Proceedings of the 17th World Congress
The International Federation of Automatic Control
Seoul, Korea, July 6-11, 2008

Visual Navigation of a Mobile Robot in a Cluttered Environment ⋆

Meenakshi Gupta * **Balaji Uggirala** * **Laxmidhar Behera** *

* *Department of Electrical Engineering, Indian Institute of Technology Kanpur, Uttar Pradesh, India - 208016.*
e-mail: meenu@iitk.ac.in, uggirala.balaji@gmail.com, lbehera@iitk.ac.in, lbehera@ulster.ac.uk

Abstract: Visual navigation of a mobile robot consists of following tasks (1) object tracking (2) depth estimation and (3) obstacle avoidance. In this paper, we present a modified version of CAMShift algorithm for object tracking, which makes tracking of fast moving object possible. A fuzzy based scheme is used for estimating depth as well as rotation angle necessary for reaching the object. A sonar based obstacle avoidance scheme is also demonstrated. The entire scheme is implemented in real time on Patrolbot, a mobile robot platform from active media robotic.

1. INTRODUCTION

Navigation is the process of planning and controlling the movement of a craft or vehicle from one place to another. Navigation in the context of robot include three basic tasks (1) object tracking (2) estimation of depth and angle of object from the robot and (3) obstacle avoidance.

Object tracking in a cluttered environment is a challenging problem. Difficulties in tracking object can arise due to abrupt object motion , changing appearance patterns of both the object and the scene, and camera motion. Real time applications such as perceptual user interface [Bradski, 1998], surveillance and monitoring and video compression all require the ability to track moving object without hitting any obstacle.

Mean shift algorithm for object tracking suggested by Ramesh [V. Ramesh and Meer, 2000], can only be used for the images with static distribution. CAMShift algorithm [Bradski, 1998] is proposed to track the head and face movement using a one-dimensional histogram (hue) consisting of quantized channel from the HSV color space. Difficulty may arise when one wishes to use CAMShift to track objects where the assumption of single hue cannot be made. To overcome this difficulty, a *Hybrid CAMShift Algorithm* is proposed, which uses all the three channels in HSV plane for tracking the target.

Extraction of accurate position information of the target from the mobile robot is necessary in vision based autonomous mobile robot navigation. Most of the existing classical techniques in the literature are based on either processing a pair of stereo images [Lucas and Kanade, 1981] or a monocular sequence of images taken from a single moving camera [Dalmia and Trivedi, 1996]. The depth estimation techniques using camera model, face either the problems of excessive computational requirements and occluding boundaries or limited accuracy. Soft computing methodologies like fuzzy logic and neural networks can be used for estimating the depth. In this paper a fuzzy based depth estimation scheme is implemented. The proposed scheme uses only the object position information from the camera reading in contrast to [Bien and Park, 1996], where the camera coordinate is also considered.

An essential issue in robotics is to endow the robot with a capability of interaction with the environment. In mobile robot applications, the different types of sensors like Sonar, Laser range finder, Infrared is used to detect the presence of an obstacle in the path. When the obstacle is located, a decision-taking process is carried out in order to choose a new path if necessary [J.R. Lalta, 1998]. In this paper Sonar is used for detecting the obstacle.

The paper is organized as follows. Vision based object tracking algorithms are developed and explained in section 2. Section 3 introduces the fuzzy based depth estimation and determination of rotational angle. Section 4 discuss about the sensor based obstacle avoidance. Real time experiments and results are given in section 5, and finally the paper is concluded in section 6.

2. VISION BASED OBJECT TRACKING ALGORITHMS

2.1 CAMShift Algorithm

The Continuously Adaptive Mean Shift Algorithm (CAMShift) proposed by Bradski [Bradski, 1998], is an adaptation of the Mean Shift Algorithm [Comaniciu and Meer, 1997] for object tracking. CAMShift algorithm operates on a probability density image obtained by histogram back-projection. In CAMShift, the mode in the probability distribution image is detected by applying mean shift while dynamically adjusting the parameters of the target distribution. CAMShift algorithm is enumerated as follow:

(1) Choose the initial location of the 2D mean shift search window.
(2) Calculate the color probability distribution in the 2D region centered at the search window location in an ROI slightly larger than the mean shift window size.
(3) Run the Mean Shift algorithm to find the search window center. a) Compute the zeroth moment as

⋆ This work was supported by Department of Science and Technology (DST), Govt. Of India under the project titled "Intelligent Control Schemes and application to dynamics an visual control of redundant manipulator systems". The project number is DST/EE/20050331. Dr. L. Behera is an associate professor at department of Electrical Engineering, IITK. Currently, he is a reader at school of computing and intelligent system, university of Ulster, Uk.

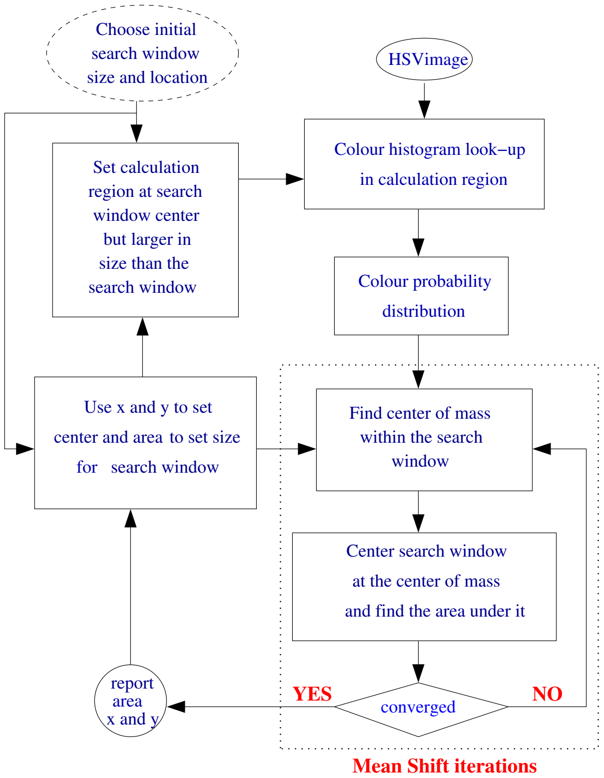


Fig. 1. Original CAMShift Algorithm

$$M_{00} = \sum_x \sum_y I(x,y) \tag{1}$$

b) Compute the first moment for x and y as

$$M_{10} = \sum_x \sum_y xI(x,y) \tag{2}$$

$$M_{01} = \sum_y \sum_y yI(x,y) \tag{3}$$

c) Compute and store the mean location in the current search window

$$x_c = \frac{M_{10}}{M_{00}}\,; y_c = \frac{M_{01}}{M_{00}} \tag{4}$$

Here $I(x, y)$ is the pixel (probability) value at the pixel location (x, y) in the back-projected image and x and y range over entire search window.

(4) For the next video frame center the search window at the mean location stored in Step 3 and set the window's size to a function of zeroth moment found there.

Modifications to CAMShift Algorithm

(1) In the original implementation of CAMShift algorithm which is intended for tracking human faces in video sequences [Bradski, 1998], the backprojected image of only *Hue* channel is used for tracking. But the two remaining channels viz., *Value* and *Saturation* do contain information about the target which can be used for better object detection. In our implementation all the three channel in HSV plane are used for forming the backprojected image.
(2) The inherent problem with CAMShift algorithm is that it can not track a fast moving target. Because it tries to locate the target only in the neighborhood of the previous search window. CAMShift needs an overlap of the object position in two successive frames for efficient tracking. The overlapping condition may not be met in case of fast moving targets. When the target is lost, CAMShift algorithm fixates on some small region of the background. Even if the target were to reappear in the scene at a different location, CAMShift could not detect the target. Once the track is lost, one has to reselect the target.
To remedy this problem the following strategy is used. When CAMShift loses tracking, the current frame is searched for a rough estimate of object location using Mean Matching algorithm (MM) which is described later in this section. Loss of track is indicated by either zero area or very large area. The area of the ellipse (an ellipse drawn around the detected object) is checked for whether it is with in the thresholds. Once it is found that track is lost, the current HSV frame and the target image are passed to the MM. The MM locates the target by matching the target mean. The CAMShift search window coordinates are reinitialized with the target coordinates found by MM.

2.2 *Hybrid CAMShift Algorithm*

The above two modifications has resulted a new tracking algorithm. We call this as **Hybrid CAMShift Algorithm**. The steps involved in the Hybrid CAMShift Algorithm can be enumerated as follows.

(1) The target is selected by the mouse in the first frame of the video sequence. The image under the mouse selection is saved as target image.
(2) The target image which is in RGB format is converted into HSV image and then split into its constituent Hue, Saturation and Value images.
(3) 1D-histograms of the three channels are computed and saved as target histograms. Each histogram is quantized into bins. This reduces the computational and space complexity and allows similar color values to be clustered together.
(4) Now the frame in which the target to be located is split into individual single channel Hue, Saturation and Value images. Histogram back-projection is carried out on these images using corresponding target histogram. A median filter can be applied on this image to smooth out the spurious bright spots.
(5) Instead of using a single back-projected image, various combinations of two or more back-projected images are used for tracking the target. This is done by ANDing two individual binary images (i.e., back-projected images) at a time. The OpenCV function **cvAnd()** is used for this operation. The final back-projected HSV image is used to carry out the CAMShift iterations.
(6) CAMShift operates on the HSV back-projected image using mean shift iterations to detect the target coordinates. If CAMShift fails to locate the target, the current frame will be passed to Mean Matching algorithm.
(7) The Mean Matching Algorithm takes the target image and computes the mean of H,S and V channels and stores it as "Target Mean Vector".
(8) The HSV frame is divided into a number of smaller windows of the size same as target image. The mean of H,S and V channels for each window are computed and matched against the target mean vector.
(9) The coordinates of the center of best matching window will be passed back to CAMShift function. The CAMShift

search window coordinates are reinitialized to this new coordinates in the next iteration.

(10) It may happen that the target is not at all there in the current HSV frame passed to the Mean Matching algorithm. It can be detected by thresholding the "Mean Match Error".

2.3 Mean Matching Algorithm (MM)

In this algorithm, first the RGB image containing only the target to be detected is taken. This image is split into its constituent single channel R,G and B images. The mean of each color channel is computed and stored as *Target Mean Vector* as shown below.

$$T^{\mu} = [R_t G_t B_t]^t \tag{5}$$

Then the image sequence in which the location of the target to be detected is acquired. Each frame is divided into small windows of size 60x60 pixels. The mean value of each color channel in each window is computed and stored as *Window Mean Vector* given as below.

$$W^{\mu} = [R_w G_w B_w]^t \tag{6}$$

Checking for best mean match and centroid calculation:

Now one window in the current frame is considered at a time. The mean of respective color channel is subtracted from the mean of corresponding channel of the target. The square root of sum of mean differences of the three channels is computed. We call this measure as ***Mean Match Error***, simply ***MME***.

$$MME_i = \|(T^{\mu} - W_i^{\mu})\| \tag{7}$$

where i the index of the current window in the frame. The lesser the error, the better is the match. Whole image is traversed while looking for best matching window giving minimum ***MME***.

$$MME_{min} = \min\{MME_i\} \tag{8}$$

The coordinates of the window giving the best match are stored. The center of the best matching window is taken as the centroid of the target.

3. FUZZY BASED DEPTH ESTIMATION AND DETERMINATION OF ROTATIONAL ANGLE

The task of reaching the target comprises of two subtasks:

(1) The robot has to rotate by an angle so that the target appear in the middle of camera view. We call this angle as " Rotational angle".

(2) Once the rotation is over, the robot has to move forward by a distance equal to the distance between the target and the robot.

3.1 Fuzzy based Depth Estimation

It is observed that the distance of the target is a function of the Y-coordinate C_y of the centroid. If the target is farther from camera, it appears in the top portion of the image and C_y coordinate is small. If it moves towards the camera, C_y coordinate will increase. In short, the depth information is embedded in the C_y coordinate of the centroid. This relationship can be used for estimating the distance of the target from robot. We used a fuzzy network for mapping the C_y coordinate to depth [Bien and Park, 1996]. The Y-axis is divided into nine fuzzy regions as shown in the figure 2. Triangular membership functions are defined for these nine fuzzy zones as shown in the figure 3. Data for depth estimation is shown in table 1. A rule base consisting of nine fuzzy rules is formed. The rule base is so chosen that at most two rules will be fired at a time. Defuzzification is carried out by using Centroid method.

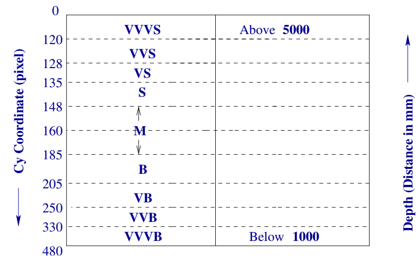


Fig. 2. Schematic diagram for Depth Estimation

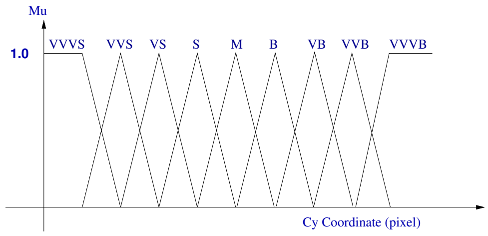


Fig. 3. Membership functions for Depth Estimation

S.No.	Distance in mm	C_y coordinate	Fuzzy zone
1	1000	330	VVVB
2	1500	250	VVB
3	2000	205	VB
4	2500	185	B
5	3000	160	M
6	3500	148	S
7	4000	135	VS
8	4500	128	VVS
9	5000	120	VVVS

Table 1. Data for Fuzzy Depth Estimation

3.2 Determination of Rotational Angle

For determining the angle of rotation, the same methodology for depth estimation discussed in the previous section can be used. By how much angle the mobile robot has to rotate is calculated by using the C_x coordinate of the centroid.

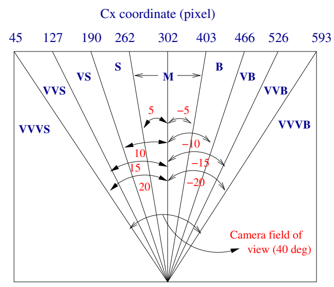


Fig. 4. Schematic diagram for Rotational angle

The robot has to turn by a large angle if the target in the image lies near the edges. The robot has to turn left or right depending upon the location of the target in the image plane. This direction can easily be decided by C_x pixel coordinate of the centroid. If the target is in the first half of the image plane (C_x value will be less than 320), the robot takes a left turn and if the target lies in the second half, it takes right turn. If C_x pixel coordinate lies around 320 no rotation is needed as the target is in the middle of the camera view.

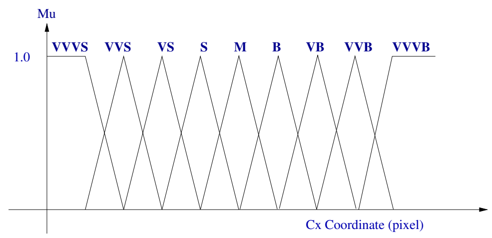


Fig. 5. Membership functions for Rotational Angle

The X-axis of the image plane is divided into nine fuzzy zones as shown figure 4.Triangular membership functions are defined for each zone as shown in figure 5. Data for rotational angle determination is shown in the table 2. A total of nine fuzzy rules constitute the rule base. The rule base is so chosen that at any point of time at most two rules will fire. Centroid defuzzification method is used to get the crisp output rotational angle.

S.No.	Rotation Angle in deg	C_x coordinate	Fuzzy zone
1	-20	593	VVVB
2	-15	526	VVB
3	-10	466	VB
4	-5	403	B
5	0	302	M
6	5	262	S
7	10	190	VS
8	15	172	VVS
9	20	45	VVVS

Table 2. Data for Determination of Rotational Angle

4. SENSOR BASED OBSTACLE AVOIDANCE

Real-time obstacle avoidance is one of the key issues in autonomous mobile robot navigation systems. A mobile robot should be furnished with some kind of collision avoidance behavior so that it can complete the assigned tasks without being a hazard to itself or other. These obstacle avoidance algorithms ranges from primitive algorithms that detect an obstacle and stop the robot short of it in order to avoid a collision, through sophisticated algorithms, that enable the robot to detour obstacles. The latter algorithms are much more complex, since they involve not only the detection of an obstacle, but also some kind of quantitative measurements concerning the obstacle's dimensions. Once these have been determined, the obstacle avoidance algorithm needs to steer the robot around the obstacle and resume motion toward the original target.

Methodology

For detecting the obstacles usually Sonars are used. The Sonars will continuously read the distances of the objects over 360 degrees. The six sonars on the front side of the robot are considered for obstacle avoidance. But we cannot fired all the six sonars simultaneously due to crosstalk problem.

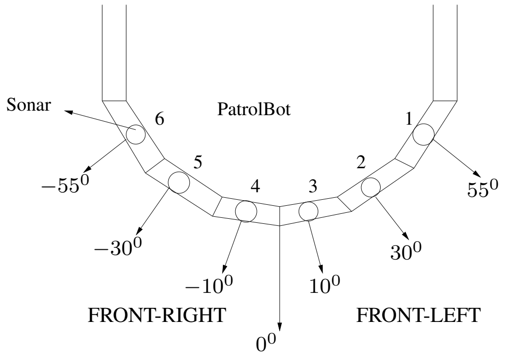


Fig. 6. Sonar Panel on PatrolBot

Time of flight(TOF) ranging system measure the round trip time required for a pulse of emitted energy to travel to a reflecting object and than the echo back of own transmitter. Therefore, if one receiver obtain the echo back of another transmitter, erroneous distance that is shorter than actual value will be observed. This is crosstalk [Emaru and Tsuchiya, 2000]. So, instead of firing all the sonars simultaneously, we adopt a scheduled firing method, where sensors are activated in sequence of (s1,s2,s3...,s6). So, there is no problem of crosstalk. The sonars are divided into two groups, namely

(1) FRONT-LEFT
(2) FRONT-RIGHT

Each group contains three sonars as shown in figure 6.

If the reading of any sonar falls bellow the threshold, we say that sonar has fired. Whenever the sonar on the FRONT-LEFT panel fires, the robot has to turn right to avoid the obstacle and whenever the sonar on the FRONT-LEFT panel fires, the robot has to turn right to avoid the obstacle. The robot has to move forward by a distance d (which is the minimum detectable distance by sonars) after the rotational movement while avoiding the obstacle.

A special case is when the sonars on the both panels fires simultaneously, indicating the presence of the obstacle directly on the front. In such a case the robot would turn 90 degrees and the direction of rotation will be decided by the centroid location of the target in the image plane. The robot has to move forward by a distance d after the rotational movement.

5. REAL-TIME EXPERIMENTS

The real-time experiments are carried out on PatrolBot, a general purpose mobile robot acquired from *MobileRobots Inc.*

5.1 Mobile Robot Setup

PatrolBot is a programmable autonomous general purpose Service robot rover built by MobileRobots Inc [Robotics].

Technical Specifications

PatrolBot has a 59cm x 48cm x 38cm, CNC aluminum body. Its 19cm dia tires handle nearly any indoor surface. The two motor shafts hold 1000-tick encoders. This differential drive platform is holonomic so it can turn in place. Moving wheels on one side only, it forms a circle of 29cm radius. The robot is equipped with *1.6 GHz Intel Pentium* processor and *500 MB* of RAM. PatrolBot is equipped with an Hitachi SH-based 32-bit

RISC micro controller, two 1000-tick motor encoders and other electronics. There are three different types of sensors integrated with PatrolBot, namely

(1) SONAR : Total 16 sonars distributed in two sonar arrays of eight sonar each are so placed that nearly 360 degrees of range sensing is possible. The maximum range of each sonar is 5000 mm.
(2) SICK LMS200 Laser Range Finder : The laser beam being highly focused and not readily distorted or absorbed by the reflecting medium, the precision of range finding with Laser Range Finder (LRF) is much superior to sonar.
(3) Canon VC-C50i PTZ Camera : This robotic camera attaches and extends the capabilities of a variety of video and vision systems and applications. It can pan, tilt and zoom over a wide range.

Software Specifications

A small proprietary μ**ARCS** transfers sonar readings, motor encoder information and other I/O via packets from the micro controller server to the PC client and returns control commands. PatrolBot can be run from the client or users can design their own programs under Linux or under WIN32 using C/C++ compiler. ARIA and ARNL software supply library functions to handle navigation, path planning, obstacle avoidance and many other robotic tasks.

Integration of Vision algorithm and Motion algorithm

The real-time implementation set up comprises of two modules, namely

(1) Vision Module
(2) Motion Module

The strategy is that the Vision module locates the target in each frame. The Motion module reads the centroid coordinates of the target from the Vision module. The centroid coordinate are used to estimate the Depth of the target and Rotational angle. The sonar data is continuously read and obstacles are checked for. If an obstacle is found in its path toward the target, the robot rotates by an angle to surpass the obstacle. Once the path is clear of any obstacle, the robot moves forward to reach the target. The Vision and Motion modules run as two separate threads, accessing the same target centroid information. A locking mechanism is constituted so that at a time only one module can access the centroid information.

Vision Module

The Vision module is composed of the on-board camera and Hybrid CAMShift algorithm. The camera captures the video sequence at rate of *25 fps* with an image size of *640x480* pixels. Hybrid CAMShift algorithm runs on this image sequence to find out the centroid location of the target. This module is run as the main thread.

Motion Module

The Motion module runs as a separate thread. It accesses the the centroid and area information computed by the Vision algorithm. Then it calls the fuzzy depth estimation function to determine the distance of the target from robot. It also determines the angle of rotation by calling fuzzy angle estimation function. Once the depth and angle of rotation are calculated, motion commands are issued to the robot to move.

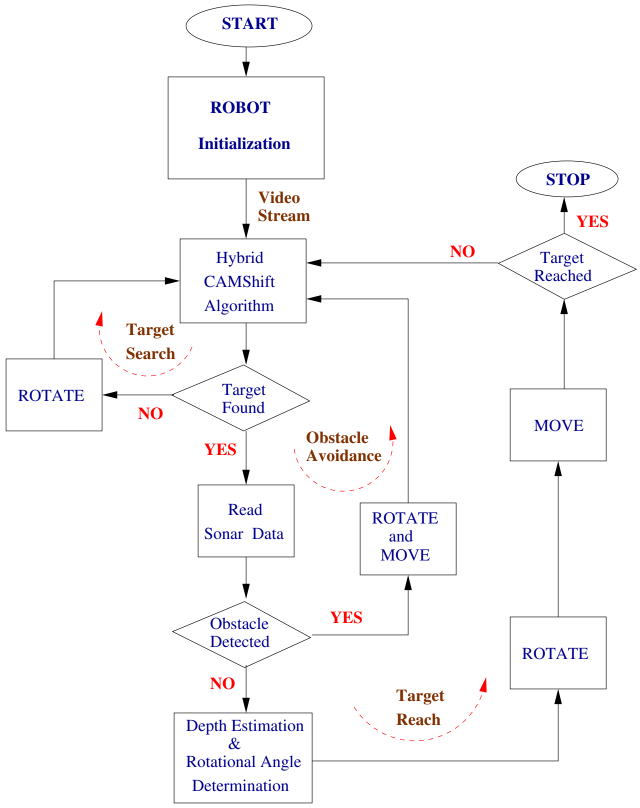


Fig. 7. Flow Chart for Real-time Experiment

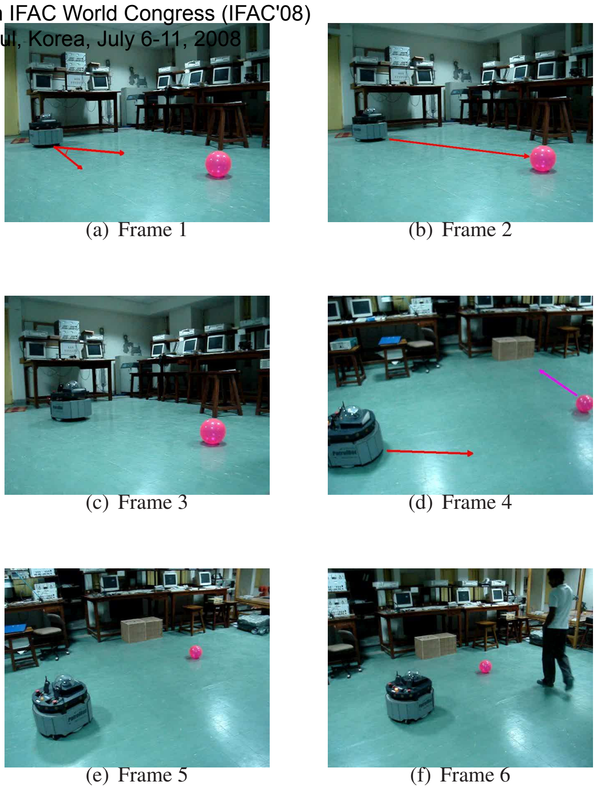
5.2 Priority based Mobile Robot Navigation

The tasks that the mobile robot has to perform can be classified into three main categories.

(1) Target Reaching: Task One
The mobile robot has to reach the target once it locates the same using vision algorithm. The Vision algorithm detects the target and provides the centroid information to the Motion algorithm. Then the Motion algorithm estimates the target distance from the robot and computes how much angle should it rotate before it moves forward to reach the target. Then the Motion algorithm issues commands to the robot first to turn by computed angle and then move toward the target by the estimated distance. We call this as Task-1.
(2) Target Searching: Task Two
If the Vision algorithm fails to locate the target, the mobile robot has to rotate about its own axis till it finds the target in the on-board camera's field of view. Let us call this as Task-2.
(3) Obstacle Avoidance: Task Three
Once the target is located by the Vision algorithm, the mobile robot will try to reach it by first turning and then moving toward the target (Task-1). While doing so, the mobile robot has to detect and surpass the obstacles avoiding any collision. This task we call as Task-3.

The above three tasks constitute the Motion module of the mobile robot. For achieving proper coordination of the above mentioned three tasks, we assign *priority* to each task depending upon its relative importance.

Priority Assignment



(a) Frame 1 (b) Frame 2

(c) Frame 3 (d) Frame 4

(e) Frame 5 (f) Frame 6

Fig. 8. Real-time Experiment Results

Of all the three tasks, we have given Task-2 the highest priority. This is because there is nothing to do with Vision based navigation, if the target is not in camera's field of view. So once the target is *Out Of View*, the mobile robot has to rotate around while the Vision module continuously tries to locate the object elsewhere in the environment. Till this task is finished the remaining two tasks will be suspended from action.

The second highest priority goes to Task-3. The Mobile robot has to detect the obstacles so that it can navigate without hitting any objects in the environment. For detecting the obstacles the robot relies on the Sonar data. The Sonars continuously scan the environment over 360 degrees and the robot reads this sonar data every second. Once the reading of the front panel sonars falls below the set threshold, the motion algorithm suspends the task of target reaching(Task-1) and obstacle avoidance routine will take up the show. In this task, the Motion algorithm makes the mobile robot to turn until the obstacle is completely out of its path. Once the obstacle is overcome, the Motion algorithm will resume the task of target reaching provided that the target is in camera's field of view. Otherwise the task of target search(Task-1) will take over the show.

The least priority is given to Task-1. Once the target is in view and the path is clear of obstacles, then only the task of target reaching is performed.

5.3 Real-time Experiments Results

The demonstration of real-time experiments is shown in figure 8. These are snaps taken while the mobile robot is tracking a pink ball. The red arrows in figure 8(a) shows robot current direction and the direction of the target from the robot. The angle subtended by the two red arrows is "Rotational angle". Figure 8(b) shows the instant when the robot has turned by an angle of rotational and the red arrow shows the current direction of robot. The length of this arrow is equal to the distance between the target and the robot(depth). Figure 8(c) shows the instant while the robot is approaching the target.

In figure 8(d), the red arrow shows the current direction of the mobile robot. The pink arrow is pointing the direction in which the ball is moving at that instant. Figure 8(e) shows the instant when the robot has turned and facing the target. Figure 8(f) give the instance while the robot is approaching the target.

6. CONCLUSION

Vision based object tracking algorithms are developed in the context of Mobile robot navigation. Mean Matching Algorithm for colored object detection is developed. The CAMshift algorithm is modified so that it can be used for mobile robot navigation. This is achieved by integrating the original CAMShift algorithm with Mean Matching algorithm. A Fuzzy Logic based Depth Estimation scheme is proposed for determining the distance of the target from the robot. Proposed Hybrid CAMShift algorithm is implemented in real-time on PatrolBot. Future work will estimate the depth using vision algorithm and avoid obstacle using both sonar and laser range finder.

REFERENCES

Z. Bien and J. Park. Depth estimation for visual servoing based on fuzzy logic. In *Proceedings of the Fifth IEEE International Conference on Fuzz Systems*, volume 2, pages 1043–1047. IEEE, September 1996.

G. R. Bradski. Computer vision face tracking for use in a perceptual user interface. *Intel Technology Journal*, 1998. 2nd Quarter.

D. Comaniciu and P. Meer. Robust analysis of feature spaces: color image segmentation. In *Proceedings of the 1997 Conference on Computer Vision and Pattern Recognition (CVPR '97)*. IEEE Computer Society, 1997.

A. K. Dalmia and M. Trivedi. Depth extraction using a single moving camera: An integration of depth from motion and depth from stereo. *Machine Vision and Applications*, 9(2): 43–55, March 1996.

T. Emaru and T. Tsuchiya. Avoid crosstalk of sonar by using sliding mode system. In *IECON 2000. 26th Annual Confjerence of the IEEE*, pages 2377–2382. IEEE, August 2000.

J.P. Oria J.R. Lalta, E.G. Sarabia. Fuzzy controller for obstacle avoidance in robotic manipulators using ultrasonic sensors. In *IEEE 5th International Workshop on Advanced Motion Control*, pages 647–652. IEEE, 1998.

B. Lucas and T. Kanade. An iterative image registration technique with an application to stereo vision. In *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI '81)*, pages 674–679, 1981.

ActivMedia Robotics. Mobile robots. `http://mobilerobots.com/`.

D. Comanciu V. Ramesh and P. Meer. Real-time tracking of non-rigid objects using mean shift. In *IEEE Conference on Computer Vision and Pattern Recognition*, volume 2, pages 142–149. IEEE, 2000.