical flow experienced by the observer, and this effect has been studied in detail. Eckert and Buchsbaum (1993) demonstrated that tracking an object greatly reduces the variance of optical flow at the point of tracking while generally leading to increased variance of optical flow with increasing eccentricity from the point being tracked. Since in our case a looming object being tracked is by definition much nearer than background objects and thus will result in much larger optical flow speeds, this effect does not confuse the overall optical flow pattern, but rather emphasizes details of target optical flow, thereby mak-

ing the collision avoidance computation easier. Warren and Hannon (1990) investigated the optical flow pattern generated by a moving observer while fixating on a point on a plane. The resulting complex flow field is nontrivially decomposed into translational and rotational components, but since the optical speeds generated by a looming target are much greater than those generated by the background during fixation, this decomposition is unnecessary in our case. Daniilidis (1997) showed that fixation on a stationary point simplifies estimation of self-motion parameters from the optical flow pattern, a fact that is closely related to our analysis below.

Let us examine how tracking an object affects the computation in the case of our two representative collision avoidance models. We will first consider the case of an object moving in the real world with speed (V_X, V_Y, V_Z) relative to the observer. Let the speed of approach be zero ($V_Z = 0$). In this case the object is moving in a two dimensional plane parallel to the viewing plane. Let us also assume that the observer is equipped with an algorithm that can exactly track the object and is able to compensate for the movement of the object instantly. In this scenario, as long as the observer is able to reject self-motion components of optical flow induced due to rotation of its sensor plane, the only visual motion it experiences in the sensor plane is due to the change in the viewing angle of the object. This component is usually much smaller than actual motion in any direction. Therefore, the response of the collision avoidance models for this scenario will be greatly diminished relative to approach scenarios.

Next, let us consider the case when $V_Z \neq 0$. Since the observer is able to compensate for the motion in the $x - y$ plane (and again neglecting self-motion components), the only visual motion registered by the sensor array is due to movement of the object in the z direction and due to a change in the viewing angle. If we neglect the much smaller motion component due to the change in the viewing angle as compared to the expansive motion, the problem of 3D motion gets converted into a 1D motion problem. This is the same scenario as when the object is always maintained in the center of the visual field. Over small distances, the tracking of the object effectively makes $V_X = V_Y = 0$. Thus, the equation for motion of a projected point p on the sensor plane, as shown in Figure 3, is now simplified to:

$$v_x = \frac{dx}{dt} = \frac{V_X}{Z_a} - \frac{xV_Z}{Z_a} = -\frac{xV_Z}{Z_a} \quad (6)$$

$$v_y = \frac{dy}{dt} = \frac{V_Y}{Z_a} - \frac{yV_Z}{Z_a} = -\frac{yV_Z}{Z_a} \quad (7)$$

From these equations, while tracking it is possible to compute the ratio of the speed of approach V_Z to

the distance Z_a of the object from the sensor plane provided we can estimate the correct v_x and v_y values at the sensor plane location (x, y) . This ratio represents the theoretical limit of what can be reconstructed about target 3D motion from a 2D flow field (Koenderink and van Doorn, 1987). For this reason, tracking an object in the visual field may solve most of the problems that biological collision avoidance models are faced with.

4 Methods

4.1 Software Simulations

We simulated the Rind and STI models, with and without a tracking algorithm, using Matlab (The Mathworks, Natick, MA). The simulation environment was designed to test the response of the models for collision and non-collision scenarios. In all the cases, a single two-dimensional object, oriented parallel to the x - y cartesian plane, moved relative to the observer inside a three-dimensional world. Only the object was visible to the observer.

Simulations took place in an arena of size 45 m length \times 45 m width \times 55 m height. A 40×40 hemispherical array of visual sensor units implementing either the Rind or the STI model was used to compute a response to the moving object. This array spanned 180° in both azimuth and elevation. The outputs of both models are scalar values which are updated at each timestep. In both cases, a decision to make an avoidance turn must be made based on the time course of these values. 1 ms was used as a timestep in all simulations, resulting in a effective frame rate of 1000 Hz.

The size of the object was chosen to be 2.2 m \times 2.2 m. For simulations of the Rind and STI models with tracking, the sensor array was kept pointing towards the approaching object as long as the object was in front of it. The sensor plane was allowed to rotate only $\pm 45^\circ$ about either axis, and so could not turn backwards to follow the object after it passed the observer. The orientation of the sensor array was determined by computing the centroid of the intensity map of the visual field using a feed-forward linear system. The tracking algorithm computed the centroid (x_c, y_c) of object position on the imaging array using the following formulae:

$$x_c = \frac{\sum_i x_i \cdot \sum_j E(x_i, y_j)}{\sum_i \sum_j E(x_i, y_j)} \quad (8)$$

$$y_c = \frac{\sum_i y_i \cdot \sum_j E(x_i, y_j)}{\sum_i \sum_j E(x_i, y_j)} \quad (9)$$

where $E(x_i, y_j)$ is the intensity of a point at image location (x_i, y_j) , i and j span the x and y dimensions of

the sensor plane, and the midpoints of the sensor plane are taken to be the origin.

Rather than perfectly orienting the sensor on the target position, tracking was performed more realistically by adjusting the azimuth and elevation angles of the sensor plane based on the centroid location at each timestep using the following formulae:

$$\theta_{az}(t) = \theta_{az}(t-1) + g_{az} \times x_c \quad (10)$$

$$\theta_{el}(t) = \theta_{el}(t-1) + g_{el} \times y_c \quad (11)$$

where the time instant t refers to the current frame and $(t-1)$ to the last frame, θ_{az} and θ_{el} are azimuth and elevation angles respectively, and g_{az} and g_{el} are the empirically-set gains for the respective cases.

Due to the simplicity of the visual environment, this algorithm computed an exact location of the target as long as the object remained in the visual field of the observer. We terminated the simulation whenever the object went outside the visual field of the observer, either because the sensor plane had to turn more than $\pm 45^\circ$ or if the object hit one of the walls in the 3D arena. The projected angular size of the object on the sensor plane was updated at each timestep such that its size grew inversely with its distance from the sensor array.

At the start of a simulation run, the simulated object was assigned a starting (x, y, z) position. The z position of the object specified the distance of the object from the viewer along a direction normal to the sensor plane. The x and y positions were within ± 5.6 m from the center of the visual field. The z positions were between 44.8 m and 56 m from the observer. The speed of the object along all three cartesian axes was then generated from a uniform random distribution. The range of speeds was chosen from three different sets for the x and the y axes. The first set was chosen between ± 1.68 m/s, the second between ± 2.24 m/s, and third between ± 3.36 m/s. The speed along the z axis varied from -2 to -20 m/s, with the negative sign of the z speed denoting that the object always approached the viewing plane. A total of 1500 simulations were performed for each model. Due to the random generation of the speed, most trajectories resulted in non-collision scenarios. The object was considered close to an imminent collision at a distance less than 5 m from the observer (about twice the size of the object). The peak response (R_{peak}) of the model output was recorded along with the object distance (d_p) at which the response peaked and the minimum distance (d_m) the object reached to the observer. We histogrammed model peak response data against distance in 1.1 m bins and computed the mean (\bar{R}_{peak}) and standard deviation (σ_R) in each bin. \bar{R}_{peak} and σ_R were plotted versus both d_p and d_m . The

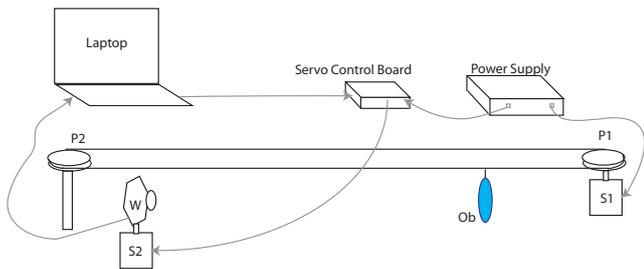


Fig. 6 Experimental setup of the collision detection experiment. Two pulleys (P1 and P2) control the trajectory of the obstacle (Ob), a blue colored disc. Pulley P1 was mounted on a DC motor (S1) and the speed of the obstacle was set by varying the voltage of a power supply. A webcam (W) mounted on a servo motor (S2) sent visual input to a laptop. The laptop runs custom developed software to process the visual information and detect an impending collision. For the models with tracking, the laptop sent control signals to the servo motor S2 via a servo control board.

data from the models with tracking capability were fit with an inverse distance equation for all but one case:

$$\bar{R}_{peak} = \frac{K_1}{d - d_0} \quad (12)$$

where d is either d_p or d_m , and K_1 and d_0 are fitting constants.

For the Rind model with tracking, \bar{R}_{peak} versus d_p curve was linear and was fitted with the equation of a line:

$$\bar{R}_{peak} = m \cdot d_p + c \quad (13)$$

where m is the slope and c is a fitting constant.

In Section 5.5, we present a case of noisy tracking to examine how sensitive the performance of the models with tracking capability are to noise in centroid computation. This was done by adding a Gaussian-distributed noise signal (mean = 0, variance = 8 pixels) to the centroid output of the tracking algorithm. We must note that the size of the sensor array is 40×40 and a variance of 8 is 20% of its size. Therefore, these simulations represent a very unreliable tracking algorithm. These simulations were performed with x and y speeds randomly chosen from a uniform distribution with speed between ± 2.24 m/s. The z speed varied from -2.68 to -13.44 m/s and a total of 500 simulation runs were performed for both the Rind and STI models with tracking.

4.2 Implementation of collision models in a physical system

In addition to simulations, a physical system was developed to test the collision avoidance models. All models were tested for collision and non-collision scenarios. The tracking-capable collision detection models were

implemented so as to keep the approaching object in the center of the visual field. The setup of the experiment is shown in Figure 6. The hardware comprises a web-camera (a Labtec webcam) as a visual sensor, a servo motor (Futaba HS700BB) connected at the base of the webcam for tracking, a pulley and DC motor arrangement from which the obstacle is suspended, and an Acer laptop (Microsoft Windows XP OS, 1.66 GHz, dual-core Intel Centrino processor) for processing the models. The collision detection models and tracking algorithms were implemented in software using the Visual C++ package (Visual Studio, Microsoft).

The setup shows a DC motor and pulley arrangement that allows for repeatability of experiments and was used to control the speed at which the object translates in the visual field. This was done by adjusting a variable power supply which controlled the rotational speed of the pulleys. The object was suspended from a string which went around the two pulleys (see Figure 6). The motion of the string added an additional swinging movement (parallel to the camera plane) to the object, which was random in nature. This made each run of an experiment slightly different even when the speed of the servo and pulley system were the same. The swinging motion also affected the orientation of the object with respect to the ceiling lights that were used for illumination. This in turn affected the contrast of the object as it moved and added more complexity and randomness to each experimental run.

The collision detection system used a webcam as a visual sensor. The field of view of the camera was $\pm 23^\circ$ about an axis normal to its sensor plane. The webcam could operate at up to 20 frames per second (fps) while capturing images of size 352×288 . In order to process images at 15 fps, it was necessary to downsample each frame by a factor of two such that the models operated on an image size of 176×144 .

We implemented a simple tracking algorithm to follow a specifically colored (blue) object by using a centroid detection scheme. A more sophisticated tracking algorithm could be used to tackle more general scenarios (Higgins and Pant, 2004). The simple algorithm is beneficial in that it does not critically affect the computational speed of the system, and still provides reasonable performance to compare the operation of the models with and without tracking capability. The algorithm tracks an object by utilizing a servo motor connected at the webcam's base. This single servo motor allows for only one degree of freedom for the webcam: rotation about its vertical axis. The input frame from the webcam has three color channels: red, green, and blue. We used only the blue color plane to track the object within the frame. The object itself was chosen

to be a bright blue colored disc. Since the webcam can only rotate about its vertical axis, the tracking algorithm computed a positional parameter based on the centroid of the blue pixels in the blue channel of every frame. The centroid was computed by using the following formula:

$$x_c = \frac{\sum_i x_i \cdot \sum_j T(I(x_i, y_j))}{\sum_i \sum_j T(I(x_i, y_j))} \quad (14)$$

where $I(x_i, y_j)$ is the intensity of the blue pixel at image location (x_i, y_j) , and the thresholding function $T(u)$ is defined by

$$T(u) = \begin{cases} u & u > 0.95I_{max} \\ 0 & u \leq 0.95I_{max} \end{cases} \quad (15)$$

with I_{max} as the maximum intensity in the entire blue channel of the image.

This centroid value was then processed by a PD (proportional-derivative) controller treating $x_e = (x_c - x_m)$ as an error signal, where x_m is the center of the image along the horizontal axis and is considered the origin. The differential error signal \dot{x}_e was generated by taking the difference between the error signal of the current and previous frame. The output signal S_{cnt} was generated by using the following relationship:

$$S_{cnt} = K_P \cdot x_e + K_D \cdot \dot{x}_e \quad (16)$$

where K_P and K_D are proportional and derivative gain constants, respectively. The values of K_P and K_D were experimentally determined. A servo control board (USB 16-servo controller, Pololu Corporation) was used to send this signal to the servo motor that controlled the orientation of the webcam.

The response of the models with tracking were affected by the frame rate at which the webcam captured images. A frame rate of 15 fps was not sufficient to show large motions near the camera in a smooth manner. This caused large adjustment in the camera angle to track the object. This jerkiness in the motion of the camera detracted from the response of the model.

In an actual collision avoidance system, the decision to make a course correction has to be made at some point. We have used the peak value from the running average of the response to discriminate collision versus non-collision. The length of the running average window was set to three frames. This peak running-average was used to compare the performance of the models.

5 Simulation results

In our investigations, we observed the peak response of the two models and the corresponding distance between the object and the observer. The aim was to analyze whether or not a thresholding scheme might be

employed with a collision detection model to judge an imminent collision.

5.1 Rind model without tracking

In this set of simulations a single 2D object approached the viewing plane from different starting positions within the virtual arena as described in Methods. Figure 7a shows the mean and standard deviation of the peak response (\bar{R}_{peak} and σ_R) of the Rind model versus the minimum distance the object ever reached from the observer. The plot was truncated to show data only nearer than 25 m from the observer. For distances greater than 25 m, the peak response was close to zero. The variances (σ_R^2) for distances smaller than 7 m from the observer are 72.5% to 113% of the mean \bar{R}_{peak} . These large variances suggest that the model often responded more weakly to a closer object than it did to a farther one.

Another way of looking at the data is by plotting \bar{R}_{peak} versus the distance at which that maximum response occurred. Figure 7b shows these data. We again note that for distances up to 12.5 m from the observer, the variance is sometimes larger than the \bar{R}_{peak} . The first data points in Figure 7b and 7c do not show any variance because they represent two individual cases and, therefore, we do not include them in our analysis. For the case of an imminent collision (distance from the observer < 5 m, shown by a dashed vertical line in the figure), the variance of the model is 1.87 times \bar{R}_{peak} and overlaps with the peak response values at distances greater than 10 m. The large variance of the peak response makes the determination of a simple threshold upon which to make a collision avoidance maneuver impossible in this case.

5.2 STI model without tracking

The STI model was simulated utilizing the same visual setup used for the Rind model. Figure 8a shows \bar{R}_{peak} and σ_R of the STI model versus the minimum distance from the observer. Figure 8b show \bar{R}_{peak} and σ_R versus the distance at which that response was elicited. As before, for distances greater than 25 m, the peak response was close to zero. Similar to the Rind model, we find that the \bar{R}_{peak} of the STI model has large variance. The first two data points in Figure 8b do not show any variance because they represent two individual cases and, therefore, we do not include them in our analysis. For the case of an imminent collision (distance from the observer < 5 m, shown by a dashed vertical line in the figure), σ_R is as large as 53% to 71% of the \bar{R}_{peak} value.

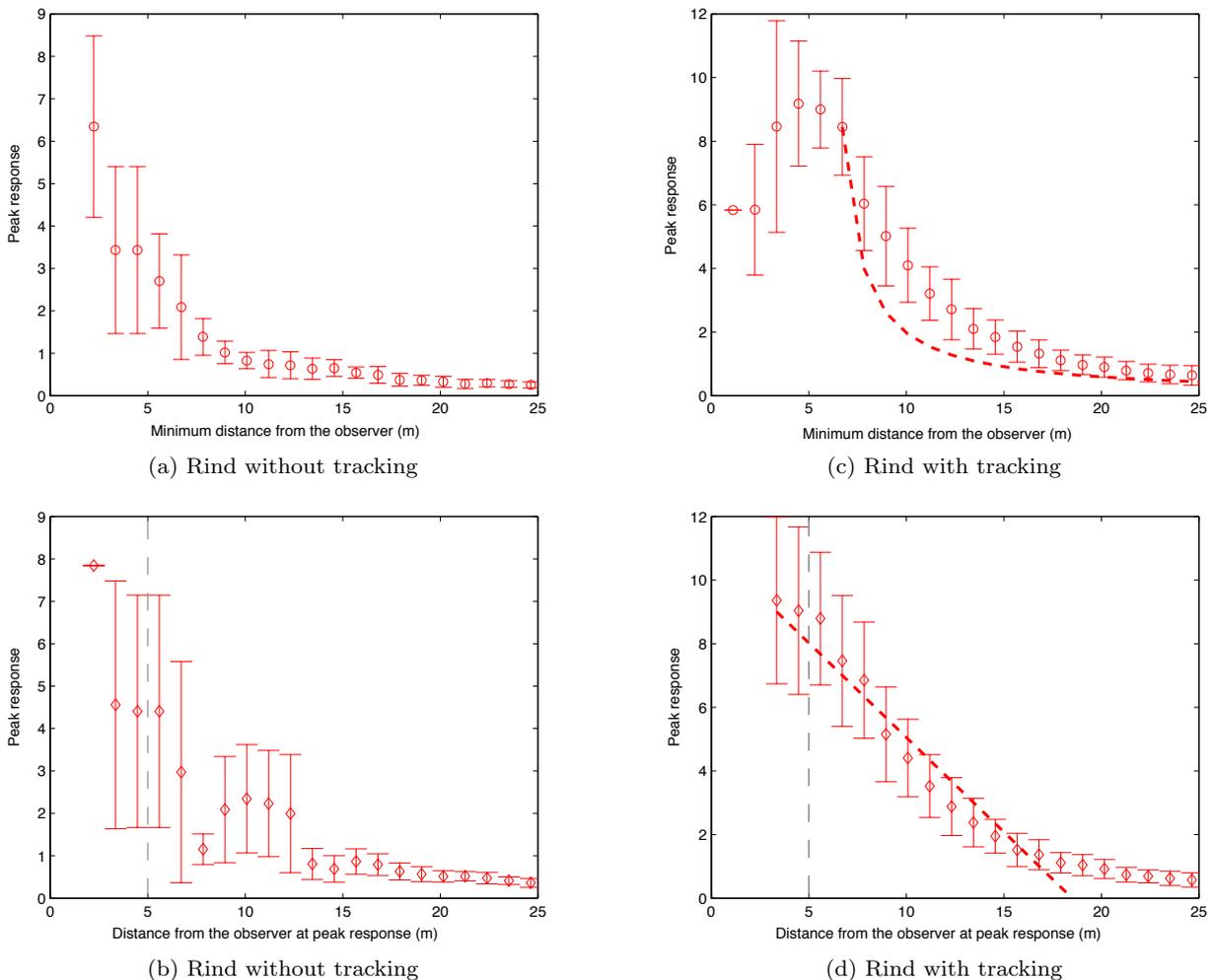


Fig. 7 Mean peak response \bar{R}_{peak} (denoted by circles) and standard deviation σ_R (denoted by error bars) of the Rind model versus the distance from the observer. For comparison, panels a and b are without tracking; panels c and d are with tracking. Panels (a) and (c): \bar{R}_{peak} of the Rind model versus the minimum distance from the observer the object ever reached. Panels (b) and (d): \bar{R}_{peak} of the Rind model versus the distance from the observer at which the peak response was attained. The dashed vertical line denotes the distance at which collision is imminent. The thick dashed traces in panels c and d are theoretical fits as described in the text.

The variance of \bar{R}_{peak} for these distances can be seen to overlap with the variance at distances larger than 10 m. Again, a simple threshold determination for the STI model for all possible collision scenarios is impossible.

5.3 Rind model with tracking

We simulated the Rind model equipped with a tracking algorithm as described in Methods. The simulation sets were the same as those used for testing the Rind model without tracking. \bar{R}_{peak} and σ_R for the Rind model with tracking versus the minimum distance from the object are shown in Figure 7c. \bar{R}_{peak} values reach their maximum at a distance of 4.48 m from the observer, and

then unexpectedly decrease (see below). We have used Equation 12 to fit the data up to the peak with fitting parameters $K_1 = 8.45$ and $d_0 = 5.72$ m.

To better understand why we see a decrease in response peak when objects are closest in Figure 7c, we plotted \bar{R}_{peak} versus the distance at which the response was recorded as shown in Figure 7d. We find that \bar{R}_{peak} increases linearly as a function of distance. A linear fit as described by Equation 13 with fitting parameters $m = 0.5945$ and $c = 11$ m is also plotted in the figure (dashed line). This plot does not show a decrease at small object distances, which means that objects that came close may have been the ones that were passing close by the observer but not actually threatening a collision. In non-collision cases, the Rind model has a de-

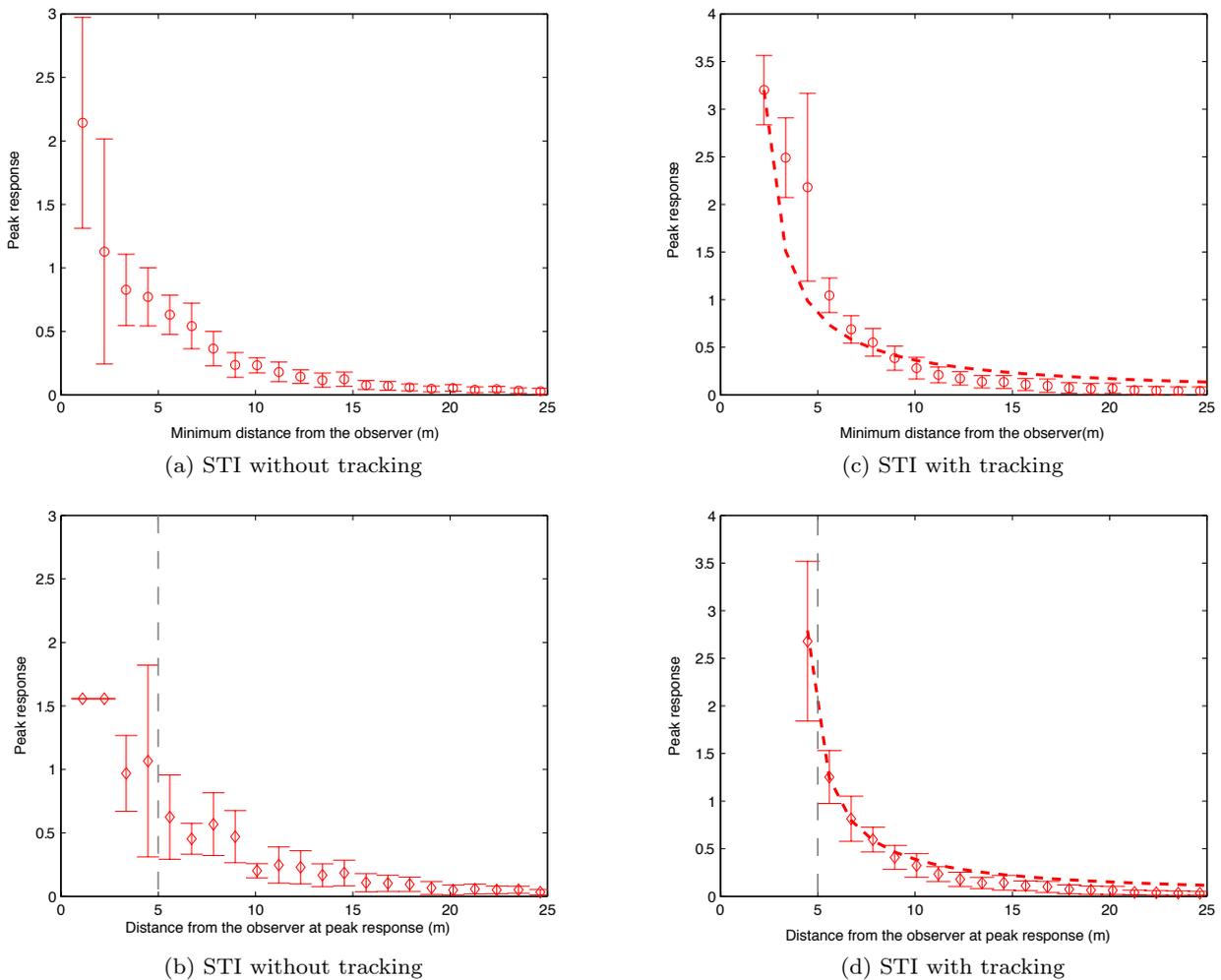


Fig. 8 Mean peak response \bar{R}_{peak} (denoted by circles) and standard deviation σ_R (denoted by error bars) of the STI model versus the distance from the observer. For comparison, panels a and b are without tracking; panels c and d are with tracking. Panels (a) and (c): \bar{R}_{peak} of the STI model versus the minimum distance from the observer the object ever reached. Panels (b) and (d): \bar{R}_{peak} of the STI model versus the distance from the observer at which the peak response was attained. The dashed vertical line denotes the distance at which collision is imminent. The thick dashed traces in panels (c) and (d) represent theoretical fits as described in the text.

creased response. The tracking algorithm allows the rotation of the sensor plane up to an angle of $\pm 45^\circ$ beyond which the simulation is terminated and the approaching object is deemed not a threat. These cases are responsible for the low values of \bar{R}_{peak} at distances less than 5 m in Figure 7c. However, the linear trend of \bar{R}_{peak} with large variance shows that even though the model performs much better than the one without tracking, the determination of the threshold is not trivial. The threshold may be set at an \bar{R}_{peak} value of 6. However, this may raise false alarms for some objects as far away as 8.96 m. We must note that we simulated scenarios that varied in approach speeds by almost one order of magnitude (2 m/s to 20 m/s) at random approach an-

gles, and even though the response of the Rind model with tracking is far from ideal, it is still usable.

5.4 STI model with tracking

In this set of simulations, we simulated an STI model with tracking. The simulation parameters were the same as the ones used in simulating the STI model without tracking. The plot of \bar{R}_{peak} and σ_R for the STI model with tracking versus the minimum distance from the object is shown in Figure 8c. \bar{R}_{peak} closely follows an inverse relation as seen in Equation 12 with fitting parameters $K_1 = 3.2$ and $d_0 = 1.24$ m.

In Figure 8d, \bar{R}_{peak} is plotted with the distance at which the response occurred. The data almost exactly follows the above mentioned equation with fitting parameters $K_1 = 2.5$ and $d_0 = 3.584$ m as shown by the dashed trend line in the figure. We find that even though the variance for the maximum \bar{R}_{peak} is large (distance < 5 m), it does not overlap with the range of values for distances outside that range. Therefore, the STI model with tracking may be used as a reliable threshold based collision avoidance system and the improvement with respect to the STI model without tracking is dramatic. Figure 8d shows that if the threshold is set at 1.6, all objects that pose a threat of imminent collision (distance < 5 m) will be detected. The better performance of the STI model with tracking with respect to the Rind model with tracking is due to its explicit computation of expansive motion. Once the tracking model neutralizes any motion in the x - y plane, the expansive motion computed on the projected image exactly estimates the motion towards the viewing plane as discussed in Section 3.2.

5.5 Effect of noisy tracking

In all the simulations with tracking discussed above, the tracking algorithm was able to exactly compute the position of the approaching object. This might not necessarily be achievable in a complex real-world scenarios with a non-ideal tracking algorithm. To test whether a less accurate tracking algorithm would be as effective as an exact algorithm, we introduced measurement noise in the centroid computation as detailed in Section 4. The results from this set of simulations for both the Rind and the STI model with tracking are shown in Figure 9a and 9b, respectively. The plots indicate that the imprecise tracking algorithm was still sufficient to make the response of the collision avoidance models respond inversely with the distance from the object. The \bar{R}_{peak} values for the Rind model with tracking in this case almost follow an inverse relation with distance as per Equation 12, with fitting parameters $K_1 = 25$ and $d_0 = 4.6$ m. For the STI model with tracking, \bar{R}_{peak} also follows the same equation with fitting parameters $K_1 = 3$ and $d_0 = 2.36$ m. We note that the STI model with tracking is more affected by the noisy tracking algorithm than the Rind model with tracking. This is expected because the response of the STI model with tracking relies totally on the tracking parameters to cancel the effect of motion parallel to the sensor plane. The Rind model with tracking on the other hand has a local inhibitory network that can suppress weak horizontal motion due to noise in tracking.

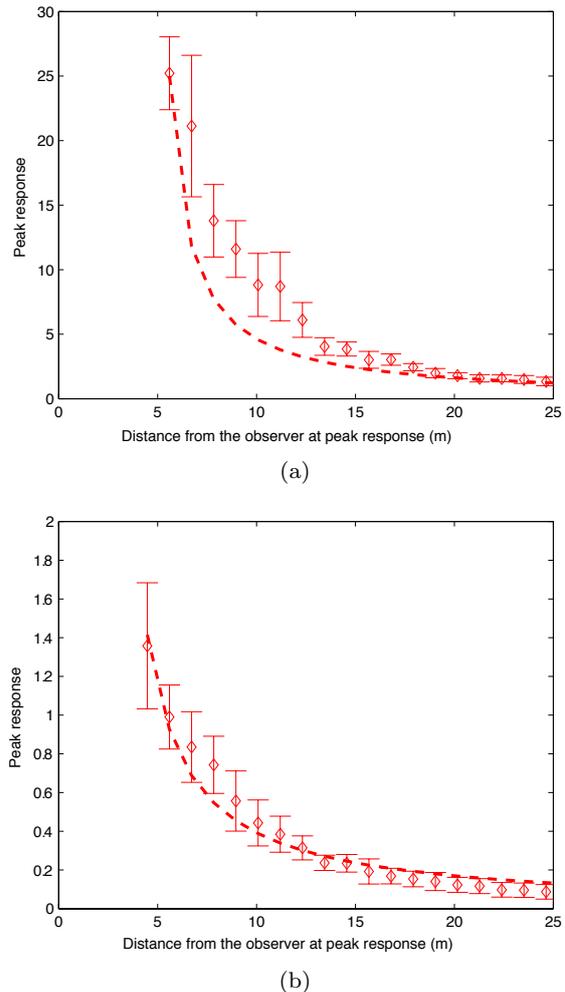


Fig. 9 Mean peak response \bar{R}_{peak} (denoted by circle) and standard deviation σ_R (denoted by error bar) versus the distance from the sensor plane for noisy tracking. The dashed line in both the cases represents respective non-linear fits as described in the text. (a) \bar{R}_{peak} versus the distance from the observer at which the peak response was attained for the Rind model with a noisy tracking algorithm. (b) \bar{R}_{peak} versus the distance from the observer at which the peak response was attained for the STI model with a noisy tracking algorithm.

6 Physical implementation of collision avoidance models

The simulations described above show that the response of the Rind and the STI models with tracking increase inversely with the distance between the object and the observer. To verify our simulation results in more practical scenarios, we implemented these models into a physical collision avoidance system. Results are only shown for the STI model, but qualitatively similar results were obtained for the Rind model.

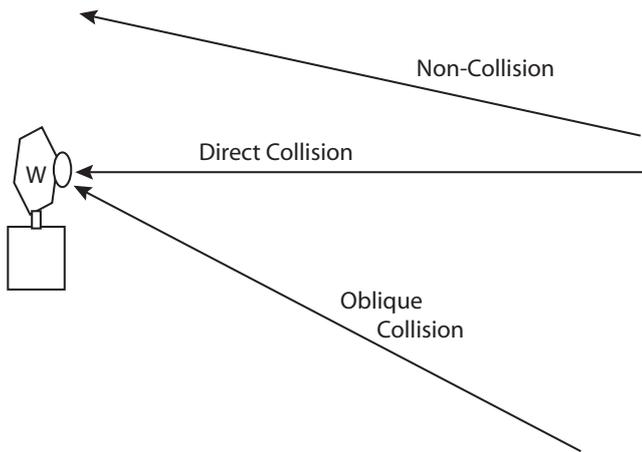


Fig. 10 Different approach trajectories of an object. The lines indicate a direct collision, an oblique collision, and a non-collision approach towards a webcam (W).

6.1 The STI model

The STI model with and without tracking was evaluated under the same test conditions. The object (a blue disc) was made to move at a speed of 17 cm/sec. The flat surface of the disc was made to face the webcam at the start of every experiment. However, any small change in the orientation due to the self-motion of the object was not corrected for. We first present response traces for typical direct-collision, oblique-collision, and non-collision cases as diagrammed in Figure 10.

Response traces were computed by taking an average of ten direct-collision approach cases. In Figure 11a, the thin line shows the average response of the STI model without tracking. The thick line shows the response of the STI model with tracking. As expected, the responses for both with and without tracking cases are almost the same since the object is almost always in the center of the visual field. The slight difference in the two traces originates due to the random swinging of the object as it approaches the webcam. For the model with tracking, the camera moves to compensate this motion and thereby affects the response. The response peaks at roughly 1.2 seconds before the collision at which instant the object occupies the entire visual field of the webcam. The dashed line in the figure shows the distance of the object from the webcam.

Figure 11b shows the average response for the object approaching at a slightly oblique trajectory. Ten experiments were performed for the STI model with and without tracking and the response was averaged. The angle of approach was 10° off the line normal to the camera plane. Some part of the object was always within the visual field of the camera throughout its approach. The thin line in the figure once again represents

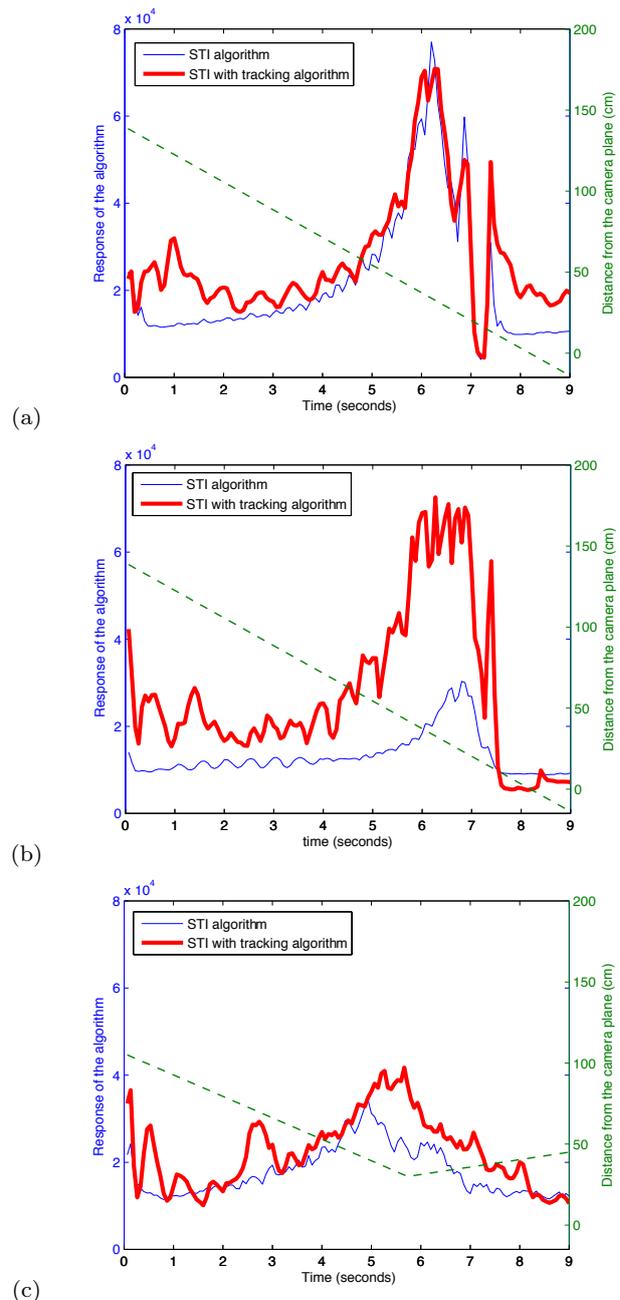


Fig. 11 Average response of the STI model with and without tracking. The thin line represents the frame-by-frame response of the simple STI model; thick line represents the response of the STI model with tracking. The response magnitude is shown on the left y-axis. The dashed line shows the distance of the obstacle from the camera plane as a function of time. The distance scale is shown on the right y-axis. Negative distances mean that the object is behind the camera. (a) Response for a direct-collision scenario. The response tracks the approach of the obstacle and peaks roughly a 1.2 sec before collision. (b) Response for an oblique approach scenario, with the angle of approach set at 10° . Note that the response of the model with tracking is similar to the direct-collision case, while it is much diminished for the non-tracking STI model. (c) Response for a non-collision scenario. The responses of the model for both with and without tracking cases are significantly smaller than the direct collision cases.

the response of the STI model without tracking. The thick line represents the response of the same model with tracking capability. It is clear that even with a slightly oblique approach, the response of the simple STI model is much smaller as compared to a direct collision approach. However, the response trace for the STI model with tracking is similar to the case of direct collision trajectory. This is because the webcam was able to center the approaching object in its visual field for the STI model with tracking, thereby increasing the net expansive motion seen by the camera. The peak in this case is more sustained because of the camera motion which increased the response of the model with tracking.

We compare the above responses with the average response to a non-collision case. Again, ten experiments were performed for each case and the response traces were averaged. The trajectory was set such that the object was visible in the view field for virtually the entire length of the experiment as shown in Figure 10. It entered the view field from the right side and exited from the left, and was always at a distance of ≥ 30 cm. Figure 11c shows the response traces. We find that the response of the STI model with tracking (thick line) is greater in magnitude when compared with the STI model (thin line). However, compared to the responses for direct and oblique collision, these responses were much smaller in magnitude.

Next, we compared the STI model with and without tracking quantitatively by using peak running average (as described in Methods) as the figure of merit for various collision and non-collision scenarios. Is it possible to set a threshold to reliably discriminate collision from non-collision cases without raising a false alarm? Figure 12 shows the mean and standard deviation of the peak running-average values recorded for multiple experiments with the STI model with and without tracking. The leftmost pair of bars in the figure show the data for non-collision experiments. Only ten experiments were conducted for each case because the standard deviation of the peak running-average responses was small. The object translated at a speed of 17 cm/sec in a trajectory such that it was always at a distance ≥ 30 cm. The mean response of the STI model with tracking (gray bar) is about twice as large as the response of the non-tracking model (white bar). The response of the model with tracking is larger because of the motion of the camera while tracking.

The second pair of bars from the left in Figure 12 show the mean and standard deviation for ten direct-collision experiments each. The responses for both the tracking and without tracking cases are comparable for this set. The mean of the peak running-average values

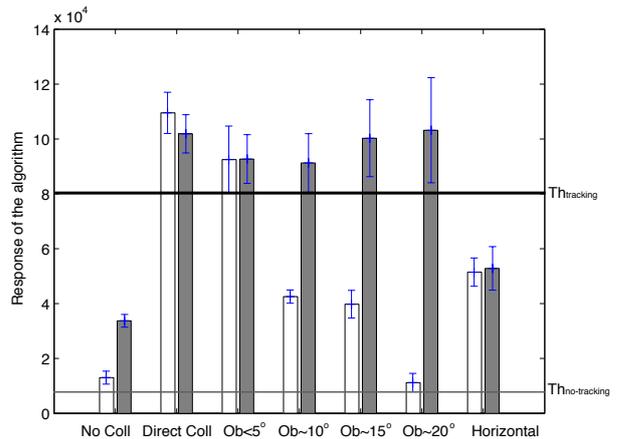


Fig. 12 Experimental data from the STI model with and without tracking for seven scenarios. The white bars show data for the STI model without tracking, while the gray bars show data for the STI model with tracking. Standard deviation is indicated for each experiment. The thick line shows the minimum threshold that needs to be set to detect all collision scenarios for the STI model with tracking, while the thin line shows the minimum threshold for the STI model without tracking. The leftmost pair of bars show the mean response and the standard deviation for non-collision cases. The second pair from the left shows the mean and standard deviation for direct-collision cases. The third pair shows the response for an oblique trajectory where the angle of approach was within $\pm 5^\circ$ of the camera axis. The fourth pair shows the mean response for an oblique approach with the angle of approach between 8° - 10° off the camera axis. The fifth and sixth pair show the responses at 15° and 20° , respectively. The rightmost pair shows the mean and standard deviation of the response for a horizontally moving obstacle.

for the direct-collision case is at least three times that for the non-collision case.

The third pair of bars in Figure 12 show the mean values of the peak running-average response of the model for an oblique trajectory where the starting position of the object was within $\pm 5^\circ$ of the axis normal to the sensor plane. Each model was tested for 20 experiments because the standard deviation for oblique trajectories was larger than the standard deviation for direct and no collision trajectories. The responses for the model with and without tracking are not too different from each other and are comparable to the mean values of the direct-collision trajectories. In this case, the angle of approach is narrow and during the last phase of approach, the obstacle is almost in the center of the visual field and hence elicits a strong response for both the cases. However, if the angle of approach is slightly increased to be between 8° and 10° off-axis, the mean of the peak running-average values are very different for the tracking and non-tracking cases as seen in the fourth pair of bars in Figure 12. Data were collected

from 20 experiments for each case. The mean of the peak running-average value for the STI model without tracking reduced by almost half of its value as compared to the case of oblique trajectories within 5° of the normal axis. The mean value of the STI model with tracking, on the other hand, is almost unchanged. This was due to the ability of the model with tracking to keep the object in the center of its visual field, thereby increasing the peak running-average response. This is also witnessed in the fifth and sixth pair of bars in Figure 12, for which the angle of approach was increased to 15° and 20° , respectively. The mean of the peak running-average value for the STI model without tracking for the latter case is even smaller than the no collision case. This was due to the wide angle of approach of the object at the start of the trajectory. The mean of the peak running-average for the STI model with tracking remains almost unchanged from the direct collision case.

Except for the oblique approach case from wide angles (20°), the mean value of the STI model without tracking is still much larger than its mean for the non-collision cases shown by the first pair of bars in Figure 12. This begs a question: is there any other advantage to the STI model with tracking if we can separate most of the collision cases from the non-collision cases by setting a threshold for the STI model without tracking?

To answer this question, we conducted another experiment where the object was made to move horizontally across the visual field at a distance of 20-25 cm from the camera, at a speed of 17 cm/sec. The motion of the object was parallel to the webcam and the object crossed the visual field of the stable camera (in the STI model without tracking) in less than 500 ms. The mean and standard deviation of the peak running-average response for these non-collision experiments is shown by the rightmost pair of bars in Figure 12. The responses for the STI model with and without tracking are almost equal in magnitude. The STI model with tracking should have negated the horizontal motion by following the object across the visual field of the webcam. However, due to the fast motion of the object in the visual field, the webcam was unable to smoothly track the object. This self-induced motion is responsible for the mean peak response of the STI model with tracking being comparable with its non tracking counterpart. This peak response is about half of the mean peak response for all collision cases shown in the figure. The mean value of the peak running-average response for the STI model, on the other hand, is larger than its mean value for the oblique approach at angles wider than 8° !

Since a unique threshold cannot be set for discrimination, the STI model without tracking is unable to reliably detect collisions for approach angles larger than 20° and raises a false alarm when an object moves horizontally across the webcam at fast speeds. The thick horizontal line in the figure shows the minimum threshold that is needed to detect all collisions for the STI model with tracking and the thin horizontal line shows the same for the same model without tracking. It is easy to see that the STI model with tracking is able to distinguish all the collision scenarios from the non-collision ones. However, the STI model without tracking cannot do so without raising false alarms. Tracking increases the capability of the STI model to respond reliably to collision scenarios and at the same time, reduce false alarms. We see that even with a simple tracking algorithm the performance of the STI model improves remarkably and is in agreement with our simulation analysis.

7 Discussion

In this study, we evaluated the limitations of biological collision avoidance models. Optical flow based algorithms have been widely used for this purpose. However, the computation of optical flow is mathematically ill-defined and requires additional assumptions about the visual scene which are not always warranted. The use of correlation-based models in place of optical flow based algorithms is advantageous only to the extent that it presents a mathematically stable and computationally simple alternative (Reichardt *et al.*, 1988; Verri and Poggio, 1989). However, it does not address the issue that the local speed of a point on the sensor plane may not be used to disambiguate between the infinitely many solutions for the corresponding motion in the real world. We see from the simulation results in Section 5 that both the Rind and the STI models fail to reliably detect an impending collision and may lead to false alarms or fail to report an imminent collision. The major reason behind the failure in the Rind model is its inability to reject horizontal motion. In case of the STI model, it is the manner in which it computes expansion by dividing the visual field into four quadrants without regard to the position of the obstacle in it.

The major shortcoming of any collision avoidance model is due to an inherent limitation of representing three-dimensional motion in the real world on a two-dimensional sensor array. The analysis presented here shows that if the response of collision avoidance models can be made dependent on the distance of an obstacle from the observer, this problem may be solved. We have

shown how tracking may be used with the Rind and STI models to relate their responses to distance.

The results from various collision and non-collision experiments clearly illustrate that by letting the visual sensor align itself with an approaching object, the effectiveness of the model is dramatically improved. In our study, the performance of the STI model with tracking was better than the Rind model with tracking. This we attribute largely to two factors. Firstly, the Rind model relies on edge detection and in our implementation, the low frame rate of the camera sometimes blurred the motion which made edge detection difficult. Secondly, the lateral inhibitory network in the Rind model *always* tries to suppress the response magnitude of the model. In order to get a strong response from this model, the activation of its excitatory units must be large enough to swamp the effect of inhibitory units. This goes back to the problem of edge detection. Since the frame rate of our input was not very high, the edge response was weak and so was the response of the excitatory units. The STI model with tracking, on the other hand, does better on both accounts, and in general is a more practical model for collision avoidance.

The primary effect of the fact that our simulation and physical implementation of these collision avoidance algorithms was discrete-time (to be compared to continuous-time processing in brains) was in detection of edges in the physical implementation as described above. Due to the very high frame rate possible in the simulations (1000 Hz), the simulation results are virtually indistinguishable from continuous-time. The limited frame rate of the physical implementation and the relatively low quality of the webcam images were an intentional attempt to determine if either of these biological models make practical algorithms for collision detection. The STI model with tracking performs rather well even in these non-ideal circumstances.

It should be noted that in all our simulation experiments there was no background visible to the observer (only the target was visible), and in our physical implementation only relatively faraway background objects (the laboratory wall, for example) could be seen. This raises the question as to whether the visual motion of the background caused by movement of the camera during tracking could interfere with collision detection performance. The apparent visual motion due to rotation of the camera is independent of distance to the object (Gibson, 1950), so rapid camera movement during tracking could generate large apparent motion regardless of distance to the background objects. Our physical implementation addresses this question implicitly in that an unaltered laboratory background was used without significant effects on the results as com-

pared to simulation. Additionally, our tracking control system was designed to track as smoothly as possible, thus eliminating unnecessary camera motion artifacts. However, a highly textured and/or high-contrast background would clearly interfere more with collision detection performance than any background we used in our experiments, and we have not explicitly experimented with this.

The two biological models of insect collision avoidance we have analyzed in this study are well accepted by the biological community but fail to perform well when used without tracking in practical collision scenarios. Despite the lack of evidence that insects actively direct their gaze to looming objects, this fact suggests that insects and other organisms may alter their gaze to fixate on looming objects, which based on our results will dramatically improve their ability to distinguish collision from non-collision scenarios.

Acknowledgements This work was partially supported by a grant from the National Institutes of Health (National Center for Research Resources, grant number 5R01RR008688-21).

References

- Aloimonos, Y., I. Weiss, and A. Bandopadhyay (1988). Active vision. *International Journal of Computer Vision* 2: 333–356.
- Bajcsy, R. (1985). Active perception vs. passive perception. In *Proc. IEEE Workshop on Computer Vision*, pp. 55–59.
- Bandopadhyay, A. and D.H. Ballard (1991). Egomotion perception using visual tracking. *Computational Intelligence* 7: 39–47.
- Boeddeker, N., L. Dittmar, W. Strzyl, and M. Egelhaaf (2010). The fine structure of honeybee head and body yaw movements in a homing task. *Proceedings of the Royal Society of London B* 277(1689): 1899–1906.
- Boeddeker, N. and J.M. Hemmi (2010). Visual gaze control during peering flight manoeuvres in honeybees. *Proc. R. Soc. B* 277: 1209–1217.
- Borst, A. (1990). How do flies land? *Bio-science* 40(2): 292–299.
- Borst, A. and S. Bahde (1986). What kind of movement detector is triggering the landing response of the housefly. *Biol. Cybern.* 55: 59–69.
- Bruckstein, A., R.J. Holt, I. Katsman, and E. Rivlin (2005). Head movements for depth perception: Praying mantis versus pigeon. *Autonomous Robots* 18: 21–42.
- Daniilidis, K. (1997). Fixation simplifies 3D motion estimation. *Comp. Vis. & Image Understanding* 68(2): 158–169.
- Duric, Z., A. Rosenfeld, and J. Duncan (1999). The applicability of Greens theorem to computation of rate of approach. *International Journal of Computer Vision* 31(1): 83–98.
- Eckert, M.P. and G. Buchsbaum (1993). Effect of tracking strategies on the velocity structure of two dimensional image sequences. *J. Opt. Soc. Am. A* 10(7): 1582–1585.
- Fermuller, C. and Y. Aloimonos (1992). Tracking facilitates 3-d motion estimation. *Biol. Cybern.* 67: 259–268.
- Foley, J.D., J. Hughes, A. van Dam, and S. Feiner (1995). *Computer Graphics: Principles and Practice*. Addison-Wesley Professional.

- Franz, M. O. and H. G. Krapp (2000). Wide-field, motion-sensitive neurons and matched filters for optic flow fields. *Biol. Cybern.* 83: 185–197.
- Franz, M.O. and H.G. Krapp (1998). Wide-field, motion-sensitive neurons and optimal matched filters for optic flow. *Biological Cybernetics* 83(3).
- Gibson, J. J. (1950). *The Perception of the Visual World*. Houghton Mifflin Co., Boston.
- Harrison, R.R. (2005). A biologically inspired analog IC for visual collision detection. *IEEE Transactions on Circuits and Systems-I* 52: 2308–2318.
- Hassenstein, B. and W. Reichardt (1956). Systemtheoretische analyse der Zeit-, Reihenfolgen- und Vorzeichenbewertung bei der Bewegungserkennung des Rüsselkäfers *Chlorophanus*. *Zeitschrift für Naturforschung* 11b: 513–524.
- Hatsopoulos, N, F Gabbiani, and G Laurent (1995). Elementary computation of object approach by a wide-field visual neuron. *Science* 270: 1000–1003.
- Hemmi, J.M. and D. Tomsic (2011). The neuroethology of escape in crabs: from sensory ecology to neurons and back. *Current Opinion in Neurobiology* 22: 1–7.
- Higgins, C.M. and V. Pant (2004). An elaborated model of fly small target tracking. *Biological Cybernetics* 91(6): 417–428.
- Horn, B. (1986). *Robot Vision*, chapter 12. MIT Press, Cambridge, MA.
- Horn, B.K.P. and B.G. Schunck (1981). Determining optical flow. *Artificial Intelligence* 17: 185–203.
- Hyslop, A. M. and J. S. Humbert (2010). Autonomous navigation in three-dimensional urban environments using wide-field integration of optic flow. *Journal of Guidance, Control, and Dynamics* 33(1): 147–159.
- Koch, C., J. Marroquin, and A. Yuille (1986). Analog “neuronal” networks in early vision. *Proc. Natl. Acad. Sci., USA* 83: 4263–4267.
- Koenderink, J.J. and A. J. van Doorn (1987). Facts on optic flow. *Biol. Cybern.* 56(4): 247–254.
- Krapp, H.G., B. Hengstenberg, and R. Hengstenberg (1998). Dendritic structure and receptive-field organization of optic flow processing interneurons in the fly. *J. Neurophysiol.* 79: 1902–1917.
- Land, M.F. (1973). Head movements of flies during visually guided flight. *Nature* 243: 299–300.
- Laurent, G. and F. Gabbiani (1998). Collision-avoidance: Nature’s many solutions. *Nature Neuroscience* 1(4): 261–263.
- Lindemann, J.P., R. Kern, J.H. Hateren, H. Ritter, and M. Egelhaaf (2005). On the computations analyzing natural optic flow: Quantitative model analysis of the blowfly motion vision pathway. *The Journal of Neuroscience* 25(27): 6435–6448.
- Milde, J.J. and N.J. Strausfeld (1990). Cluster organization and response characteristics of the giant fiber pathway of the blowfly *Calliphora erythrocephala*. *J. Comp. Neurol.* 294: 59–75.
- Pant, V. (2007). Biomimetic visual navigation architectures for autonomous intelligent systems. Ph.D. diss., University of Arizona, Tucson.
- Reichardt, W. (1961). Autocorrelation, a principle for the evaluation of sensory information by the central nervous system. In Rosenblith, WA, editor, *Sensory Communication*, pp. 303–317. MIT Press, New York.
- Reichardt, W., M. Egelhaaf, and R.W. Schlögl (1988). Movement detectors provide sufficient information for local computation of 2-D velocity field. *Naturwiss.* 75: 313–315.
- Reiser, M. and M. Dickinson (2003). A test bed for insect-inspired robotic control. *Phil. Transact. Roy. Soc. Lond. A* 361: 2267–2285.
- Rind, F.C. (1984). A chemical synapse between two motion detecting neurones in the locust brain. *Journal of Experimental Biology* 110(1): 143–167.
- Rind, F.C. and D.I. Bramwell (1996). Neural network based on the input organization of an identified neuron signaling impending collision. *J. Neurophysiology* 75: 967–985.
- Rind, F.C. and P.J. Simmons (1992). Orthopteran DCMD neuron: A reevaluation of response to moving objects. I. Selective responses to approaching objects. *J. Neurophysiology* 68: 1654–1666.
- Schilstra, C. and J.H. van Hateren (1998). Stabilizing gaze in flying blowflies. *Nature* 395: 654.
- Simmons, P.J. and F.C. Rind (1992). Orthopteran DCMD neuron: A reevaluation of response to moving objects. II. Critical cues for detecting approaching objects. *J. Neurophysiology* 68: 1667–1682.
- Stocker, A. (2004). Analog VLSI focal-plane array with dynamic connections for the estimation of piecewise-smooth optical flow. *IEEE Trans. on Circuit and Systems* 51(5): 963–973.
- Sun, H. and B.J. Frost (1998). Computation of different optic variables of looming objects in pigeon nucleus rotundus neurons. *Nat Neurosci* 1: 296–303.
- Tammero, L.F. and M.H. Dickinson (2002). Collision-avoidance and landing responses are mediated by separate pathways in the fruit fly, *Drosophila melanogaster*. *The Journal of Experimental Biology* 205: 2785–2798.
- van Hateren, J.H., R. Kern, G. Schwerdtfeger, and M. Egelhaaf (2005). Function and coding in the blowfly H1 neuron during naturalistic optic flow. *J. Neurosci.* 25(17): 4343–4352.
- van Hateren, J.H. and C. Schilstra (1999). Blowfly flight and optic flow. II. Head movements during flight. *J. Exp. Biol.* 202: 1491–1500.
- Verri, A. and T. Poggio (1989). Motion field and optical-flow - qualitative properties. *IEEE Trans. Pattern Anal. Mach. Intell.* 11: 490–498.
- Warren, W.H. and D.J. Hannon (1990). Eye movements and optical flow. *J. Opt. Soc. Am. A* 7(1): 160–169.
- Wertz, A., B. Gaub, J. Plett, J. Haag, and A. Borst (2009). Robust coding of ego-motion in descending neurons of the fly. *J. Neuroscience* 29(47): 14993–15000.