IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Date of publication xxxx 00, 0000, date of current version xxxx 00, 0000.

Digital Object Identifier 10.1109/ACCESS.2017.Doi Number

Monocular Vision-Based Obstacle Detection and Avoidance for a Multicopter

Hsiang-Chieh Chen[1], Member, IEEE

[1]Department of Electrical Engineering, National United University, Miaoli 36063, TAIWAN

Corresponding author: Hsiang-Chieh Chen (e-mail: chc@nuu.edu.tw).

This work was supported in part by the Ministry of Science and Technology of Taiwan under Grant MOST-107-2221-E-239-034.

ABSTRACT This article presents a monocular vision-based algorithm for detecting obstacles and identifying obstacle-aware regions, developed to be used for collision avoidance by a multicopter. The first step of our algorithm is to predict a disparity image from a single-view image via implementing a deep encoder-decoder network. All pixels in this disparity prediction are then categorized as one of three classes, obstacle, road, or obstacle-free, by combining V-disparity analysis and a fuzzy inference system. For pixels belonging to obstacle objects, obstacle-aware regions are generated within the field of visual perception. To accommodate the safety margins of a multicopter, intermediate waypoints are then added to obtain a new flyable path that passes through an unknown environment safely. Experimental results verified the effectiveness of the detection of obstacles and the identification of obstacle-aware regions. The accuracy of disparity prediction and monocular depth estimation were quantitatively compared to support the feasibility of monocular vision in obstacle avoidance. Furthermore, the entire algorithm was successfully tested on a robotic platform, autonomously flying a hexacopter in an outdoor space with obstacles. In conclusion, the proposed monocular algorithm performs well for obstacle detection and depth estimation and is potentially an alternative to a binocular solution.

INDEX TERMS Collision avoidance, machine learning, monocular depth estimation, multicopter, obstacle detection, robot vision systems, unmanned aerial vehicles.

I. INTRODUCTION

Over the past decade, the use of unmanned aerial vehicles (UAVs) has rapidly grown, particularly in non-military purposes. A large variety of UAVs have entered the commercial market, finding uses in industry, agriculture, geomorphologic survey, and inspection. With the development of microsystems, small and micro UAVs (sUAVs and MAVs) such as multicopters have achieved widespread distribution. So far, they are mainly deployed outdoors with the assistance of a global positioning system (GPS). However, GPS is not always available indoors or in a GPS-denied environment. Moreover, information about dangerous obstacles near a UAV cannot be obtained from GPS alone.

Many navigation and obstacle-avoidance methodologies exist to address this issue, using various sensors, such as cameras, infrared, and laser rangefinders. Obstacle detection is a vital feature for an autonomous UAV and still presents a challenging problem. This challenge is growing as commercial UAVs become smaller. Most rangefinders are too heavy and power-consuming for use in sUAVs and MAVs. By contrast, an onboard camera device has potential in this role, due to its lighter weight and relatively low power consumption. Additionally, such a vision sensor typically offers more space information than rangefinders. Accordingly, various vision-based approaches have been introduced to address the needs of obstacle detection and avoidance in UAVs. Both monocular and binocular vision methods are popular and can be found in many studies.

Due to size, weight, and power constraints, a monocular camera is more practical than binocular one for us in a multicopter. Accordingly, this study focuses on prior work based on monocular vision. In [1], a 3D map of the UAV's surroundings is estimated by matching several monocular images and is then employed to localize a target for autonomously traversing an unknown environment. Al-Kaff et al. [2] propose a method for analyzing the size changes of obstacles from consecutive image frames. This method then judges whether an obstacle may cause a collision by considering the area ratio of an obstacle and the UAV's

position. And in [3], the authors present an approach to the computation of a regularized depth map that is subsequently used for waypoint generation. The sensory device in question contains only an inertial measurement unit (IMU) and a monocular camera, and thus is compatible with small-scale aerial vehicles.

Before the rise of deep learning, various methods of feature point detection and matching were frequently used such as structure from motion (SfM) [4], optical flow [5], [6], [7], [8], and so forth. A comprehensive survey of existing methods for navigating aerial vehicles is summarized in [9], including obstacle detection systems. The limitation of such methods is that the feature points extracted in any given image frame are sparse, and sometimes are not dense enough to describe the appearance of obstacles. Such a situation may cause a collision when the UAV is under automatic guidance. Hence, obtaining a dense depth map is essential for autonomous travel in cluttered environments. Binocular/stereo vision is often employed to predict a depth map by running global matching algorithms on two-view images [10]. Unfortunately, any binocular vision module needs a real baseline between two cameras. Due to size and weight considerations, binocular vision is not a feasible choice in our work. Thus, we attempt to obtain a dense depth map with a single camera, even if only given a single image. Ranftl et al. [11] present a method to produce a dense depth map from two consecutive frames. Moving objects are then reconstructed by leveraging optical flow to segment dynamic scenes. There are other studies that obtain depth based on geometric methods [12], [13], [14].

In recent years, deep learning techniques have been widely adopted in computer vision applications. Learning-based methodologies have been verified effectively in designated scenarios, though they have not been demonstrated to generalize well. There is a large amount of research that investigates the topic of learning-based monocular depth estimation from input images. For depth estimation problems, a supervised learning method often suffers from a shortage of true depth data as a supervisory signal. In practice, it is difficult to collect the dense ground truth of depth. Thus, we focus on the methods that can be trained without using real depth values but with self-supervision.

One class of self-supervision comes from a pair of well-rectified images. Eigen et al. [15] utilize a convolutional neural network (CNN) with a deep architecture to predict a depth map from a single-view image. More recently, several methods exploiting CNN-based techniques have been introduced. Liu et al. [16] present a unified framework that combines a deep CNN with a continuous conditional random field, retaining good performance in general scenes without geometric priors. Godard et al. [17] exploit epipolar geometry constraints to predict disparity images by training their network with an image reconstruction loss. Given a pair of rectified images I^l and I^r corresponding to left and right color images, their presented model attempts to reconstruct the right image from the left image. First, two disparity images $\widetilde{\mathcal{D}}^l$ and $\widetilde{\mathcal{D}}^r$ are generated using an encoder-decoder architecture from the left image I^l. By utilizing bilinear sampler [18], the reconstructed left image is obtained from $\widetilde{\mathcal{D}}^l$ and I^r. Given the parameters of the binocular camera used for collecting training data, a real distance is easily obtained. Godard's approach has been extended by additionally involving trinocular assumptions [19], and utilizing semantic information, as discussed in [20] and [21].

Another form of self-supervision is to employ monocular video sequences, in which the consecutive image frames play the role of supervisory signals. In this case, the model additionally estimates the camera pose between two frames. Zhou et al. [22] use a structure consisting of depth CNN and pose CNN, respectively, for single-view depth prediction and for multi-view camera pose estimation. Furthermore, differentiable 3D loss functions are proposed to improve depth and ego-motion estimation by considering the geometry of adjacent frames [23]. Based on the model of [17], Godard et al. subsequently present the Monodepth2 model, which can be trained using binocular pairs and monocular sequences [24]. From experimental comparison, training with image pairs tends to obtain better quality results.

Surveying learning-based monocular depth estimation methodologies over the past few years, the Monodepth model presented in [17] outperformed the majority of previous works extant at that time and motivated several subsequent studies. Based on the monocular disparity or depth (inverse to each other) estimation, we attempted to improve the Monodepth model and then apply the result to avoid collisions while navigating a small multicopter. The main contributions and innovations of this paper are briefly summarized below.

(1) We present a modified loss function for learning a monocular disparity estimation model. The modified loss considers obstacles within the field of visual perception. Consequently, the accuracy of disparity estimation results for frontal obstacles is improved.
(2) We combine a disparity analysis approach and a fuzzy inference system to classify each disparity pixel into one of three different classes: obstacle, road, and obstacle-free. The pixels belonging to obstacles are then used in constructing obstacle-aware regions.
(3) A simple path selection scheme is introduced in the application part for piloting a small multicopter to avoid obstacles when flying to a destination.

The remainder of this paper is organized as follows. Section II introduces the main algorithm of the proposed method. In Section III, we describe the implementation details and give some experimental discussions. Section IV provides more evaluation to verify the effectiveness of our method. Finally, the conclusions are drawn.

II. THE PROPOSED METHOD

This section mainly presents a monocular vision-based method for detecting obstacles and their depth information.

The major procedures of our proposed method include (1) monocular disparity estimation, (2) pixel classification in disparity image, and (3) generation of obstacle-aware regions.

A. MONOCULAR DISPARITY ESTIMATION

Inspired by [17], this subsection aims to learn a model ψ for predicting a disparity image $\widetilde{\mathcal{D}}^l$ from a given left-view image I^l. It is denoted by

$$\psi: I^l(u,v) \to \widetilde{\mathcal{D}}^l(u,v), \tag{1}$$

where (u,v) is pixel coordinate, with $1 \le u \le w$ and $1 \le v \le h$, where w and h are the width and height of these two images. In this work, the model ψ is achieved by a deep encoder-decoder architecture that is designed to be trained in a self-supervised manner. Each training sample is composed of a pair left and right well-rectified images, $I^l(u,v)$ and $I^r(u,v)$, respectively. Fig. 1(a) illustrates the self-supervised training scheme that was initially introduced in [17]. Learning a disparity image corresponding to the left-view image is regarded as a stereo image reconstruction problem. Given a left-view image I^l, the model ψ first predicts two disparity images, $\widetilde{\mathcal{D}}^l$ and $\widetilde{\mathcal{D}}^r$. For the right image I^r, the left disparity estimation $\widetilde{\mathcal{D}}^l$ is used to reconstruct its opposite left-view correspondence $\tilde{I}^l$. Similarly, $\widetilde{\mathcal{D}}^r$ is used for reconstructing $\tilde{I}^r$ from I^l. Let $\mathcal{L}$ be the total loss of learning model ψ, defined by $\mathcal{L} = \sum_{s=1}^{4} \mathcal{L}_s$, where

$$\begin{aligned}\mathcal{L}_s = \alpha_{\mathrm{a}}(\mathcal{L}_{\mathrm{a}}^l + \mathcal{L}_{\mathrm{a}}^r) + \alpha_{\mathrm{d}}\big(\mathcal{L}_{\mathrm{d}}^l + \mathcal{L}_{\mathrm{d}}^r\big) \\ +\alpha_{\mathrm{c}}(\mathcal{L}_{\mathrm{c}}^l + \mathcal{L}_{\mathrm{c}}^r).\end{aligned} \tag{2}$$

Here, the subscript s indicates four different scales of disparity pyramid and $1 \le s \le 4$. The superscript (either l or r) stands for the left- or right-view image, α_{a}, α_{d} and α_{c} are the weights of their corresponding loss terms, $\mathcal{L}_{\mathrm{a}}$ aims to make the reconstructed image similar to its corresponding supervisory input, $\mathcal{L}_{\mathrm{d}}$ is an edge-aware term to enforce locally smooth in disparity estimation, and $\mathcal{L}_{\mathrm{c}}$ is for enhancing the estimated left and right disparity maps to be consistent.

Each above-mentioned loss in terms of superscript l is described in detail in the following contents. The opposite version for superscript r could be derived in the same way.

1) APPEARANCE MATCHING LOSS

The model ψ first predicts $\widetilde{\mathcal{D}}^l$ whereas inputting a left-view image I^l, followed by a bilinear sampler that generates a reconstructed left image $\tilde{I}^l$.

$$\tilde{I}^l = I^r \otimes \widetilde{\mathcal{D}}^l \tag{3}$$

Here, the symbol $\otimes$ represents the operation of bilinear sampler accomplished by a spatial transformer network [18]. The single scale structural similarity (SSIM) and L1 penalty introduced in [25] are included to build the appearance matching loss, as formulated by

$$\begin{aligned}\mathcal{L}_{\mathrm{a}}^l = \frac{1}{w \times h}\Bigg\{\sum_{u,v}\beta\left(1 - \mathrm{SSIM}_{(u,v)}(I^l, \tilde{I}^l)\right) \\ +\sum_{u,v}(1-\beta)\left|I^l(u,v) - \tilde{I}^l(u,v)\right|\Bigg\},\end{aligned} \tag{4}$$

where β is a coefficient ranged in $[0,1]$, w and h are image width and height. A pixel-wise measure $\mathrm{SSIM}_{(u,v)}$ is used and is calculated within a 3×3 mask centered at the pixel (u,v).

$$\mathrm{SSIM}_{(u,v)}(I^l, \tilde{I}^l) = \frac{(2\mu_l\tilde{\mu}_l + c_1)(2\sigma_{\mathrm{cov}}^2 + c_2)}{(\mu_l^2 + \tilde{\mu}_l^2 + c_1)(\sigma_l^2 + \tilde{\sigma}_l^2 + c_2)} \tag{5}$$

with the conditions:

- The mask covers nine pixels (x,y) for $x = u-1, u, u+1$ and $y = v-1, v, v+1$.
- μ_l is the average in the mask of $I^l(x,y)$.
- $\tilde{\mu}_l$ is the average in the mask of $\tilde{I}^l(x,y)$.
- σ_l^2 is the variance in the mask of $I^l(x,y)$.
- $\tilde{\sigma}_l^2$ is the variance in the mask of $\tilde{I}^l(x,y)$.
- σ_{cov}^2 is the covariance in the mask of $I^l(x,y)$ and $\tilde{I}^l(x,y)$.
- The constants are predefined as $c_1 = 0.01^2$ and $c_2 = 0.03^2$.

2) DISPARITY SMOOTHNESS LOSS

This term is a smoothness constraint for penalizing large disparity changes along the horizontal u and vertical v directions except if strong intensity gradients occur at pixel (u,v) in the input image I^l.

$$\begin{aligned}\mathcal{L}_{\mathrm{d}}^l = \frac{1}{w \times h}\Bigg\{\sum_{u,v}\left|\nabla_u \widetilde{\mathcal{D}}^l(u,v)\right| e^{-\left|\nabla_u I^l(u,v)\right|} \\ +\sum_{u,v}\left|\nabla_v \widetilde{\mathcal{D}}^l(u,v)\right| e^{-\left|\nabla_v I^l(u,v)\right|}\Bigg\},\end{aligned} \tag{6}$$

where ∇_u and ∇_v are difference operators along the u and v axes, respectively. This loss is first presented in [26].

3) LEFT-RIGHT DISPARITY CONSISTENCY LOSS

The model ψ simultaneously predicts both the left and right disparity images, whereas only inputting a left-view color image. To enhance the left-right consistency between the estimated disparities, $\widetilde{\mathcal{D}}^l$ and $\widetilde{\mathcal{D}}^r$, the following loss term is included in the total loss.

$$\mathcal{L}_{\mathrm{c}}^l = \frac{1}{w \times h}\sum_{u,v}\left|\widetilde{\mathcal{D}}^l(u,v) - \widetilde{\mathcal{D}}^l\big(u + \widetilde{\mathcal{D}}^l(u,v), v\big)\right|. \tag{7}$$

4) OBSTACLE-WEIGHTED LOSS

The main objective of this work is to acquire obstacle information that can be then exploited to assist collision-avoidance for a small multicopter. Our modified learning scheme is plotted in Fig. 1(b), in which the novelty is to consider the information of obstacles when building the

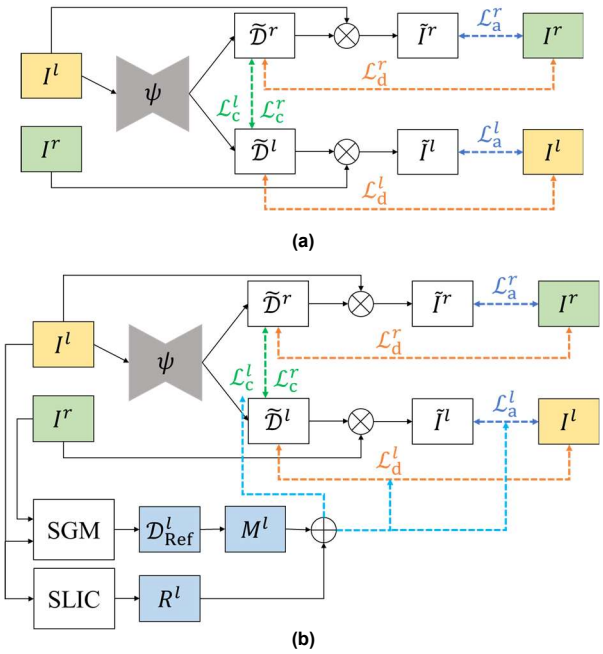

FIGURE 1. Self-supervised learning scheme for monocular disparity estimation by stereo pairs: (a) the original version, and (b) the modified version.

training loss. Consequently, the accuracy of disparity estimation results for obstacles is improved, whereas the effectiveness of the original model is preserved. Similar to [17], the proposed model also outputs dense disparity images at four scales. Hence, the loss at a specified scale s is presented by

$$\mathcal{L}'_s = \alpha_{\mathrm{a}}\left(\mathcal{L}'^{l}_{\mathrm{a}} + \mathcal{L}'^{r}_{\mathrm{a}}\right) + \alpha_{\mathrm{d}}\left(\mathcal{L}'^{l}_{\mathrm{d}} + \mathcal{L}'^{r}_{\mathrm{d}}\right) + \alpha_{\mathrm{c}}\left(\mathcal{L}'^{l}_{\mathrm{c}} + \mathcal{L}'^{r}_{\mathrm{c}}\right). \tag{8}$$

The composition of this loss function is the same as (2), but each loss term is replaced with a modified one that will be described hereafter. For the purpose of obstacle avoidance, the objects having larger disparities should be more critical than backgrounds. Hence, we propose a scheme to generate a weighting map by combining semi-global matching (SGM) [27] and simple linear iterative clustering (SLIC) [28]. The SGM algorithm first outputs a reference disparity map $\mathcal{D}^{l}_{\mathrm{Ref}}$ from the stereo images, I^l and I^r. This disparity map is immediately used for obtaining a weighting map M^l, as defined by

$$M^l(u,v) = \begin{cases} p, \text{ if } d^l(u,v) \le \dfrac{f^l b}{z_{\mathrm{f}}} \\ q, \text{ if } d^l(u,v) \ge \dfrac{f^l b}{z_{\mathrm{n}}} \\ p + \dfrac{(q-p)\left(d^l(u,v) - \dfrac{f^l b}{z_{\mathrm{f}}}\right)}{f^l b\left(\dfrac{1}{z_{\mathrm{n}}} - \dfrac{1}{z_{\mathrm{f}}}\right)}, \text{otherwise.} \end{cases} \tag{9}$$

where z_{n} and z_{f} are the predefined nearest and furthest obstacle-perceptual distances along the camera's principal axis, f^l is the focal length (in pixels) of the left camera, and b is the baseline between left and right cameras (in millimeters). Notably, each pixel value in this weighting map is limited in the interval $[p, q]$.

Meanwhile, the SLIC approach segments the left-view image into an image R^l that is full of superpixels. To preserve the consistent disparity in a superpixel that belongs to a small part of an obstacle, the mean of pixel values within the superpixel $\mathcal{S}$ is adopted to replace the original $M^l(u,v)$.

$$M^l(u,v) \leftarrow \frac{\sum_{(u,v)\in\mathcal{S}} M^l(u,v)}{\#(\mathcal{S})}, \forall\, (u,v) \in \mathcal{S} \tag{10}$$

where $\#(\mathcal{S})$ is the number of pixels in $\mathcal{S}$. The last step of our proposed method is to use the weighting map to regulate the training loss. Thereby, the loss is modified as follows.

$$\mathcal{L}'^{l}_{\mathrm{a}} = \frac{1}{w\times h}\left\{\sum_{u,v}\beta M^l(u,v)\left(1 - \mathrm{SSIM}_{(u,v)}(I^l, \tilde{I}^l)\right) + \sum_{u,v}(1-\beta)M^l(u,v)\left|I^l(u,v) - \tilde{I}^l(u,v)\right|\right\} \tag{11}$$

$$\mathcal{L}'^{l}_{\mathrm{d}} = \frac{1}{w\times h}\left\{\sum_{u,v} M^l(u,v)\left|\nabla_u \widetilde{\mathcal{D}}^l(u,v)\right| e^{-|\nabla_u I^l(u,v)|} + \sum_{u,v} M^l(u,v)\left|\nabla_v \widetilde{\mathcal{D}}^l(u,v)\right| e^{-|\nabla_v I^l(u,v)|}\right\} \tag{12}$$

$$\mathcal{L}'^{l}_{\mathrm{c}} = \frac{1}{w\times h}\sum_{u,v} M^l(u,v)\left|\widetilde{\mathcal{D}}^l(u,v) - \widetilde{\mathcal{D}}^l\left(u + \widetilde{\mathcal{D}}^l(u,v), v\right)\right| \tag{13}$$

The standard network of the model ψ is designed as a deep encoder-decoder architecture, as illustrated in Fig. 2, in which the encoder is a ResNet-50 [29] and the decoder is composed of the layers of deconvolution, up-sampling, and max-pooling. The encoder can be replaced by certain other network backbones such as a VGG network [30]. Fig. 3 shows an exemplary result of the original color image (selected from the Yonsei's RGB-D dataset [31]) and its dense disparity estimation $\widetilde{\mathcal{D}}^l$ by the model ψ. Here, both images are sized of 640×384 pixels. Implementation details are described in Subsection III-A.

B. PIXEL CLASSIFICATION IN DISPARITY IMAGE

Based on observations from the disparity estimation, both the road plane and obstacles are inferred to show large disparities because they are physically close to the camera. However, the obstacles are dangerous and have to be avoided, whereas the road plane need not be. Thus, an algorithm for differentiating

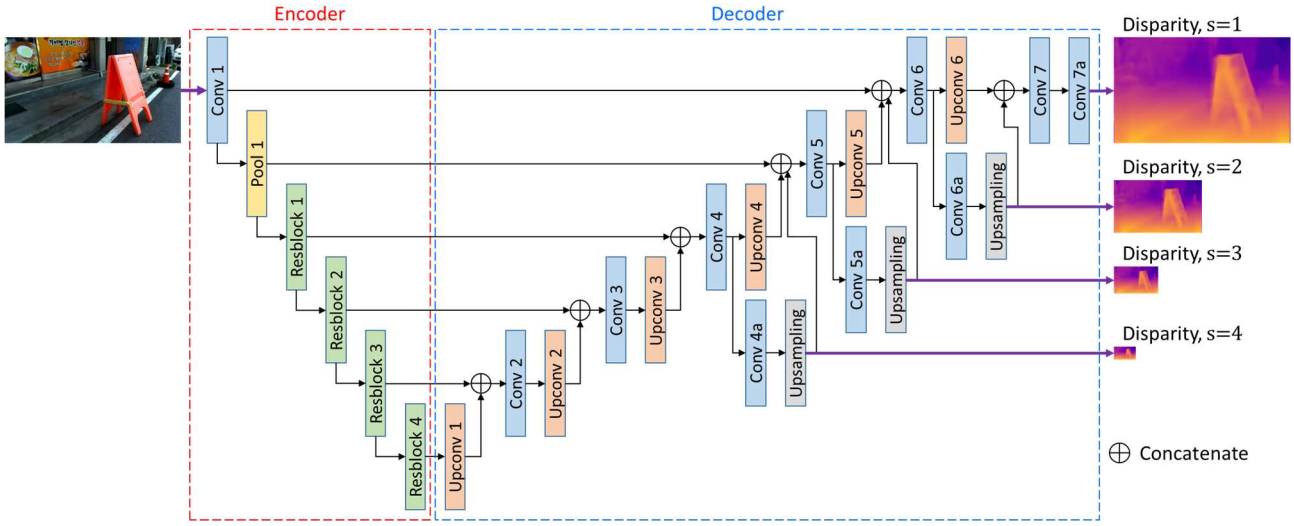

FIGURE 2. **The standard architecture of our proposed monocular disparity estimation model used in this work.**

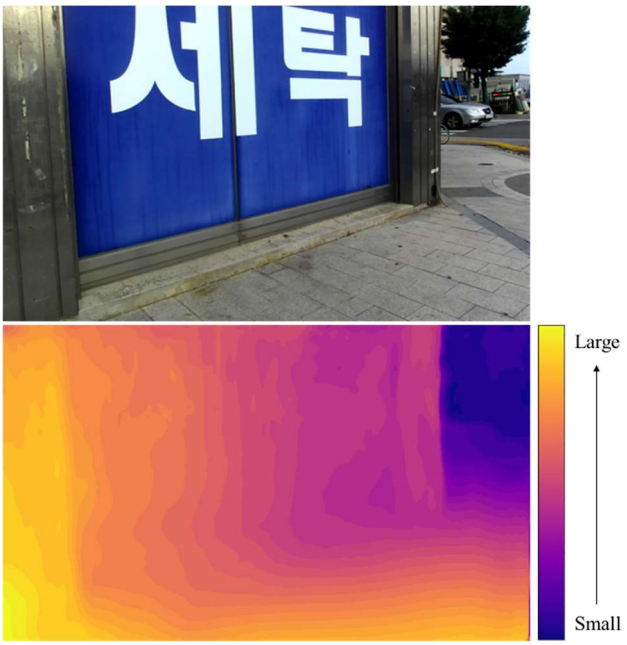

FIGURE 3. **A street-view example: the original color image (upper), and its disparity estimation by the proposed method.**

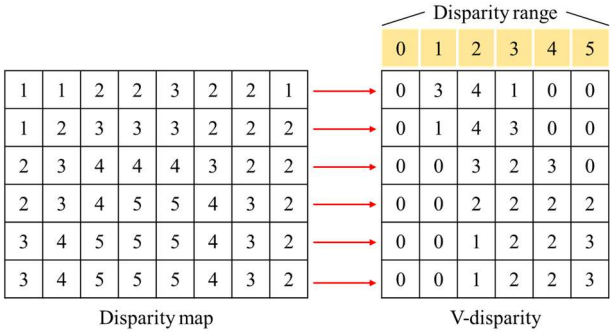

FIGURE 4. **A synthesis example of a disparity map and the V-disparity generation.**

the road region from obstacles is necessary. Using an extra deep CNN for semantic segmentation is useful in order to draw the road region, but computationally intensive. Instead, we present a computation-efficient approach for classifying each pixel in the disparity image into three classes: road, obstacle, and obstacle-free. We will now describe our method in detail.

Let $\Gamma: \widetilde{\mathcal{D}}^l \longrightarrow \widetilde{\mathcal{V}}^l$ be a mapping function that transforms the estimated disparity image to its corresponding V-disparity map [32]. The function Γ accumulates the number of pixels with the same disparity along a given line j in $\widetilde{\mathcal{D}}^l$. Fig. 4 illustrates a simple example of generating a V-disparity map from a disparity image. Notably, the V-disparity map is of size $(d_{\max}+1)\times h$, where $d_{\max}$ is the maximal disparity value in $\widetilde{\mathcal{D}}^l$, and h is the height of the disparity map. The pixel value at $(i+1, j)$ in V-disparity indicates the number of occurrences of disparity value i along line j in $\widetilde{\mathcal{D}}^l$. Next, the Canny edge detector is applied to this V-disparity, including the process of Gaussian blurring. Fig. 5 shows the results of V-disparity generation and the following edge detection, in which (a) is the disparity image identical to the lower part of Fig. 3, and the size of (b) and (c) is 39×384 pixels due to $d_{\max} = 38$. When necessary, images can be stretched wider for ease of observation. From Fig. 5(b), we observed that the road plane is accumulated to form a straight line with a specified slope in V-disparity, which is the bright yellow line segment (its slope is from top-left to bottom-right in this map). Theoretically, the pixel (u, v) in $\widetilde{\mathcal{D}}^l$ is considered as a road pixel if its corresponding V-disparity pixel $\left(\widetilde{\mathcal{D}}^l(u, v), v\right)$ lies on (or is nearby) a line whose slope ranges in a specified interval. Fig. 6 presents the procedures of obtaining the road profile for such a line segment, emphasized by a red color. Table I summarizes the main steps of our proposed pixel classification algorithm. Each pixel in the estimated disparity image is classified into one of three classes: road, obstacle, and obstacle-free. Fig. 7 shows the result of performing this pixel-

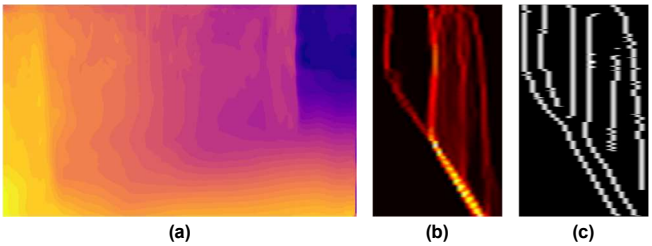

FIGURE 5. An example of disparity analysis: (a) the disparity map estimation in Fig. 3, (b) the V-disparity generation result, and (c) the Canny edge detection result.

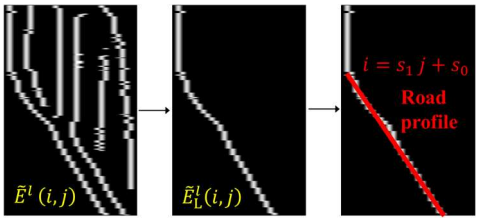

FIGURE 6. V-disparity analysis for finding the road profile.

FIGURE 7. The result of pixel classification of Fig. 5(a).

TABLE I
MAIN PROCEDURES OF PIXEL CLASSIFICATION IN DISPARITY IMAGE

Algorithm 1. Pixel classification for dense disparity map		
Inputs:		Predicted disparity image $\widetilde{\mathcal{D}}^l$ sized of $w \times h$ pixels.
Outputs:		Pixel classification result of $\widetilde{\mathcal{D}}^l$, denoted by $\tilde{C}^l$.
Steps ↓	1:	Generate V-disparity map $\tilde{\mathcal{V}}^l$ of size $(d_{\max}+1) \times h$ pixels.
	2:	Perform Canny edge detector on $\tilde{\mathcal{V}}^l$, and obtain a binary edge map $\tilde{E}^l$.
	3:	For each image line j in $\tilde{E}^l$, only preserve the leftmost edge point, so that $\tilde{E}^l \longrightarrow \tilde{E}^l_{\mathrm{L}}$.
	4:	Perform Hough line transform on $\tilde{E}^l_{\mathrm{L}}$, and select the straight line $i = s_1 j + s_0$ to be the road profile candidate, and $s_1 > T_s$ (this line slope is from top-left to bottom-right).
	5:	Classify each point (u, v) in $\widetilde{\mathcal{D}}^l$ as:
	6:	Class 0: Obstacle-free pixel, if $\widetilde{\mathcal{D}}^l(u,v) < T_{\mathrm{of}}$;
	7:	Class 1: Road pixel, if $\widetilde{\mathcal{D}}^l(u,v)$ is close to $s_1 v + s_0$ and $\widetilde{\mathcal{D}}^l(u,v) \geq T_{\mathrm{of}}$;
	8:	Class 2: Obstacle pixel, if $\widetilde{\mathcal{D}}^l(u,v) \geq T_{\mathrm{of}}$ and is not a road pixel.

Let (u, v) denote the pixel coordinate in the disparity estimation $\widetilde{\mathcal{D}}^l$, whereas $1 \leq u \leq w$ and $1 \leq v \leq h$. Let (i, j) denote the pixel coordinate in the V-disparity map $\tilde{\mathcal{V}}^l$, whereas $1 \leq i \leq d_{max}+1$ and $1 \leq j \leq h$. Here, d_{max} is the maximal disparity value in $\widetilde{\mathcal{D}}^l$.

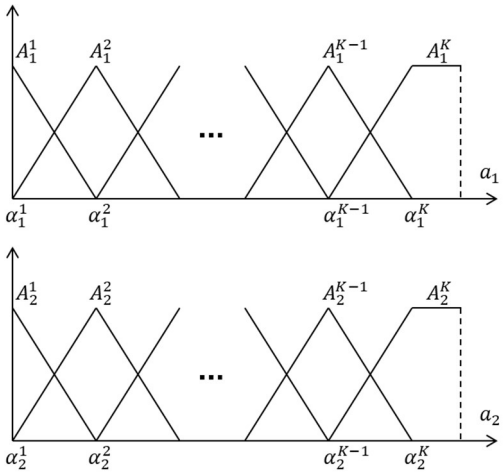

FIGURE 8. Fuzzy sets of premise variables.

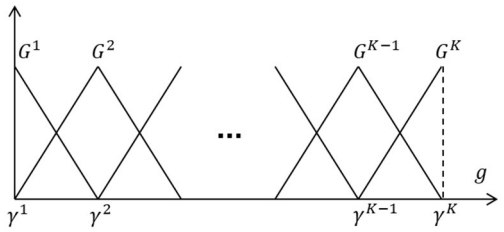

FIGURE 9. Fuzzy sets of consequent variable.

wise classification algorithm on the disparity in Fig. 5(a), in which the yellow region indicates obstacles that must be avoided, and the green is road region. Meanwhile, the black is obstacle-free pixels whose disparity values are less than a predefined threshold T_{of}. Here, this threshold is directly calculated from the furthest obstacle-perceptual distance.

It is necessary to mention that Step 7 in Table I uses a linguistic description "is close to" for classifying whether a pixel belongs to the road plane or not. We employ fuzzy logic and inference to implement such a linguistic statement. The primary configuration of completing a fuzzy inference system (FIS) comprises the fuzzification, fuzzy rules, inference engine, and defuzzification. The details of the proposed FIS are described as follows.

Let a_1 and a_2 be two premise variables of this FIS, and g denote its consequent variable. For the two premise variables, trapezoidal and triangular functions are chosen to be the membership functions of fuzzy sets, as shown in Fig. 8, and K is the number of fuzzy sets of each variable. For the consequent part, g is represented by triangular membership functions, as plotted in Fig. 9. According to the physical characteristics of image formation, the pixels of the road plane always distribute in the lower half of an image and are near the road profile that was obtained in Fig. 6. Based on the above assumptions, we formulate the two premise variables for each pixel (u, v) in $\widetilde{\mathcal{D}}^l$, as expressed by

TABLE II
FUZZY RULE TABLE FOR ROAD PIXELS CLASSIFICATION

a_2 \ a_1	A_1^1	A_1^2	...	A_1^K
A_2^1	G^K	G^{K-1}	...	G^1
A_2^2	G^{K-1}	G^{K-2}	...	G^1
⋮	⋮	⋮	⋯	⋮
A_2^K	G^1	G^1	G^1	G^1

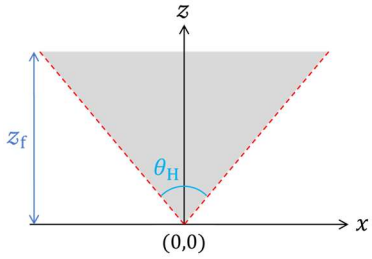

FIGURE 10. The field of visual perception (gray region) of an image sensor.

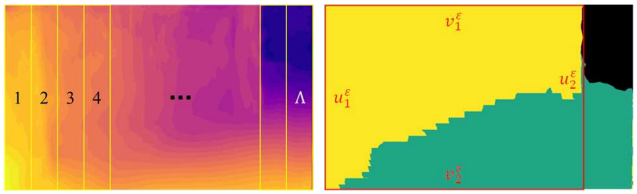

FIGURE 11. (a) The grids in the estimated disparity, and (b) the labeled object(s) in classified result.

$$\begin{cases} a_1 = \max\left(0, \left|\widetilde{\mathcal{D}}^l(u,v) - (s_1 v + s_0)\right|\right) \\ a_2 = h - v \end{cases} \tag{14}$$

where s_1 and s_0 are the road profile parameters in V-disparity. Then we objectively construct fuzzy rules using linguistic terms. For example,

IF a_1 is *Small* AND a_2 is *Small*, THEN g is *Large*.

Here, the consequent variable g means the degree that pixel (u, v) is considered to be the road plane. By complying with this concept, the m-th fuzzy rule can be formally written in the canonical format:

Rule m: IF a_1 is $\hat{A}_1^m$ AND a_2 is $\hat{A}_2^m$, THEN g is $\hat{G}^m$.

Here, $m = 1,2,\dots,K^2$; $\hat{A}_1^m$, $\hat{A}_2^m$, and $\hat{G}^m$ are respectively selected from fuzzy sets $\{A_1^1, A_1^2, \dots, A_1^K\}$, $\{A_2^1, A_2^2, \dots, A_2^K\}$, and $\{G^1, G^2, \dots, G^K\}$. Following the human intuition of road pixel classification, the fuzzy rules are tabulated in Table II. An imported input pair $(\tilde{a}_1, \tilde{a}_2)$ goes into the FIS and triggers some of the fuzzy rules. By minimum inference engine and center-of-gravity defuzzification method [33], the nonfuzzy output is finally obtained as follows.

$$\tilde{g} = \frac{\int_{\mathbb{G}} G'(g) g dg}{\int_{\mathbb{G}} G'(g) dg}, \tag{15}$$

and

$$G'(g) = \max_m \{A_1^m(\tilde{a}_1) \wedge A_2^m(\tilde{a}_2) \wedge G^m(g)\}, \tag{16}$$

where $\wedge$ is the minimum operator, and $\mathbb{G}$ is the universe of discourse. This outlines the configuration of the proposed FIS. The hyper-parameters and implementation details will be discussed in Subsection III-B.

C. GENERATION OF OBSTACLE-AWARE REGIONS

The goal of this subsection is to identify the obstacle-aware regions where line of sight is blocked by obstacles. These regions involve obstacles and areas behind obstacles, and thus should be regarded to be unsafe when flying a multicopter. The field of visual perception of a monocular camera is first defined and depicted in Fig. 10, in which z_f denotes a predefined furthest obstacle-perceptual distance and θ_H is the horizontal angle of view computed by

$$\theta_H = 2\tan^{-1}\left(\frac{w_s}{2f_e}\right), \tag{17}$$

where w_s is the width of the image sensor and f_e is the effective focal length. Both of them are measured in millimeters. With this said, the proposed method for generating obstacle-aware regions and its main steps are described in detail below, and tabulated in Table III as well.

1) The disparity estimation $\widetilde{\mathcal{D}}^l$ is first split into $\Lambda \times 1$ grids, as illustrated by the yellow rectangles in Fig. 11(a). The λ-th grid is bounded in $[u_1^\lambda, u_2^\lambda] \times [v_1^\lambda, v_2^\lambda]$ with its left, right, top, and bottom boundaries u_1^λ, u_2^λ, v_1^λ, and v_2^λ, and can be computed as follows.

$$\begin{cases} u_1^\lambda = \dfrac{w}{\Lambda}(\lambda - 1) + 1 \\ u_2^\lambda = \dfrac{w}{\Lambda}\lambda \\ v_1^\lambda = 1 \\ v_2^\lambda = h \end{cases} \tag{18}$$

To ease computation, we require that w is evenly divisible by Λ, i.e., w/Λ is a positive integer.

2) For the classification result $\tilde{C}^l$, the ε-th obstacle object is bounded in $[u_1^\varepsilon, u_2^\varepsilon] \times [v_1^\varepsilon, v_2^\varepsilon]$. As plotted in Fig. 11(b), its four boundaries are u_1^ε, u_2^ε, v_1^ε, and v_2^ε.

3) For every possible combination of (λ, ε):
   - Find the intersection of $[u_1^\lambda, u_2^\lambda]$ and $[u_1^\varepsilon, u_2^\varepsilon]$.

$$[u_1^{\lambda,\varepsilon}, u_2^{\lambda,\varepsilon}] = [u_1^\lambda, u_2^\lambda] \cap [u_1^\varepsilon, u_2^\varepsilon].$$

   - Find the top and bottom boundaries of the ε-th obstacle in $[u_1^{\lambda,\varepsilon}, u_2^{\lambda,\varepsilon}]$, denoted as $v_1^{\lambda,\varepsilon}$ and $v_2^{\lambda,\varepsilon}$.
   - Define the ROI in the λ-th grid as $[u_1^{\lambda,\varepsilon}, u_2^{\lambda,\varepsilon}] \times [v_1^{\lambda,\varepsilon}, v_2^{\lambda,\varepsilon}]$, if $[u_1^{\lambda,\varepsilon}, u_2^{\lambda,\varepsilon}]$ and $[v_1^{\lambda,\varepsilon}, v_2^{\lambda,\varepsilon}]$

are nonempty.

- Find the 95-th percentile value d_{95} of the disparity set $\{\widetilde{\mathcal{D}}^l(u,v) \mid \forall\, u_1^{\lambda,\varepsilon} \le u \le u_2^{\lambda,\varepsilon} \text{ and } v_1^{\lambda,\varepsilon} \le u \le v_2^{\lambda,\varepsilon}\}$.
- Calculate the representative depth for this tuple (λ, ε) by

$$\tilde{z}^{\lambda,\varepsilon} = f_{\text{mono}} \frac{b}{d_{95}}, \tag{19}$$

where f_{mono} (in pixels) is the focal length of the monocular camera that is used for online testing, and b is the baseline of the binocular camera that is used for collecting training data.

4) For each grid λ, find the minimal depth of all considering obstacles by

$$\tilde{z}^{\lambda} = \min\left(\tilde{z}^{\lambda,1}, \dots, \tilde{z}^{\lambda,\varepsilon}, \dots, \tilde{z}^{\lambda,\Omega}\right), \tag{20}$$

where $1 \le \varepsilon \le \Omega$ are indices of obstacles.

5) For the λ-th grid, we have its left and right boundaries, u_1^{λ} and u_2^{λ}, and a representative depth of detected obstacles, $\tilde{z}^{\lambda}$, so far. The representative depth equals to the furthest obstacle-perceptual distance z_{f} if there is no obstacle in this grid. In addition to the split grids, the field of visual perception can be split into Λ sectors as well, as plotted in Fig. 12. Notably, any realistic obstacle appears in the grids with a specified disparity distribution, and also in the sectors with a distance variation. Obviously, each grid has the information $\left(u_1^{\lambda}, u_2^{\lambda}, \tilde{z}^{\lambda}\right)$ that can be applied to depict an obstacle-aware region in its corresponding sector. In this work, this obstacle-aware region is represented by a trapezoid defined by four corners which are formulated as follows.

$$\text{Left-front point: } \left(x_1^{\lambda}, \tilde{z}^{\lambda}\right) = \left(\frac{\left(u_1^{\lambda} - u_o\right) \times \tilde{z}^{\lambda}}{f_{\text{mono}}}, \tilde{z}^{\lambda}\right)$$

$$\text{Right-front point: } \left(x_2^{\lambda}, \tilde{z}^{\lambda}\right) = \left(\frac{\left(u_2^{\lambda} - u_o\right) \times \tilde{z}^{\lambda}}{f_{\text{mono}}}, \tilde{z}^{\lambda}\right)$$

$$\text{Left-rear point: } \left(\frac{z_{\text{f}}}{\tilde{z}^{\lambda} \times x_1^{\lambda}}, z_{\text{f}}\right)$$

$$\text{Right-rear point: } \left(\frac{z_{\text{f}}}{\tilde{z}^{\lambda} \times x_2^{\lambda}}, z_{\text{f}}\right) \tag{21}$$

where f_{mono} and u_o are the focal length and image center along u-direction obtained by a common camera calibration process.

These four corners are in the camera coordinate system (measured in millimeters) centered at the optical center. The reason for using a trapezoid is that the line of sight radiates out from the camera's optical center and is blocked by obstacles. Applying this step to each grid, the obstacle-aware region of each sector is first drawn. Followed by unification of the obstacle-aware regions of all sectors, the overall obstacle-aware regions in the field of visual perception are finally obtained. Fig. 13 shows the result of obstacle-aware regions for the disparity image Fig. 11, in which $\Lambda = 32$ and $z_{\text{f}} = 600\,\text{cm} = 6000\text{mm}$ are predetermined.

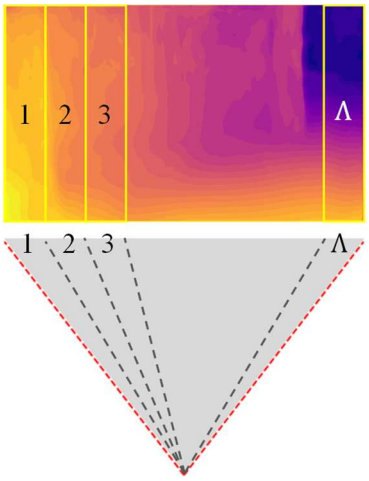

FIGURE 12. **The relation between grids and sectors.**

TABLE III
MAIN PROCEDURES OF OBSTACLE-AWARE REGION GENERATION

Algorithm 2 Obstacle-aware region generation

Inputs: Monocular disparity estimation $\widetilde{\mathcal{D}}^l$, and its pixel classification $\tilde{C}^l$.

Outputs: Obstacle-aware regions in the field of visual perception.

Steps ↓

1: Split the estimated disparity map $\widetilde{\mathcal{D}}^l$ into Λ grids. The λ-th grid is bounded in $[u_1^{\lambda}, u_2^{\lambda}] \times [v_1^{\lambda}, v_2^{\lambda}]$.

2: Label the obstacle objects in $\tilde{C}^l$ using the connected component method. The ε-th obstacle object is bounded in $[u_1^{\varepsilon}, u_2^{\varepsilon}] \times [v_1^{\varepsilon}, v_2^{\varepsilon}]$, and $1 \le \varepsilon \le \Omega$.

3: For the λ-th grid:
For the ε-th obstacle:
- Calculate the intersection of $[u_1^{\lambda}, u_2^{\lambda}]$ and $[u_1^{\varepsilon}, u_2^{\varepsilon}]$. Thus, this overlapping interval is $[u_1^{\lambda,\varepsilon}, u_2^{\lambda,\varepsilon}]$.
- Find the upper and lower boundaries of obstacle ε in $[u_1^{\lambda,\varepsilon}, u_2^{\lambda,\varepsilon}]$. The upper and lower boundaries are $v_1^{\lambda,\varepsilon}$ and $v_2^{\lambda,\varepsilon}$, respectively.
- Find the ROI bounded in $[u_1^{\lambda,\varepsilon}, u_2^{\lambda,\varepsilon}] \times [v_1^{\lambda,\varepsilon}, v_2^{\lambda,\varepsilon}]$.
- For the ROI:
  - Find the 95-th percentile value of disparities.
  - Find its representative depth $\tilde{z}^{\lambda,\varepsilon}$

Calculate the minimal obstacle depth for grid λ by $\tilde{z}^{\lambda} = \min(\tilde{z}^{\lambda,1}, \dots, \tilde{z}^{\lambda,\varepsilon}, \dots, \tilde{z}^{\lambda,\Omega})$ for $\varepsilon = 1,2,\dots,\Omega$.

4: For the λ-th grid:
The left and right boundaries u_1^{λ} and u_2^{λ}, and its representative distance $\tilde{z}^{\lambda}$ are already obtained.
- Calculate the four corners using (21), and form an obstacle-aware region using a trapezoid.

5: Draw the union of every trapezoid in the last step, and form the obstacle-aware regions in the field of perception.

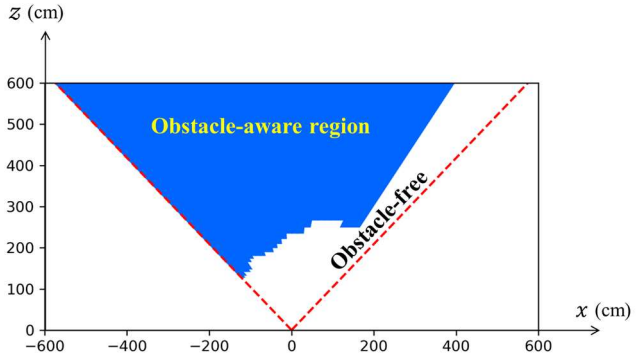

FIGURE 13. The obstacle-aware region of taking Fig. 11 as an example.

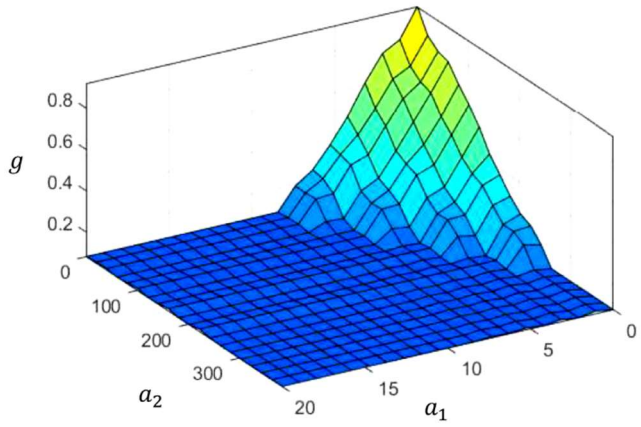

FIGURE 14. The input to output mapping surface of the proposed FIS with determined parameters in subsection III-B.

III. IMPLEMENTATION AND EXPERIMENTS

In this section, we implemented the proposed algorithm on a GPU-accelerated computer. The following discussion is focused on: 1) training and testing the proposed monocular disparity estimation model, 2) the fuzzy inference system used in pixel classification and its computation, and 3) two different situations of generating obstacle-aware regions.

A. IMPLEMENTATION OF MONOCULAR DISPARITY ESTIMATION MODEL

The proposed model is implemented in TensorFlow and Python and is trained using well-rectified image pairs. The standard network, as plotted in Fig. 2, contains over 55 million trainable parameters and takes about 6-8 hours to pretrain using a CUDA-based GPU engine, GeForce GTX 1080 Ti, on the outdoor sample data of RGB-D dataset provided by Yonsei University [31]. Next, this pretrained model is refined by our own dataset, collected outdoors. Notably, both of Yonsei's and our datasets are collected by a camera called ZED, which has built-in binocular cameras.

In this subsection, over 6000 pairs of rectified images were used in learning the model, and 500 pairs in testing. We set the hyperparameters as: $\alpha_{\mathrm{a}} = 1$, $\alpha_{\mathrm{d}} = 0.1/s$ ($s = 1,2,3,4$ with respect to different scales), $\alpha_{\mathrm{c}} = 1$, the estimated disparity limited in $d_{\max} \leq 0.3w$, $w = 640$, $h = 384$, batch size $= 8$, epochs $= 200$, $\beta = 0.5$, $p = 1$, $q = 3$, optimized by Adam [34] with the commonly-used settings $\beta_1 = 0.9$, $\beta_2 = 0.999$, and $\epsilon = 10^{-8}$, and the initial learning rate $\eta = 10^{-4}$ which is kept for the first 50 epochs and halved every 50 epochs until the end.

Certain commonly-used data augmentation techniques are employed in our works, such as horizontal flipping with a half probability, and brightness and color shifts by sampling from uniform distributions ranging in $[0.7, 1.3]$ for brightness, and $[0.8, 1.2]$ for each color channel. The complete compositions of the encoder and decoder parts in Fig. 2 are separately presented in Tables IV and V.

TABLE IV
THE STANDARD ARCHITECTURE OF ENCODER IN THE PROPOSED MODEL

Encoder- ResNet-50				
Layer name	Kernel size	Stride	Channels	Repeat times
Conv 1	7×7	2	3→64	1
Pool 1	3×3	2		1
Resblock 1	$\begin{bmatrix}1\times1\\3\times3\\1\times1\end{bmatrix}$	1	64→64	3
Resblock 2	$\begin{bmatrix}1\times1\\3\times3\\1\times1\end{bmatrix}$	1	64→128	4
Resblock 3	$\begin{bmatrix}1\times1\\3\times3\\1\times1\end{bmatrix}$	1	128→256	6
Resblock 4	$\begin{bmatrix}1\times1\\3\times3\\1\times1\end{bmatrix}$	1	256→512	3

TABLE V
THE STANDARD ARCHITECTURE OF DECODER IN THE PROPOSED MODEL

Decoder				
Layer name	Kernel size	Stride	Channels	Repeat times
Upconv 1	3×3	2	512→512	1
Conv 2	3×3	1	768→512	1
Upconv 2	3×3	2	512→256	1
Conv 3	3×3	1	384→256	1
Upconv 3	3×3	2	256→128	1
Conv 4	3×3	1	192→128	1
Conv 4a	3×3	1	128→2	1
Upconv 4	3×3	2	128→64	1
Conv 5	3×3	1	130→64	1
Conv 5a	3×3	1	64→2	1
Upconv 5	3×3	2	64→32	1
Conv 6	3×3	1	98→32	1
Conv 6a	3×3	1	32→2	1
Upconv 6	3×3	2	32→16	1
Conv 7	3×3	1	18→16	1
Conv 7a	3×3	1	16→2	1

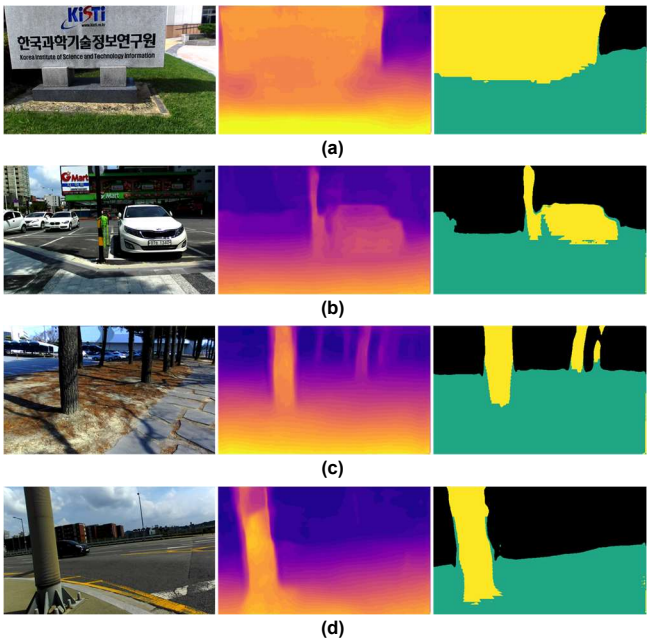

FIGURE 15. **Four examples of disparity estimation and analysis: the original image (left), the estimated disparity (middle), and the pixel classification (right).**

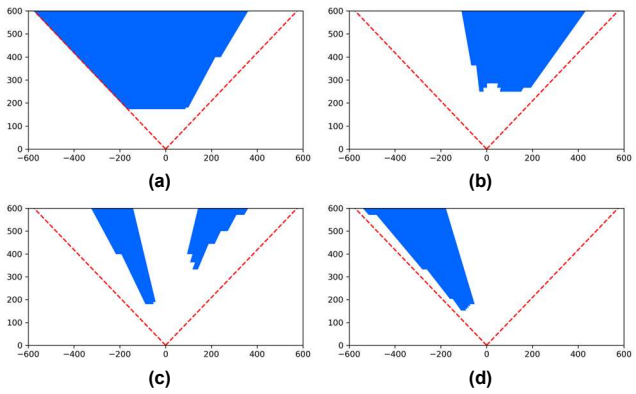

FIGURE 16. **The obstacle-aware regions corresponding to Fig. 15. The unit is measured in cm.**

B. FURTHER DISCUSSION ON DISPARITY ANALYSIS

There is an FIS for performing the pixel classification within an estimated disparity image. The fuzzy sets strongly affect the performance of the proposed FIS, as shown in Figs. 8 and 9, and can be completely determined by the critical points $\{\alpha_1^1, \alpha_1^2, \ldots \alpha_1^K\}$, $\{\alpha_2^1, \alpha_2^2, \ldots, \alpha_2^K\}$, and $\{\gamma^1, \gamma^2, \ldots, \gamma^K\}$. Evidently, the first premise variable a_1 is always limited in $[0, d_{\max}]$, and thus we assume that $\{\alpha_1^1, \alpha_1^2, \ldots \alpha_1^K\}$ are uniformly spaced in $[0, \alpha_1^K]$.

$$\alpha_1^k = \frac{k-1}{K-1}\alpha_1^K, \text{ for } k = 1,2, \ldots, K, \text{ with } \alpha_1^K \leq d_{\max}. \quad (22)$$

Similarly, for the second premise variable a_2, the critical points are determined by

$$\alpha_2^k = \frac{k-1}{K-1}\alpha_2^K, \text{ for } k = 1,2, \ldots, K, \text{ with } \alpha_2^K \leq h. \quad (23)$$

The output variable g denotes the degree limited in $[0,1]$ of a considered pixel belonging to the road region. Hence, the same way is employed.

$$\gamma^k = \frac{k-1}{K-1}, \text{ for } k = 1,2, \ldots, K. \quad (24)$$

Therefore, the total number of parameters is significantly reduced because only the values of α_1^K, α_2^K, and K need to be determined later. To study the influence of parameters α_1^K, α_2^K, and K, we select ten representative images from the Yonsei's outdoor dataset and then analyze our proposed Algorithm 1 (Table I) with different combinations of α_1^K, α_2^K, and K. In this work, they are finally determined by trial-and-error analysis method as: $\alpha_1^K = 8$, $\alpha_2^K = 2h/3$, and $K = 5$.

After an input goes into an FIS, several steps, including fuzzification, firing rules, inferencing, and defuzzification, are needed to derive the final output. To reduce the computation time, the FIS is regarded to be transformed as an input-output mapping: $\tilde{g} = \text{func}(\tilde{a}_1, \tilde{a}_2)$. This mapping can be pre-determined and constructed in a lookup table for $\tilde{a}_1 = 0,1, \ldots, d_{\max}$, and $\tilde{a}_2 = 0,1, \ldots, h-1$. Fig. 14 shows the pre-computed mapping surface, in which the horizontal plane is the a_1 vs. a_2 plane, and the vertical axis is the crisp output of g. Each pixel in $\tilde{\mathcal{D}}^l$ is classified as road class if its output derived from the FIS is greater than a predefined threshold $T_{\text{road}} = 0.4$. For ease of observation, only the range $[0,20]$ of a_1 is plotted. Accordingly, the time cost of such an FIS is significantly reduced by replacing the inferencing process with a lookup table. Fig. 15 shows four representative results of the proposed pixel classification, where the left, middle, and right images are the original color images (selected from Yonsei's dataset [31]), the estimated disparity images, and the pixel classification results, respectively. The obstacles are indicated by yellow regions in the right figures.

C. DISPARITY TO OBSTACLE-AWARE REGIONS

Calculating the depth from a given disparity is straightforward for a binocular/stereo vision system with known intrinsic and extrinsic parameters. In this work, the model ψ is trained using Yonsei's and our own datasets, whose left-right pairs of training images are collected by ZED cameras. Given a left-view image, its corresponding disparity map is first predicted by our model ψ. Consequently, the obstacle-aware regions as mentioned in Subsection II-C are generated. Here, the parameters of the ZED camera are the focal length of the left camera $f_{\text{left}} = f_{\text{mono}} = 335$ (in pixels) and its baseline $b = 120$ (in millimeters). With these parameters, the depth (in millimeters) is easily calculated and thus the obstacle-aware regions with real distance information can be obtained. Fig. 16 shows the results of the obstacle-aware regions corresponding to Fig. 15, in which the red dashed lines represent the horizontal angle of view. Notably, the measurement unit in this figure is centimeters.

The second experiment in this subsection discusses the usage of another monocular camera, whose focal length differs from that of ZED and does not provide a baseline for using

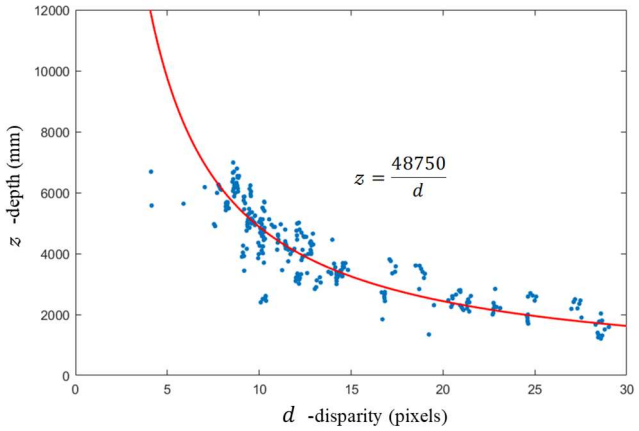

FIGURE 17. The mapping function from disparity to depth prediction when using the Logitech C930e webcam.

TABLE VI
ERROR COMPARISON OF METHODS WITH DIFFERENT ENCODERS

Method	Error (lower is better) Abs rel	Sqr rel	RMSE	RMSE-log
Monodepth-VGG	0.2435	2.3647	7.5629	0.6125
Monodepth- ResNet50	0.2318	1.9150	6.6315	0.4974
Ours-VGG	0.2105	2.2817	7.4109	0.5903
Ours-ResNet50	0.2005	1.7994	6.5820	0.4665

TABLE VII
ACCURACY COMPARISON OF METHODS WITH DIFFERENT ENCODERS

Method	Accuracy (higher is better) $\delta < 1.25$	$\delta < 1.25^2$	$\delta < 1.25^3$
Monodepth-VGG	0.5992	0.8010	0.8327
Monodepth- ResNet50	0.6084	0.8807	0.9130
Ours-VGG	0.7086	0.8072	0.8526
Ours-ResNet50	0.7305	0.8842	0.9214

stereoscopic vision. The depth cannot be directly calculated by a formula. Therefore, we need to obtain a mapping relation from disparity to true depth. In this experiment, we choose a low-cost webcam Logitech C930e to grab single-view images. For each capture going into our model ψ, a disparity image is predicted. Furthermore, we use a light detection and ranging (LiDAR) device to measure the true distance of an obstacle. By fusing the webcam and LiDAR, the estimated disparity and the real distance are acquired simultaneously. Fig. 17 presents data that are collected by such a joint camera-LiDAR device. In this plot, the disparity-to-depth mapping can be simply fitted as $z = 48750/d$, with a relatively high score $R^2 = 0.9963$. Experimental results thus demonstrate that the proposed method is feasible to a camera with different intrinsic parameters, if the process of camera-LiDAR calibration is performed first. The evaluation of the depth measurement will be given in Subsection IV-B.

IV. PERFORMANCE EVALUATION

A. ACCURACY OF DISPARITY ESTIMATION

We adopt several frequently-used metrics for quantitative evaluation of the proposed monocular disparity estimation. The first group of metrics contains the errors between the ground truth and the estimated disparities:

- Average of relative absolute error (Abs rel)

$$\text{Abs rel} = \frac{1}{N}\sum_{\mathbf{n}\in N}\left(\frac{\left|\mathcal{D}^{\text{GT}}[\mathbf{n}] - \widetilde{\mathcal{D}}[\mathbf{n}]\right|}{\mathcal{D}^{\text{GT}}[\mathbf{n}]}\right) \tag{25}$$

- Average of relative squared error (Sqr rel)

$$\text{Sqr rel} = \frac{1}{N}\sum_{\mathbf{n}\in N}\left(\frac{\left\|\mathcal{D}^{\text{GT}}[\mathbf{n}] - \widetilde{\mathcal{D}}[\mathbf{n}]\right\|^2}{\mathcal{D}^{\text{GT}}[\mathbf{n}]}\right) \tag{26}$$

- Root mean squared error (RMSE)

$$\text{RMSE} = \sqrt{\frac{1}{N}\sum_{\mathbf{n}\in N}\left\|\mathcal{D}^{\text{GT}}[\mathbf{n}] - \widetilde{\mathcal{D}}[\mathbf{n}]\right\|^2} \tag{27}$$

- $\log_{10}$ root mean squared error (RMSE-log)

$$\text{RMSE-log} = \sqrt{\frac{1}{N}\sum_{\mathbf{n}\in N}\left\|\log(\mathcal{D}^{\text{GT}}[\mathbf{n}]) - \log(\widetilde{\mathcal{D}}[\mathbf{n}])\right\|^2} \tag{28}$$

where $\mathcal{D}^{\text{GT}}$ and $\widetilde{\mathcal{D}}$ denote the ground truth and estimated disparity, N is the set of pixels (u, v) with $\mathcal{D}^{\text{GT}}(u, v) \geq 3$ on the disparity ground truth (discarding the faraway pixels), $\mathbf{n}$ indicates any pixel in N, and $\#(N)$ denotes the number of pixels in N.

The second group is accuracy-based metrics with three different thresholds, defined as

$$\delta = \max\left(\frac{\mathcal{D}^{\text{GT}}[\mathbf{n}]}{\widetilde{\mathcal{D}}[\mathbf{n}]}, \frac{\widetilde{\mathcal{D}}[\mathbf{n}]}{\mathcal{D}^{\text{GT}}[\mathbf{n}]}\right) < \text{Threshold}, \tag{29}$$

where Threshold $= 1.25, 1.25^2$, and 1.25^3. The evaluation is conducted using the sample data of Yonsei's outdoor dataset, including 500 pairs of stereo images and their disparity ground truth. Table VI summarizes the quantitative results of the first-group metrics, and Table VII does so for the second group. The reason for choosing the *Monodepth* method [17] in this experiment is that it is one of the state-of-the-art methods for learning with stereo images, and motivates a number of subsequent works recently. In addition to the standard model plotted in Fig. 2, we implement another lightweight encoder-decoder architecture, in which the encoder is replaced by the common VGG deep network. Observed from Tables VI and VII, encoding by ResNet-50 performs slightly better than VGG. Fig. 18 presents an example of the improvement gained by our method, in which (a) is done by Monodepth and (b) is by ours. Obviously, the outer appearance produced by our method is much clear. In this experiment, we used the SGM to obtain a reference disparity ground truth and computed the

TABLE VIII
COMPARISON OF DISPARITY ESTIMATION ERROR BY STATISTICAL ANALYSIS

Methods	Mean of Er	Standard deviation of Er
Monodepth- ResNet50	1.643	3.017
Ours- ResNet50	0.831	1.738

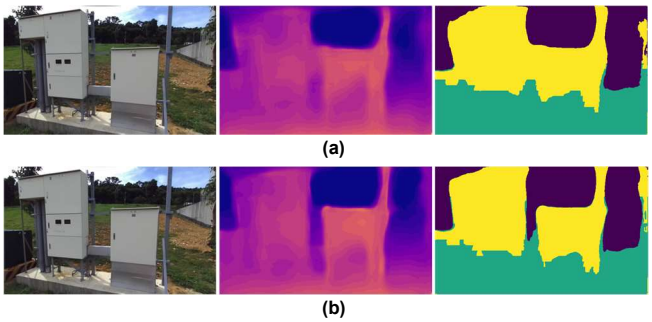

FIGURE 18. The quality comparison of disparity estimation: (a) results of Monodepth, and (b) results of ours.

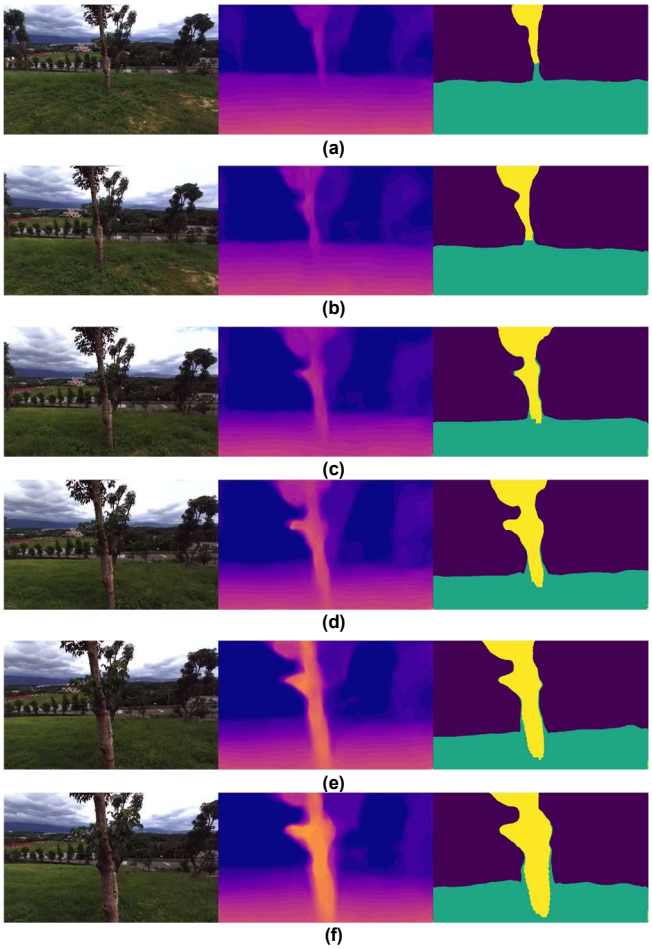

FIGURE 19. The results of disparity estimation and pixel classification at different distances.

estimation error as follows. Here, the absolute error is defined by

$$Er(u,v) = \left|\mathcal{D}^{\mathrm{GT}}(u,v) - \widetilde{\mathcal{D}}(u,v)\right|, \tag{30}$$

where $1 \le u \le w$, and $1 \le v \le h$. Table VIII lists the mean and standard deviation of this absolute error Er for the testing set.

B. ACCURACY OF MEASURING AN OBSTACLE

As mentioned before, there are two different situations, including the cases of (1) using the left camera of ZED device as a monocular camera, and (2) using another monocular camera, described so far. Generally, for the first case, it is straightforward to obtain the depth from disparity because the baseline and focal length are already known. The accuracy of depth measurement completely depends on the performance of the model ψ, discussed in the last subsection. Consequently, the following content focuses on the second situation, which uses another monocular camera for obstacle detection. We conducted an experiment that uses a Logitech C930e webcam to grab images of a tree, the obstacle-aware region, and its relevant distance range can be obtained by our method. Notably, the transformation from disparity to depth follows the fitted curve in Fig. 17. Besides, the true distance of the tree is measured by a laser range finder. Fig. 19 presents the results of disparity estimation and analysis of single-view images captured at different distances ranging from 150 cm to 500 cm. The left, middle and right parts are the original color image, the disparity estimation, and the disparity classification results, respectively. The resulting obstacle-aware regions are additionally generated in Fig. 20, in which each subplot is relevant to a distance measured in centimeters.

Table IX tabulates the effectiveness of the proposed obstacle detection and measurement. The average absolute error percentage ranges from $3\%-9.2\%$, which we deem acceptable to navigate our multicopter for collision avoidance. Note that some amount of error is unavoidable. The real distance is measured using a laser range finder aiming at the center of the obstacle. However, the depth of any obstacle should vary in a range since its outer appearance contains a surface structure. Therefore, there will always be differences between modeled outcomes and those measured as described.

C. DISCUSSION OF COMPUTATION TIME

The proposed method contains three main steps, including the monocular disparity estimation, the pixel classification and the generation of obstacle-aware regions. In this work, the whole algorithm is tested on a CUDA-accelerated computer with MS Windows 10 installed, equipping Intel i7-8750 2.2G processor, GeForce GTX 1060 GPU and DRAM DDR4 16G. The testing images are grabbed using a USB webcam and are sized of 640×480 pixels. Table X lists the average execution time. By accumulating the computation time of each step, the framerate is about 8 to 10 frames per second (FPS) which we deem acceptable to guide a multicopter in general applications.

TABLE IX
DISTANCE MEASUREMENT ON AN OBSTACLE: TREE

Subplot in Fig. 20	Real distance (cm)	Estimated distance (cm)	Absolute error percentage
(a)	500	447-503	1% – 11%
(b)	450	402-447	1% – 11%
(c)	350	335-365	4% – 4%
(d)	250	251-268	0% – 7%
(e)	200	191-212	5% – 6%
(f)	150	161-174	7% – 16%
		Average relative error	3% – 9.2%

TABLE X
TEST ON COMPUTATION TIME

Process	Average time (ms)
Monocular disparity estimation	≈ 28-35
Pixel classification	≈ 30-40
Generation of obstacle-aware regions	≈ 35-45

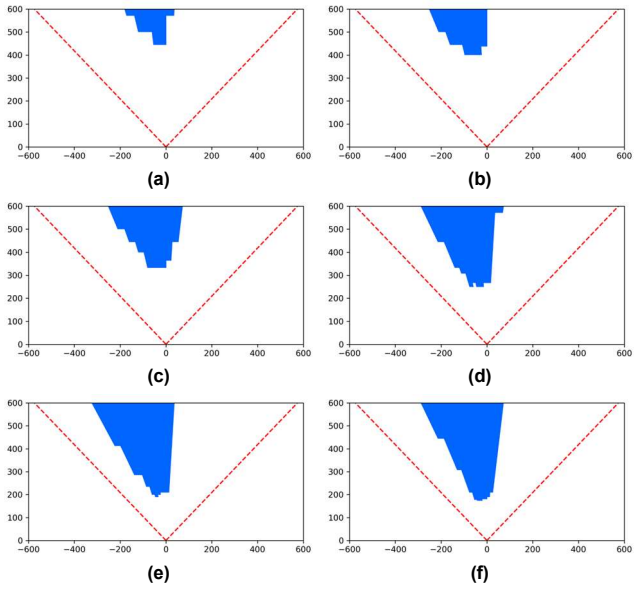

FIGURE 20. The results of obstacle-aware region generation separately from the subplots of Fig. 19.

V. APPLICATION: GUIDING A MULTICOPTER

Since the obstacles have been identified via the presence of obstacle-aware regions, the next is to find a safe path for guiding a multicopter for collision avoidance. We realized a robotic platform that includes a hexacopter with a webcam, a microcomputer, and a flight controller. The webcam is stabilized by a camera gimbal to stay horizontally forward. In addition, a laptop is used for establishing the ground control station (QCS) for flight data monitoring, and it is also for runnning the proposed algorithm.

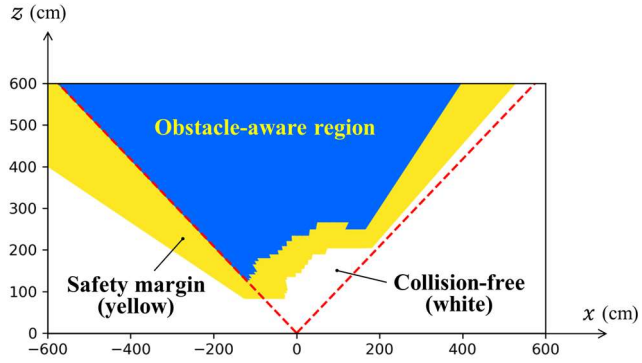

FIGURE 21. The obstacle-aware region (blue), safety margin (yellow), and flying direction (red arrow).

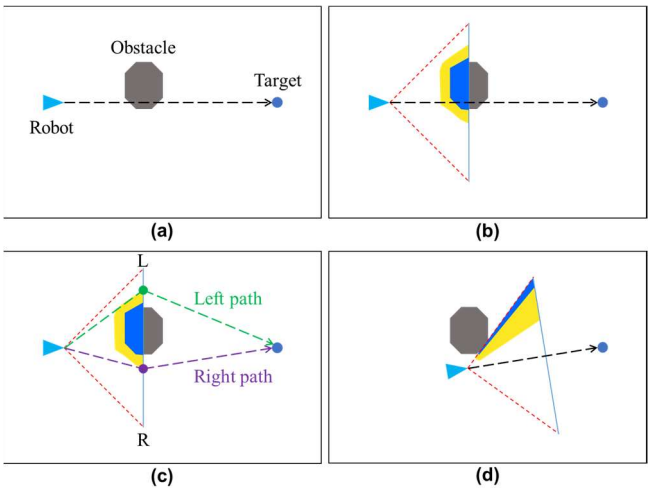

FIGURE 22. The steps of finding a new path for avoiding an obstacle.

A simple but reliable method of employing the obstacle-aware regions for finding a new flying path is introduced below. First, the obstacle-aware regions are dilated to be collision-free zone by considering the body width of the hexacopter. The four corners of each obstacle trapezoid are updated as follows.

New front-left point: $\left(x_1^\lambda, \tilde{z}^\lambda\right) \leftarrow \left(x_1^\lambda - w_r, \tilde{z}^\lambda - w_r\right)$

New front-right point: $\left(x_2^\lambda, \tilde{z}^\lambda\right) \leftarrow \left(x_2^\lambda + w_r, \tilde{z}^\lambda - w_r\right)$

New left-rear point: $\left(\frac{z_\mathrm{f}}{\tilde{z}^\lambda \times x_1^\lambda}, z_\mathrm{f}\right) \leftarrow \left(\frac{z_\mathrm{f}}{(\tilde{z}^\lambda - w_r) \times (x_1^\lambda - w_r)}, z_\mathrm{f}\right)$

New right-rear point: $\left(\frac{z_\mathrm{f}}{\tilde{z}^\lambda \times x_2^\lambda}, z_\mathrm{f}\right) \leftarrow \left(\frac{z_\mathrm{f}}{(\tilde{z}^\lambda - w_r) \times (x_2^\lambda + w_r)}, z_\mathrm{f}\right)$

where w_r is the half of hexacopter's body width.

Fig. 21 illustrates the dilated obstacle-aware region (the union of the blue and yellow regions), and the remaining part (white) in the field of visual perception is regarded as a collision-free zone. In our work, using a multirotor robotic platform, we are able to suggest a computation-efficient algorithm to select a new path for avoiding obstacles, while

meeting the real-time requirement. The key idea is to add an intermediate waypoint which locates on the boundary of collision-free zone and safety margin, at the furthest depth z_f. The main steps of adding a new waypoint and finding a flyable path are briefly explained below and are plotted in Fig. 22.

1) Subplot (a) shows the initial state when an obstacle blocks off the moving trajectory towards the target point.
2) Subplot (b) shows the moment when an obstacle is detected by our proposed method.
3) Scan along the side LR, referring to subplot (c), and find the boundary points of safety margin (yellow) and collision-free region (white). In this example, there are two points, plotted as green and purple circles, found on the side LR.
4) Considering the relative positions of the robot, the target point, and two intermediate points, there are two paths simply connected. Subplot (c) shows the green and purple dashed lines that indicate the left and right paths bypassing the obstacle.
5) Select the shortest path among all generated path candidates.
6) The multicopter moves to the intermediate waypoint on the selected path, and then front faces to the target point by turning. Subplot (d) shows the state when the robot completes this step.
7) Repeat the above Steps 1) to 6) until the robot safely reaches the target point.

If there is no collision-free zone in the field of visual perception, the robot is supposed to rotate in hopes of finding a collision-free zone allowing approach to the target point. In this experiment, we used the function of telemetry logs for recording the entire journal in QCS. Observed from the visualized flight trajectory, our hexacopter successfully passed through obstacles and reached the destination.

VI. CONCLUSION

In this paper, we propose a working algorithm for detecting obstacles and identifying obstacle-free regions that can be then used for guiding a multicopter. The main achievements of our method include the new learning scheme for monocular disparity estimation, the pixel classification for finding obstacles in a disparity image, and the generation of obstacle-free regions. The resulting obstacle-free regions of our algorithm are then applied to draw a collision-free zone for finding a safe path. The effectiveness of our proposed method was evaluated in several experiments by comparing qualitative and quantitative results. The whole algorithm was deployed on a hexacopter platform and successfully tested in an unknown outdoor environment with obstacles and a target.

REFERENCES

[1] R. Carloni *et al.*, "Robot vision: Obstacle-avoidance techniques for unmanned aerial vehicles," *IEEE Robot. Autom. Mag.*, vol. 20, no. 4, pp. 22–31, Dec. 2013, DOI. 10.1109/MRA.2013.2283632
[2] A. Al-Kaff *et al.*, "Obstacle detection and avoidance system based on monocular camera and size expansion algorithm for UAVs," *Sensors*, vol. 17, no. 5, May 2017, DOI. 10.3390/s17051061
[3] H. Alvarez *et al.*, "Collision avoidance for quadrotors with a monocular camera," in *Experimental Robotics*, Springer Tracts in Advanced Robotics, 2016, vol. 109, pp. 195–209. [Online]. Available: https://doi.org/10.1007/978-3-319-23778-7_14
[4] D. J. Lee *et al.*, "Two-frame structure from motion using optical flow probability distributions for unmanned air vehicle obstacle avoidance," *Mach. Vision and Appl.*, vol. 21, no. 3, pp. 229–249, Apr. 2010, DOI. 10.1007/s00138-008-0148-9
[5] W. E. Green and P. Y. Oh, "Optic-flow-based collision avoidance," *IEEE Robot. Autom. Mag.*, vol. 15, no. 1, pp. 96–103, Mar. 2008, DOI. 10.1109/MRA.2008.919023
[6] P. C. Merrell, D. J. Lee, and R. W. Beard, "Obstacle avoidance for unmanned air vehicles using optical flow probability distributions," in *Mobile Robots XVII*, Philadelphia, PA, USA, 2004, pp. 13–22.
[7] D. W. Yoo, D. Y. Won, and M. J. Tahk, "Optical flow based collision avoidance of multi-rotor UAVs in urban environments," *Int. J. Aeronaut. Space.*, vol. 12, no. 3, pp. 252–259, 2011.
[8] S. Zingg, D. Scaramuzza, S. Weiss, and R. Siegwart, "MAV navigation through indoor corridors using optical flow," in *2010 IEEE Int. Conf. on Robot. and Autom.*, Anchorage, AK, USA, 2010, pp. 3361-3368.
[9] F. Kendoul, "Survey of advances in guidance, navigation, and control of unmanned rotorcraft systems," *J. Field Robot.*, vol. 29, no. 2, pp. 315-378, Feb. 2012, DOI. 10.1002/rob.20414
[10] D. Scharstein and R. Szeliski, "A taxonomy and evaluation of dense two-frame stereo correspondence algorithms," *Int. J. Comput. Vision*, vol. 47, no. 1-3, pp. 7-42, Apr. 2002.
[11] R. Ranftl *et al.*, "Dense monocular depth estimation in complex dynamic scenes," in *IEEE Conf. on Comput. Vision and Pattern Recogn.*, Las Vegas, NV, USA, 2016, pp. 4058-4066.
[12] M. Pizzoli, C. Forster, and D. Scaramuzza, "REMODE: Probabilistic, monocular dense reconstruction in real time," in *IEEE Int. Conf. Robot. Autom.*, Hong Kong, China, 2014, pp. 2609-2616.
[13] J. Engel, T. Schöps, and D. Cremers, "LSD-SLAM: Large-scale direct monocular SLAM," *European Conf. on Comput. Vision*, Zurich, Switzerland, 2014, pp. 834-849.
[14] R. A. Newcombe, S. J. Lovegrove, and A. J. Davison, "DTAM: Dense tracking and mapping in real-time," *2011 Int. Conf. on Comput. Vision*, Barcelona, Spain, 2011, pp. 2320-2327.
[15] D. Eigen, C. Puhrsch, and R. Fergus, "Depth map prediction from a single image using a multi-scale deep network," arXiv:1406.2283v1 [cs.CV], Jun. 2014.
[16] F. Liu, C. Shen, G. Lin, and I. Reid, "Learning depth from single monocular images using deep convolutional neural fields," *IEEE T. Pattern Anal. Mach. Intell.*, vol. 38, no. 10, pp. 2024–2039, Oct. 2016.
[17] C. Godard, O. M. Aodha, and G. J. Brostow, "Unsupervised monocular depth estimation with left-right consistency," in *IEEE Conf. on Comput. Vision and Pattern Recogn.*, Honolulu, Hawaii, USA, 2017, pp. 270-279.
[18] M. Jaderberg, K. Simonyan, and A. Zisserman, "Spatial transformer networks," in *Advances in Neural Information Processing Systems*, Montreal, Canada, 2015, pp. 2017-2025.
[19] M. Poggi, F. Tosi, and S. Mattoccia, "Learning monocular depth estimation with unsupervised trinocular assumptions," in *2018 Int. Conf. on 3D Vision*, Verona, Italy, 2018, pp. 324-333.
[20] V. Casser *et al.*, "Unsupervised monocular depth and ego-motion learning with structure and semantics," in *IEEE Conf. on Comput. Vision and Pattern Recogn. Workshops*, Long Beach, CA, USA, 2019, to be published.
[21] P. Y. Chen *et al.*, "Towards scene understanding: unsupervised monocular depth estimation with semantic-aware representation," in *IEEE Conf. on Comput. Vision and Pattern Recogn.*, Long Beach, CA, USA, 2019, to be published.
[22] T. Zhou *et al.*, "Unsupervised learning of depth and ego-motion from video," in *IEEE Conf. on Comput. Vision and Pattern Recogn.*, Honolulu, Hawaii, USA, 2017, pp. 1851-1858.
[23] R. Mahjourian, M. Wicke, and A. Angelova, "Unsupervised learning of depth and ego-motion from monocular video using 3d geometric constraints," in *IEEE Conf. on Comput. Vision and Pattern Recogn.*, Salt Lake City, UT, USA, 2018, pp. 5667-5675.
[24] C. Godard *et al.*, "Digging into self-supervised monocular depth estimation," arXiv:1806.01260 [cs.CV], Jun. 2018.

[25] H. Zhao, O. Gallo, I. Frosio, and J. Kautz, "Loss functions for image restoration with neural networks," *IEEE T. Comput. Imag.*, vol. 3, no. 1, pp. 47-57, Mar. 2017, DOI. 10.1109/TCI.2016.2644865

[26] P. Heise, S. Klose, B. Jensen, and A. Knoll, "Pm-huber: Patchmatch with huber regularization for stereo matching," in *IEEE Int. Conf. on Comput. Vision*, Sydney, Australia, 2013, pp. 2360-2367.

[27] H. Hirschmuller, "Stereo processing by semiglobal matching and mutual information," *IEEE T. Pattern Anal. Mach. Intell.*, vol. 30, no. 2, pp. 328-341, Feb. 2008, DOI. 10.1109/TPAMI.2007.1166

[28] R. Achanta *et al.*, "SLIC superpixels compared to state-of-the-art superpixel methods," *IEEE T. Pattern Anal. Mach. Intell.*, vol. 34, no. 11, pp. 2274-2282, May 2012, DOI. 10.1109/TPAMI.2012.120

[29] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *IEEE Conf. on Comput. Vision and Pattern Recogn.*, Las Vegas, NV, USA, 2016, pp. 770-778.

[30] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," arXiv: 1409.1556 [cs.CV], Sep. 2014.

[31] Y. Kim, H. Jung, D. Min, and K. Sohn, "Deep monocular depth estimation via integration of global and local predictions," *IEEE T. on Image Process.*, vol.27, no. 8, pp. 4131-4143, Aug. 2018, DOI. 10.1109/TIP.2018.2836318

[32] R. Labayrade, D. Aubert, and J. P. Tarel, "Real time obstacle detection in stereovision on non flat road geometry through v-disparity representation," in *Intell. Vehicle Symposium*, Versailles, France, 2002, pp. 646-651.

[33] H. J. Zimmermann, "Fuzzy set theory and its applications," Springer Science & Business Media, 2011.

[34] D.P. Kingma and J. Ba, "Adam: A method for stochastic optimization," arXiv:1412.6980 [cs.LG], Dec. 2014.

Hsiang-Chieh Chen (M'16) received his B.S. degree in engineering science from National Cheng Kung University, Tainan, Taiwan, in 2002 and his M.S. and Ph.D. degrees in electrical engineering from National Central University, Taoyuan, Taiwan, in 2005 and 2009. From Aug. 2010 to Sep. 2014, he was with the Industrial Technology Research Institute, where he worked on intelligent video analytics and energy saving. From Sep. 2014 to Mar. 2015, he was with the Altek Semiconductor and focused on stereo vision R&D. From Mar. 2015 to Jul. 2016, he served as a technical leader of computer vision with the Coretronic Corporation and was in charge of innovative technology development. Since Aug. 2016, he joined the faculty of the Department of Electrical Engineering, National United University, Miaoli, Taiwan, where he is currently an assistant professor. His research interests include computer vision, computational intelligence, robotics, and automated optical inspection (AOI).