A Bio-Inspired Model for Visual Collision Avoidance on a Hexapod Walking Robot

Hanno Gerd Meyer[1(✉)], Olivier J.N. Bertrand[2], Jan Paskarbeit[1], Jens Peter Lindemann[2], Axel Schneider[1,3], and Martin Egelhaaf[2]

[1] Biomechatronics, Center of Excellence 'Cognitive Interaction Technology' (CITEC), University of Bielefeld, Bielefeld, Germany
hanno.meyer@uni-bielefeld.de
[2] Department of Neurobiology and Center of Excellence 'Cognitive Interaction Technology' (CITEC), University of Bielefeld, Bielefeld, Germany
[3] Embedded Systems and Biomechatronics Group, Faculty of Engineering and Mathematics, University of Applied Sciences, Bielefeld, Germany

Abstract. While navigating their environments it is essential for autonomous mobile robots to actively avoid collisions with obstacles. Flying insects perform this behavioural task with ease relying mainly on information the visual system provides. Here we implement a bio-inspired collision avoidance algorithm based on the extraction of nearness information from visual motion on the hexapod walking robot platform HECTOR. The algorithm allows HECTOR to navigate cluttered environments while actively avoiding obstacles.

Keywords: Biorobotics · Bio-inspired vision · Collision avoidance · Optic flow · Elementary motion detector

1 Introduction

Compared to man-made machines, insects show in many respects a remarkable behavioural performance despite having only relatively small nervous systems. Such behaviours include *complex flight or walking manoeuvres*, *avoiding collisions*, *approaching targets* or *navigating in cluttered environments* [11]. Sensing and processing of environmental information is a prerequisite for behavioural control in biological as well as in technical systems. An important source of information is *visual motion*, because it provides information about self-motion, moving objects, and also about the 3D-layout of the environment [7].

When an agent moves through a static environment, the resulting visual image displacements (*optic flow*) depend on the speed and direction of ego-motion, but may also be affected by the nearness to objects in the environment. During *translational* movements, the optic flow amplitude is high if the agent moves fast and/or if objects in the environment are close. However, during *rotational* movements, the optic flow amplitude depends solely on the velocity of ego-motion and, thus, is independent of the nearness to objects. Hence, information

N.F. Lepora et al. (Eds.): Living Machines 2016, LNAI 9793, pp. 167–178, 2016.
DOI: 10.1007/978-3-319-42417-0_16

about the depth structure of an environment can be extracted parsimoniously during translational self-motion as distance information is immediately reflected in the optic flow [7]. To solve behavioural tasks, such as *avoiding collisions* or *approaching targets*, information about the depth structure of an environment is necessary. Behavioural studies suggest, that flying insects employ flight strategies which facilitate the neuronal extraction of depth information from optic flow by segregating flight trajectories into translational and usually much shorter rotational phases (*active-gaze-strategy*, [4]). Furthermore, the neuronal processing of optic flow has been shown to play a crucial role in the control of *flight stabilisation*, *object detection*, *visual odometry* and *spatial navigation* [3].

In flying insects, optic flow is estimated by a mechanism that can be modelled by *correlation-type elementary motion detectors* (EMDs, [2]). A characteristic property of EMDs is that the output does not exclusively depend on velocity, but also on the pattern properties of a moving stimulus, such as its contrast and spatial frequency content. Hence, nearness information can not be extracted unambiguously from EMD responses [5]. Rather, the responses of EMDs to pure translational optic flow have been concluded to resemble a representation of the *relative contrast-weighted nearness* to objects in the environment, or, in other words, of the contours of nearby objects [18].

Recently, a simple model for collision avoidance based on EMDs was proposed [1]. The model is based on three successive processing steps: (a) the extraction of (contrast-weighted) nearness information from optic flow by EMDs, (b) the determination of a collision avoidance direction from the map of nearness estimates and (c) the determination of a collision avoidance necessity, i.e. whether to follow (i.e. potential obstacles are close) or not to follow the collision avoidance direction (i.e. potential obstacles are still rather distant). When coupled with a goal direction, the algorithm is able to successfully guide an agent to a goal in cluttered environments without collisions.

In this study, this collision avoidance model was implemented on the insect-inspired hexapod walking robot HECTOR [15]. In contrast to flight, walking imposes specific constraints on the processing of optic flow information. Due to the mechanical coupling of the agent to the ground the perceived image flow is superimposed by continuous rotational components about all axes correlated to the stride-cycle [9]. Therefore, nearness estimation from optic flow during translational walking might be obfuscated, potentially reducing the reliability of the collision avoidance algorithm. Further, in contrast to the 360° panoramic vision used in [1], a fisheye lens was mounted on the front segment of the robot with its main optical axis pointing forward, limiting the field of view for retrieving optic flow information.

In the following, the implementation of the collision avoidance and vision-based direction control on the robotic platform will be described and the performance assessed in artificial and natural cluttered environments in a simulation framework of HECTOR. After optimisation of parameters, HECTOR will be able to successfully navigate to predefined goals in cluttered environments while avoiding collisions.

2 The Simulation Framework

The performance of the model of collision avoidance and direction control was assessed in a dynamics simulation of HECTOR which is coupled with a rendering module. The simulation allows parameter optimisation and tests in different virtual environments. The simulation framework is depicted in Fig. 1 and can be separated into four processing modules: (a) *walking controller*, (b) *robot simulation*, (c) *renderer* and (d) *vision-based directional controller*.

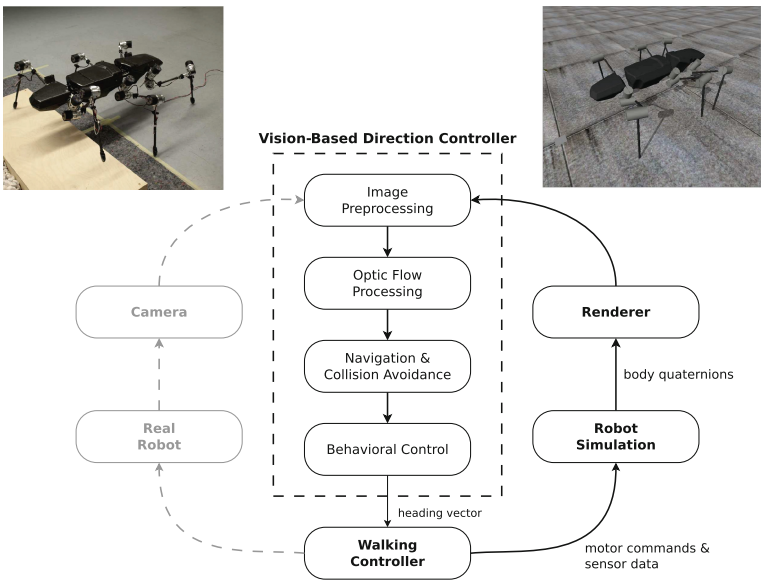

Fig. 1. The simulation framework used for testing the implementation of the visual collision avoidance model on HECTOR. The dashed box (*Vision-Based Direction Controller)* indicates the algorithm used for controlling the robot's behaviour based on nearness estimation from optic flow.

2.1 Robot Simulation and Walking Controller

The hexapod robot HECTOR is inspired by the stick insect *Carausius morosus*. For its design, the relative positions of the legs as well as the orientation of the legs' joint axes have been adopted. The size of the robot was scaled up by a factor of 20 as compared to the biological example which results in an overall length of roughly 0.9 m. This size allows the robot to be used as an integration platform for several hard- and software modules. All 18 drives for the joints of the six legs are serial elastic actuators. The mechanical compliance of the drives is achieved by an integrated, sensorised elastomer coupling [14] and is the foundation for the robot's ability to passively adapt to the structure of the substrate during walking. The bio-inspired walking controller is a conversion of the WALKNET

approach [17] and allows the robot to negotiate rough terrain [15]. Abstracting the complex task of leg coordination, only the heading vector must be provided externally, e.g. by a vision-based direction controller as proposed here.

To simulate a multitude of controller parameters, a dynamics simulation has been set up based on ODE (Open Dynamics Engine) which also simulates the elastic joint actuation. The HECTOR simulator is controlled by the same controller framework as the physical robot.

2.2 Renderer

To obtain optic flow information resulting from ego-motion in different virtual environments the images of a camera attached to the robot's main body are rendered using the graphics engine Panda3D [6]. The robot's orientation and position are obtained from the robot simulation module. To emulate the wide field of view of insect eyes [21], virtual camera images are rendered simulating an equisolid fisheye lens [12]. The lens is parametrised to a horizontal and vertical field of view of 192°. The images obtained have a resolution of 400×400 pixels with a resolution of 10 bit per RGB color channel and a sampling frequency of 20 Hz. Although blowflies possess color vision, evidence suggests that the pathways involved in motion detection are monochromatic [20]. Therefore, only the green color channel is used (Fig. 2A).

Head Stabilisation. During walking, the extraction of distance information on the basis of optic flow processing may be impaired by stride-coupled image shifts. For example, walking blowflies hardly ever show purely translational locomotion phases. Rather, they perform relatively large periodic rotations of their body around all axes due to walking [9]. While stride-induced body rotations around the roll and pitch axes are compensated by counter-rotations of the head, body rotations around the yaw axis are not [8]. To minimise stride-coupled image displacements, movements of the camera around the roll and pitch axis are compensated in simulation. This is achieved by setting the roll and pitch angles of the camera to fixed values independent of the movement of the robot's main body, effectively keeping the center of the optical axis of the camera parallel to the ground plane.

2.3 Vision-Based Direction Controller

The sequences of camera images obtained from the renderer are processed by the vision-based direction controller, which can be subdivided into four processing steps:

(a) *preprocessing of images*, in order to emulate the characteristics of the visual input of flying insects,
(b) estimation of a relative nearness map by *processing of optic flow* via EMDs,
(c) computation of a *collision avoidance direction* based on the relative nearness of objects and a goal direction, and
(d) *controlling the walking direction* of the robot.

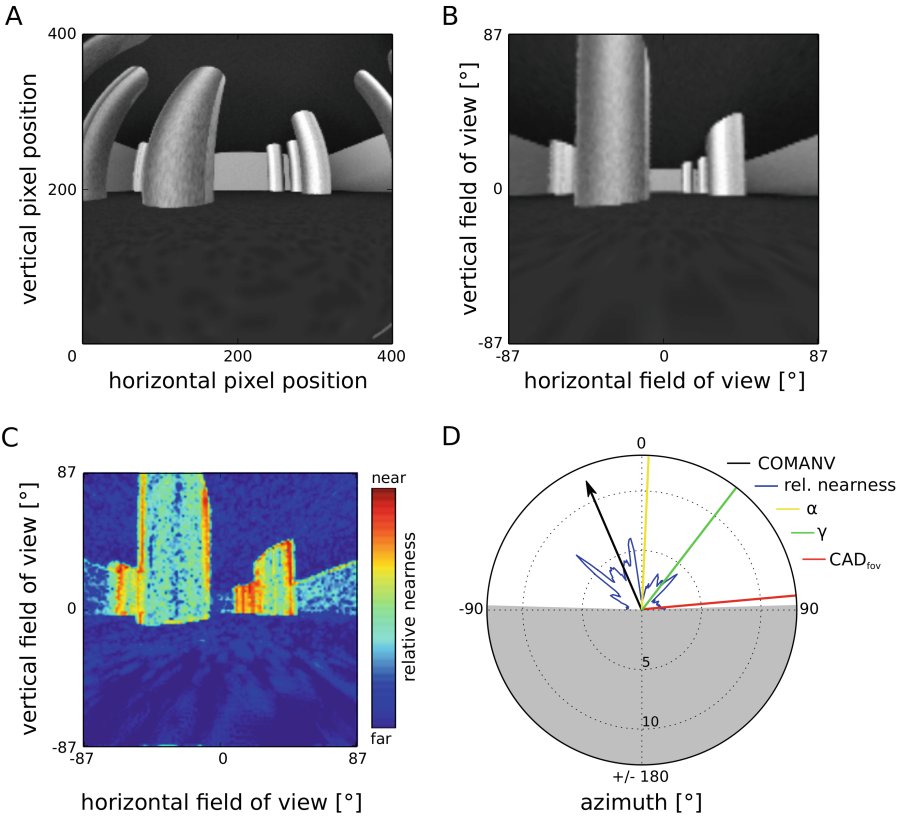

Fig. 2. (A) *Camera image* of a virtual environment rendered with an equisolid fisheye lens. Image has been resized and reduced to 8-bit dynamic range for reproduction. (B) Camera image *remapped to a rectilinear representation*, *spatially filtered* and *scaled to an array of photoreceptors*. Each pixel position represents a luminance value as perceived by the according photoreceptor. Image has been resized and reduced to 8-bit dynamic range for reproduction. (C) *Relative contrast-weighted nearness map* μ_r obtained from optic flow estimation via horizontally and vertically aligned EMDs. The color-code depicts near (red) and far (blue) estimated relative nearnesses. (D) *Polar representation of the relative nearness averaged over the azimuth* (blue). The arrow (black) depicts the sum of nearness vectors ($COMANV$) used for determining the *collision avoidance direction* CAD_{fov} (red). Based on the weighted sum of the direction to a goal α (yellow) and the CAD_{fov}, a heading direction γ (green) is computed. The greyed out area indicates the surrounding of the robot *not* covered by the field of view. (Color figure online)

(a) Image Preprocessing. The non-linear fisheye lens, used here, possesses distortion characteristics which are such that objects along the optical axis of the lens occupy disproportionately large areas of the image. Objects near the periphery occupy a smaller area of the image. Since the lens enlarges objects in the vicinity of the optical axis, those objects are transmitted with much greater detail than objects in the peripheral viewing region, thus, obfuscating nearness estimation from optic flow. Hence, images obtained from the fisheye lens are remapped to a rectilinear representation [12].

The compound eye of insects consists of a two-dimensional array of hexagonally aligned ommatidia comprising the retina. Each ommatidium contains a lens and a set of photoreceptor cells. The lattice of ommatidia has characteristics of a spatial low-pass filter and blurs the retinal image. To mimic the spatial filtering of the eye, the remapped images are filtered by a two-dimensional Gaussian-shaped spatial low-pass filter according to Shoemaker et al. [19]. After spatial filtering, each input image is scaled down to a rectangular grid of photoreceptors, with an interommatidial angle of 1.5°. The acceptance angle of an ommatidium is set to 1.64° in order to approximate the characteristics of the eyes of the blowfly [16]. The grid covers a field of view of 174° horizontally and vertically, resulting in an array of 116×116 photoreceptors (i.e. luminance values) (Fig. 2B).

(b) Optic Flow Processing. In the model used here (see [18]), optic flow estimation is based on two retinotopic arrays of either *horizontally* or *vertically* aligned EMDs. Individual EMDs are implemented by a multiplication of the delayed signal of a receptive input unit with the undelayed signal of a neighbouring unit. Only interactions between direct neighbours are taken into account, for both horizontally and vertically aligned EMDs. The luminance values from the photoreceptors are filtered with a first-order temporal high-pass filter ($\tau_{hp} = 20$ ms) to remove the mean from the overall luminance of the input. The filtered outputs are fed into the horizontally and vertically aligned EMD arrays. The delay operator in each half-detector is modelled by a temporal first-order low-pass filter ($\tau_{lp} = 35$ ms). Each EMD consists of two mirror-symmetric subunits with opposite preferred directions. Their outputs are subtracted from each other. For each retinotopic unit the motion energy is computed by taking the length of the motion vector given by the combination of the responses of a pair of the horizontal h_{EMD} and the vertical v_{EMD} at a given location (x,y) of the visual field:

$$\mu_{r(x,y)} = \sqrt{v^2_{\mathrm{EMD}}(x,y) + h^2_{\mathrm{EMD}}(x,y)} \tag{1}$$

The array of the absolute values of these local motion vectors μ_r resembles a map of *contrast-weighted relative nearness* to objects in the environment [18], providing information about the contours of nearby objects (Fig. 2C).

(c) Navigation and Collision Avoidance. Once the relative nearness map μ_r is known, collision avoidance is achieved by moving away from the maximum nearness value (e.g. objects that are close) (see [1]). However, the contrast-weighted

nearness map also depends on the textural properties of the environment. To reduce the texture dependence, the nearness map is averaged along the elevation ϵ, giving the average nearness for a given azimuth ϕ. Each of these averaged nearness values can be represented by a vector in polar coordinates, where the norm of the vector is the averaged nearness, and its angle corresponds to the azimuth. The sum of these vectors points towards the average direction of close objects (Fig. 2D). This vector is denoted *center-of-mass-average-nearness-vector* ($COMANV$; [1])

$$COMANV = \sum \left(\begin{pmatrix} \cos(\phi) \\ \\ \sin(\phi) \end{pmatrix} \frac{1}{n} \sum \mu_r \left(\epsilon, \phi\right) \right), \tag{2}$$

where n is the number of elements in the azimuth. The inverse of the $COMANV$ vector, scaled to the horizontal field of view θ of the photoreceptor array, points away from the closest object and, thus, can be used as the direction of the robot to avoid collisions (*collision avoidance direction*, CAD_{fov}; Fig. 2D; [1]):

$$CAD_{fov} = \frac{-\arctan\left(COMANV_y, COMANV_x\right)}{\frac{2\pi}{\theta}} \tag{3}$$

The length of the $COMANV$ vector increases with nearness and apparent size of objects. Its length is a measure of the *collision avoidance necessity* (CAN; [1]):

$$CAN = \parallel COMANV \parallel . \tag{4}$$

The CAN measure is used to control the heading direction γ of the robot between *avoiding collisions* and *following the direction to a goal* (α; Fig. 2D) [1]:

$$\gamma = W(CAN) \cdot CAD_{fov} + (1 - W(CAN)) \cdot \alpha \tag{5}$$

W is a sigmoid weighting function based on the CAN:

$$W(CAN) = \frac{1}{1 + \left(\frac{CAN}{n_0}\right)^{-g}}, \tag{6}$$

and driven by a gain g and a threshold n_0 [1].

(d) Behavioural Control of the Walking Robot. The walking direction of the robot is controlled based on the heading direction γ obtained from estimating relative nearness values to objects from optic flow and the goal direction. Information about the spatial structure of an environment can only be extracted from optic flow during translational movements, as rotational flow components do not provide distance information [7]. Inspired by the *active-gaze-strategy* employed by flying insects [4], the control of walking direction is implemented by segregating the motion trajectory into *translational* and *rotational phases*. During translation, the robot moves forward with a constant velocity of 0.2 m/s for 4 s (corresponding to 80 camera images), while averaging the heading direction γ.

After that, the robot switches to a rotational state and turns towards the averaged heading direction γ until the vector is centred in the field of view. When the optic flow field is estimated by EMDs, the nearness estimations also depend on the motion history due to the temporal filters. Hence, the optic flow obtained during the rotational phase interferes with the optic flow measurements during the translational phase. This effect decreases over time. Therefore, the heading direction γ is only averaged for the last 3 s of the translational phase.

Due to the restricted horizontal field of view, no information about the nearness of objects outside of the camera's field of view can be obtained (grey area in Fig. 2D). However, as the camera is pointing forward along the direction of walking during translation, information about the nearness of objects sidewards or behind the robot is not essential. In situations where the goal direction does *not* reside within the field of view, the CAN is set to zero, effectively inducing a turn of the robot in the rotational phase until the goal direction is centered in the field of view.

3 Visual Collision Avoidance in Cluttered Environments

Parameter Optimisation. The implementation of the collision avoidance model in the dynamics simulation of HECTOR was tested in several cluttered environments. In a first step, the threshold n_0 and gain g of the weighting function W [see Eq. (6)] were optimised in an artificial environment. The environment consisted of a cubic box with a cylindrical object placed in the center (see Fig. 3B–D). Both, the box and the object were covered with a Perlin noise texture. The robot was placed at a starting position (S) in front of the object, facing a goal position (G) behind the object. The distance between starting position and goal was set to 10 m. For each of the possible parameter combinations of the gain $g = [1.0, 2.0, ..., 10.0]$ and threshold $n_0 = [0.0, 1.0, ..., 20.0]$ the trajectory length for reaching the goal (G) without colliding with the object was taken as a benchmark of the performance of the collision avoidance model (see Fig. 3A). A collision was assumed if the position of the camera crossed a radius of 1.5 m around the center of the object (*black dashed circle*, Fig. 3B–D) and the respective combination of parameters was discarded. For each combination of parameters 3 trials were performed.

If the threshold n_0 is set to *low* values, the computation of the heading direction γ [see Eq. (5)] mainly depends on the collision avoidance direction CAD_{fov}, whereas the goal direction α is only taken into account to a small extent. Hence, the robot will more likely avoid collisions than navigate to the goal (G). Further, a steeper slope of the sigmoid weighting function W, set by the gain g, leads to higher temporal fluctuation of the heading direction γ. As a consequence, when setting the threshold to $n_0 = 0.0$ and the gain to $g = 10.0$, the resulting trajectories were relatively long (Fig. 3A) and showed erratic movement patterns. However, all trajectories reached the goal position for the given parameter combination (Fig. 3B). Due to the robot following the collision avoidance direction CAD_{fov}, in several cases the goal direction did not reside within the field of view,

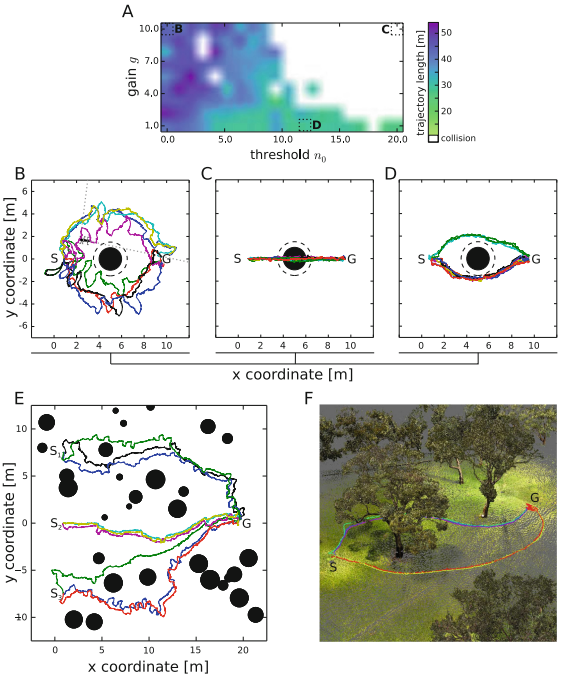

Fig. 3. (A) *Length of simulated trajectories* (color-coded) in a cubic box with a single object (see B–D) for different combinations of the weighting function parameters gain $g = [1.0, 2.0, ..., 10.0]$ and threshold $n_0 = [0.0, 1.0, ..., 20.0]$ [see Eq. (6)]. The size of the box was 14 m × 14 m × 10 m (length × width × height) and the radius of the object $r = 1$ m (height h =10 m). The walls of the box and the object were uniformly covered with a Perlin noise texture (scale = 0.05). When the trajectory crossed a circle of a radius of 1.5 m around the center of the object (dashed line in B–D) a collision was assumed (white areas). (B–D) *Simulated trajectories (n = 10) in a cubic box with a single object* (filled circle). Starting positions are given as S and goal positions as G. Weighting function parameters were set to (B) $g = 10.0$ and $n_0 = 0.0$, (C) $g = 10.0$ and $n_0 = 20.0$ and (D) $g = 1.0$ and $n_0 = 12.0$. The *grey dotted lines* in B indicate the main optical axis before and after recentering the goal direction in the visual field. (E) *Simulated trajectories in a cubic box with randomly placed objects* (filled circles) for different starting positions (S_1–S_3). The size of the box was 25 m × 25 m × 10 m (length × width × height). The radius of each object ($n = 30$; height: 10 m) was set randomly in a range from 0.25 m to 1.0 m. The walls of the box and the objects were uniformly covered with a Perlin noise texture (scale = 0.05). Weighting function parameters were set to $g = 1.0$ and $n_0 = 12.0$ (see A and D). For each starting position 3 trajectories are shown. It is notable, that the variability for trajectories with the same starting positions arises due to the initialization of the robot with differing body postures, effectively influencing the initial perceived image flow. (F) *Simulated trajectories* ($n = 5$) *in a reconstructed natural environment.* Weighting function parameters were set to $g = 1.0$ and $n_0 = 12.0$ (see A and D). The distance between starting position (S) and goal position (G) was 48.83 m.

resulting in a recentering of the goal vector along the main optical axis (as indicated by the *grey dashed lines* in Fig. 3B). This strategy led to reaching the goal position for all trajectories.

In contrast, when setting the threshold n_0 to *high* values, the computation of the heading vector γ mainly takes the goal direction α into account, whereas the influence of the collision avoidance direction (CAD_{fov}) is reduced. As a consequence, the robot will more likely follow the direction to the goal without avoiding obstacles. Therefore, when setting the threshold to $n_0 = 20.0$ and the gain to $g = 10.0$, the robot directly approached the goal position, consequently, colliding with the object (Fig. 3A and C).

Figure 3D shows the trajectories for a combination of the parameters gain and threshold which resulted in short trajectory lengths without collisions ($n_0 = 12.0$, $g = 1.0$). Here, the robot almost directly approached the goal, while effectively avoiding the object. This combination of the threshold and gain parameters was used in subsequent simulations in more complex cluttered environments.

Artificial Cluttered Environment. After optimisation of the threshold and gain parameters the performance of the collision avoidance model was tested in an Artificial cluttered environment (Fig. 3E) which was set up in a cubic box. Several cylindrical objects ($n = 30$) were placed at random positions in the x,y-plane. The radius of the objects was set individually to a random number of $r = [0.25, 1.0]$ m. Both, the box and the objects were covered with a texture generated from Perlin noise. The robot was placed at different starting positions (S_1–S_3), with the main optical axis oriented in parallel with the x-axis. The distance to the goal position was $d_{1,3} = 21.36$ m for the starting positions S_1 and S_3 and $d_2 = 20$ m for the starting position S_2. The parameters of the sigmoid weighting function W were set to $n_0 = 12.0$ and $g = 1.0$ according to Fig. 3A. For each starting position 3 trajectories were simulated. In all cases the robot successfully reached the goal position without collisions and without encountering local minima (see however [1] for a more detailed analysis).

Natural Cluttered Environment. We further tested the performance of the collision avoidance model in a reconstructed natural environment (Fig. 3F). The environment consisted of a dataset obtained from several laser scans [22]. The starting (S) and goal position (G) were set so that the robot had to avoid collisions with trees to reach the location of the goal. Also in the natural environment – which substantially differs in the textural pattern properties from the tested artificial environment – for all trials ($n = 5$) the combination of weighting function parameters $n_0 = 12.0$ and $g = 1.0$ resulted in trajectories successfully leading to the goal, without colliding with objects.

4 Conclusion

A prerequisite for autonomous mobile robots is to navigate their environments while actively avoiding collisions with obstacles. Whereas, to perform collision

avoidance, autonomous robots nowadays normally rely on active sensors (e.g. laser range finders [13]) or extensive computations (e.g. Lucas-Kanade optic flow computation [10]), insects are able to do so with minimal energetic and computational expenditure by relying mainly on visual information. We implemented a bio-inspired model of collision avoidance in simulations of the hexapod walking robot HECTOR solely based on the processing of optic flow by correlation-type elementary motion detectors (EMDs). EMDs have previously been accounted for playing a key role in the processing of visual motion information in insects [3]. As could be shown, although the responses of EMDs to visual motion are entangled with the textural properties of the environment [5], the relative nearness information obtained from optic flow estimation via EMDs is sufficient to direct HECTOR to a goal location in cluttered environments without colliding with obstacles. This holds true either for *artificially* generated environments as well as for a *reconstructed natural* environment, which substantially differ in their textural pattern properties. Moreover, by employing behavioural strategies such as (a) an *active-gaze strategy* and (b) *active head stabilisation* – both also found in insects – the influence of rotational optic flow components which potentially obfuscate the estimation of relative nearness information from optic flow is reduced. Hence, on the physical robot a prototype for mechanical gaze-stabilisation has been implemented and is currently compared to a software implementation.

The simulation results shown here will serve as a basis for the implementation of more complex bio-inspired models for visually-guided navigation in hardware which is currently under development. These models will comprise strategies for navigation and search behaviour based on the insect-inspired processing of optic flow.

Acknowledgments. This work has been supported by the DFG Center of Excellence Cognitive Interaction TEChnology (CITEC, EXC 277) within the EICCI-project. We thank Dr. Wolfgang Stürzl for kindly providing us with a dataset of a laser scanned outdoor environment.

References

1. Bertrand, O.J., Lindemann, J.P., Egelhaaf, M.: A bio-inspired collision avoidance model based on spatial information derived from motion detectors leads to common routes. PLoS Comput. Biol. **11**(11), e1004339 (2015)
2. Borst, A.: Modelling fly motion vision. In: Feng, J. (ed.) Computational Neuroscience: A Comprehensive Approach, pp. 397–429. Chapman and Hall/CTC, Boca Raton, London, New York (2004)
3. Borst, A.: Fly visual course control: behaviour, algorithms and circuits. Nat. Rev. Neurosci. **15**(9), 590–599 (2014)
4. Egelhaaf, M., Boeddeker, N., Kern, R., Kurtz, R., Lindemann, J.P.: Spatial vision in insects is facilitated by shaping the dynamics of visual input through behavioral action. Front. Neural Circuits **6**(108), 1–23 (2012)

5. Egelhaaf, M., Kern, R., Lindemann, J.P.: Motion as a source of environmental information: a fresh view on biological motion computation by insect brains. Front. Neural Circuits **8**(127), 1–15 (2014)
6. Goslin, M., Mine, M.R.: The panda3d graphics engine. Computer **37**(10), 112–114 (2004)
7. Koenderink, J.J.: Optic flow. Vis. Res. **26**(1), 161–179 (1986)
8. Kress, D., Egelhaaf, M.: Head and body stabilization in blowflies walking on differently structured substrates. J. Exp. Biol. **215**(9), 1523–1532 (2012)
9. Kress, D., Egelhaaf, M.: Impact of stride-coupled gaze shifts of walking blowflies on the neuronal representation of visual targets. Front. Behav. Neurosci. **8**(307), 1–13 (2014)
10. Lucas, B.D., Kanade, T., et al.: An iterative image registration technique with an application to stereo vision. In: IJCAI, vol. 81, pp. 674–679 (1981)
11. Matthews, R.W., Matthews, J.R.: Insect Behavior. Springer, Netherlands (2009)
12. Miyamoto, K.: Fish eye lens. JOSA **54**(8), 1060–1061 (1964)
13. Montano, L., Asensio, J.R.: Real-time robot navigation in unstructured environments using a 3d laser rangefinder. In: Proceedings of the 1997 IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS 1997, vol. 2, pp. 526–532. IEEE (1997)
14. Paskarbeit, J., Annunziata, S., Basa, D., Schneider, A.: A self-contained, elastic joint drive for robotics applications based on a sensorized elastomer coupling - design and identification. Sens. Actuators A Phys. **199**, 56–66 (2013)
15. Paskarbeit, J., Schilling, M., Schmitz, J., Schneider, A.: Obstacle crossing of a real, compliant robot based on local evasion movements and averaging of stance heights using singular value decomposition. In: 2015 IEEE International Conference on Robotics and Automation (ICRA), pp. 3140–3145. IEEE (2015)
16. Petrowitz, R., Dahmen, H., Egelhaaf, M., Krapp, H.G.: Arrangement of optical axes and spatial resolution in the compound eye of the female blowfly calliphora. J. Comp. Physiol. A **186**(7–8), 737–746 (2000)
17. Schilling, M., Hoinville, T., Schmitz, J., Cruse, H.: Walknet, a bio-inspired controller for hexapod walking. Biol. Cybern. **107**(4), 397–419 (2013)
18. Schwegmann, A., Lindemann, J.P., Egelhaaf, M.: Depth information in natural environments derived from optic flow by insect motion detection system: a model analysis. Front. Comput. Neurosci. **8**(83), 1–15 (2014)
19. Shoemaker, P.A., Ocarroll, D.C., Straw, A.D.: Velocity constancy and models for wide-field visual motion detection in insects. Biol. Cybern. **93**(4), 275–287 (2005)
20. Srinivasan, M., Guy, R.: Spectral properties of movement perception in the dronefly eristalis. J. Comp. Physiol. A **166**(3), 287–295 (1990)
21. Stürzl, W., Böddeker, N., Dittmar, L., Egelhaaf, M.: Mimicking honeybee eyes with a 280 field of view catadioptric imaging system. Bioinspir. Biomim. **5**(3), 036002 (2010)
22. Stürzl, W., Grixa, I., Mair, E., Narendra, A., Zeil, J.: Three-dimensional models of natural environments and the mapping of navigational information. J. Comp. Physiol. A **201**(6), 563–584 (2015)