Adaptive Visual Obstacle Detection for Mobile Robots Using Monocular Camera and Ultrasonic Sensor

İbrahim K. İyidir, F. Boray Tek, and Doğan Kırcalı

Robotics and Autonomous Vehicles Laboratory
Department of Computer Engineering,
Işık University, 34980, İstanbul, Turkey
{ibrahim.iyidir,boray,dogan}@isikun.edu.tr
http://ravlab.isikun.edu.tr

Abstract. This paper presents a novel vision based obstacle detection algorithm that is adapted from a powerful background subtraction algorithm: ViBe (VIsual Background Extractor). We describe an adaptive obstacle detection method using monocular color vision and an ultrasonic distance sensor. Our approach assumes an obstacle free region in front of the robot in the initial frame. However, the method dynamically adapts to its environment in the succeeding frames. The adaptation is performed using a model update rule based on using ultrasonic distance sensor reading. Our detailed experiments validate the proposed concept and ultrasonic sensor based model update.

Keywords: Obstacle detection, ViBe, ultrasonic sensor, mobile robot.

1 Introduction

In the last decade the mobile robot technologies have made a significant progress. Beside big budget commercial efforts, we see that small projects which are conducted by individuals or small groups who have limited budgets provide a noticeable addition to its momentum. This work is a part of our efforts where we aim to transform a model car kit into an autonomous navigating robot using a camera, electronic sensors and an onboard PC.

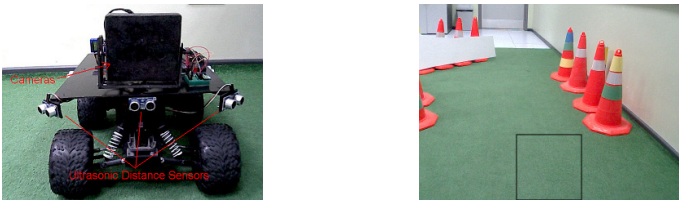
For a ground vehicle, an obstacle may be defined as a protrusion or extrusion of the ground surface, which segments the current planned path. However, in practice it is not simple to construct a generic method to implement obstacle detection due to huge variation in environmental conditions [1].

In line with human sensory organization the most used sensor for obstacle detection is visible light camera. Though analyzing two dimensional images for obstacle detection is challenging, in the last two decades there are significant amount of attempts [1,2]. Several concepts have been investigated using two dimensional images for obstacle avoidance problem such as edge detection [3], optical flow [4] and pixels values [5]. Additional third dimension information

A. Fusiello et al. (Eds.): ECCV 2012 Ws/Demos, Part II, LNCS 7584, pp. 526–535, 2012.

can be obtained using more than one camera which involves a pre-calibration process but it required more computation until recent commercial development of low-cost depth sensors such as MS-Kinect[1].

In addition to a visible light camera various other sensors (e.g. radar, lidar, and sonar) can be used for obstacle detection [6]. One of the earlier works that has done by Borenstein and Koren [7] uses only ultrasonic sensors for obstacle avoidance. Laser based sensors such as 3D-lidar are quite successful in scanning the environment however they are expensive and require significant computing resource.



(a) (b)

Fig. 1. (a)Model car equipped with camera and sensors (b)Obstacle free region

Ultrasonic distance sensors provide a cheaper way to measure distance in an unstructured environment. Any obstacle surface which is parallel to the sensor and inside the ultrasonic beam region reflects the sound emitted by the sensor. The reflected wave is captured by the sensor and time-of-flight is calculated to measure the distance. However, the measurements can only be localized to a limited precision (i.e. location in the beam pattern). Moreover, ultrasonic sensors may not give reliable results on reflective surfaces. In [7] Borenstein and Koren investigated the limitations of the ultrasonic range finders for obstacle avoidance task.

Our robot is based on a model car, equipped with an RGB color camera, an ultrasonic distance sensor, a compass and a gps sensor (Figure 1(a)). The obstacles that we aim to detect may appear in any shape or color in the scene.

Several other algorithms have assumptions about the operating environment [1,2] and we unavoidably make an assumption about expecting an obstacle free bounded region in front of the car in the first frame. However, the method dynamically adapts to its environment in the consecutive frames. Therefore it can be suitable for both indoor and outdoor environments even in places where there is no structural nature. Our method mainly relies on a monocular camera to model the ground plane and other obstacles in the scene. Since the robot is mobile and also that there are no assumptions about the scene structure, the ground plane model must be updated continuously. This model update is controlled based on the ultrasonic distance sensor reading to avoid ground plane like obstacle regions included in the model. Authors of [3,8] also used combination of

[1] `http://www.microsoft.com/en-us/kinectforwindows/develop/overview.aspx`

a monocular camera and ultrasonic range finder for obstacle avoidance purposes. In contrast to their work, we utilize ultrasonic sensor's data only for the decision of updating the ground plane model.

Our ground plane modelling scheme is inspired from ViBe [9] which is a background subtraction algorithm that is proposed for motion detection. ViBe keeps positive samples for every pixel location and classifies a previously unseen pixel according to its distance from stored samples. In other words, every pixel location has an independent classifier to decide whether a given candidate pixel for that location is from the ground plane or not. Through time, some random samples are updated with some random new pixels collected and classified from the scene. This update mechanism gives ViBe its dynamic adaptive behavior and robustness against local illumination changes, reflections and shadows. ViBe [9] is simple, efficient and faster than its counterparts which suits well to our robot where we cannot operate with high performance computers due to limitations of size and payload.

We also incorporate a methodology of a simple obstacle detection algorithm developed by Ulrich and Nourbakhsh [5] which assumes that the area in front of the mobile robot is the ground plane and can be modeled by hue and intensity histograms. A previously unseen pixel is classified as a ground plane pixel if corresponding index in one of the histograms has accumulated sufficient samples (i.e. above a threshold). Moreover, they propose a post-update rule based on odometry information and successful passing from a region. Thus the method relies on a feedback mechanism which has some trial and error nature which cannot be assumed to be safe under many conditions. In contrast to their method, our ground plane model update rule is based on a pre-update mechanism that uses the ultrasonic distance sensor to control the ground plane model update.

This paper is organized as follows: In Section 2, we present our proposed method. In Section 3, we discuss the parameters of our method and give the results of our experiments. Our conclusions and future work are presented in Section 4.

2 Method

2.1 Initialization

Following the work of [5], our obstacle detection technique is based on the assumption that the square region in front of the mobile robot is free of obstacles at the initialization step of the algorithm (Figure 1(b)). As in [9], we define and maintain a sample set $\mathbf{S}(x, y)$ for each pixel location (x, y) of the image $\mathbf{I}$ of dimensions WxH where W is width and H is height of the image. Each sample set $\mathbf{S}(x, y)$ stores N samples.

$$\mathbf{S}(x, y) = \{p_1, p_2, p_3, ...p_N\} \tag{1}$$

For initialization, in the first frame, random pixels are chosen from the obstacle-free square region (OFR) in front of the robot and their values are added to

$\mathbf{S}(x, y)$ for every (x, y) in $\mathbf{I}$. The set of sample sets $\mathbf{S}$ whose size is $NxWxH$ forms our ground plane model. The size of OFR must be determined based on the width of the car and camera's field of view. If the region that we collect samples from gets larger, the chance of including obstacle pixels in the ground plane model increases whereas a smaller area would mean less variation in the model. Because the ground plane model $\mathbf{S}$ is created using only the first frame, the initialization is very fast.

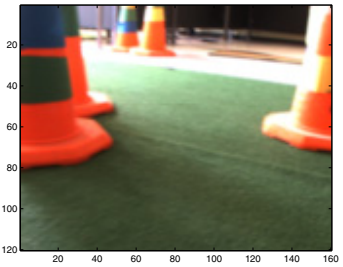


Fig. 2. Original Image

2.2 Obstacle Detection

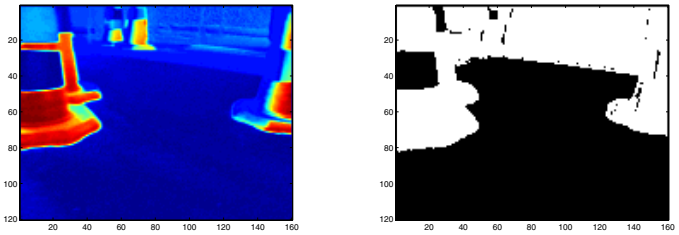
After initialization, each new frame is processed to detect ground plane and obstacle pixels. A given pixel value at location (x, y) is classified as an obstacle or not using its $\mathbf{S}(x, y)$. The classification is based on finding similar values to the pixel's own value in its $\mathbf{S}(x, y)$. Finding at least M similar pixels in $\mathbf{S}(x, y)$ is sufficient to mark the pixel in location (x, y) as ground plane pixel.

Let the pixel value at position (x, y) be q, then we can express the number of similar pixels (SPC) as follows:

$$SPC = \|dist(q, p_i) < R)\|, \quad where\ p_i\, \epsilon\, \mathbf{S}(x, y)\ , i = 1, ..., N \tag{2}$$

dist is the Euclidean distance metric which measures how close two sample are and R is the distance threshold which defines being similar with a limiting distance value.

Our camera produces images in RGB (Red-Green-Blue) color space (Figure 2). We operate our algorithm in normalized RGB color space in order to provide illumination invariance (Figure 4(a) (b)). Figure 3(a) shows an example distance image which is formed by pixel values that indicate the distance between $\mathbf{S}(x, y)$ and corresponding pixel value q at location (x, y). Distance image is obtained by calculating average of N distance values between the values in $\mathbf{S}(x, y)$ and q. Euclidean metric is used when calculating the distances. After classifying each pixel of the input image $\mathbf{I}$, a binary obstacle map is generated (Figure 3(b)).



(a) (b)

Fig. 3. (a)Distance image (b)Binary obstacle map

2.3 Updating Sample Sets

ViBe applies a random update rule to the background model [9]. If the pixel q at position (x, y) is classified as background, its value is used to update a random value in $\mathbf{S}(x, y)$. The update is not carried out at every frame, instead the update decision depends on a term called "subsampling factor". For each found new background pixel the probability of inclusion in the sample set is (1/subsampling factor) and decision is made using a random value. Moreover, one of the neighboring pixel's sample set of q is updated using the same probabilistic approach. Motivation here is that the adjacent pixels often share a similar temporal distribution [10].

This continuous update rule provides a dynamic adaptation to environmental changes in illumination, reflections. However, to adapt the sudden significant changes in the ground plane model, (e.g. from carpet to tiles) we needed another mechanism. We keep histograms of both OFR and sample pixels $\mathbf{S}(x, y)$. We re-initialize $\mathbf{S}(x, y)$ if these histograms differ significantly and also if ultrasonic sensor indicates that OFR is free. Significant difference of one of the histograms (normalized red or normalized green) is sufficient for the model re-initialization. The significant change is detected by a percentage threshold (Figure 4(c)-(d)). Figure 5 shows a situation where both histograms are similar therefore there is no need to re-initialize sample sets.

Effect of Ultrasonic Distance Sensor on Update. In our work, the update mechanism of $\mathbf{S}$ is dependent on the ultrasonic distance sensor data (UD). At each frame, we read ultrasonic distance sensor (UD). The reading gives us the distance to the closest obstacle in front of the robot in terms of centimeters in range (10,300). We let the update of $\mathbf{S}$ if the mobile robot can safely move one step forward afterwards. In other words, we control the update using a "safe-to-move-threshold" value (t).

Thus, this pre-update rule mechanism that is based on the safe distance check prevents our robot to update the model unselectively. If the ultrasonic distance sensor is not used, the ground plane model continues updating from

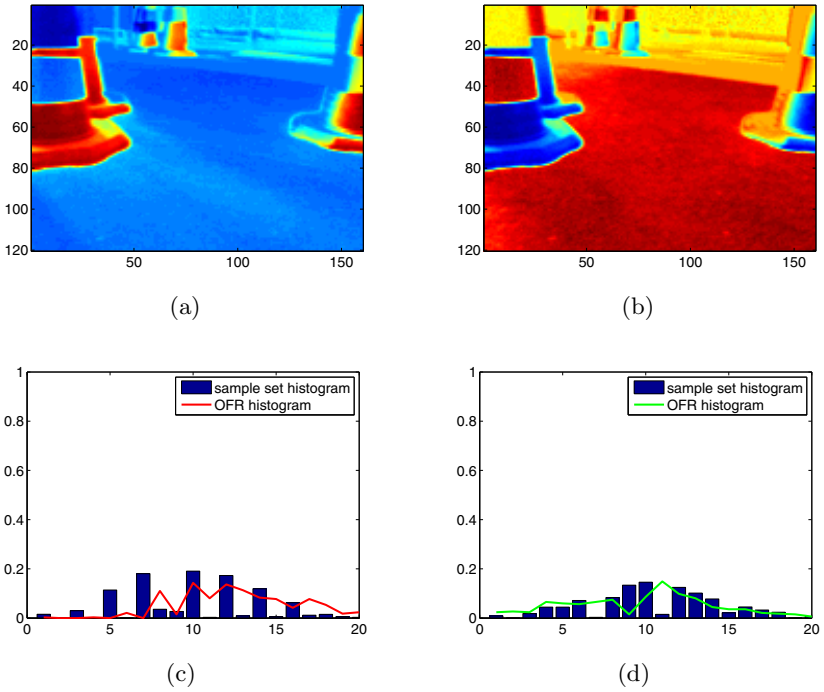


Fig. 4. (a)Normalized red image (b)Normalized green image (c)Normalized red histogram (d)Normalized green histogram

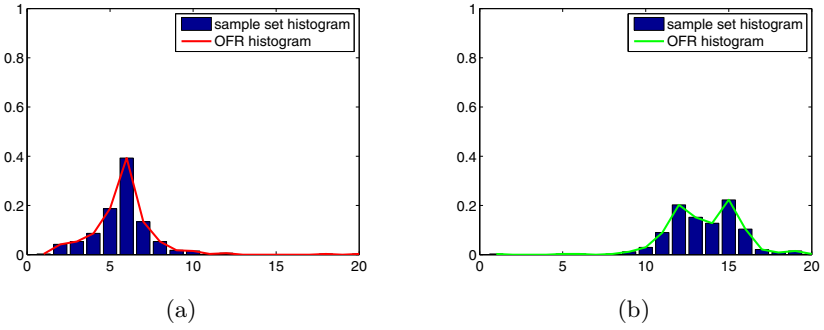


Fig. 5. (a)Normalized red histogram (b)Normalized green histogram. Both histogram pairs (sample set and OFR) are similar.

OFR whether there is an obstacle or not in the front. Moreover, if the appearance of the ground plane changes suddenly (e.g. carpet to tile), the robot cannot move towards to the different appearing ground plane because the robot treats this new appearance as an obstacle. However, when the ultrasonic distance sensor gives a safe-to-move-threshold for the new ground plane, the robot updates

its ground plane model effectively. If UD is above t, we replace two samples from $\mathbf{S}(x, y)$ with values one is randomly chosen from OFR and the value of q at the location (x, y). Also one of the adjacent pixel's sample set is updated in the same way. Note that, during the update process, only the sample sets of pixel locations that are classified as ground plane are updated. Figure 6 shows the distribution of $\mathbf{S}$ at two different time lines. Only two samples for each $\mathbf{S}(x, y)$ are drawn on the scatters.

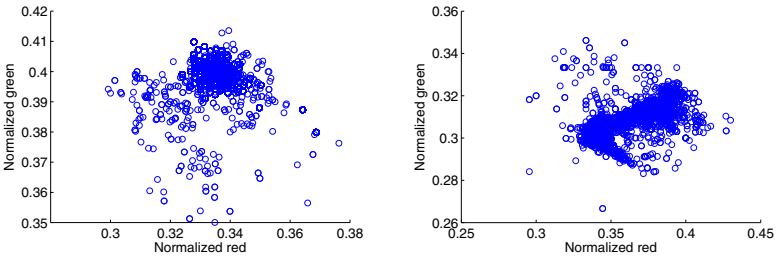


Fig. 6. Distribution of sample sets at different time lines

3 Experiments

During the tests, we run the algorithm in normalized RGB color space. Normalizing red and green channels eliminate problems that occur due to differently illuminated ground plane areas. However, we also performed some preliminary experiments using HSI (Hue-Saturation-Intensity) color space but we observed that using Hue value creates more problems than normalized red-green does in dark or saturated bright regions. We validated our proposed method using three different recorded videos which included manually segmented ground truth data that can be reached through our website[2]. The videos were recorded in our laboratory resting pitch which has illumination reflections, shadows, different colors of floor and it contains many obstacles that have different colors and shapes.

Each RGB frame in the recorded dataset has an image size of 160x120 pixels where the square that we collect samples from (OFR) contains only 20x20 pixels. During the experiments, we set the size of the sample sets (N) as 20 and UD threshold (t) as 100cm. In addition, we set the threshold value for the number of close samples (M) as 2, and we set distance threshold value (R) as 0.05

In addition to first Figures 2, 3(b), Figure 8 shows more input-output-ground truth triplets of our algorithm. We defined the ground plane as the negative class and the obstacles as the positive class. We observed that the method detects obstacles generally successfully. However, false negatives occur when the obstacle is in the same color with the ground plane (see Figures 2-3(b)). We also observed that the algorithm performs not well specifically for gray colored ground plane and gray colored obstacles (e.g walls, doors) with the above parameter settings.

[2] `http://ravlab.isikun.edu.tr`

Table 1. Performance of the algorithm on different data sets

Dataset Name	True Positive	False Positive	Accuracy
Dataset-1	0.8251 ±0.1897	0.1685 ±0.1153	0.8221 ±0.1026
Dataset-2	0.8102 ±0.1949	0.1853 ±0.2826	0.7737 ±0.2268
Dataset-3	0.7269 ±0.2301	0.1471 ±0.1726	0.8119 ±0.1532

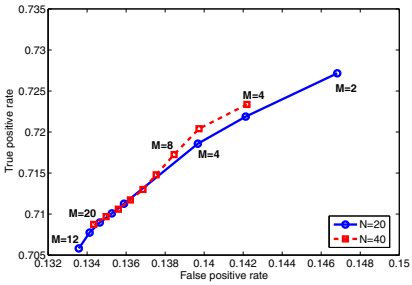


Fig. 7. ROC curve for varying M values

In order to quantify the results, we measured the overall accuracy, true and false positive rates for each frame in the different datasets (Table 1). As it can be seen, we achieved over 70% of true positive rate while false positive rate is under 20% in all datasets. Note that, these are quite challenging sequences and many errors occur during frame transitions from one ground plane character to another.

The distance threshold R value was set to 0.05 for all datasets to provide consistency through processing all the datasets. However, we observed that changing this value could improve performance of different datasets because lower R values perform better for gray colored ground plane and grayish obstacles. This is possibly due to the fact that a constant threshold value is not optimal through normalized red-green color space.

Furthermore, to observe the effect of parameters N and M, we plot a ROC curve for their different values for dataset-3. It can be seen in Figure 7 that choosing larger values of M lowers both false and true positive rates whereas lower values produce opposite condition. From different ROC curves obtained from different $N = 20, 40$ values, we observed that the algorithm is not so sensitive to the sample set size N, so we set it as 20.

In overall, we observed that our combined obstacle detection algorithm performed acceptable while we noted above mentioned points for improvement. Furthermore, most false positive detections were either isolated pixels or very small regions. Hence, they can be removed using morphological operations like erosion. During above experiments, such post-processing was not used to show the reader the exact outputs of our algorithm.

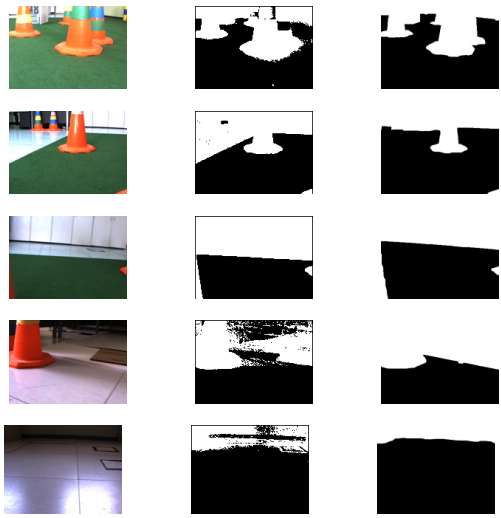


Fig. 8. Input image, obstacle map and ground truth triplets are shown in left, center and rights columns respectively

4 Conclusion

In this paper, we introduced a novel vision based obstacle detection method. Our method relies on images acquired by a monocular color camera and an assisting ultrasonic distance sensor to update its ground plane model over time. The method is adapted from ViBe which is an efficient background subtraction algorithm that is used for motion detection. ViBe model gives robustness against local illumination changes, reflections and shadows. Ultrasonic distance sensor is used to control the update decision of the ground plane model. Additionally, we detect sudden significant changes in the ground plane appearance using histograms and let re-initialization of the ground plane model.

This paper can be seen as an initial step to develop a vision based obstacle detection algorithm which has no assumptions about the scene structure and ground plane. The only assumption we made is an obstacle free region in front of the car in the first frame. However, this assumption is not a strong necessity, instead of "expecting", it is possible to "seek" an obstacle free region. If the car happens to be in front of an obstacle in the first frame (that can be detected by the ultrasonic sensor), it can perform small maneuvers through backwards and search for an obstacle free region.

The only visual feature that was used by the algorithm is the normalized red-green pixel values. It may be possible to extend this work by adding a computationally feasible textural feature. Also, we did not use any supervised learning mechanism nor a history of observed previous ground planes. Addition of such features can lead to more accurate detection of obstacles.

References

1. Desouza, G., Kak, A.: Vision for mobile robot navigation: a survey. IEEE Transactions on Pattern Analysis and Machine Intelligence 24, 237–267 (2002)
2. Bonin-Font, F., Ortiz, A., Oliver, G.: Visual navigation for mobile robots: A survey. J. Intell. Robotics Syst. 53, 263–296 (2008)
3. Ohya, I., Kosaka, A., Kak, A.: Vision-based navigation by a mobile robot with obstacle avoidance using single-camera vision and ultrasonic sensing. IEEE Transactions on Robotics and Automation 14, 969–978 (1998)
4. Ohnishi, N., Imiya, A.: Featureless robot navigation using optical flow. Connection Science 17, 23–46 (2005)
5. Ulrich, I., Nourbakhsh, I.R.: Appearance-based obstacle detection with monocular color vision. In: Proceedings of the Seventeenth National Conference on Artificial Intelligence and Twelfth Conference on Innovative Applications of Artificial Intelligence, pp. 866–871. AAAI Press (2000)
6. Discant, A., Rogozan, A., Rusu, C., Bensrhair, A.: Sensors for obstacle detection - a survey. In: 30th International Spring Seminar on Electronics Technology, pp. 100–105 (2007)
7. Borenstein, J., Koren, Y.: Obstacle avoidance with ultrasonic sensors. IEEE Journal of Robotics and Automation 4, 213–218 (1988)
8. Yu, M., Shu-qin, L.: A method of robot navigation based on the multi-sensor fusion. In: 2010 2nd International Workshop on Intelligent Systems and Applications (ISA), pp. 1–4 (2010)
9. Barnich, O., Van Droogenbroeck, M.: Vibe: A universal background subtraction algorithm for video sequences. IEEE Transactions on Image Processing 20, 1709–1724 (2011)
10. Jodoin, P.M., Mignotte, M., Konrad, J.: Statistical background subtraction using spatial cues. IEEE Transactions on Circuits and Systems for Video Technology 17, 1758–1763 (2007)