

# Development of a Bio-inspired Vision System for Mobile Micro-robots

Cheng Hu, Farshad Arvin and Shigang Yue  
Computational Intelligence Lab (CIL), School of Computer Science  
University of Lincoln, Brayford Pool  
Lincoln, LN6 7TS, UK  
Email: {chu, farvin, syue}@lincoln.ac.uk

**Abstract**—In this paper, we present a new bio-inspired vision system for mobile micro-robots. The processing method takes inspiration from vision of locusts in detecting the fast approaching objects. Research suggested that locusts use wide-field visual neuron called the lobula giant movement detector to respond to imminent collisions. We employed the locusts' vision mechanism to motion control of a mobile robot. The selected image processing method is implemented on a developed extension module using a low-cost and fast ARM processor. The vision module is placed on top of a micro-robot to control its trajectory and to avoid obstacles. The observed results from several performed experiments demonstrated that the developed extension module and the inspired vision system are feasible to employ as a vision module for obstacle avoidance and motion control.

**Index Terms**—Bio-inspired, Collision avoidance, Locusts vision, Mobile robot

## I. Introduction

The ability to avoid a collision is an important issue for the mobile robots. There are different sensory systems which are used for collision avoidance such as ultrasonic [1], infrared [2, 3], laser [4], radar [5] and vision system [6]. These sensory systems are mainly used on mobile robots for obstacle detection and avoiding the obstacles. However, it is still not easy task for mobile robots to run autonomously in complex environments without human intervention. One of the greatest challenges is to understand and to deal with the dynamic scenes [7] like complex background or rapidly changing the ambient light. Therefore, we need to apply fast and reliable methods to address the problem.

Nature demonstrates variety of the successful methods in collision avoidance, which are employed by very small insects and other animals [8, 9]. For insects such as locusts, an ability to detect the fast approaching objects is important to avoid bumping into each other or caught by predators. After millions of years evolving, these vision-based collision avoidance systems are both reliable and efficient [10, 11]. Therefore, it can be a feasible approach if we take inspiration from nature and apply on autonomous mobile robots.

It is identified anatomically that, there is a wide-field visual neuron that is located in the lobula layer of the locust nervous system called the Lobula Giant Movement Detector (LGMD) [12]. The LGMD spikes when objects approach in a direct collision course rapidly [13]. The LGMD's firing rate

is related to the approaching speed and the texture details of the moving object. Therefore, locusts can act quickly to avoid an imminent collision. LGMD is tightly tuned to respond to objects approaching in a direct collision course [14], however it produces a little or no response to receding objects [13]. Compared to the vision processing systems in the large mammals like humans, LGMD is much simpler. All these features make it an ideal model for developing the specialized fast and low-cost sensory system for autonomous collision avoidance [15-17].

A functional neural network based on the LGMD's input circuitry was developed by Rind and Bramwell [18]. This neural network showed the same selectivity as the LGMD neuron for approaching objects and responded best to objects approaching on collision rather than near-miss trajectories. The expanding edges of colliding objects and the lateral inhibition were the key features computed by the model. This neural network has also been used to mediate collision avoidance in a real-world environment by incorporating it into the control structure of a miniature robot [15, 19].

As a miniature robot, the size and power consumption should be controlled carefully due to the limitations of computing power and hardware resources. In the previous collision avoidance researches based on LGMD [15-17], robots only served the captured images and the motion control, hence the major LGMD processing task is done by the PC-based software like MATLAB. Then the software controls the robot remotely using cable or wireless signals. The whole system is cumbersome and complicated to be utilized in miniature multi-robot systems such as swarm robotics scenarios [20]. In other words, the robots are not independent (or fully autonomous). Therefore, we need a new implementation to avoid the remote controllers and to provide fully autonomy via bio-inspired LGMD-based collision avoidance on a low-cost miniature mobile robot.

In this paper, an extension vision module is developed to imply by a low-cost micro-robot (*Colias*) [21]. The extension module enables the robot to implement collision avoidance method based on the LGMD. The vision module along with the robot is tested in various scenarios and environments to demonstrate the reliability of the system in obstacle avoidance. The observed results showed that the developed vision system for a miniature robot is amenable to be used in real-world autonomous motion applications.

The rest of this paper is organised as follow. Section II

gives a brief description of LGMD algorithm, which also explains our initial work. Section III describes the hardware implementation. Following that, in section IV, we explain the proposed software design. The experiments and results are described in Section V.

## II. Algorithm description

The LGMD algorithm used in this robot is based on our previous model described in Yue and Rind [16] as Fig. 1 shows, with some simplification and approximation. The model is composed of five groups of cells, which are P-cells (photoreceptor), I-cells (inhibitory), E-cells (excitatory), S-cells (summing) and G-cells (grouping) and also two individual cells, namely, the feed-forward inhibitory and LGMD.

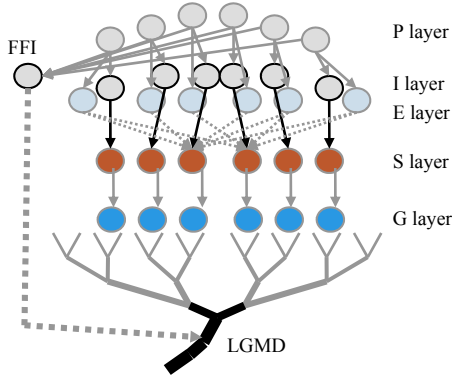


Fig. 1 A schematic of the LGMD based neural network for collision detection. The input of the P cells is the luminance change. Lateral inhibition is indicated with dotted lines and has one frame delay. Excitation is indicated with black lines and has no delay. The FFI also has one frame delay.

The first layer of the neuron is composed by the P cells, which are arranged in a matrix, formed by the change of luminance in adjacent frames captured by camera. Comparing to the former algorithm, the visual persistence part which occupies a lot of computation power is removed. The output of P layer can be described as below:

$$P_f(x, y) = L_f(x, y) - L_{f-1}(x, y) \quad (1)$$

where  $P_f(x, y)$  is the change of luminance of pixel  $(x, y)$  at frame  $f$ ,  $L_f(x, y)$  and  $L_{f-1}(x, y)$  are the luminance in current frame and the previous frame.

Subsequent processing is quite similar to our previous model as well. The output of P cells forms the inputs to two separate cell types in the next layer. One type is the excitatory cells, through which excitation is passed directly to the retinotopic counterpart of the cell in the third layer. The second type of cells are lateral inhibition cells which pass inhibition after one image frame delay to their retinotopical counterpart's neighbouring cells in the E layer.

$$[I_f] = [P_{f-1}] \otimes [w_i] \quad (2)$$

where  $w_i$  is the local inhibiting weight matrix.

The excitation of E cells and the inhibition of I cells are combined in the S layer.

$$S_f = E_f - I_f W_i \quad (3)$$

where  $W_i$  is the inhibiting coefficient.

When reaches the G layer, the expanded edges which represented by clustered excitations are enhanced to extract colliding objects against complex backgrounds. This mechanism is implemented with a passing coefficient for each cell, which is defined by a convolution operation in the S layer. The excitation correspond to each cell becomes:

$$G_f(x, y) = S_f(x, y) C_{e_f}(x, y) \omega^{-1} \quad (4)$$

where  $\omega$  is a scale and computed at every frame. The passing coefficient can be described as:

$$C_{e_f}(x, y) = \sum_{i=-1}^1 \sum_{j=-1}^1 S_f(x+i, y+j) w_e(i, j) \quad (5)$$

where  $w_e(i, j)$  represents the influence of its neighbours and this operation can be simplified as a convolution mask and the passing coefficients can be computed in a matrix.

Following by a threshold set to filter decayed excitations. The membrane potential of the LGMD cell  $K_f$  at frame  $f$  is summed after G layer with a rectifying operation, which will turn the responses in negative values to positive before summing. Then  $K_f$  is then transformed through a normalizer. Considering of saving computing power and easier tuning, a new normalizing function is adopted, given by:

$$\kappa_f = k \cdot (\tanh(K_f - C_1) + C_2) \quad (6)$$

The excitation  $\kappa_f$  varies within  $[0, 1]$ .  $C_1$ ,  $C_2$  and  $k$  are constant. If  $\kappa_f$  exceeds the threshold, a spike is produced.

An impending collision is confirmed after several (in our tests, four) successive spikes generated. However, spikes may be suppressed by the FFI cell when the robot is turning.

It is not surprised during turning, the neuron network may produce spikes and even false collision alerts because of the sudden change in the visual scene. The feed forward inhibition and lateral inhibition work together to cope with such whole field movement. The FFI is proportional to the summation of excitations in all cells with one frame delay.

$$F_f = \sum_{x=1}^n \sum_{y=1}^n (|P_{f-1}(x, y)|) n_{cell}^{-1} \quad (7)$$

Once  $F_f$  exceeds its threshold  $T_{FFI}$ , spikes in the LGMD are inhibited immediately.

As described in the above subsections, the LGMD based collision detection system only involves low level image processing, such as excitation transferring and neighbouring operation; computationally expensive methods, such as object recognition or scene analysis, are not used. Because of this, the collision detection system is able to work in real time and is independent of object classification.

## III. Hardware description

### A. Camera

Camera is the most important sensor in the vision based control of a robot. We chose a low voltage CMOS image

sensor OV7670 module, which has a compact package size of 8x8x4 mm with 24-pin flexible flat cable (FFC) connector. The power supply is from 1.7 V to 3.0 V with active power consumption of 60 mW. The camera is capable of operating up to 30 frames per seconds (fps) in VGA mode with output support for RGB422, RGB565 and YUV422. The viewing angle is approximately 70 degrees. All these features make the camera suitable to be used in miniature mobile robots. As a trade-off for image quality and data consumption, we choose a resolution of 72x99 pixel at 30 fps, with output format of 8-bit YUV422.

Three groups of digital interfaces are used, which are serial camera control bus (SCCB) for camera configuration, clock/timing signals and 8-bit parallel port for image data.

### B. The microcontroller

An ARM® Cortex™-M4F core microcontroller is deployed as the main processor for monitoring all the modules and serves the image processing method. The 32-bit MCU STM32F407 clocked at 168 MHz provides the necessary computational power to have a real-time image processing. It contains 192 Kbyte SRAM that provides enough spaces for image buffering and computing.

The images captured by the camera are transmitted through the digital camera interface (DCMI) which is an embedded camera interface. It can connect to the camera modules and CMOS sensors through an 8-bit to 14-bit parallel interface to receive image data. The camera interface can sustain a data transfer rate up to 54 Mbyte/s at 54 MHz.

Images received by DCMI are transmitted into SRAM through a specific DMA channel. There are two image buffers in SRAM to store the raw images which can be read and written simultaneously. To maintain the efficiency, when CPU deals with one of the buffers, the other one receives the next frame. Fig. 2 shows the proposed architecture for the vision module.

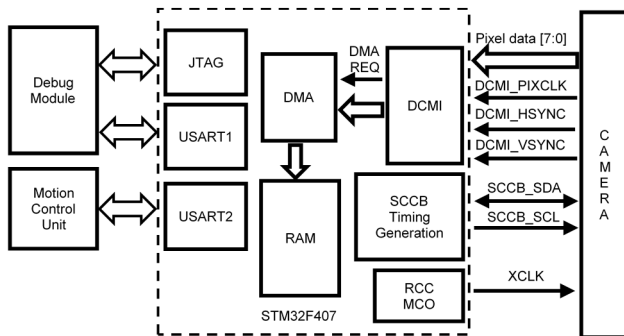


Fig. 2 Hardware architecture of the extension vision module

### C. Robot Platform

Robot platform *Colias* with diameter of 4 cm is the platform which is employed in this research. The developed extension vision module is placed on top of the basic *Colias*.

The bottom board provides motion and power management systems. It uses an AVR microcontroller with 8 MHz clock source. Two micro DC motors employing direct gears and two wheels with diameter of 2.2 cm actuate *Colias* with a maximum speed of 35 cm/s. Motors are controlled individually using a pulse-width modulation (PWM)

technique [22]. Fig. 3 shows the utilized robot platform

### D. Debugging module

A full-duplex serial port is applied as the debugging interface. The robot sends image samples and algorithm data to PC when debugging is enabled. Meanwhile, the robot can receive configuration commands from PC, including commands for camera, the algorithm and reaction behaviour. The baud rate is set at 256 Kbps, with eight data bits and one stop bit.

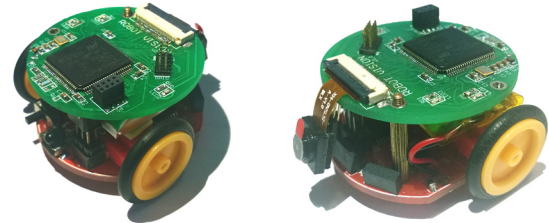


Fig. 3 The pictures of the developed micro-robot from back and front side. Upper board (green) executes LGMD based control and the bottom part (red) is the actuator.

## IV. Software design

The software design mainly includes four parts: i) image acquisition, ii) the LGMD algorithm, iii) the decision making and iv) motion control. The flowchart of the system is shown in Fig. 4.

### A. Camera configuration and image acquisition

The camera has hundreds of build-in registers, providing various functions like scaling, image enhancement, output format and data communication rate. Most of them remain default, while some need to be configured separately.

### B. The LGMD algorithm

As described in the previous sections, the LGMD algorithm is the most complicated computing task of the system. Working at 30 fps, CPU has only about 30 ms to accomplish the whole process. Due to the limited computing power, minor simplifications and approximations are taken. Convolution operations at edge pixels are skipped. Consequently, black frames of layer S and G appeared, as shown in Fig. 5.

### C. Decision making and control commands

There are three types of motion control commands, which are 'F' for going forward, 'L' or 'R' for turn left or right and 'S' for stop. The decision is determined by both LGMD and FFI outputs. The control commands are defined in TABLE I.

1. By default, LGMD and FFI stays under threshold, which means the environment is safe for robot to go forward. Thus, the command 'F' is given to the motion control unit.

2. If some objects move towards the robot or the robot attempts to move towards them, the image of certain object in the robot's eye expands, which leads to an output spike of LGMD cell increases under appropriate expanding speed. Meanwhile, the FFI cell does not produce output since there is no sudden change visually. This situation is judged as a risk of collision, thus the robot turns immediately.

It should be noted that, LGMD cell cannot tell where the object exactly is, so the turning direction can be chosen

randomly. To imitate a real animal and avoid swing from side to side, the robot is designed to have a higher chance to turn right (75%) than left (25%).

3. During turning phase, there would be whole-frame movement, which activates the outputs of FFI cell. In this case, a command 'S' is sent to the motion control unit causing a pause in its motion after turning action. When confirmed the new direction is safe, the robot will move on.

TABLE I. CONTROL COMMANDS DEFINITION

Neuron Status		Decision	Command word
LGMD	FFI		
0	0	Go forward	'F'
1	0	Turn left or right	'L' or 'R'
X	1	Stop	'S'

## V. Experiments and results

Several experiments are performed to test the sensitivity and robustness of the system. One is LGMD processing test which mainly tests the performance of the algorithm. Then the system is challenged when combined with motion controller when wandering in a certain arena.

### A. LGMD processing

#### 1) Approaching object

First we tested the sensitivity and robustness of the LGMD output. A plank was set on a plain table with adjustable inclination  $\theta^\circ$ , as illustrated in Fig. 6. A guide track was set diagonally along the plank that allows a tennis ball ( $\varphi=66$  mm) which starting from rest at the top of the track roll down to the table in a certain trajectory with different acceleration when  $\theta$  changes. The robot is placed in fix position on the table.

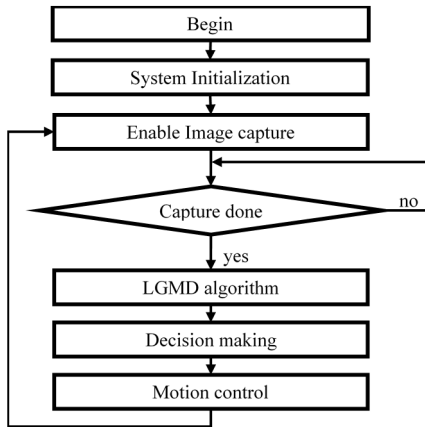


Fig. 4 The flow chart of software processing.

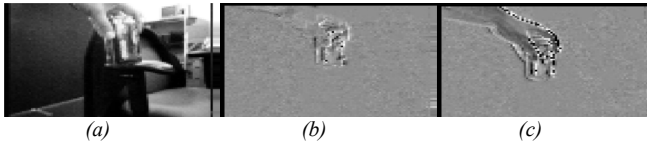


Fig. 5 Different layers of LGMD processing. (a) shows the original image, which is a hand holding a jar waving in front of the camera; (b) shows the output of P layer. The background detail is inhibited, while the hand with the jar stands out; (c) shows the output of G layer. The most significant difference comparing to the P layer is the enhanced edges. Note the black frames in the image of P and G layer, that is because of those incomplete convolution which are eliminated.

Several sets of tests were taken respectively. Each set were repeated 15 times. The average of the obtained results are shown in Fig. 7 (a). We can see that the model works well on every set of experiments, which offers alerts on approaching balls, although their speed are different.

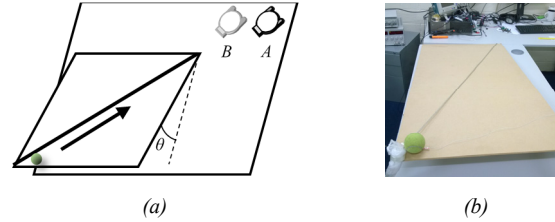


Fig. 6 Testing table for LGMD processing. In (a), the robot (A) is placed on the table surface, fixed in the trajectory of the tennis ball in the first experiment; and different distances away from the trajectory in the second experiment (B). (b) shows a photo of experiment setup. The vision module is at the upright corner of the photo.

#### 2) Passing object

Another experiment is designed for testing the behaviour of the model when object pass by. Based on the first testing environment, we put the robot aside from the trajectory with certain offset. Experiments are executed in 5 sets, each set is repeated 15 times. The average of captured results are shown in Fig. 7 (b).

From the records we can find that the LGMD output increases at first, indicates the approaching ball, but soon after the ball moves out of sight, the LGMD output drops immediately. FFI outputs also increase along with the LGMD, with smaller growth. Comparing the distance from the robot to the trajectory of tennis ball and the LGMD output, they showed a negative correlation as expected.

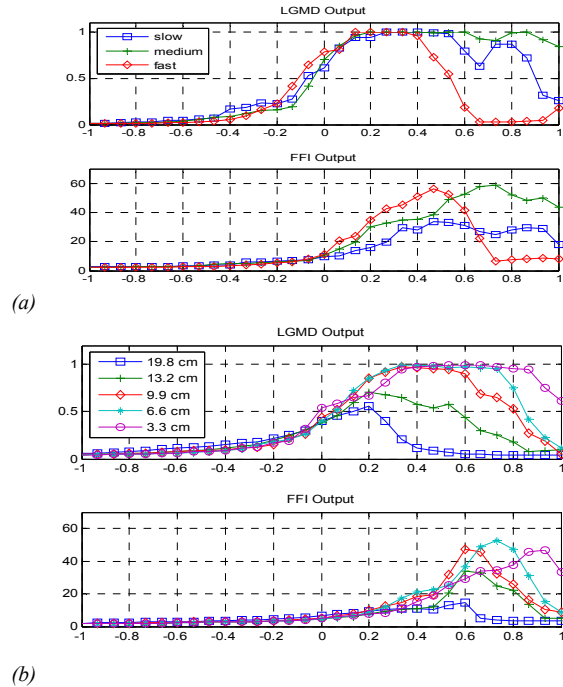


Fig. 7 Average records for each set in experiment A. LGMD and FFI outputs are shown. The x axis represents time in seconds, y axis is for neuron network output. Records are aligned at when the outputs exceed LGMD threshold, which are set time zero. (a) records of approaching object experiments with different speed; (b) records of passing object experiments with different offset from the robot.



### B. Vigilance distance at different speed

The obstacle avoidance behaviour performances differently under different moving speed. It is important to estimate the distance between the robot and obstacle when LGMD exceeds the threshold during the approaching a certain obstacle at different speeds. We called this distance the vigilance distance.

We decorated a wall with texture of black and white squares that provides clear edges as shown in Fig. 8. Robot starts running towards the wall 50 cm away until the vigilance distance is reached, then it halts. Experiments are taken in nine groups, with different speed ranging from 1.5 to 17 cm/s. The results are shown in Fig. 9.

As expected, the vigilance distance increases as the robot moves faster. When speed is between 5 and 14 cm/s, the robot performed coherent. When the robot moves at a high speed (e.g., 17 cm/s), the movement becomes unstable and image is blurred, causing fake alarms.

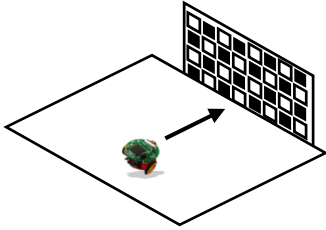


Fig. 8 The environment set for testing the vigilance distance. Textured wall provides clear and stable clues for the robot.

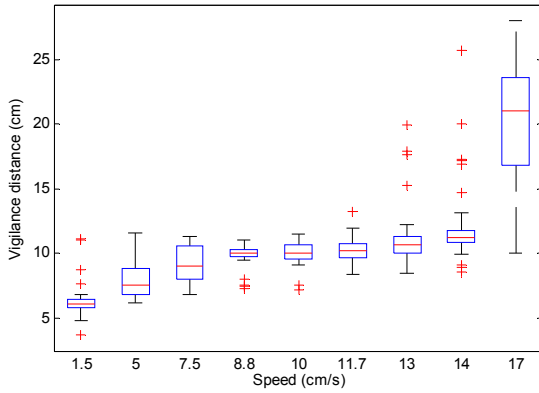


Fig. 9 Result in diagram of the nine groups of experiment B. For each group, the central mark is the median, blue square is formed by the first and third quartiles. Outliers are represented by red pluses.

### C. Real time tests

In the previous experiments, obstacles and scenes remained unchanged for each test. However, the real world test provides unpredictable situations anytime. Thus it is the best way to test if the collision detection system works reliably.

In the real time tests, the LGMD algorithm works together with collision avoidance commands. As mentioned earlier, the LGMD based collision detection can deal with more complex situations. The background used in the experiments are kept complex as well as inhomogeneous ambient light. The robot is allowed to move at the speed of about 10 cm/s in the arena. Once an imminent collision is detected, the robot

turns without the debugging unit. The experiment lasts for at least 5 minutes without collision as in Fig. 10. Because of the biases of turning direction, the robot behaves more like an insect. Images taken from the robot's eye in the test are shown in Fig. 12. Fig. 11 shows a series of the LGMD and FFI outputs during the test.

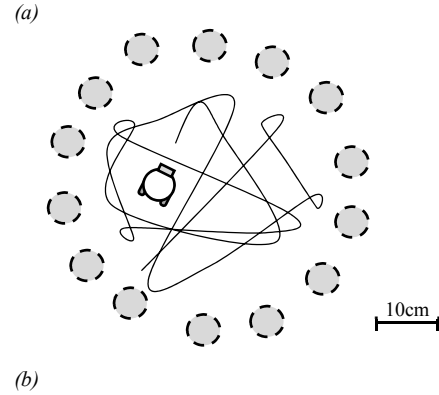
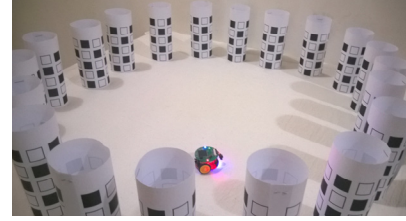


Fig. 10 (a) the photo of experiment set for the real time test. The robot moves within an arena surrounded by several textured paper rolls; (b) A brief example of the experiment. The solid line indicates the trajectory. Note that right turns are more frequent than left turns.

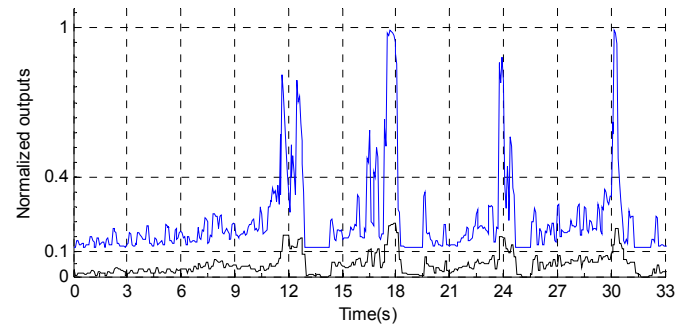


Fig. 11 A series of both LGMD and FFI normalized outputs under real time test. The robot has moved for more than 30 seconds. The X axis represents the time in seconds and Y axis is for the normalized outputs within [0,1]. The upper blue trace shows the LGMD output; the lower black trace shows the FFI output. During the time period, 4 successive turning was executed at around 10s, 17s, 23s and 30s.

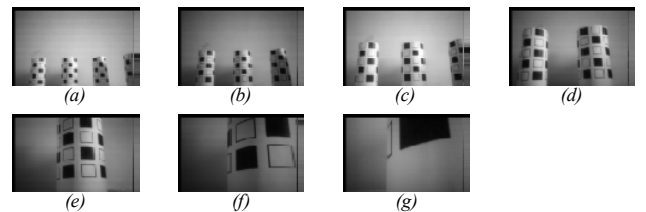


Fig. 12 Sample frames taken by the robot while approaching paper poles. Distances from the robot are 60cm(a), 50cm(b), 40cm(c), 30cm(d), 20cm(e), 10cm(f) and 5cm(g).

## VI. Conclusion

An extended vision module for a new low-cost micro-robot is presented in this paper, which enables implementing of collision avoidance method based on bio-inspired LGMD. The robot uses its on-board LGMD algorithm to detect imminent collision with obstacles using tiny camera module and executes avoiding behaviour. Results of the performed experiments demonstrated that the developed module has strong robustness to adapt environment and it can be employed in real-world applications. For the future work, several number of robots will be equipped with the vision module to test the method in multi-robot scenarios.

## ACKNOWLEDGMENT

This work was supported by EU FP7-IRSES projects EYE2E (269118), LIVCODE (295151) and HAZCEPT (318907).

## REFERENCES

- [1] H. Everett, *Sensors for mobile robots: theory and application*: AK Peters, Ltd., 1995.
- [2] G. Benet, F. Blanes, J. E. Simó, and P. Pérez, "Using infrared sensors for distance measurement in mobile robots," *Robotics and autonomous systems*, vol. 40, pp. 255-266, 2002.
- [3] F. Arvin, K. Samsudin, and R. Ramli, "Development of IR-based short-range communication techniques for swarm robot applications," *Advances in Electrical and Computer Engineering*, vol. 10, pp. 61-68, 2010.
- [4] G. v. Wichert, "Can robots learn to see?," *Control engineering practice*, vol. 7, pp. 783-795, 1999.
- [5] R. Manduchi, A. Castano, A. Talukder, and L. Matthies, "Obstacle detection and terrain classification for autonomous off-road navigation," *Autonomous robots*, vol. 18, pp. 81-102, 2005.
- [6] S. Yue, R. D. Santer, Y. Yamawaki, and F. C. Rind, "Reactive direction control for a mobile robot: a locust-like control of escape direction emerges when a bilateral pair of model locust visual neurons are integrated," *Autonomous Robots*, vol. 28, pp. 151-167, 2010.
- [7] H. Buxton, "Learning and understanding dynamic scene activity: a review," *Image and vision computing*, vol. 21, pp. 125-136, 2003.
- [8] F. C. Rind and P. J. Simmons, "Seeing what is coming: building collision-sensitive neurones," *Trends in neurosciences*, vol. 22, pp. 215-220, 1999.
- [9] R. D. Santer, P. J. Simmons, and F. C. Rind, "Gliding behaviour elicited by lateral looming stimuli in flying locusts," *Journal of Comparative Physiology A*, vol. 191, pp. 61-73, 2005.
- [10] P. S. Bhagavatula, C. Claudianos, M. R. Ibbotson, and M. V. Srinivasan, "Behavioral Lateralization and Optimal Route Choice in Flying Budgerigars," *PLoS computational biology*, vol. 10, p. e1003473, 2014.
- [11] A. C. Paulk, J. A. Stacey, T. W. Pearson, G. J. Taylor, R. J. Moore, M. V. Srinivasan, *et al.*, "Selective attention in the honeybee optic lobes precedes behavioral choices," *Proceedings of the National Academy of Sciences*, vol. 111, pp. 5006-5011, 2014.
- [12] M. O'shea, C. Rowell, and J. Williams, "The anatomy of a locust visual interneurone; the descending contralateral movement detector," *Journal of Experimental Biology*, vol. 60, pp. 1-12, 1974.
- [13] F. C. Rind and P. J. Simmons, "Orthopteran DCMD neuron: a reevaluation of responses to moving objects. I. Selective responses to approaching objects," *Journal of Neurophysiology*, vol. 68, pp. 1654-1654, 1992.
- [14] S. Judge and F. Rind, "The locust DCMD, a movement-detecting neurone tightly tuned to collision trajectories," *Journal of Experimental Biology*, vol. 200, pp. 2209-2216, 1997.
- [15] M. Blanchard, F. C. Rind, and P. F. Verschure, "Collision avoidance using a model of the locust LGMD neuron," *Robotics and Autonomous Systems*, vol. 30, pp. 17-38, 2000.
- [16] S. Yue and F. C. Rind, "Collision detection in complex dynamic scenes using an LGMD-based visual neural network with feature enhancement," *Neural Networks, IEEE Transactions on*, vol. 17, pp. 705-716, 2006.
- [17] S. Yue and F. C. Rind, "Near range path navigation using LGMD visual neural networks," in *Computer Science and Information Technology, 2009. ICCSIT 2009. 2nd IEEE International Conference on*, 2009, pp. 105-109.
- [18] F. C. Rind and D. Bramwell, "Neural network based on the input organization of an identified neuron signaling impending collision," *Journal of Neurophysiology*, vol. 75, pp. 967-985, 1996.
- [19] M. Blanchard, P. F. Verschure, and F. C. Rind, "Using a mobile robot to study locust collision avoidance responses," *International Journal of Neural Systems*, vol. 9, pp. 405-410, 1999.
- [20] F. Arvin, A. E. Turgut, F. Bazyari, K. B. Arikan, N. Bellotto, and S. Yue, "Cue-based aggregation with a mobile robot swarm: a novel fuzzy-based method," *Adaptive Behavior*, vol. 22, pp. 189-206, 2014.
- [21] F. Arvin, J. Murray, C. Zhang, and S. Yue, "Colias: An Autonomous Micro Robot for Swarm Robotic Applications," *International Journal of Advanced Robotic Systems*, vol. 11, pp. 1-10, 2014.
- [22] F. Arvin and M. Bekravi, "Encoderless position estimation and error correction techniques for miniature mobile robots," *Turkish Journal of Electrical Engineering & Computer Sciences*, vol. 21, pp. 1631-1645, 2013.