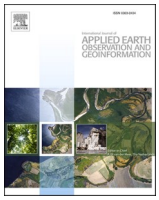
International Journal of Applied Earth Observations and Geoinformation 108 (2022) 102739

Contents lists available at ScienceDirect

International Journal of Applied Earth Observations and Geoinformation

journal homepage: www.elsevier.com/locate/jag



A novel region-based expansion rate obstacle detection method for MAVs using a fisheye camera

Samira Badrloo [a,b], Masood Varshosaz [b], Saied Pirasteh [a,c,*], Jonathan Li [c]

[a] Department of Surveying and Geoinformatics, Faculty of Geosciences and Environmental Engineering, Southwest Jiaotong University, Chengdu 611756, China
[b] Department of Photogrammetry and Remote Sensing, K.N. Toosi University of Technology, Tehran 19697, Iran
[c] Geospatial Sensing and Data Intelligence Lab, Department of Geography and Environmental Management, University of Waterloo, Waterloo, ON N2L 3G1, Canada

ARTICLE INFO

Keywords:
MAVs
Obstacle detection
Expansion-based method
Fisheye image
Obstacle region

ABSTRACT

Expansion-based methods are among the fastest algorithms to detect obstacles for safe navigation of Micro-Aerial Vehicles (MAVs). These methods are based on estimating an enlarging rate which is mostly computed using point features. Using points alone may result in situations where obstacles are only partially identified. This paper presents a new technique that uses image regions, instead of points, for estimating the expansion rate. The proposed algorithm utilises a fisheye camera that can be installed in front of a drone to detect obstacles in all directions. The camera takes images on which obstacles are identified. At each point in time, we extract the regions on the latest fisheye image and check to determine whether or not its regions belong to an obstacle. This step is completed by matching region points with those within the previous image. If at least three points are matched, then the convex hulls of the matched points on both images are formed. The expansion ratio of the convex hull areas is then estimated. If this ratio is bigger than a certain threshold, the region on the latest image is determined as an obstacle; otherwise, it is disregarded. This process is repeated until all pixels of the image are labelled either as obstacle or non-obstacle. Experiments were carried out using 50 pairs of fisheye images that covered a variety of obstacles, including people, pillars, trees, walls, and so on. The findings showed between 74%, and 84% of pixels were labelled correctly. By comparing these results with those obtained by the method developed by Al-Kaff et al. (2017), it is clear that the proposed method produces more stable and accurate results.

1. Introduction

Due to their small size, light weight, Micro-Aerial Vehicles (MAVs) are ideal platforms for indoor and outdoor applications (Bi et al., 2019; Chataigner et al., 2020; Gao et al., 2020). However, obstacles, including buildings, trees, people, and other structures, may cause a MAVs to crash. The usual step in detecting an obstacle is to build a three-dimensional map of the area (McGuire et al., 2017; Pestana et al., 2019) which might be difficult, due to MAVs' limited processing and energy storage capabilities (McGuire et al., 2017; Barry et al., 2018).

Obstacle detection techniques can be classified into sensor-based (Giannì et al., 2017) and vision-based (Zeng et al., 2016; Al-Kaff et al., 2017; Häne et al., 2017; Barry et al., 2018). In sensor-based techniques, active sensors such as a laser (Díaz-Vilariño et al., 2016; Giannì et al., 2017; Li et al., 2019), radar (Giannì et al., 2017), sonar (Qin et al., 2021), ultrasonic (Singh and Kapoor, 2021), and Kinect (Kucukyildiz et al., 2017) can be used. Active sensors are not ideal for MAVs having limited weight and cost (Gao et al., 2020) or those with high energy consumption (Zahran et al., 2018), and sensitivity to weather conditions (Lee et al., 2016).

In contrast to active sensors, cameras (as passive sensors) provide a lot of information about the environment (Figorito and Tarantino, 2014; Yin et al., 2021). Vision-based methods use cameras to detect obstacles using grayscale values (Mashaly et al., 2016), point features (Al-Kaff et al., 2017), and edge details (Huh et al., 2008; Padhy et al., 2019). They can be divided into monocular (Al-Kaff et al., 2017; Badrloo and Varshosaz, 2017) and stereo (Huang et al., 2015; McGuire et al., 2017; Barry et al., 2018). In stereo methods, two cameras are used to capture stereo images in real-time to generate the three-dimensional map of the surroundings, then used for obstacle detection. This approach has been

* Corresponding author at: Department of Surveying and Geoinformatics, Faculty of Geosciences and Environmental Engineering, Southwest Jiaotong University, Chengdu 611756, China.
E-mail addresses: badrloo.samira@email.kntu.ac.ir (S. Badrloo), varshosazm@kntu.ac.ir (M. Varshosaz), sapirasteh@swjtu.edu.cn (S. Pirasteh), junli@uwaterloo.ca (J. Li).

https://doi.org/10.1016/j.jag.2022.102739
Received 24 October 2021; Received in revised form 13 February 2022; Accepted 5 March 2022
Available online 17 March 2022

vastly employed to guide visually impaired people (Simões et al., 2020), vehicles (Jung et al., 2007; Kim et al., 2015), Unmanned Aerial Vehicles (UAVs) (Huang et al., 2015; Tijmons et al., 2017; McGuire et al., 2017; Barry et al., 2018), and MAVs (McGuire et al., 2017; Pestana et al., 2019). Unfortunately, such methods are not computationally cost-effective for MAV microprocessors (McGuire et al., 2017; Barry et al., 2018). To resolve this issue, a powerful Graphics Processing Unit (GPU) is needed (Aguilar et al., 2017). Also, since a method of such relies mostly on accurate system calibration, a tiny error could potentially affect the system's stability over time (Barry et al., 2018).

On the other hand, monocular vision-based obstacle detection methods make use of a single camera. They detect obstacles using images taken by a camera mounted in front or at the sides of an unmanned vehicle (Zeng et al., 2016; Al-Kaff et al., 2017; de Croon and De Wagter, 2018; Lin et al., 2018; de Croon et al., 2021). Monocular techniques are divided into four categories (Singh and Kaur, 2017): appearance-based (Lee et al., 2016), motion-based (Gharani and Karimi, 2017; Ho et al., 2018), depth-based (Häne et al., 2017; Lin et al., 2018), and expansion-based (Al-Kaff et al., 2017; Badrloo and Varshosaz, 2017). Appearance-based methods only involve the appearance properties of objects and, thus, usually struggle in outdoor environments with complex objects present (Lee et al., 2016; Mashaly et al., 2016). Similarly, motion-based techniques cannot usually detect obstacles directly in front of the camera (Al-Kaff et al., 2017). Depth-based algorithms try to produce a complete three-dimensional map (Häne et al., 2017) of objects through unnecessary yet, costly computations.

Overall, expansion-based methods are among the simplest and fastest algorithms that use an object's enlarging rate (across consecutive frames) (Badrloo and Varshosaz, 2017) to determine obstacles. These methods follow a similar concept used by human eyes, where geometric information, including scale (Mori and Scherer, 2013) and convex hull area (Al-Kaff et al., 2017) of objects are used to detect the obstacles. They do not have a detailed map of objects. Unlike motion-based methods, they can detect frontal obstacles and, thus, can be more appropriate for the safe navigation of MAVs.

Most expansion-based approaches rely on detecting object points for obstacle detection (Mori and Scherer, 2013; Al-Kaff et al., 2017; Badrloo and Varshosaz, 2017; Häne et al., 2017). However, using points alone may not be as efficient as needed; thus, the MAV may crash into undetected obstacles. To resolve this problem, we present a new method that works based on the same concept but using region-enlarging rates (across multiple images). We use a fisheye camera that can be mounted in front of a drone to cover large portions of the surrounding areas. As will be shown, the presented approach outperforms the current techniques and produces better and more reliable results.

The main contributions of this study are summarized in

(a) Developing a new expansion-based method: instead of points, our technique uses the expansion rate of regions across successive images to accurately and reliably detect obstacles. Thus, compared to similar approaches, we obtain more complete and accurate results.
(b) Implementation of the suggested approach on fisheye images: the implementation of our algorithm on fisheye images allows for larger portions of the surroundings to be simultaneously investigated for obstacles. Fisheye images present a lot of distortions and follow a different calibration model than frame-based images, and we had to introduce a number of additional steps in the proposed obstacle detection technique.

The rest of this paper is organised as follows. **Section 2** reviews the related work about monocular obstacle detection techniques. **Section 3** describes the proposed method for detecting obstacles. In **Section 4**, experiments are presented to evaluate the proposed method and compare its results with those obtained with one of the best point-based techniques developed by Al-Kaff et al. (2017). Finally, the paper is concluded in **Section 5**, where recommendations and suggestions for future works are presented.

2. Related work

Numerous techniques fall within the mono obstacle detection category (Huh et al., 2008; Lee et al., 2016; Mashaly et al., 2016; de Croon and De Wagter, 2018). As our technique also falls within this category, we discuss the relevant literature in this section. Such methods can be divided into four types (Singh and Kaur, 2017): appearance-based (Lee et al., 2016), motion-based (Gharani and Karimi, 2017), depth-based (Häne et al., 2017; Lin et al., 2018), and expansion-based (Al-Kaff et al., 2017) and are discussed in more detail.

2.1. Appearance-based

An obstacle is defined as a foreground object against a uniformly-coloured background thus it can be identified using edge (Huh et al., 2015), colour (Mashaly et al., 2016), texture (Ulrich and Nourbakhsh, 2000), and object shape (Lee et al., 2016) information. Ulrich and Nourbakhsh (2000) classified each pixel in an image as an obstacle or terrain based on its grayscale value. Lee et al. (2016) used Markov random field segmentation to detect obstacles in indoor environments. Huh et al. (2015) defined a horizon line to separate the sky from the ground. To extract moving obstacles, they used a particle filter algorithm. In a similar study, Mashaly et al. (2016) created a binary image to separate obstacles from the sky. De Croon and De Wagter (2018) used a self-supervised learning method to extract the horizon line.

Appearance-based methods are generally limited to situations in which obstacles can easily be distinguished from the background. Unfortunately, this assumption is frequently violated, especially in complex environments with objects of varying shape and colour, such as buildings, trees, and humans (Zeng et al., 2016).

2.2. Motion-based

In these methods, depth is calculated using motion information. Several studies have been conducted in this area (Gharani and Karimi, 2017; Tsai et al., 2018). Optical flow is a type of data used in most motion-based approaches (Ho et al., 2018; Cho et al., 2019). Gharani and Karimi (2017) created an obstacle detection system that measures optical flow on two consecutive images and chooses some points for obstacle detection. The optical flow is proportional to insect flight patterns. Bees, for example, never travel in a straight line but rather in a zigzag pattern. The zigzag motion produces motion vectors that can be used to calculate the depth and detect obstacles. Therefore, detecting frontal obstacles using optical flow in direct movements is difficult (de Croon et al., 2021; Urban and Caplier, 2021).

2.3. Depth-based

To detect nearby obstacles, in depth-based methods, the depth information is obtained from a single image. It is used to obtain a complete three-dimensional map of the environment (de Croon and De Wagter, 2018; Gao et al., 2020; Silva et al., 2020; de Croon et al., 2021). Motion stereo is used in some of these methods (Häne et al., 2017; Lin et al., 2018), which can be acquired using fisheye cameras (Häne et al.; 2017). Lin et al. (2018) navigated a MAV using a fisheye camera and an Inertial Measurement Unit (IMU). They used key frames and motion stereo to calculate depth.

Recently, researchers have estimated depth using artificial neural networks (Kumar et al., 2018; Mancini et al., 2018). For example, Kumar et al. (2018) used four fisheye cameras and a Convolutional Neural Network (CNN) to estimate the depth around a car. CNNs learn various characteristics of the desired features during a training step, for which the more data used, the more accurate results obtained. They heavily

rely on the availability of a large number of high-quality images (Lee et al., 2021), which may not always be easy to obtain, particularly in complex environments.

Due to the need for a three-dimensional map of the environment, depth-based methods are computationally intensive (Al-Kaff et al., 2017; Silva et al., 2020). Thus, unless a powerful GPU is used, this can be a problem, especially with MAVs (Aguilar et al., 2017).

2.4. Expansion-based

Expansion-based employs a concept similar to human perception, i.e. the enlargement rate of the objects between successive images. Several studies have been conducted using expansion-based methods (Mori and Scherer, 2013; Zeng et al., 2016; Al-Kaff et al., 2017; Badrloo and Varshosaz, 2017; Häne et al., 2017; Escobar-Alvarez et al., 2018; Padhy et al., 2019; Lee et al., 2021). Different enlargement rate estimation criteria have been used. For example, Mori and Scherer (2013) detected obstacles using point scale variations extracted from Speeded-Up Robust Features (SURF). To detect the obstacles, Zeng et al. (2016) used the expansion of an object boundary. In a more recent study, Al-Kaff et al. (2017) used the scale changes of points extracted using the Scale-Invariant Feature Transform (SIFT) algorithm and their convex hull area ratio. In contrast to other algorithms, which only provide obstacle points (Mori and Scherer, 2013; Badrloo and Varshosaz, 2017; Häne et al., 2017), they use the convex hull of points as the obstacle region. Furthermore, Badrloo and Varshosaz (2017) proposed a method for detecting obstacles based on distance ratios, while Aguilar et al. (2017) used a SURF template of obstacles and points. Padhy et al. (2019) used

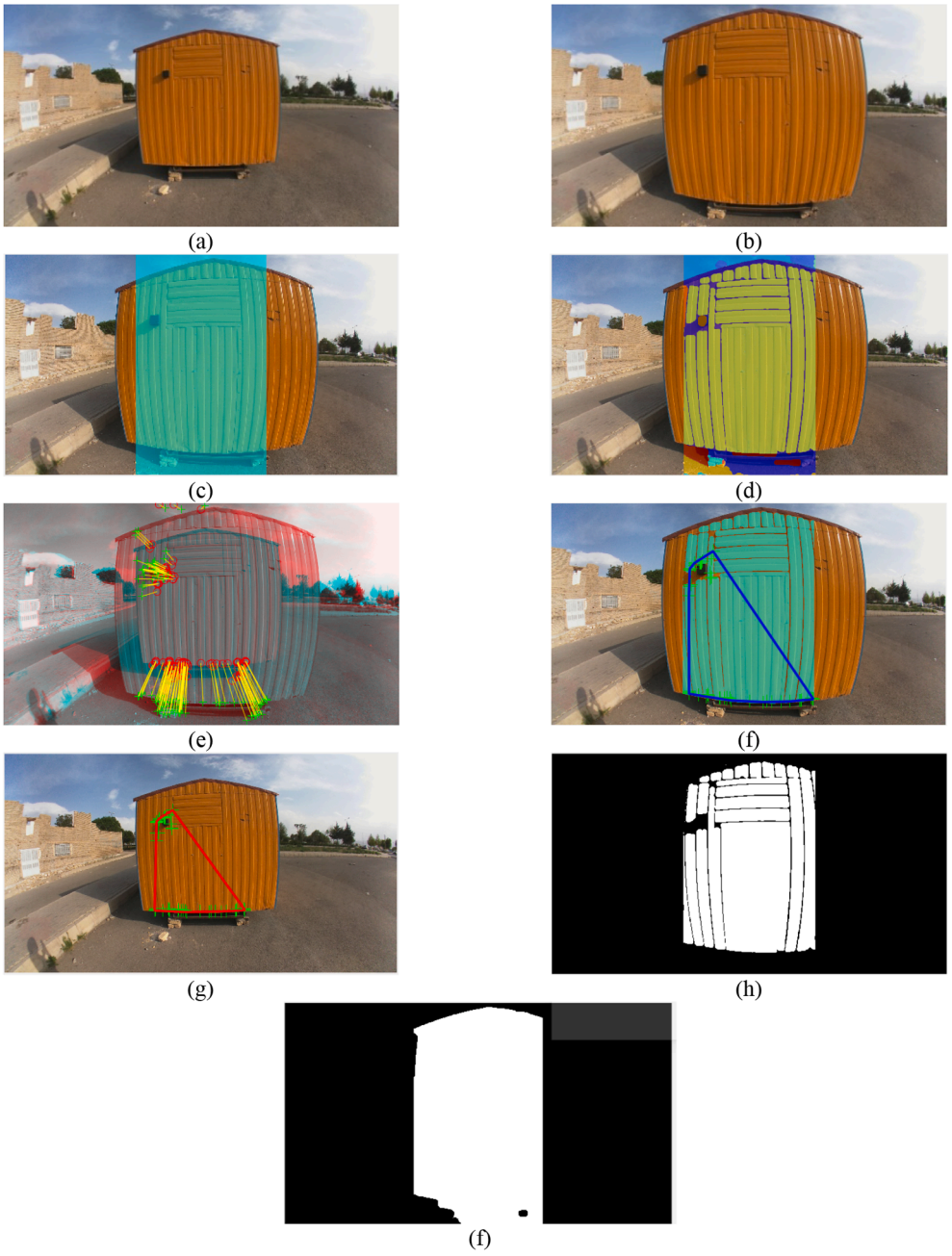


Fig. 1. Obstacle detection process. (a) The first image. (b) The second image. (c) The selected part in the second image. (d) Extracted regions in the selected part. (e) Matched points. (f) Convex hull of the points of a region in the second image. (g) Convex hull of the corresponding points in the first image. (h) The region's binary obstacle image. (f) The final binary obstacle image.

the Euclidean distance ratio of matched points to detect obstacles. Later, Lee et al. (2021) identified an obstacle tree by calculating the ratio of its length over image to the relevant image dimension.

Compared to the previous three methods, expansion-based are preferred as they do not require a complete map of the surroundings. Furthermore, because they employ point or distance features, they can be used to detect a wide range of objects (Badrloo and Varshosaz, 2017; Lee et al., 2021). Nevertheless, there might not be enough points for to completely identify an object in many cases. A door or a wall, for example, may have a featureless smooth texture, making the feature algorithm fail. Thus, the resulting obstacle points may be incomplete or contain gaps. Thus the main goal of this paper is to present a new technique that employs a number of criteria/approaches to make sure the obstacle detection is as complete as possible. Also, MAVs may collide with objects from all directions. Using more cameras can be considered as an immediate solution. However, as they are small, this may not be possible (Gao et al., 2020). As a result, the proposed algorithm employs a fisheye camera that can be installed in front of a drone, allowing detection to occur in all directions. as will be shown, we have implemented our algorithm on fisheye images so obstacles can be identified in all directions

3. Methodology

In this section, various steps of our obstacle detection technique are described. We use a fisheye camera to take the images. Before the images are captured, the fisheye camera is calibrated in the laboratory (Section 3.1). Then, as shown in Fig. 1, at any time during the drone's flight, a pair of consecutive images are taken and used to detect the obstacles (Fig. 1 (a) and Fig. 1 (b)). We call them the first and second images. Then, the second image is divided into three left, middle, and right parts. Based on the drone's flight direction, one of the parts (Fig. 1 (c)) is selected for obstacle detection. If the drone moves forward, the middle part is used; otherwise, one of the other two is chosen accordingly. Then, the regions within the selected part are extracted (Fig. 1 (d)). In the meantime, the pixels of the selected part is matched with those in the first image (Fig. 1 (e)). At this stage, the convex hulls of the points detected in each region are determined in both the first and second images (Fig. 1 (f) and Fig. 1 (g)). The ratio of the two convex hull areas is then estimated. If it is bigger than a certain threshold, the region is regarded as an obstacle, which is saved as a part of the final obstacle binary image (Fig. 1 (h)). The above process is repeated for all of the regions until the final binary obstacle image is completed (Fig. 1 (f)).

The steps described above can be summarised in (a) data acquisition and preparation, (b) region extraction and matching region points, and (c) obstacle detection. These are shown in Fig. 2 and are detailed in the following sections.

3.1. Data acquisition and preparation

Fisheye camera images have a significant amount of distortion (especially on the sides). Thus, it affects the precision of the measurements. As a result, calibration parameters must be accurately computed and applied to the image coordinates before any calculations are performed (Liang et al., 2021). In this study, the calibration was performed once in the laboratory, using a series of images from a chessboard. Sample images are shown in Fig. 3.

There are many calibration models specifically proposed for fisheye cameras (Ricolfe-Viala and Sánchez-Salmerón, 2010; Urban et al., 2015; Choi et al., 2019; Jarron et al., 2019; Ji et al., 2020). Scaramuzza and Ikeuchi (2014) presented a model that can be applied to both catadioptric and fisheye cameras in a unified framework. It can model a fisheye camera with very large Fields of View (FoV) of up to 195° (Scaramuzza and Ikeuchi, 2014); as a result, this model is used in this study. The camera parameters can be calculated by:

$$\lambda.\begin{bmatrix} u'' \\ v'' \\ w'' \end{bmatrix} = \lambda.g(\alpha.U') = \lambda.\begin{bmatrix} (\alpha.u') \\ (\alpha.v') \\ f(\alpha.\rho') \end{bmatrix} = \lambda.\alpha.\begin{bmatrix} (u') \\ (v') \\ a_0 + \cdots + a_N \rho'^N \end{bmatrix} = P.X\lambda, \alpha > 0 \quad (1)$$

As shown in Fig. 4 (b) and Fig. 4 (c), the variables *(u ', v')* and *(u", v")* in Eq. (1) represent the image of the ground point X on the image plane and the sensor plane, respectively. λ and α denote the scale parameter, and ρ' is the Euclidean distance between the point and the centre of the image. Additionally, $X \in R^4$ is expressed as homogeneous coordinates, and $P \in R^{3\times4}$as perspective projection matrix. The variable α can be combined with λ, and in general, the coefficients a_i, $i = 0, 1, 2, \ldots, N$ corresponds to the same number of $N + 1$ polynomial parameters that must be specified in the calibration step.

Once the calibration is carried out, the next step is to take the images. Essentially, the images should be captured using a camera mounted on a real drone and in a real flight. This requires time and resources that, unfortunately, in this research, the authors did not have. Thus, in this paper, the image acquisition is carried out using a camera. However, they are taken in exactly the same manner as those taken in real-time applications.

For this, we need a pair of sequential images at each point of time (Fig. 1 (a) and Fig. 1 (b)). To give MAV enough time to interact safely with the surroundings, we process consecutive frames, the time interval of which is equal to:

$$T = 1/fbs \quad (2)$$

3.2. Extracting regions and matching region points

To detect the obstacles, the second fisheye image is divided into three horizontally equal sections: left, middle, and right (Fig. 5 (b)). To detect the obstacles in the proposed approach, we use the expansion rate of homologous regions across a pair of successive frames, which we call the first and second images.

Homologous regions can be identified by detecting an object region in the first image and tracing it forward in the second image. However, this requires precise extraction and matching of the object boundaries, which is a challenging and difficult task, especially in complex environments (Chen et al., 2020). To resolve the problem, the regions are initially identified in the second image to ensure they have counterparts in the first image (as the time interval between the frames is short). To identify the regions in the second image, the Seeded Region Growing (SRG) algorithm (Asmussen et al., 2015) is used, with image pixels taken as seed points. This process is repeated until all pixels inside a specific region are assigned to it. At this stage, any region smaller than a given threshold is disregarded (see Section 4.1). Also, we fill in the gaps within the regions using a closing morphological operation (Said et al., 2016) by:

$$Closing = (IM \oplus SE) \ominus SE \quad (3)$$

where $\oplus$ and $\ominus$ denote the dilation and erosion, respectively, SE is the Structural Elements, and IM stands for image. Fig. 1 (d) shows one of the images used in our experiments and its extracted regions.

In addition to delineating the regions, pixels of the selected part in the second image are matched with those in the first one. In this paper, since fisheye images contain a lot of distortions, we used the Affine-SIFT (ASIFT) algorithm (Yu and Morel, 2011) for matching. ASIFT is an improvement to the SIFT method (Lowe, 2004), and is invariant to the four parameters of the affine transform. To filter out the incorrect matches, we first correct their coordinates using the calibration parameters estimated before. Then, we use the fundamental matrix for an estimation of which LMedS (Rusiecki, 2012) algorithm is used due to its low sensitivity to incorrect match points.

There are situations where the number of matched points in a region

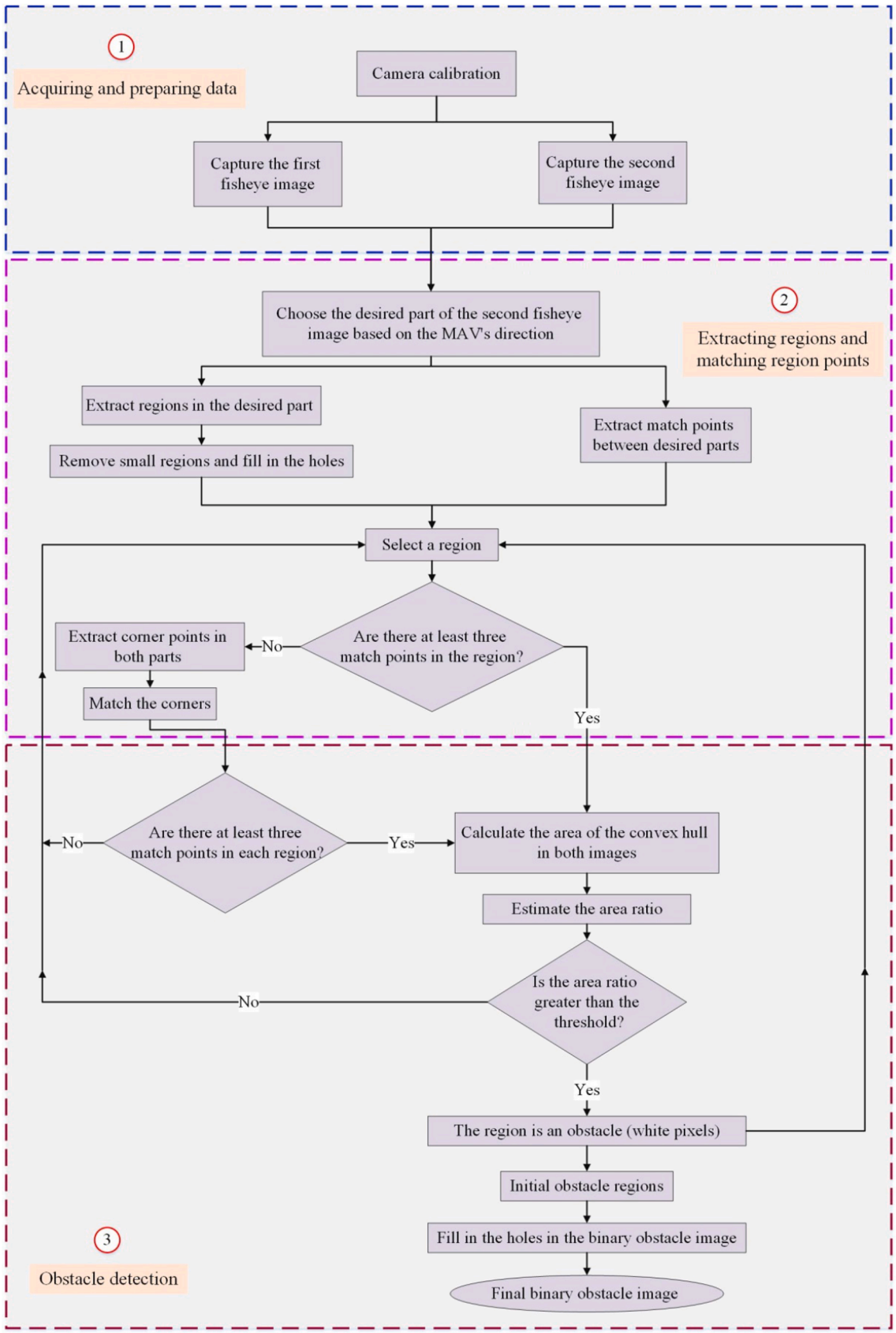


Fig. 2. The framework of the proposed method.

Fig. 3. Sample of the images for the fisheye camera calibration.

is insufficient (i.e. less than 3 non-collinear points). Thus, the region is searched for additional corner points using the Shi-Tomasi algorithm (Mu and Li, 2018), which after being matched in the first image, are added to the region points. It is hoped that now the region contains enough matched points. Otherwise, it is disregarded and considered as a non-obstacle object

3.3. Obstacle detection

Any region that contains at least three non-collinear points is considered as a potential obstacle. To determine this, the convex hull of its match points is generated in both images (Fig. 1 (f) and Fig. 1 (g)). The convex hull is regarded as an irregular polygon. As a result, the area of a given C as a convex hull can be determined as follows:

$$C_{area} = \frac{1}{2}\begin{vmatrix} x_1 & y_1 \\ x_2 & y_2 \\ x_3 & y_3 \\ \vdots & \vdots \\ x_n & y_n \\ x_1 & y_1 \end{vmatrix} = \frac{1}{2}[(x_1y_2 + x_2y_3 + x_3y_4 + \cdots + x_ny_1) - (y_1x_2 + y_2x_3 + y_3x_4 + \cdots + y_nx_1)] \quad (4)$$

where $\times_{(1:n)}$ and $y_{(1:n)}$ are vertices, and n is the number of sides of the polygon. To determine if the region should be considered as an obstacle, we use the following equation:

$$\text{Ratio}\ (C_{area}) = C_{2area}/C_{1area} \quad (5)$$

where C_{2area} and C_{1area} denote the convex hull areas in the second and first images, respectively. A region is considered to be an obstacle, if the ratio (C_{area}) exceeds a certain threshold. Assuming that a1, and a2 are the areas the convex hulls in the first image and in the second image, from Fig. 6, it can easily be shown that:

$$\frac{a_2}{a_1} = \left(\frac{H+h}{H}\right)^2 \quad (6)$$

where H is the distance between the object and the second image, and h is the spatial distance between the two images (Fig. 6). The minimum value for H, is called the reaction distance H_m which is the minimum distance an MAV needs to preserve to avoid obstacles. This means if the ratio of the areas is equal to or bigger than that obtained by using $H=H_m$. In practice, we may consider a larger value to be on the safe side.

The preceding procedure is repeated for all of the regions until all pixels in the second image are labelled as either obstacle or non-

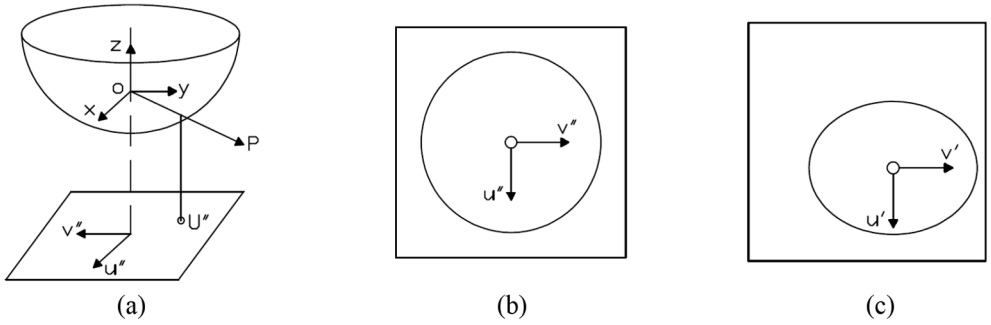


Fig. 4. Relationship between the points u' ,u'' and the vector P (which connects the image's centre O to the ground point X), (a) Catadioptric coordinate system, (b) sensor plane, and (c) image plane.

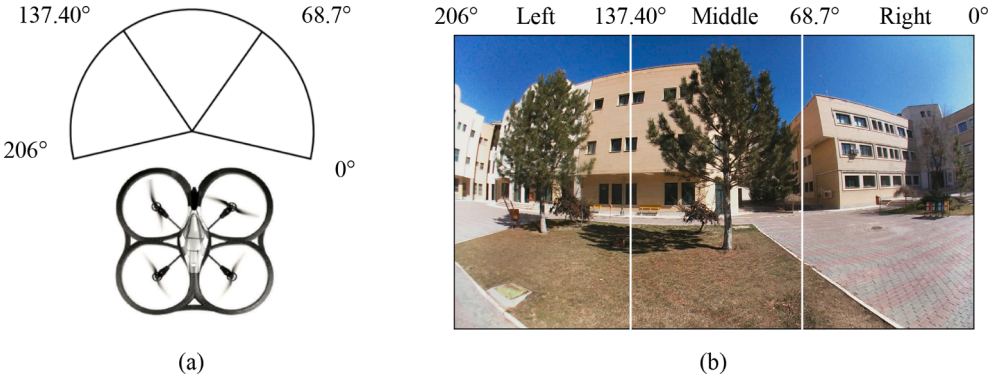


Fig. 5. Left, front, and right of the fisheye image. (a) FoV for each section. (b) Each section's image.

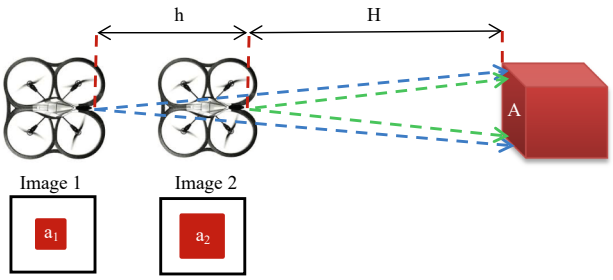


Fig. 6. Increasing the area of an object in consecutive images.

obstacle. The final output is a binary image with highlighted obstacle locations in white and non-obstacle regions in black. Following that, using the closing morphological algorithm, the holes in the obstacle image are closed (Said et al., 2016).

4. Results and discussion

In this section, the proposed algorithm for obstacle detection is evaluated. For this, a fisheye camera was moved in front and side directions in a relatively complex environment that included various objects to perform the experiments. In addition, the results were compared with those obtained by Al-Kaff et al. (2017). In each experiment, a binary image was produced on which obstacles and non-obstacles were presented by white and black pixels, respectively. For evaluations, the following terms were used:

TP: The number of pixels that are indeed obstacles and detected as obstacles.

FP: The number of pixels that are actually not obstacles but detected as obstacles.

FN: The number of pixels that are actually obstacles but are detected as non-obstacles.

TN: The number of pixels that are actually not obstacles and are detected as non-obstacles.

The above terms can be used to compute the recall, precision, and overall accuracy of the results (Hong and Oh, 2021) by the following equations.

$$Recall = \frac{TP}{TP + FN} \times 100\% \tag{7}$$

$$Precision = \frac{TP}{TP + FP} \times 100\% \tag{8}$$

$$Overallaccuracy = \frac{TP + TN}{TP + FP + FN + TN} \times 100\% \tag{9}$$

A low recall rate suggests obstacle detection is not carried out completely, which is risky for an MAV. The precision parameter measures the algorithm's accuracy in detecting non-obstacles. When precision is poor, non-obstacles are assumed to be obstacles. As a result, MAV's manoeuvrability is limited by low precision. In addition to the above parameters, the time spent on each experiment was also examined.

4.1. Experimental results

We used an LG 360 CAM fisheye camera to take fifty pairs of images, twenty-five pairs of which were taken with the camera moving forward (Fig. 7) and the other twenty-five were taken with the camera moving to the sides (Fig. 8). Images were analysed in pairs to compute the enlarging rate between sequential images. To define the distance between the images of each pair, we assumed that the drone's speed is 10 m/s, and the image acquisition is carried out at 30 fps. Thus, h is equal to 33 cm. Also, we assumed the drone's minimum reaction time is equal to 250 msec. Thus, H_m is equal to 2.5 m. As a result, regions having a convex hull ratio bigger than 1.28 must be considered as immediate obstacles. To be on the safe side, we used 1.20 as the minimum value (the threshold) that identifies a region as an obstacle. Fig. 7 and Fig. 8 show several samples, with the corresponding outputs at each step as discussed in Section 3. Fig. 7 shows samples for the forward motion images, while Fig. 8 presents samples of the images taken assuming the drone moves to the right.

The dimensions of the images were 1260 × 2560 pixels, while the camera's field of view was 206°. As discussed in Section 3.1, we used the Scaramuzza and Ikeuchi (2014) method for camera calibration that works for images of up to a maximum of 195°. Thus, in our experiments, the far left and right 6°s of the images have not been used. The Scaramuzza and Ikeuchi (2014) technique assumes an optimal Calibration yielded five parameters: a_0, a_1, a_2, a_3, and a_4 (Table 1). Calibration parameter values were derived using N = 4, which is used by Scaramuzza and Ikeuchi (2014) method,

The regions were extracted using a grayscale difference threshold of 10, with those smaller than 300 pixels disregarded. To fill in the holes, we determined that 20 was the best closing morphological parameter threshold value. To remove incorrect matches, we chose 2000 random points to calculate the fundamental matrix parameters. Also, Shi-Tomasi algorithm (Mu and Li, 2018) to detect additional corner points were where then matched using the Least Square Matching (LSM) algorithm using a 15 × 15 pixels window size and a threshold of 0.5.

Fig. 7 (e) and Fig. 8 (e) show the final results in which obstacles are highlighted in white and non-obstacles in black. The implementation results of each step for forwarding and right motion are illustrated in Table 2.

It is evident that the average number of extracted regions varies between 7 and 48, depending on the complexity of features in the image. Moreover, the third row of Table 2 shows that 64% of the regions had at least three ASIFT match points. This finding shows that 36% of the regions still do not have a corresponding point and cannot be used for obstacle detection. Therefore, the new extracted corners provide at least three match points for 18% of the regions. As a result, 18% of regions can also be used in obstacle detection by matching corner points, and only18% of the regions remain without corresponding points. For this reason, the seventh row of Table 2 shows that 18% of regions are not an obstacle due to a lack of match points. These regions are mostly on the right side of the image. Because of the low quality of the fisheye images on the sides, the number of matched points is reduced.

At this stage, we need to evaluate the results because it is necessary to compare the identified obstacles with the real ones. Therefore, a binary image of the obstacles was manually constructed. The pixels belonging to less than 2.5 m away were then manually identified and marked in white in the binary image (Fig. 7 (f) and Fig. 8 (f)). Finally, the percentages of recall, precision, and overall accuracy were calculated by comparing the overlap between the real binary images of the obstacle and the binary image created using the suggested method. Table 3 displays the average of evaluation results and computation time. Please note that the time was taken to run the algorithm in MATLAB software R2018b using a system with the following specifications:

- CPU: Intel (R) Core (TM) i7-8550U CPU @ 1.80 GHz
- RAM: 12 GB
- Graphics card: NVIDIA GeForce MX130

It is evident that the average recall accuracy for the people, wall, tree, pillar and other obstacles data sets are 98.5%, 77.4%, 83.6%, 78.6%, and 83.5%, respectively. Fig. 7 (e) shows the completeness of obstacle regions. It is observed that the wall and pillar data sets have lower average recall accuracy than the other four. For the people, wall, tree, pillar and other obstacles data sets, average precision accuracy is 83%, 79.4%, 63.4%, 60.8%, and 97%, respectively. Average precision in the tree data set and especially the pillar data set is not good. Finally, the algorithm's average overall accuracy is 91.4%, 77.4%, 77.6%, 78%, and

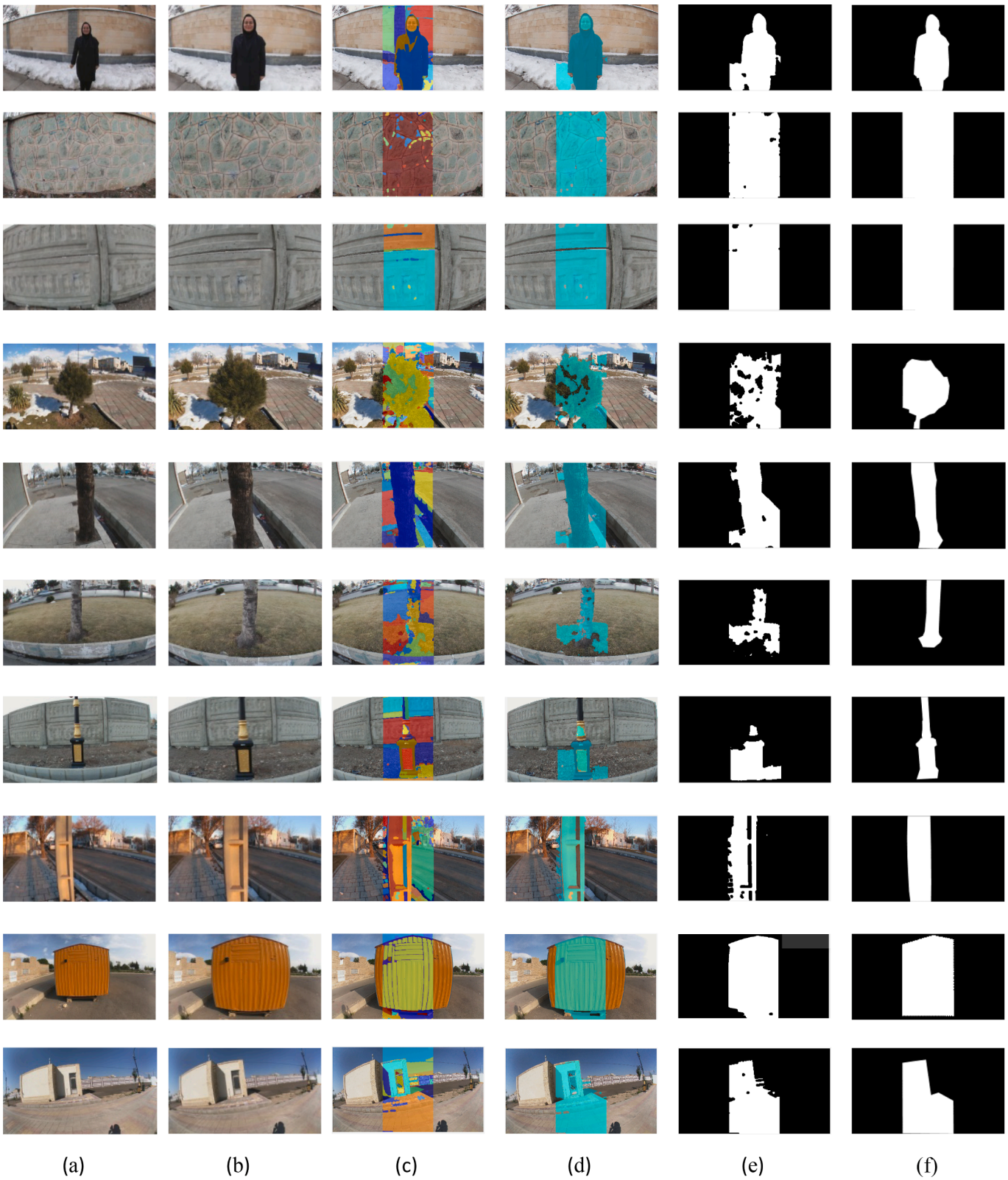


Fig. 7. Sample of the images for forward motion. (a) The first image. (b) The second image. (c) Extracted regions of the second image. (d) Obstacle regions obtained using the proposed method. (e) The final obstacle binary image after closing morphological algorithm. (f) The true binary image of the obstacles.

84% for the first twenty-five data sets.

For the data taken when moving to the right, average recall accuracy is 96.5%, 56.6%, 57%, 71.4%, and 90.5% for the people, wall, tree, pillar and other obstacles data sets, respectively. The wall and tree data sets have lower average recall accuracy. The people, wall, tree, pillar, and other obstacles data sets had average precision accuracy of 99%, 59.4%, 47%, 77.8%, and 91.5%, respectively. The tree data set's average precision accuracy is less than the other data sets. This finding is due to the grayscale similarity of trees and distant features. Finally, the people, wall, tree, pillar and other obstacles data sets exhibit 97.3%, 56.8%,

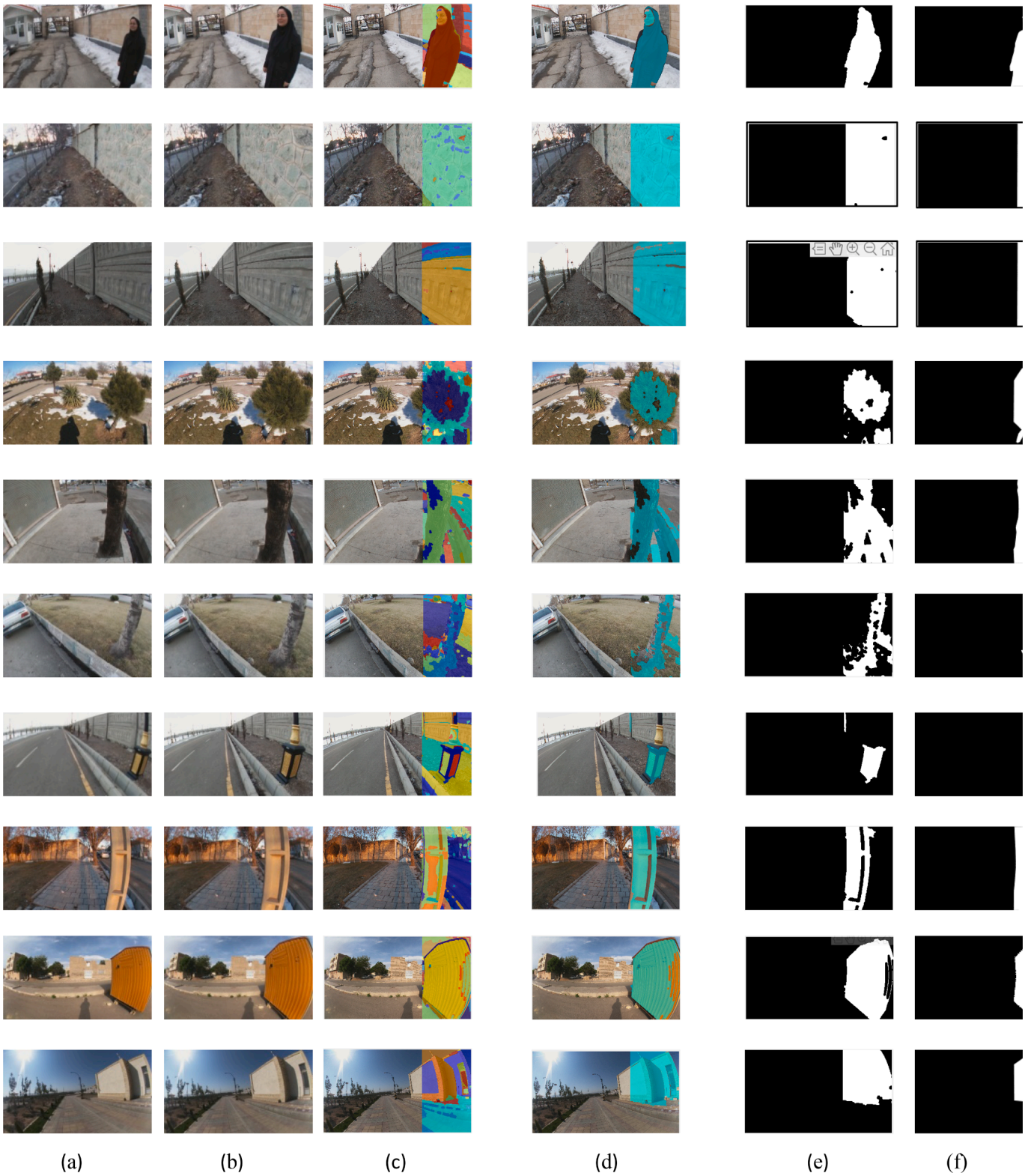


Fig. 8. Sample of the images for rightward motion. (a) The first image. (b) The second image. (c) Extracted regions of the second image. (d) Obstacle regions obtained using the proposed method. (e) The final obstacle binary image after closing morphological algorithm. (f) The true binary image of the obstacles.

56.6%, 78%, and 88.5% average overall accuracy. Overall, the results indicate that the proposed technique enables the identification of the obstacles with an average of 84.3% and 74.4% accuracy in forward and right motion mode, respectively.

4.2. Comparative study

In this section, the proposed method is compared with a similar algorithm. Al-Kaff et al. (2017) published a study on expansion-based methods. Unlike previous algorithms that only provide obstacle points (Mori and Scherer, 2013; Aguilar et al., 2017; Badrloo and Varshosaz,

Table 1
Calibration parameters.

Calibration Parameters	Values
Average reprojection error [pixels]	1.206645
a_0	−926.71402
a_1	100
a_2	0.0004200359242
a_3	−0.0000001970037
a_4	0.0000000001967

2017; Padhy et al., 2019), their methods considered the convex hull as an obstacle region in addition to obstacle points. Thus, similar to our method, their technique displays the obstacle as a region. For this reason, we compared our method with their technique.

In Al-Kaff et al.'s (2017) method, keypoints and the descriptors are extracted from two successive images using SIFT (Lowe, 2004). After detecting the keypoints and the descriptors, the Brute-Force algorithm with a distance-ratio threshold of 0.28 was applied to match the keypoints from the two images. Afterwards, the algorithm returns the matched keypoints if and only if its size is growing. Then, the size ratio of the matched keypoints from the second image to the first image was calculated by Eq. (10). In this equation, $\mathrm{mkp}_1(\mathrm{i})$ and $\mathrm{mkp}_2(\mathrm{i})$ denote the *i-point* scale sizes in the first and second images, respectively.

$$\mathrm{Ratio(mkp)} = \frac{1}{\mathrm{N}} \sum_{\mathrm{i}=1}^{\mathrm{N}} \frac{size(mkp_2(i))}{size(mkp_1(i))} \tag{10}$$

Furthermore, the area ratio of the convex hull from the second to the first images was calculated by applying Eq. (11). *Size (c1)* and *size (c2)* in Eq. (11) are also the convex hull areas of the first and second images, respectively.

$$\mathrm{Ratio(c)} = \frac{\mathrm{size}(c_2)}{\mathrm{size}(c_1)} \tag{11}$$

Finally, if the size ratio of the matched keypoints is greater than 1 and the area ratio of the convex hull is greater than 1.45, the convex hull points and regions are identified as obstacles. Thus, the obstacle is less than 2.5 m away from the MAV when the thresholds of 1 and 1.45 are used.

Fig. 9 depicts the comparison results. As shown in the figure, only a few points on the obstacle and a piece of the convex hull were detected by Al-Kaff et al. (2017) (Fig. 9 (a) and Fig. 9 (b)). Therefore, obstacle regions have not been completely detected. As shown in Table 4, the recall value of our algorithm is bigger than that of the available method in all of the data sets used in this research. As can be seen, on average, the proposed technique outperforms the Al-Kaff et al. (2017) approach by 60.7%, 2.5%, and 32.4% in forward motion and 57.6%, 56%, and 53.6% in rightward motion with respect to recall, precision, and overall accuracy metrics, respectively. This suggested that, although Al-Kaff approach (2017) may have been successful in working with frame images, our region-based approach produces more accurate and complete results when fisheye images are utilised for obstacle detection.

4.3. Discussion

The experiments in Section 4.1 analysed outcomes in two tests. In the first test, the evaluation was completed while the camera moved ahead. In the second test, evaluations were also completed in the rightward direction. The suggested algorithm was then compared to a currently available method (Section 4.2).

When moving forward (first test), the obstacle's regions are fully detected. However, the wall and pillar data sets had lower obstacle detection average recall accuracy than the others, as some obstacle parts were not detected completely. This problem appears to be due to a lack of match points in one area and the probability of inaccurate match points in other areas. Moreover, because obstacles and distant objects have similar grey levels, the SRG algorithm extracted them as a single area. As a result, the proposed method failed to detect the obstacle pixels correctly. Furthermore, an appropriate grayscale difference threshold variable for each image is required to extract the regions using the SRG algorithm. This means that setting the threshold to a fixed value may lead to the combination of near and far regions or the production of many tiny regions.

Also, the findings of the second test showed that the wall and tree data sets have lower average recall accuracy than the others. As previously stated, this error was caused by a lack of sufficient match points and the possibility of incorrect match points, particularly at the edges of

Table 2
Results obtained during the implementation of the proposed algorithm.

Parameters	Forward movement data					Right movement data				
	People	Wall	Tree	Pillar	Other Obstacles	People	Wall	Tree	Pillar	Other Obstacles
Average number of extracted regions	23	7	35	15	47	32	6	48	23	42
Average number of regions with at least 3 ASIFT points	17	4	22	11	38	21	3	29	14	24
Regions with at least three ASIFT points (%)	74	57	63	73	81	66	50	60	61	57
Average number of regions with at least three corner match points	5	1	9	3	7	8	1	8	3	5
Regions with at least three corner match points (%)	20	14	26	20	15	25	17	17	13	12
Average number of regions that do not have three match points	1	2	4	1	2	3	2	11	6	13
Regions that do not have three match points (%)	4	29	11	7	4	9	33	23	26	31

Table 3
The proposed obstacle detection method's results.

Movement direction	Parameters	Data					Total
		People	Wall	Tree	Pillar	Other Obstacles	
Forward	Recall (%)	98.5	77.4	83.6	78.6	83.5	84.3
	Precision (%)	83.0	79.4	63.4	60.8	97.0	76.7
	Overall accuracy (%)	91.4	77.4	77.6	78.0	84.0	81.7
	Time (s)	96	49	149	141	75	102
Rightward	Recall (%)	96.5	56.6	57.0	71.4	90.5	74.4
	Precision (%)	99	59.4	47.0	77.8	91.5	74.9
	Overall accuracy (%)	97.3	56.8	56.6	78.0	88.5	75.4
	Time (s)	134	33	113	124	77	96

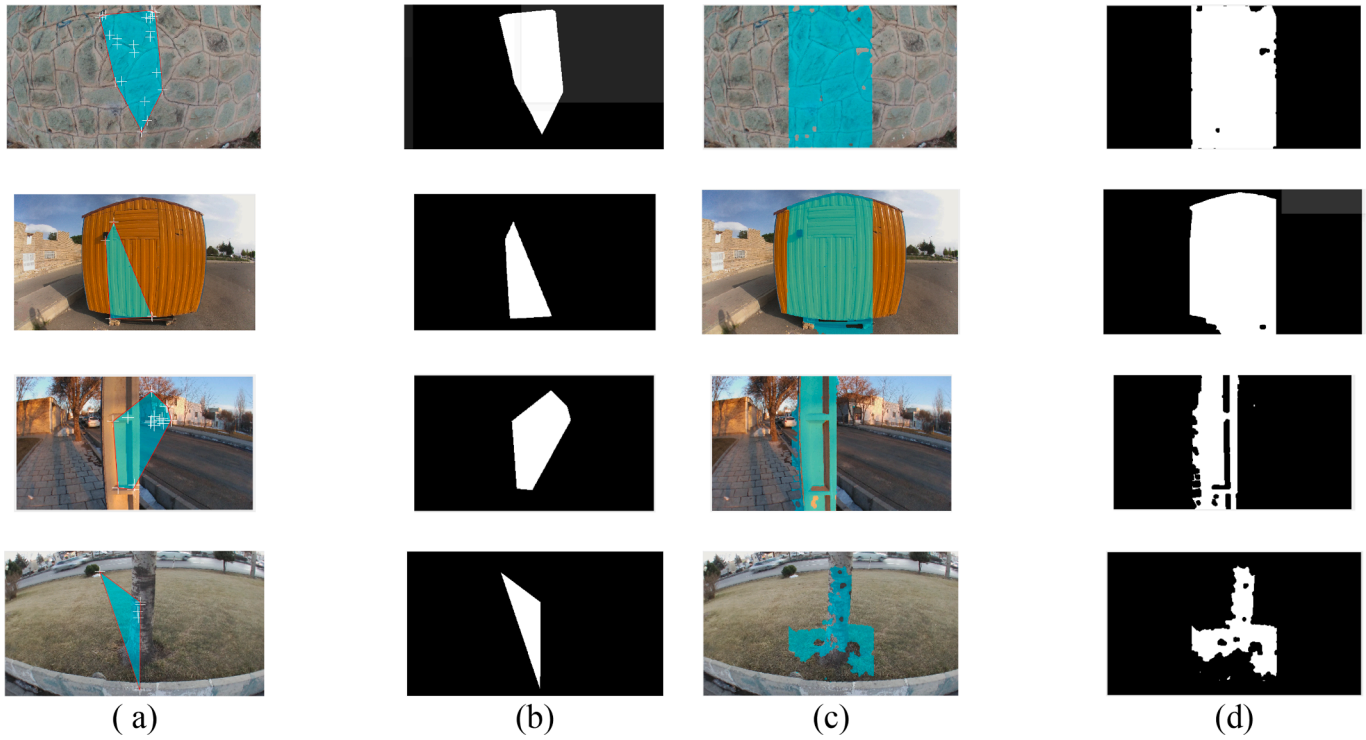


Fig. 9. Comparison of the proposed method's results to those of a currently available method (Al-Kaff et al., 2017). (a) The currently available method shows the results with white obstacle points and a red convex hull region. (b) The currently available method results as a binary image. (c) Obstacle regions obtained using our approach. (d) The final obstacle binary image in our approach. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

Table 4
Comparison of the proposed method's results to those of Al-Kaff et al. (2017).

Movement direction	Parameters	Proposed in this study	Al-Kaff et al.'s (2017)
Forward	Recall (%)	84.3	23.6
	Precision (%)	76.7	74.2
	Overall accuracy (%)	81.7	49.3
	Time (s)	102	65
Rightward	Recall (%)	74.4	16.8
	Precision (%)	74.9	18.9
	Overall accuracy (%)	75.4	21.8
	Time (s)	96	57

the fisheye image. Additionally, the comparison of our method to Al-Kaff et al. (2017) suggested that our technique has a better recall in all experiments. Due to the use of regions, the proposed method has a relatively large recall parameter.

Another issue examined was the computation time. Although the primary intention of this paper was to improve the accuracy of obstacle detection, we observed that running the proposed method took much longer than the intended 250 msec. Thus, the speed issue is still challenging and needs further improvement in future studies. In addition, the proposed method is suitable for rigid stationary objects. Due to the critical importance of detecting moving obstacles, future studies will generalize the proposed method for detecting moving obstacles.

5. Conclusion

Expansion-based obstacle detection techniques use an enlarging rate which is mostly calculated using point features. However, using points alone may result in situations where obstacles are only partially identified. Aiming at increasing the accuracy and completeness of obstacle detection, we proposed a novel approach that uses image regions to compute the expansion rate across successive images. In the proposed algorithm, a fisheye camera is used that can be installed at the front of a drone to detect obstacles in all directions. For this, regions on the most recent image are extracted and checked for matched points within the previous image. An object is defined as an obstacle if it contains at least three matched points. Various techniques were incorporated to ensure the maximum number of obstacles are identified and the minimum number of holes remain in the results.

The proposed method was evaluated in several experiments with images taken in forward and sideward directions. The results indicated that, the accuracy of the proposed technique in identifying the obstacles, ranged from 74% to 84%. Furthermore, in another experiment, our results were compared with those obtained by a strong point-based obstacle detection approach developed by Al-Kaff et al. (2017). It was observed that, on average, the proposed technique outperforms the Al-Kaff et al. (2017) approach by 60.7%, 2.5%, and 32.4% in forward motion and 57.6%, 56%, and 53.6% in rightward motion with respect to recall, precision, and overall accuracy metrics, respectively. This suggested that, although the Al-Kaff approach (2017) is successful in working with frame images, our region-based approach produces more accurate and complete results when fisheye images are utilised.

Despite offering promising results, our technique failed to detect some of the obstacles correctly when the obstacle colour was similar to those of distant objects. Therefore, it is suggested that future research focus on using semantic maps for obstacle detection. It was also observed; sometimes obstacles were not detected at all due to a lack of match points and/or the possibility of incorrect match points. Thus, we recommend utilising a matching method explicitly developed for fisheye images.

Another issue examined was the computation time. Although the primary intention of this paper was to improve the accuracy of obstacle

detection, we observed that running the proposed method took much longer than the intended 250 msec. Thus, the speed issue is still challenging and needs further improvement in future studies.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgements

The authors appreciate the GeoAI Smarter Map and LiDAR Lab of the Faculty of Geosciences and Environmental Engineering, Southwest Jiaotong University (SWJTU), technical research support. This work is the outcome of a joint research study between K.N. Toosi University of Technology (KNT), Iran and the University of Waterloo (UW), Canada. We thank KNT and UW for their active collaboration in this research.

Data and code availability

Data and code are available upon request.

References

Aguilar, W., Casaliglla, V., Pólit, J., 2017. Obstacle avoidance based-visual navigation for micro aerial vehicles. Electronics 6 (1), 10. https://doi.org/10.3390/electronics6010010.

Al-Kaff, A., García, F., Martín, D., de la Escalera, A., Armingol, J.M., 2017. Obstacle detection and avoidance system based on monocular camera and size expansion algorithm for AVs. Sensors 17 (5), 1061.

Asmussen, P., Conrad, O., Günther, A., Kirsch, M., Riller, U., 2015. Semi-automatic segmentation of petrographic thin section images using a "seeded-region growing algorithm" with an application to characterize wheatheredsubarkose sandstone. Comput. Geosci. 83, 89–99. https://doi.org/10.1016/j.cageo.2015.05.001.

Badrloo, S., Varshosaz, M., 2017. Monocular vision based obstacle detection. Earth Obs. Geomatics Eng. 1, 122–130. https://doi.org/10.22059/eoge.2017.244709.1015.

Barry, A.J., Florence, P.R., Tedrake, R., 2018. High-speed autonomous obstacle avoidance with pushbroom stereo. J. Field Rob. 35 (1), 52–68. https://doi.org/10.1002/rob.21741.

Bi, Y., Lan, M., Li, J., Lai, S., Chen, B.M., 2019. A lightweight autonomous MAV for indoor search and rescue. Asian J. Control 21 (4), 1732–1744. https://doi.org/10.1002/asjc.2162.

Chataigner, F., Cavestany, P., Soler, M., Rizzo, C., Gonzalez, J.-P., Bosch, C., Gibert, J., Torrente, A., Gomez, R., Serrano, D., 2020. ARSI: An aerial robot for sewer inspection. In: Grau, A., Morel, Y., Puig-Pey, A., Cecchi, F. (Eds.), Advances in robotics research: From lab to market. Springer Tracts in Advanced Robotics, vol. 132, pp. 249–274. https://doi.org/10.1007/978-3-030-22327-4_12.

Chen, Y.-C., Lin, Y.-Y., Yang, M.-H., Huang, J.-B., 2020. Show, match and segment: Joint weakly supervised learning of semantic matching and object co-segmentation. IEEE Trans. Pattern Anal. Mach. Intell. 43 (10), 3632–3647. https://doi.org/10.1109/TPAMI.2020.2985395.

Cho, G., Kim, J., Oh, H., 2019. Vision-based obstacle avoidance strategies for MAVs using optical flows in 3-D textured environments. Sensor 19 (11), 2523. https://doi.org/10.3390/s19112523.

Choi, K.H., Kim, Y., Kim, C., 2019. Analysis of fish-eye lens camera self-calibration. Sensors (Basel) 19, 1218. https://doi.org/10.3390/s19051218.

De Croon, G., De Wagter, C., 2018. Learning what is above and what is below: Horizon approach to monocular obstacle detection. Arxiv prepr. 08007. Arxiv1806.08007.

De Croon, G., De Wagter, C., Seidl, T., 2021. Enhancing optical-flow-based control by learning visual appearance cues for flying robots. Nature Mach. Intell. 3 (1), 33–41. https://doi.org/10.1038/s42256-020-00279-7.

Díaz-vilariño, L., Boguslawski, P., Khoshelham, K., Lorenzo, H., Mahdjoubi, L., 2016. Indoor navigation from point clouds: 3D modelling and obstacle detection. Int. Arch. Photogramm. Remote Sens. Spatial Inform. Sci. 41, 275–281. https://doi.org/10.5194/isprs-archives-xli-b4-275-2016.

Escobar-Alvarez, H.D., Johnson, N., Hebble, T., Klingebiel, K., Quintero, S.A.P., Regenstein, J., Browning, N.A., 2018. R-advance: Rapid adaptive prediction for vision-based autonomous navigation, control, and evasion. J. Field Rob. 35 (1), 91–100. https://doi.org/10.1002/rob.21744.

Figorito, B., Tarantino, E., 2014. Semi-automatic detection of linear archaeological traces from orthorectified aerial images. Int. J. Appl. Earth Obs. Geoinf. 26, 458–463. https://doi.org/10.1016/j.jag.2013.04.005.

Gao, W., Wang, K., Ding, W., Gao, F., Qin, T., Shen, S., 2020. Autonomous aerial robot using dual-fisheye cameras. J. Field Rob. 37 (4), 497–514. https://doi.org/10.1002/rob.21946.

Gharani, P., Karimi, H.A., 2017. Context-aware obstacle detection for navigation by visually impaired. Image Vis. Comput. 64, 103–115. https://doi.org/10.1016/j.imavis.2017.06.002.

Giannì, C., Balsi, M., Esposito, S., Fallavollita, P., 2017. Obstacle detection system involving fusion of multiple sensor technologies. Int. Arch. Photogram. Remote Sens. Spat. Inform. Sci. 42, 127–134. https://doi.org/10.5194/isprs-archives-xlii-2-w6-127-2017.

Häne, C., Heng, L., Lee, G.H., Fraundorfer, F., Furgale, P., Sattler, T., Pollefeys, M., 2017. 3D visual perception for self-driving cars using a multi-camera system: Dalibration, mapping, localization, and obstacle detection. Image Vis. Comput. 68, 14–27. https://doi.org/10.1016/j.imavis.2017.07.003.

Ho, H.W., De Wagter, C., Remes, B.D.W., de Croon, G., 2018. Optical-flow based self-supervised learning of obstacle appearance applied to MAV landing. Rob. Auton. Syst. 100, 78–94. https://doi.org/10.1016/j.robot.2017.10.004.

Hong, C.S., Oh, T.G., 2021. TPR-TNR plot for confusion matrix. Commun. Stat. Appl. Meth. 28 (2), 161–169.

Huang, H.-C., Hsieh, C.-T., Yeh, C.-H., 2015. An indoor obstacle detection system using depth information and region growth. Sensors 15 (10), 27116–27141. https://doi.org/10.3390/s151027116.

Huh, K., Park, J., Hwang, J., Hong, D., 2008. A stereo vision-based obstacle detection system in vehicles. Opt. Lasers Eng. 46 (2), 168–178. https://doi.org/10.1016/j.optlaseng.2007.08.002.

Huh, S., Cho, S., Jung, Y., Shim, D.H., 2015. Vision-based sense-and-avoid framework for unmanned aerial vehicles. IEEE Trans. Aerosp. Electron. Syst. 51 (4), 3427–3439. https://doi.org/10.1109/TAES.2015.140252.

Jarron, D., Shahbazi, M., Lichti, D., Radovanovic, R., 2019. Modelling wide-angle lens cameras for metrology and mapping applications. ISPRS Ann. Photogram. Remote Sens. Spat. Inform. Sci. 4, 79–86. https://doi.org/10.5194/isprs-annals-iv-2-w7-79-2019.

Ji, S., Qin, Z., Shan, J., Lu, M., 2020. Panoramic SLAM from a multiple fisheye camera rig. ISPRS J. Photogramm. Remote Sens. 159, 169–183. https://doi.org/10.1016/j.isprsjprs.2019.11.014.

Jung, H.G., Lee, Y.H., Kim, B.J., Yoon, P.J., Kim, J.H., 2007. Stereo vision-based forward obstacle detection. Int. J. Automot. Technol. 8 (4), 493–504.

Kim, D., Choi, J., Yoo, H., Yang, U., Sohn, K., 2015. Rear obstacle detection system with fisheye stereo camera using HCT. Expert Syst. Appl. 42 (17–18), 6295–6305. https://doi.org/10.1016/j.eswa.2015.04.035.

Kucukyildiz, G., Ocak, H., Karakaya, S., Sayli, O., 2017. Design and implementation of a multi sensor based brain computer interface for a robotic wheelchair. J. Intell. Rob. Syst. 87 (2), 247–263.

Kumar, V.R., Milz, S., Simon, M., Witt, C., Amende, K., Petzold, J., Yogamani, S., Pech, T., 2018. Monocular fisheye camera depth estimation using sparse lidar supervision. In: 21st International Conference on Intelligent Transportation Systems (ITSC), pp. 2853–2858. https://doi.org/10.1109/ITSC.2018.8569665.

Lee, H.Y., Ho, H.W., Zhou, Y., 2021. Deep learning-based monocular obstacle avoidance for unmanned aerial vehicle navigation in tree plantations. J. Intell. Rob. Syst. 101 (1), 1–18. https://doi.org/10.1007/s10846-020-01284-z.

Lee, T.-J., Yi, D.-H., Cho, D.-I., 2016. A monocular vision sensor-based obstacle detection algorithm for autonomous robots. Sensors 16 (3), 311. https://doi.org/10.3390/s16030311.

Li, F., Wang, H., Akwensi, P.H., Kang, Z., 2019. Construction of obstacle element map based on indoor scene recognition. Int. Arch. Photogram. Remote Sens. Spat. Inf. Sci. XLII-2/W13, 819–825. https://doi.org/10.5194/isprs-archives-XLII-2-W13-819-2019.

Liang, A., Li, Q., Chen, Z., Zhang, D., Zhu, J., Yu, J., Fang, X., 2021. Spherically optimized RANSAC aided by an IMU for fisheye image matching. Remote Sens. 13 (10), 2017. https://doi.org/10.3390/rs13102017.

Lin, Y., Gao, F., Qin, T., Gao, W., Liu, T., Wu, W., Yang, Z., Shen, S., 2018. Autonomous aerial navigation using monocular visual-inertial fusion. J. Field Rob. 35 (1), 23–51. https://doi.org/10.1002/rob.21732.

Lowe, D.G., 2004. Distinctive image features from scale-invariant keypoints. Int. J. Comput. Vision 60 (2), 91–110. https://doi.org/10.1023/B:VISI.0000029664.99615.94.

Mancini, M., Costante, G., Valigi, P., Ciarfuglia, T.A., 2018. J-MOD 2: joint monocular obstacle detection and depth estimation. IEEE Rob. Autom. Lett. 3 (3), 1490–1497. https://doi.org/10.1109/LRA.2018.2800083.

Mashaly, A.S., Wang, Y., Liu, Q., 2016. Efficient sky segmentation approach for small UAV autonomous obstacles avoidance in cluttered environment. In: In: 2016 IEEE International Geoscience and Remote Sensing Symposium (IGARSS), pp. 6710–6713. https://doi.org/10.1109/IGARSS.2016.7730752.

McGuire, K., de Croon, G., De Wagter, C., Tuyls, K., Kappen, H.J., 2017. Efficient optical flow and stereo vision for velocity estimation and obstacle avoidance on an autonomous pocket drone. IEEE Rob. Autom. Lett. 2 (2), 1070–1076. https://doi.org/10.1109/LRA.2017.2658940.

Mori, T., Scherer, S., 2013. First results in detecting and avoiding frontal obstacles from a monocular camera for micro unmanned aerial vehicles. In: 2013 IEEE International Conference on Robotics and Automation, pp. 1750–1757. https://doi.org/10.1109/ICRA.2013.6630807.

Mu, Z., Li, Z., 2018. A novel Shi-Tomasi corner detection algorithm based on progressive probabilistic hough transform. In: 2018 Chinese Automation Congress (CAC) IEEE, pp. 2918–2922. https://doi.org/10.1109/CAC.2018.8623648.

Padhy, R.P., Choudhury, S.K., Sa, P.K., Bakshi, S., 2019. Obstacle avoidance for unmanned aerial vehicles: Using visual features in unknown environments. IEEE Consum. Electron. Mag. 8 (3), 74–80. https://doi.org/10.1109/MCE.2019.2892280.

Pestana, J., Maurer, M., Muschick, D., Hofer, M., Fraundorfer, F., 2019. Overview obstacle maps for obstacle-aware navigation of autonomous drones. J. Field Rob. 36 (4), 734–762. https://doi.org/10.1002/rob.21863.

Qin, R., Zhao, X., Zhu, W., Yang, Q., He, B., Li, G., Yan, T., 2021. Multiple receptive field network (MRF-Net) for autonomous underwater vehicle fishing net detection using forward-looking sonar images. Sensors 21 (6), 1933. https://doi.org/10.3390/s21061933.

Ricolfe-Viala, C., Sánchez-Salmerón, A.-J., 2010. Robust metric calibration of non-linear camera lens distortion. Pattern Recogn. 43 (4), 1688–1699.

Rusiecki, A., 2012. Robust learning algorithm based on iterative least median of squares. Neural Process. Lett. 36 (2), 145–160. https://doi.org/10.1007/s11063-012-9227-z.

Said, K.A.M., Jambek, A.B., Sulaiman, N., 2016. A study of image processing using morphological opening and closing processes. Int. J. Control Theor. Appl. 9 (31), 15–21.

Scaramuzza, D., Ikeuchi, K., 2014. Omnidirectional Camera. Springer US, New York. https://doi.org/10.1007/978-0-387-31439-6.

Silva, A., Mendonça, R., Santana, P., 2020. Monocular trail detection and tracking aided by visual SLAM for small unmanned aerial vehicles. J. Intell. Rob. Syst. 97 (3), 531–551. https://doi.org/10.1007/s10846-019-01033-x.

Singh, B., Kapoor, M., 2021. A framework for the generation of obstacle data for the study of obstacle detection by ultrasonic sensors. IEEE Sens. J. 21 (7), 9475–9483. https://doi.org/10.1109/JSEN.2021.3055515.

Singh, Y., Kaur, L., 2017. Obstacle detection techniques in outdoor environment: Process, study and analysis. Int. J. Image Graph. Signal Process. 9 (5), 35–53. https://doi.org/10.5815/ijigsp.2017.05.05.

Simões, W.C.S.S., Silva, Y.M.L.R., Pio, J.L.d.S., Jazdi, N., de Lucena, V.F., 2020. Audio guide for visually impaired people based on combination of stereo vision and musical tones. Sensors 20 (1), 151. https://doi.org/10.3390/s20010151.

Tijmons, S., de Croon, G.C.H.E., Remes, B.D.W., De Wagter, C., Mulder, M., 2017. Obstacle avoidance strategy using onboard stereo vision on a flapping wing MAV. IEEE Trans. Rob. 33 (4), 858–874.

Tsai, C.-C., Chang, C.-W., Tao, C.-W., 2018. Vision-based obstacle detection for mobile robot in outdoor environment. J. Inform. Sci. Eng. 34 (1), 21–34.

Ulrich, I., Nourbakhsh, I., 2000. Appearance-based obstacle detection with monocular color vision. In: AAAI/IAAI, pp. 866–871.

Urban, D., Caplier, A., 2021. Time- and resource-efficient time-to-collision forecasting for indoor pedestrian obstacles avoidance. J. Imag. 7 (4), 61. https://doi.org/10.3390/jimaging7040061.

Urban, S., Leitloff, J., Hinz, S., 2015. Improved wide-angle, fisheye and omnidirectional camera calibration. ISPRS J. Photogramm. Remote Sens. 108, 72–79. https://doi.org/10.1016/j.isprsjprs.2015.06.005.

Yin, J., Dong, J., Hamm, N.A.S., Li, Z., Wang, J., Xing, H., Fu, P., 2021. Integrating remote sensing and geospatial big data for urban land use mapping: A review. Int. J. Appl. Earth Obs. Geoinf. 103, 102514. https://doi.org/10.1016/j.jag.2021.102514.

Yu, G., Morel, J.-M., 2011. ASIFT: An algorithm for fully affine invariant comparison. Image Process. Line 1, 11–38. https://doi.org/10.5201/ipol.2011.my-asift.

Zahran, S., Moussa, A.M., Sesay, A.B., El-Sheimy, N., 2018. A new velocity meter based on Hall effect sensors for UAV indoor navigation. IEEE Sens. J. 19 (8), 3067–3076. https://doi.org/10.1109/JSEN.2018.2890094.

Zeng, Y., Zhao, F., Wang, G., Zhang, L., Xu, B., 2016. Brain-inspired obstacle detection based on the biological visual pathway. In: International Conference on Brain and Health Informatics, pp. 355–364. https://doi.org/10.1007/978-3-319-47103-7_35.