

Bioinspired event-driven collision avoidance algorithm based on optic flow

Moritz B. Milde^{*§¶}, Olivier J.N. Bertrand^{*§}, Ryad Benosman[‡], Martin Egelhaaf^{*§} and Elisabetta Chicca^{*†}

^{*}Cognitive Interaction Technology - Center of Excellence, Bielefeld University

[§]Neurobiology, Bielefeld University

[‡]Institut de la Vision, University Pierre and Marie Curie, Paris

[†]Faculty of Technology, Bielefeld University

[§]Neurobiology, Bielefeld University

[¶]Mail to: moritz.milde@uni-bielefeld.de

Abstract—Any mobile agent, whether biological or robotic, needs to avoid collisions with obstacles. Insects, such as bees and flies, use optic flow to estimate the relative nearness to obstacles. Optic flow induced by ego-motion is composed of a translational and a rotational component. The segregation of both components is computationally and thus energetically expensive. Flies and bees actively separate the rotational and translational optic flow components via behaviour, i.e. by employing a saccadic strategy of flight and gaze control. Although robotic systems are able to mimic this gaze-strategy, the calculation of optic-flow fields from standard camera images remains time and energy consuming. To overcome this problem, we use a dynamic vision sensor (DVS), which provides event-based information about changes in contrast over time at each pixel location. To extract optic flow from this information, a plane-fitting algorithm estimating the relative velocity in a small spatio-temporal cuboid is used. The depth-structure is derived from the translational optic flow by using local properties of the retina. A collision avoidance direction is then computed from the event-based depth-structure of the environment. The system has successfully been tested on a robotic platform in open loop.

I. INTRODUCTION

The role of autonomous robots in our daily lives has increased through the years. The field of robotics faces several challenges for the robot to be more flexible, smaller, and more energy efficient. Animals often show a better performance than artificial agents in this regard. Therefore robotics may take inspiration from biology for achieving a similar performance. Avoiding collisions is one of the most basic needs of any moving robot. This task has been solved until now, with either expensive and relatively large devices, such as 3D range finders [1], or computationally and energetically challenging techniques, such as optic-flow computation from camera images [2], [3].

In a static environment, a stationary robot will not have to avoid collisions. Thus, stationary images are not relevant for a collision avoidance system, and their processing would be a waste of computational power and energy. Therefore the sensory system, i.e. the camera, should only process changes in the environment. To measure only the changes in the environment, an event-based visual sensor has been introduced in previous studies [4], [5]: the dynamic vision sensor (DVS). The DVS only transmits information when a change in contrast over time is observed at a specific location. The energy consumption is thus reduced compared to classical cameras [5]. Any motion of

an agent induces on the vision system apparent movement of the surroundings, i.e. optic flow. During translational motion of the agent, nearby objects induce large optic flow amplitudes. Optic flow amplitudes have already been used to design a collision avoidance algorithm [2], [6]–[8]. However collision avoidance has never been performed so far with an event-based system, although a time-to-contact algorithm [9] and an altitude, pitch and forward speed control algorithm [10] have been developed on this basis.

In the present paper, we propose a model of collision avoidance coding the nearness to objects, measured by the spatio-temporal variation of events emitted by the DVS. The collision avoidance algorithm can be subdivided into three processing steps: (1) Computation of optic flow from DVS events, (2) extraction of nearness from optic flow, and (3) determination of a collision avoidance direction from the nearness information, i.e. where to go. We validated the algorithm on a robotic platform (Pioneer 2-DX). The robot, equipped with a DVS and a standard webcam, moved in open field with eight objects. The data produced by the DVS constitutes the input for the algorithm which computes the collision avoidance direction. The movement of the robot was not yet controlled by the collision avoidance direction computed in this way. In the next step of the analysis the collision avoidance algorithm will be tested also under closed-loop conditions.

A. Optic flow

The fundamental properties of translational optic flow can easily be observed when looking out of a moving train: The nearby objects like trees along the track appear to move faster than a far away wind turbine. This phenomenon is called motion parallax and represents the image displacements on the eye during translational self-motion [11], [12]. The optic flow experienced during translational motion in a static environment depends on the agent's speed, its nearness to objects, and its motion direction. In contrast, the rotational optic flow depends only on the self-motion of the agent and, thus, is independent of the spatial layout of the environment.

Since the work of Gibson [11], several algorithms have been developed to estimate the optic flow from image motion. One of the first algorithms was proposed by Hassenstein and Reichardt at the end of the 1950's [13]. This relatively simple algorithm (correlation-type movement detector; EMD) is based on the spatio-temporal correlation between pixels. However it



Fig. 1. The Dynamic Vision Sensor (DVS). Left) The DVS mounted on the Pioneer 2-DX robot Right) Experimental setup of 8 objects in open field covered with black/white checkerboard pattern.

does not code the optic flow unambiguously, i.e. it depends, in addition to velocity, on the textural properties of the scenery [14]. To overcome this problem, several algorithms have been introduced [15]. However these methods are based on classical camera images, and are high-power and time consuming. Benosman et al. have introduced a method to compute optic flow from the DVS [16]. This camera combines low power consumption with a high sampling rate, two requirements for fast moving robots. Due to the response to local contrast changes the DVS has a high intra-scene dynamic range. Hence, the detection of contrast changes is independent of the overall brightness of the environment [17].

B. The flight characteristic of flying insects

Flying insects, such as bees and flies, perform complex flight manoeuvres at high speeds in a wide range of environments [18]–[22]. Collision avoidance decisions are produced by an insect’s brain with very limited neural resources [23], [24] and thus, with low energy consumption. Hence, flying insects can be assumed to have a fast and low power consumption collision avoidance system. Experiments with flies and bees have shown that they rely on optic flow to avoid collisions [18], [25], and to control their flight speed [18], [19], [21].

A moving agent normally perceives a superposition of rotational and translational optic flow, although only translational optic flow contains nearness information useful for collision avoidance. The translational optic flow has therefore to be extracted from the superimposed optic flow. This separation can either be done by computation (e.g. [26]) or via an active gaze strategy [27]. Flies and bees show a characteristic flight style, i.e. a sequence of straight segments (intersaccades) and sharp turns (saccades). Intersaccades are accompanied by slight sideways motion (drift) due to inertia. Saccades are characterised by high angular velocities of the head of up to 4000 deg/s performed mainly around the yaw axis of the animal, whereas intersaccades are periods where body and head orientation stays relatively constant [18], [22], [25], [27], [28]. This active gaze strategy segregates the optic flow into its rotational and translational components and, therefore, reduces the computational load to extract relevant collision avoidance information. This strategy may therefore be beneficial for small and fast moving robots. Our aim is to mimic this strategy with a control approach which relies on optic flow during translational motion only, ignoring input data generated during turning manoeuvres.

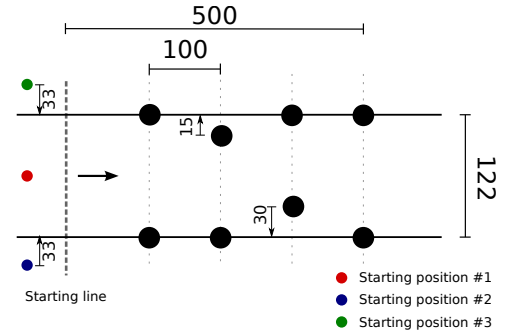


Fig. 2. Object constellation formed by 8 cylindrical objects (black dots) in open field. The robot started from three different starting positions (red: between the objects, blue: on right side of the objects, green: on the left side of the objects). The distance between each object along the trajectory is 100 cm . In total the robot travelled 500 cm . The distance of each object to the white lines (Fig. 1 right), varies between 0 and 30 cm .

II. MATERIAL & METHODS

A. Neuromorphic silicon retina

A Pioneer 2-DX robot was used to acquire data in open field with 8 cylindrical objects ($\Phi \times H = 9 \times 74$ cm), covered with a 9 mm black and white checkerboard pattern (Fig. 2). The robot moved along a straight line at 0.7 ms^{-1} . The DVS camera (Fig. 1) was mounted on top of the robot, and faced in the direction of motion. The data were recorded with three different starting positions (Fig. 2). The optic flow (subsection B), nearness, and collision avoidance direction (subsection C) were computed using Python 2.7. To interact with the camera system the jAER graphical user interface was used [29], which forwarded the raw data via user datagram protocol (UDP) to Python. Although biological systems seem to use asynchronous processing, whereas a classical computer relies on a clock to process information, a classical computer has been used to analyse the data, to show the principle of the algorithm. The DVSs chip has 128×128 pixel resolution, and its power consumption is 23 mW . One pixel of the DVS produces a positive spike when a contrast increment occurs (ON-Event) and a negative spike for contrast decrements (OFF-Event) (for more details see [5]). If a pixel of the DVS generates an event (or spike), the pixel’s location on the sensor (i.e. the x and y coordinates) and its polarity (pol), i.e. “on” or “off”, are transmitted. Time represents itself and, thus after generation of a spike a time stamp is attached to the event (Eq. 1). An event is, therefore, defined by a 4D vector:

$$e = (x, y, t, pol) \quad (1)$$

B. From events to optic flow

Once several events are received by the computer, the optic flow is computed based on the method proposed by Benosman et al. [16]. This method can be best understood by considering an artificial square (5×5 px) moving in the x and y direction at a speed of ($1px\mu s^{-1}$) (Fig. 3a). If an event e is detected a spatio-temporal cuboid Γ_e (5 $px \times 5$ $px \times 70$ $msec$) is generated around the detected event (Fig. 3b). A principal component analysis (PCA) was used to estimate the direction and the strength of variation of the events belonging to Γ_e (Fig. 3c). The spatio-temporal cuboid Γ_e has three dimensions. Therefore, the PCA operates on three variables (x, y, t), and

three eigenvectors are calculated. The first component, which is the first eigenvector times the corresponding eigenvalue, accounts for the direction and strength along x , y and the time of the greatest variation in Γ_e . The first two principal components span a plane Π (Eq. 2), accounting for the strongest variations in the spatio-temporal cuboid.

$$\begin{aligned} \Pi &= ax + by + ct + d \\ \boldsymbol{\theta} &= \begin{pmatrix} a \\ b \\ c \end{pmatrix}; \boldsymbol{\theta} \times \Pi = 0 \end{aligned} \quad (2)$$

$\boldsymbol{\theta}$ is the normal to the plane Π , i.e. the third eigenvector. The velocity along x (v_x) and along y (v_y) is defined by the change in x and y , respectively, divided by the change in t . The change in x , y and t is defined by the component of $\boldsymbol{\theta}$ along x , y and t , respectively. Therefore the velocity $\boldsymbol{v} = (v_x, v_y)$ is $\boldsymbol{v} = -\frac{1}{c}(a, b)$. The PCA has the advantage that the third principal component describes entirely the plane spanned by the first two principal components (Eq. 2). A similar approach was done in [30], and presents advantages compared to the inverse function theorem as suggested in [16].

To make the algorithm robust against noise, outliers need to be rejected. The PCA can be calculated for two different subsets of events in the spatio-temporal cuboid. The PCA quality is defined by the difference between the third eigenvectors computed from the two subsets. Unless the PCA quality is below 1×10^{-5} arbitrary unit (a.u.), the distance d from each event to Π is calculated (Eq. 3). Events with a distance larger than 0.7 a.u. are rejected. The PCA is then recalculated on the remaining events, and the PCA quality is calculated with the current and previous eigenvectors.

$$d = \frac{\boldsymbol{\theta} \times \Gamma_{e(i)}}{\|\boldsymbol{\theta}\|} \quad (3)$$

C. From optic flow to collision avoidance

The optic flow experienced during translation is linked to the nearness of the agent to objects in the environment and to its self-motion. It has been shown that the relative nearness can be locally computed with a pseudo norm of the optic flow, independently of the motion direction [8]. Therefore, an optic flow vector measured at a given viewing direction contains enough information to compute the relative nearness in that direction. By using the norm of the optic flow computed from the events emitted by the DVS, the relative nearness is estimated. Each event generated by the DVS may generate an optic flow vector and, therefore, a relative nearness measurement. The relative nearness measurements are thus event-like. Estimating a collision avoidance direction from relative nearness information is a common problem [1], [31]–[33]. Using a vector sum of the relative nearness measurements, i.e. the center of mass of the nearness vectors, a collision avoidance direction can be computed [8]. The relative nearness measurements are associated with given pixels (x, y) on the chip at a given time t . Each pixel on the chip is associated with an “virtual angle” ϕ , here $\phi = 2\pi \times x/128 - \pi$, e.g. the pixel location on the DVS is mapped to an angle between $-\pi$ and π . The relative nearness vectors calculated at $t_i = t_j \pm 35ms$, with t_j being the time stamp of one relative nearness vector, are then selected to compute the collision avoidance direction. The

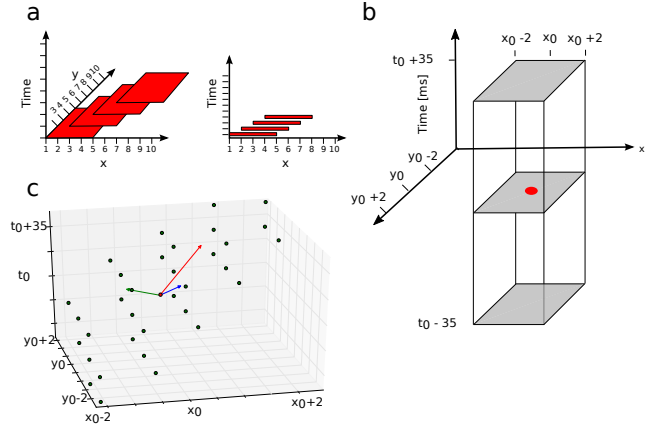


Fig. 3. Principle of optic flow computation from events. The motion on a square, in x and y at a constant velocity of $1 \text{ px}\mu\text{s}^{-1}$, was simulated. a) Square position in the spatio-temporal domain, i.e. along x , y , and time. A projection of the square position in the plane (x -time) is shown on the right. The plane is positive x and y , because x and y increase with the time. b) Every time the DVS detects an event (red dot, $e_0(x_0, y_0, t_0)$), a spatio-temporal cuboid around e_0 is created. All events belonging to the resulting spatio-temporal cuboid are used to estimate the velocity of the event e_0 . c) The event positions in the spatio-temporal domain, within the spatio-temporal cuboid centred at $e_0(x_0, y_0, t_0)$, are represented by green dots. The coloured arrows represent the eigenvectors of the PCA. The first (red arrow) and second (green arrow) principal component, forming a plane Π , explain the strongest variation within the spatio-temporal cuboid. The slope in the x -direction of Π is directly related to the velocity v_x and v_y in the y -direction. The slope of Π is described by the third principal component (blue arrow).

vector sum of these vectors, with arguments ϕ_i , points toward close objects. Therefore the argument of the opposite vector determines a suitable collision avoidance direction. During the collision avoidance, e.g. turning manoeuvres, optic flow contains no spatial information about the environment, due to rotational motion (see Section I-A & B). Hence, no collision avoidance direction will be computed in this short period of time.

III. RESULTS

The collision avoidance algorithm introduced by Bertrand et al. [8], based on optic flow measured by correlation-type motion detectors (EMD), has been shown to be successful under different environmental conditions. However, for EMD-based optic flow estimates each frame of the camera has to be processed. The EMD signal is a continuous function of time, i.e. information is always transmitted from the detector, even when the agent is stationary and relative nearness measurements are available even if they are not needed for collision avoidance. The relative nearness computed from the DVS output is only available if the contrast changes, e.g. when the agent is moving. Therefore, it’s a challenge of the collision avoidance algorithm to cope also with sparse relative nearness information.

The event-based data from the DVS is encoded asynchronously. However the computation of optic flow is done on a computer, i.e. synchronously. The asynchronous computation of optic flow is beyond the scope of the present paper. The real time communication between synchronous and asynchronous devices is a challenging task due to buffer problems. The collision avoidance algorithm has thus been tested in an open loop experiment, to highlight the principle of the algorithm

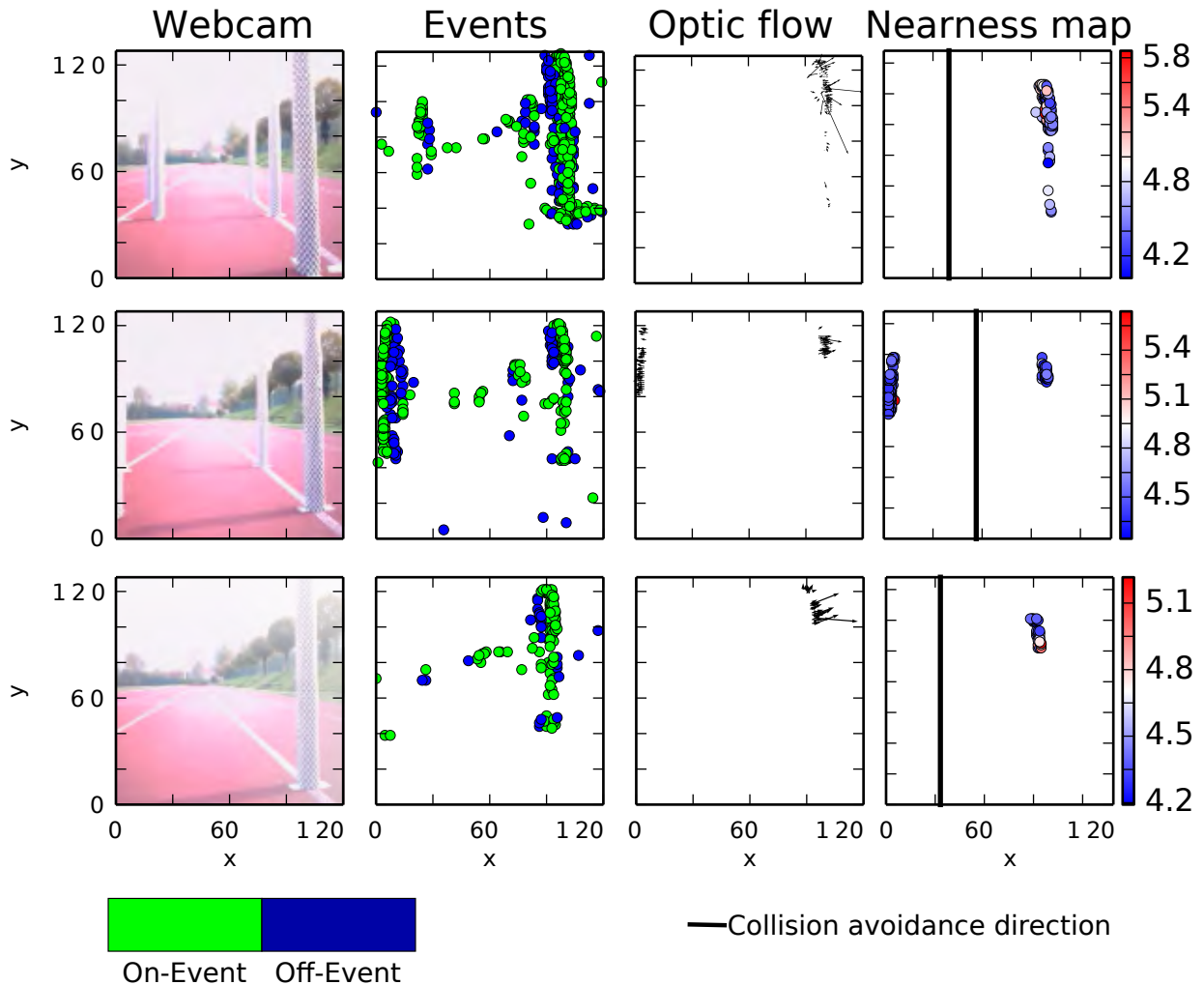


Fig. 4. The output of every step of the collision avoidance in an open loop experiment for three different time points along the robot’s trajectory. The robot moves in open field; here eight objects were present (starting position # 1). The time increases from top to bottom row. First column: RGB-Images acquired by a standard webcam. Second column: The projection of the events within a time $t_i \pm 35ms$ on the $(x - y)$ plane, i.e. the camera coordinates. On and off events are represented by green and blue dots, respectively. Object boundaries can be seen by neighbouring on and off edges. Third column: Computed optic flow within the spatio-temporal cuboid, for each event shown in the second column. Note that certain events do not lead to an optic flow vector, due to the lack of events within the spatio-temporal cuboid. Fourth column: Relative nearness derived from optic flow shown by colour-coded dots. The 10-based-log of the norm is plotted for each optic flow vector. The norm is used to compute the collision avoidance direction, which is indicated by the black bar. Note that the collision avoidance direction points away from nearby objects for the different time points along the trajectory. The unit of the colourbar are arbitrary.

with event-based optic flow.

A robot equipped with the DVS and a standard webcam (Fig. 4, first column) moved in open field with a given constellation of objects to gather events emitted from the DVS. The acquired data during the motion of the robot was used offline to compute the collision avoidance direction. As the robot moves through the environment, the projections of the objects’ positions move across the sensor. On and off events are emitted at the objects’ boundaries (Fig. 4, second column). The optic flow computed from these events represents mainly the displacements of object boundaries (Fig. 4, third column). The relative nearness is calculated for each optic flow vector (Fig. 4, fourth column). Far objects move at lower velocities than near ones and, thus may produce smaller changes in contrast. Thus, they are unlikely to contribute much to the collision avoidance direction.

The collision avoidance direction, in general, points away from nearby objects along the trajectory of the robot. The collision

avoidance direction points along the trajectory always in front of the robot (Fig. 6). Similar results were obtained with the two other starting positions, e.g. object constellations (Fig. 5). These results are similar to the simulation results presented in [8] although the optic flow was computed from the sparse event-driven information provided by the DVS instead of a continuous correlation-type EMD model.

IV. CONCLUSION

We developed an event-based collision avoidance algorithm. The collision avoidance direction, computed from this algorithm, points away from nearby objects. Although the algorithm has been tested, so far, only under open loop conditions in a corridor (data not shown) and open field with a small number of objects, our results show that this algorithm could be used to control a robotic platform.

One of the main advantages of using an event-driven vision

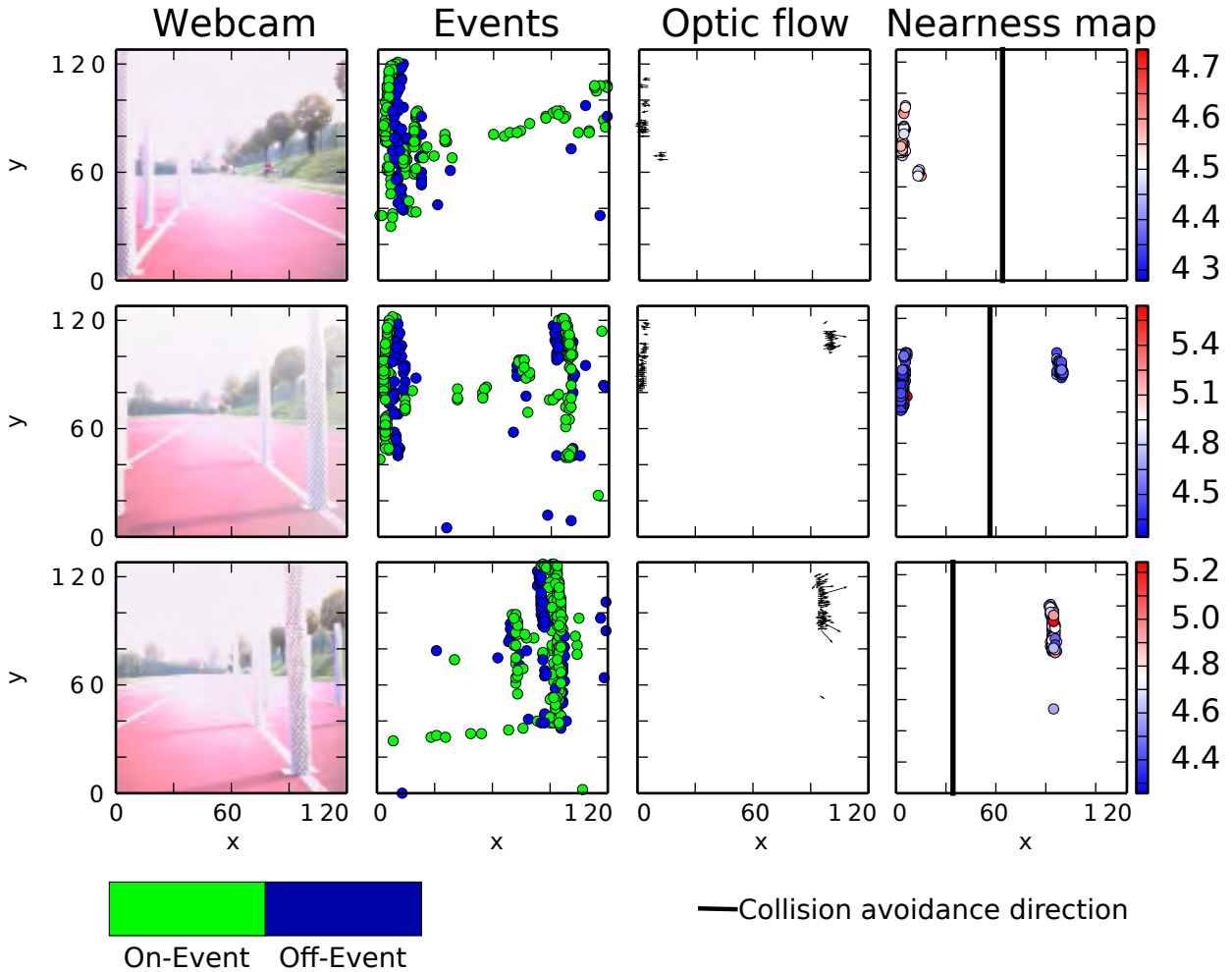


Fig. 5. The output of every step of the collision avoidance in an open loop experiment for three different starting positions at a given time point along the robot trajectories. The robot moves in open field, here eight objects were present. First column: RGB-Images acquired by a standard webcam. Second column: The projection of the events within a time $t_i \pm 35ms$ on the $(x - y)$ plane, i.e. the camera coordinates. On and off events are represented by green and blue dots, respectively. Object boundaries can be seen by neighbouring on and off edges. Third column: Computed optic flow within the spatio-temporal cuboid, for each event shown in the second column. Note that certain events do not lead to an optic flow vector, due to the lack of events within the spatio-temporal cuboid. Fourth column: Relative nearness derived from optic flow shown by colour-coded dots. The 10-based-log of the norm is plotted for each optic flow vector. The norm is used to compute the collision avoidance direction, which is indicated by the black bar. Note that the collision avoidance direction points away from nearby objects for the different starting positions, e.g object constellations. First to bottom row: Starting position # 2, # 1, # 3. The unit of the colourbar are arbitrary.

sensor is the suppression of redundancy already at the sensory level. The environment is thus sparsely represented, i.e. represented by events only if the sensory image changes. Despite this sparseness an appropriate collision avoidance direction is computed.

Our method remains to be validated by controlling a robotic platform under closed-loop conditions and real-time data processing. This preliminary study showed that the collision avoidance algorithm proposed by [8] works in principal with event-driven optic flow and will be the starting point of an asynchronous real-time implementation. One beneficial step to implement real-time processing is to keep the overall system asynchronous by using a neural network implemented on a neuromorphic chip of the kind described in [34]. This neural network would carry out the computation of optic flow, the relative nearness, and the collision avoidance direction. The above mentioned computational steps have then to be translated into asynchronous hardware. The main advantage

would be the decreased time between information acquisition and extracting the collision avoidance direction. This means that as soon as a contrast change is encoded by a voltage change, the voltage is processed by the neural network, i.e. neuromorphic circuit. The extraction of optic flow based on PCA and neural networks requires learning on a large amount of data. Thus we might have to test alternative approaches to estimate optic flow with a neuromorphic circuit [35], [36]. The collision avoidance direction could then be extracted from optic flow by a winner-takes-all network based on the norm of optic flow. A winner-takes-all network was already used to extract a collision avoidance direction from sonar-based input [33]. The low power consumption as well as the small and compact design of such a circuit would open the potential of implementing it on light weight agents. More generally, the application of the present collision avoidance algorithm to an analogue neural network would offer the opportunity to investigate and understand principles of information processing

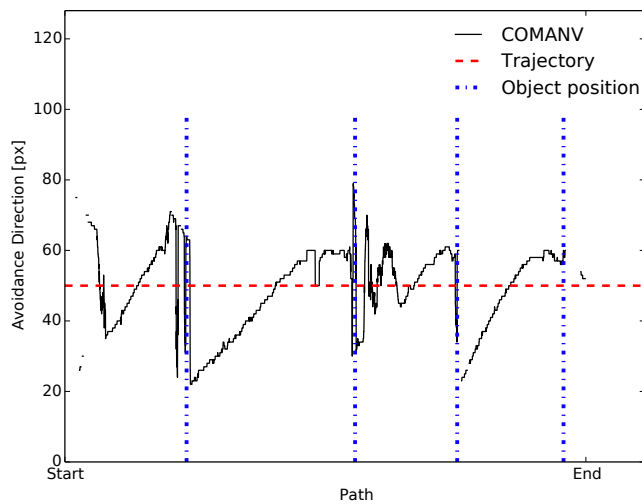


Fig. 6. Center of mass collision avoidance direction (COMANV) over the robots trajectory (see specific time points Fig. 4). Black line: COMANV, red dashed line: robots trajectory, blue dot dashed line: object position along the trajectory (see Fig. 2 for detailed object position). The collision avoidance direction points for every time point in relatively small area in front of the robot.

with spiking neural networks. Furthermore, we might obtain a better understanding of the working operation of biological neural networks.

ACKNOWLEDGEMENT

This work was supported by the Deutsche Forschungsgemeinschaft (DFG). The authors would like to thank Stephen Nease for helpful language improvements.

REFERENCES

- [1] J. Minguez and L. Montano, "Nearness diagram navigation (nd): A new real time collision avoidance approach," in *Intelligent Robots and Systems (IROS 2000). Proceedings. IEEE/RSJ International Conference on*, vol. 3. IEEE, 2000, pp. 2094–2100.
- [2] A. M. Hyslop and J. S. Humbert, "Autonomous navigation in three-dimensional urban environments using wide-field integration of optic flow," *Journal of guidance, control, and dynamics*, vol. 33, no. 1, pp. 147–159, 2010.
- [3] B. D. Lucas and T. Kanade, "An Iterative Image Registration Technique with an Application to Stereo Vision," in *Proceedings of Imaging Understanding Workshop*, pp. 121–130, 1981.
- [4] P. Lichtsteiner and T. Delbruck, "A 64x64 aer logarithmic temporal derivative silicon retina," *Research in Microelectronics and Electronics*, vol. 2, pp. 202–205, 2005.
- [5] P. Lichtsteiner, C. Posch, and T. Delbruck, "A 128x128 120 db 15 μ s latency asynchronous temporal contrast vision sensor," *Solid-State Circuits, IEEE Journal of*, vol. 43, no. 2, pp. 566–576, 2008.
- [6] J.-M. Pichon, C. Blanes, and N. Franceschini, "Visual guidance of a mobile robot equipped with a network of self-motion sensors," in *Advances in Intelligent Robotics Systems Conference*. International Society for Optics and Photonics, 1990, pp. 44–55.
- [7] J. R. Serres and F. Ruffier, "Biomimetic autopilot based on minimalistic motion vision for navigating along corridors comprising u-shaped and s-shaped turns," *Journal of Bionic Engineering*, vol. 12, no. 1, pp. 47–60, 2015.
- [8] O. J. N. Bertrand, J. P. Lindemann, and M. Egelhaaf, "A bio-inspired collision avoidance model based on spatial information derived from motion detectors leads to common routes," *Plos Computational Biology*, "in revision".

- [9] X. Clady, C. Clercq, S.-H. Ieng, F. Houseini, M. Randazzo, L. Natale, C. Bartolozzi, and R. Benosman, "Asynchronous visual event-based time-to-contact," *Frontiers in neuroscience*, vol. 8, 2014.
- [10] F. Expert and F. Ruffier, "Flying over uneven moving terrain based on optic-flow cues without any need for reference frames or accelerometers," *Bioinspiration & biomimetics*, vol. 10, no. 2, 2015.
- [11] J. Gibson, *The Perception of the visual world*. Houghton Mifflin, 1950.
- [12] J. J. Koenderink and A. J. van Doorn, "Optic flow," *Vision Research*, vol. 26, no. 1, pp. 161–180, 1986.
- [13] W. Reichardt, "Autocorrelation, a principle for evaluation of sensory information by the central nervous system," *Sensory Communication*, pp. 303–317, 1961.
- [14] A. Borst and M. Egelhaaf, "Principles of visual motion detection," *Trends in neurosciences*, vol. 12, no. 8, pp. 297–306, 1989.
- [15] S. S. Beauchemin and J. L. Barron, "The computation of optical flow," *ACM Computing Surveys*, vol. 27, no. 3, pp. 433–466, 1995.
- [16] R. Benosman, C. Clercq, X. Lagorce, S.-H. Ieng, and C. Bartolozzi, "Event-based visual flow," *Neural Networks and Learning Systems, IEEE Transactions on*, vol. 25, no. 2, pp. 407–417, 2014.
- [17] "Products of inilabs: Dynamic vision sensor," iniLabs Ltd web-site, 2009. [Online]. Available: <http://www.inilabs.com/products/dynamic-vision-sensors>
- [18] R. Kern, N. Boeddeker, L. Dittmar, and M. Egelhaaf, "Blowfly flight characteristics are shaped by environmental features and controlled by optic flow information," *The Journal of experimental biology*, vol. 215, no. P14, pp. 2501–14, 2012.
- [19] K. Weber, S. Venkatesh, and M. Srinivasan, "Insect inspired behaviours for the autonomous control of mobile robots," in *Pattern Recognition, Proceedings of the 13th International Conference on*. IEEE, 1996, pp. 156–160.
- [20] J. Zeil, "Orientation flights of solitary wasps (cerceris; sphecidae; hymenoptera)," *Journal of Comparative Physiology A*, vol. 172, no. 2, pp. 207–222, 1993.
- [21] E. Baird, M. V. Srinivasan, S. Zhang, and A. Cowling, "Visual control of flight speed in honeybees," *Journal of Experimental Biology*, vol. 208, no. 20, pp. 3895–3905, 2005.
- [22] C. Schilstra and J. H. van Hateren, "Blowfly flight and optic flow I. Thorax kinematics and flight dynamics," *The Journal of experimental biology*, vol. 202, pp. 1481–1490, 1999.
- [23] A.-S. Chiang, C.-Y. Lin, C.-C. Chuang, H.-M. Chang, C.-H. Hsieh, C.-W. Yeh, C.-T. Shih, J.-J. Wu, G.-T. Wang, and Y.-C. Chen, "Three-dimensional reconstruction of brain-wide wiring networks in *Drosophila* at single-cell resolution," *Current Biology*, vol. 21, no. 1, pp. 1–11, 2011.
- [24] W. Witthöft, "Absolute Anzahl und Verteilung der Zellen im Hirn der Honigbiene," *Zeitschrift für Morphologie der Tiere*, vol. 61, no. 1, pp. 160–184, 1967.
- [25] L. F. Tammero and M. H. Dickinson, "The influence of visual landscape on the free flight behavior of the fruit fly *Drosophila melanogaster*," *The Journal of experimental biology*, vol. 205, no. 3, pp. 327–43, 2002.
- [26] J. Lim, N. Barnes, and H. Li, "Estimating relative camera motion from the antipodal-epipolar constraint," *Pattern analysis and Machine Intelligence, IEEE Transactions on*, vol. 32, no. 10, pp. 1907–1914, 2010.
- [27] M. Egelhaaf, N. Boeddeker, R. Kern, R. Kurtz, and J. P. Lindemann, "Spatial vision in insects is facilitated by shaping the dynamics of visual input through behavioral action," *Frontiers in neural circuits*, 2012.
- [28] N. Boeddeker, L. Dittmar, W. Stürzl, and M. Egelhaaf, "The fine structure of honeybee head and body yaw movements in a homing task," *Proceedings. Biological sciences / The Royal Society*, vol. 277, no. 1689, pp. 1899–1906, 2010.
- [29] "The jAER open source project," SourceForge web-site, November 2006. [Online]. Available: <http://sourceforge.net/projects/jaer/>
- [30] T. Brosch, S. Tschechne, and H. Neumann, "On event-based optical flow detection," *Frontiers in Neuroscience*, vol. 9, 2015.
- [31] J. Borenstein and Y. Koren, "The vector field histogram-fast obstacle avoidance for mobile robots," *Robotics and Automation, IEEE Transactions on*, vol. 7, no. 3, pp. 278–288, 1991.

- [32] W. A. Lewinger, M. S. Watson, and R. D. Quinn, "Obstacle avoidance behavior for a biologically-inspired mobile robot using binaural ultrasonic sensors," in *Intelligent Robots and Systems, IEEE/RSJ International Conference on*. IEEE, 2006, pp. 5769–5774.
- [33] T. K. Horiuchi, "A spike-latency model for sonar-based navigation in obstacle fields," *Circuits and Systems I: Regular Papers, IEEE Transactions on*, vol. 56, no. 11, pp. 2393–2401, 2009.
- [34] E. Chicca, F. Stefanini, C. Bartolozzi, and G. Indiveri, "Neuromorphic electronic circuits for building autonomous cognitive systems," *Proceedings of the IEEE*, pp. 1 – 22, 2014.
- [35] S.-C. Liu, "A neuromorphic avlsi model of global motion processing in the fly," *Circuits and Systems II: Analog and Digital Signal Processing, IEEE Transactions on*, vol. 47, no. 12, pp. 1458–1467, 2000.
- [36] J. Serres, F. Ruffier, S. Viollet, and N. Franceschini, "Toward optic flow regulation for wall-following and centring behaviours," *International Journal of Advanced Robotic Systems*, vol. 3, no. 2, pp. 147–154, 2006.