Drosophila-inspired visual orientation model on the Eye-Ris platform: experiments on a roving robot

P. Arena[a], S. De Fiore[a], L. Patané[a], L. Alba[b], and R. Strauss[c].

[a] Dipartimento di Ingegneria Elettrica, Elettronica e Informatica, Universitá degli Studi di Catania, Viale A. Doria 6, 95125 Catania, Italy.

[b] AnaFocus - Innovaciones Microelectrnicas S.L. Av. Isaac Newton s/n. Pabelln de Italia 7th Floor. PT Isla de la Cartuja E-41092 Sevilla, Spain

[c] University of Mainz, Inst. f. Zool. III - Neurobiology D-55099 Mainz, Germany

ABSTRACT

Behavioral experiments on fruit flies had shown that they are attracted by near objects and they prefer front-to-back motion. In this paper a visual orientation model is implemented on the Eye-Ris vision system and tested using a roving platform. Robotic experiments are used to collect statistical data regarding the system behaviour: followed trajectories, dwelling time, distribution of gaze direction and others strictly resembling the biological experimental setup on the flies. The statistical analysis has been performed in different scenarios where the robot faces with different object distribution in the arena. The acquired data has been used to validate the proposed model making a comparison with the fruit fly experiments.

Keywords: hybrid robot, parallax motion-based navigation, Eye-Ris, Drosophila melanogaster.

1. INTRODUCTION

Walking fruit flies are attracted by nearby objects. Behavioral experiments[1] on fruit flies had shown that Drosophila prefers front-to-back motion over back-to-front. They estimate the distance to these objects by the parallax motion of their images on the retina. The process in flies is not only selective to image motion created by the self-motion of the fly but also sensitive to object motion. Added visual motion (e.g. oscillations) makes objects more attractive than their stationary counterparts. Front-to-back motion, the natural parallax motion on the eyes of a forward-translating fly, is preferred. A group of several more distant objects can be more attractive than one close object. Parallax motion of the images of stationary objects is produced only during locomotion of the fly; close objects move with higher speed and larger amplitude over the retina than objects that are further away. Moreover, objects that are most attractive in the fronto-lateral eye-field, act as repulsive in the rear visual field.

In a first step of processing, visual motion is extracted by means of elementary motion detectors (EMDs) of the correlation type.[2] Visual motion for distance estimation is most likely perceived by the well-studied array of correlation-type elementary movement detectors (EMDs) of the fly visual system.[2–4] Within known limits and characteristics these detectors increase their output with increasing speed of visual motion. If object motion interferes with distance estimation, then flies should prefer an oscillating object over a stationary counterpart at the same distance and a fast oscillating object over a slowly oscillating object. In flies, the correlation-type motion detector connects exclusively immediately neighboring photoreceptors. A schematic representation of a minimal model describing orientation behavior in fruit-flies was recently introduced by some of the authors and is reported in Fig.1.[5]

Visual motion information is extracted from a horizontal ring of ommatidia. Each Drosophila ommatidium has an acceptance angle of 4.6°. Correlation-type visual motion detectors connect exclusively immediately adjacent ommatidia. Each elementary motion detector is composed of two mirror-symmetrically oriented motion detectors. The visual input of the first detector is filtered by a high-pass filter and multiplied with the high-pass filtered signal of the second photoreceptor, which will be delayed by means of a low pass filter (EMD in Fig. 1). Motion output is integrated space-wise and time-wise in four compartments (azimuth angles, both sides frontal 0° to 100° sideways and 100° sideways to 170° in the rear).

Further author information:
Send correspondence to Luca Patané, E-mail: lpatane@diees.unict.it

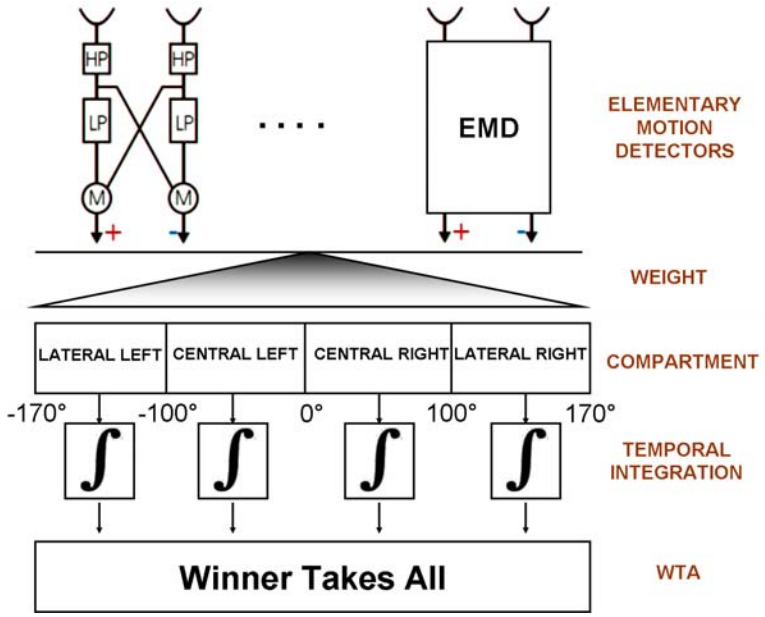


Figure 1. Minimal model of the orientation behavior towards visual objects (HP: high pass filter; LP: low pass filter; M: multiplication stage).

Visual orientation in flies is completely controlled by the Central Complex, an important neural assembly specifically devoted to visual tasks, and in particular the basic behaviours, orienting towards nearest blobs, are coordinated by the Protocerebral Bridge (PB) and the Fan-shaped Body (FB) through descending pathways that reach the ventral lobes for motor control.

In this paper a CNN-based version of Drosophila control orientation model is implemented on a roving robot using the Eye-Ris CNN-based vision system.

A CNN-based algorithm for fly-inspired orientation had been formerly implemented on a roving platforms,[6] and experimental results were already obtained. The CNN architecture reported in this manuscript is optimised, as regards the parameters involved, and implemented on a different robotic platform, used to reproduce the behavioral experiments on flies, as these are performed in a Neurobiological laboratory. The Neurobiological experimental setup was taken into primary account to perform statistical comparison between the roving robot emerging behavior and the experimental results on the fly. These comparison allows to appreciate the efficiency of the proposed CNN algorithm and opens the way to a unifying methodology, based on spatial-temporal dynamics, for the implementation of a complete insect brain model.

The roving platform used for the experiments and named in the following the ”Rover”, is a simple differential drive robot equipped with the Eye-Ris 1.3 version and with a series of sonar sensors for collision detection. The panoramic view, as acquired by the Eye-Ris focal plane processor, is divided into 78 azimuth areas (in analogy with the fly visual field) and collected in four macro sectors. A series of elementary motion detector (EMD) has been simulated to correlate the information coming from two adjacent visual areas. The weighted EMD outputs are then integrated in space and time for each compartment until a certain threshold is reached and an action is performed. The action, depending on the winning sector, could be attractive or repulsive accordingly to the fruit fly behaviour. The implementation of the visual orientation algorithm on the Eye-Ris guarantees high performances in terms of processing time due to the parallel processing capability of the smart visual processor.

An experimental campaign was performed on the Rover placed in a squared arena where objects of interest have been placed on the walls in different configurations. The experimental result has been chosen to perform a direct comparison with fly experiments.

2. THE CNN IMPLEMENTATION

The algorithm, introduced in the previous section, has been implemented on a Cellular Nonlinear Network structure. In order to emulate the acceptance angle of 4.6° of the drosophila ommatidium, a circle of 78 points equally spaced is extracted from the input pictures and translated in a row of a 78x78 image. In this way it is possible, exploiting the CNN parallel computation capabilities, to process in a full image all the information coming from the ommatidia. All the operations described in Fig. 1 have been implemented as CNN templates. However, some modifications to the algorithm were required to allow a full CNN implementation, obtaining an equivalent result in terms of the robot performance. The first step is the high pass filter calculation:

$$HP(t) = I(t) - I(t-1) \tag{1}$$

that is the subtraction between the image acquired during the actual step $I(t)$ and the image acquired the previous step $I(t-1)$. This operation was performed applying the template in (2), using $I(t)$ as initial state and $I(t-1)$ as input image and performing only one iteration.

$$A = \begin{bmatrix} 0 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \quad B = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \quad I = 0 \tag{2}$$

In Fig. 1, LP stands for a temporal low-pass filter with a time constant $\tau = 1.25$. The low-pass filtering leads to a temporal delay of the receptor signals and it is defined according to:

$$LP(t) = (1 - 1/\tau)LP(t-1) + (1/\tau)HP(t) \tag{3}$$

This operation was performed using the template in (4). In this case the initial state is $HP(t)$ while the input is $LP(t-1)$. The output of the operation is $LP(t)$.

$$A = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1/\tau & 0 \\ 0 & 0 & 0 \end{bmatrix}, \quad B = \begin{bmatrix} 0 & 0 & 0 \\ 0 & (1-1/\tau) & 0 \\ 0 & 0 & 0 \end{bmatrix}, \quad I = 0 \tag{4}$$

The subsequent step of the EMD implementation, i.e. the calculation of the cross product ϕ between HP and LP (Eq. 5), cannot be implemented on CNN since it requires a convolution between images that cannot be implemented using a spatio-invariant template:

$$\phi(t) = HP_i * LP_{i+1} - HP_{i+1} * LP_i \tag{5}$$

for i from 1 to 78.

To overcome the infeasibility of a convolution between images on CNN, it was chosen to perform an equivalent calculation of the Φ converting the output of the previous steps in binary images. In this way it is still possible to perform parallel processing on all the ommatidia, at the expenses of the introduction of other filtering parameters (i.e. the threshold values), to be chosen suitably. To calculate Φ, the multiplication has been substituted by a *AND* operation while the subtraction and absolute value with a *XOR*. Since the two multiplications should be performed between *HP* and *LP* of contiguous ommatidia, the *LP* (*HP*) was shifted to the right before performing the first (second) multiplication.

The next steps are the spatial integration in the various compartments and the temporal integration among the various iterations of the algorithm. To perform the spatial integration, the output of the previous step (Φ) is *AND*ed with four different masks which identify the different compartments. The black pixels output of each *AND* operation represent the points where a transition (i.e. a moving object) was identified inside the compartment. The total number of black pixels in each compartment represents, then, the spatial integration inside the compartment. To emphasize the output of the frontal part of the retina, the outputs of the central compartments are multiplied by two. The outputs of the weighted spatial integration are converted into four grayscale images which are then used among the various iterations of the algorithm to

perform the temporal integration. At the end of each iteration, a threshold operation is performed on the outputs of the temporal integration. According to the sector where the fixed threshold is overcome an action is performed. In particular if the two outermost compartments exceed the threshold first, the corresponding action is to turn away from the winning compartment, while if the innermost compartments win, the corresponding action is to turn towards the local motion.

Figure 2 depicts a flow diagram of the whole algorithm as it was implemented on CNN.

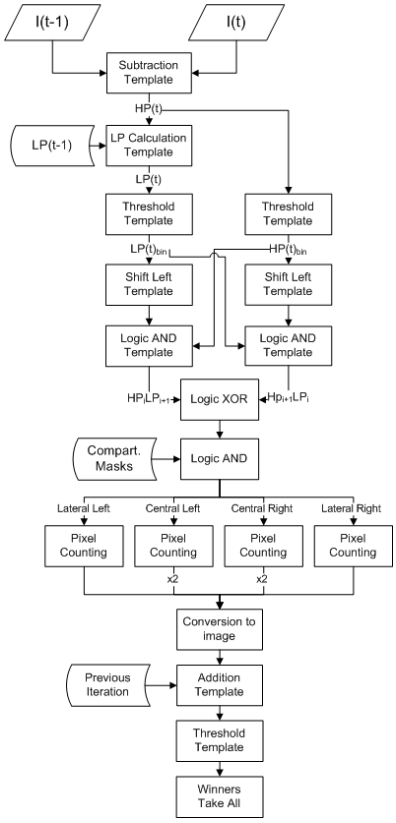


Figure 2. Flow diagram of the whole algorithm as implemented on CNN

3. THE EYE-RIS 1.3 PLATFORM

An hardware implementation of CNN is represented by the Eye-Ris platform,[7] a visual system, made by Anafocus,[8] that uses a fully-parallel mixed-signal array sensor-processor chip. The Eye-RIS system implements, indeed, a bio-inspired architecture represented by the retina-like front-end which combines signal acquisition and embedded processing on the same physical structure. This is the Q-Eye chip, an evolution of the previously adopted Analogic Cellular Engines(ACE),[9] the family of stand-alone chips developed in the last decade and capable of performing analogue and logic operations on the same architecture. The Q-Eye was devised to overcome the main drawbacks of ACE chips, such as lack of robustness and large power consumption.

Eye-RIS is a multiprocessor system since it employs two different processors: the Anafocus' Q-Eye Focal Plane Processor and the Altera's Nios II Digital Soft Core processors. The AnaFocus Q-Eye Focal Plane Processor (FPP) acts as an Image Coprocessor: it acquires and processes images, extracting the relevant information from the scene being analyzed, usually with no intervention of the Nios II processor. Its basic analog processing operations among pixels are linear convolutions with programmable masks. Size of the acquired and processed image is the Q-CIF (Quarter Common Intermediate Format) standard 176×144.

Altera NIOS II digital processor is a FPGA-synthesizable digital microprocessor (32-bit RISC μP at 70 MHz- realized on a FPGA). It controls the execution flow and processes the information provided by the FPP. Generally, this information

is not an image, but image features processed by Q-Eye. Thus, no image transfer are usually needed in Eye-RIS, increasing in this way the frequency of operation. The platform is programmed through the Eye-RIS Application and Development Kit (ADK), an Eclipse-based software development environment. The Eye-RIS ADK is integrated into the ALTERA Nios II Integrated Development Environment (Nios II IDE). In order to program the Q-Eye, FPP code, a specific programming language, was developed. The Nios II is programmed, instead, using standard C/C++ programming language. In addition, the Eye-RIS ADK includes two different function libraries to ease developing applications. The FPP Image Processing Library which provides functions to implement some basic image processing operators such as arithmetic, logic and morphologic operations, spatio-temporal filters, thresholding, etc. The Eye-RIS Basic Library which is composed by several C/C++ functions to execute and debug FPP code and to display images. All of these features allow Eye-RIS Vision System to process images at ultra-high speed but still with very low power consumption.

4. THE IMPLEMENTATION ON THE EYE-RIS

The algorithm described in the previous sections was implemented on the Eye-RIS platform. The integrated CMOS sensor was exploited to acquire the image through a fish-eye lens positioned vertically upward. The acquired image provides a 360 grayscale view of the environment. The outputs of the EMDs are calculated according to a formula described in eq. 5. As already stated, according to the model, the acquired image should be divided to emulate the acceptance angle of the drosophila ommatidium. However, in order to exploit the capability of the Eye-RIS visual processing system, when possible the processing was performed on the full image obtaining a parallel processing of the ommatidia.

A first implementation was performed considering the original algorithm reported in the block scheme of Fig. 1. All the blocks are implemented in c code on the Nios II microprocessor and the Q-Eye chip is used only to acquire the image from the fish-eye. The execution time of the whole algorithm, from the acquisition of the frame to the output of the robot action, is 100 ms about. To further exploit the parallel computational capabilities of the Q-eye chip, another implementation was proposed considering the simplified algorithm reported in Fig. 2 where the most problematic function, the cross product, was substituted with logical operations considering the input data not with gray-scale but with binary images. This modified version produces results qualitatively equivalent to the standard model but needs a fine tuning of the thresholds used during the image conversion.

The code used to implement the high pass and low pass filter on the Eye-RIS is here reported:

```
void section HP(int img, int hp, int t)
Move_loadImage(img, LAM_0, GREY); (i.e. I(t))
Move_shiftImage(LAM_0, LAM_1, SHIFT_DOWN, 1, 0); (i.e. I(t-1))
Arith_absDifference(LAM_1, LAM_0, LAM_2); (i.e. HP(t))
Move_downloadImage(LAM_2, hp, GREY, 0, t);

void section LP(int lp, int hp, int t)
Move_loadImage(lp, LAM_0, GREY);
Move_loadImage(hp, LAM_1, GREY);
Move_shiftImage(LAM_0, LAM_2, SHIFT_DOWN, 1, 0); (i.e. LP(t-1))
Arith_macOperation(LAM_2, 0, LAM_3, 0, 1/2); (i.e. 1/2 * LP(t-1))
Arith_macOperation(LAM_3, 0, LAM_4, 0, 1/2); (i.e. 1/4 * LP(t-1))
Arith_macOperation(LAM_1, 0, LAM_2, 0, 1/2); (i.e. 1/2 * HP(t))
Arith_macOperation(LAM_2, 0, LAM_3, 0, 1/2); (i.e. 1/4 * HP(t))
Arith_add(LAM_2, LAM_3, LAM_5); (i.e. 3/4 * HP(t))
Arith_add(LAM_4, LAM_5, LAM_6); (i.e. 1/4 * LP(t-1) + 3/4 * HP(t))
Move_downloadImage(LAM_6, lp, GREY, 0, t);
```

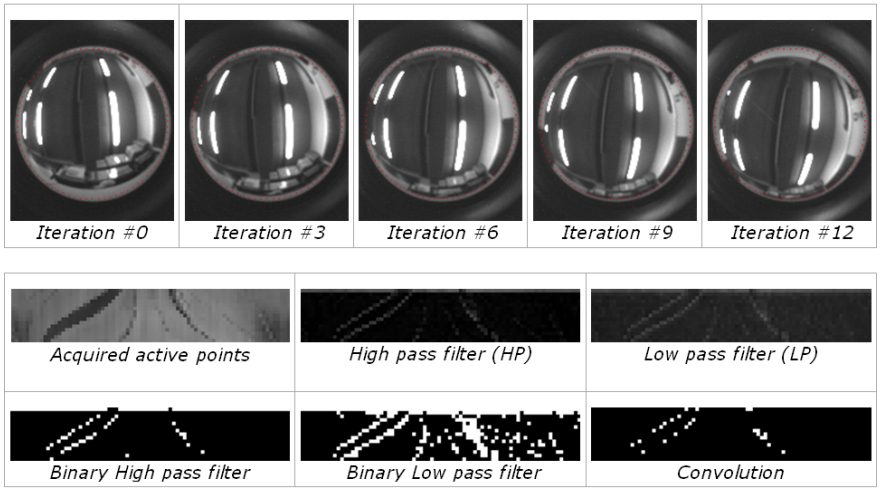


Figure 3. Data flow of the images processed through the Eye-RIS vision system.

Moreover the convolution operation performed with logical operator in the digital domain together with the final integration stage is reported in the following:

```
void section Convolution(int src1, int src2, int dst, int t)
Move_loadImage(src1, LDM_0, BINARY);
Move_loadImage(src2, LDM_1, BINARY);
//shift images
Move_shiftImage(LDM_0, LDM_2, SHIFT_LEFT, 1, BINARY_BLACK);
Move_shiftImage(LDM_1, LDM_3, SHIFT_LEFT, 1, BINARY_BLACK);
//logic AND
Logic_and(LDM_0, LDM_3, LDM_0);
Logic_and(LDM_1, LDM_2, LDM_1);
//logic XOR
Logic_xor(LDM_0, LDM_1, LDM_2);
Move_downloadImage(LDM_2, dst, BINARY, 0, t);

int section Integration(int src, int compartment)
Move_loadImage(src, LDM_0, BINARY);
Move_loadImage(compartment, LDM_1, BINARY);
Logic_and(LDM_0, LDM_1, LDM_2);
return ActivePoints_count(LDM_2);
```

The input images with the active points (i.e. 78 points are acquired on the horizon) and the step by step output of the different processing stages are reported in Fig.3.

The time needed to acquire the image from the panoramic lens and to extract the relevant information corresponding to a ring across the horizon is about 26 ms. This includes also the creation of a new image composed by only 78 pixels that will be used in all the successive steps as input data. The rest of operations that are reported in Fig.2 needs about 3 ms.

The improvement obtained is important to generate more reliable command actions increasing the image acquisition rate having, in this way, a smoother control of the robot.

5. THE ROBOTIC PLATFORM

The roving platform used for navigation experiments, is a modified version of the dual drive Lynx Motion rover, called Rover (Fig. 4). It is equipped with an on-board netbook based on Intel Atom 455, two sonar range sensors (detection range 3 cm to 3 m), a low level target sensor used to detect color spots on the ground, the Eye-RIS v1.3 visual system with a panoramic lens, a 640x480 webcam and the Eye-RIS v2.1 visual system mounted on a pan-tilt module for frontal view purpose.

Figure 4. Robot Rover.

The robot is also equipped with the STEVAL-MKI062V2 board.[10] It is the second generation of the iNEMO module family that combines accelerometers, gyroscopes and magnetometers with pressure and temperature sensors to provide 3-axis sensing of linear, angular and magnetic motion, complemented with temperature and barometer/altitude readings. In the robot, the iNemo inertial motion unit (IMU) is used to close the control loop while the robot is performing movements (i.e. forward movements and rotations). Finally the robot is completely autonomous for the power supply. One 14V, 5A Li Poly battery packs is used to guarantee an autonomy of about 2.5 hours.

The complete control architecture, reported in fig. 5, shows the distributed architecture used to control all the activities of the robot. All the module of the robot are independent and the communication is implemented via a Master-Slave protocol on RS485 bus at 1 Mbit. The protocol permits to exchange information among different nodes and supports connections such as TTL, RS485, and XBee. A generic node on the network is identified with a unique identifier (ID) and it is controlled by Packet communication. The main controller (in our case the netbook) and a generic slave communicate each other by sending and receiving data called Packet. We can consider two kinds of packets:

- **Instruction Packet**: set of instructions that the Main Controller sends to control a slave;
- **Status Packet**: response of the slave to the Main Controller.

ID is a specific number used to distinguish each slave when several slaves are linked to one bus. By giving IDs to Instruction and Status Packets, the Main Controller can control only the addressed slave.

The Rover was designed and built to serve as an autonomous setup where to easily develop and test the behavior of new control models inspired to an insect brain architecture. Due to the expected complexity of the whole architecture, the net book was conceived to act as the master for facing with all the computational burden, with the aim to use all the other devices as slave peripherals useful for efficiently interact with the environment. In this specific application, the main attention was focussed towards outlining the capabilities of the embedded Eye-Ris architecture. Therefore the whole visual orientation algorithm was completely implemented on the visual processor, sharing the load between the analog focal plane processor and the embedded digital control part. The netbook in this case acts merely as a bridge, used to acquire sensory data and to handle the motor control layer. This further enhances and exploits the characteristics of the visual processors that implements a key part of the insect brain devoted to orientation control.

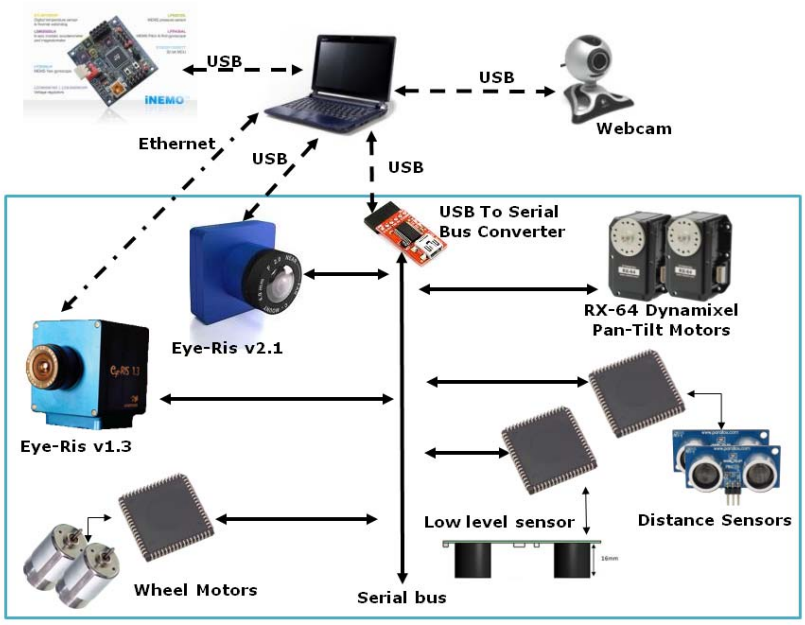


Figure 5. Rover control architecture.

6. EXPERIMENTAL SETUP AND RESULTS

The testing environments, like for the fly experiments,[5] consist in a squared arena with black stripes placed on the walls in different configurations, arranged in such a way to be clearly distinguishable by the robot visual system. This result was obtained considering a compromise between the limited resolution of the Eye-Ris system, further affected by the panoramic configuration and by the fact that the useful area of the whole image was, in this application, restricted to an azimuthal ring. The robot navigates in the environment attracted by these black lines that create a high contrast with the white walls. To visualize in real-time the visual flow coming processed by the algorithm, a graphical interface is available on the computer (see Fig. 6).

The typical parameters considered relevant to derive quantitatively analyze experimental data in the fly are:

- the dwelling time, i.e. the time spent by the animal in a particular site of the arena. This quantity, integrated over a sufficiently long time window, allows to capture the positions more often visited by the animal during the experiments;
- the gaze orientation, i.e. the direction of heading of the animal during the experiment.

Also this is an integrated measure which outlines, in our case, the fraction of time spent by the animal to head towards particular directions. Direct comparisons between fly and robot experimental campaign allow to analyze the suitability of the model implemented, outlining possible discrepancies for further refinement.

The robot is equipped with distance sensors to avoid collisions with the walls and with the Eye-Ris visual processor where the control architecture is implemented. The camera includes a fish-eye lens useful for mimicking the 320^o visual field present in flies. Statistical data acquired from biological experiments can be compared with data acquired from the robotic campaign.

A typical trajectory obtained while the robot is approaching a black stripe on the wall, is reported in Fig. 7 where the acquired images and the corresponding winning sectors are reported during the path.

Upon detection of an obstacle, the Rover adopt an avoiding action and the orientation algorithm is reset and starts again from the new position. Several experiments were conducted, fig. 8 shows the experimental results obtained in an arena

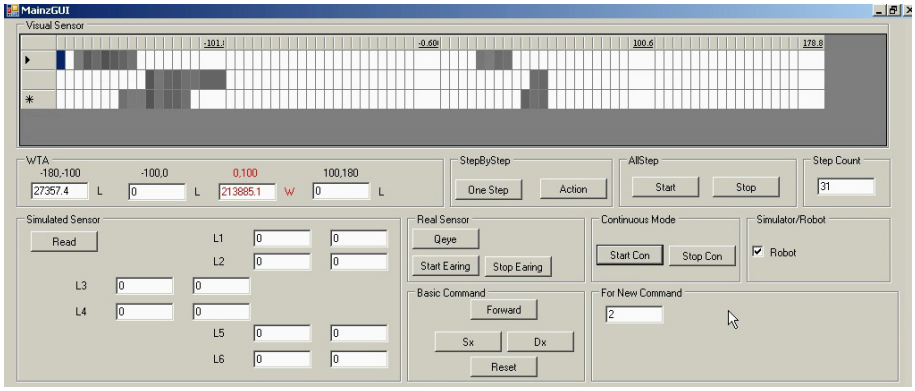


Figure 6. Graphical interface showing in real time the visual input for the EMD network. Each row is constituted by 78 elements and consecutive rows represents the evolution of the visual flow in time.

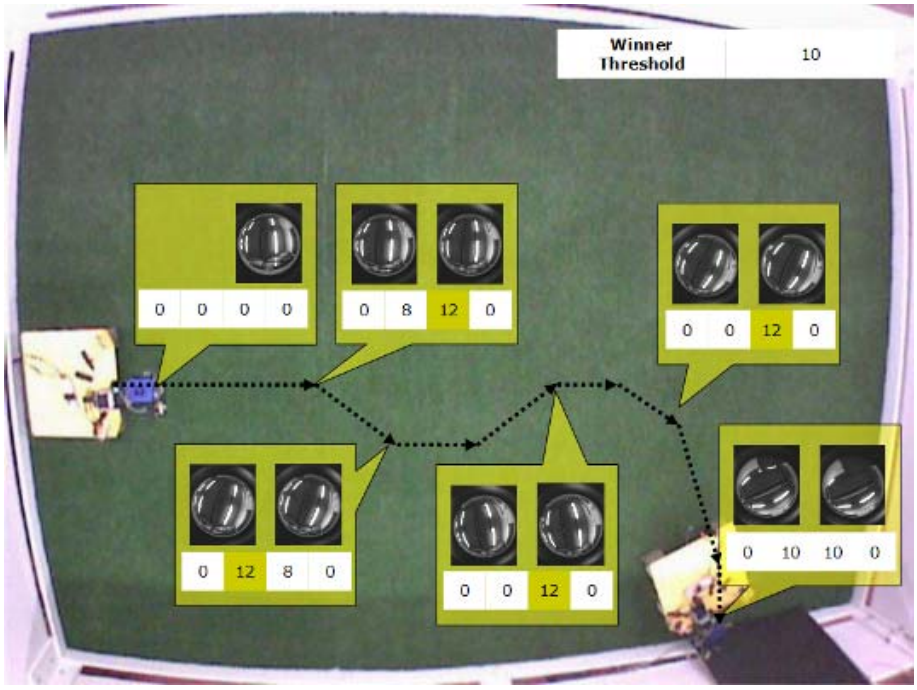
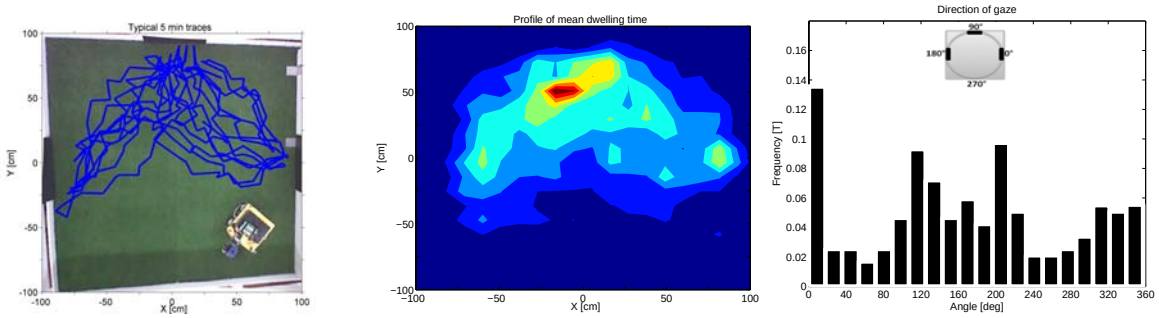
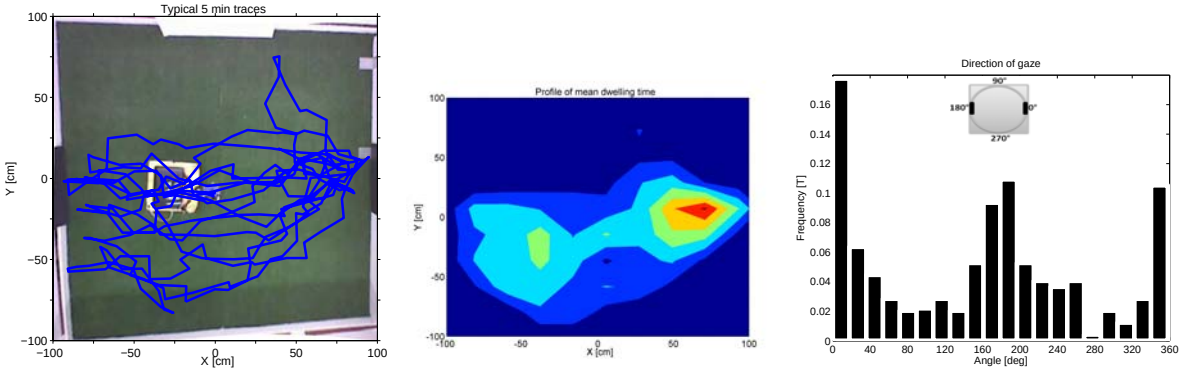


Figure 7. Typical trajectory shown by the robot in approaching a black stripe on the wall. The acquired images and the corresponding winning sectors are also reported during the trajectory.



(a) (b) (c)

Figure 8. Behaviour of a roving robot involved in a visual orientation task with 3 landmarks. (a) robot trajectory, (b) density probability of visits in the arena, (c) gaze distribution.



(a) (b) (c)

Figure 9. Behaviour of a roving robot involved in a visual orientation task with 2 landmarks located in opposite walls. (a) robot trajectory, (b) density probability of visits in the arena, (c) gaze distribution.

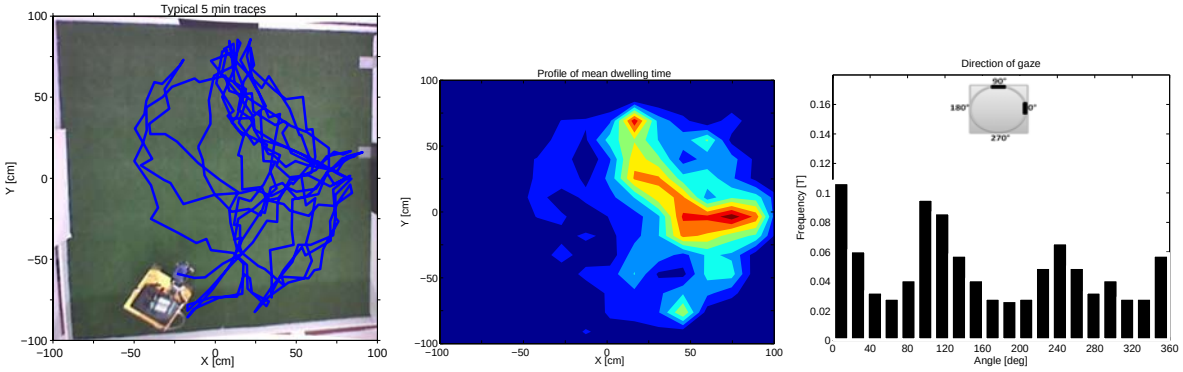
with a constellation of three landmarks, used to make a comparison with the fly experiments reported in literature.[5] As already introduced, the variable of interest are: the trajectory followed by the robot during a single trial consisting of 5 minutes recording; the dwelling time that represents the density probability of visits in the arena obtained averaging the results over 15 trials and the gaze orientation that shows a statistical distribution of the robot heading angle (i.e. where the robot attention is mostly directed).

The matching between the real fly and the Rover controlled through the proposed orientation architecture allows the fine tuning of the model parameters and validation could be easily performed modifying the set-up of the arena.

The implementation on a smart visual processor guarantees high speed performances: using the Eye-Ris a controlling cycle needs 29 ms for the sensor acquisition and processing, 32.5 ms for the motor commands and 16.6 ms for communication routines obtaining a time of about 78 ms for the complete processing.

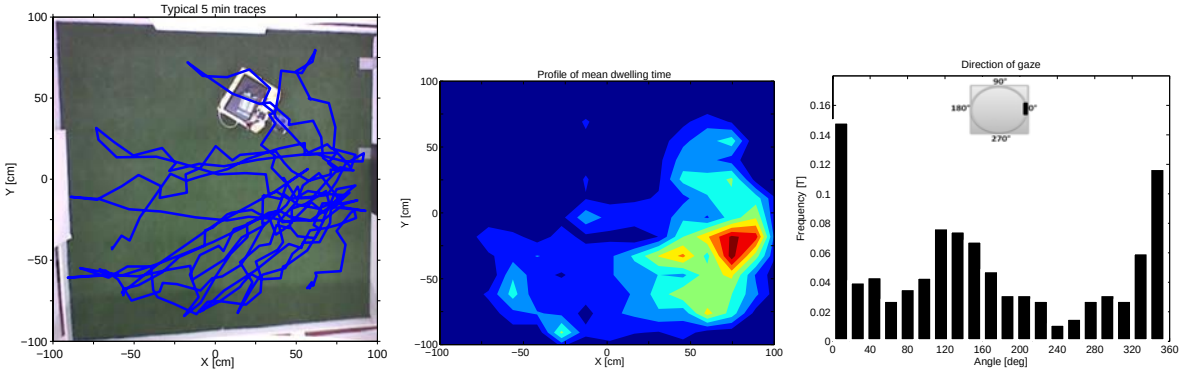
Depending on the configuration of the attractive landmarks in the arena, the behaviour of the fly changes and the robot behaviour changes accordingly as shown in[5] and in fig. 9 where two objects are placed on opposite walls and in Figure 10 where the objects are located in two adjacent walls.

Finally the behaviour with a single landmark was also investigated and the results are reported in Figure 11.



(a) (b) (c)

Figure 10. Behaviour of a roving robot involved in a visual orientation task with 2 landmarks located on two adjacent walls. (a) robot trajectory, (b) density probability of visits in the arena, (c) gaze distribution.



(a) (b) (c)

Figure 11. Behaviour of a roving robot involved in a visual orientation task with only 1 landmark. (a) robot trajectory, (b) density probability of visits in the arena, (c) gaze distribution.

7. CONCLUSIONS

In this paper the Eye-Ris vision system capabilities are investigated to implement the Drosophila orientation model. The Eye-Ris vision system is equipped with the Q-Eye vision chip that is a CNN-based CMOS high-speed image sensor-processor able to sense and process on-the-fly images at extreme frame rates. It provides high parallelism between image sensing, pre-processing and readout. The Rover autonomous robot was used and it is equipped with Eye-Ris with fish-eye lens in order to acquire a 360 view of the environment in a unique step. Also a simplified version of the Drosophila orientation model is implemented directly on the Eye-Ris permitting to complete a control loop in a time compatible with real time application. Finally a set of experiments are reported in order to show the matching between the fly and robot behaviour.

ACKNOWLEDGMENTS

The authors acknowledge the support of the European Commission under the project FP7-ICT-2007-1 216227 SPARK II and FP7-ICT-2009-6 270182 EMICAB.

REFERENCES

[1] Neuse, K., Triphan, T., Mronz, M., Poeck, B. and Strauss, R., "Analysis of a spatial orientation memory in Drosophila", *Nature*, 453, 1244-1247 (2008).

[2] Borst, A., Egelhaaf, M., "Principles of visual motion detection", *Trends Neurosci*, 12, 297-306 (1989).

[3] Ilda, F., "Biologically inspired visual odometer for navigation of a flying robot", *Robotics and Autonomus System*, 44, 204-208 (2003).

[4] Neumann, TR., Bulthoff, HH., "Insect inspired control of Translatory flight", *Advances in Artifgicial Life - Proceedings of the 6th European Conference*, Springer-Verlag, Berlin, 627-636 (2001).

[5] Mronz, M. and Strauss, R., "Visual motion integration controls attractiveness of objects in walking flies and a mobile robot", *IEEE / RSJ International Conference on Intelligent Robots and Systems*, Nice, 3559-3564, (2008).

[6] Alba, L., Arena, P., De Fiore, S., Patané, L., Strauss, R., Vagliasindi, G., "Implementation of a drosophila-inspired orientation model on the Eye-Ris platform", *Cellular Nanoscale Networks and Their Applications CNNA*, (2010).

[7] Rodriguez-Vazquez, A., et al, "The Eye-RIS CMOS Vision System", in *Analog Circuit Design*, Springer, 15-32 (2007).

[8] *ANAFOCUS home page* [Online]. Available: http://www.anafocus.com (2011)

[9] Liñàn, G., Domìguez-Castro, R., Espejo, S. and Rodrìguez-Vàzquez, A., "ACE16K: A programmable focal plane vision processor with 128×128 resolution", in *Proc. of ECTD 01*, Espoo, Finland, (2001).

[10] *iNemo producer home page* [Online]. Available: http://www.st.com/inemo (2011)