A Hybrid Approach for Dynamic Observer to Detect and Track Dynamic Obstacles

Sudeepta Ranjan Sahoo and P. V. Manivannan

***Abstract*—This paper deals with the detection of a dynamic obstacle with the use of a low-cost monocular camera mounted on a dynamic vehicle. An algorithm has been developed to integrate feature tracking, Image transformation, and optical flow. Optical flow method is used to detect a moving obstacle in a static environment. Camera motion causes the environment to be dynamic with the point of view of the observer thereby leading to the failure of optical flow algorithm. The difficulty in distinguishing the difference between moving and a stationary body can be overcome with the use of image transformation technique, which transforms the newly captured image to the coordinate frame of the previous image. The optical flow algorithm can now be applied to detect a dynamic obstacle. To further find the position of the obstacle in 3D space with respect to the camera, Stereo vision system is used. The developed algorithm has been tested in a virtual environment using V-rep and Matlab. The algorithm has also been validated experimentally with the use of a stereo camera system on a mobile platform-P3-DX.**

***Index Terms*—Optical flow, feature point matching, first corner method, image transformation, stereo vision, V-rep, P3-DX, VICON.**

I. Introduction

With the increase of automation in the transport industry, it is very important to examine the use of various navigation procedures. To navigate, we require the vehicle's position as well as its surroundings. Currently, a wide variety of sensors are being used to understand environment such as IR sensor, Ultrasonic sensor, LIDAR, etc. The major drawbacks of these sensors are their susceptibility the various environmental conditions such as vibration, environmental noise, measurement noise, etc. [1]. With the increase in flexibility of vision sensors, cameras are a good alternative due to their advantages.

For autonomous vehicular motion, we require the knowledge of the environment which includes both static as well as dynamic obstacles. For static obstacles, its position alone with respect to the moving vehicle is required. For dynamic obstacles, its current position, and velocity with respect to the vehicle is required. Research has been carried out to overcome the above issues by using various techniques that depend on different sensors such as LiDAR Distance Sensor, SONAR, and cameras [1] . Each of these sensors has its own advantages and disadvantages. In terms of accuracy, laser sensors provide good results in comparison to other types but has a higher cost. In contrast, SONAR is cost-effective but has low accuracy. The alternative to these traditional sensors is to use cameras. Using a stereo vision system, the 3D position of the obstacles can be found using disparity mapping [2].

Even though stereo vision uses two cameras to find out depth information of the environment, we cannot capture information of all the points in the environment. Many researchers use RGB-D sensor (Kinect develop by Microsoft) instead of a stereo vision for depth mapping, which can provide more accurate results [3] but it is not very cost effective. Both, stereo vision system and RGB-D sensor when used for dynamic obstacle detection cause a large increase in the computational complexity. The developed algorithm takes advantage of the low-cost monocular camera to detect a dynamic obstacle and a stereo vision system to get the position and direction of it.

Optical flow method takes advantage of the feature matching algorithm for the detection of a moving obstacle for a stationary camera condition [4]. A combination of optical flow method and a stereo vision system can be used to produce a hybrid vision system that can track the motion of an obstacle in a 3D environment [5]. Optical flow method is applicable to a stationary obstacle. The motion of the camera causes a dynamic environment thereby leading to a difficulty in distinguishing between stationary and moving obstacles caused by random optical flow values of feature points. Every small motion experienced by the camera will cause a coordinate transformation in the image captured with respect to the previous one. To overcome this difficulty, we make use of an image transformation using the feature points to transform the captured image to the coordinate frame of the previous image. The above-mentioned method has been used to provide video stabilization [6], [7].

The algorithm used in this paper makes use of this image transformation technique to stabilize two consecutive images causing the two images to appear as a single one. Hence, Optical flow method is applied to detect a dynamic obstacle, and a stereo vision system to track a dynamic obstacle in a 3D environment. The values obtained from the algorithm has been verified using Vicon cameras that provide us the values of position and speed of the dynamic obstacles.

II. Simulation Setup

The algorithm developed can also be used in real time to detect and observe a moving vehicle or obstacle and its behavior. A virtual environment called V-REP (Virtual Robot Experimental Platform) is used to verify the functioning of the algorithm. V-REP robot test system contains coordinated and advanced environment settings,

Manuscript received March 2019; revised November 14, 2019.
The authors are with Department of Mechanical Engineering IIT Madras, Chennai-600036 India (e-mail: sahoo.sudeepta71@gmail.com, pvm@iitm.ac.in).

doi: 10.18178/ijmlc.2020.10.2.948

where each module or object can be individually controlled by an embedded script as well as remote-API client. Therefore, V-REP ensures an extremely adaptable and suitable virtual environment for testing multi-vehicle applications like navigation and control. Through remote API, vehicle control can be achieved by using third party software or languages like C/C++, Python, Matlab, etc. We have used MATLAB to remotely control the speed of the vehicle in V-REP.

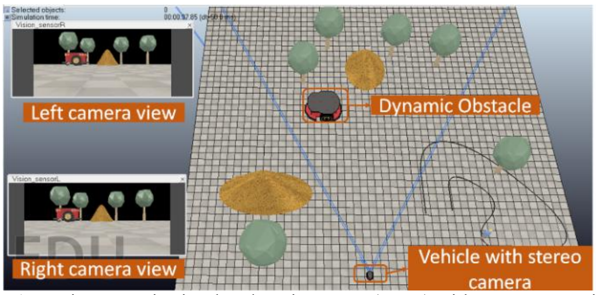


Fig. 1. Environment in simulated environment (v-rep) with two camera view.

P3DX, which is a standard autonomous vehicle is used for the experiments. Two P3DXs have been used, one as an autonomous vehicle that is navigating the environment and the other as a dynamic obstacle as shown in Fig. 1. Two cameras (stereo camera) with a pixel resolution of 512×256 are used to test the algorithm. The developed code in MATLAB obtains the image from the stereo camera and using these captured images, detects the dynamic obstacle.

III. Experimental Setup

For experimental validation of the developed algorithm, Amigobot is used as the autonomous vehicle and a P3DX is used as the dynamic obstacle as shown in Fig. 2. Both the amigobot and the P3DX are controlled using MATLAB via Aria plugin. The same code that was used in the simulation is validated by using it for the experiment. Two low-cost (Logitech) camera are used for the development of stereo vision system as seen in Fig. 2.

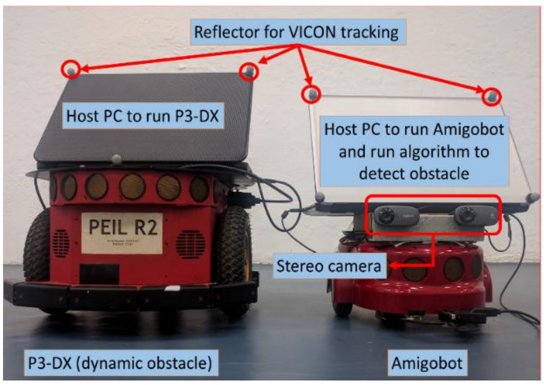


Fig. 2. Experimental setup.

To validate the displacement and speed obtained from the algorithm, Vicon tracking System is used. The system captures the motion parameters of the object accurately. We used a set of four BONITA cameras to capture the motion of the reflectors that have been placed on the autonomous vehicle and the obstacles as shown in Fig. 3. The reflectors can be observed in Fig. 2. The entire test setup with the cameras, amigobot, P3DX, and the static obstacles appear on the Vicon interface as seen in Fig. 4. Using a remote connection between Vicon software and Matlab, we can track and obtain the motion parameters of vehicles as shown in Fig. 5.

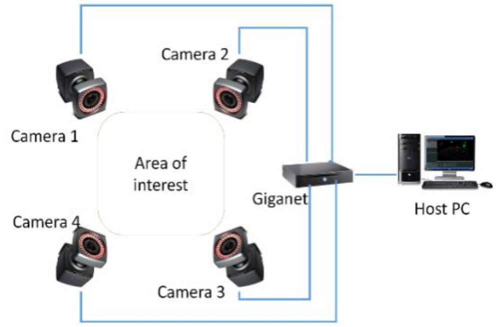


Fig. 3. Vicon camera setup.

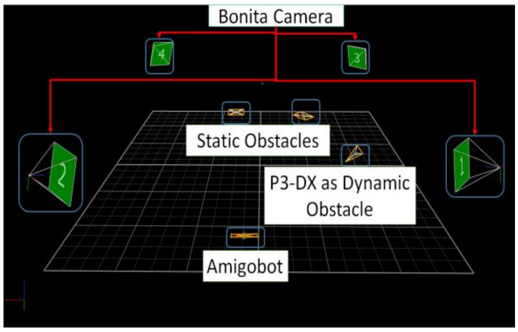


Fig. 4. Vicon system environment.

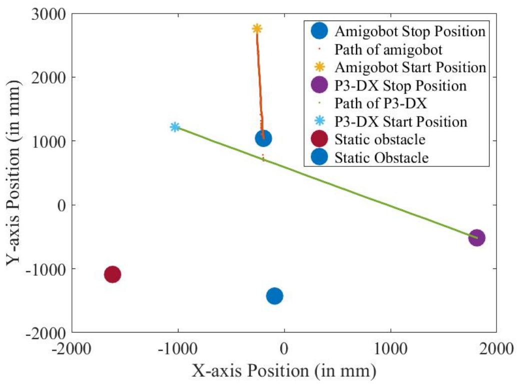


Fig. 5. Tracked object by Vicon system.

IV. Procedure

Motion of the camera causes a dynamic environment due to the motion of the surroundings with respect to the frame of reference of the camera. Observing from the global reference frame, all points except dynamic obstacle looks stationary. We can apply optical flow from a global reference frame for detecting a dynamic obstacle but for the navigation of an autonomous vehicle, we require the dynamic obstacle to be detected locally in the environment. Application of the optical flow method on in-camera reference frame is difficult and is solved by applying image transformation from the current frame to the previous frame. After coordinate transformation, we can now apply optical flow method to these two frames to detect the dynamic obstacle. The procedure to obtain the above-mentioned results are mentioned below:

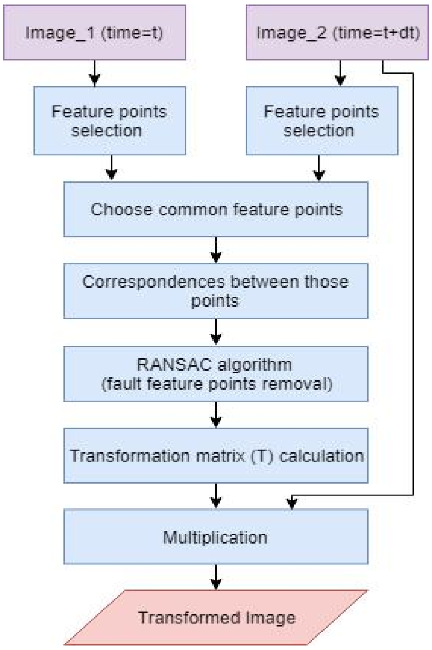


Fig. 6. Image transformation procedure.

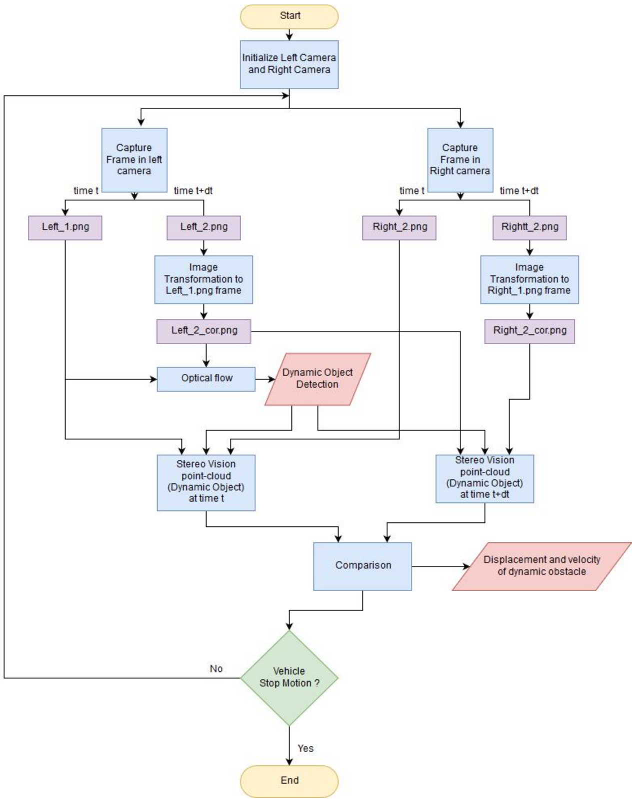


Fig. 7. Algorithm flowchart.

A. Image Transformation

Images are captured using live streaming stereo vision system at regular time intervals (or frame). We then transform the image captured at time '*t+dt*' to the frame of reference of the image captured at time '*t*'. In order to perform this image transformation, a transformation matrix should first be calculated. To calculate the transformation matrix, we need some common feature points between the two images that are considered to be stationary in the global reference frame. A variety of features points are available to choose from a few of which are described below:

1) Corner Points – Corner of objects in images are good feature points because a drastic change of pixel intensity is present.
2) BRIS Points – Binary Robust Invariant Scalable Key points
3) SURF Points – Speeded up robust features
4) KAZE Points – Kaze is derived from a Japanese word which describes wind
5) MSER Regions – Maximally stable external Regions

It is advantageous to choose the corner points method due to its simplicity and less computational cost. There are some algorithms which are used to detect Corner points:

1) Feature from accelerated segment test (FAST) algorithms.Minimum
2) Eigenvalue Algorithm
3) Haris-Stephens Algorithm

The method chosen for the purpose of this research is the FAST algorithm which makes use of the specific feature values of the feature points. The feature values depend upon the grayscale intensity difference between the point and its neighborhood. The next step is to find out the correlation between common feature points of the two captured images in time by making use of the which can be solved by matching the feature values between those feature points. Using these common feature points, we can now estimate the transformation matrix by applying the RANSAC Algorithm [8]. Out of the obtained feature points, not all of them can be used directly due to the presence of a few faulty points. The RANSAC algorithm can estimate the transformation matrix by eliminating these faulty feature points. This transformation obtained is known as an Affine transformation.

The simplest kind of image transformation is a linear transformation, and all linear transformations are Affine having the following two properties:

1) A straight line will be straight after transformation.
2) The parallels line stays parallel after transformation

Six parameters are required to define the affine transformation shown below in the form of equations:

$$x' = t_{11}x + t_{12}y + t_{13}$$
$$y' = t_{21}x + t_{22}y + t_{23}$$
$$\Rightarrow \begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} t_{11} & t_{12} & t_{13} \\ t_{21} & t_{22} & t_{23} \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

Using the above six parameters (t_{ij}), various types of coordinate transformations can be achieved. Table I shows the relation between parameters and the different transformation techniques (where $C(\alpha)$ = Cos(α), S(α) = Sin(α)):

TABLE I: RELATION BETWEEN PARAMETERS AND THE DIFFERENT TRANSFORMATION TECHNIQUES

Operation	t_{11}	t_{21}	t_{12}	t_{22}	t_{13}	t_{23}
Translation	1	0	0	0	Δ_x	Δ_y
Rotation	C(α)	-S(α)	S(α)	C(α)	0	0
Shear	1	Sh_y	Sh_x	1	0	0
Scale	S_x	0	0	S_y	0	0
Horizontal Reflection	-1	0	0	1	X_c	0
Vertical Reflection	1	0	0	-1	0	Y_c

These parameters can now be used to apply the affine transformation on all feature points between the two captured images using the transformation matrix. Fig. 8 shows the schematic of the above transformation process. The first image (Fig. 8(a)) is the image captured at time ‘t’ and the second image (Fig. 8(b)) is the image captures at time ‘t+dt’. There is clear change in the geometry of the object i.e. rectangle becomes a random quadrilateral. Using the corner points of the rectangle as feature points and their affine transformation matrix (6 parameters) we can obtain the transformed image. Therefore the quadrilateral again looks like a rectangle.

Fig. 9 and Fig. 10 show this image transformation operation in simulated and practical environment respectively. In this case, the vehicle (where camera mounted) moves only in the forward direction, so the transformed image looks smaller than the original image. This transformation can be observed by the decrease in pixel size therefore causing the boundary black portion. Slight rotation of the image can also be observed from the images.

Above method have a limitation of the only alteration of image 2d plane. Thus this method is very ill-suitable to finding general destruction taken place in the 3D scene. For which it is not a good option for image transformation in moving vehicle condition. But if two nearest frames of video taken, then there is a very little 3D destruction capture in camera which can be neglected. So for this paper, we ignore this 3D destruction in the environment.

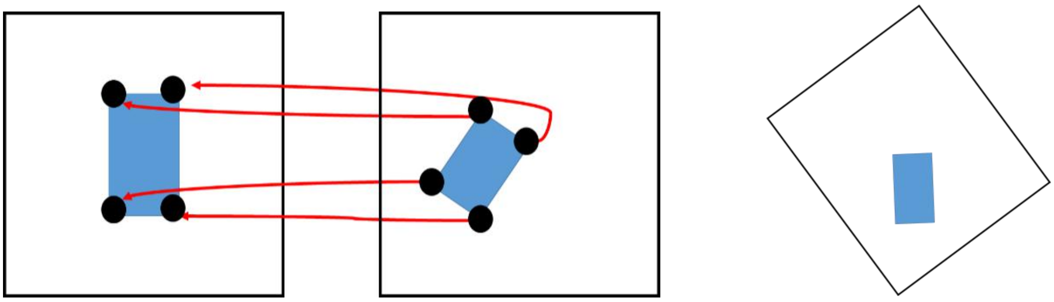


Fig. 8(a). Line diagram at time = *t*. Fig. 8(b) Line diagram at time = *t+dt*. Fig. 8(c) Line diagram after image transformation.

Fig. 8. Line diagram for understanding image transformation.

Fig. 9(a). Image at time = *t*. Fig. 9(b). Image at time = *t+dt*. Fig. 9(c). After image transformation.

Fig. 9. Image transformation in simulation.

Fig. 10(a). Image at time = *t*. Fig. 10(b). Image at time = *t+dt*. Fig. 10(c). After image transformation.

Fig. 10. Image transformation in real time application.

B. Dynamic Object Detection

After the application of image transformation, even though the vehicle and the camera are in motion, all the static obstacles will appear static. Only the actual dynamic obstacles will appear as a dynamic obstacle. Our objective now is to ensure we can detect these dynamic obstacles. Optical flow algorithm which follows the locus Kanade method is used to detect the dynamic obstacle between the first image (at time=*t*) and transformed the second image captured at time ‘*t+dt*’.

This procedure is similar to the previous method which begins with the selection of good feature points and using their feature values. A feature matching algorithm is then applied using KLT method [9] applied to the images. The relative positions between these feature points can then be obtained as shown in Fig. 11. It is observed that the few feature points remain stationary (no relative movement) while other feature points having motion (some relative movement). These feature points that have relative movement provide the position of the dynamic obstacle.

C. Stereo Vision

The stereo vision system is used to obtain the depth map by using two identical cameras. The depth calculation depends on the disparity of a point. Disparity is the relative displacement of a point which is visible in a binocularly visible region, by keeping one camera on at a time. Depending upon the camera configurations and placement, we experience two types of disparity: Horizontal Disparity and Vertical Disparity. If the cameras are identical and are positioned horizontally, then there will be no vertical disparity. There are also nonlinearity errors in a stereo system and these can be avoided by calibrating the stereo vision system to obtain the corresponding intrinsic and extrinsic parameters. These parameters are used at each frame to calculate the disparity, thereby leading to depth generation and point cloud mapping [10].

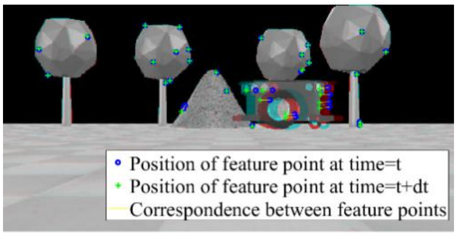


Fig. 11 (a). Simulation result for optic flow.

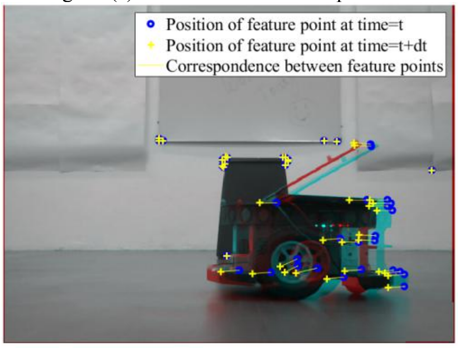


Fig. 11 (b). Experimental result for optic flow.
Fig. 11. Correspondence between feature points of two conjugative frame.

With using the above-explained stereo vision algorithm, two-point clouds environments prepared. One with images taken by two cameras at time=*t*. Another one with the image formed by image transformation of two images taken at time=*t*+*dt*.

D. Obstacle Speed Calculation

Take pointclouds value of detected dynamic feature points of previously prepared two set of point clouds (from images at time=*t* and from transformed images at time=*t*+*dt*). As image transformation done both point clouds have stereo camera origin at same point. Static points having same value of point clouds and dynamic points are having different. So the distance between those two point clouds of dynamic points gives the displacement occurs at that span of time. As time period of each examined frame known, speed can easily be calculated.

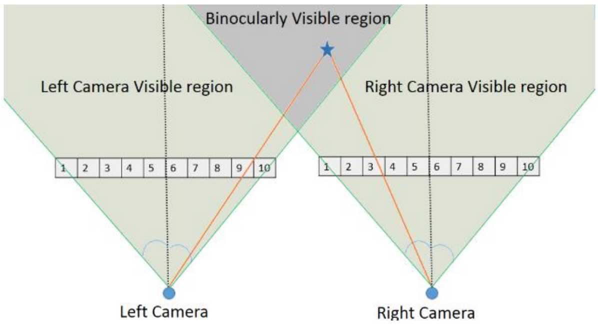


Fig. 12. Stereo vision principle.

V. Result and Discussion

The verification of the developed algorithm to detect a dynamic obstacle in a virtual environment is shown in Fig. 13 and the experimental validation is shown in Fig. 14. The green star points above vehicle describe the chosen feature points that help us distinguish between the dynamic obstacles and the other stationary obstacles. The yellow rectangular box shows the detected dynamic obstacle in the image.

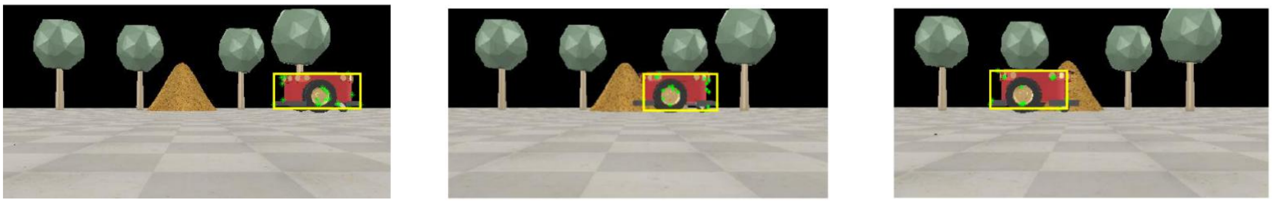


Fig. 13. Dynamic obstacle detection in V-REP.

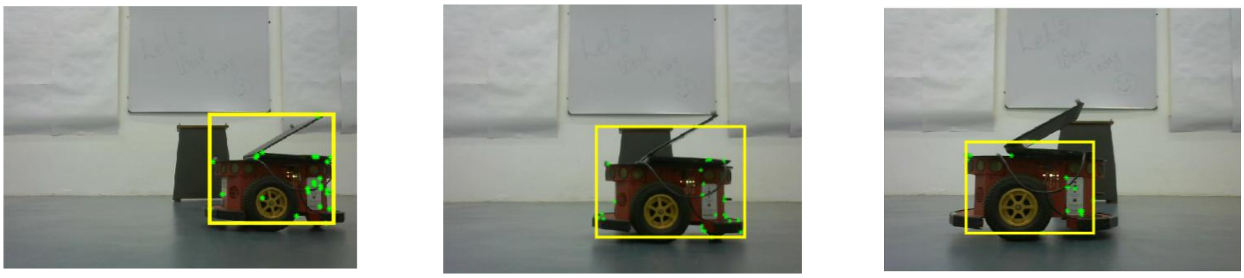


Fig. 14. Dynamic obstacle in real-time application.

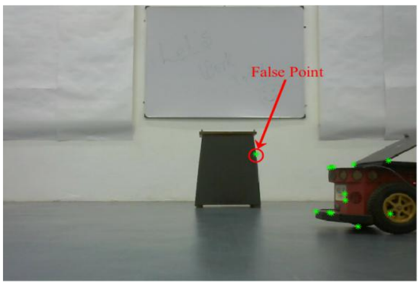


Fig. 15. Detection of the non-moving feature point.

Fig. 15 shows the error in the algorithm, wherein a static point was also detected as a dynamic obstacle. This occurs when the background feature points have a significantly large optical flow value. This issue is a drawback of the image transformation method. This technique works well when there is a change in the image in two dimensions but fails in a three-dimensional geometry change. The above problem can be solved by making use of a high-speed camera which will reduce the changes in geometry in 3D.

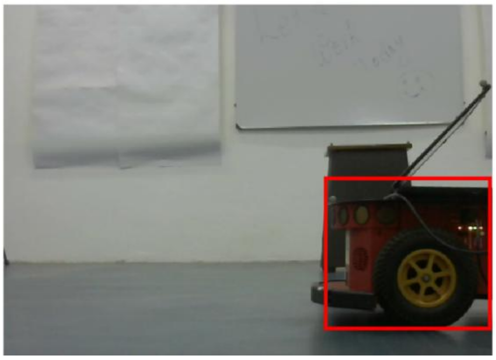


Fig. 16(a). Detected Obstacle in the image.

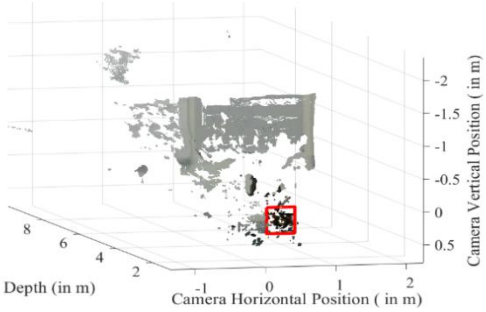
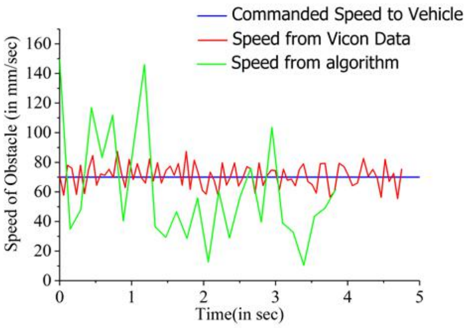
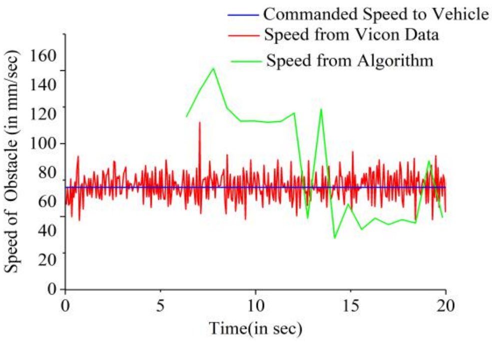


Fig. 16(b). Representation of detected obstacle in the 3D point cloud.



(a). Experiment 1



(b) Experiment 2

Fig. 17. Comparision of speed dynamic obstacle.

Fig. 16(b) shows the 3D point cloud representation of the whole environment viewed by the stereo camera. The Origin (0,0,0) defines the position of the stereo camera. The red box seen in the same Fig. shows the detected dynamic obstacle, for which the point cloud position is known. Displacement and speed are calculated by the previously described method. Fig. 17 shows the comparison between the speed obtained from the algorithm, Speed from Vicon data and commanded speed in two experiments having a different direction of motion. As P3-DX is a standard vehicle having very high stability, it is assumed that its speed is same as commanded speed.

The errors in the results are caused by majorly due to the following reason:

- Point cloud developed by stereo vision is not accurate due to calibration error.
- Matching of the point depends on pixel intensity so when more than one point has the same property, then stereo vision fails.

VI. Future Scope

- 3D depth information can find out by using single camera using monocular stereo so one camera can use instead of two cameras.
- The developed algorithm only talks about the detection of dynamic obstacles but the navigation is yet to be performed using this information.
- This experiment was carried out using a ground vehicle and the algorithm can also be tested on aerial vehicles.
- Computational complexity is very high for this process. With the use of a different computational method, the computational cost can decrease.

VII. Conclusion

With the use of several image processing methods such as Image transformation, optical flow and stereo vision, an algorithm was developed to detect a dynamic obstacle with a dynamic camera condition. This algorithm has been verified in a virtual environment using V-REP and also validated experimentally on a P3DX and an amigobot. The detected dynamic vehicle's position in the 3D environment is found out using the Stereovision algorithm. From this obtained position of the dynamic obstacle, its velocity profile is also found out and verified using Vicon data reading. Although the above method is verified experimentally, the computational time can be minimized by using optimization technique.

Conflict of Interest

The authors declare no conflict of interest.

Author Contributions

The research work is carried by Sudeepta Ranjan Sahoo under the guidance and support of P. V. Manivannan. Both the authors have equally contributed for this paper. While the experimentation, collection of data and writing the paper part is done by Sudeepta Ranjan Sahoo, evaluation, analysis and review is done by P. V. Manivannan.

References

[1] B. Peasley and S. Birchfield, "Real-time obstacle detection and avoidance in the presence of specular surfaces using an active 3D sensor," in *Proc. 2013 IEEE Workshop on Robot Vision*, 2013, pp. 197–202.

[2] K. Huh, J. Park, J. Hwang, and D. Hong, "A stereo vision-based obstacle detection system in vehicles," *Optics and Lasers in Engineering*, vol. 46, pp. 168–178, 2008.

[3] X. V. M. L. Rivera, R. M. D. Cadubla, J. M. Alemania *et al*., "Depth map characterization of RGB-D sensor for obstacle detection system," in *Proc. 8th International Conference on Humanoid, Nanotechnology, Information Technology, Communication and Control, Environment and Management,* 2016, pp. 1–5.

[4] R. Cucchiara, C. Grana, M. Piccardi, and A. Prati, "Detecting moving objects, ghosts, and shadows in video streams.," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 25, pp. 1337–1342, 2003.
[5] K. S. Sharma, S. R. Sahoo, and P. V. Manivannan, "A hybrid vision system for dynamic obstacle detection," *Procedia Computer Science*, 2018.
[6] A. Litvin, J. Konrad, and W. C. Karl, "Probabilistic video stabilization using Kalman filtering and mosaicing," in *Proc. SPIE*, 2003.
[7] Y. Matsushita, E. Ofek, X. Tang, and H. Y. Shum, "Full-frame video stabilization.," in *Proc. 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. I, pp. 50–57, 2005.
[8] K. Lebeda, "Robust Sampling Consensus," master's thesis, Czech Technical University in Prague, 2013.
[9] J. Hariyono, V. D. Hoang, and K. H. Jo, "Motion segmentation using optical flow for pedestrian detection from moving vehicle," *Lecture Notes in Computer Science*, vol. 8733, pp. 204–213, 2014.
[10] F. Zhao and Z. Jiang, "A new algorithm for three-dimensional construction based on the robot binocular stereo vision system," in *Proc. in 2012 4th International Conference on Intelligent Human-Machine Systems and Cybernetics*, vol. 2, 2012, pp. 302–305.

Sudeepta R. Sahoo graduated in mechanical engineering in 2014 from the NMIET, Bhubaneswar. He completed his master in mechanical system design at IIT Bhubaneswar. Sudeepta is currently a Ph.D. scholar in Mechanical Engineering Department at IIT Madras, India. He has done his master project on motion planning for the ground vehicle. Currently, he is working on navigation and control of Swarm Aerial vehicles as a part of Ph.D. research work. His research focus is on visual navigation and communication between autonomous aerial vehicles.

P. V. Manivannan received his Ph.D. in control system for SI Engines from IIT Madras, India and master in applied electronics from College of Engineering, Anna University, Chennai, India. Currently, he is an associate professor at the Department of Mechanical Engineering, Indian Institute of Technology Madras, Chennai, India. He has also taught summer term course "Mechatronic Systems" in University of Nebraska, Lincoln, USA and University of Kaiserslautern, Germany as a visiting faculty. He is also a recipient of prestigious DAAD fellowship and ERASMUS MUNDUS Teaching fellowship. His teaching research interests include mechatronics, robotics, automotive control systems, embedded system design, sensor network.