

Vision-Based Obstacle Detection for Mobile Robot in Outdoor Environment^{*}

CHUNG-CHIH TSAI¹, CHIA-WEN CHANG² AND CHIN-WANG TAO¹

¹*Department of Electrical Engineering*

National Ilan University

Yi-lan, 260 Taiwan

E-mail: cwtao@niu.edu.tw

²*Department of Information and Telecommunications Engineering*

Ming Chuan University

Taipei, 111 Taiwan

The research in the robot vision becomes more and more attractive since the demand for robot is growing. To detect and avoid the obstacles in the outdoor environment is an important task for a moving robot. In this paper, we propose a real-time detection of the obstacle based on the computer vision with single camera. The dense optical flow method is adopted to extract the training data for a classifier model using support vector machine (SVM). The speeded-up robust features method is used to detect the interest points to be verified as the obstacle points or not by a SVM classifier. Moreover, a measurement of the spatial weighted saliency map is proposed to highlight the pixels of the obstacle. Finally, the obstacle points and the saliency map are combined to locate the region of the obstacle. The experimental results show that the proposed algorithm can effectively detect the obstacle in the outdoor environment.

Keywords: obstacle detection, optical flow, saliency, SURF, SVM

1. INTRODUCTION

Up to now, several kinds of the sensors were adopted to detect the obstacles in the natural environments, such as laser scanner [1], infrared detector [2], ultrasonic sensor [3], image sensor [45], etc. Besides, there are many image-based methods are proposed to detect the obstacles. It is found that to use the image or the video streams to detect the solid objects is more popular. Lueng *et al.* [6] used the stereo image sequences of the outdoor scene to evaluate several motion estimation methods. Bertozzi and Broggi [7] utilized a stereo vision-based method to detect the generic obstacle on the moving vehicles. The stereo vision-based algorithms are able to detect the obstacle in the unknown environment. However, more computing power is needed than those based on the monocular images. There are two major scenarios of the obstacle detection. One is that the background in an image sequence is fixed, and the other is that both foreground and background are dynamic. In the former situation, the candidate obstacle can be detected by subtracting two images in the sequence [8, 9]. When the camera is set up on the moving vehicle, the foreground is difficultly segmented. Consequently, the more complex algorithms are needed to be developed to detect the obstacle. Zhu [10] used the Hidden Markov Model to describe and predict the obstacle motion. Pangyu and Nedevschi proposed an obstacle detection method on the depth images [11]. Kruger *et al.* [12] adopted

Received June 21, 2016; revised September 20, 2016; accepted October 25, 2016.

Communicated by Chia-Feng Juang.

* This work was supported by the Ministry of Science and Technology (MOST) of Taiwan under Contract MOST 102-2221-E-197-024-MY3.

the information of the estimated optical flow vectors for obstacle detection. The obstacle avoidance of the unmanned aerial vehicles based on the optical flow method and the Lucas-Kanade algorithm is proposed by Gosiewski *et al.* [13]. Mori and Scherer used the speeded-up robust features (SURF) to detect the obstacles [14]. Jia *et al.* [15] used the motion of the local feature points to separate the obstacles from shadows and road surfaces. Lee *et al.* [16] proposed an obstacle detection method which combines the inverse perspective mapping and the Markov random field model for autonomous robot with a low-mounted camera. In recent years, the deep learning became a most popular topic. Jia *et al.* [17] also adopted the deep neural networks to detection the obstacle in the images.

Challenges in this topic of the vision-based obstacle detection are the various noises in the captured images, such as the varied size of the object, the different angle of view, the variation of the contrast, the camera shake, and so on. These noises will influence the detection rate. In this paper, we propose a robust and real-time system for the obstacle detection in a moving bio-robot that is also a small vehicle. The single camera that is mounted near the floor is used to capture the image sequence in this system. This system is in order to imitate the vision of the smaller animals. Since the computing power of the hardware for a small bio-robot is always lower, the proposed method must be efficient and simple. The proposed approach combines the optical flow method [18], the SURF [19], and the support vector machines [20] to detect candidate points of the obstacle. Moreover, the histogram-based contrast method (HC) [21] is modified to verify the candidate points and to localize the region of the obstacles. There are two major contributions in this paper. One is that the original HC method is modified to the spatial weighted HC which improves the saliency map more useful to highlight the obstacles in the natural scene. And the other is the proposed system does not require the powerful hardware to detect the obstacle in the testing stage.

The remainder of this paper is organized as follows. Section 2 roughly describes the optical flow, SURF, and SVM, those are combined to detect the candidate points of the obstacle. The modified HC algorithm is proposed in Section 3. Experiment of obstacle detection in illustrated in Section 4. Section 5 concludes this paper.

2. OBSTACLE POINTS DETECTION

In this section, the optical flow, SURF and SVM are first reviewed. Then we propose a method to combine these techniques to detect the obstacle points in the image sequence. For the proposed method, the dense optical flow technique is adopted to obtain the training data to train a SVM model. Then the SURF method is used to detect and estimate the motion of each interest point that is verified whether an obstacle point or not by the SVM. The algorithms including the optical flow method, the SURF, and the SVM classifier are roughly described in the follows.

2.1 Optical Flow

The concept of the optical flow was proposed by Gibson in 1950 [18]. The optical flow can be used to estimate the motion of the object in the image sequence. This method is also called image flow or optical velocity field. Assume that the intensity of an image frame at position (x, y) and at time t is $I(x, y, t)$. The optical flow can be represented as following equation:

$$I_x u + I_y v + I_t = 0 \quad (1)$$

where I_x , I_y , and I_t are the derivatives of the image at (x, y, t) in the corresponding direction and u and v are the optical flow of $I(x, y, t)$ in x and y directions, respectively.

There are two types of the optical flow: dense [22] and sparse [23] methods. For the dense technique, the velocity (optical flow) of each pixel need be calculated, hence it is more accurate but slower. In contrast, the sparse optical flow can be estimated by using some particular points in the image, and therefore it is more suitable for the real-time detection. However, the object with less texture is hard to detect by the sparse method. In this paper, we adopted the dense method to extract the optical flow of each pixel. And then, four information of each optical flow including the moving length (d), the orientation (θ), and the position (x, y) are normalized to $[0, 1]$ as the training data (x, y, d, θ) .

2.2 Speeded-Up Robust Features

Bay *et al.* proposed the speeded-up robust features (SURF) [19] which is inspired by the scale-invariant feature transform (SIFT) [24]. The SURF algorithm includes a local feature detector and a feature descriptor. Since the SURF adopts the Hessian matrix and the integral image method to detect the interest points in an image, it is always faster than the SIFT. The details of the SURF algorithm can refer to original literature. In our system, the SURF method is used to detect the feature points in each frame. Then the detected points in the current frame are matched with those in the previous image. Finally, each matching pair of the feature points can be used to calculate the moving distance, the shift angle, and the position of the matched point. These four elements are normalized to form a 4-D feature vector that is the test data of the classifier.

2.3 Classifier of Obstacle Points

In this paper, we utilize the SVM as the classifier to judge whether the interest points are on the obstacles in an image. The SVM is the supervised learning algorithm used for classification and it is a popular research in the machine learning area. The SVMs are successfully applied to many vision-based applications, such as the character recognition, the shape classification, the image retrieval, and the biometrics. In our experiments, the non-linear SVM with RBF kernel is adopted to verify the points of the obstacles.

In the training stage, the dense optical flow method is used to collect a lot of the training data as described in Section 2.1. Each training sample is composed of a training vector (x, y, d, θ) and its labeled category (obstacle or non-obstacle). As shown in Fig. 1, the training data is then used to train the non-linear SVM model.

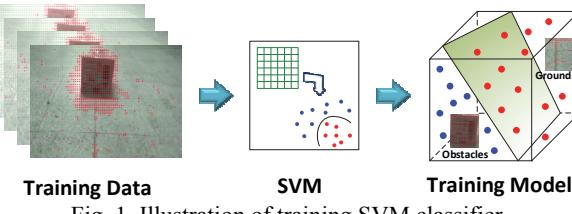


Fig. 1. Illustration of training SVM classifier.

2.4 Real-time Detection of Obstacle Points

As described in section 2.3, the SVM model with non-linear kernel is trained by using the training data that are extracted by the dense optical flow method. When the mobile robot moves in the outdoor environment, the computing time is too long for the dense optical flow. So that, the SURF is adopted to detect the interest points and the interest points of two different frames are matched to estimate the motions of these points. The interest points of the image can be quickly detected by SURF method. And the moving information and the position of the points which are matched between two image frames can form the test data. After that, each test vector is judged whether it indicates an obstacle point by the SVM. We note that each element of the test vector should also be normalized to [0, 1].

3. SPATIAL WEIGHTED HC

The saliency map is combined with the detection of the obstacle points to locate the region of the obstacle in the proposed system. We first describe how the original HC method does estimate the saliency map in this section. Then the proposed measurement for the saliency image with the spatial weighting is presented.

3.1 Original Histogram-based Contrast

In order to estimate the salient regions in an image without a priori knowledge, Cheng *et al.* [21] proposed a histogram-based contrast method using the color statistics of the input image to measure the saliency map. In the image I , the salient value of a pixel (I_k) is measured by using its color contrast to each other pixel:

$$S(I_k) = \sum_{\forall I_i \in I} D(I_k, I_i) \quad (2)$$

where $D(I_k, I_i)$ indicates the color distance between pixels I_k and I_i in the $L \times a \times b$ color space [24]. Eq. (2) can be rewritten as following form:

$$S(I_k) = S(c_l) = \sum_{j=1}^m f_j D(c_l, c_j) \quad (3)$$

where c_l indicates the color of the pixel I_k , m is the number of colors in the image, and f_j is the frequency of the color c_j appearing in I .

It is time-consuming to calculate the saliency by using Eq. (2) which is pixel-based measurement. Since the number of colors is always fewer than the number of pixels in an image, Eq. (3) is adopted to estimate the salient value. However, the number of colors is still large in the color space, *e.g.* the number is 256^3 in the RGB space. Thus, to reduce the number of colors is an important task of a real-time system. In the original literature of the HC method, each color channel is quantized into 12 bins by using the histogram method. Thus, the number of colors is reduced to 12^3 . Moreover, the colors those rarely appear in an image are merged into the similar colors. By preserving the significant colors, the number of colors is dramatically decreased to reduce the compu-

ting time. The salient values that are represented by the quantized color histogram are than smoothed to reduce the noise caused by the quantization. The smoothing procedure of the salient value of color c is shown in Eq. (4).

$$S'(c) = \frac{1}{(n-1)T} \sum_{i=1}^n (T - D(c, c_i)) S(c_i) \quad (4)$$

where n is the number of selected colors those near the color c and $T = \sum_{i=1}^n D(c, c_i)$.

3.2 Spatial Weighted Saliency Image

When the obstacle is far from the camera, the frequency of the colors on the obstacle is lower. Thereby, the obstacle is not significant in the saliency map and it is difficultly detected in the image. To enhance the saliency of the obstacle, the higher frequency of pixel color should be adjusted. In our experiment, if the ranking of color is in front of 35%, its frequency is decreased. Moreover, the camera of our system is put on a robot to detect the obstacles in the outdoors. In this scenario, the obstacle should be detected in the middle zone of the image in the vertical direction and so we can partition the image into three regions. As shown in Fig. 2, the upper region and the lower region are always covered by sky and ground, respectively. According to this observation, we set the middle zone of the image in the y direction to be the region of interest (ROI) and proposed a spatial weighted HC method (SWHC) to improve the saliency map for easily detecting the obstacles.

In the proposed SWHC, three additional tasks are involved in the original HC method. First, the spatial weight of each pixel is calculated to generate a weighting map. Then, the smoothing histogram of the salient values need be mapped to the image space, which is so-called the saliency image. Finally, the salient value of each pixel is multiplied by the spatial weight according to the pixel's position.

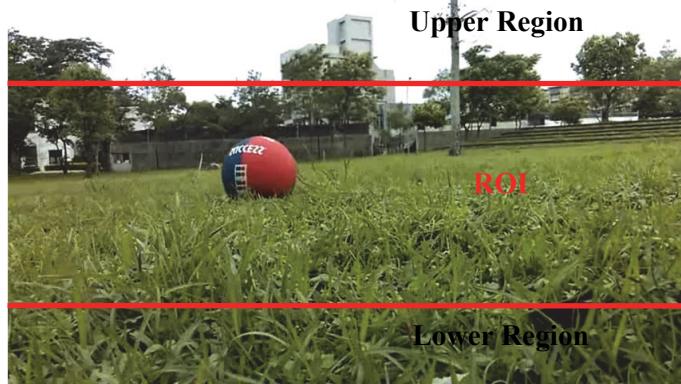


Fig. 2. The partitioned regions of an image.

When we calculate the saliency map by using Eq. (3), we can also count the number of colors occurring in the upper and lower regions and count the number of pixels that

belongs to each color in both regions. Suppose there are s and t colors in the upper region and the lower region, respectively. Let n_p^{UR} denote the number of pixels of p th color in the upper region and n_p^{LR} be the number of pixels of q th color in the lower region. Then the spatial weight of each pixel can be composed of four components according to location of the pixel. The weights corresponding to the different regions are defined as follows:

$$w_{(x,y)}^R = \begin{cases} w_p^{UR}, & \text{if } (x,y) \text{ in upper region.} \\ w_p^{LR}, & \text{if } (x,y) \text{ in lower region.} \\ 1, & \text{if } (x,y) \text{ in ROI.} \end{cases} \quad (5)$$

$$w_p^{UR} = \frac{\sqrt{n_p^{UR}}}{\sqrt{n_1^{UR}} + \sqrt{n_2^{UR}} + \dots + \sqrt{n_s^{UR}}}. \quad (6)$$

$$w_q^{LR} = \frac{\sqrt{n_q^{LR}}}{\sqrt{n_1^{LR}} + \sqrt{n_2^{LR}} + \dots + \sqrt{n_t^{LR}}}. \quad (7)$$

In addition, as shown in Fig. 3, we defined the fuzzy-based weights related to the y position of the pixels:

$$w_y^{upside} = \frac{H-y}{H}, \quad (8)$$

$$w_y^{downside} = \frac{y}{H}, \quad (9)$$

where H indicates the height of the image. The fuzzy-based weighting is used to adjust the influence of the region-based weighting that defined in Eq. (5).

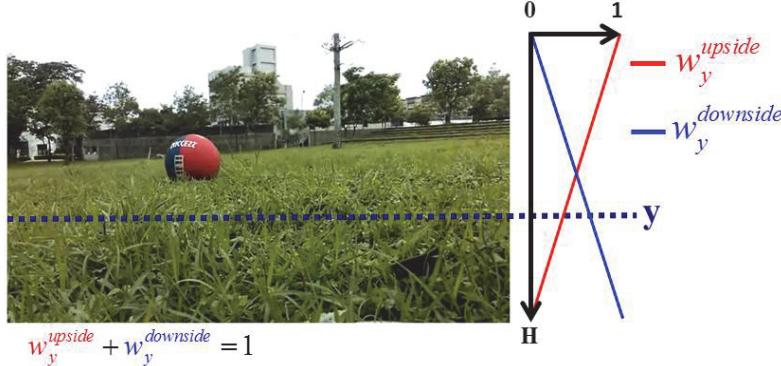


Fig. 3. Illustration of the fuzzy-based weighting.

In order to simplify the represent of the spatial weighting, the redundant region is used to indicate both the upper and lower regions. Assume that the coordinate of a pixel of the image is (x, y) and the color of this pixel is c , the spatial weight of the salient value

of this pixel can be represented as follows:

$$w_{(x,y)}^{sw} = \begin{cases} w_y^{upside} \times w_p^{UR} + w_y^{downside} \times w_q^{LR}, & \text{if } c \text{ appears in redundant region.} \\ w_y^{upside} \times w_p^{UR} + w_y^{downside} \times 1, & \text{if } c \text{ appears in upper region.} \\ w_y^{upside} \times 1 + w_y^{downside} \times w_q^{LR}, & \text{if } c \text{ appears in lower region.} \\ w_y^{upside} \times 1 + w_y^{downside} \times 1, & \text{if } c \text{ appears only in ROI.} \end{cases} \quad (10)$$

As mentioned early, the smoothing histogram of the salient values should be mapped to the image space ($S'(c) = S'(x, y)$). Then the saliency map of the SWHC can be define as the following formula:

$$S''(x, y) = w_{(x,y)}^{sw} \times S'(x, y), \quad I \in I(x, y). \quad (11)$$

where $I(x, y)$ is the pixel of the image at (x, y) . The procedure of the SWHC can refer to Fig. 4.

4. EXPERIMENTS

In this paper, we propose an obstacle detection for the robot in the outdoor environment. This scheme is composed of the optical flow method, the SURF algorithm, the SVM classifier, and the spatial weighted saliency map. In this section, the experimental results are presented and discussed. The procedure of the obstacle detection is shown in Fig. 4.

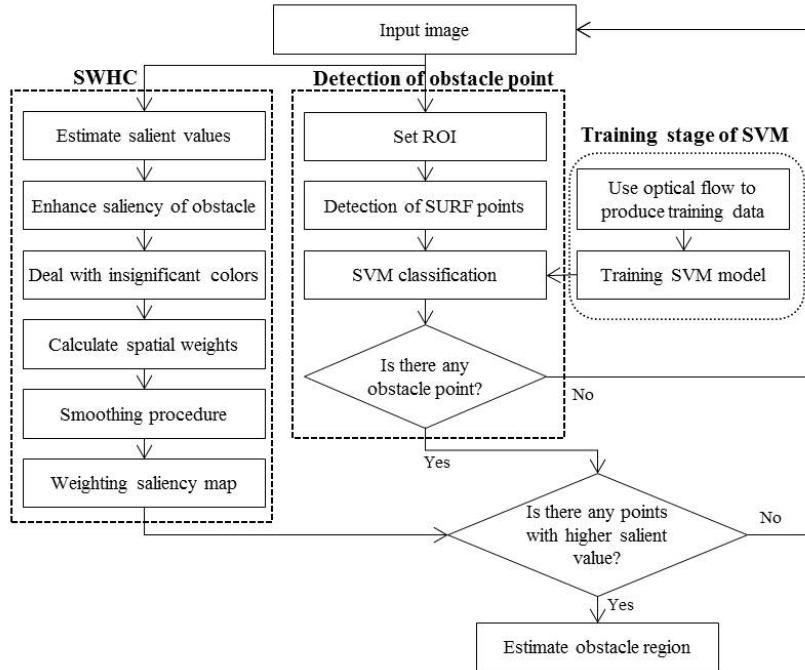


Fig. 4. Flowchart of obstacle detection in outdoor environment.

In our experiments, the image sequence is captured by the camera on the robot. The frame per second (FPS) is generally set to 30, and it is time-consuming to detect the obstacles by using every frames. Therefore, the interval between the selected frames is 15. In other words, the system is set to detect obstacle every half a second. The selected frame is the input image that is used to estimate the saliency map and to detect the obstacle points.

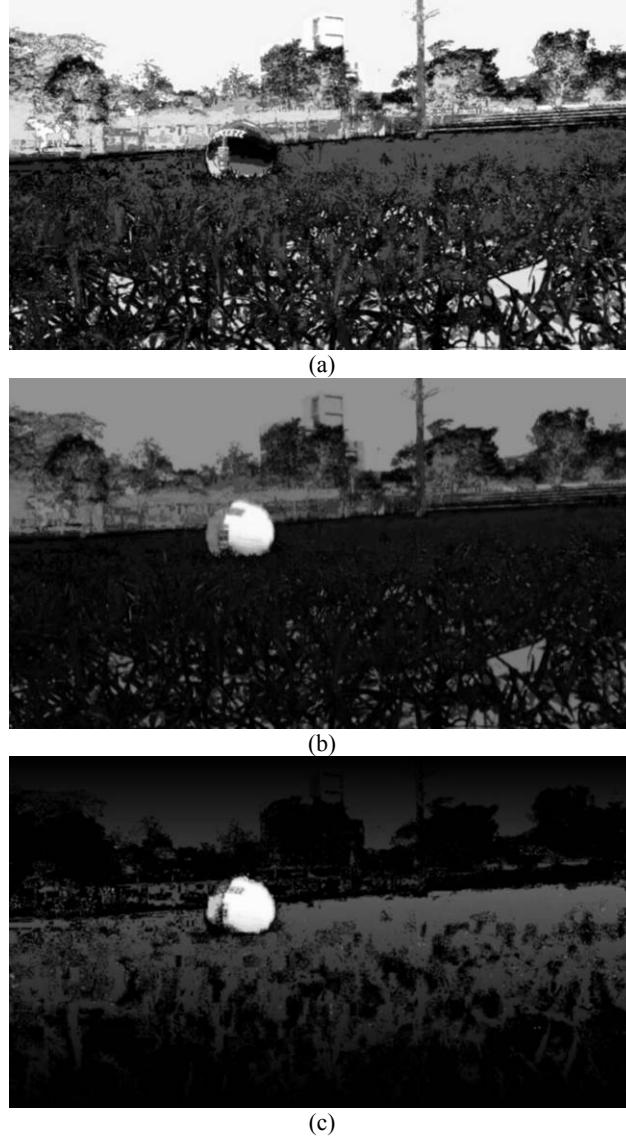


Fig. 5. The saliency maps: (a) Original HC method; (b) Enhanced by adjusting frequency; (c) Spatial weighting.

We first present and compare the experimental results of the proposed SWHC method step by step. As shown in Fig. 5 (a), it is the saliency map of Fig. 2 which is estimated by the original HC method. The nearest obstacle is a basketball in this image, however, the saliency of the ball is not significant. In Fig. 5 (b), after the enhancement of the saliency, the salient values of the obstacle are raised. As described in section 3, the method of the enhancement is to adjust the frequency of the color. Finally, the saliency map is multiplied by the spatial weights. As shown in Fig. 5 (c), we can observe that the obstacle is more significant to the surrounds.

The fuzzy-based weighting is designed to the adverse effects of the region-based weighting. Briefly, if the color of the obstacle also occurs in the upper or lower region, the obstacle may disappear in the saliency map. Observing the results shown in Fig. 6, the fuzzy-based weighting effectively compensate the influence of the region-based weighting.

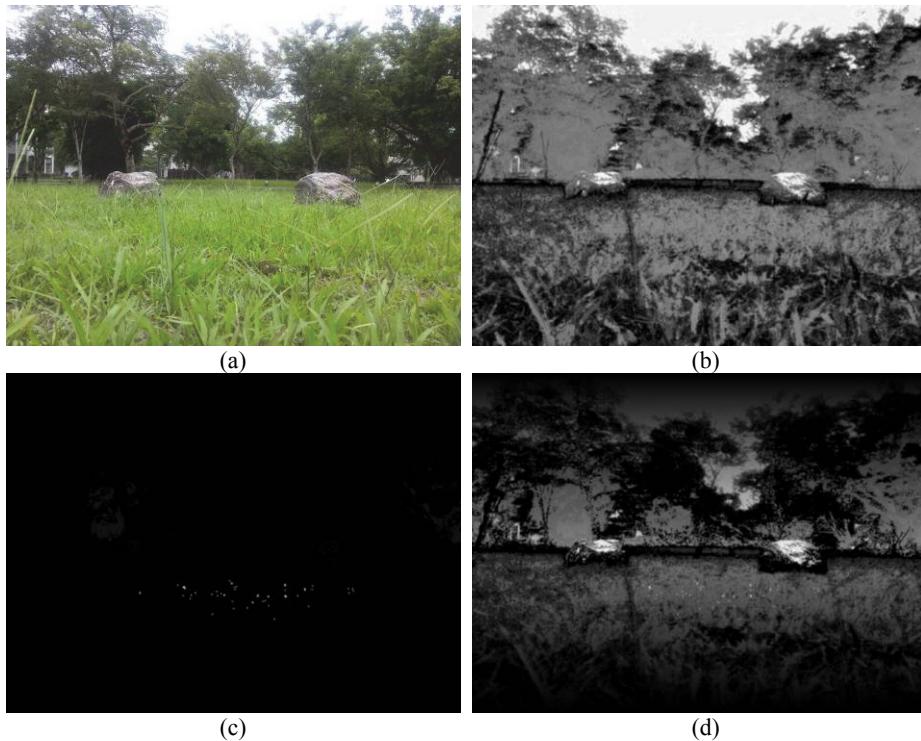


Fig. 6. Comparison with saliency maps: (a) is the original image; (b) is the saliency map of (a) using HC method; (c) is the region-based weighted saliency map; (d) is the saliency image based on the SWHC.

The saliency map that is estimated by the proposed SWHC method can make the obstacle prominent. However, the region that is not the obstacle may still own higher salient values. In order to detection the obstacle precisely, we proposed a detection of the obstacle points that is described in section 2. This algorithm contains the dense optical flow, the SURF method, and the SVM classifier. The training stage of the SVM is men-

tioned in section 2.3. In the test stage, SURF method is used to detect the interest points in the ROI of the current frame. These points of the current frame and the previously selected frame are then compared to find the matching points. Each pair of the matching point can be used to estimate the optical flow. Finally, the trained SVM model is utilized to verify whether each matching point is obstacle point or not. As shown in Fig. 7 (a), the green and red points are the matching points those are detected and matched by the SURF and the white line represents the motion (optical flow) of each matching point in the current frame. Moreover, the green points are the obstacle points and the red points are non-barriers which are verified by the SVM model. It can be observed that the obstacle points verified by SVM may occur on the far objects or produce the false results. Therefore, the proposed detection of the obstacle points is combined with the SWHC algorithm. If the salient value of an obstacle point verified by SVM is lower than a pre-define threshold is discarded. In our experiments, the threshold is set to 0.7 and an example is shown in Fig. 7 (b) where the blue points indicate the preserved obstacle points.

This combined method can find the obstacle points with higher salient values. The centroid of these points is calculated. And then, a rectangular region near the centroid with higher salient value is defined as the region of the obstacle. Figs. 7 (c) and (d) represent the region of an obstacle which is drawn as a yellow rectangle. In our experiments, the system is set to detect one obstacle. The simple clustering method can be used to category the obstacle points to detect more obstacles. Furthermore, the center of the obstacle region that is estimated on the previous selected frame can be used to predict and detect the obstacle region when there is no obstacle point. Since the obstacle cannot disappear suddenly, the center of the previous region of the obstacle can be used to search

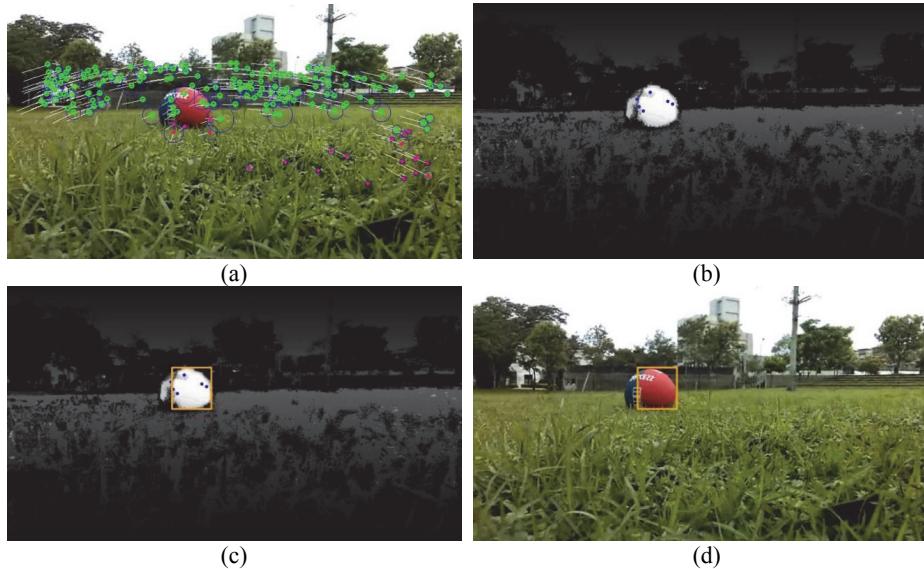


Fig. 7. The procedure for detecting the obstacle region; (a) illustrates the detection and classification of the interest points; (b) represents the obstacle points with higher salient value; (c) and (d) display the detected obstacle region on saliency map and original image, respectively.

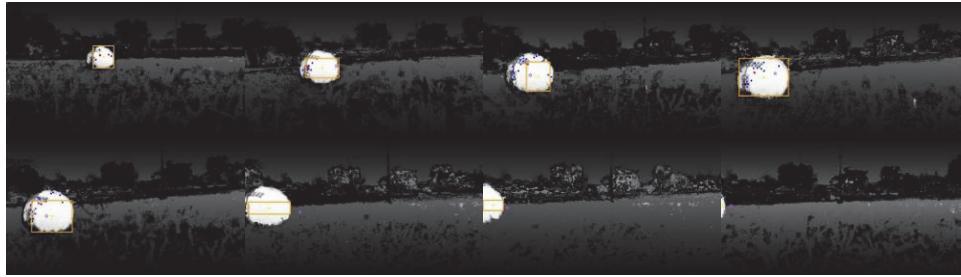


Fig. 8. Examples of the obstacle detection on the sequence. The blue points and a red point represent the obstacle points and the centroid of these points. The yellow rectangle and point indicate the obstacle region and the center of this region. Moreover, the purple circle is the center of the obstacle region detected on the previous selected frames.

the current region around the center. If there is no obstacle point has been detected, the system will search the saliency map near the center of previous region to find whether existing the higher saliency. Fig. 8 represents the detection results of an image sequence. According to the experimental results, the proposed system can effectively and correctly detection the obstacle region in the outdoor environment.

The proposed real-time detection of the obstacle is used by a small bio-robot in which the computing capability is limited. The detection performance of the popular research used the deep learning could be better than our method, however, the deep learning need a powerful GPU to perform the algorithm. So the proposed method is more suitable for a small robot. In addition, the camera of our system is mounted near ground. This condition is similar to Lee *et al.* proposed system [16] but their experiments are only performed indoor.

5. CONCLUSION

It is a difficult task to detect the obstacle using single camera in the outdoor environment. In order to assist a robot to avoid the obstacle, we proposed a vision-based obstacle detection. The detection system is composed of two major schemes. One is the extraction of the obstacle points in the image sequence, and the other is the generation of a saliency map in which the obstacle region is significant. The dense optical flow technique is used to extract the training data of the SVM. In the test stage, the trained SVM model is utilized to verify the obstacle points that are detect by the SURF method. Moreover, we proposed the SWHC to generate a saliency map that is combined with the detection of the obstacle point to locate the region of the obstacle. Our experimental results show that the proposed system can quickly and effectively detect the obstacle position in the outdoor environment.

REFERENCES

1. Z. Xu, Y. Zhuang, and H. Chen, “Obstacle detection and road following using laser scanner,” in *Proceedings of the 6th World Congress on Intelligent Control and Automation*, Vol. 2, 2006, pp. 8630-8634.

2. R. Malik and H. Yu, "The infrared detector ring: obstacle detection for an autonomous mobile robot," in *Proceedings of the 35th Midwest Symposium on Circuits and Systems*, Vol. 1, 1992, pp. 76-79.
3. Y. Han and H. Hahn, "Localization and classification of target surfaces using two pairs of ultrasonic sensors," in *Proceedings of IEEE International Conference on Robotics and Automation*, Vol. 1, 1999, pp. 637-643.
4. A. Ohya, A. Kosaka, and A. Kak, "Vision-based navigation by a mobile robot with obstacle avoidance using single-camera vision and ultrasonic sensing," *IEEE Transactions on Robotics and Automation*, Vol. 14, 1998, pp. 969-978.
5. P. Foggia, A. Limongiello, and M. Vento, "A real-time stereo-vision system for moving object and obstacle detection in AVG and AMR applications," in *Proceedings of the 7th International Workshop on Computer Architecture for Machine Perception*, 2005, pp. 58-63.
6. M. Leung, Y. Liu, and T. Huang, "Estimating 3D vehicle motion in an outdoor scene from monocular and stereo image sequences," in *Proceedings of IEEE Workshop on Visual Motion*, 1991, pp. 62-68.
7. M. Bertozzi and A. Broggi, "GOLD: A parallel real-time stereo vision system for generic obstacle and lane detection," *IEEE Transactions on Image Processing*, Vol. 7, 1998, pp. 62-81.
8. H. Liu, W. Pi, and H. Zha, "Motion detection for multiple moving targets by using an omnidirectional camera," in *Proceedings of IEEE International Conference on Robotics, Intelligent Systems and Signal Processing*, Vol. 1, 2003, pp. 422-426.
9. K. Yamazawa and N. Yokoya, "Detecting moving objects from omnidirectional dynamic images based on adaptive background subtraction," in *Proceedings of the 10th IEEE International Conference on Image Processing*, Vol. 3, 2003, pp. 953-956.
10. Q. Zhu, "Hidden markov model for dynamic obstacle avoidance of mobile robot navigation," *IEEE Transactions on Robotics and Automation*, Vol. 7, 1991, pp. 390-397.
11. P. Jeong and S. Nedevschi, "Obstacle detection based on the hybrid road plane under the weak calibration conditions," in *Proceedings of IEEE Intelligent Vehicles Symposium*, 2008, pp. 446-451.
12. W. Kruger, W. Enkelmann, and S. Rossle, "Real-time estimation and tracking of optical flow vectors for obstacle detection," in *Proceedings of IEEE Intelligent Vehicles Symposium*, 1995, pp. 304-309.
13. Z. Gosiewski, J. Ciesluk, and L. Ambroziak, "Vision-based obstacle avoidance for unmanned aerial vehicles," in *Proceedings of IEEE International Congress on Image and Signal Processing*, Vol. 4, 2011, pp. 2020-2025.
14. T. Mori and S. Scherer, "First results in detecting and avoiding frontal obstacles from a monocular camera for micro unmanned aerial vehicles," in *Proceedings of IEEE International Conference on Robotics and Automation*, 2013, pp. 1750-1757.
15. B. Jia, R. Liu, and M. Zhu, "Real-time obstacle detection with motion features using monocular vision," *The Visual Computer*, Vol. 31, 2015, pp. 281-293.
16. T. J. Lee, D. H. Yi, and D. I. Cho, "A monocular vision sensor-based obstacle detection algorithm for autonomous robots," *Sensors*, Vol. 16, 2016, p. 311.
17. B. Jia, W. Feng, and M. Zhu, "Obstacle detection in single images with deep neural networks," *Signal, Image and Video Processing*, Vol. 10, 2015, pp. 1033-1040.

18. J. J. Gibson, *The Perception of the Visual World*, Houghton Mifflin, Boston, 1950.
19. H. Bay, A. Essa, T. Tuytelaarsb, and L. V. Gool, “Speeded-up robust features (SURF),” *Computer Vision and Image Understanding*, Vol. 110, 2008, pp. 346-359.
20. S. R. Gunn, “Support vector machines for classification and regression,” Technical Report, Image Speech and Intelligent Systems Research Group, University of Southampton, 1998.
21. M.-M. Cheng, G.-X. Zhang, N. J. Mitra, X. Huang, and S.-M. Hu, “Global contrast based salient region detection,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 37, 2015, pp. 569-582.
22. G. Farnebäck, “Two-frame motion estimation based on polynomial expansion,” in *Proceedings of Scandinavian Conference on Image Analysis*, 2003, pp. 363-370.
23. J. Y. Bouguet, “Pyramidal implementation of the affine Lucas Kanade feature tracker description of the algorithm,” Technical Report, OpenCV Document, Intel Corporation, Microprocessor Research Labs, 2001.
24. D. G. Lowe, “Distinctive image features from scale-invariant keypoints,” *International Journal of Computer Vision*, Vol. 2, 2004, pp. 91-110.
25. Y. Zhai and M. Shah, “Visual attention detection in video sequences using spatio-temporal cues,” in *Proceedings of the 14th Annual ACM International Conference on Multimedia*, 2006, pp. 815-824.



Chung-Chih Tsai (蔡仲智) received the B.S., M.S., and Ph.D. degrees in Electrical Engineering from the National Chung Hsing University, Taiwan, in 2001, 2003 and 2009, respectively. He served as a Postdoctoral Fellow with the Department of Electrical Engineering, National I-Lan University, Taiwan, from 2013 to 2015. His research interests include biometrics, image processing, pattern recognition, neural networks, and machine learning.



Chia-Wen Chang (張嘉文) received the B.S. degree in Electrical Engineering from National Ilan Institute of Technology, Yilan, Taiwan, in 2003 and the M.S. and Ph.D. degrees in Electrical Engineering from Chang Gung University, Taoyuan, Taiwan, in 2005 and 2011, respectively. He is currently an Assistant Professor with the Department of Information and Telecommunications Engineering, School of Information Technology, Ming Chuan University, Taoyuan, Taiwan. His research interests include fuzzy sliding-mode control, intelligent computation, and multi-robot systems.



Chin-Wang Tao (陶金旺) received the B.S. degree in Electrical Engineering from National Tsing Hua University, Hsinchu, Taiwan, in 1984 and the M.S. and Ph.D. degrees in Electrical Engineering from New Mexico State University, Las Cruces, NM, USA, in 1989 and 1992, respectively. He is currently a Professor with the Department of Electrical Engineering, College of Electrical Engineering and Computer Science, National Ilan University, I-lan, Taiwan. His research interests include fuzzy neural systems, including fuzzy control systems and fuzzy neural image processing. Dr. Tao is an Associate Editor of the IEEE Transactions on Systems, Man, and Cybernetics.