# Identifying free space in a robot bird-eye view

Maillette de Buy Wenniger, G.E.; Schmits, T.; Visser, A.

**Publication date**
2009

**Published in**
Proceedings of the 4th European Conference on Mobile Robots, ECMR '09, September 23 - 25, 2009, Mlini/Dubrovnik, Croatia

# Identifying Free Space in a Robot Bird-Eye View

Gideon Emile Maillette de Buy Wenniger* Tijn Schmits† Arnoud Visser†
*Autonomous Intelligent Systems Group, Rheinische Friedrich-Wilhelms-Universität Bonn, Germany
†Intelligent Systems Laboratory Amsterdam, Universiteit van Amsterdam, the Netherlands

***Abstract*— Free space detection based on visual clues is an upcoming approach in robotics. Our working domain is the Virtual Rescue League of the RoboCup. In this domain efficient obstacle avoidance is crucial to find victims under challenging conditions. In this study a machine-learning approach is applied to distinguish the difference in visual appearance of obstacles and free space. Omnidirectional camera images are transformed to bird-eye view, which makes comparison with local occupancy maps possible. Bird-eye view images are automatically labeled using Laser Range information, allowing completely autonomous and continuous learning of accurate color models. Two color-based models are compared; a Histogram Method and a Gaussian Mixture Model. Both methods achieve very good performances, with results in a high precision and recall on a typical map from the Rescue League. The Gaussian Mixture Model achieves the best scores with much less parameters on this map, but is beaten by the Histogram Method on real data collected by our Nomad robot. Additionally, the importance of the right color normalization scheme and model parameters is demonstrated in this study.**

***Index Terms*— visual obstacle detection, machine learning, omnidirectional vision**

Collecting accurate information about a robot's environment is a very important aspect of robotics, especially when the environment in which the robot performs its task is unstructured. Hence, many methods have been devised to extract information about a robot's environment based on a broad spectrum of sensor types (i.e. visual, sonar, laser, radar, etc.). There can be many aspects of interest in an environment which detection methods have been designed to detect. This paper focuses on the extension of the perception system of a rescue robot which could be used in the Virtual Robot competition of the RoboCup Rescue League.

The motivation behind using a visual sensor to detect free space is the fact that visual sensors are passive sensors. Until now, to detect obstacles and free space, rescue robots used active sensors which emit a beam and analyze the reflections (i.e. sonar scanners, laser scanners, etc). Though being very accurate, active sensors have limited range and most active sensor implementations have a limited field of view. Additionally, active sensors are relatively heavy and consume considerable amounts of energy, which makes them less attractive for small mobile robots. In contrast, the limit of a visual sensor range can lie as far as the horizon and omnidirectional vision methods can provide a 360° view of the environment. A method to identify free space based on visual sensor data could well expand the environment observation quality of a rescue robot. An instance of such a method has been applied previously with great success to the DARPA Grand Challenge, were automobile robots must detect and avoid obstacles at great speeds [13].

Indoor this technique is typically applied in combination with sonar measurements. With laser range scanning they training can be much more accurate. Further, two existing techniques for object classification based on color are compared, including an extensive study of the sensitivity of the parameters. Such a sensitivity study was not available for robot applications. We combine bird-eye view images vision with laser scanning measurements. This innovative combination makes the automatic labeling of training images very accurate and straightforward. Maps can be effectively build from the the labeled bird-eye view images, in combination with localization. With normal perspective images this task would be very complex. This makes our approach also particularly suitable for smaller autonomous mobile robots were online map building is an important issue.

In this paper two color based statistical models are compared which are used to identify free space in bird-eye perspective images of the robot's environment: a RGB color histogram pixel classifier and a hue/saturation mixture of Gaussians pixel classifier. Using as reference laser range scanner data to identify free space in the immediate surroundings of the rescue robot, these statistical model based classification methods can be trained on the spot. The same laser range data, acquired elsewhere on the map, is used as reference to accurately test the precision and recall of the methods. In this paper a comparison is provided of their performance in different environments and under different circumstances.

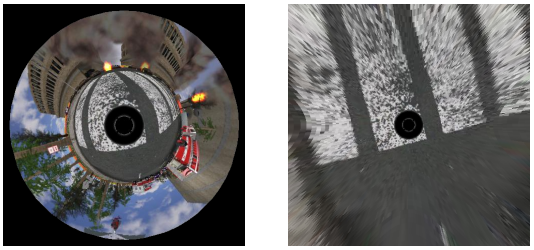
In Sect. I we first describe the environment in which the identifiers have been tested as well as all the methods which contributed to the development and validation of the free space identifier. In this section we also describe the measures which have been used to represent the performance of the free space identifier methods. In Sect. II an extensive analysis of the test results is made. In Sect. III we discuss the consequences of these test results. We end this paper with a section containing our conclusions and an indication of the limits and possible usage of these methods.

### A. Related Work

Shakey, the first autonomous mobile robot, already used a simple form of visual obstacle detection [8]. In the artificial environment of textureless floor tiles, simple edge detection sufficed to detect obstacles. Ulrich and Nourbakhsh [14] describe a color-based obstacle detection method used on a mobile robot. In their work they use a mobile robot with a normal perspective camera, combined with information about the robot's trajectory to find image pixels which are certain

to belong to empty space. Sun et al. [12] combine color-based obstacle modeling with a "Learning from Example" method, to perform automatic region-preferability learning, planning and navigation in unstructured outdoor terrains. Their approach relies on a combination of binocular and monocular vision. In contrast, the work of Michels et al. [6] focuses on using only monocular information in combination with color and texture modeling to perform obstacle avoidance. First they train a classifier on a combination of real and synthetic labeled data, that estimates depths from single monocular images. Reinforcement learning is next applied to a simulator of synthetic scenes to learn a control policy that selects a steering direction as a function of the vision system's output. Sridharan and Stone [11] use color information to perform structure based learning of colors for specific objects, with special attention to automatic model adaptation in case of changing illumination conditions. Rauskolb et al. [9] discuss an extension to the original *Stanley* [13] vision based obstacle detection used in the DARPA Grand Challenge. They improve the original algorithm to make it usable for the Urban Environments, were the usability of laser range scanners is decreased, and more preprocessing of the visual information is required.

For omnidirectional cameras, obstacles are typically detected from the optical flow, after removing the egomotion [2]. A nice example is the visual obstacle detection developed in the PERSES project [3]. Visual obstacle detection is accomplished by first creating bird-eye view transformations, and uses the difference to create panoramic optical flow images.



(a) Omnidirectional image data obtained from the simulation environment.

(b) A bird-eye view projection of Fig. 1a.

Fig. 1: Images depicting bird-eye view image transformation.

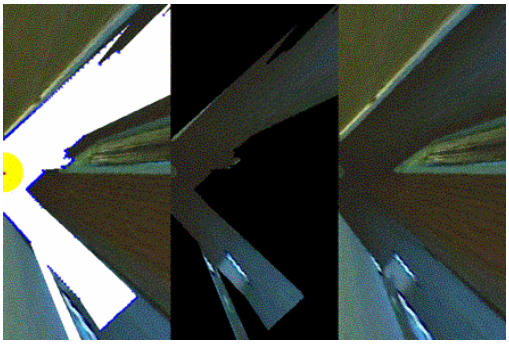


Fig. 2: Combination of bird-eye view and laser range data for the real dataset. On the left the occupancy grid obtained from the laser range data drawn on top of the bird-eye view image, in the middle the contained pixels in the free space and on the right the original bird-eye view image of a corridor in a home environment.

## I. Method

In this section the method is described with which free space pixel identification has been performed. First, omnidirectional views were collected from a simulated environment[1] or from a real dataset[2]. Image data obtained in this manner has then been transformed into bird-eye views. Then, two color pixel classifiers have been trained using laser range data to identify free space pixels. Finally, those pixel classifiers have been used to identify free space pixels and the results have been compared to free space pixels identified with laser range data to measure the performance of the pixel classifiers.

*1) Bird-Eye Views:* The simulation environment supports simulation of a catadioptric omnidirectional camera providing a 360° view of the robot's environment, which is used to create a bird-eye view of the environment, as depicted in Fig. 1. The omnidirectional camera uses texture projection to simulate the reflecting surface of a hyperbolic convex mirror and the data which the camera simulation model generates has been validated in [10].

Bird-eye views (both simulated and real) are obtained by radial correction around the image center which is the result of a scaled perspective projection of the ground plane. Nayar describes a direct relation between a location in a 3D environment and the location in the omnidirectional image where this point can be seen if nothing obstructs the view [7]. He describes the correspondence between a pixel in the omnidirectional image $p_{omni} = (x_{omn}, y_{omn})$ and a pixel in the birds-eye view image $p_{be} = (x_{be}, y_{be})$ to be defined by the following equations:

$$\theta = \arccos \frac{z}{\sqrt{x_{be}^2 + y_{be}^2 + z^2}}, \tag{1}$$

$$\phi = \arctan \frac{y_{be}}{x_{be}}, \quad \rho = \frac{h}{1 + \cos\theta} \tag{2}$$

$$x_{omn} = \rho \sin\theta \cos\phi, \quad y_{omn} = \rho \sin\theta \sin\phi \tag{3}$$

where $h$ is the radius of the circle describing the 90° incidence angle on the omnidirectional camera effective viewpoint. The variable $z$ is defined by the distance between the effective viewpoint and the projection plane in pixels. These equations can be used to construct perspectively correct images based on omnidirectional camera data by translating 3D projection plane pixel locations to omnidirectional pixel locations.

*2) Laser Range Data:* In the simulation environment the laser range scanner is simulated by ray tracing multiple lines from the sensor position in the simulated world. The sensor returns the distance between the sensor and the first line intersection with a surface (a 'hit point'), though if the range is beyond the sensor's detection range, the sensor will return the

[1]USARSim, the simulation environment used in the Virtual Robot competition of the RoboCup Rescue League.

[2]Radish dataset provided as part of the IROS 2006 Workshop "From sensors to human spatial concepts".

maximum detection range for that line. Before the simulation provides the data, a random number is added to simulate random noise (uniformly distributed) and a distortion curve is used to interpolate the range data to simulate a real laser range scanner.

Multiple laser-range measurements (both real and simulated) can be accumulated in an occupancy grid, which indicates the probability that free space is present as a number between 0 and 1. A binary image with the same dimensions as the bird-eye view can be created from this probability information, by thresholding the probability values. The information from the omnidirectional camera and the laser range scanner can easily be fused, as illustrated in Fig. 2. Note that the top of the robot is always visible in the omnidirectional camera. The pixels representing the robot have been indicated in yellow, and have not been used to train the classification methods.

### A. Color Pixel Classification Methods

The two classification methods studied in this article use statistical models to identify color pixels based on their $R$, $G$ and $B$ values. The concept behind these classifiers is using a large collection of pixels of which the class (i.e. *free space* or *non free space*) is known to determine the likelihood that a certain $rgb$ value belongs to a certain class. The same methods were successfully applied earlier in another domain by Jones and Rehg [4], to perform skin detection. As has been explained in Sect. I-.2, laser range data has been used to obtain a reference classification for this large set of pixels, which will be referred to as training data and testing data.

*1) Histogram Method:* The first method used to identify free space pixels was a pixel classifier based on a well established statistical model, the color histogram. The color histogram classifier uses training data to create a 3D histogram with a specified number of bins per color channel. The total number of bins ($n$), bins per color channel ($B_{cc}$) and bin-length per color channel ($L_{cc}$) are three interchangeable ways to specify the amount of bins used by the Histogram Method, as expressed by the following equation:

$$B_{cc} = 256/L_{cc}, \quad n = B_{cc}^3 \tag{4}$$

The histogram counts were then converted into a discrete probability distribution $P_{HIST}(\cdot)$:

$$P_{HIST}(rgb) = \frac{c[rgb]}{T_c} \tag{5}$$

where $c[rgb]$ gives the count in the histogram bin associated with the color $rgb \in RGB$ and $T_c$ is the total count obtained by summing the counts in all of the bins. A particular color $rgb$ is labeled positive by the classifier if

$$P_{HIST}(rgb) \geq \Theta \tag{6}$$

where $0 \leq \Theta \leq 1$ is a threshold which can be adjusted to trade-off between correct classifications and false positives. The sensitivity of this threshold is studied in section II. As the classifier is trained to detect the colors of free space, all pixels in the image can be labeled either positive or negative, indicating the presence of free space.

*2) Gaussian Mixture Model Method:* The second pixel classifier is based on another established statistical model, a Mixture of Gaussians. Initially, a number of three dimensional Gaussians is initiated in the $RGB$ color space by a K-Mean algorithm using all observed free space color values. Then, an EM algorithm optimizes the likelihood of the distributions with respect to these color values until convergence of the likelihood is observed. These Gaussian distributions can subsequently be converted to the continuous probability distribution $P_{GMM}(\cdot)$:

$$P_{GMM}(rgb) = \sum_{i=1}^{n} w_i \mathcal{N}_i(rgb) \tag{7}$$

where $n$ is the total number of normal distributions $\mathcal{N}(\cdot)$, and $w_i$ is the weight applied to the distribution $i$ and for which $\sum_{i=1}^{n} w_i = 1$. A particular color $rgb \in RGB$ is labeled positive by the classifier if

$$P_{GMM}(rgb) \geq \Theta \tag{8}$$

where $0 \leq \Theta \leq 1$ is again a threshold which can be adjusted to trade-off between missed detections and false positives. As the classifier is trained to detect the colors of free space, all pixels in a normalized image can be labeled either positive or negative, indicating the presence of free space.

### B. Normalized RGB Color Spaces

Important for obstacle detections applications is the sensitivity to shadows. A solution could be to perform the training in another color space, such as HSI or YUV. In the field of image processing many advanced techniques exist to enhance or modify image data for specific purposes. Multiple of these image modification techniques belong to the normalization category, which is based on normalizing the range of color values in an image. While there are multiple types of normalization, normalizing color intensity is one of the most commonly used [1]. This normalization is implemented by division through the total intensity, and provides invariance to lightning intensity differences. The method transforms each color $col : \{r, g, b\} \in [0, 255]^3$ in an image according to the following formula:

$$col_{norm} : \{r_{norm}, g_{norm}, b_{norm}\} = \{r, g, b\} * \frac{255}{r + g + b} \tag{9}$$

For this project the influence of this type of normalization on the effectiveness of the classification methods was investigated and compared to the performance of the classification methods based on the standard $\{r, g, b\}$ values.

### C. The Experimental Setup

Using the free space pixel identification method described in Sect. I-.2, both color pixel classifiers have been trained and tested in the following settings.

*1) Map 1: the Maze:* Synthetic testing data was obtained by the traversal of a simulated P2DX, a 2-wheel drive pioneer robot from ActivMedia Robotics, LLC., through a maze. The P2DX was mounted with both a SICK Laser Scanner LMS200 simulation model to obtain laser range data and a Catadioptric Omnidirectional Camera simulation model to obtain omnidirectional camera image data.

The maze is present on a map used in the 2006 RoboCup competition; DM-compWorldDay1_250.ut2[3]. At the start of each test run, the robot was spawned on a location inside the maze. Tele-operation was then used to drive around, collecting data for training and testing purposes. In this maze, the walls are dark green hedges and the floor consists of light green grass. The lighting conditions are synthetic, but realistic, with result in several shades of green inside the maze.

*2) Map 2: the Home:* Real testing data was obtained by traversal of our Nomad Scout II robot through a home environment [15]. The Nomad was equipped with a SICK Laser Scanner LMS200, and a Catadioptric Omnidirectional Camera constructed from a Dragonfly2 camera combined with a convex hyperbolic mirror from Accowle. For this dataset[4], the measurements collected in the corridor of the home were used. Here the walls are white and yellow, although many other colors are present (e.g. a brown coat rack). The floor is dark blue. Natural light was coming from the large windows in the living room and bed room, but the main source of light were the ceiling lamps.

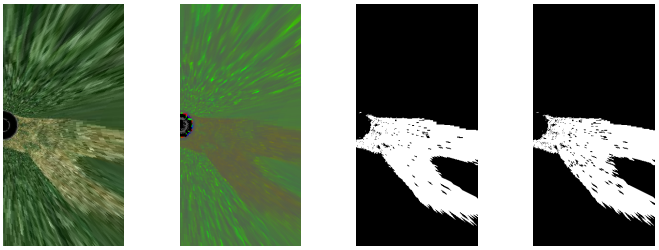
*3) Testing Parameters:* To properly analyze the behavior of both classification methods, method performance differences have been measured by varying the following parameters:

- Number of Mixtures/Bins $n$ - The amount of Mixtures or Bins influences the classifier performance, as too few mixtures/bins might make the classifier incapable of discerning required $rgb$ value differences, while too many mixtures/bins will result in overfitting of the statistical model on the training data.
- Normalized Color Space - As lighting in the environment greatly influences the intensity with which colors are observed, normalization is a classic method to improve classification of colored pixels [1]. However, as the normalization process effectively reduces the 3-D color space to a 2-D color plane, the Mixture Components can be trained with only two of the three normalized color channels. The Histogram Method keeps three dimensions but will effectively use only part of its bins.
- Threshold $\Theta$ - varying the threshold described in Equations 6 and 8 influences the amount of $rgb$ values labeled free-space pixels. The threshold value $\Theta$ dictates a trade off between correct classifications and false positives and produces optimal classification performance between 0 and 1.

Each combination of testing parameters $n$ and $\Theta$ were tested by performing twenty runs on images from the synthetic and real dataset which resulted in an estimate of the method performance including the variance. The used image set was deliberately kepts small to allow running tests with all parameter combinations within a reasonable time, it contained 28 images and was randomly split into half train half test images for every run. In practice a few images are already sufficient to train reliable color models.

[3]Available for download on: http://downloads.sourceforge.net/usarsim/

[4]Available for download on: http://radish.sourceforge.net/



(a) Bird-eye view image from the maze. (b) The Normalized version of Fig. 3a. (c) Free space detected by GMM classifier. (d) Free space detected by HIST classifier.

Fig. 3: Bird-eye view images from the maze and free space detection results.

## II. Experimental Results

In this section we provide detailed descriptions of experimental results obtained in the test runs described in the previous section.

### A. Performance Measurement

The performance of the two free space color pixel classifiers (Sect. I-A.1) is based on measurements of true positive $tp$ and true negative $tn$ pixel classifications versus false classifications $fn$ of both classes.

With these measurements the precision and recall can be calculated. Because both measures are important to build obstacle avoidance teams, our evaluation is based on the combination of the precision and recall. The F-measure is the harmonic mean of precision and recall. This measure is defined by the following formula:

$$precision = \frac{tp}{tp + fp}, \quad recall = \frac{tp}{tp + fn} \tag{10}$$

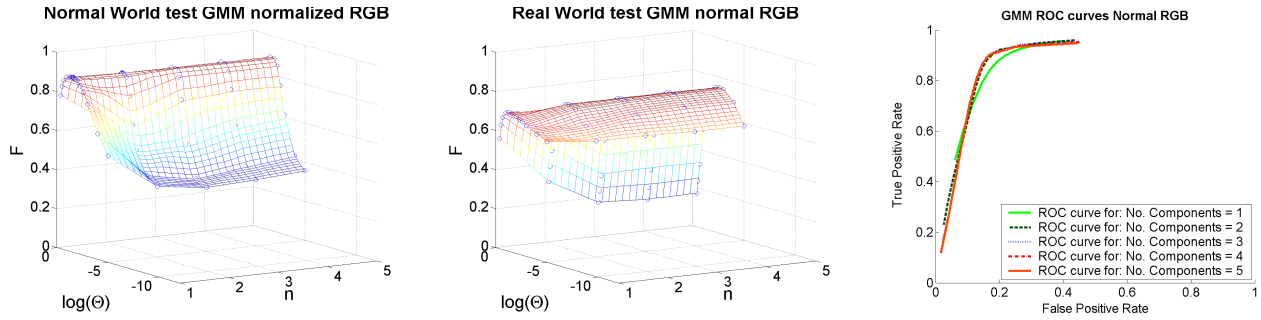
$$F = \frac{2 \cdot precision \cdot recall}{precision + recall} \tag{11}$$

Performance scores are computed directly from the free-space images generated by the classifiers, without any further post processing.

### B. Gaussian Mixture Model

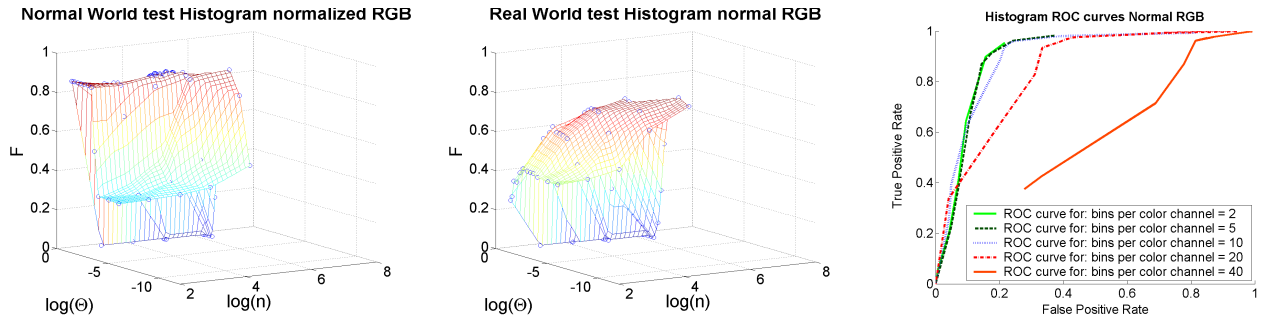
Figures 4a and 4b display 3D plots of the measured F-measure set against the logarithmic scale of the threshold, $log(\Theta)$ and the normal scale of the number of Mixtures, $n$. The two images display results obtained in the optimal color space for each map. Figure 4c displays the corresponding ROC curves [5]. The Gaussian Mixture Model is not very sensitive for the number of Mixtures, which is visible in the flat top in both 3D plots and overlap in the ROC curves.

The optimal F-measure measurements obtained in all four settings have been recorded in Table Ia. For map 1, the Maze, the GMM performed very well, with F-measure scores of nearly 90% for both the standard and normalized RGB.




(a) Map 1, Normalized RGB (b) Map 2, Standard RGB (c) Map 2, Standard RGB

Fig. 4: GMM Test Results. Note that the number of mixture components ($n$) is represented on a normal scale and threshold values $\Theta$ are represented on a logarithmic scale.




(a) Map 1, Normalized RGB (b) Map 2, Standard RGB (c) Map 2, Standard RGB

Fig. 5: HIST Test Results. Note that both the number of bins ($n$) and threshold values $\Theta$ are represented on a logarithmic scale.

For map 2, the Home, the performance is lower, but still respectable. Normalization decreases the performance for the real data. This is explained by the greater similarity between the ground color and the wall color after normalization for the real data.

### C. Histogram Method

Figures 5a and 5b display 3D plots of the measured F-measure set against the logarithmic scale of the threshold, $log(\Theta)$ and the logarithmic scale of the number of bins, $log(n)$. The two images display results obtained in the optimal color space for each map. Figure 5c displays the corresponding ROC curves. The Histogram method is clearly sensitive for the number of bins, which is visible in the curverture of the 3D plot in 5b and difference between the ROC curves.

The optimal F-measure measurements obtained in all four settings have been recorded in Table Ib. Using (4), the optimal $log(n)$ for the four settings can be translated back into the following bin-length $L_{cc}$ for the different settings: 12 for the Maze with Standard RGB, 7 for the Maze with Normalized RGB, and 4 for the Home with Standard and Normalized RGB.

For the Maze, the HIST method performed nearly as well as GMM on the synthetic data. For the Home, the HIST method performed slightly better than the GMM on the real data, while color normalization lowers the results again.

| **F**$\pm\sigma$ $\{n, log(\Theta)\}$ | Standard RGB | Normalized RGB |
|---|---|---|
| Map 1 the Maze | **0.867**$\pm 1.3e^{-2}$ $\{1, -1.602\}$ | **0.896**$\pm 6.2e^{-3}$ $\{1, -1.523\}$ |
| Map 2 the Home | **0,747**$\pm 2.1e^{-2}$ $\{4, -2.602\}$ | **0.640**$\pm 2.2e^{-2}$ $\{2, -1\}$ |

(a) Gaussian Mixture Model Results

| **F**$\pm\sigma$ $\{log(n), log(\Theta)\}$ | Standard RGB | Normalized RGB |
|---|---|---|
| Map 1 the Maze | **0.858**$\pm 1.7e^{-2}$ $\{3.967, -2.523\}$ | **0.890**$\pm 3.5e^{-3}$ $\{4.669, -2.155\}$ |
| Map 2 the Home | **0.753**$\pm 8.7e^{-3}$ $\{5.419, -3.301\}$ | **0.653**$\pm 2.7e^{-3}$ $\{5.419, -2.398\}$ |

(b) Histogram Method Results

TABLE I: Overview of the highest average F-measure scores $F$ in the 8 setups.

## III. Discussion

The test results reveal the success of using both methods to detect free space. Figure 3 gives an illustrative example of the final classification results obtained on the Maze using the best settings for both classifiers. The figure illustrates respectively the classification of the Gaussian Mixture Model (GMM) using only one component and the threshold with resulted in an F-measure of 0.896. This F-measure corresponds with a precision of 0.933 and a recall of 0.863. The Histogram Method (HIST) classifier has an almost equally high performance.

The best HIST classifier in this test has a bin size of 7. It thus has approximately $(256/7)^3 \approx 50000$ bins and as much parameters. In comparison, the GMM classifier with only one mixture component has only 12 parameters. The HIST classifier requires more memory and has a higher likeliness to overfit the data. Training a Histogram has complexity linear in the data size, while training a GMM with the EM algorithm has complexity quadratic in the number of mixture components and linear in the data size. With a small number of mixture components, this makes no big difference however. Given that the GMM classifier achieves a higher accuracy using a far lower amount of parameters, we believe it is the preferred classifier in the simulated world.

In the real world there should be enough variation in the perceived colors to train both the Gaussian Mixture Model and the Histogram Method more extensively. The likeliness of overfitting the Histogram Method on real data is strongly reduced. The optimal F-measure of 0.753 is achieved with the Histogram Method, with a corresponding precision of 0.917 and a recall of 0.644. When comparing the surfaces in the upper row of Fig. 5 with those on the lower rows it is clear that the F-measure surface is much more smooth for the real dataset.

Normalization of the color space eliminates the effect of shadows, and therefore positively effects the results in the synthetic world, where the color variations are mainly due to shadows. For the real data however, it also creates more overlap between the free space and non-free space color models, so there its total effect on performance is negative.

## IV. Conclusion

In this article a method is described which is able to learn to classify free space based on color information. The applicability of this method is demonstrated in a simulated world. This world allows to test in a controlled environment, with a constant texture on the floor. The simulated world has a constant but realistic lighting, and the same texture is seen in variety of shades. This explains why both tested methods work better for the Normalized color space in the Maze. Both methods where able to visually recognize free space with a F-measure of nearly 90%.

A real dataset was used to validate if this approach also valuable for real robots. The algorithm was still able to classify free space, although the performance dropped with nearly 15%. For the real data the Histogram Method outperformed the Gaussian Mixture Model. Note that the performance measure was calculated on pixel-level. Further processing of the free space pixels can remove solitary regions of free space, and group pixels into the large closed areas attractive for navigation algorithms.

We realize that this is only a proof by example. In the real world it is easy to find situations too difficult for this approach: corridors with walls and floor in nearly the same color, obstacles difficult to detect with range scanners and appearance (stairs), fast changing lighting conditions. On the other hand, situations suitable for this approach are also easy to find. Floors, walls and ceilings have often a constant color scheme. Corridors typically have no natural lighting conditions. More extensive experiments are needed to find the appropriate usage of this free space algorithm in robotics.

This vision based free space detection can be learned by robots equipped with both a camera and a laser range scanner and distributed wirelessly to other robots equiped with only a camera. This is a realistic scenario in the Rescue League where heterogeneous robot teams are needed to face all mobility challenges in a disaster setting.

## References


[1] T. Gevers and A. W. M. Smeulders, "Color-based object recognition", *Pattern Recognition*, volume 32(3):pp. 453–464, March 1999.

[2] J. Gluckman and S. K. Nayar, "Ego-Motion and Omnidirectional Cameras", in "ICCV '98: Proceedings of the Sixth International Conference on Computer Vision", pp. 999–1005, IEEE Computer Society, Washington, DC, USA, 1998, ISBN 81-7319-221-9.

[3] H.-M. Gross, H.-J. Boehme and T. Wilhelm, "A contribution to vision-based localization, tracking and navigation methods for an interactive mobile service-robot", in "Proceedings of the IEEE International Conference on Systems, Man, and Cybernetics", volume 2, pp. 672–677, 2001, doi:10.1109/ICSMC.2001.972991.

[4] M. Jones and J. Rehg, "Statistical Color Models with Application to Skin Detection", *International Journal of Computer Vision*, volume 46(1):pp. 81–96, 2002.

[5] W. J. Krzanowski and D. J. Hand, *ROC Curves for Continuous Data*, Chapman & Hall/CRC, Boca Raton, FL, USA, 2009, ISBN 1-43-980021-9.

[6] J. Michels, A. Saxena and A. Y. Ng, "High Speed Obstacle Avoidance using Monocular Vision and Reinforcement Learning", in "Proceedings of the 22nd International Conference on Machine Learning", pp. 593–600, 2005.

[7] S. K. Nayar, "Catadioptric Omnidirectional Camera", in "IEEE Conference on Computer Vision and Pattern Recognition (CVPR)", pp. 482–488, Jun 1997.

[8] N. J. Nilsson, "Shakey The Robot", Technical Report 323, AI Center, SRI International, 333 Ravenswood Ave., Menlo Park, CA 94025, Apr 1984.

[9] F. W. Rauskolb, K. Berger, C. Lipski *et al.*, "Caroline: An Autonomously Driving Vehicle for Urban Environments", *Journal of Field Robotics*, volume 25(9):pp. 674–724, 2008.

[10] T. Schmits and A. Visser, "An Omnidirectional Camera Simulation for the USARSim World", in "Proceedings of the 12th RoboCup International Symposium", July 2008, proceedings CD. To be published in the Lecture Notes on Artificial Intelligence series.

[11] M. Sridharan and P. Stone, "Structure-based color learning on a mobile robot under changing illumination", *Autonomous Robots*, volume 23(3):pp. 161 – 182, October 2007.

[12] J. Sun, T. Mehta, D. Wooden *et al.*, "Learning from Examples in Unstructured, Outdoor Environments", *Journal of Field Robotics*, volume 23(11-12):pp. 1019–1036, 2007.

[13] S. Thrun, M. Montemerlo, H. Dahlkamp *et al.*, "Stanley: The robot that won the DARPA Grand Challenge", *Journal of Field Robotics*, volume 23(9):pp. 661–692, 2006.

[14] I. Ulrich and I. Nourbakhsh, "Appearance-Based Obstacle Detection with Monocular Color Vision", in "Proceedings of AAAI Conference", pp. 866–871, 2000.

[15] Z. Zivkovic, O. Booij and B. Kröse, "From images to rooms", *Robotic and Autonomous Systems*, volume 55(5):pp. 411–418, 2007.