

Integrated moment-based LGMD and deep reinforcement learning for UAV obstacle avoidance

Lei He¹, Nabil Aouf², James F. Whidborne³ and Bifeng Song¹

Abstract—In this paper, a bio-inspired monocular vision perception method combined with a learning-based reaction local planner for obstacle avoidance of micro UAVs is presented. The system is more computationally efficient than other vision-based perception and navigation methods such as SLAM and optical flow because it does not need to calculate accurate distances. To improve the robustness of perception against illuminance change, the input image is remapped using image moment which is independent of illuminance variation. After perception, a local planner is trained using deep reinforcement learning for mapless navigation. The proposed perception and navigation methods are evaluated in some realistic simulation environments. The result shows that this light-weight monocular perception and navigation system works well in different complex environments without accurate depth information.

I. INTRODUCTION

Collision avoidance is an essential feature for Unmanned Aerial Vehicle (UAV) to navigate autonomously in complex environments [1]. Traditional methods for collision avoidance are mainly based on accurate distance measurements obtained by ranging sensors [2], RGB-D cameras [3], [4] and LiDAR [5]. For small or micro UAVs, most ranging sensors are too heavy and power-consuming; hence some researchers have considered binocular/stereo vision [6], [7], which can provide a depth estimate with acceptable accuracy. However, the detection range and the accuracy of depth prediction are related to its baseline length which is limited on micro UAVs. Monocular cameras, on the other hand, can provide rich visual information about the environment and are low-cost and light-weight. They have potential for depth perception by using monocular cues and are suitable for micro UAVs with Size, Weight, and Power (SWaP) constraints [8].

Obstacle detection and avoidance using monocular vision is an interesting but challenging task because it is hard to get depth information directly from a single image. Visual Simultaneous Localization And Mapping (VSLAM) is a promising technique to obtain reliable environment awareness, and it can be applied with only monocular vision, e.g. ORB-SLAM [9], or monocular vision combined with IMU information, e.g. VINS-Mono [10], [11]. Based

*This work was supported by China Scholarship Council. (No. 201806290175)

¹Lei He and Bifeng Song are with the School of Aeronautics, Northwestern Polytechnical University, Xi'an, 710072, China heleidsn@gmail.com, bfsong@nwpu.edu.cn

²Nabil Aouf is with the Dept of Electrical and Electronic Engineering, City, University of London, London EC1V 0HB, UK nabil.aouf@city.ac.uk

³James F. Whidborne is with the Centre for Aeronautics, Cranfield University, Bedfordshire, MK43 0AL, UK j.f.whidborne@cranfield.ac.uk

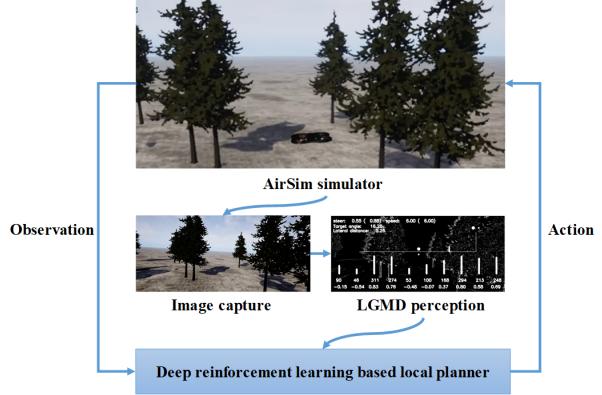


Fig. 1: Simulation environment.

on the local position and map provided by SLAM, many path planners have been proposed to navigate UAVs in an unknown complex environment with collision avoidance [12], [13]. However, these methods require large amount of computation and memory resource for processes such as feature extraction, mapping and optimization. Hence SLAM-based methods are unsuitable for applications with limited computing resources.

To reduce the computation, some other monocular vision cues, e.g. optical flow [14] and expansion rate [15], have been used to get a perception of the environment without mapping process. Hector *et al* [16] proposed a robust low size, weight and power system that can navigate quickly in a complex unknown cluttered environment without GPS. The core perception algorithm uses the concept of expansion rate which is calculated using optical flow. Although it achieves very fast flight speeds of up to 19 m/s in the real-world environment, the calculation of dense optical flow is too computationally expensive for some micro UAVs. In addition, some researchers achieve depth estimation from single image using deep convolutional neural networks (CNN) [17], [18] and apply it for obstacle avoidance [19], [20].

In this work, we focus on the problem of monocular vision-based autonomous navigation and obstacle avoidance system for micro UAVs which have a very low SWaP capacity. More specifically, the proposed method consists of a bio-inspired obstacle detection network for perception and a navigation network trained by deep reinforcement learning. The framework of the whole system is shown in Figs 1 and 2. It will be described in more detail the following section.

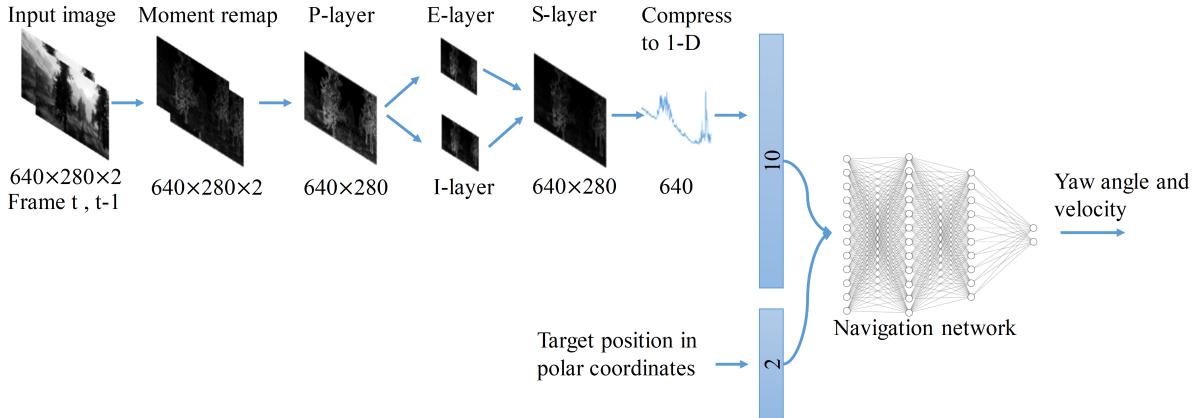


Fig. 2: Framework of image-moment based LGMD perception and navigation system.

II. BACKGROUND

The proposed perception network is inspired by insect visual systems. To improve the reliability and robustness for perception, the grayscale input image is remapped to a moment scale image which is independent of the illumination variation. In addition, a reactive local planner is trained using deep reinforcement learning (DRL).

A. The Lobula Giant Moment Detector (LGMD)

The LGMD is a looming-object detector discovered in the locust brain which responds most strongly to approaching objects such as predators [21]. A computational model of the LGMD was pioneered by Rind *et al* [22] and has continuously evolved over the years. The model is composed of five groups of cells: P-cells (photoreceptor), I-cells (inhibitory), E-cells (excitatory), S-cells (summing) and G-cells (grouping) along with two individual cells, namely, the Feed-Forward Inhibition (FFI) and LGMD. See [21] for a detailed description of LGMD and other insect visual systems.

So far, LGMD is mainly used for the small indoor scenarios. Yue *et al* [23] proposed an LGMD-based neural network with a feature enhancement mechanism to detect the expanded edges of the colliding object and tested it by real-time robotics experiments. Meng *et al* [24] introduced a modified neural model for LGMD that provides additional depth direction information for the movement. The method has been implemented on a Field Programmable Gate Array (FPGA) and tested on real-time video streams which can achieve 220fps at frame size 300×200 and 55fps at 600×400 . Inspired by LGMD, Hamid *et al* [25] achieved a secondary reactive collision avoidance system for micro-robots based on Furcate Luminance-Difference Processing (FLDP) and tested this algorithm using an RC car equipped with a Raspberry Pi. In [26] the authors used LGMD for obstacle avoidance on a UAV which is similar to our work. However, they used an event-based camera as the vision sensor and only show the result of obstacle detection in indoor environment. Our method extends the original LGMD model proposed in [23] to outdoor application and improves

this method by adding image moment to get robustness perception against illuminance variation.

B. Image moment

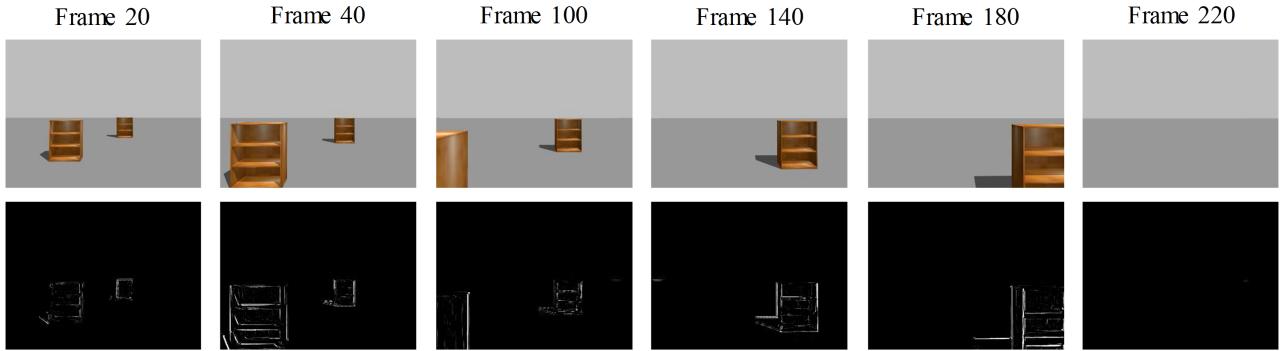
Image moment is a major approach in pattern recognition and has been an active research field in image processing since 1962 [27]. For micro UAVs operating in complex outdoor environment, illumination variation is very common. Recently, Kharbat *et al* [28] described the grey-level of each pixel by the content of its neighbourhood using a geometric moment rather than its intensity function value. They applied this method to optical flow estimation and improved the reliability and robustness against varying illuminations.

C. Deep Deterministic Policy Gradient (DDPG)

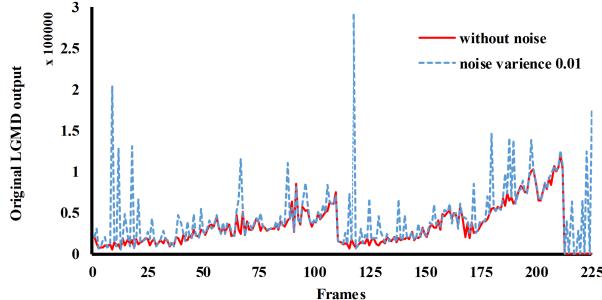
Benefiting from the advance of artificial neural networks, end-to-end learning approaches have been proposed to directly train networks from raw images for robot navigation, e.g. supervised learning [8] and imitation learning [29]. After the success of Deep Q-Networks (DQN) [30], many researchers have tried to solve the obstacle avoidance and navigation problem with learning methods. Xie *et al* [31] applied a duelling architecture-based Deep Double-Q Network (D3QN) for obstacle avoidance using only monocular RGB vision. They use depth information predicted from the RGB image as the perception. Zhang *et al* [32] provide a solution for robot navigation based on depth image trained with DQN, where successor features are used to transfer the strategy to an unknown environment.

To extend DQN to the continuous domain, Lillicrap *et al* proposed DDPG [33] algorithm for continuous control problems. DDPG is an actor-critic learning algorithm. It uses off-policy data and the Bellman equation to learn the Q-function and uses the Q-function to learn the policy. Such an actor-critic architecture makes it suitable to work in a continuous action domain which is difficult for DQN and it is beneficial for robotics. The training for the critic network is almost the same as DQN, but the actor network is updated with the policy gradient.

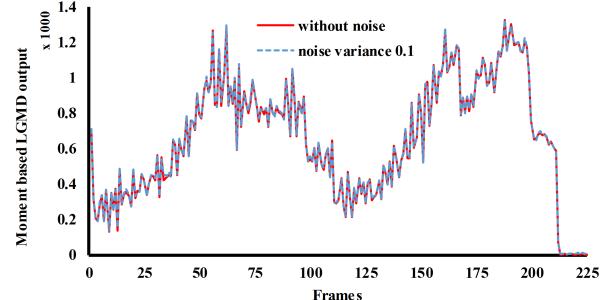
Based on DDPG, Tai *et al* [34] proposed a mapless motion planner by taking sparse 10-dimensional range findings and



(a) Raw input image (upper row) and LGMD output image (lower row) extracted from the video recorded in Gazebo. The robot is moving at a linear velocity of 2m/s towards two static obstacles.



(b) Original LGMD output.



(c) Moment-based LGMD output.

Fig. 3: Comparison of the original LGMD and the proposed method against illuminance variation.

the target position to the mobile robot as the input and trained end-to-end without any manually designed features and prior demonstrations. The result shows that the trained planner can be directly applied in unseen virtual and real environments.

III. ROBUST LOCAL PERCEPTION

Illumination variation is very common among images obtained by UAVs because of different and changing operating environments and the automatic exposure mechanism of the camera. The original LGMD method relies on the illuminance difference of consecutive image frames which is strongly affected by the frequent illumination variation. To tackle this problem, image moment is introduced. The main idea of the proposed method is to remap the input image of LGMD with normalized local geometric moments which are independent from the illumination variation, rather than using the greyscale image directly.

A. Image remapping using local image moment

For a 2D continuous function $f(x, y)$, the moment of order $(p + q)$ is defined as:

$$M_{pq} = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} x^p y^q f(x, y) dx dy \quad (1)$$

where $p, q = 0, 1, 2, \dots$. Adapting this to a greyscale image with pixel intensities $I(x, y)$, the raw image moments M_{ij} can be calculated by

$$M_{ij} = \sum_x \sum_y x^i y^j I(x, y) \quad (2)$$

Equation (2) defines the geometric moment for the whole image. It is also available to calculate the same moments over a small window around each pixel. These descriptions are associated with image features in a pixel neighbourhood known as local moments. For a region R of size $(2N + 1, 2N + 1)$ at pixel (x, y) , the corresponding $(p + q)^{\text{th}}$ order local moment is defined as:

$$m_{pq}(x, y) = \sum_{i=-N}^N \sum_{j=-N}^N i^p j^q I(x - i, y - j) \quad (3)$$

When the image region R suffers from illuminance variation, the intensity function of the new image $I'(x, y)$ can be approximated equal to the initial intensity function $I(x, y)$ multiplied by a factor c :

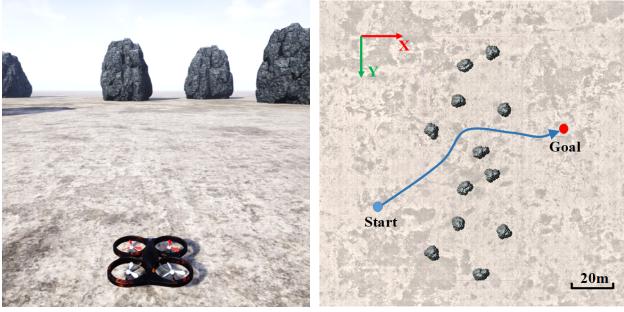
$$I'(x, y) \approx c \cdot I(x, y) \quad (4)$$

When affected by the change of intensity, the new local moment changes by the same factor c according to (3):

$$m'_{pq} \approx c \cdot m_{pq} \quad (5)$$

The local moments description of the image can be independent of the factor c divided by any non-zero moment. In our work, the input grayscale image is remapped using the sum of two first-order local moments m_{01} and m_{10} , and divided by its zero-order local moment m_{00} :

$$M^{\text{norm}}(x, y) = \frac{m_{01}(x, y) + m_{10}(x, y)}{m_{00}(x, y)} \quad (6)$$



(a) Training environment. (b) Environment top view.
Fig. 4: The training environment.

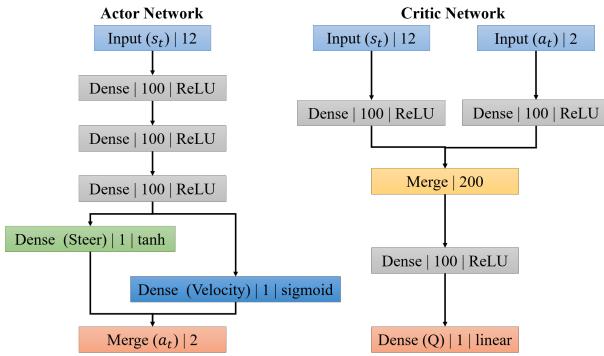


Fig. 5: The network structure for DDPG network.

From (6) we can see that the illuminance variation factor c is eliminated in the expression for the normalized local moment $M_f^{\text{norm}}(x, y)$. Although this transform slightly increases the computation load, it provides a better and more robust perception in complex environments with illuminance variation which is very important for outdoor flight.

B. LGMD perception

After image remapping with the local moment, a computation model of LGMD is utilized to get perception of the environment. Different from the original LGMD which can only get obstacle information, we split the image Field of View (FoV) vertically to get obstacle information in different directions.

1) P-layer: The first layer of the LGMD network consists of the photoreceptor P cells arranged in a matrix. The normalized local moment $M_f^{\text{norm}}(x, y)$ of each pixel (x, y) in the input image is captured by each photoreceptor cell. Then, the absolute difference of P_f between frames of the image sequence is calculated. The output of the P-layer is defined by:

$$P_f(x, y) = \sum_{i=1}^{n_p} p_i P_{f-i} + (M_f^{\text{norm}}(x, y) - M_{f-1}^{\text{norm}}(x, y)) \quad (7)$$

where $P_f(x, y)$ is the change of normalized local moment corresponding to pixel (x, y) at frame f , x and y are indices of the image, n_p defines the maximum number of frames the persistence can last, and $p_i \in (0, 1)$ is the persistence

coefficient. In order to reduce the computation, we set $n_p = 0$.

2) IE-layer: The output of the P-layer forms the input to two parallel layers of separate cell types, I (inhibitory) cells and E (excitatory) cells. The E-layer passes the P-layer output directly to the S-layer, while the I-layer pass the inhibitory flow convolved by the surrounding delayed excitation:

$$E_f(x, y) = P_f(x, y) \quad (8)$$

$$I_f(x, y) = \sum_i \sum_j P_{f-1}(x+i, y+j) \cdot W_I(i, j) \quad (9)$$

where $W_I(x, y)$ is the local inhibition weight. The I-layer can also be treated as a simplified convolution operation:

$$I_f = P_f \otimes W_I \quad (10)$$

where W_I is the convolution mask representing the local inhibition weight distribution:

$$W_I = \begin{bmatrix} 0.125 & 0.25 & 0.125 \\ 0.25 & 0 & 0.25 \\ 0.125 & 0.25 & 0.125 \end{bmatrix} \quad (11)$$

3) S-layer: The next layer is the Sum layer (S-layer) which combines the outputs of the E-layer and I-layer by

$$S_f(x, y) = E_f(x, y) - I_f(x, y) \cdot w_i \quad (12)$$

where w_i denotes the inhibition coefficient, that is 0.7 in our case. To ensure the output of S-layer is positive for every pixel, an additional condition is included:

$$S_f(x, y) = \begin{cases} S_f(x, y), & \text{if } S_f(x, y) = 0 \\ 0, & \text{otherwise} \end{cases} \quad (13)$$

4) LGMD cell: The LGMD model can only detect whether or not an object in the field of view is looming. To get perception of surrounding environment, the output of S-layer is split into N_{LGMD} parts vertically and implement an LGMD cell for each section. For the i th part of S-layer output, the membrane potential of the LGMD cell is defined as the sum of the relevant S-layer part:

$$K_f^i = \sum_x \sum_y S_f^i(x, y) \quad (14)$$

The membrane potential of the LGMD cell is then transformed into a sigmoid function:

$$k_f^i = (1 + e^{-K_f^i / N_{\text{cell}}})^{-1} \quad (15)$$

where N_{cell} is the activate coefficient. Since K_f^i is greater than zero, the output of each LGMD cell $k_f^i \in (0.5, 1)$. The final output of the perception model is a 1D vector with N_{LGMD} elements. The higher the magnitude, the closer an obstacle is relative to the vehicle in that specific azimuth angle. In our experiment, the input image is 640×280 resolution with 90° FoV, $N_{\text{LGMD}} = 10$ and $N_{\text{cell}} = 100$.

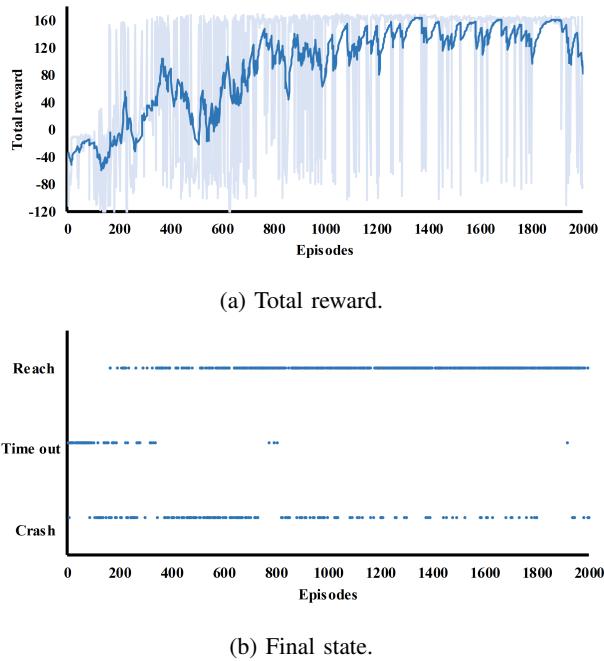


Fig. 6: Training result of each episode.

C. Validation against illuminance variation

To illustrate the effect of the image moment, both the original LGMD algorithm and the proposed moment-based LGMD algorithm are executed with video as shown in Fig. 3(a). A noise factor c is utilized to simulate the illuminance variation where $c \sim N(0, 0.01)$ for the original LGMD and $c \sim N(0, 0.1)$ for the moment-based LGMD because the former is more sensitive to illuminance variation. The results shown in Fig. 3(b) and (c) indicate that both methods can detect obstacles in the video without noise. However, with added Gaussian noise, the output of the original LGMD algorithm becomes noisy and is hard to detect obstacles. In contrast, the moment-based LGMD performs well because it is independent of the illuminance variation.

IV. LEARNING-BASED REACTIVE PLANNER

Unlike classical planning methods, reactive control approaches have deterministic and short computation times and can cope with high frequency dynamics and unpredictable environments. It computes just one next action at every sampling instant based on the current state – this is very computationally efficient and suitable for micro UAVs operating in complex environments.

To investigate the feasibility of the proposed perception method, a reactive control navigation network is trained in a simulation environment based on Microsoft AirSim [35] as shown in Fig. 4. AirSim is based on Unreal Engine which can provide realistic visual scenes. This is beneficial for transferring the trained network from simulation to the real world. The input of the navigation network consists of two parts as shown in Fig. 2, one is the perception using moment-based LGMD, the other is the relative position from the

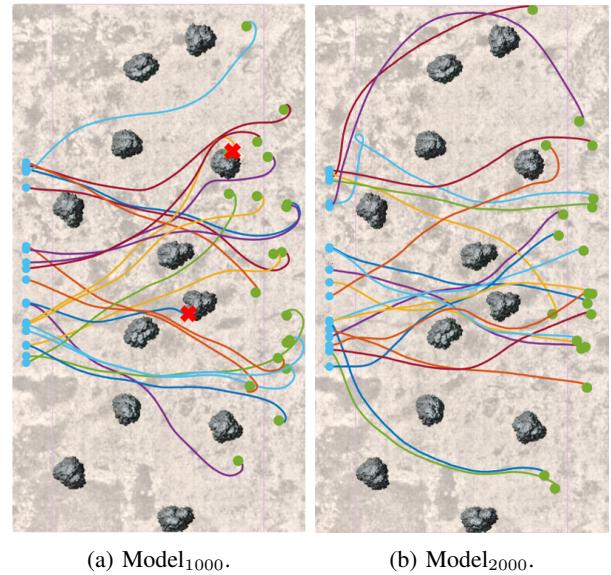


Fig. 7: Trajectories of evaluation in training environment. Start points (blue) and target points (green) are generated randomly for each episode. The red cross means collision.

UAV to the target point represented in polar coordinates. At the beginning of each episode, a simulated multirotor takes off from a random start point and the objective is to move towards the target position without collision.

A. Reward function

The reward function is crucial for deep reinforcement learning. To simplify the training process, a hand-designed reward function consists of two sparse parts and a dense part is utilized for training:

$$r_t = \begin{cases} R_{\text{reach}}, & \text{if goal reached} \\ R_{\text{crash}}, & \text{if collision} \\ R_{\text{positive}} - R_{\text{negative}}, & \text{otherwise} \end{cases} \quad (16)$$

where

$$R_{\text{positive}} = d_{t-1} - d_t \quad (17)$$

$$R_{\text{negative}} = \left[w_1 \left(\frac{a_t^1}{a_{\max}^1} \right) + w_2 |e_{\text{angle}}| \right] \Delta t + C \quad (18)$$

and where $R_{\text{reach}} = 100$ and $R_{\text{crash}} = -100$ means that the multirotor obtains a large positive reward for reaching the goal and a large negative reward for colliding with an obstacle, d_{t-1} and d_t indicate the distance from current position to the target at two consecutive sample time instants, a_t^1 is the first output of the navigation network which means the expected yaw for steering control, a_{\max}^1 is the maximum yaw output which is set to 30° , e_{angle} is the error between current yaw angle and the target azimuth, Δt is the time interval between two consecutive frames, w_1 and w_2 are penalty weight terms which are set to 0.5 and 1 respectively, and C is a constant penalty on the flight time.



(a) Env1.

(b) Env2.

(c) Env3.

Fig. 8: Environments for model evaluation.

TABLE I: Results of model evaluation

Environment	Description	Complexity	Success ratio	
			Model ₁₀₀₀	Model ₂₀₀₀
Training environment	Random distribution of rocks	simple	90%	100%
Env1	Random distribution of pine trees	simple	100%	75%
Env2	Random distribution of pine trees and snow trees	medium	90%	70%
Env3	Landscape Mountains environment with complex background	complex	80%	60%

B. RL training

The navigation network is trained in the environment shown as in Fig. 4. To increase the stochastic, the start and target points are generated randomly at the beginning of each episode. The start point is set to $(0, y)$, where y is randomly selected between -20m and 20m. The angle between the start-target line and x-axis is chosen randomly between -30° and 30° . The distance from the start point to target point is fixed to 100m so that a maximum $R_{\text{positive}} = 100$ can be obtained if the multirotor reaches the target point. The structure of the DDPG network is shown in Fig. 5 where s_t is the observation consists of the LGMD perception and relative target position represented in polar coordinate. a_t is the network output which includes the expected yaw angle $a^1 \in (-30^\circ, 30^\circ)$ and forward velocity $a^2 \in (2, 6)$ m/s. The acceptable radius of the target point is set to 2m.

For the hyperparameters, the Adam optimizer is used for learning for both actor and critic network with a learning rate of 10^{-4} and 5×10^{-4} respectively. The reward discount factor is $\gamma = 0.98$ and $\tau = 10^{-3}$ for the soft update. The system was trained with mini-batch sizes of 64 and a replay buffer size of 4×10^{-5} . The Ornstein-Uhlenbeck process for the expected yaw angle a^1 is $\theta = 0.15$ and $\sigma = 0.1$ and for forward velocity a^2 is $\theta = 0.05$ and $\sigma = 0.4$.

After training for 2000 episodes, the result is shown in Fig. 6. The total reward converged after 1000 episodes and the multirotor can reach the target point with a high success rate. Because the Ornstein-Uhlenbeck process is used to keep exploring the environment during training, the multirotor still sometimes collides with the obstacles.

C. Evaluation

To evaluate the training result and test the robustness of proposed method, the trained model is tested in the training environment as shown in Fig. 4 as well as three new environments as shown in Fig. 8. Two different models

saved at different training periods are used for evaluation, Model₁₀₀₀ is saved after training for 1000 episodes and Model₂₀₀₀ after 2000 episode. Both models are executed 20 episodes in every environment. The description of each environment and the test result is listed in Table I.

The result shows that Model₂₀₀₀ performs better than Model₁₀₀₀ in training environment. However, in other evaluation environments, Model₁₀₀₀ performs better. Model₂₀₀₀ maybe be over-fitted to the training environment. In addition, Fig. 7 shows the trajectory of each model in the training environment. It shows that Model₂₀₀₀ can reach the target point faster than Model₁₀₀₀ while Model₁₀₀₀ has some overshoot near the target point. Because the task includes two objectives, obstacle avoidance and navigation, there is a conflict in these two objectives and it needs a good reward function to get a good trade-off between them. In our reward function configuration, the multirotor learns to avoid obstacles faster than to navigate to the target point.

V. CONCLUSIONS

In this work, an obstacle avoidance system inspired by insect visual system with a learning-based reactive local planner for mapless navigation is proposed and evaluated in several simulation environments. The system is lightweight which is suitable for micro UAVs with little SWaP constraint. Our work indicates that bio-inspired perception algorithms such as LGMD can be applied to the complex outdoor environment with illuminance variation. Moreover, deep reinforcement learning can be applied to learn a navigation policy without accurate depth information.

In the future, a micro UAV platform equipped with only monocular camera and limited computation resources will be built to evaluate proposed method in a real environment. Also, we will keep studying for a good reward function design method to balance the weight of obstacle avoidance and navigation and prevent over-fitting during training.

REFERENCES

- [1] M. Radmanesh, M. Kumar, P. H. Guentert, and M. Sarim, "Overview of path-planning and obstacle avoidance algorithms for UAVs: a comparative study," *Unmanned Systems*, vol. 6, no. 02, pp. 95–118, 2018.
- [2] N. Gageik, P. Benz, and S. Montenegro, "Obstacle detection and collision avoidance for a UAV with complementary low-cost sensors," *IEEE Access*, vol. 3, pp. 599–609, 2015.
- [3] M. Odelga, P. Stegagno, and H. H. Bülfhoff, "Obstacle detection, tracking and avoidance for a teleoperated UAV," in *IEEE International Conference on Robotics and Automation (ICRA)*, Stockholm, Sweden, 2016, pp. 2984–2990.
- [4] M. Iacono and A. Sgorbissa, "Path following and obstacle avoidance for an autonomous UAV using a depth camera," *Robotics and Autonomous Systems*, vol. 106, pp. 38–46, 2018.
- [5] S. Ramasamy, R. Sabatini, A. Gardi, and J. Liu, "LIDAR obstacle warning and avoidance system for unmanned aerial vehicle sense-and-avoid," *Aerospace Science and Technology*, vol. 55, pp. 344–358, 2016.
- [6] A. J. Barry and R. Tedrake, "Pushbroom stereo for high-speed navigation in cluttered environments," in *IEEE International Conference on Robotics and Automation (ICRA)*, Seattle, WA, 2015, pp. 3046–3052.
- [7] Y. Lyu, Q. Pan, C. Zhao, and J. Hu, "Autonomous stereo vision based collision avoid system for small UAV," in *AIAA Information Systems-AIAA Infotech@Aerospace*, Grapevine, TX, 2017, p. 1150.
- [8] J. Michels, A. Saxena, and A. Y. Ng, "High speed obstacle avoidance using monocular vision and reinforcement learning," in *Proceedings of the 22nd International Conference on Machine Learning (ICML '05)*, Bonn, Germany, 2005, pp. 593–600.
- [9] R. Mur-Artal, J. M. M. Montiel, and J. D. Tardos, "ORB-SLAM a versatile and accurate monocular SLAM system," *IEEE Transactions on Robotics*, vol. 31, no. 5, pp. 1147–1163, 2015.
- [10] S. Shen, N. Michael, and V. Kumar, "Tightly-coupled monocular visual-inertial fusion for autonomous flight of rotorcraft MAVs," in *IEEE International Conference on Robotics and Automation (ICRA)*, Seattle, WA, 2015, pp. 5303–5310.
- [11] T. Qin, P. Li, and S. Shen, "VINS-Mono: A robust and versatile monocular visual-inertial state estimator," *IEEE Transactions on Robotics*, vol. 34, no. 4, pp. 1004–1020, 2018.
- [12] O. Esrafilian and H. D. Taghirad, "Autonomous flight and obstacle avoidance of a quadrotor by monocular SLAM," in *4th International Conference on Robotics and Mechatronics (ICROM)*, Tehran, Iran, 2016, pp. 240–245.
- [13] Y. Lin, F. Gao, T. Qin, W. Gao, T. Liu, W. Wu, Z. Yang, and S. Shen, "Autonomous aerial navigation using monocular visual-inertial fusion," *Journal of Field Robotics*, vol. 35, no. 1, pp. 23–51, 2018.
- [14] X.-Z. Peng, H.-Y. Lin, and J.-M. Dai, "Path planning and obstacle avoidance for vision guided quadrotor UAV navigation," in *12th IEEE International Conference on Control and Automation (ICCA)*, Kathmandu, Nepal, 2016, pp. 984–989.
- [15] A. Al-Kaff, F. García, D. Martín, A. De La Escalera, and J. Armingol, "Obstacle detection and avoidance system based on monocular camera and size expansion algorithm for UAVs," *Sensors*, vol. 17, no. 5, p. 1061, 2017.
- [16] H. D. Escobar-Alvarez, N. Johnson, T. Hebble, K. Klingebiel, S. A. Quintero, J. Regenstein, and N. A. Browning, "R-ADVANCE: Rapid adaptive prediction for vision-based autonomous navigation, control, and evasion," *Journal of Field Robotics*, vol. 35, no. 1, pp. 91–100, 2018.
- [17] F. Liu, C. Shen, and G. Lin, "Deep convolutional neural fields for depth estimation from a single image," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Boston, MA, June 2015.
- [18] C. Godard, O. Mac Aodha, and G. J. Brostow, "Unsupervised monocular depth estimation with left-right consistency," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Honolulu, HI, 2017, pp. 270–279.
- [19] P. Chakravarty, K. Kelchtermans, T. Roussel, S. Wellens, T. Tuytelaars, and L. Van Eycken, "CNN-based single image obstacle avoidance on a quadrotor," in *IEEE International Conference on Robotics and Automation (ICRA)*, Singapore, 2017, pp. 6369–6374.
- [20] S. Dionisio-Ortega, L. O. Rojas-Perez, J. Martinez-Carranza, and I. Cruz-Vega, "A deep learning approach towards autonomous flight in forest environments," in *2018 International Conference on Electronics, Communications and Computers (CONIELECOMP)*, Cholula Puebla, Mexico, 2018, pp. 139–144.
- [21] Q. Fu, H. Wang, C. Hu, and S. Yue, "Towards computational models and applications of insect visual systems for motion perception: A review," *Artificial Life*, vol. 25, no. 3, pp. 263–311, 2019.
- [22] 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, no. 3, pp. 967–985, 1996.
- [23] S. Yue and F. C. Rind, "Collision detection in complex dynamic scenes using an LGMD-based visual neural network with feature enhancement," *IEEE Transactions on Neural Networks*, vol. 17, no. 3, pp. 705–716, 2006.
- [24] H. Meng, K. Appiah, S. Yue, A. Hunter, M. Hobden, N. Priestley, P. Hobden, and C. Pettit, "A modified model for the Lobula Giant Movement Detector and its FPGA implementation," *Computer Vision and Image Understanding*, vol. 114, no. 11, pp. 1238–1247, 2010.
- [25] H. Isakhani, N. Aouf, O. Kechagias-Stamatis, and J. F. Whidborne, "A furcated visual collision avoidance system for an autonomous micro robot," *IEEE Transactions on Cognitive and Developmental Systems*, 2018, early Access.
- [26] L. Salt, G. Indiveri, and Y. Sandamirskaya, "Obstacle avoidance with LGMD neuron: towards a neuromorphic UAV implementation," in *IEEE International Symposium on Circuits and Systems (ISCAS)*, Baltimore, MD, 2017, pp. 1–4.
- [27] M.-K. Hu, "Visual pattern recognition by moment invariants," *IRE Transactions on Information Theory*, vol. 8, no. 2, pp. 179–187, 1962.
- [28] M. Kharbat, N. Aouf, A. Tsourdos, and B. A. White, "Robust brightness description for computing optical flow," in *BMVC*, Leeds, U.K., 2008, pp. 1–10.
- [29] S. Ross, N. Melik-Barkhudarov, K. S. Shankar, A. Wendel, D. Dey, J. A. Bagnell, and M. Hebert, "Learning monocular reactive UAV control in cluttered natural environments," in *2013 IEEE International Conference on Robotics and Automation (ICRA)*, Karlsruhe, Germany, 2013, pp. 1765–1772.
- [30] V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski, et al., "Human-level control through deep reinforcement learning," *Nature*, vol. 518, no. 7540, p. 529, 2015.
- [31] L. Xie, S. Wang, A. Markham, and N. Trigoni, "Towards monocular vision based obstacle avoidance through deep reinforcement learning," *arXiv preprint arXiv:1706.09829*, 2017.
- [32] J. Zhang, J. T. Springenberg, J. Boedecker, and W. Burgard, "Deep reinforcement learning with successor features for navigation across similar environments," in *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Vancouver, BC, Canada, 2017, pp. 2371–2378.
- [33] T. P. Lillicrap, J. J. Hunt, A. Pritzel, N. Heess, T. Erez, Y. Tassa, D. Silver, and D. Wierstra, "Continuous control with deep reinforcement learning," *arXiv preprint arXiv:1509.02971*, 2015.
- [34] L. Tai, G. Paolo, and M. Liu, "Virtual-to-real deep reinforcement learning: Continuous control of mobile robots for mapless navigation," in *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Vancouver, BC, Canada, 2017, pp. 31–36.
- [35] S. Shah, D. Dey, C. Lovett, and A. Kapoor, "Airsim: High-fidelity visual and physical simulation for autonomous vehicles," in *Field and Service Robotics – Results of the 11th International Conference*. Springer, 2018, pp. 621–635.