A novel floor segmentation algorithm for mobile robot navigation

Soumabha Bhowmick
Advanced Technology Development Center, IIT Kharagpur
Email:soumabha.bhowmick@gmail.com

Abhishek Pant
Computer Science and Engineering,IIT Kharagpur
Email: abhishek.pant93@gmail.com

Jayanta Mukherjee
Computer Science and Engineering,IIT Kharagpur
Email: jay@cse.iitkgp.ernet.in

Alok Kanti Deb
Electrical Engineering,IIT Kharagpur
Email: alokkanti@ee.iitkgp.ernet.in

***Abstract*—The task of detection of floor area for mobile robot navigation has received immense importance over the years. The main challenging problem as long as the path planning of robots is concerned, is the obstacle avoidance. Obstacle detection and avoidance in real time is a complex and computationally expensive process as a result of which the robotics researchers opted to segment out floors, which is comparatively easier process and at the same time very much feasible in real time applications. The proposed floor segmentation algorithm detects floor from a scene irrespective of the change of illumination as well as shadows. A conventional breadth first search based region growing technique has been used with histogram based features in YCbCr color space and floor junction masking to detect the floor pixels from the scene. The algorithm has been compared with various other existing techniques to showcase the improved results of the proposed one.**

Keywords*—*Histogram, floor junction masking, breadth first search.

I. Introduction

The most important part for a mobile robot navigation is detection of the travarsable region. Typically, most robots rely on range data such as that from ultrasonic sensors [1], laser range finders [2], [3] or stereo vision [4], [5] to detect obstacles. However, sensors, especially the laser range finder, produce good results. The laser range finder is expensive, and hence is not suitable for consumer use or for low-cost robots. Ultrasonic sensors are cheaper, but, they generally suffer from low angular resolution. Stereo vision based approaches are computationally expensive. They produce a sparse point cloud, and require precise calibration. Moreover, range data based approaches are unable to distinguish between different surfaces of the same height (e.g. between pavement and rocky areas in a park), or small/flat objects lying on the ground. Existing floor segmentation approaches suffer from mainly the following problems.

I *Shadow Casting*: This is a very common problem almost in any environment, where an upright obstacle casts a shadow on the floor.
II *Patterned floor*: The floor patterns are not always uniform.
III *Illumination change*: Illumination is not always uniform over the image; hence the intensity values are very much liable to change.

In this paper, a novel method for floor segmentation using a region growing method has been presented. The proposed technique can efficiently segment the patterned as well as uniform floors independent of the illumination and shadow. Section II describes the related significant works done in the field. Section III describes the floor segmentation algorithm in detail, and section IV shows the results of the floor segmentation algorithm. Section V concludes the paper and presents the future scope of the paper.

II. Prior Works

Floor segmentation is an open problem targeted by many roboticists over the time. One of the most classical work was done by Piotr Jasiobedzk [6] in the year 1996, where the author integrated vision as well as laser range finder to segment the travarsable region. Another significcant work that has been reported, was by G. Cheng and A. Zelinsky in the year 1996 [7]. They have presented a template matching technique for segmenting the floor. A grid of pixels were matched with available template to segment out the carpeted floor. S. D. Jones et. al [8] also used a template matching technique for segmenting floors. In [9] Xue-Nan Cui et. al proposed a segmentation algorithm by computing the plane normals. In relatively recent times, a few more works were reported. L. F. Posada et. al [10] first converted an image from RGB colorspace to HSV colorspace, and computed the histogram for all the channels to capture the floor patterns, and finally used a region growing technique from a seedpoint to segment floor. Ma ling et. al. [11] used K-means clustering approach. They assumed the floor color to be uniform, and converted the colorspace from RGB to YCbCr. K means clustering was applied after dividing the image in segments of rectangular blocks, and calculating the features. The centroid of the rectangle was calculated, and it was used for clustering. Li et. al [12] proposed a technique, where they computed a score considering several factors for delcaring a candidate region as floor. In [13] the authors have presented a graphical approach to detect the floor. Authors have assumed that for every frame the upper half is less probable in containing obstacles. Moreover they have also assumed, that the central columns of the bottom rows are free from obstacles. The algorithm proposed here in this work segments

978-1-4673-8564-0/15/$31.00 ©2015 IEEE

floor irrespective of the intensity variation due to illumination difference as well as shadow of the obstacles.

III. Floor Segmentation algorithm

Floor segmentation algorithm mainly addresses two issues, namely, detecting the floor independent of patterns, and illumination. Computational steps of the proposed method are described in the following subsections.

A. Histogram Satisfiability criteria

The image is first converted from RGB to YCbCr colorspace. The histogram of YCbCr has been used to test a satisfiability criteria. The Y-channel contains the luminance information, and the Y histogram shows very high contrast for which the color space has been converted to YCbCr. The histogram satisfiability criteria starts with the following two assumptions

i The floor has a repetitive pattern.
ii The lower rows of the central columns of the camera image is floor, and is termed safe zone. The centroid of the safezone is called the seed point.

The conversion of RGB to YCbCr color space has been done using the following equations [14].

$$Y = 0.299R + 0.587G + 0.114B \tag{1}$$

$$Cr = (R - Y)0.713 + 128 \tag{2}$$

$$Cb = (B - Y)0.564 + 128 \tag{3}$$

The safe zone is selected checking the histogram. The histogram may not have uniform distribution even if the floor is uniform. This is due to varied illumination across the environment. Hence as the algorithm proceeds it encounters intensity values, which differ from the values present in the safe zone. To overcome this problem, the pixels, which have an intensity value within a threshold of the safezone values are added to the histogram. This helps in detecting the floor pixels, with significantly varying intensity values from the safe zone. The histogram is computed to capture the floor pattern, and all the intensity values which are lying within the histogram was listed. These values were used for the purpose of histogram based sampling method discussed in the next section. The whole algorithm has been described in Algorithm 1.

Algorithm 1 Safe Zone Sampling (SZHS)

Input: image, safe zone coordinates
Output: Safe-zone histogram
Image is divided into Y, Cb, Cr planes .
Create three empty histograms (Y, Cb, Cr), where each bin count is initialized to 0.
For each pixel in the safe zone, obtain its Y,Cb,Cr components, and increment the count of corresponding bins in the respective histograms.
return Safe-Zone histogram

Fig. 1 is a frame of the robot with the safe zone marked as green. The red point is the seed point of the safe zone, which was used in region growing.

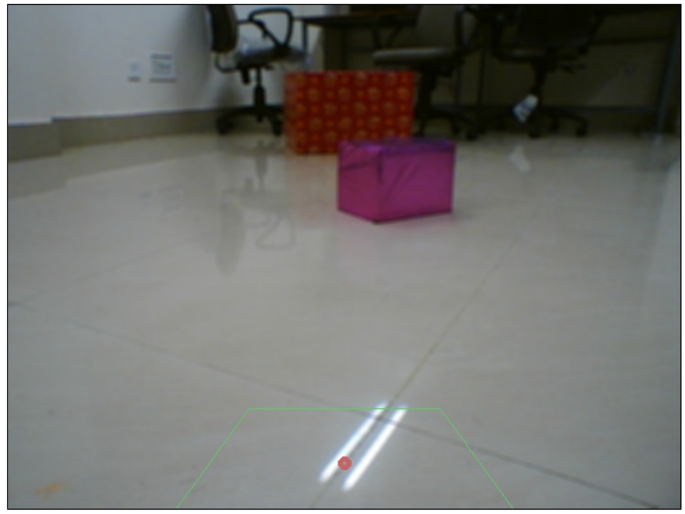

Fig. 1: Safe zone enclosed in green straight lines and the seed point

B. Region Growing

BFS based region growing has been used which starts from a seed point, and all the four neighbors were searched. The diagonal neighbors were not taken under consideration. The algorithm for region growing has been shown in the Algorithm 2.

Algorithm 2 Region Growing (RG)

Input: image, YCbCr histogram, seed point
Output: binary occupancy image(occ)
enqueue($Tovisit \leftarrow seed\ pixel$)
Initialize occupancy image $occ \leftarrow BLACK$ for all pixels.
while $Tovisit$ is not empty **do**
  $seed\_point \leftarrow dequeue(Tovisit)$
  for all unvisited 4-neighbors nbr of $seed\_point$ **do**
    if safe-zone-proximity-condition(nbr) **or** varying-illumination-check($seed\_point, nbr$) **then**
      $Tovisit \leftarrow enqueue(nbr)$
      $occ[nbr] \leftarrow WHITE$
      Update YCbCr histogram.
    end if
  end for
end while
return occ

Algorithm 2 has in it three important functions, which are described below.

I *Safe Zone proximity condition*: Three histograms each for Y, Cb and Cr have been computed. Each pixel is taken and all the intensity values of the three channels were taken. If these values matches any of the intensity values present in the safe zone, or if the values differ by a threshold amount then the pixel will be said to be travarsable, and marked to be in the safe zone.
II *Varying illumination check*: The intensity may vary due to varied illumination. If the current pixel under consideration is greater than the previous pixel by a threshold then we consider the pixel to be floor pixel.

III *Histogram update*: Once a pixel is detected as floor from the varying illumination condition, the YCbCr histogram is updated. The new intensity values are incorporated in the histogram as a result of which detection of floor area becomes more efficient. As the region grows and encounters floor pixels with different intensity values updated histogram improves the performance of safe zone proximity condition as well as varying illumination check.

C. Floor Junction Masking

This is an improvement over the results achieved over the histogram based floor junction detection algorithm. Firstly Canny edge algorithm [15] and Suzuki's contour detection algorithm [16] have been used followed by the detection of the straight lines using probabilistic hough transformation [17]. It is assumed that all the vertical lines are true vertical lines irrespective of the robot position. For each vertical line, it detects a horizontal line within a fixed radius indicating a floor junction. Accordingly it masks the corresponding region as non-traversable. The approach is explained in the Algorithm 3.

Algorithm 3 Floor Junction Masking (FJM)

Input: Camera image
Output: Binary occupancy mask

```
Initialize occupancy mask, occ_m ← WHITE, for all pixels.
Initialize floor junctions list. FJ ← {}
Convert input image to grayscale.
Detect edges E in input image using Canny edge detection [15].
Detect contours C from E using Suzuki85 algorithm [16].
Obtain lines L from C using probabilistic Hough transform [17].
for all vertical lines vl in L do
    for all non-vertical lines fl in L do
        if fl lies within circle of threshold radius from bottom of vl then
            FJ.add((vl, fl))
        end if
    end for
end for
for all fj in FJ do
    p1, p2 ← fj.fl
    ROI ← region bounded by p1(x1, y1), p2(x2, y2), p3(x1, 0) and p4(x2, 0)
    m[ROI] ← BLACK
end for
return m
```

IV. Experimental results

The results of the previously reported segmentation algorithms have been shown in Fig. 2. Fig. 2a shows the image of the environment and Fig. 2c is the simple template matching technique suggested by [7], and Fig. 2d is using histogram based region growing [18]. Fig. 2b is the output of Ulrich method [19].

The images in Fig. 3 show the results of the proposed floor segmentation algorithm. It is observed that the algorithm

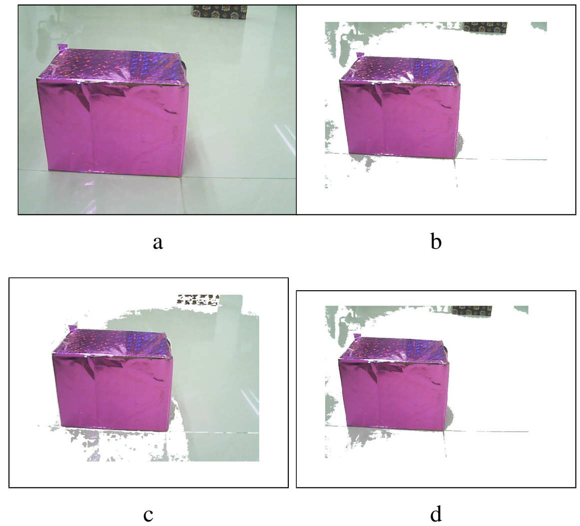

Fig. 2: Results of some existing techniques, (a) Input image,(b) Ulrich method [19],(c) Templete matching based technique [7],(d) Simple RGB Histogram region growing

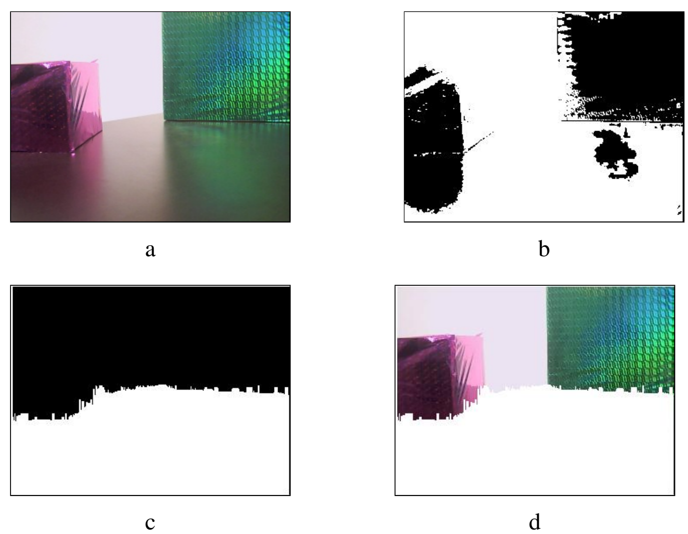

Fig. 3: Performance of the proposed algorithm (a) Input image,(b) before floor junction masking,(c) after floor junction masking,(d) segmented floor

can effectively segment the floor irrespective of patterns and illumination changes over the environment. The Fig. 3a shows the input image, Fig. 3b the output of the region growing technique, and Fig. 3c and 3d output after floor junction masking.

Fig. 4 shows the output of the proposed algorithm. It is found that the proposed approach outperforms the existing approaches. The images were taken from live webcam feed of the robot. The Table 1 shows the performance measure of the proposed technique, where the ground truth was generated manually by thresholding the images, and the numbers of floor pixels non floor pixels were calculated. Three different kinds

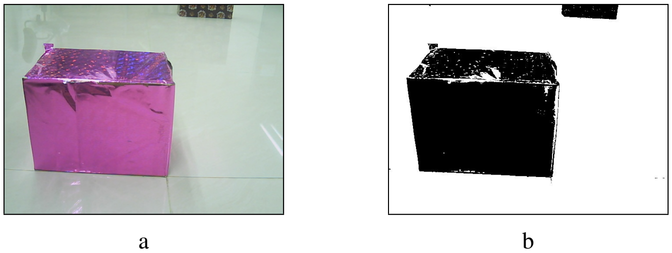

Fig. 4: Runtime performance (a) input frame (b) floor segmented binary image

TABLE I: Comparison with existing techniques

Name of the method	Accuracy %age	Recall %age
Template matching technique [7]	54.71	34.95
RGB histogram based region growing [18]	87.04	81.411
Ulrich method [19]	90.8	86.81
Our approach	94.65	92.42

of floors were taken into account for testing the algorithm. The average accuracy and recall over those three different scenarios were computed for the proposed as well as reported algorithms. The proposed method outperforms most of the existing approaches with average accuracy being 94.65% and recall as 92.42%.

V. Conclusion

In this work a robust method for detection of floor junction has been suggested which takes care of the illumination change quite efficiently and it segments the floor in spite of the variation of the illumination. The second advantage of the algorithm is that it masks out the wall and floor junctions, which results in improved performance. The floor is segmented from the wall even if the floor colors and the wall colors are similar. However, the process is slow, and takes into account many conditions and constraints.

The algorithm fails in situations, where an object casts a shadow on the floor specially in case of shiny floor. For improved performance, it is also necessary to identify the reflection of the light source, and to take necessary corrective actions. In some of the images the reflection of the ceiling light sources has been falsely treated as obstacles because of the high difference in intensity values of those regions from the neighbouring regions. In future we propose to work on mitigating these effects.

Acknowledgment

The authors would like to thank the Sponsored Research and Industrial Consultancy (SRIC, IIT Kharagpur) for setting up the Centre for Excellence in Robotics, and sponsoring the project "Robot as a Social Assistant", under the ambit of which this work has been done.

References

[1] J. Borenstein and Y. Koren, "Obstacle avoidance with ultrasonic sensors," *IEEE Journal of Robotics and Automation*, vol. 4, no. 2, pp. 213–218, 1988.

[2] J. Hancock, M. Hebert, and C. Thorpe, "Laser intensity-based obstacle detection," in *IEEE/RSJ International Conference on Intelligent Robots and Systems,*, vol. 3, 1998, pp. 1541–1546.

[3] A. Ewald and V. Willhoeft, "Laser scanners for obstacle detection in automotive applications," in *Proceedings of the IEEE Intelligent Vehicles Symposium, 2000.*, 2000, pp. 682–687.

[4] M. Bertozzi and A. Broggi, "Gold: A parallel real-time stereo vision system for generic obstacle and lane detection," *IEEE Transactions on Image Processing*, vol. 7, no. 1, pp. 62–81, 1998.

[5] A. Broggi, C. Caraffi, R. I. Fedriga, and P. Grisleri, "Obstacle detection with stereo vision for off-road vehicle navigation," in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition-Workshops, 2005.*, 2005, pp. 65–65.

[6] P. Jasiobedzki, "Detecting driveable floor regions," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 1995, pp. 264 – 270.

[7] G. Cheng and A. Zelinsky, "Real-time visual behaviours for navigating a mobile robot," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 1996, pp. 973 – 980.

[8] S. D. Jhones, C. Andersen, and J. L. Crowley, "Appearance based processes for visual navigation," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 1997, pp. 551 – 557.

[9] Y. G. Cui, X.-N.and Kim and H. Ki, "Detecting driveable floor regions," *International Journal of Control, Automation, and Systems*, pp. 788–798, 2009.

[10] L. Posada, K. Narayanan, F. Hoffmann, and T. Bertram, "Floor segmentation of omnidirectional images for mobile robot visual navigation," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2010, pp. 804 – 809.

[11] M. Ling, W. Jianming, Z. Bo, and W. Shengbei, "Automatic floor segmentation for indoor robot navigation," in *IEEE International International Conference on Signal Processing Systems*, 2010, pp. 684 – 689.

[12] Y. Li and S. T. Birchfield, "Image-based segmentation of indoor corridor floors for a mobile robot," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2010, pp. 837 – 843.

[13] F. Rodriguez-Telles, L. Abril Torres-Mendez, and E. Martinez-Garcia, "A fast floor segmentation algorithm for visual-based robot navigation," in *International Conference on Computer and Robot Vision (CRV)*, 2013, pp. 167 – 173.

[14] G. Bradski, "The OpenCV Library," *Dr. Dobb's Journal of Software Tools*, 2000.

[15] J. Canny, "A computational approach to edge detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence,*, vol. 8, pp. 679 – 698, 1986.

[16] S. Suzuki *et al.*, "Topological structural analysis of digitized binary images by border following," *Computer Vision, Graphics, and Image Processing*, vol. 30, no. 1, pp. 32–46.

[17] N. Kiryati, y. Eldar, and M. Bruckstein, A., "A probabilistic hough transform," *IEEE Transactions on Pattern Analysis and Machine Intelligence,*, vol. 8, pp. 679 – 698, 1986.

[18] D. Holz and S. Behnke, "Approximate triangulation and region growing for efficient segmentation and smoothing of range images," *Robotics and Autonomous Systems*, pp. 1282–1293, 2014.

[19] I. Ulrich and I. Nourbakhsh, "Appearance-based obstacle detection with monocular color vision," in *AAAI/IAAI*, 2000, pp. 866–871.