Obstacle Detection on Heterogeneous Surfaces Using Color and Geometric Cues

Luis J. Manso, Pablo Bustos, Pilar Bachiller and José Moreno

***Abstract*—Autonomous navigation is one of the most essential capabilities of autonomous robots. In order to navigate autonomously, robots need to detect obstacles. While many approaches achieve good results tackling this problem with lidar sensor devices, vision based approaches are cheaper and richer solutions. This paper presents an algorithm for obstacle detection using a stereo camera pair that overcomes some of the limitations of the existing algorithms and performs better on heterogeneous circumstances. We use both geometric and color based cues in order to improve its robustness. The contributions of the paper are improvements to the state of the art on single and multiple cue obstacle detection algorithms and a new heuristic method for merging its outputs.**

***Index Terms*—autonomous robots, visual navigation, obstacle detection.**

I. Introduction

OBSTACLE detection is one of the most fundamental needs for an autonomous navigation system to work. In order to avoid obstacles, the majority of approaches use laser or sonic range sensor devices. While sonic sensors are imprecise, short-sighted and usually unreliable, lidar devices are expensive. The use of stereo vision systems for obstacle detection in autonomous robots might be not only cheaper but free, as they are usually used for object detection and other tasks. This way, the cost of a laser device can be saved up by using a stereo camera pair for one more task. While vision systems can detect objects in their whole visual field, common laser sensors only sweep a plane, so those objects not intersecting the plane are missed. Two axis sweeping lidars or integrated image and lidar sensors are even more expensive. Besides, lidar sensors are not error free, they may get wrong measures on shiny, or black objects not reflecting light as expected[8].

Experimental results show that neither appearance or geometric vision approaches are enough by themselves. Previous geometric ground-obstacle classifiers produce a high rate of false positives if they are not provided with a precise stereo calibration (e.g. in the borders of the paper sheet of figure 1). Our stereo algorithm, as most geometric approaches, relies on the homography induced by a *locally* planar ground. This homography can be stored for a set of known or static camera configurations or calculated in real time using the information from a stereo head. However, our method is able to work within a considerable uncertainty range, performing well even with inaccurate calibrations. By relying on both germetric and appearance information in an appropriate way, our algorithm achieves higher reliability than those just working with one type of information. In addition, since our method does not use ranging sensor devices, it can detect floor downward discontinuities in addition to obstacles lying on the floor.

There has been previous research on visual detection of obstacles based on: color [5] and geometric[2], [4], [12] information. In most of the color based approaches a hue histogram is created and used to classify pixels as obstacle or as free space. This is useful on very restricted environments, but leads to frequent false positives when approaching planar objects lying on the floor so that the robot can actually walk over them (see figure 1), and to false negatives as well (see figure 2). Color based approaches are not useful on grounds of heterogeneous colors. Geometric, homography based, approaches do not perform well with non-textured obstacles (see figure 3).

The most similar works we have found are [6], [15], where both color and geometric cues are used. However, they do not present any improvement over single cue approaches, they justs suggest to use both methods by using *OR/AND* operators. If using an OR operator, it will fail where any of its cues get a false positive. If using an AND operator, it will fail where any of its cues get a false negative. Thus, despite its simplicity, this kind of cue integration is doomed to fail because it does not take into account the nature of its cues. In addition, [6] follows the same color-based classification as [5], which accuracy is discussed on section III.

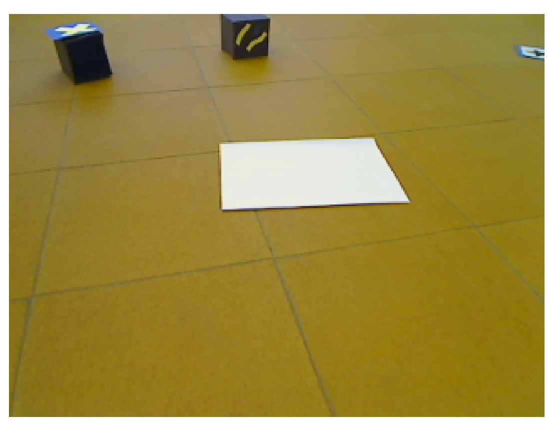


Fig. 1: Color-only approaches can lead to false positives.

Our algorithm uses two obstacle detectors, based on two

Luis J. Manso is with University of Extremadura.
E-mail: lumafe04@alumnos.unex.es
Pablo Bustos is with University of Extremadura.
E-mail: pbustos@unex.es
Pilar Bachiller is with University of Extremadura.
E-mail: pilarb@unex.es
Jose Moreno is with University of Extremadura.
E-mail: josemore@unex.es

Fig. 2: Color-only approaches can lead to false negatives.

Fig. 3: Geometric-only approaches can lead to false negatives.

different visual cues: color-based and stereo-based. The color obstacle detection algorithm produces a binary image, in which pixels are marked as white if its color is not present enough in the histogram, in a similar way as seen in [5]. The stereo cues are gathered in two steps: the first one is similar to [6], and the second one filters the output. We also suggest a better alternative to merge the binary output images. In addition, improvements to each of the two single cue obstacle detectors are presented.

The rest of this paper is organized as follows: Sections II and III describe the process of classification using a single cue approach, geometric and color respectively. Section IV describes how the fusion of cues is made. Section V presents the robot platform used, the experiments performed and their results. Conclusions are detailed on section VI.

II. Geometric-based detection

The geometric-based obstacle detection algorithm is based on the idea that the floor is approximately planar. Given this assumption, the floor induces a planar homography between the two camera images. A planar homography is a projective geometry transformation that maps points between two image planes assuming that they lie on a particular plane[1]. Pixels are mapped by premultiplying its homogeneous coordinates by the *homography matrix*:

$$x' = Hx,$$

so that for any pixel position in an image, a new position is determined for the other view point. In other words, the homography allows to estimate how a plane would be viewed from other perspective. When the intrinsic and extrinsic parameters are known or estimated, the homography matrix can be calculated by the following equation[1]:

$$H = K'(R - tn^T/d)K^{-1},$$

where:

- K and K' are the intrinsic parameters matrix of the initial and the new viewpoints, respectively.
- R and t are the rotation matrix and translation vector that lead from the initial viewpoint to the new one, respectively.
- n is a normalized vector perpendicular to the plane inducing the homography.
- d is the minimum distance from the initial viewpoint to the plane.

If the cameras of the robot are whether static or its only degree of freedom is a common pan movement, that movement will not change the homography, so explicit knowledge of the intrinsic or extrinsic parameters are not needed. In that case, approaches to directly estimate the homography matrix for a static configuration can be used[9], [10], [7], [2].

The ground-obstacle classification is done by warping one of the views to the other and comparing the result of the warping with the actual image seen in the last point of view. Under ideal conditions, provided that the camera can be modeled by the pin-hole model, every warped pixel corresponding to the floor would have the same value in both images. However, in real conditions, we have to face the following problems:

- Light reflections.
- Camera - camera desynchronization.
- Camera - head position sensor desynchronization.
- Different camera responses to the same color.
- Floor is not actually a plane.
- Stereo head pose uncertainty.
- Imprecise homography estimation.

In [6], pixels are classified as obstacles if the distance between its values in the warped image and the actual one is above a threshold. The quality of his method relies heavily on the accuracy of the homography, on a floor free of light reflections, and on good lighting conditions. If the homography accuracy is not good enough, false positives often appear on edges. Our method divides the classification in two stages in order to improve the reliability by including a second test. The first step of our method is similar to the previously seen, the only difference is that we divide the image on small windows to reduce the computational cost. Obstacle candidate windows are then verified or discarded by the second test.

The second stage compares the maximum output of the normalized cross-correlation of candidate obstacle windows with their neighbours for each one of the color channels (red, green and blue). If the maximum for any of the channels is under a threshold, the window does not pass the test and it is classified as obstacle. Color information allows us to differentiate between different colors that have the same luminance. Here, a relatively low threshold should be used. The **key idea** is that it is safer and more stable to decide based on multiple low thresholds of different nature than complement each other, than using a single highly tuned one.

Since only candidate windows are tested, this second stage improves significantly the classification process without adding too much computational overload. This improvement not only decreases the occurrences of false positives on edges of textured floors, which is the bigger disadvantage of similar techniques, it also allows to dismiss low objects that do not actually represent an obstacle for the robot. In any case, no matter how approximate they are, there will be two different homographies, the actual homography that the floor induces in the two cameras, and the homography we are estimating. The second step allows these two homographies to be reasonably different without compromising the system.

Light reflections depend on the point of view of the camera so, from different points of view, the reflections appear to be at different positions on the floor. These reflections are very common in human envrionments. Even though they cannot be detected and identified as proper reflections, and despite they can ruin the geometric classification, the second test reduces the impact.

The main improvement of this algorithm to previous approaches is that it is partially immune to inaccurate homographies. Little variations on the actual homography produce mostly a translation of the image and little projective deformation. For every obstacle candidate window, the best correlation match is searched in its neighbourhood. If a window is not an actual obstacle it should have a good correlation match and the candidate should be discarded. As outlined before, this problem is very common when using mobile robotic heads. The scenarios in which this might be helpful are:

- **Loose or wrong camera position estimation:** Small translations or rotations of the robotic head, due to looseness or small impacts, may change the actual homography. In such cases, other approaches would give lots of false positives.
- **Motorized stereo configurations:** When the homography is recalculated in real-time in a stereo system, wrong pose and angle estimations, due to the imperfections of hardware or software, may lead to inaccurate homographies. This would also lead to false positives if using other approaches.
- **Camera desynchronization:** Even slight camera desynchronizations often make previous algorithms useless. This might or might not be common depending on the hardware used.

Once the second test is finished a binary image where pixels represent small windows of the original images is obtained. Assuming that the destination point of view is the one of the right camera, the process can be summarized as follows:

1) **Copy** the right image, I^R, to a temporal image I^{T1}.
2) Use H to **warp** the left image, I^L, to I^{T1}.
3) Compute the **absolute value** of I^{T1} - I^R. Store the result in I^{T2}.
4) For each window having a pixel over a threshold:
   a) **If** the maximum value for the normalized cross-correlation value in a bigger window is over a threshold for all channels, set the window as floor.
   b) **Else** set the window as obstacle.

The first step can be skipped if, after the algorithm is finished, pixels outside binocular space are ignored. Optionally, if a window size resolution is not enough, a flood-fill operation can be started on the windows that passed the second test after resizing the output image.

III. Color-based detection

The color-based obstacle detection is inspired on the approach seen in [5]. The training consists of the selection of several image regions of the ground and the computation of a three dimensional histogram from those image regions. Region selection can be done manually by a human operator or, if the environment obstacles are textured (there are not untextured walls), by the robot itself selecting floor regions using the geometric-based classification previously detailed.

Instead of using two separate 1-dimensional histograms as suggested in [5], we use a single 3-dimensional one. The use of a three dimensional histogram is justified in terms of discrimination power. Theshold cuts on 1-dimensional histograms sum up to axis paralell decision boundaries on corresponding multidimensional spaces. We build our 3-dimensional histogram with values for normalized red and green components and an approximated value of luminance defined to achieve a certain invariance. Depending on the number of bins in which an axis is divided, we will get a different invariance to the values stored on it, the more bins an axis has the less invariance it gets. In particular, luminance invariance is desirable to some extent, but a complete invariance would entail a loss of discriminative power (see figures 9a and 9b).

Assuming that pixels are represented as RGB bytes (their values range from 0 to 255), the 3D histogram has 128 bins on the X and Y axis and 16 on the Z axis. Thus, the X, Y and Z axis correspond to:

$$X : 128R3/(R+G+B),$$

$$Y : 128G3/(R+G+B),$$

$$Z : (R+G+B)/6$$

Once the histogram is generated, it is low-filtered using a 5x5x5 mask and the training is finished.

In [15] a similar approach is taken, but it uses a two dimensional histogram, with normalized red and green values only. Thus, it is fully luminance invariant, which, as previously seen, is rarely a desired feature.

The classification is also divided in two stages. In the first stage, each pixel is classified according to the presence of its color in the histogram. If the quantity is lower than a threshold, the pixel is classified as obstacle. This threshold can be set as a percentage of the histogram population so it does not depend on the size of the training set. Despite most color based obstacle detectors use HSV color space, we decided not to use it because hue values are usually very noisy at low saturation or low luminance. While other approaches as [5], [6] do not take into account pixels with low saturation, experimental results proved that greyish pixels should not be ignored. Our three dimensional approach histogram shows better performance in the experiments.

The second classification stage windowizes the output of the first step by using a low-valued neighbour interpolation algorithm with 4 pixel width windows (i.e. a 4x4 window is classified as obstacle only if every pixel of the window is). This step removes false positive produced by noise.

IV. Multiple cue obstacle detection

An obstacle detection system based on only one of the described methods would perform well under certain circumstances: planar floor and textured obstacles in the geometric approach; and disjoint color sets for obstacles and floor in the color-based one. Since this conditions are seldom found, the quality of the classification can be improved by using both at the same time, taking advantage of the different properties of the cues. A color-based only classification would often lead to non desirable classifications such as classifying as obstacle a paper sheet lying on the floor. On the other hand, a geometric-based only classification would not see any obstacle in an untextured wall. We propose that the use of both cues produces higher quality classifications. Nevertheless, we also claim that merging the results of both classifiers should be carried in a more sophisticated way than *AND/OR* operators.

A. Geometry wins on small obstacles

Here, the main assumption is that a small region classified as obstacle by its color that is not classified by the geometry obstacle detector is probably a planar object which does not represent an obstacle to the robot. Thus, it should not be classified as obstacle. According to this, isolated regions classified as obstacle by its color and not by the geometry obstacle detector are ignored.

B. Color wins on large obstacles

On indoors environments, untextured walls are very common, so the planar perspective mapping approach might not classify walls correctly. Despite a isolated small region classified as obstacle only because of its color is ignored, big regions reaching the top of the images are suspicious enough to assume its an untextured wall. Thus, this kind of regions are classified as obstacles despite they are not detected by the geometry algorithm.

C. Multiple cue detection result

Both, planar perspective mapping and color based cues are merged in order to get a single binary image as output, where each pixel is related to one of the windows of the single cue classifiers. As we will see in section V, the fusion of cues of different nature allows us to navigate through unstructured as well as structured ones. The only restriction is that the ground must be *locally* planar.

V. Experimental results

The described system leads to a binary image which can be easily used for navigation. The floor boundary can be calculated by scanning the columns of the binary image bottom-up, constructing a polyline with the first occurrences of obstacle pixels in the image. If the extrinsic camera parameters are known, assuming that obstacle pixels are near the floor, the polyline can be mapped into world coordinates. With the polyline in world coordinates, a laser measure can be estimated and so, local navigation can be solved by using existing algorithms such as VFH*[11].

Figures 5, 6 and 7 are examples of the output of the obstacle detection system. The left obstacle of figure 5 and both obstacles of figure 7 are detected by both geometric-based and appearance-based algorithms. The wall at the right of figure 5, only has texture at its bottom, which actually lies on the floor plane, thus, it can only be detected by its appearance. The system is also able to discard the planar objects lying on the floor which appear on figures 6 and 7. In all of these figures floor plane is represented by light colored lines and obstacles are represented by dark ones.

The rest of the section will introduce the robot platform used and the results of the experiments which lead us to the conclusions that follow in section VI.

A. Description of the used robot platform

The obstacle detection system has been tested on RobEx, a low-cost differential robot platform developed at the Robotics and Artificial Vision Laboratory of the University of Extremadura. The used cameras are USB webcams working at 20Hz. Both cameras are mounted on a three degrees of freedom head system. The computation is done on-board in real-time, by a laptop carried by the robot. Even using a low cost robot, and cheap cameras, the robot is able to navigate autonomously it real time on a wide variety of environments. The robot is shown in figure 4.

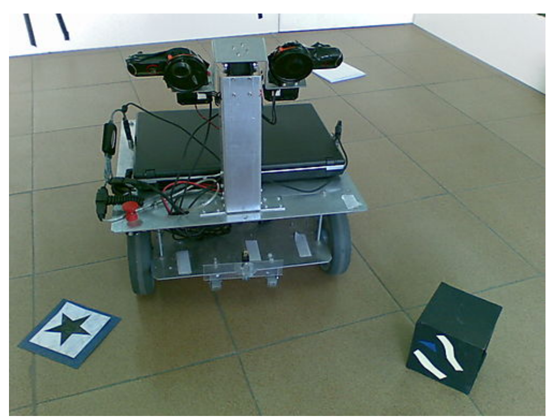


Fig. 4: RobEx is the robot we used in our experiments.

As seen in section II, light reflections depend on the point of view of the camera, so they may lead to a wrong classification. Despite the impact of light reflections on the geometry algorithm is minimized using the second test, it may not be enough depending on the material of the floor. A useful property of electromagnetic waves, such as light, is that when they are reflected, they get polarized at some extent. We use this property to decrease the appearance of these reflections on the images by providing the cameras with circular polarizing filters, which remove polarized light reflections. Reflections are closer to full polarization as they approach the so called *Brewster's angle*[14], which depends, in this case, on the refractive index of air and floor. Although angles far from the Brewster's angle are only partially polarized and hence the polarizing filters do not remove the reflections completely, they have been proved to be helpful.

B. Single vs. Multiple cue navigation

Single cue navigation was performed on both structured and unstructured environments. The results obtained with the color based approach outperformed the ones obtained with previous similar methods, but it is still not reliable by itself. Ground and obstacles some times share the same color.

Results obtained with the geometric approach clearly outperformed the ones obtained with previous similar methods. The overhead of the second test is negligible, as it is done only on candidate windows. In spite that the geometric obstacle detection is enough on unstructured environment where untextured image areas are very rare (with exception of the sky which is not an obstacle), on structured environments may be problematic.

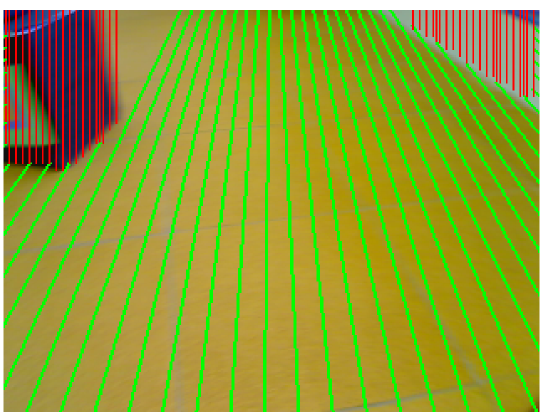


Fig. 5: The obstacle and the wall are detected.

C. Sixty minutes challenge using multiple cues

When the obstacle detection system development was started, a sixty minutes challenge was established as a measure of reliability. The challenge was passed in exceed when using cue fusion. In fact, it has passed several times a sixty minutes challenge. With geometric single-cue navigation, the challenge was also passed in unstructured environments (textured), but it failed in structured scenarios with untextured walls. On single cue navigation, the results depended heavily on the color of the ground and the obstacles.

The scenario in which the sixty minutes tests have been succesfully accomplished is a specially tailored environment

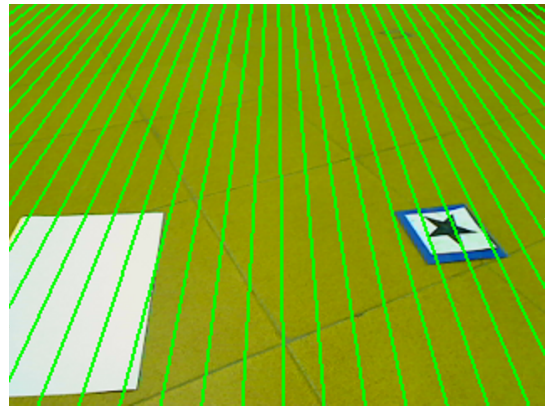


Fig. 6: Both, the paper sheet and the landmark, are discarded as obstacles.

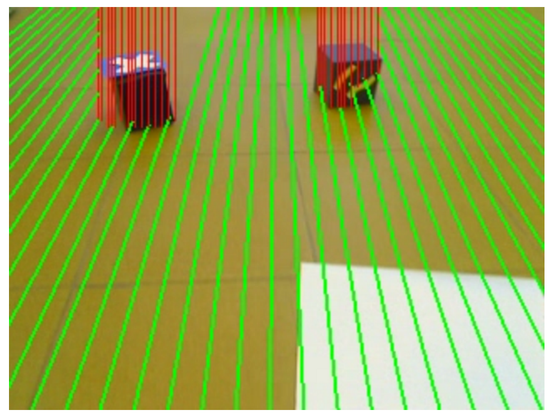


Fig. 7: Obstacles are detected, the paper sheet lying on the floor is discarded.

with tricky obstacles and non-obstacle objects with different heights, shapes and colors, as well as light reflections. A picture of this environment can be seen in figure 8, where the trajectory of a 7-minute wander is also shown.

D. Real time homography estimation

Due to the problems seen in II, specifically camera-camera desynchronization, camera-head position sensor desynchronization, and stereo head pose uncertainty, the approaches in [6], [15] are error prone when using motorized stereo heads. Because of the second test of the geometric classifier, our method discards all false positives with reasonable desynchronizations or pose estimation errors.

In our experiments, the real time homography estimation performed successfully even with rough camera calibrations.

VI. Conclusions

In this paper we present two improvements to previous single cue detectors and a new technique for merging their outputs. The new single cue detectors, appearance and geometry based, outperform their previous counterpart. In addition, the proposed method for cue fusion clearly outperforms any

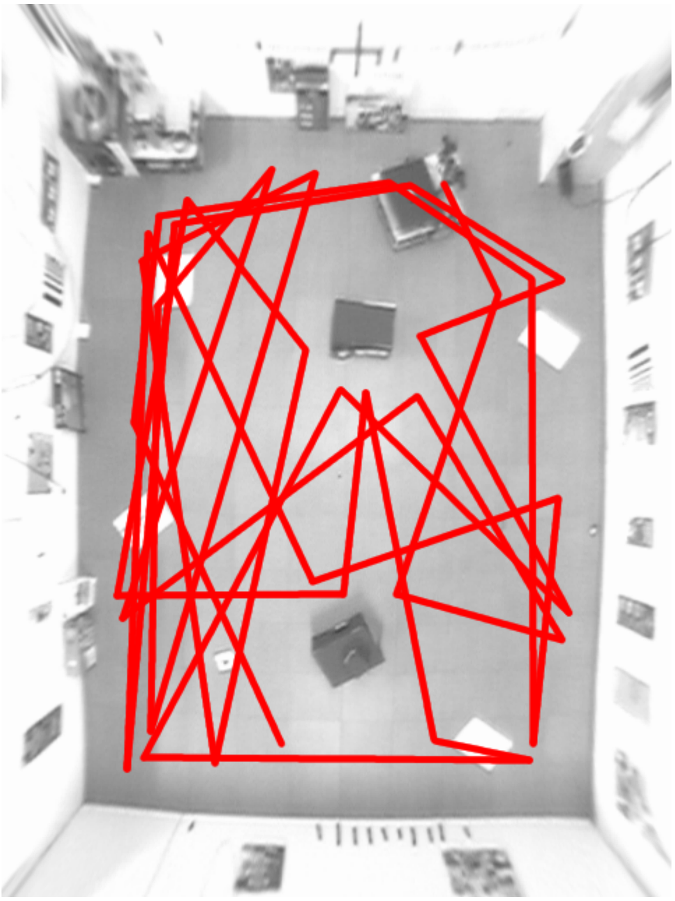
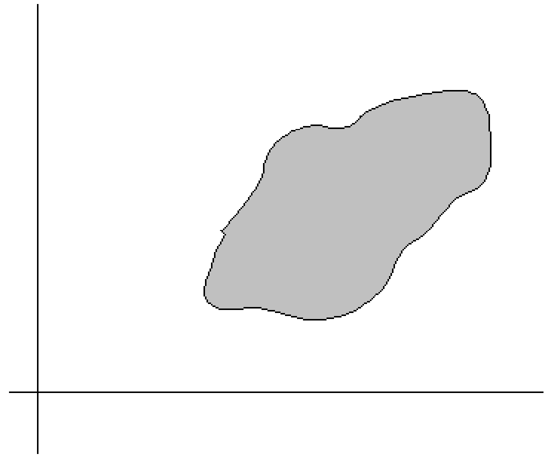
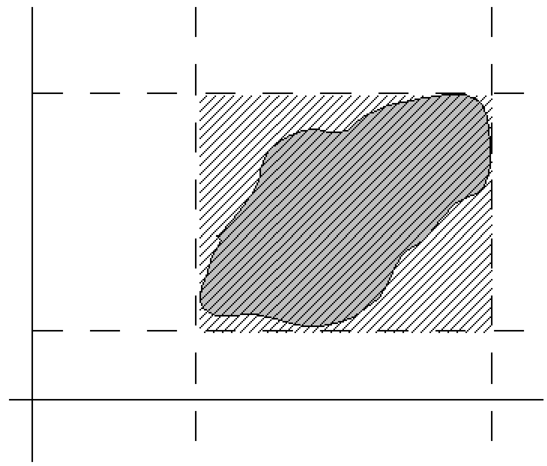


Fig. 8: RobEx robot trajectory using a naive wandering algorithm.



(a) Discriminative power of a 2-dimensional histogram.



(b) Discriminative power of two 1-dimensional histograms.

Fig. 9: Comparison between the discriminative power of a 2-dimensional histogram and two 1-dimensional histograms. This can be extended to a third dimension. The striped area of figure delimits the boundary of two 1-dimensional histogram while the grey area delimits the frontier of the 2-dimensional histogram.

single cue detector, as well as the multiple-cue method detailed in [6]. Due to the variety of obstacles present in the tests, we have been able to demonstrate that our approach allows the robot to navigate through a wide repertoire of conditions. The computation is light enough to be carried in real-time on an average laptop, coexisting with other heavier software components.

In order to improve significantly the system performance, our view is that it will be required a richer environment representation. The only situation in which the algorithm does not work as it would be desirable is when facing untextured obstacles with the same color as the ground (see picture 2). Generally these sitations can be easily overcome with a dense, symbolic, geometric representation of the environment that can only be achieved using knowledge-based modelling techniques in conjunction with state-estimation algorithms. Steps are being taken towards this goal.

REFERENCES

[1] Richard Hartley and Andrew Zisserman, "Multiple View Geometry in Computer Vision", Cambridge University Press, 2004.

[2] J. Gaspar and J. Santos-Victor and J. Sentieiro, "Ground plane obstacle detection with a stereo vision system", 1994.

[3] Jorge Lobo and Jorge Dias, "Ground Plane Detection using visual and inertial data fusion", 1998.

[4] Zhou and Baoxin Li", "Robust ground plane detection with normalized homography in monocular sequences from a robot platform", 2006.

[5] Iwan Ulrich and Illah Nourbakhsh, "Appearance-Based Obstacle Detection with Monocular Color Vision", *"in Proceedings of the AAAI National Conference on Artificial Intelligence*, 2000.

[6] Parag Batavia and Sanjiv Singh, "Obstacle Detection Using Adaptive Color Segmentation and Color Stereo Homography", *in International Conference on Robotics and Automation*, IEEE; 2001.

[7] Y.I. Abdel-Aziz and H.M. Karara, "Direct linear transformation from comparator coordinates into object space coordinates in close-range photogrammetry", *in Proceedings of the Symposium on Close-Range Photogrammetry*, American Society of Photogrammetry, 1971.

[8] Wolfgang Boehler and Andreas Marbs, "Investigating Laser Scanner Accuracy", 2003.

[9] Jin Zhou and Baoxin Li, "Homography-based ground detection for a mobile robot platform using a single camera", *in International Conference on Robotics and Automation*, IEEE; 2006.

[10] Thomas Bergener, Carsten Bruckhoff and Christian Igel, "Evolutionary parameter optimization for visual obstacle detection", *in The International Institute for Advanced Studies in Systems Research and Cybernetics*, 2000.

[11] Iwan Ulrich and Johann Borenstein, "VFH*: Local Obstacle Avoidance with Look-Ahead Verification", *in International Conference on Robotics and Automation*, IEEE, 2000.

[12] Darius Burschka, Stephen Lee and Gregory Hager, ”Stereo-based obstacle avoidance in indoor environments with active sensor re-calibration”, *in Proceedings of Intelligent Robots and Systems*, 2003.

[13] Scott Lenser and Manuela Veloso, ”Visual Sonar: Fast Obstacle Avoidance Using Monocular Vision”, *in International Conference on Robotics and Automation*, IEEE, 2002.

[14] A. Lakhtakia, ”General schema for the Brewster conditions”, *Optik*, Hecht, 1992.

[15] Yong Chao and Zhu Changan, “Obstacle Detection using Adaptive Color Segmentation and Planar Projection Stereopsis for Mobile Robots”. International Conference on Robotics, Intelligent Systems and Signal Processing, 2003.