Image Feature Based Navigation of Omni Directional Robot

Nagadeesh N, Sanjeev Kaushik R, Shailesh U, S Ravishankar, Sudeesh B

***Abstract*—A novel obstacle detection and avoidance algorithm using image features for Omni directional robot (FireBird V) is proposed in this paper. The proposed algorithm uses image segmentation along with edge features to navigate the robot autonomously over ahomogeneous surface without any collisions during its motion.A monocular camera is used to sense obstacles which are present in the vicinity of the robot. Since the designed algorithm employs more features in comparison to similar works carried out in the past, the motion control of the robot becomes dynamic and robust. The algorithm has been successfully trained with minimum test patterns and tested on Omni directional robot FireBird V. The results indicate that the robot autonomously navigates with zero collisions.**

***Index Terms*—Omni-directional, robot, frame, region of interest (ROI), GMM based segmentation.**

I. INTRODUCTION

Robotic Vision is one of the most important areas for realising autonomous navigation of robots. Compared to other on-board sensing techniques, vision based control continue to create much research interest in the area of robot navigation. Thus image processing plays a vital role in the design of vision systems for robots because of its ability to provide detailed information about the environment, which is not possible to be realised using other types of sensors.

The current research [1], [2] in the area of robot navigation usually involved single features such as edge/texture to detect objects in the vicinity of the robot. These methods suffer from poor accuracy while detecting objects, since it employs single feature to detect an object.

Edge detection based path planning using SOBEL operator as reported in [1] assumes the edges formed in the images are due to straight lines while interpolating the edges and further assumes that the images are symmetric which is a serious drawback of this technique. Further the author has experimented in a closed room environment, which does not provide accurate results in a practical external scenario

Manuscript received May 12, 2014; revised May 29, 2014.

Nagadeesh N, student, ECE Dept. Amrita Vishwa Vidyapeetham (University), Bangalore campus. Fax:+918028440092. Email id: nagdeesh@gmail.com.

Sanjeev Kaushik R, student, ECE Dept. Amrita Vishwa Vidyapeetham (University), Bangalore campus. Fax:+918028440092. Email id: sanjeevr93@gmail.com.

Shailesh U, student, ECE Dept. Amrita Vishwa Vidyapeetham (University), Bangalore campus. Fax:+918028440092. Email id: shaileshu2007@gmail.com.

S. Ravishankar, Professor, SM.IEEE,ECE Dept. Amrita Vishwa Vidyapeetham (University), Bangalore campus. Fax:+918028440092. Email id:s_ravishankar@blr.amrita.edu

Sudeesh B, Scientist, Indian Space Research Organisation, Bangalore, India, email id: sudbalan@gmail.com.

The authors in [2] have used Gaussian mixture model (GMM) to segment the images into road and non-road type classification. GMM parameters are trained by selecting a set of images with large variation in the greyscale values of the road and the parameters are trained. The Gaussian distribution with least overlapping between road and non-road regions is considered for training the GMM parameter. Once the GMM parameter is trained, these parameters are not updated. In this method, human intervention in choosing the Gaussian distribution parameters to segment the image plays a major role which is a major drawback in this algorithm.

The motion detection in old film sequences in [3] uses adaptive Gaussian mixture models to detect the motion of objects in a video. The author has accounted for changes in the background by updating the GMM parameters throughout the video sequence. The learning rate is considered as the reciprocal of the current frame number, where the learning rate specifies the time taken to learn the changes in the background. The drawback of this method is for larger duration video sequences, the time required to learn changes in the background is high.

Section II describes image processing techniques for detection and classification of various obstacles that are found in the field of vision of the robot.

In Section III, the construction and control circuit details of FireBird V robot is described.

Section IV gives the flowchart for the autonomous navigation control of the robot.

In Section V, the results of the proposed algorithm are described.

II. PROPOSED ALGORITHM

A. *Pre-processing*

Pre-processing involves the following steps:

(i) Cropping of captured image
(ii) Down scaling by a factor of 4
(iii) Format conversion
(iv) Adaptive histogram equalization
(v) Filtering

The captured image of size (720x480) pixel is cropped by removing the top half portion of the frame from the captured image. This helps in reducing the field of vision. Fig.1 shows the image captured by the camera.

The cropped image is further scaled down by a factor of 4 resulting into an image of size (90x120). Fig.2 shows the cropped and scaled down image. This procedure enhances the computational speed of the algorithm.

ISBN: 978-988-19252-0-6
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

Fig.1 : Captured image

Fig.2 : Cropped image

The captured image is in RGB format. This is converted to gray-scale image using (1). Fig.3 shows the gray-scale image.

$$Y = 0.229*R + 0.587*G + 0.114*B \qquad (1)$$

Fig.3 : Conversion to gray-scale

Equalization is done to enhance the contrast of the image. Contrast Limited adaptive histogram equalization (CLAHE) is used to adaptively equalise the image.

While using Contrast Limited adaptive histogram equalization (CLAHE), a uniform distribution window/tile of size 8x8 is used with contrast limit of 0.01 and is shown in Fig.4.

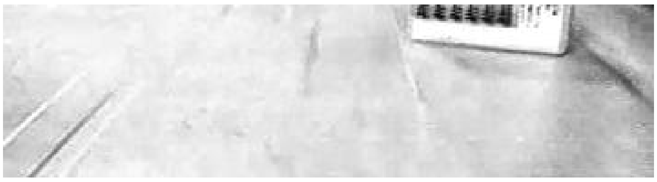

Fig.4 : Result of equalization using CLAHE

Image captured and transmitted by the camera may contain noise. We model this noise as a zero mean Additive White Gaussian Noise (AWGN). To filter out this noise, a Gaussian filter having standard deviation 3 and window size of 24x24 is used.

Fig.5 shows the resultant image after adaptive equalization.

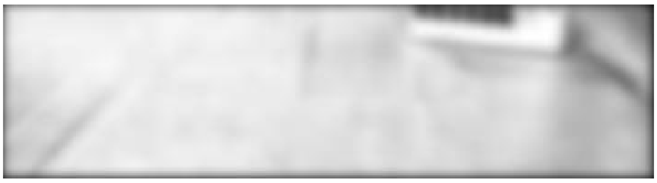

Fig.5 : Filtered image

B. Proposed algorithm

The algorithm uses two important features extracted from the image / video captured during the motion of the robot. The edges present in the image forms one of the features that are filtered adaptively using SOBEL operator. This information clearly indicates the presence of an obstacle.

Fig.6 shows the SOBEL kernal to get the edge information. Gx produces the vertical edges and Gy produces horizontal edges.

-1	0	+1
-2	0	+2
-1	0	+1

Gx

+1	+2	+1
0	0	0
-1	-2	-1

Gy

Fig.6 : Sobel Kernal

Combining the horizontal and vertical images produce the filtered image as in (2).

$$G = \sqrt{G_x^2 + G_y^2} \qquad (2)$$

The edges are identified with the help of a threshold that is dynamically computed using (3) where G_{max} is the maximum value of G.

$$\text{Threshold} = 0.7*G_{max} \qquad (3)$$

The pixels in 'G' having values less than threshold is made zero, the rest are made one. The resultant image contains the edge information present in the original captured image.

Fig.7, shows the edge detection applied on Fig.6.

Fig.7 : Edge detection output

The adaptive GMM [6] of the captured image is used to segment the image into:

(i) Floor information
(ii) Non-floor information

The parameters of the GMM are initialized with

(i) $w_k = 0.2$
(ii) $\mu_k = 0$
(iii) $\sigma_k = 0.0013$

GMM is a multinomial Gaussian distribution having the parameters as in (4) & (5).

ISBN: 978-988-19252-0-6
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

$$P(x_N) = \sum_{k=1}^{K} w_k \phi(x_N; \mu_k, \sigma_k) \quad (4)$$

$$\phi(x_N; \mu_k, \sigma_k) = \frac{1}{\sqrt{2\pi}\sigma_k} e^{-\frac{1}{2}(x_N-\mu_k)^2 \sigma_k^{-2}} \quad (5)$$

Where K=5 indicates the number of partitions, the weight $\mathbf{W_k}$ can be regarded as the probability P(k), which tells the amount of background covered by that distribution. **N** is the current preprocessed frame number.

The above GMM parameters are extracted from the floor image and used for iteratively training the classifier. During the training of the classifier the floor image patterns containing edges of the obstacles in the frame will be discarded. Approximately 50 frames are used to train the classifier initially. This method of training GMM gives better segmentation accuracy.

During the training period, the GMM parameters are updated from each frame of the images using the (6), (7) & (8).

$$\hat{w}_k^{N+1} = \hat{w}_k^N + \frac{1}{N+1}\left(\hat{p}(\omega_k \mid \mathbf{x}_{N+1}) - \hat{w}_k^N\right) \quad (6)$$

$$\hat{\boldsymbol{\mu}}_k^{N+1} = \hat{\boldsymbol{\mu}}_k^N + \frac{\hat{p}(\omega_k \mid \mathbf{x}_{N+1})}{\sum_{i=1}^{N+1} \hat{p}(\omega_k \mid \mathbf{x}_i)}\left(\mathbf{x}_{N+1} - \hat{\boldsymbol{\mu}}_k^N\right) \quad (7)$$

$$\hat{\boldsymbol{\Sigma}}_k^{N+1} = \hat{\boldsymbol{\Sigma}}_k^N + \frac{\hat{p}(\omega_k \mid \mathbf{x}_{N+1})}{\sum_{i=1}^{N+1} \hat{p}(\omega_k \mid \mathbf{x}_i)}\left((\mathbf{x}_{N+1} - \hat{\boldsymbol{\mu}}_k^N)(\mathbf{x}_{N+1} - \hat{\boldsymbol{\mu}}_k^N)^T\right. \quad (8)$$

P(k) is the probability mass function and '1/(N+1)' represent the time taken to learn the background, $\hat{w}_k^N, \hat{\boldsymbol{\mu}}_k^N, \hat{\boldsymbol{\Sigma}}_k^N$, is the current Gaussian paramenters of k^th distribution. $\hat{w}_k^{N+1}$, $\hat{\boldsymbol{\mu}}_k^{N+1}$, $\hat{\boldsymbol{\Sigma}}_k^{N+1}$ are the updated gaussian parameters.

Once the training of the GMM parameter is completed, the segmented image can provide information about the obstacle present in the vicinity of the robot.

The segmentation of image is done using (9) & (10).

$$p(k/xi) = \frac{p(xi/k)p(k)}{\sum_{k=1}^{K} p(xi/k)p(k)} \quad (9)$$

$$p(x/k) = \frac{\exp(-0.5*(x-\mu_k)^T \sum_k^{-} (x-}{\sqrt{|\sum_k|}*(2\pi)^{d/2}} \quad (10)$$

Edge detection [5] and GMM based segmentation is performed sequentially on each processed frame captured. During this phase, obstacles are detected and the GMM parameters are updated dynamically to account for changes in the environment. The update is performed using (11), (12) & (13). **L** is the Learning rate and is chosen as **20** for experimental purpose.

$$\hat{w}_k^{N+1} = \hat{w}_k^N + \frac{1}{L}\left(\hat{p}(\omega_k \mid \mathbf{x}_{N+1}) - \hat{w}_k^N\right) \quad (11)$$

$$\hat{\boldsymbol{\mu}}_k^{N+1} = \hat{\boldsymbol{\mu}}_k^N + \frac{1}{L}\left(\frac{\hat{p}(\omega_k \mid \mathbf{x}_{N+1})\mathbf{x}_{N+1}}{\hat{w}_k^{N+1}} - \hat{\boldsymbol{\mu}}_k^N\right) \quad (12)$$

$$\hat{\boldsymbol{\Sigma}}_k^{N+1} = \hat{\boldsymbol{\Sigma}}_k^N + \frac{1}{L}\left(\frac{\hat{p}(\omega_k \mid \mathbf{x}_{N+1})(\mathbf{x}_{N+1} - \hat{\boldsymbol{\mu}}_k^N)(\mathbf{x}_{N+1} - \hat{\boldsymbol{\mu}}_k^N)}{\hat{w}_k^{N+1}}\right. \quad (13)$$

C. *Validating presence of Obstacle*

In this phase of the algorithm, the field of view is partitioned into two equal halves viz, left region (LR) and right region (RR). If an obstacle is found in LR, the camera setup is moved to capture more information about the obstacle. This ensures the detected obstacles is a true obstacle. The robot turns accordingly to avoid the obstacle. The same procedure is repeated if the obstacle is detected in the RR. This sequence of operation is shown clearly in the state diagram as in Fig. 8.

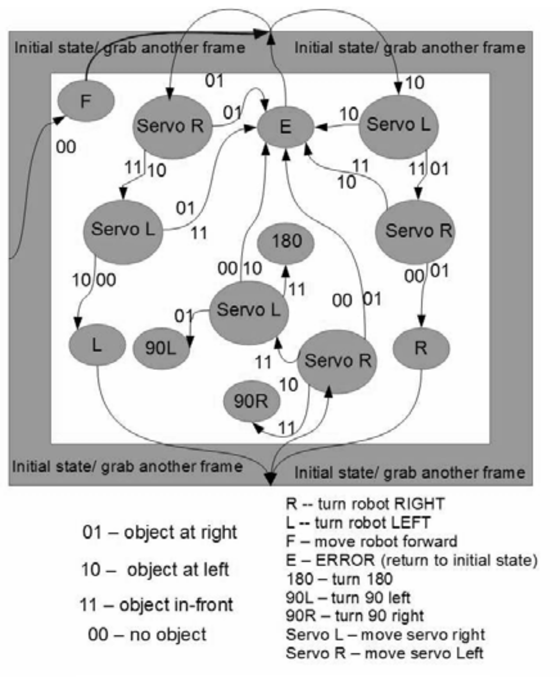

Fig.8 ; State diagram for Validation process

The object code is written in MATLAB and the flowchart used is as shown in the Fig.9.

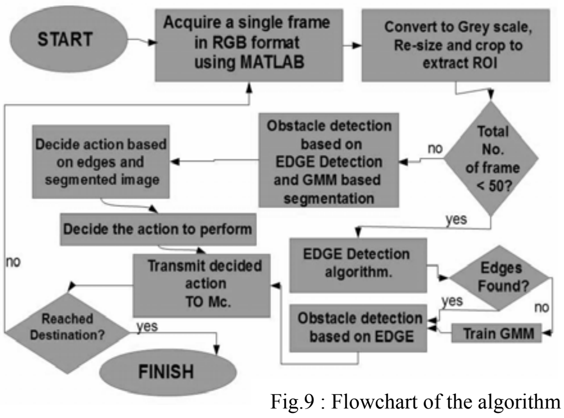

Fig.9 : Flowchart of the algorithm

ISBN: 978-988-19252-0-6
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

III. EXPERIMENTAL SETUP

Fig.10 : Experimental Setup

Fig.10 shows the system used during the experimentation. The robot, FireBird V is an Omni-directional robot with three Omni-directional wheels which are placed at 120^0 with respect to one another. This robot is controlled using Atmega 2560 microcontroller (master), which is an eight bit microcontroller. The robot can be upgraded by adding several sensors and Atmega 8 (slave) microcontroller can be used to interface the sensors and master microcontroller. A wireless colour camera is mounted on the robot and uses an AV transmitter to transmit the images in RGB format to the workstation (here, a computer).

The entire program is source coded in MATLAB R2013a and it is used to acquire and process the image transmitted by the camera. The acquired image is (720x480) pixels. MATLAB functions are used to re-scale, convert the image to gray-scale, and enhance the image and to extract the region of interest for the robot.

The decision made using the above algorithm is sent to the robot using a Wi-Fi module.

The Wi-Fi module receives the data from the workstation in UART protocol at baud rate of 115200, converts it to TCP/IP protocol and transmits wirelessly. The Wi-Fi module present on the robot receives the data as packets and converts it to UART data and sends it to microcontroller Atmel Atmega 2560's UART port. Microcontroller actuates the robot based on the data sent by the workstation.

IV. RESULTS

Initially when the robot starts to navigate it uses only the edge information in the current processed frame and tries to navigate. At the same time GMM parameters are trained as explained in section II.

Few results obtained during this phase are shown in Fig. 11(a) and 11(b). Fig.11(a) represents the frame obtained after pre-processing the image. Fig.11(b) is results obtained by the algorithm on applying the SOBEL operator.

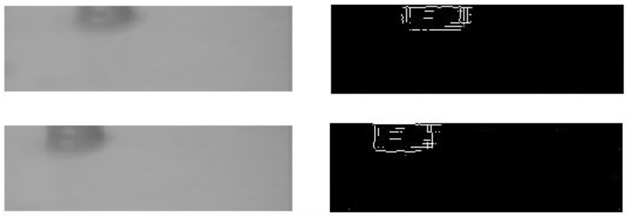

Fig.11(a) : Pre-processed image

Fig.11(b) : Results on applying Sobel operator

After completion of the training period, the algorithm uses both edge information and GMM classified output to navigate the robot as discussed before. Few results obtained when the combination of the algorithm was used are as shown in Fig. 12(a), 12(b) and 12(c).

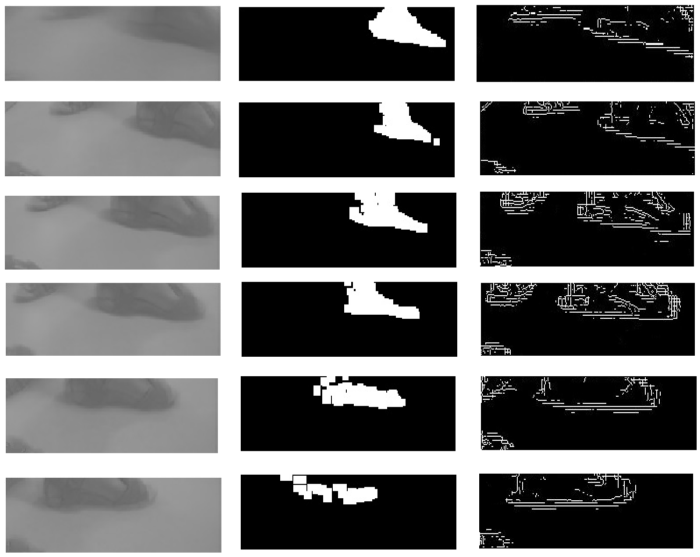

Fig.12(a) : Gray-scale image

Fig.12(b) : Results based on GMM segmentation

Fig.12(c) : Results based on edge-detection

V. CONCLUSION

Since two features of the image are utilized to detect object present in the path/vicinity of the robot and correspondingly in decision making, accurate results are obtained. As discussed in section II, the GMM classifier is trained as adaptively with only the required floor images. All the K distributions capture the variation in the floor and the training is done extensively so as to achieve precise segmentation of floor and non-floor regions once the training is complete. After the initial navigation based on edge information and simultaneous training of the GMM classifier for approximately 50 frames, navigation is carried out using both the methods running simultaneously. Each frame obtained was validated for the presence of obstacles and decision was taken accordingly. On implementation of the algorithm, it was observed that the robot was able to detect and avoid the obstacles in its path.

ISBN: 978-988-19252-0-6
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

ACKNOWLEDGEMENT

We acknowledge the satellite division of Indian Space Research Organization (ISAC, Bangalore), for providing the hardware and the required support to carry out this work.

REFERENCES

[1] Qin Zhang, et.al, "***Single image-based path planning for a spherical robot***," *in 5th IEEE Conference on Industrial Electronics and Applications (ICIEA)*, 15-17 June 2010, pp. 1879–1884, DOI:10.1109/ICIEA.2010.5515441.

[2] Savitha DK and Subrata Rakshit, "***Gaussian mixture model based road signature classification for robot navigation,***" *in IEEE International Conference on Emerging Trends in Robotics and Communication Technologies (INTERACT)*, 3-5 Dec. 2010, pp. 230-233, DOI: 10.1109/INTERACT.2010.5706145.

[3] Xiaoyong Zhang et.al, "***Motion detection in old film sequences using adaptive Gaussian mixture model,***"*in18th IEEE International Conference on Image Processing (ICIP)*, 11-14 Sept. 2011,pp. 2337-2340, DOI:10.1109/ICIP.2011.6116108.

[4] P. Kaewtrakulpong, R. Bowden, "***An Improved Adaptive Background Mixture Model for Real-time Tracking with Shadow Detection***", In Proc. *2nd European Workshop on Advanced Video Based Surveillance Systems*, AVBS01, VIDEO BASED SURVEILLANCE SYSTEMS: Computer Vision and Distributed Processing (September 2001)

[5] G N DeSouza, A C Kak. "Vision for Mobile Robot Navigation: ASurvey". IEEE Trans on PAMI, 2002, 24(2): 237-267Ziou D., Tabbone S. "Edge detection techniques-an overview".

[6] C. Stauffer and W.E.L. Grimson, "Adaptive background mixture models for real-time tracking," in *IEEE CVPR*, 1999, vol. 2.