Monocular Depth Estimation for UAV Obstacle Avoidance

Zhenghong Zhang, Mingkang Xiong, Huilin Xiong
School of Electronic Information and Electrical Engineering
Shanghai Jiao Tong University
Shanghai, China
e-mail: {art_zzh, mkxiong, hlxiong}@sjtu.edu.cn

***Abstract*—In this paper, we present a novel method for obstacle avoidance of a quadrotor equiped with a single front camera. The proposed method is composed of three parts: depth estimation, obstacle detection, and obstacle avoidance controll. We use convolutional neural networks (CNN) to estimate depth from RGB image. Then the depth image is fed into the obstacle avoidance system, in which proposed control algorithm steers the quadrotor to fly away from obstacles, and after that, continue towards the destination. We conduct a lot of experiments, either in virtual environment with a simulated drone, or in real world with a quadrotor Parrot Bebop2, to verify the effectiveness of our method.**

***Index Terms*—quadrotor; monocular; obstacles avoidance algorithm; CNN; depth estimation**

I. Introduction

A large number of cheaper, lightweight quadrotors are available on the market, which are convenient to control through Wi-Fi by smartphone or laptop. Usually, the small quadrotor is only equiped with a single front camera as the primary sensing facility. Such types of small quadrotors as DJI Phantom and Parrot bebop, have gained enormous popularity over past years because of its suitability for various applications, e.g. package delivery, filming and visual surveillance. However, the risk of a small quadrotor crashing obstacles is pretty high, especially in cluttered unknown environments. Therefore, perceiving and avoiding obstacles are the necessary abilities for a small UAV, and furthmore, are also highly challenging tasks, espacially for a quadrotor only depending on a monocular camera to sense and to navigate in cluttered unknown environments.

We use a deep fully fonvolutional neural network (MegaDepth) [1] to predict the depth images of the given RGB images. MegaDepth [1] generates a set of 3D reconstructions using modern structure-from-motion(SfM) and multi-view stereo (MVS) methods. It can adapt to a variety of scenarios. An obstacles avoidance algorithm is developed in this paper, which use the estimated depth images to steer the quadrotor to fly away from obstacles and continue to fly towards the destination. Experiments are carried out in both virtual and real world environments to demonstrate the effectiveness of the proposed method.

The rest of the paper is organized as follows. Section II describes Previous work. Section III describes the proposed method for Obstacle avoidance. Section IV provides experimental results of the simulated and real drone flights for obstacle avoidance. Section V presents the conclusion remarks.

II. Related work

A. Traditional Sensors for Obstacle Avoidance

Obstacle avoidance is an important part of the autonomous navigation system. Perceiving obstacles is one of the key technologies for obstacle avoidance. According to the way of perception of obstacles, there are two kinds of sensors. One is non-vision sensors like LIDAR, sonar, structured light and IR [8]. The other is vision sensors such as binocular cameras sensing stereo vision information [4], [9].

Structured light sensors and LIDAR are not suitable for small flying platforms, in terms of weight and power requirements. The depth sensing of Sonar and IR is limited to a short range (within one meter). So they are also not suitable for fast, reactive flying. Depth images from binocular cameras with more visual information, can be used for mapping and path planning. But binocular cameras possible on a quadrotor, with a small baseline, has the same problem of limited range.

B. Monocular Vision-Based Obstacle Detection

Bills et al. [10] proposed a obstacle detection method in the indoor environments with single image perspective cues. This work uses a trained classifier to detect the type of environments, and then, extracts different perspective cues for computing the desired direction to fly. In [11], multi-scale-oriented-patches (MOPS) and scale-invariant feature transform (SIFT) descriptor are combined to estimate 3D information of the obstacles. MOPS is used to detect the edges and corners of the object, and SIFT is used to extract the internal outline information. In order to improve the speed of computation, another approach [12] used template matching and speeded-up robust features (SURF) matching to compare the relative size of obstacles under different image spacing. In which, SURF is faster than SIFT. In [13], the authors presented a mathematical model to estimate the relative distance from the camera. In which, position of the camera is estimated based on the data from the Inertial Measurement Unit (IMU) using the Extended Kalman Filter (EKF). Then the 3D position of the obstacle is estimated from its 2D matched SURF feature points. More recently, there has been some work that uses a deep learning approach to solve the problem of depth estimation from a single image [15], [16], [17]. Therefore, the depth estimation CNN was increasingly used for obstacle avoidance on the quadrotors [2].

III. Method

A. System

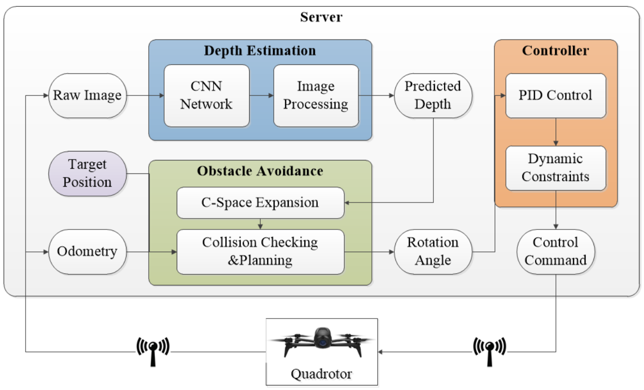


Fig. 1. Obstacle avoidance system architecture.

As shown in Fig.1, our obstacle avoidance system consists of three major modules: depth estimation, obstacle avoidance and controller. The depth estimation module contains two components: 1) a CNN model (MegaDepth [1]) which continuously estimates unreliable depth from RGB images; 2) the image processing part which makes the depth to be more reliable. The obstacle avoidance module is composed of two parts: 1) a C-space expansion operation [4] on depth images based on the quadrotor size to configure the drone as point; 2) collision checking and planning section, which outputs rotation angle from the expanded depth images. The controller module receives the signal of the rotation angle, and then, generates the angular velocity using a PID controller. The angular velocity is limited to a certain range under dynamic constraints. Then the angular velocity is published to the quadrotor.

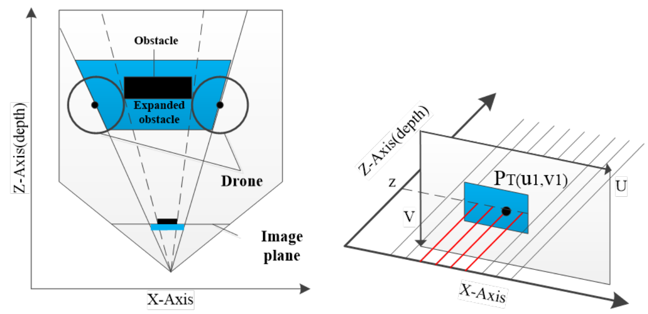
B. Monocular Depth Estimation and Image Processing

To obtain the depth information, We adopt the CNN used in MegaDepth [1]. There are three contributions in [1], namely, an original loss function and learning strategy, data augmentation techniques, and a large number of training datasets established. It [1] generates a set of 3D reconstructions using SfM and MVS from internet images. These reconstructions are used as ground truth normalized depth images for model training. Therefore, we use the network in [1] for depth estimation.

However, the CNN [1] that takes a large scale image (640×480) as input can not satisfy the real-time need in the obstacles avoidance application. Considering this, the input size for CNN is smaller in our system, which is 320×240. The side effect of contraction is that depth information is unreliable. Therefore, we propose to use Mean Filtering and Histogram Equalizaton to make depth imformation more precise, shown in Fig.2.

(a) Raw images. (b) Depth. (c) Expanded depth.

Fig. 2. Depth iamges generated by the depth estimation module.



(a) (b)

Fig. 3. (a) C-space expansion in the world. (b) Obstacle checking.

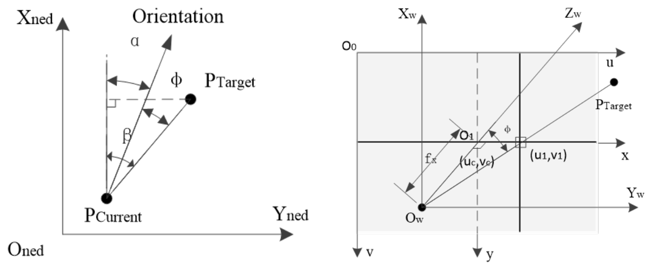
Then, C-space expansion [4] applied in the depth images means to expand the obstacle in the world. As shown in Fig.3 (a), if we use the original depth, the size of the drone should be considered in collision checking. When using the expanded depth, it allows to treat the drone as a single point in the planning. It is very convenient for collision checking, which depends on depth value z of a point $P_T(u_1, v_1)$ on the expanded depth, shown in Fig.3 (b). If the depth value of $P_T(u_1, v_1)$ is larger than the safe value d_{safe}, the point P_T is considered as safe. If the depth value of $P_T(u_1, v_1)$ is smaller than the safe value d_{safe}, the point P_T is considered as obstacles. The depth estimation module runs on a GPU at 10 fps.

C. Depth Image Based Obstacle Avoidance Algorithm

Our obstacle avoidance algorithm is shown in Algorithm 1. It takes the expanded depth, a goal position, the current position and the current orientation as input and outputs the rotation angle for obstacles avoidance. Note that, all calculations in this algorithm are based on North-East-Down (NED) coordinate system. The function $IncludedAngle()$ calculates the included angle ϕ between the current orientation and the direction of destination P_{Target}, as shown in Fig.4 (a). The function $CoordinateMapping()$ calculates the distance of the point P_{Target} in the world coordinate sysytem based on the depth image D and the included angle ϕ, as shown in Fig.4 (b). The calculation process is expressed in Equations (1), (2).

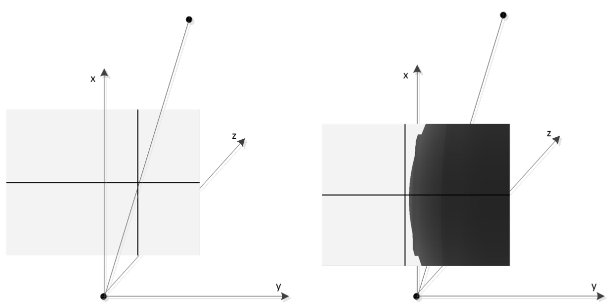
$$u_1 = tan(\phi) \times f_x + u_c \tag{1}$$

$$x = D_e[u_1, v_1] \tag{2}$$



(a) (b)

Fig. 4. Left: Shows the included angle ϕ between the current orientation and the direction of destination. The arrow represent orientation, P_C represent a current position, and P_T represent a target position. Right: Shows mapping relations between pixel coordinates and the included angle.



(a) No obstacles. (b) Find obstacles.

Fig. 5. Results of obstacle avoidance algorithm. The cross point represents the target direction.

Under the control of the algorithm 1, the quadrotor always flies towards the destination. Only when detecting obstacles, does the quadrotor rotate for avoiding. The loop part of the algorithm 1 is a search strategy to find the smallest rotation angle avoiding obstacles.

Fig.5 shows our obstacle avoidance algorithm is effective. The drone rotates right towards a target position when no obstacles are detected in Fig.5 (a). But the drone rotates left for avoiding obstacles when obstacles are detected in Fig.5 (b).

Algorithm 1: Obstacle Avoidance Algorithm

Input : Expanded depth image D_e, Target position $P_T(x, y)$, Current position $P_C(x, y)$, Current orientation α;

Output: rotation angle ϕ;

Initial: Safe distance d_{safe}, Search step size k, Field of Vision Ω
$\phi = IncludedAngle(P_T, P_C, \alpha)$;
$z = CoordinateMapping(D_e, \phi)$;
if $z < d_{safe}$ **then**
    $(tmp, step, left, right) \leftarrow (\phi, 1, true, true)$;
    while $left$ || $right$ **do**
        if $left$ **then**
            $\phi = tmp - step * k$;
            if $\phi < -\Omega/2$ **then**
                $left = false$;
            else
                $z = CoordinateMapping(D_e, \phi)$;
                if $z > d_{safe}$ **then**
                    break;
                end
            end
        end
        if $right$ **then**
            $\phi = tmp + step * k$;
            if $\phi > \Omega/2$ **then**
                $right = false$;
            else
                $z = CoordinateMapping(D_e, \phi)$;
                if $z > d_{safe}$ **then**
                    break;
                end
            end
        end
        $step = step + 1$;
    end
end

IV. Experiments

A. Performance of the Drone Obstacles Avoidance in Virtual Environments

We use Microsoft's AirSim [14] as a virtual environment which is closer to the real world, Fig.6. In the experiments, the drone takes off at one end of the trees, and arrives at the other end. Its current state is obtained from the virtual environment. When the drone reaches the destination, it is considered as a successful flight. When the drone collides with a tree, it is considered as a failed flight. To confirm the algorithm effectiveness, four trials were run for each of the three destinations, and for three different speeds 1, 2, 3 meters per seconds. The success rate of reaching the destinations is presented in TABLE I. And the flight track is shown in Fig.7. From the results, we can see that the drone was able to fly through the trees successfully at a low speed. When enhancing the speed of the drone, it crashed into a tree at few times. These failures are excusable because of low frame rate of depth estimation module. Generally, the control algorithm successfully steers the drone to fly away from obstacles and continue to fly towards the goal position.

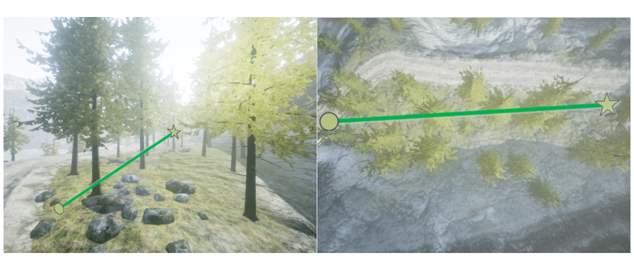


Fig. 6. Side view and top view of virtual enviroments. Circle represents the starting point and pentagram represents the goal position.

TABLE I
PERFORMANCE OF OUR METHOD IN THE SIMULATED WORLD.

Speed(m/s)	*1*	*2*	*3*
Desitnation1	4/4	4/4	4/4
Desitnation2	4/4	3/4	3/4
Desitnation3	4/4	3/4	3/4

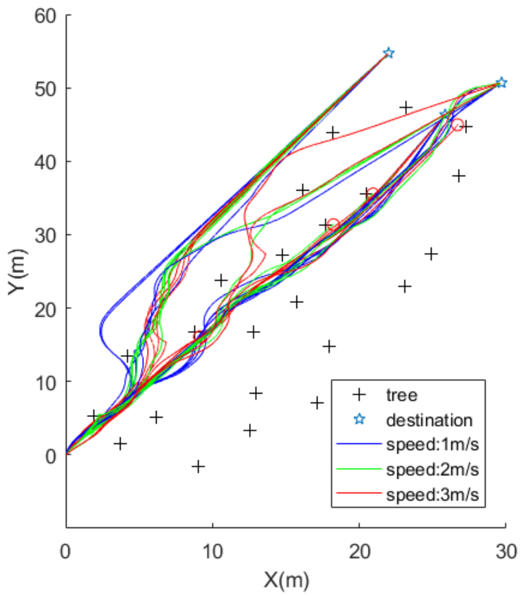
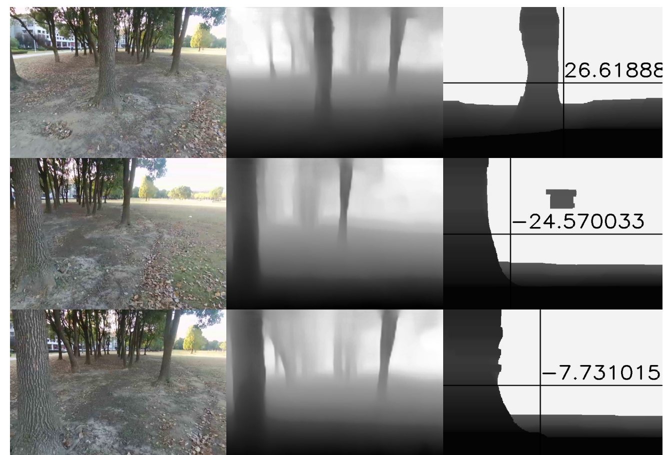


Fig. 7. Flight track in the virtual experiments. Only the failed flight is marked as a red circle.

(a) (b)

Fig. 8. The experiments in the real world. (a) The drone avoiding obstacles and reaching the destination through the trees. The red line shows the collision-free trajectory of the drone, which is based on the track in (b). (b) Shows the fight trajectory recorded by odometry.



(a) Raw images. (b) Depth images. (c) Expanded depth.

Fig. 9. Quadrotor avoiding obstacles based on the proposed method. (a) Shows three raw images every 1 second from the perspective of the quadrotor. (b) The depth images corresponding to the raw images. (c) Shows the expanded depth iamges. The number in the images is the rotation angle to turn away from obstacles. A positive value means controlling the drone to the right and vice versa.

B. Performance of the Drone Obstacles Avoidance in Real Environments

In real environments, we do the experiments with a lightweight commercial quadrotor Parrot Bebop2. We implement the obstacle avoidance algorithm on the ground server (a laptop with an Intel i5-8300H CPU @2.30 GHz and Nvidia GTX 1060 with 6 GB memory). The implementation is written in Python and C++, and uses ROS to communicate with Bebop2. The control commands are dispatched to the dorne at 10Hz. And the flight speed of UAV in the experiments is about 0.7m/s. We perform the experiments in the trees. The drone is hovering , given a target bearing direction and starts moving towards the destination through trees, Fig.8. Once the obstacles are detected (using expanded depth image) in the direction of flight, the rotation angle for avoiding obstacles is generated immediately. And then, the drone rotates to the initial direction. An instance of obstacle avoidance is shown in Fig.9.

V. CONCLUSION

This paper presents a novel method for obstacle avoidance of an UAV eqiuped with a monocular camerea. In both virtual and real world environments, the drone can automatically avoid obstacles and fly towards the destination. We generate and refine the depth image from RGB image by CNN. And the experiments demonstrate the proposed method is effctive. In the future, we will do much more experiments in both indoor and outdoor complex environments. And we make attempt to obtain much preciser depth images from a monocular camera.

ACKNOWLEDGMENT

This work was supported by Shanghai Science and Technology Commission Scientific Research Project with project Nos. 17DZ1100803.

REFERENCES

[1] Z. Li and N. Snavely,“MegaDepth: Learning Single-View Depth Prediction from Internet Photos,” 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, 2018, pp. 2041-2050.

[2] P. Chakravarty, K. Kelchtermans, T. Roussel, S. Wellens, T. Tuytelaars and L. Van Eycken, "CNN-based single image obstacle avoidance on a quadrotor," 2017 IEEE International Conference on Robotics and Automation (ICRA), Singapore, 2017, pp. 6369-6374.

[3] Yang X , Luo H , Wu Y , et al. Reactive Obstacle Avoidance of Monocular Quadrotors with Online Adapted Depth Prediction Network[J]. Neurocomputing, 2018.

[4] L. Matthies, R. Brockers, Y. Kuwata and S. Weiss, "Stereo vision-based obstacle avoidance for micro air vehicles using disparity space," 2014 IEEE International Conference on Robotics and Automation (ICRA), Hong Kong, 2014, pp. 3242-3249.

[5] K. Bipin, V. Duggal and K. Madhava Krishna, "Autonomous navigation of generic monocular quadcopter in natural environment," 2015 IEEE International Conference on Robotics and Automation (ICRA), Seattle, WA, 2015, pp. 1063-1070.

[6] A. Al-Kaff, Qinggang Meng, D. Martín, A. de la Escalera and J. M. Armingol, "Monocular vision-based obstacle detection/avoidance for unmanned aerial vehicles," 2016 IEEE Intelligent Vehicles Symposium (IV), Gothenburg, 2016, pp. 92-97.

[7] A. Al-Kaff, Qinggang Meng, D. Martín, A. de la Escalera and J. M. Armingol, "Monocular vision-based obstacle detection/avoidance for unmanned aerial vehicles," 2016 IEEE Intelligent Vehicles Symposium (IV), Gothenburg, 2016, pp. 92-97.

[8] N. Gageik, P. Benz and S. Montenegro, "Obstacle Detection and Collision Avoidance for a UAV With Complementary Low-Cost Sensors," in IEEE Access, vol. 3, pp. 599-609, 2015.

[9] H. Oleynikova, D. Honegger and M. Pollefeys, "Reactive avoidance using embedded stereo vision for MAV flight," 2015 IEEE International Conference on Robotics and Automation (ICRA), Seattle, WA, 2015, pp. 50-56.

[10] C. Bills, J. Chen, and A. Saxena, “Autonomous MAV flight in indoor environments using single image perspective cues,” in 2011 IEEE International Conference on Robotics and Automation (ICRA), May 2011, pp. 57765783.

[11] Lee, JeongOog, Lee, KeunHwan, Park, SangHeon, et al. Obstacle avoidance for small UAVs using monocular vision[J]. Aircraft Engineering and Aerospace Technology, 2011, 83(6):397-406.

[12] T. Mori and S. Scherer, “First results in detecting and avoiding frontal obstacles from a monocular camera for micro unmanned aerial vehicles,” in Robotics and Automation (ICRA), 2013 IEEE International Conference on. IEEE, 2013, pp. 17501757.

[13] S. Saha, A. Natraj and S. Waharte, "A real-time monocular vision-based frontal obstacle detection and avoidance for low cost UAVs in GPS denied environment," 2014 IEEE International Conference on Aerospace Electronics and Remote Sensing Technology, Yogyakarta, 2014, pp. 189-195.

[14] AirSim open source platform at github: Available at http://github.com/Mincrosoft/AirSim, accessed 2017

[15] Eigen D, Puhrsch C, Fergus R, et al. Depth Map Prediction from a Single Image using a Multi-Scale Deep Network[C]. neural information processing systems, 2014: 2366-2374.

[16] F. Liu, C. Shen, G. Lin and I. Reid, "Learning Depth from Single Monocular Images Using Deep Convolutional Neural Fields," in IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 38, no. 10, pp. 2024-2039, 1 Oct. 2016.

[17] C. Godard, O. M. Aodha and G. J. Brostow, "Unsupervised Monocular Depth Estimation with Left-Right Consistency," 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, 2017, pp. 6602-6611.