J Intell Robot Syst (2007) 48:253–284
DOI 10.1007/s10846-006-9095-8

UNMANNED SYSTEMS PAPER

Mobile Robot Navigation Using Sonar and Range Measurements from Uncalibrated Cameras

A. Tsalatsanis · K. Valavanis · N. Tsourveloudis

Received: 18 October 2006 / Accepted: 30 October 2006 /
Published online: 5 January 2007
© Springer Science + Business Media B.V. 2007

Abstract A novel, simple and efficient method for vision based range measurements with uncalibrated cameras is presented. Required parameters are the image size, the relative distance between two different image frames of the same scene and the field of view of the camera(s). Range measurements acquired using ultrasonic sensors and a vision system have been used to navigate a mobile robot around known colored obstacles in an indoor environment. Both sonar sensors and cameras are activated and they operate simultaneously in parallel to obtain range measurements from common search areas located in the front of the mobile robot. Experimental results using a *parallel stereoscopic*, *rotated* and *monocular* vision system with *uncalibrated* cameras confirm that the maximum computational error (as well as the normalized root mean square error) of range measurements using the vision system for obstacles lying at a distance of 27–800 cm from the robot, is smaller compared to other similar, even more advanced and state-of-the-art existing approaches, reported in Rajagopalan (*IEEE Trans. Pattern Anal. Mach. Intell.,* 28(11): 1521–1525, 2004), Mudenagudi and Chaudhuri (*Proceedings of IEEE International Conference on Computer Vision*, 1: 483–488, 1999), Umeda and Takahashi (*Proceedings of IEEE ICRA*, pp. 3215–3220, April 2000), Jiang and Weymouth (*Proceedings of IEEE CVPR*, pp. 250–255, June 1989), Lai, Fu, and Chang (*IEEE Trans. Pattern Anal. Mach. Intell.*, 14(4):405–411, 1992), Malis and Rives (*Proceedings of IEEE ICRA*, pp. 1056–1061, 2003), Derrouich, Izumida, and Shiiya (*IEEE Annual Conference on IECON*, 3: 2191–2196, Nov. 2002).

Key words depth estimation · navigation · range measurements · sonar sensors · vision system

A. Tsalatsanis · K. Valavanis (✉)
Unmanned Vehicle Systems Lab, Department of Computer Science,
University of South Florida, Tampa, FL 33620, USA
e-mail: kvalavan@csee.usf.edu

A. Tsalatsanis
e-mail: atsalats@mail.usf.edu

N. Tsourveloudis
Technical University of Crete, DPEM, Chania, Crete 73100, Greece
e-mail: nikost@dpem.tuc.gr

1 Introduction

A novel, efficient and robust approach is presented for vision based depth estimation. The method does not require any camera calibration technique that adds computational load to the system, but it is based on the image size, the field of view of the camera(s) and the relative position of two image frames of the same scene. The image size is set by the user, the field of view is known by the camera's manufacturer, while the relative distance can be measured. No additional information is required for a given application.

A case study demonstrates the efficiency of the presented method for indoors mobile robot motion planning and collision avoidance based on simultaneous ultrasonic sensor and camera range measurements used to identify obstacles and navigate around them.

Ultrasonic sensors have the advantage of fast and accurate range measurements, but in certain cases, small surface objects, or objects situated at a wide angle related to the ultrasonic sensor(s), cannot be detected. Given these limitations, a vision system may be used to identify obstacles within the field of view of the robot; in this paper obstacles located at distances up to 8 m are identified accurately. This is achieved using a color space transformation, adaptive thresholding and the proposed method.

The YC_bC_r color space has been used to retrieve the obstacle's color from the acquired image; this choice over the *HSI* and *CMYK* color spaces is justified because the YC_bC_r color space demonstrated better performance in locating the object in question (however, results using all three color spaces are included for comparison purposes).

Both sonar sensors and cameras are activated and operate simultaneously and in parallel to obtain range measurements from common search areas in the front of the mobile robot. The algorithm allows for "back and forth" sonar/camera based obstacle identification, in the sense that although sonar sensors may initially identify a potential obstacle, camera data may be used to complement and decide about the presence/absence of potential obstacles and navigate around them. The implemented vision system consists of two *uncalibrated* rotated color cameras mounted on top of the robot at a distance of 45 cm from each other. Experimental results and comparisons using a *parallel stereoscopic*, *rotated* and *monocular* vision system are presented.

The computational complexity of the camera based range measurement is of order $O(n^2)$, where n is the dimension of a square image, while the complexity of the sonar based range measurement is of order $O(n)$, where n is the number of the sonar sensors.

Experimental validation and verification of the proposed method has been demonstrated using the *ATRV-mini* skid steering differential drive robot, equipped with 350 MHz PIII processor and 256 MB of RAM. The ATRV-mini uses 24 ultrasonic sensors, a GPS, a compass unit and a vision system that consists of two pan/tilt video cameras, SONY EVI-D30. The robot runs *RedHat Linux 6.0* and *Mobility* interface. Obstacles are assumed to be of yellow, red, green, blue, or mixed color. No enhancements or modifications to the original system have been performed. Movement from an initial to a final goal point follows the fuzzy controller framework reported in [18].

Experiments confirm the effectiveness of the presented approach; the maximum computational error as well as the *Normalized Root Mean Square Error, rmse*, of range measurements using either a rotated, parallel, or monocular vision system is about 3.5% (*rmse* 0.023), 4.21% (*rmse* 0.025) and 2% (*rmse* 0.011) respectively, for obstacles lying at a distance of 27–800 cm from the robot. This computational error is considerably better compared to results presented in [5–11]; research in [5–8] deals with stereo vision, while research in [9–11] with monocular vision.

The accuracy and error margin related to sonar sensor range measurements depends heavily on the type of sonars used; however, in all cases, accurate data is returned within a 2 m distance.

The paper is organized as follows. The next Section refers to related work and comparisons. Section 3 describes the vision system algorithm, and Section 4 presents the movement algorithm. Section 5 is dedicated to experimental results. Section 6 discuses the case of monocular vision system (basically for completeness purposes) and Section 7 concludes the paper.

2 Related Work and Comparisons

In general, vision based range measurements require knowledge of the camera intrinsic and extrinsic parameters. A widely used technique to acquire this knowledge is camera calibration where a set of linear equations with 12 unknown parameters needs be solved [25]. This technique is applicable to known or unknown scenes assuming that the intrinsic parameters of the camera do not change for different views. This is a valid assumption when pinhole cameras are used. The modern CCD cameras tend to automatically adjust their intrinsic parameters in order to acquire clear images. Thus, calibration has to be repeated every time the scene or the camera orientation changes.

Reported research in [5] and [6] estimates depth by fusing defocus and stereo with reported *rmse* errors of about 0.12 and 0.34, respectively. In [7], a sub pixel stereo method is proposed with reported computational errors of more than *10%.* In [8], dynamic stereo images are used with reported computational error less than 5%. In [9], a defocus method is used with errors of less than 5%. In [10], the robustness of image based visual servoing is explored with reported errors of 6%. Research in [11] combines a monocular CCD camera and an inertial sensor with an error of about 5.42%. Reported research in [11] includes only simulation results, therefore, no comparison is made.

Regarding depth estimation, the presented approach differs from related work presented in [13] in which range through visual looming is computed using a monocular vision system. A major disadvantage of this approach is that an object's projection must fit entirely within the focal plane, as opposed to our approach where only a part of the object is required for the same computation. Also the distance between the camera and the object is assumed to be measured from the object's center. In our approach, a distance value is calculated for each of the object's corners providing more accurate range measurement especially when the object is not parallel with the focal plane. Furthermore, even no calibration is required in [13] one has to know the physical size of the image plane as opposed to our approach where no such information is required.

The approaches presented in [1–4, 19] do not use the vision system for range measurements. The research in [1] uses vision system information for robot localization and detection of static obstacles through an environment model, and sonar range measurements to detect moving obstacles. The approach followed in [2] for indoor navigation of robot teams uses sonar information to detect walls, open doors and corners through characteristic sonar footprints, while the vision system is activated to describe what has already been detected. Similarly, the work in [3] uses range information from ultrasonic sensors that is transferred to the vision system in order to provide an absolute scale for the image coordinates. In [4], the role of the vision system is to track a color target while the ultrasonic sensors are used for collision avoidance. In [19], two cooperative robots are using visual information to detect landmarks for localization. The work in [23] presents robot architecture able to perform real time vision processing using 4 cameras mounted on the robot. Approaches to robot navigation using range measurements derived from ultrasonic sensors are the topic of [12, 18, 20–22].

The presented approach demonstrates in a very simple, effective and accurate way how data derived from a vision system can be converted to depth measurements without using any camera calibration technique. As a case study sonar and vision system data are utilized simultaneously for robot navigation and collision avoidance in indoors environments.

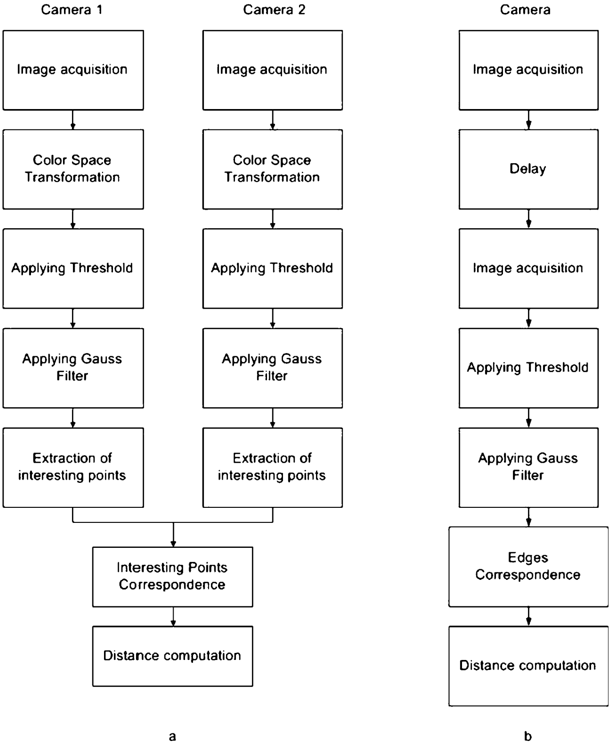
3 Vision System

The vision system of the *ATRV-Jr* consists of two *uncalibrated pan/tilt color cameras* mounted on top of the robot at a distance of 45 cm from each other. The main steps of the vision algorithm to convert image information to range data are shown in Fig. 1a. The case of monocular vision system differs in frame grabbing sequence, meaning that the pair of images is grabbed sequentially as illustrated in Fig. 1b.

3.1 Image Acquisition

Image acquisition is achieved by using the *Video4LinuxAPI* at a rate of 30 fps. Since both cameras share the same *frame grabbing* device, the frame rate is reduced to 14 fps for each camera. Each 24 bit color image has a resolution of 320×240 pixels.

Fig. 1 Block diagram of vision system function



3.2 Color Space Transformation

The YC_bC_r color space is used to retrieve the obstacle's color from the acquired image; it is chosen over the *HSI* and *CMYK* color spaces because it demonstrated better performance in locating the object in question. The transformation from the original *RGB* color space is documented in [14, 15].

3.3 Applying Threshold

A threshold technique has been implemented in the YC_bC_r, *HSI* and *CMYK* color spaces. This is basically done for comparison purposes. Experiments were conducted in an indoors lab environment under different lighting conditions. However the light was uniformly distributed on the obstacles. Images of the same color obstacles obtained in the YC_bC_r, *HSI* and *CMYK* color spaces were used for identification and comparison purposes.

At first, an image containing a yellow obstacle was considered, followed by an image containing a multi color obstacle; the purpose of this second image is to illustrate ability to recognize different colors (color segmentation) under different lighting conditions given that the obstacle is located at various distances from the robot.

3.4 YC$_b$C$_r$ Color Space

The C_b component of the YC_bC_r color space is related to the amount of the blue component. Colors containing high quantity of blue color are represented brighter in the C_b image. Yellow contains a narrow quantity of blue color; it is represented as the darkest of all colors in the C_b image. A threshold value to recognize yellow color is:

$$T = t,\ \text{when } H(t) > h \text{ and } t = \min(i) \tag{1}$$

where T is the threshold value, *H()* the function of C_b's image histogram, h a number of pixel value that shows when an obstacle is considered big enough to participate in threshold calculation, i is the intensity of the C_b component and t is the minor intensity value corresponding to the number of pixels greater than h. Because of light distribution variations, not every pixel (of the yellow obstacle represented in the C_b image) has the same intensity. A threshold area is used to distinguish the obstacle's pixels from the background, its center being the T value with boundaries between T−0.1T and T+0.1T. A new image representing only the object of interest is created according to the equation:

$$g(x,y) = \begin{cases} 255, & T - 0.1T < C_b(x,y) < T + 0.1T \\ 0 & \text{else} \end{cases} \tag{2}$$

g(x, y) is the intensity function of the new image and $C_b(x, y)$ the intensity function of the C_b component. The value of T is computed for each image separately using the image histogram. Thus, the system adapts to illumination variations.

Figure 2 illustrates a color image containing a yellow obstacle (a), the result of the threshold technique (b), and the histogram of a C_b component (c). T is the minimum pick on the histogram and the threshold area is 10% around T.

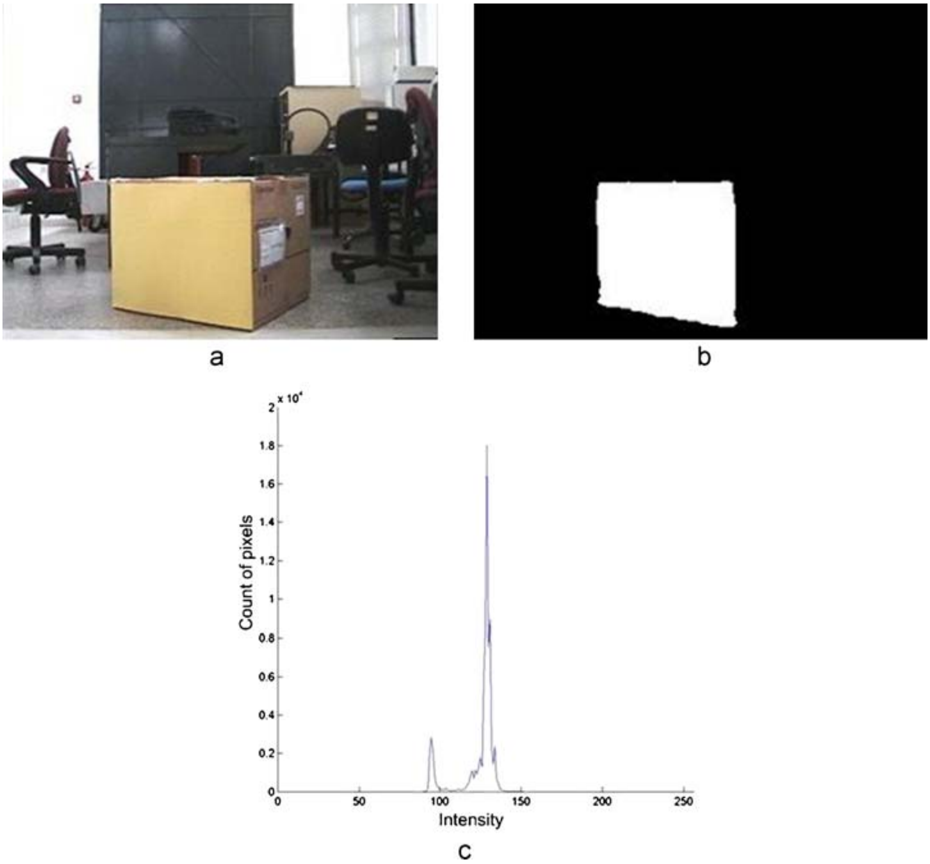


Fig. 2 Applying threshold technique in YC_bC_r color space

However, using the properties of the YC_bC_r color space more colors such as red, blue and green may be extracted from a color image. In particular, red color corresponds to the highest intensity values on the C_r component and to medium intensity values between 130 and 160 on the C_b component. Similarly, blue color corresponds to the highest values on the C_b component and to the lowest values on the C_r component. Finally, green color corresponds to C_b component values in the area between 70 and 100 and to C_r component values in the area between 85 and 120. Figure 3, shows a color image containing the four color obstacle (a), the result of the threshold technique (b) according to Eq. 3, and the histograms of the C_b and C_r components.

$$g(x,y) = \begin{cases} 255, & C_r(x,y) > T_1 \text{ and } 130 \le C_b(x,y) \le 160 \\ 200, & 70 \le C_b(x,y) \le 100 \text{ and } 85 \le C_r(x,y) \le 120 \\ 150, & C_b(x,y) > T_2 \text{ and } C_r(x,y) < T_3 \\ 100, & T_4 - 0.1T_4 < C_b(x,y) < T_4 + 0.1T_4 \\ 0, & \text{else} \end{cases} \quad (3)$$

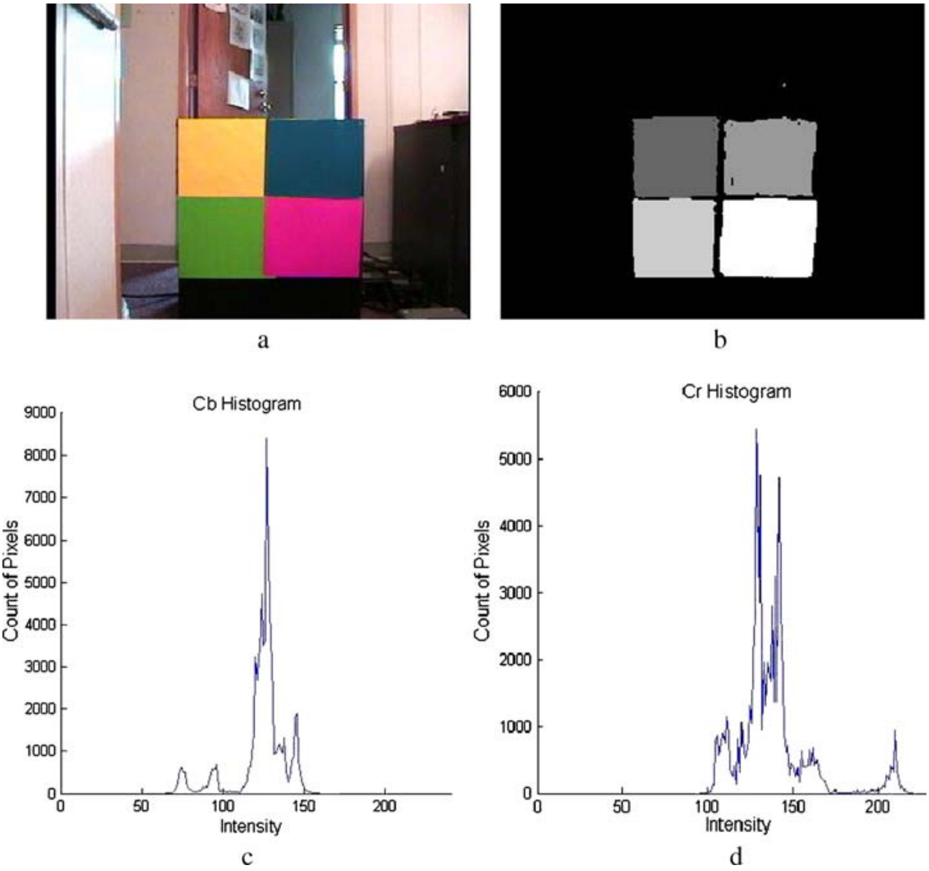


Fig. 3 Applying threshold technique to extract *red*, *green*, *blue* and *yellow* obstacles using the *YCbCr* color space

$C_b(x,y)$ and $C_r(x,y)$ are the intensity functions of C_b and C_r components, respectively. T_1, T_2 are the highest picks with the highest intensity value in C_r and C_b components, respectively, T_3 the lowest pick with the lowest intensity value in C_r component and T_4 as defined in Eq. 1.

Implementation using the *HSI* and *CMYK* color spaces is described in the Appendix.

3.5 Gauss Filter

The thresholded image may have pixels that do not belong to the object of interest. This is because some of the background elements contain a percentage of the color of interest. To eliminate these pixels a [5×5] Gauss filter is used [17].

3.6 Extraction of Interesting Points

In order to extract characteristic points in an image, the Moravec interest operator has been used [17]. The operator computes the directional variances (I_1, I_2, I_3, I_4) for each point and applies an interest value to it. The directions are computed using all pixels in a window centered about the point, as shown in Eq. 4. S represents pixels of a window centered on this point. The window size used is [5×5].

$$\begin{aligned} I_1 &= \sum_{(x,y)\in S} [f(x,y) - f(x,y+1)]^2 \\ I_2 &= \sum_{(x,y)\in S} [f(x,y) - f(x+1,y)]^2 \\ I_3 &= \sum_{(x,y)\in S} [f(x,y) - f(x+1,y+1)]^2 \\ I_4 &= \sum_{(x,y)\in S} [f(x,y) - f(x+1,y-1)]^2 \end{aligned} \tag{4}$$

An interesting value is assigned to each pixel according to:

$$I(x,y) = \min(I_1, I_2, I_3, I_4) \tag{5}$$

The image is then segmented in regions; each region's interest point is defined as the pixel with the maximum interest value.

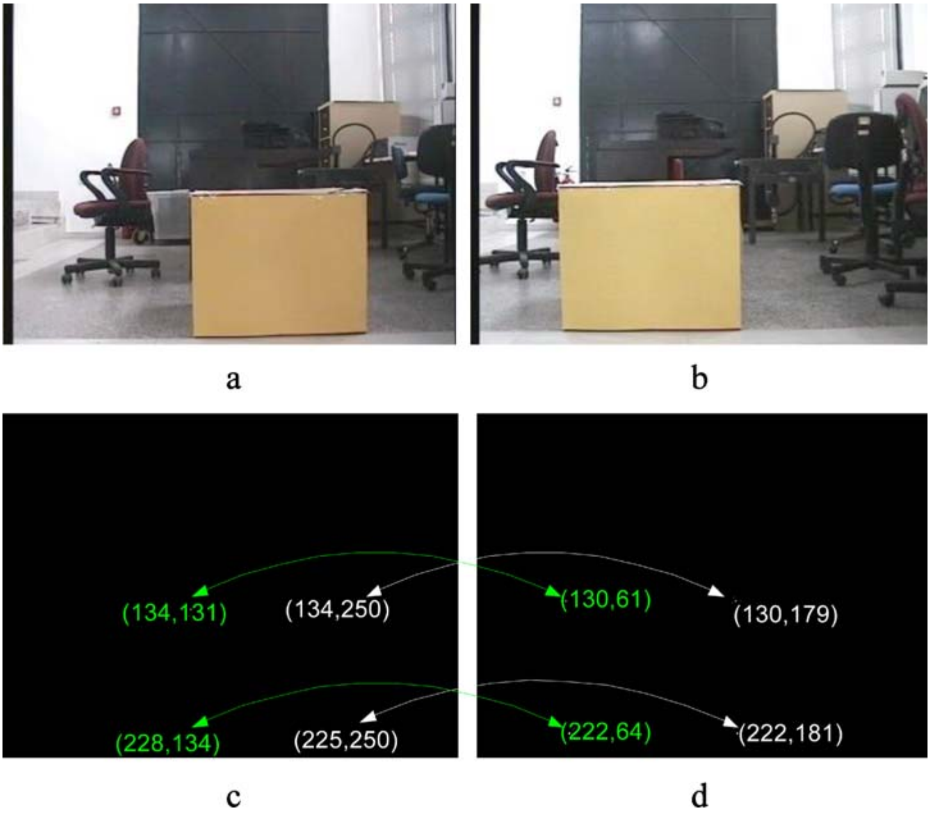


Fig. 4 Interesting point correspondence

3.7 Interesting Points Correspondence

The area of each image that contains the object of interest is segmented in four regions. An interesting point belonging in a region of the first image corresponds to the interesting point belonging in the same region of the second image. The limitation of the method is that it can only retrieve information from the two closest obstacles. However, this method reduces computational time since no correspondence algorithm is needed, which the ATRV-mini's CPU cannot spare. Figure 4 shows the images from left (a) and right (b) camera as well as their corresponding pixels (c).

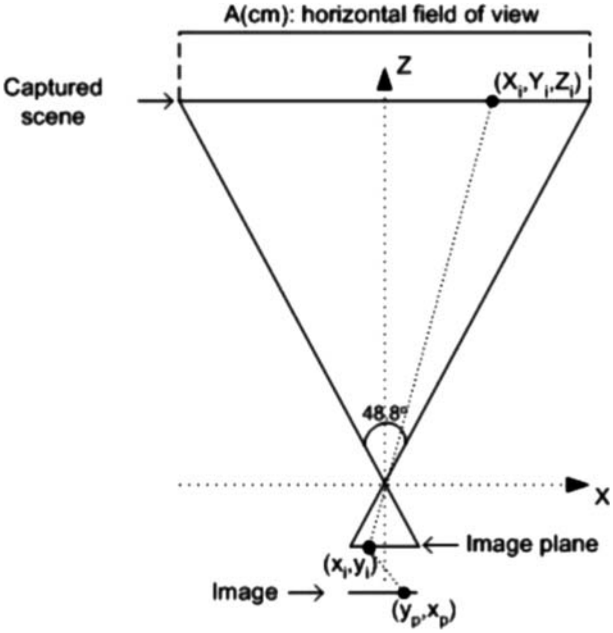
Whereas the interesting point correspondence mentioned above reports acurate results, it depends on the output of the thresholded image. For this reason, a correlation algorithm may be also used. The correlation coefficient, *r*, is defined as the sum of the products of the pixel brightness divided by their geometric mean [24]:

$$r\left(d_x, d_y\right) = \frac{\sum\limits_{(i,j)\in S} f\left(d_x + i, d_y + j\right).g(i,j)}{\sqrt{\sum\limits_{(i,j)\in S} f^2\left(d_x + i, d_y + j\right).\sum\limits_{(i,j)\in S} g^2(i,j)}} \tag{6}$$

where *f*, *g* the left and the right image respectively, *S* a pixels window centered around an interesting point (9×9), and d_x, d_y the disparity of the pair of pixels to be matched. In this case all the interesting points of the right thresholded image derived by the Moravec operator are matched with pixels on the left image.

Both methods have been tested and both provide similar accuracy. For practical reasons, only the first method can be implemented on the ATRV-mini on board computer, in order to get real time results.

Fig. 5 Horizontal field of view of monoscopic vision system



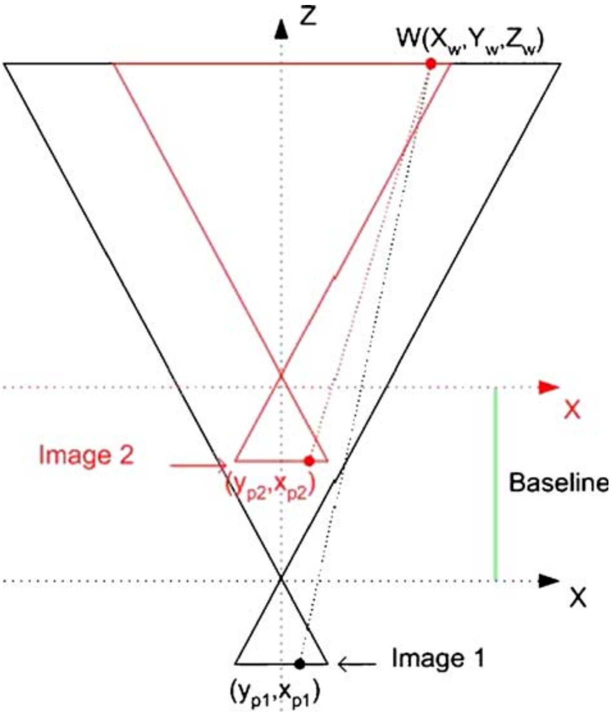


Fig. 6 Stereo from motion

3.8 Distance Computation

The presented method for depth estimation uses only the size of the image, the angle of view of the camera and the relative position of two different captures of the same scene. Figure 5 shows the field of view of a monocular camera, where A is the length in cm of the horizontal field of view in a distance of Z_i cm from the camera system, (X_i, Yi, Z_i) the world

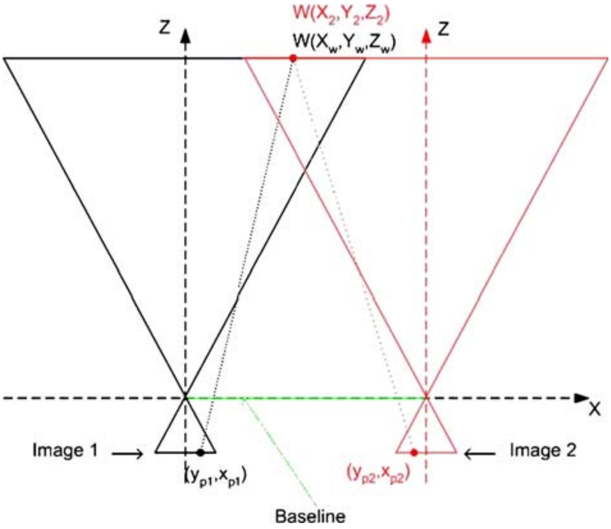


Fig. 7 Depth from stereo vision

coordinates of any point in the field of view, *(x_i, y_i)* the projection of the world point onto image plane and *(y_p, x_p)* its pixel coordinates. The origin of the pixel coordinate system coincides with the center of the image.

For an image of 320×240 pixels, one may observe that:

$$X_i = \frac{2}{320} x_p Z_i \tan(24.4)(cm) \tag{7}$$

A second image containing the same world point is acquired from a different position in order to compute depth. In case of a monocular vision system this is achieved by moving towards the world point (stereo from motion). Figure 6 demonstrates this procedure.

For both images the following equations are derived:

$$Z = \frac{x_{p2} B}{x_{p1} - x_{p2}}(cm) \tag{8}$$

where B is the relative distance between the two shots in cm (baseline) and x_{p1}, x_{p2} the coordinates of the pixel which represents the world point in the first and second image respectively.

For the *parallel* stereoscopic vision system shown in Fig. 7 the depth is derived from:

$$Z = \frac{320B}{2(x_{p1} - x_{p2}) \tan(24.4)}(\mathrm{cm}) \tag{9}$$

From Fig. 7, it is shown derives that the fields of view of the two cameras converge at a distance of 49.6 cm from the vision system. This is the distance of the closest obstacle that the vision system can locate.

To minimize this distance, both cameras are rotated such that their optical axes converge closer (see Fig. 8). In this work, the angle of rotation, α, is set to 12°, reducing the distance of the closest obstacle to 27 cm, which is the length between the front bumper of the mobile robot and the base of the vision system. For the rotating stereoscopic vision system of Fig 8, depth is calculated as:

$$Z = \frac{-\cos^2 a - (D - C)\cos a \sin a + CD \sin^2 a}{(D - C)(\cos^2 a - \sin^2 a) - CD \sin 2a - \sin 2a} B \tag{10}$$

α is the angle of rotation in degrees,

$$C = \frac{2}{320} x_{p1} \tan(24.4) \tag{11}$$

$$D = \frac{2}{320} x_{p2} \tan(24.4) \tag{12}$$

Since the area of each image containing the object of interest is segmented into four regions, there are four pairs of corresponding interesting points and, therefore, four measurements of distance. In case of one obstacle the minimum and maximum measurements are ignored and the distance is defined as the average of the other two. In case of two obstacles or one obstacle located at an angle relatively to the vision system, the distance has

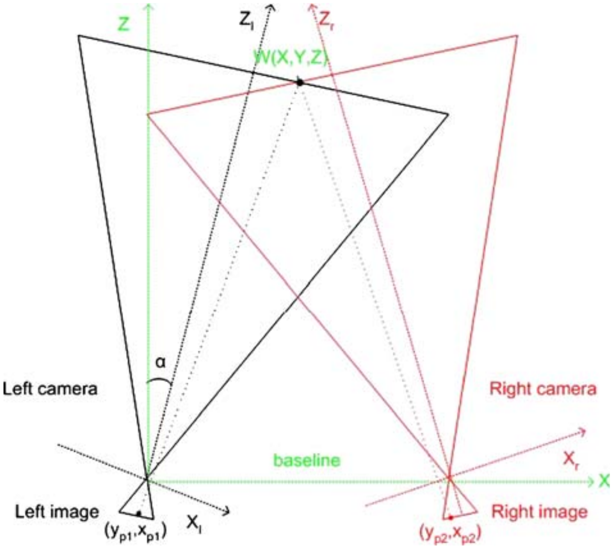


Fig. 8 Rotated stereoscopic vision system

two values and it is defined as the average of the measurements which correspond to the same vertical edge.

3.9 Computational Time

The time required converting vision system data to range measurements using the rotating or the parallel vision system is 0.43 s when running on ATRV-mini's on-board computer, while its speed reaches 0.6 m/s.

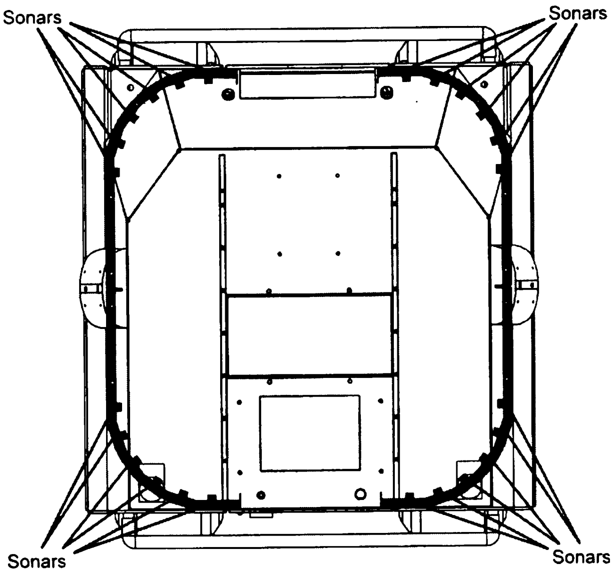


Fig. 9 Location of the sonars on board ATRV-mini

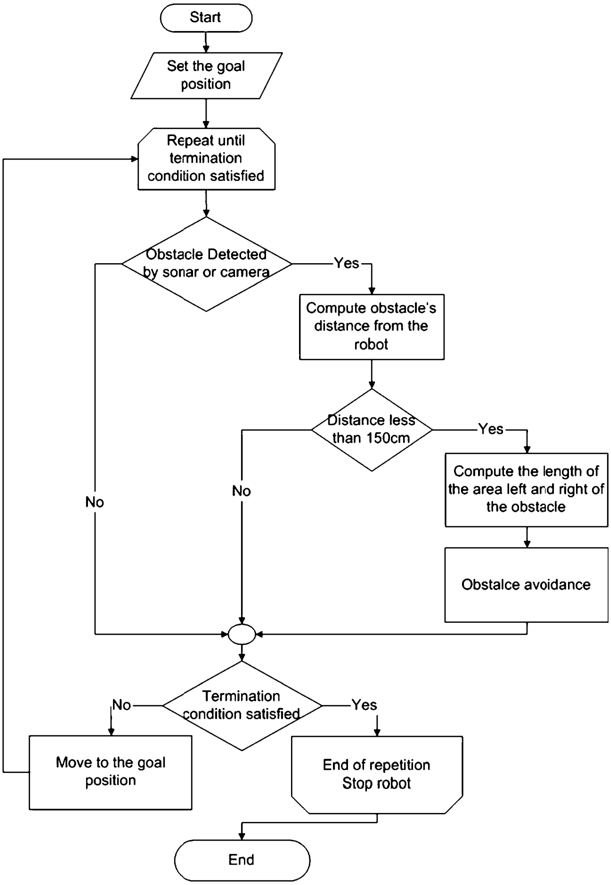

Start
Set the goal position
Repeat until termination condition satisfied
Obstacle Detected by sonar or camera
Yes
No
Compute obstacle's distance from the robot
Distance less than 150cm
Yes
No
Compute the length of the area left and right of the obstacle
Obstalce avoidance
Termination condition satisfied
No
Yes
Move to the goal position
End of repetition Stop robot
End

Fig. 10 The block diagram of motion algorithm

4 Motion Algorithm

The motion algorithm uses range data from both ultrasonic sensors and the vision system for robot motion planning and collision avoidance. Considering the ATRV-mini, Fig. 9, only the eight front ultrasonic sensors have been utilized to derive the motion algorithm. The concept of implemented “collision avoidance” is to calculate the length of the area left and right of the obstacle and to follow the route with larger free space. The cardinal sensor is the vision system which means that in case both sensors detect an obstacle the range measurements are calculated through vision data.

Using data from ultrasonic sensors, the free space calculation is done by comparing range measurement summations from sensors mounted on the robot front left and right. If the summation of the left sensor data is greater than the summation of the right one, the robot moves to the obstacle's left, otherwise, it moves to the obstacle's right, according to:

$$\text{Turn right if}: \sum_{i=0}^{3} U_i < \sum_{i=20}^{23} U_i \tag{13}$$

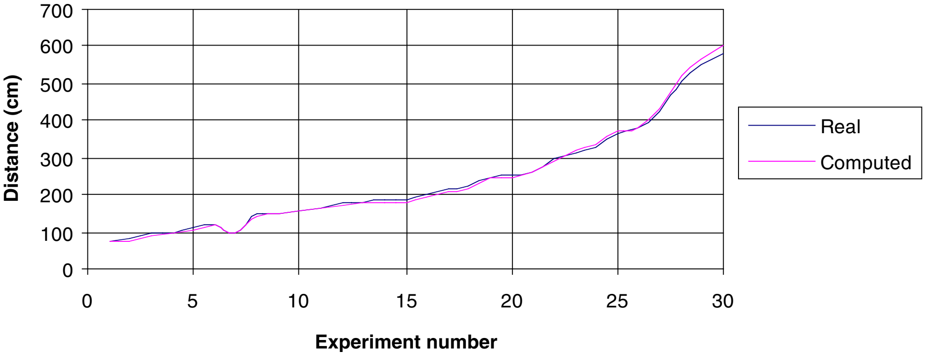


Fig. 11 Real distance vs computed distance using data from the parallel vision system

$$\text{Turn left if}: \sum_{i=0}^{3} U_i > \sum_{i=20}^{23} U_i \tag{14}$$

U_i is the range data in cm from sensor i.

Using data from the vision system, X, the horizontal length of an area corresponding to x_p pixels on a 320×240 resolution image is calculated as (see above):

$$X = \frac{2}{320} x Z \tan(24.4) \tag{15}$$

Z is the distance between the obstacle and the cameras. To calculate the area's length left of the obstacle, x represents the difference in position between the first image pixel and the first obstacle pixel, while for the calculation of the area's length right of the obstacle x represents the difference in position between the last obstacle pixel and the last image pixel such that:

$$X_{\text{left}} = \frac{2}{320}\left(x_{\text{p}} - x_0\right) Z \tan(24.4) \tag{16}$$

$$X_{\text{right}} = \frac{2}{320}\left(x_{320} - x_{\text{p}}\right) Z \tan(24.4) \tag{17}$$

x_0, x_{320} are the first and the last image pixel respectively, and x_{p} the obstacle's pixel.

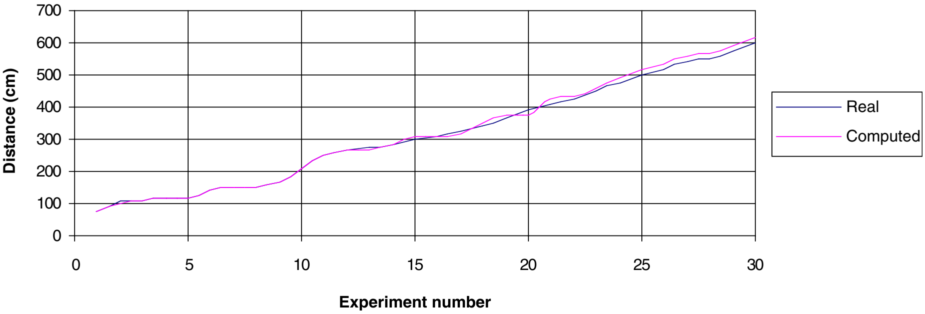


Fig. 12 Real distance vs computed distance using data from the rotated vision system

Since the vision system is under rotation it is preferred to compute the length of the left area using data from the right camera and the length of the right area using data from the left camera. Collision avoidance is completed by the robot's turn to the side with greater length, until none of the sensors detects an obstacle, and forward movement for Z cm. Figure 10 presents the flow chart of the motion algorithm.

To avoid the obstacle, the robot increases its rotational velocity and reduces its translational speed to half until none of the sensors detects the object. This response guides the robot safely left or right of the obstacle. The current position of the robot is entered to the fuzzy controller [18] and a motion vector reinstates the robot to its previous path towards the goal point.

5 Results

For implementation purposes, two types of obstacles have been used: Type I with size 50×60×30 cm and Type II with size 40×30×20 cm. Type II obstacles are colored yellow and cannot be detected from ultrasonic sensors because of their height, which is equal to the distance between the ground and the ultrasonic sensors. Type I obstacles cannot be detected from the vision system since they are not yellow. Multicolor obstacles are also used to illustrate capabilities of the approach in identifying different colors at different distances.

Experiments have been conducted in an indoor lab environment with several "furniture obstacles" of different shapes, sizes and orientation, and with different lighting conditions.

5.1 Vision System Algorithm

Experimental results showed that the threshold technique in the YC_bC_r color space can identify more than 92% of the yellow obstacle without detecting any of the background pixels. On the other hand the techniques implemented in the *HSI* and *CMYK* color spaces did detect most of the obstacle's pixels (more than 70 and 85%, respectively), but they also detected many pixels of the background. Hence, the YC_bC_r choice is justified.

Experimental results of range measurements using a parallel and a rotated stereoscopic vision system demonstrate average errors of 2.26% and standard deviation of 1.1 for the parallel system and 1.58% and standard deviation of 1 for the rotated one. The relation between the actual and the computed distance is illustrated in Figs. 11 and 12.

5.2 Motion Algorithm

The robot's trajectory is demonstrated from an initial point to a final goal point while collision avoidance is achieved using only the vision system. Figure 13 demonstrates avoidance of two yellow obstacles located in the robot's right, suggesting left movement for avoidance.

Figure 14 shows avoidance of two obstacles where the robot turns left because both obstacles are detected from the vision system.

Figure 15 demonstrates avoidance of three obstacles. The first obstacle that lies in the left of the robot, blocks the vision system from detecting the third obstacle. Thus, the robot has to move left since the vision system can detect the second obstacle.

Figure 16 shows another case with three obstacles. The first obstacle is located right of the robot and it blocks the vision system from detecting the other two. Robot moves left and

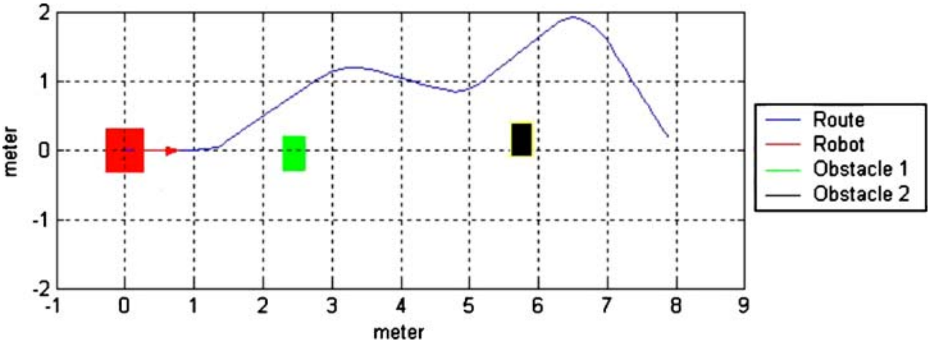


Fig. 13 Collision avoidance using vision system data

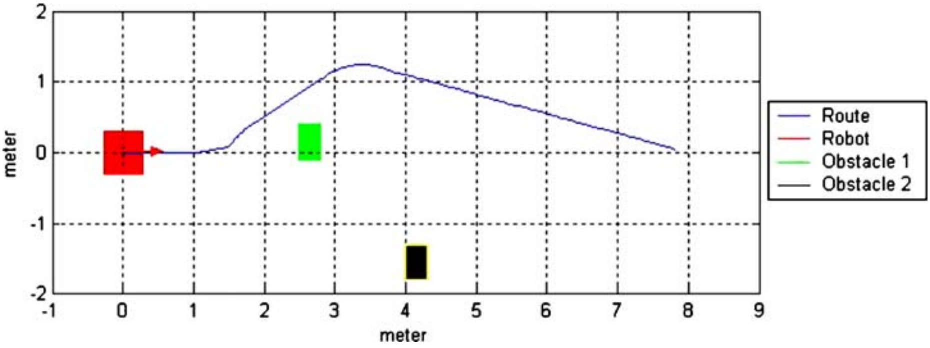


Fig. 14 Collision avoidance using vision system data

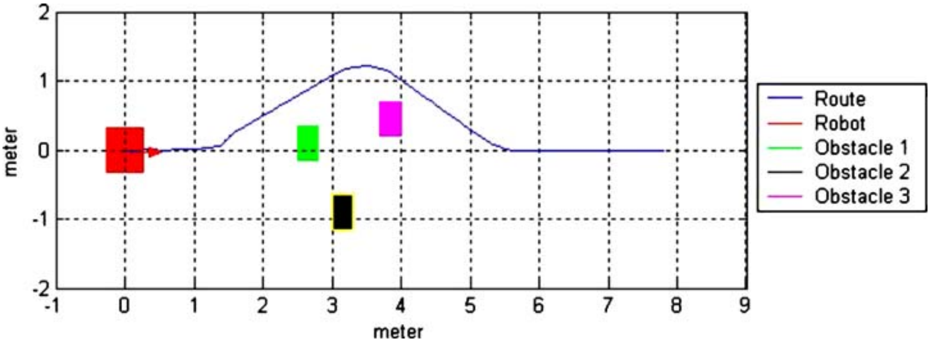


Fig. 15 Collision avoidance using vision system information

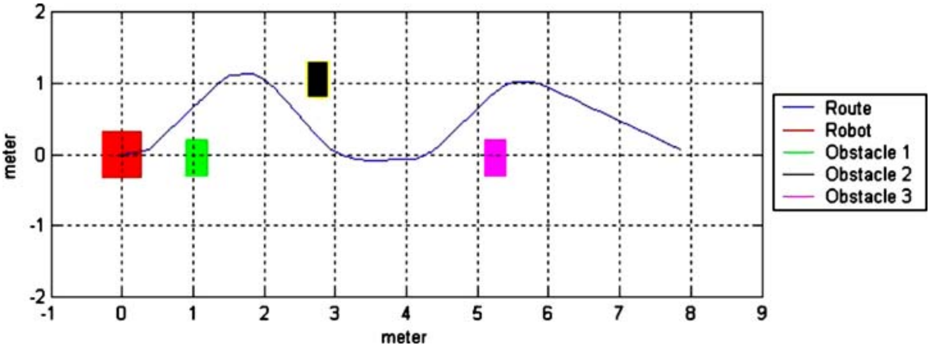


Fig. 16 Collision avoidance using vision system data

meets the second obstacle which lies left of the robot. The robot moves right and meets the third obstacle which is located right of the robot. Figures 17 and 18 demonstrate the robot trajectory while it avoids three obstacles using range measurements from ultrasonic sensors. To avoid an obstacle, the robot turns left or right according to its relative position with the obstacle.

Figures 19, 20, 21 and 22 demonstrate the robot's trajectory and collision avoidance using data from both sonar and camera. For comparison purposes, the robot's trajectory using only the ultrasonic sensors is also shown. Type I obstacles are represented with green color and Type II with cyan.

Figure 23 shows snapshots of the robot's trajectory while it avoids three yellow obstacles.

6 Discussion

As mentioned earlier, the case of monocular vision system has also been tested. Figure 1b shows the main steps of the vision algorithm where the pair of images needed for range

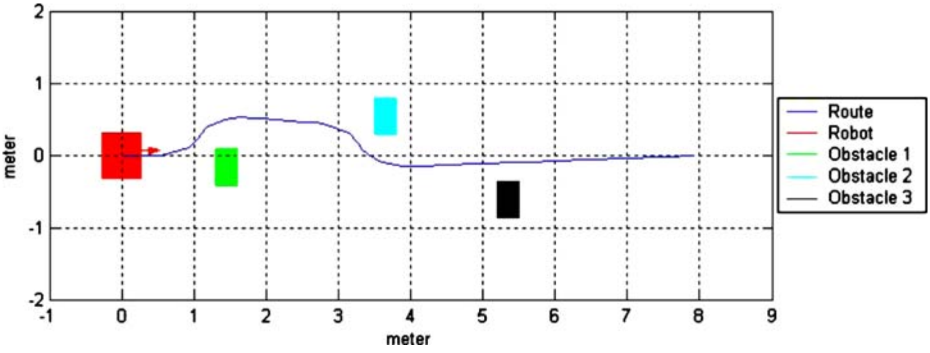


Fig. 17 Collision avoidance using range measurements from the ultrasonic sensors

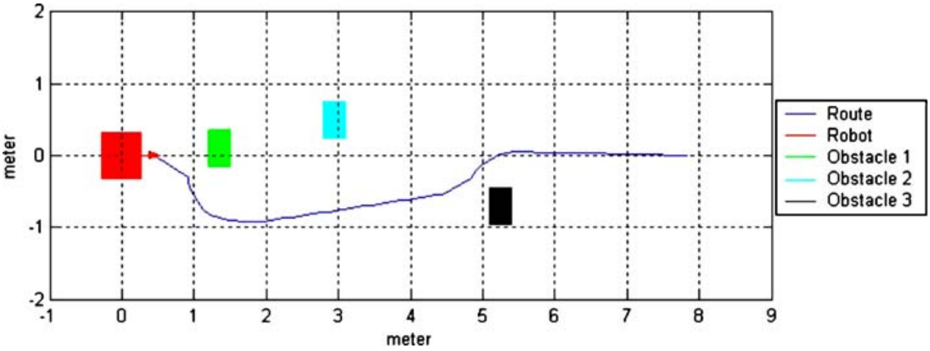


Fig. 18 Collision avoidance using range measurements from the ultrasonic sensors

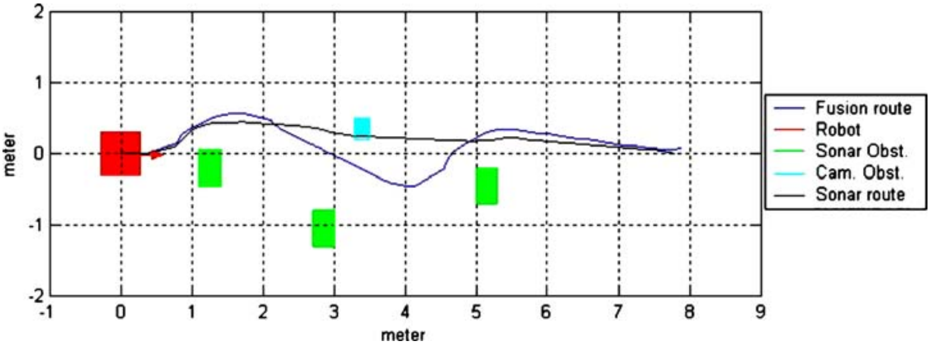


Fig. 19 Collision avoidance using data from sonar and camera

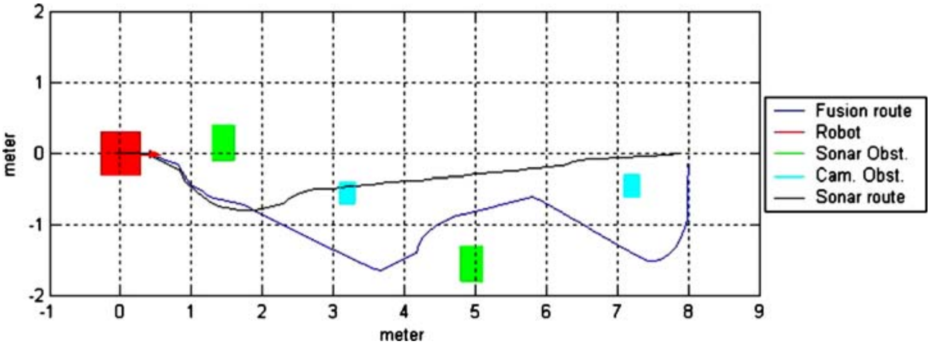


Fig. 20 Collision avoidance using data from sonar and camera

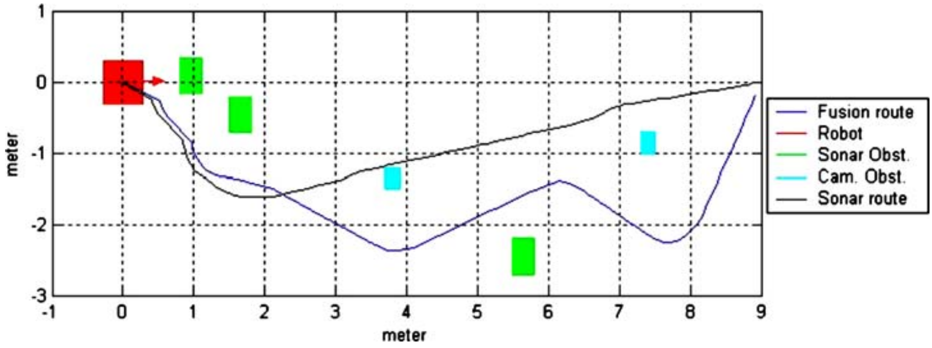


Fig. 21 Collision avoidance using data from sonar and camera

computation is grabbed sequentially. It is assumed that the mobile robot is moving directly towards the object of interest and no rotation is performed during the capture of each image pair. The equation (previously derived)

$$Z = \frac{x_{p2}B}{x_{p1} - x_{p2}} \tag{18}$$

is used for distance estimation.

The basic steps of image acquisition, color space transformation, thresholding and filtering remain the same as presented above. However, using the assumption that the mobile robot is moving directly towards the object of interest, the part of interest point correspondence can be improved.

The Moravec Interest Operator increases considerably the computational time of the system (approximately 0.11 s for each image). Instead, a method of image subtraction can be performed. The pair of the thresholded images is subtracted and the outcome is a new image which reveals the edges of the object captured in the pair of consecutive images (Fig. 24). Performing a simple line scan algorithm, the corresponding pixels can be derived.

The case of the monocular vision system requires accurate knowledge of the distance covered between two consecutive captured images; however, the robot's odometer cannot

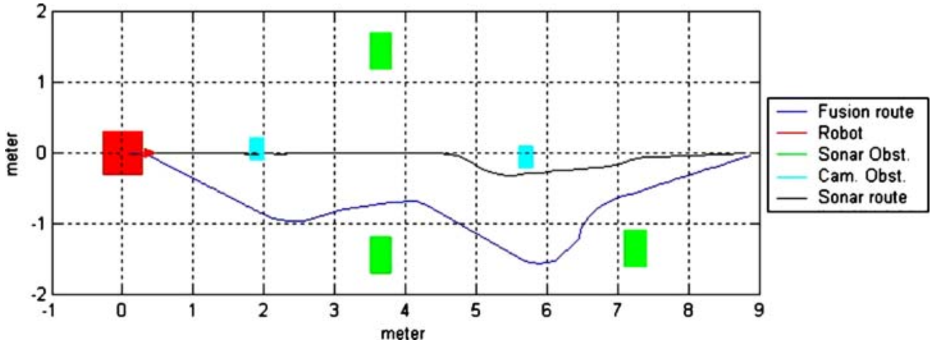


Fig. 22 Collision avoidance using data from sonar and camera

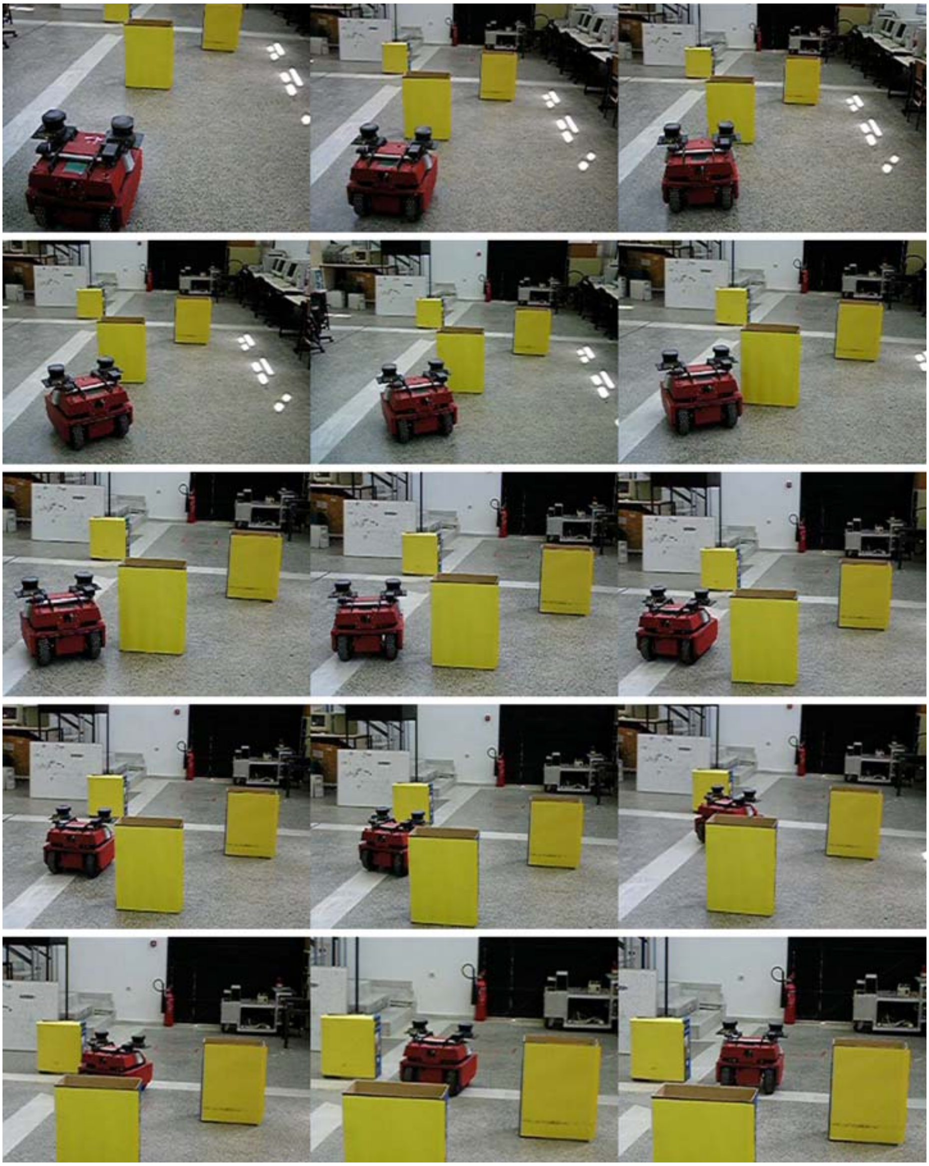


Fig. 23 Robot's trajectory snapshots

provide such accurate data. Thus, the computational error is significantly increased. This is the main reason why stereo vision is preferred.

Indicative experimental results on range measurements using a monocular vision system that conducted without the use of the robot's odometer demonstrate mean error of 1.02% and standard deviation 0.54.

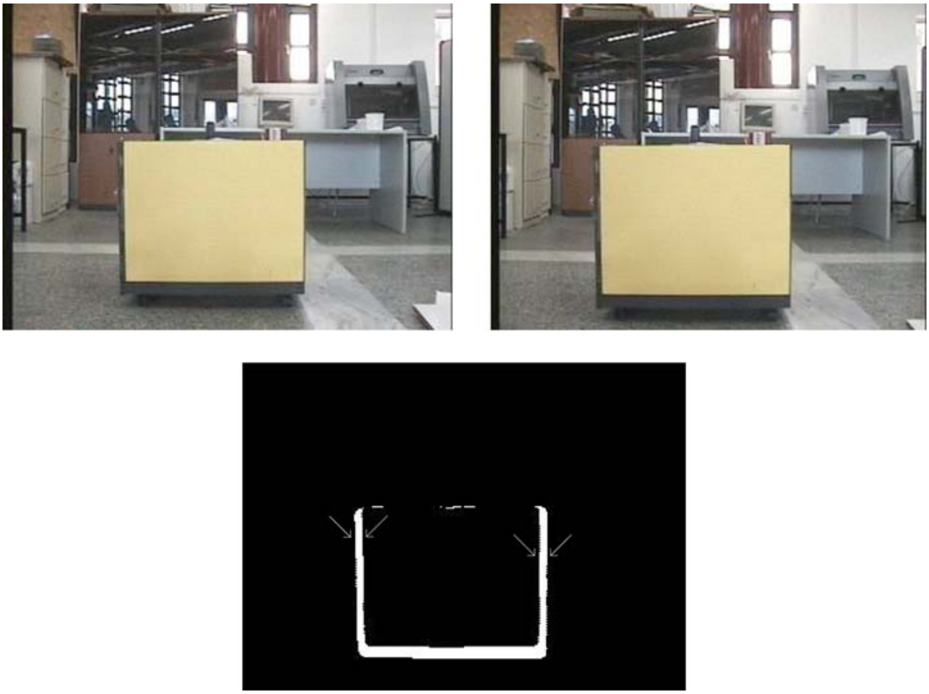


Fig. 24 Interesting point correspondence for the case of monocular vision system

7 Conclusions

This paper has presented a simple vision based depth estimation technique using the image size of an image, the angle of view of the camera and the relative position of two different captures of the same scene. The effectiveness of the proposed method has been tested for mobile robot motion planning and collision avoidance in an indoors environment based on simultaneous ultrasonic sensor and/or camera range measurements.

Compared to other related methods, experimental results have confirmed the simplicity and effectiveness of the presented approach as well as superior maximum computational errors and *Normalized Root Mean Square Errors, rmse* of range measurements using either a rotated, parallel, or monocular vision system.

Future work involves sensor fusion between laser scanner and stereoscopic vision system with the introduction of system failures.

Acknowledgement This research has been partially supported by ONR Grants N00014-03-01-786 and N00014-04-10-487 and a DOT through the USF CUTR Grant 2117-1054-02.

Appendix

The Appendix offers comparative results using the *HIS* and *CMYK* color spaces to identify obstacles and navigate around them.

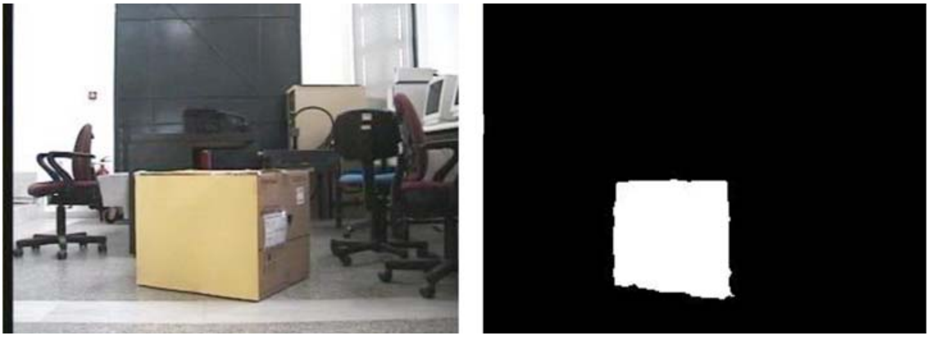


Fig. 25 Applying threshold technique in HSI color space

HSI Color Space

The hue, H, of a color is determined by its angle with respect to the red axis [16]. H=60° corresponds to yellow. Thus, a threshold on the hue component in the area of 55–65° can easily extract yellow pixels from a color image. Additionally, saturation S of a color is the degree to which the color is undiluted by white [16]. Saturation's range for yellow obstacles has been experimentally determined to be within (0.35, 0.43), where 0 corresponds to unsaturated and 1 to fully saturated colors. Applying both hue and saturation thresholds in a *HSI* color space image results in a new image in which yellow pixels are represented as white according to the equation:

$$g(x,y) = \begin{cases} 255, & 55^{\circ} \leq H(x,y) \leq 65^{\circ} \text{ and } 0.35 < S(x,y) < 0.43 \\ 0 & \text{else} \end{cases} \tag{19}$$

Figure 25 shows a color image containing a yellow obstacle and the corresponding output after thresholding.

The *HSI* color space may also be used to extract red, blue and green colors. The hue's value for red color is H=0° or H=360°, while the saturation's range has been determined to be within (0.70, 0.80). Similarly, the hue's value for green color is H=120°, while its saturation varies between 0.35 and 0.54. Finally, blue color has a value of hue H=240°, and

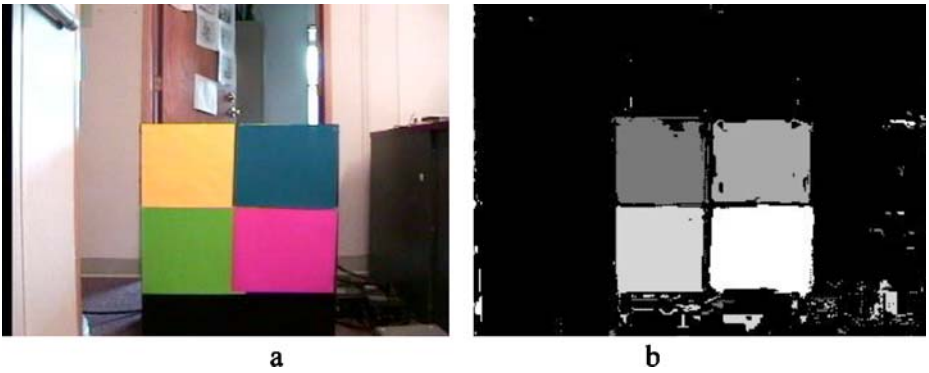


Fig. 26 Thresholding to extract *red*, *green*, *blue* and *yellow* obstacles using the HSI color space

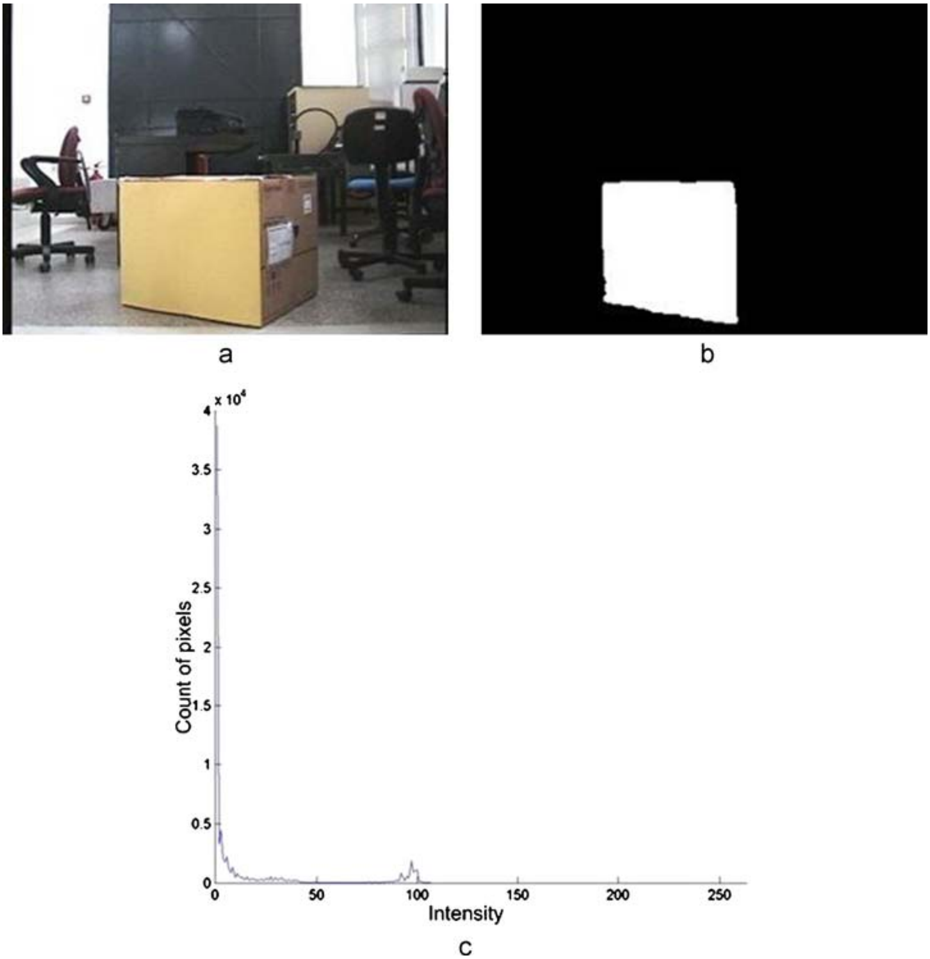


Fig. 27 Applying threshold technique in CMYK color space

saturation between 0.66 and 0.94. Figure 26 presents a color image containing the four color obstacle (a) and the result of the threshold technique (b) according to the equation:

$$g(x,y) = \begin{cases} 255, & 355^\circ \leq H(x,y) \leq 5^\circ \text{ and } 0.70 \leq S(x,y) \leq 0.80 \\ 200, & 115^\circ \leq H(x,y) \leq 125^\circ \text{ and } 0.35 \leq S(x,y) \leq 0.54 \\ 150, & 235^\circ H(x,y) \leq 245^\circ \text{ and } 0.66 \leq S(x,y) \leq 0.94 \\ 100, & 55^\circ H(x,y) \leq 65^\circ \text{ and } 0.35 \leq S(x,y) \leq 0.43 \\ 0, & \text{else} \end{cases} \tag{20}$$

CMYK Color Space

The *CMYK* color space represents each color in its secondary spectral components cyan, magenta, yellow. One of its primary components is *yellow*. Pixels of a yellow obstacle refer

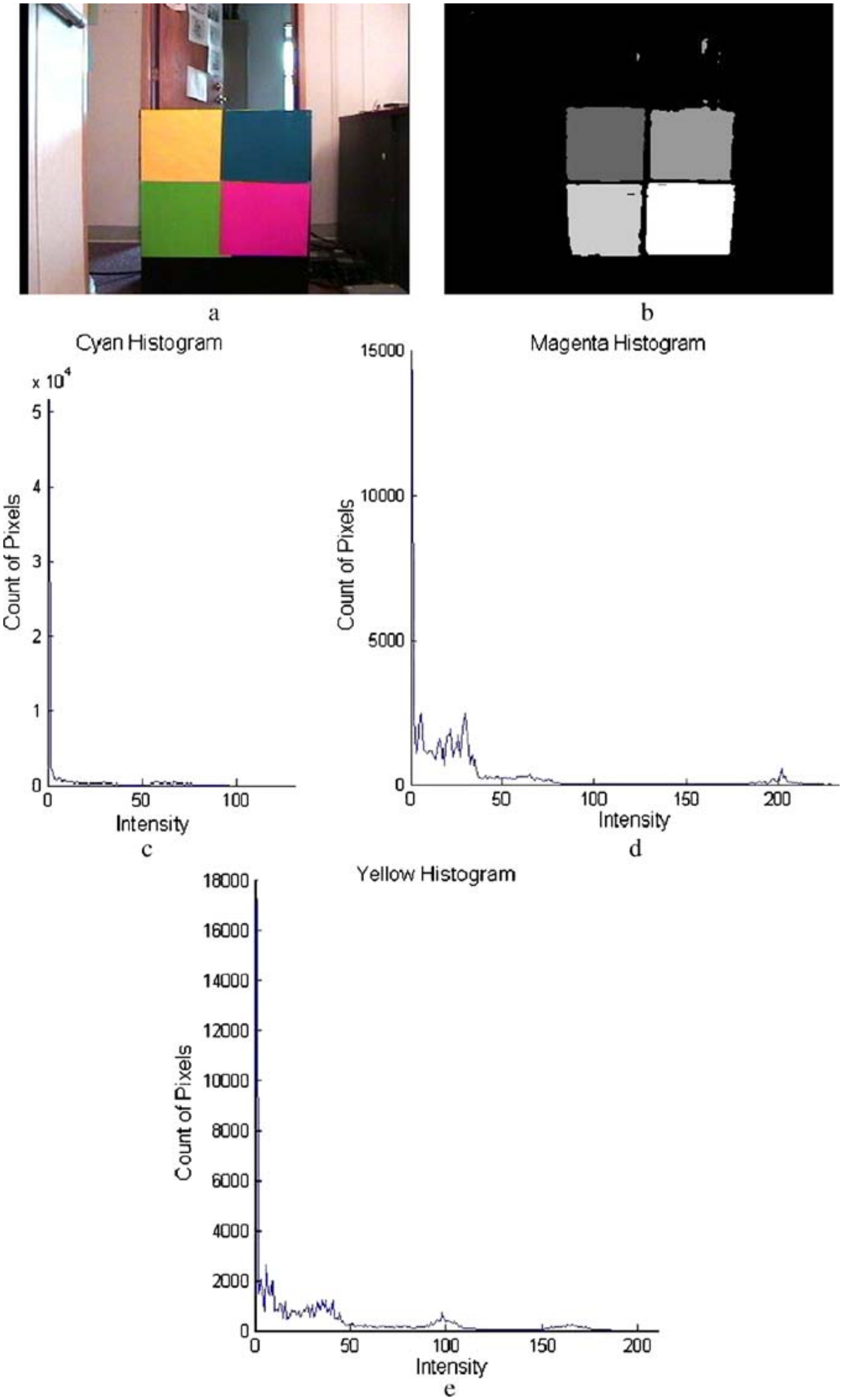


Fig. 28 Thresholding to extract *red*, *green*, *blue* and *yellow* obstacles using the CMYK color space

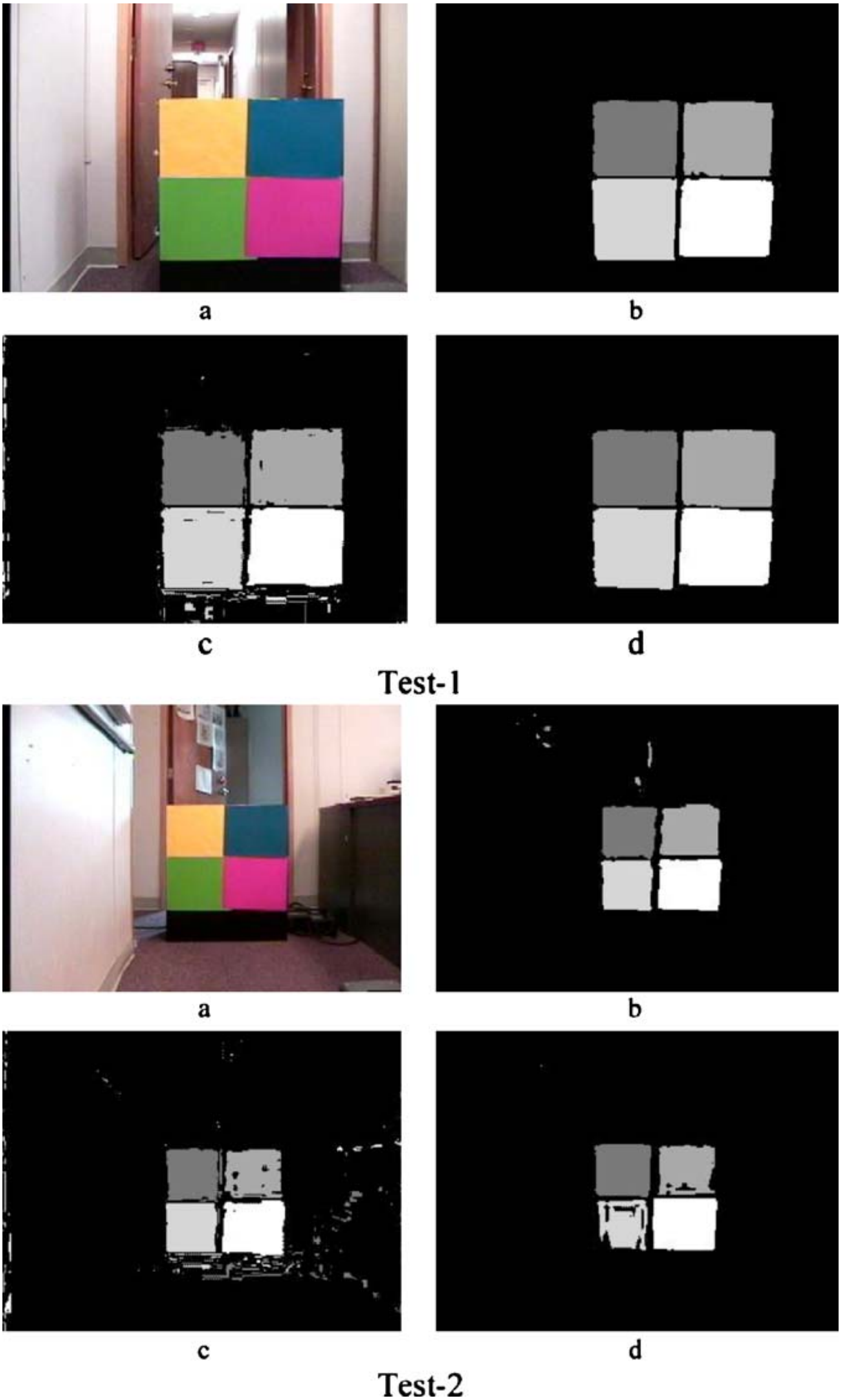


Fig. 29 Thresholding to extract *red*, *green*, *blue* and *yellow* obstacles using the three color spaces

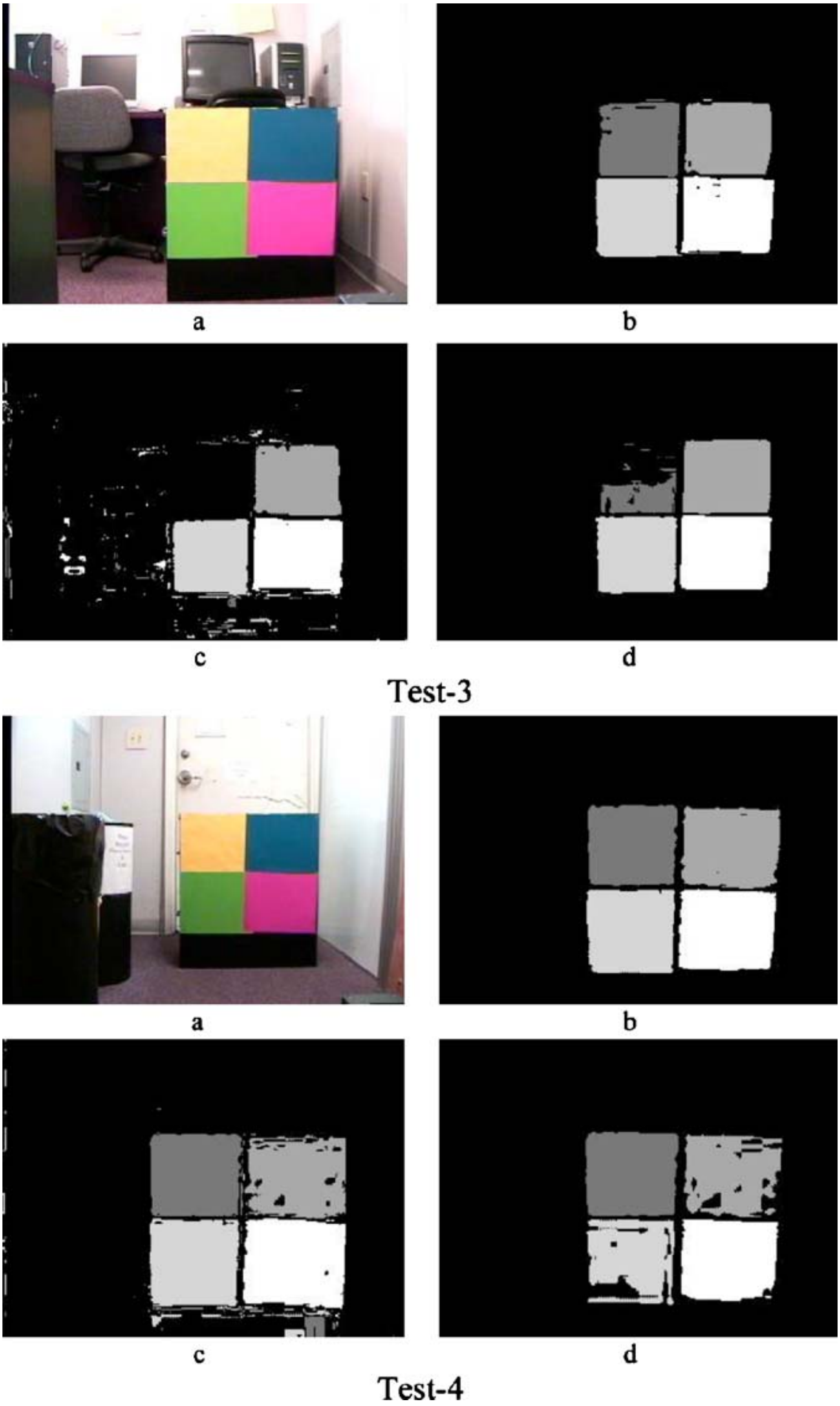


Fig. 29 (continued)

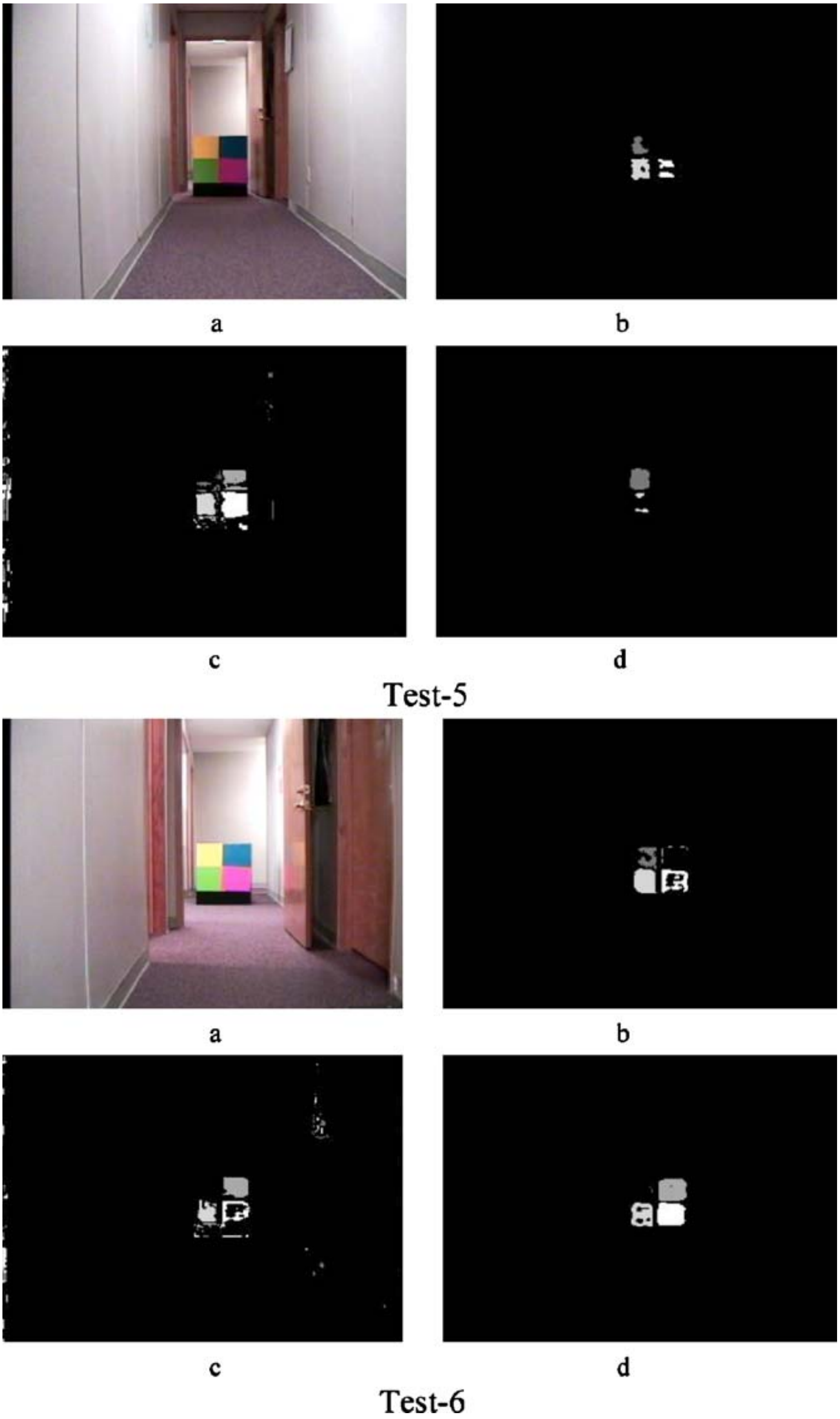


Fig. 29 (continued)

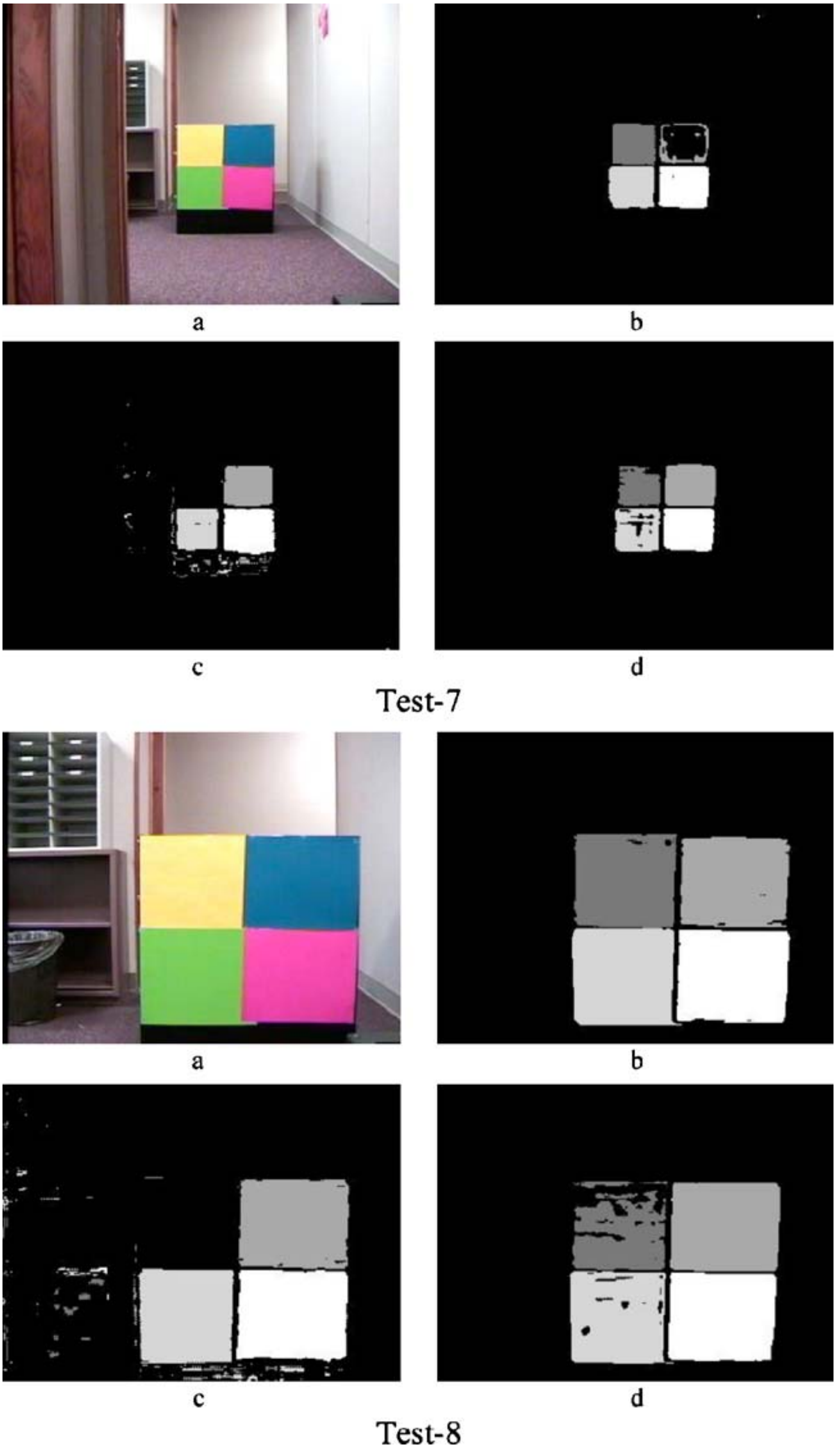
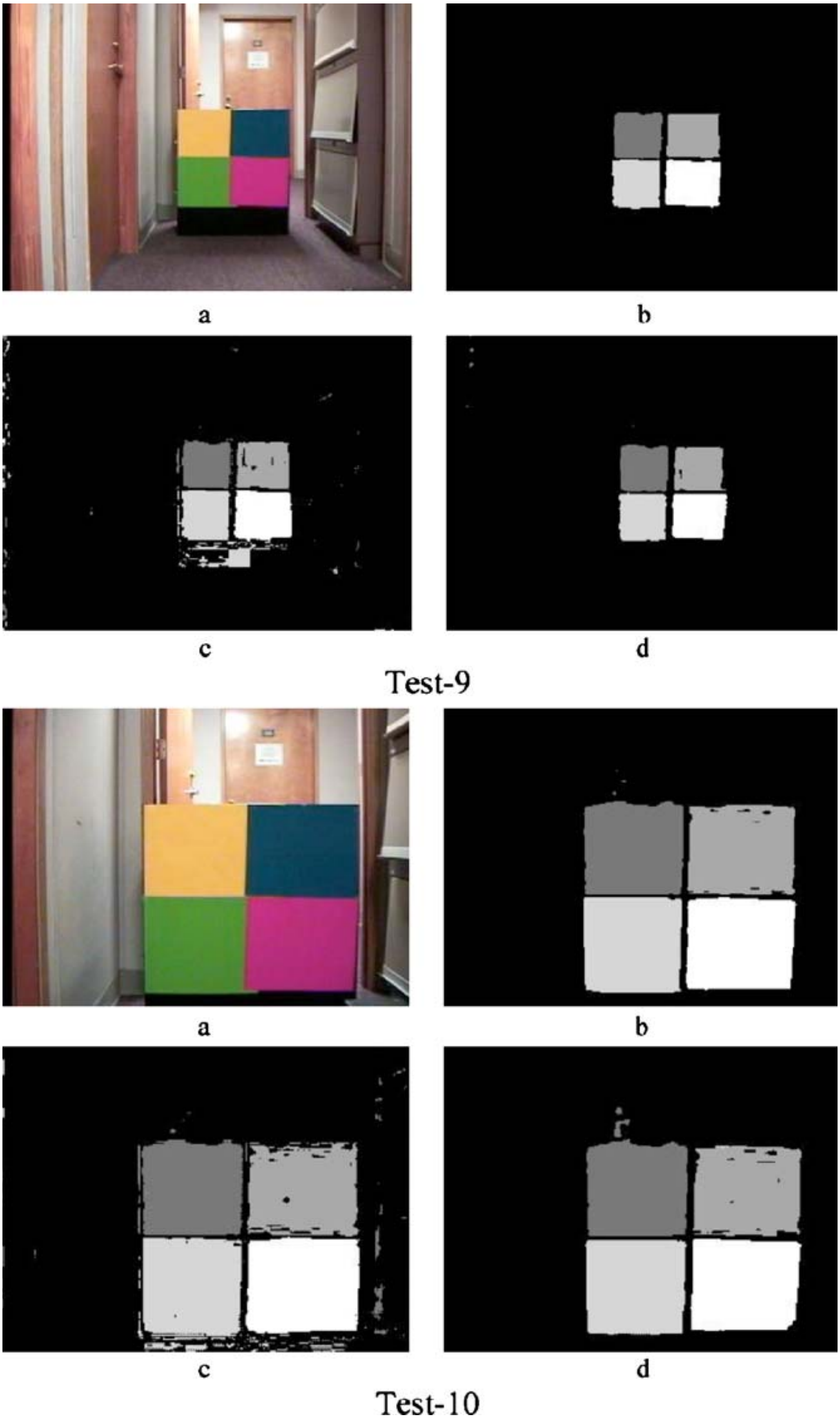


Fig. 29 (continued)



Test-9

Test-10

Fig. 29 (continued)

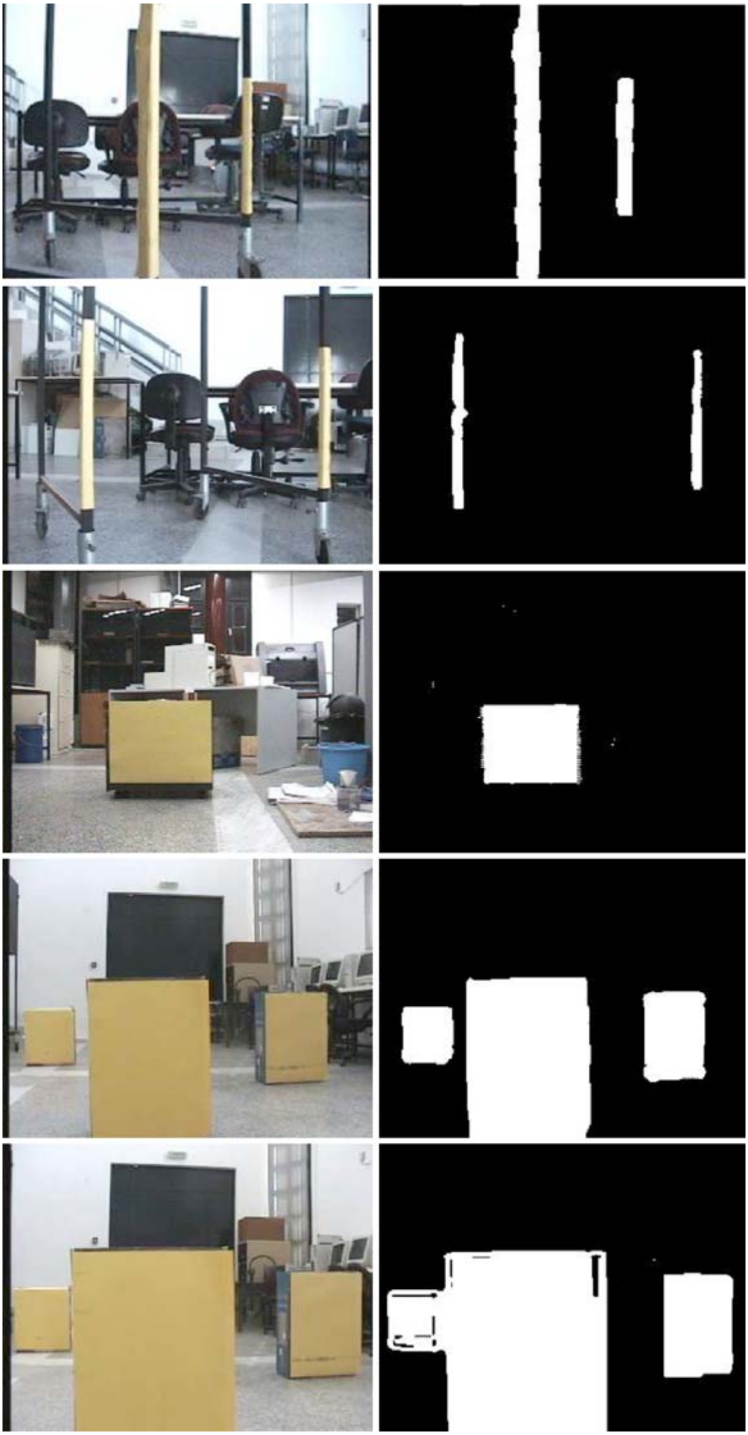


Fig. 30 Applying threshold technique to extract *yellow* obstacles using the YC_bC_r color space

to pixels with highest intensity in the yellow's component image that can be extracted from its histogram. Additionally, yellow color pixels are presented as black on the *cyan* component. A new image may be created by applying a threshold technique to the *CMYK* image according to:

$$g(x,y) = \begin{cases} f(x,y), & Y(x,y) > T_1 \text{ and } C(x,y) < 5 \\ 0 & \text{else} \end{cases} \tag{21}$$

T_1 is the highest pick in yellow's component histogram minus a tolerance of 10%, *Y(x,y)* and *C(x,y)* the intensity functions of *Yellow* and *Cyan* respectively. Figure 27 shows a color image contains a yellow obstacle (a), the thresholded image (b) and the histogram of the yellow component (c).

The *CMYK* color space may also be used to extract red, green and blue colors from a color image. Red color corresponds to the highest values on *magenta* component and its *cyan* values vary between 80 and 115. Green color corresponds to high values on the *yellow* component and its *magenta* values vary between 8 and 135. Finally, blue color corresponds to the highest values on the *cyan* component and at the same time its magenta values vary between 180 and 230. Figure 28, presents a color image containing the four color obstacle (a), the result of the threshold technique (b) according to Eq. 22 and the histograms of cyan (c), magenta (d) and yellow (e) components.

$$g(x,y) = \begin{cases} 255, & M(x,y) > T_3 \text{ and } 80 \leq C(x,y) \leq 115 \\ 200, & Y(x,y) \geq T_1 \text{ and } 80 \leq M(x,y) \leq 135 \\ 150, & C(x,y) \geq T_2 \text{ and } 180 \leq M(x,y) \leq 230 \\ 100, & Y(y,x) \geq T_1 \text{ and} C(x,y) < 5 \\ 0, & \text{else} \end{cases} \tag{22}$$

T_1, T_2 and T_3 are the highest picks with the highest intensity values in yellow's, cyan's and magenta's component histograms, respectively, minus a tolerance of 10%. *C(x,y)*, *M(x,y)* and *Y(x,y)* are the intensity functions of the *Cyan*, *Magenta* and *Yellow* component, respectively.

Color Segmentation Results

Figure 29 illustrates a multicolor obstacle located at different distances from the robot, under different lighting conditions (a), and the result of the threshold technique applied to the YC_bC_r (b), *HSI* (c), and *CMYK* (d) color spaces.

As observed, the threshold technique using the YC_bC_r color space performs better in identifying all colors, while background pixels have been considerably suppressed.

Finally, Fig. 30 shows a more realistic and complicated scene with yellow objects that demonstrate the ability of the threshold technique as applied to the YC_bC_r color space to identify objects of different size and orientation.

References

1. Ohya, A., Kosaka, A., Kak, A.: Vision-based navigation by a mobile robot with obstacle avoidance using single-camera vision and ultrasonic sensing. IEEE Trans. Robot. Autom. **14**(6), 969–978 (December 1998)

2. Duffy, B.R., Garcia, C., Rooney, C.F.B, O'Hare, G.M.P.: Sensor fusion for social robotics. In: 31st International Symposium on Robotics, Montréal, Canada, pp. 258–264, May 2000
3. Mann, R., Jones, J., Beckerman, M., Glover, C., Farkas, L., Han, J., Wacholder, E., Einstein, J.: An intelligent integrated sensor system for the ORNIL mobile robot. In: Proceedings of IEEE on Intelligent Control, pp. 170–173 (1988)
4. Gubber, G., Sahli, H.: Sensor integration on a mobile robot. In: 12th International Symposium on Measurement and Control in Robotics, France, June 2002
5. Rajagopalan, A.N.: Depth estimation and image restoration using defocused stereo pairs. IEEE Trans. Pattern Anal. Mach. Intell. **28**(11), 1521–1525 (November 2004)
6. Mudenagudi, U., Chaudhuri, S.: Depth estimation using defocused stereo image pairs. In: Proceedings of IEEE International Conference on Computer Vision, vol. 1, 483–488, September 1999
7. Umeda, K., Takahashi, T.: Subpixel stereo method: a new methodology of stereo vision. In: Proceedings of IEEE ICRA, pp. 3215–3220, April 2000
8. Jiang, F., Weymouth, T.E.: Depth from dynamic stereo images. In: Proceedings of IEEE CVPR, pp. 250–255, June 1989
9. Lai, S-H., Fu, C-W., Chang, S.: A generalized depth estimation algorithm with a single image. IEEE Trans. Pattern Anal. Mach. Intell **14**(4), 405–411 (April 1992)
10. Malis, E., Rives, P.: Robustness of image-based visual servoing with respect to depth distribution errors. In: Proceedings of IEEE ICRA, pp. 1056–1061, 2003
11. Derrouich, S., Izumida, K., Shiiya, K.: A combination of monocular CCD camera and inertial-sensor for range estimation. In: IEEE Annual Conference on IECON, vol. 3, pp. 2191–2196, November 2002
12. Fox, D., Burgard, W., Thrun, S.: The dynamic window approach to collision avoidance. IEEE Robot. Autom. Mag. **4**(1), 23–33 (March 1997)
13. Sahin, E., Gaudiano, P.: Mobile robot range sensing through visual looming. In: Proceedings of IEEE ISIC, pp. 370–375, September 1998
14. CCIR. Encoding parameters of digital television for studios. CCIR Recommendation 601-2, *International Radio Consultative Committee* (ITU), 1990
15. Dios, J.J., Garcia, N.: Face detection based on a new color space YCgCr. In: International Conference on Image Processing, vol. 3, pp. III 909–12, vol 2, 2003
16. Gonzalez, R.C., Woods, R.E.: Digital Image Processing. Addison-Wesley, Reading, MA (1992)
17. Jain, R., Kasturi, R., Schunck, B.G.: Machine Vision. McGraw-Hill, New York (1995)
18. Doitsidis, L., Valavanis, K.P., Tsourveloudis, N.: Fuzzy logic based autonomous skid steering vehicle navigation. In: Proceedings of IEEE ICRA, pp. 2171–2177, May 2002
19. Jennings, A., Murray, D., Little, J.J.: Cooperative robot localization with vision-based mapping. In: Proceedings of IEEE ICRA, vol 4, pp. 2659–2665, May 1999
20. Recken, W.D.: Autonomous sonar navigation in indoor, unknown and ustructured environments. In: Proceedings of IEEE IROS, vol 1, pp. 431–438, September 1994
21. Elfes, A.: Sonar-based real world mapping and navigation. IEEE J. Robot. Autom. **3**, 249–265 (June 1987)
22. Valavanis, K.P., Hebert, T., Kolluru, R., Tsourveloudis, N.: Mobile robot navigation in 2-D dynamic environments using an electrostatic potentional field. IEEE SMC **30**, 187–196 (March 2000)
23. Tucakov, V., Sahota, M., Murray, D., Mackworth, A., Little, J., Kingdom, S., Jennings, C., Barman, R.: Spinoza: a stereoscopic visually guided mobile robot. In: Proceedings of 30th ICSS, vol. 5, pp. 188–197, January 1997.
24. Russ, J.C.: The Image Processing Handbook. CRC Press, Boca Raton, FL (1995)
25. Sonka, M., Hlavac, V., Boyle, R.: Image Processing, Analysis and Machine Vision. PWS Publishing, New York (1999)