Identifying Free Space in a Robot Bird-Eye View

Gideon Maillette de Buy Wenniger Tijn Schmits

Universiteit van Amsterdam, 1098 SJ Amsterdam, the Netherlands

Abstract

Vision based free space detection is an upcoming approach to obstacle detection in robotics. The problem of empty space detection is typically solved with active sensors. Vision has the advantage of providing a larger range, facilitiating more efficient navigation by looking further ahead. Our working domain is the the Virtual Rescue League of the RoboCup. In this domain efficient navigation is crucial to finding the victims fast enough. In this domain a learning approach is applied to distinguish the difference in appearance of obstacles and free space. In this study two color-based models are compared; a Histogram Method and a Gaussian Mixture Model (GMM). Both methods achieve very good performances, with results in a high precision and recall on a typical map from the Rescue League. The GMM achieves the best scores with much less parameters on the normal map, but is beaten by the Histogram Method on an artificial map with many pure colors. Additionally, the importance of the right color normalization scheme and model parameters is demonstrated in this study.

1 Introduction

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 Rescue League at the RoboCup.

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 sensor 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 omnidirecional 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 succes to the DARPA grand challenge, were automobile robots must detect and avoid obstacles at great speeds [4].

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 laser-range sensor 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

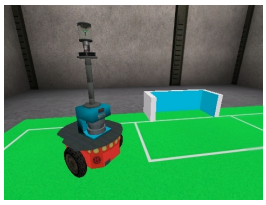
environments and under different circumstances.

In Section 2 we first describe the environment in which the identifyers 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 identifyer methods. In Section 3 an extensive analysis of the test results is made. In Section 4 we discuss these test results as well as possible future work which might improve the quality of the free space identifyer. We end this paper with a section containing our conclusions.

1.1 Related Work

Shakey, the first autonomous mobile robot, already used a simple form of appearance-based obstacle detection [12]. In the artificial environment of textureless floor tiles, simple edge detection sufficed to detect obstacles. Ulrich and Nourbakhsh [16] describe a color-based obstacle detection method used on a mobile robot. In their work they use a (real) mobile robot with a normal camera, combined with information about the robots trajectory to find image pixels which are certain to belong to empty space. Sun et al. [3] 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. [10] 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 [15] 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. [2] discuss an extension to the original *Stanley* [4] 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 scanners is decreased, and more preprocessing of the visual information is required.

(a) A view of a disaster area in the simulation environment USARSim.



(b) A simulated P2DX robot mounted with a catadioptric omnidirectional camera and a SICK laser range sensor.

Figure 1: Images taken from the USARSim simulation environment.

For omnidirectional cameras, obstacles are typically detected from the optical flow, after removing the egomotion [7]. A nice example is the visual obstacle detection developed in the PERSES project [8]. Visual obstacle detection is accomplished by first creating bird-eye view transformations, and use the difference to create panoramic optical flow images.

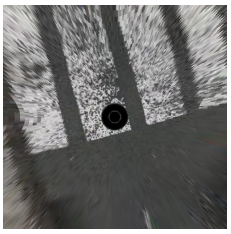
2 Method

In this section the method is described with which free space pixel indentification has been performed. First, omnidirectional views were created in a high fidelity robot simulation environment. Image data obtained in this manner has then been transformend into bird-eye views. Then, two color pixel classifiers have been trained using simulated 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 meaure the performance of the pixel classifiers.

2.1 The Simulation Environment

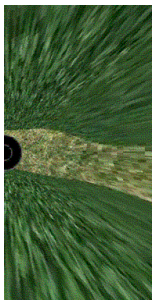
The environment chosen to do research is the simulation environment called USARSim. USARSim is a platform-independent open source project built on top of a 3D game engine created by Epic Games™. It is a high fidelity simulation environment, which means it utilizes realistic materials and

(a) Omnidirectional image data obtained from USARSim.



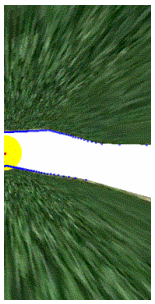
(b) A bird-eye view projection of Figure 2(a).

Figure 2: Images depicting bird-eye view image transformation.



(a) A bird-eye view.

(b) Free space identified by the laser range finder.



(c) Free space identified by the laser range finder projected on the bird-eye image.

Figure 3: Free space pixel identification using a laser range sensor.

equipment to represent the tasks that USAR robots must perform. Figure 1 depicts both a simulated disaster area as well as a simulated robot mounted with two sensors used for free space identification. Validation experiments performed on several visual perception algorithms [1] have shown that perception algorithms developed in USARSim can be directly transfered to reality. This make USARSim a valuable development environment for perception algorithms.

2.1.1 Simulated Bird Eye Views

The 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 Figure 2. The omnidirectional camera uses texture projection to simulate the reflecting surface of a parabolic convex mirror and the data which the camera simulation model generates has been validated in [14].

Bird-eye views 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 [11]. 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:

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

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

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

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.

2.1.2 Simulated Laser Range Data

The laser range sensor is simulated by ray tracing multiple lines from the sensor position in the Unreal 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 sensors detection range, the sensor will return the maximum detection range for that line. Before USARSim provides the data, a random number is added to simulate random noise and a distortion curve is used to interpolate the range data to simulate a real laser range sensor.

Multiple laser-range measurements can be accumulated in an occupancy grid, which indicates the probability that free space is present as a number between 0 and 1. This probability information can be thresholded into a binary image which the same dimensions as the bird-eye view (see Figure 3(b)). The information from the omnidirectional camera and the laserscanner can easily be fused, as illustrated in Figure 3(c)). 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.

2.2 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 likelyhood that a certain rgb value belongs to a certain class. As has been explained in Subsection 2.1.2, laser range data has been used to obtain this large set of pixels, which will be referred to as training data and testing data.

2.2.1 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 three dimensional histogram with a specified number of bins per color channel. The histogram counts were then converted into a discrete probability distribution $P_{HIST}(\cdot)$:

$$P_{HIST}(rgb) = \frac{c[rgb]}{T_c} \tag{4}$$

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 labelled positive by the classifier if

$$P_{HIST}(rgb) \geq \Theta \tag{5}$$

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

2.2.2 Gaussian Mixture Model Method

The second pixel classifier is based on another established statistical model, a Nixture 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 likelyhood of the distributions with respect to these color values until convergence of the likelyhood is observed. These Gaussian distributions can then be converted to the continuous probability distribution $P_{GMM}(\cdot)$:

$$P_{GMM}(rgb) = \sum_{i=1}^{n} w_i N_i(rgb) \tag{6}$$

where n is the total number of normal distributions $N(\cdot)$, 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

$$P_{GMM}(rgb) \geq \Theta \tag{7}$$

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.

2.2.3 Normalized RGB Color Spaces

In the field of image processing many 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. This normalisation 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

$$col_{norm} : \{r_{norm}, g_{norm}, b_{norm}\} = \{r, g, b\} * \frac{255}{r + g + b} \quad (8)$$

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.

2.3 The Experimental Setup

Using the free space pixel identification method described in Subsection 2.1.2, both color pixel classifiers have been trained and tested in the following settings.

2.3.1 USARSim Setup

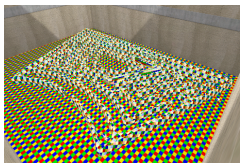
Testing data was obtained by simulating the traversal of a P2DX, a 2-wheel drive pioneer robot from ActivMedia Robotics, LLC., through two different mazes. The P2DX was mounted with both a SICK Laser Scanner LMS200 simulation model to obtain laser range data and a Catadioptric Omnidirectional Camera simulation model to obtain omnidirectional camera image data.

The first maze which the robot traversed is a specific area present in a map used in the 2006 RoboCup competition, DM-compWorldDay1_250.ut2[1], which is depicted in Figure 4(a). At the start of each test run, the robot was spawned in the middle of the maze. Tele-operation was then used to drive around, collecting data.

The second map, depicted in Figure 4(b), has specifically been designed for this study to observe a distinct difference in performance between the two classification methods and their performance both standard and normalized RGB color spaces. The wall (which defines non-free space) is textured with a simple four color image identical to the image on the floor of the maze (which defines free space) except for image intensity. This means that

[1]Available for download on: \\http://downloads.sourceforge.net/usarsim/ - last accessed: October 2008

(a) DM-compWorldDay1_250.ut2 map



(b) Modified DM-compWorldDay1_250.ut2 map

Figure 4: Views on the maze maps used for testing.

normalized rgb values defining free and non-free pixels will lie very close to each other in the normalized color space. Also, as the intensity difference of the wall texture is both positive and negative, free-space rgb values will lie in between obstacle rgb values in the standard RGB space. The influence of this environment on the classification methods is portrayed in Section 3 and will be discussed in Secion 4.

2.3.2 Testing Parameters

To properly analyse the behaviour of both classification methods, method performance differences have been measured by varying the following parameters:

- Number of Mixtures/Bins - 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 much 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 [6]. However, the influence of normalizing image data rgb values on classifier performance closely relates to the size of Mixtures/Bins used by the classifier. As the normalization process effectively reduces the threedimensional color space to a two-dimensional color plane, the Mixture Compnents 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 Θ - varying the threshold described in Equations 5 and 7 influences the amount of rgb values labeled free-space pixels. The threshold value Θ dictates a tradeoff between correct classifications and false positives and produces optimal classifaction performance between 0 and 1.

All parameter settings were tested by performing twenty runs in the simulation environment to make measurements reliable. The training set consisted of 14 images, and the test set also consisted of 14 images.

3 Experimental Results

In this section we provide detailed descriptions of experimental results obtained in the test runs described in Subsection 2.3.

3.1 Performance Measurement

The performance of the two free space color pixel classifiers (Sections 2.2.1 and 2.2.2) is based on measurements of true positive and true negative pixel classifications versus false classifications of both classes. With these measurements we can define the following performance measures:

3.1.1 Precision

The precision of the classification methods is defined by the following formula:

$$precision = \frac{tp}{tp + fp} \tag{9}$$

where tp and fp define true and false positive pixel classifications respectively. This measure defines the probability that a positive pixel classification is correctly classified.

3.1.2 Recall

The recall of the classification methods is defined by the following formula:

$$recall = \frac{tp}{tp + fn} \tag{10}$$

where tp and fn define true positive and false negative pixel classifications respectively. This measure defines the probability that a pixel representing free space is positively labeled by the classifier.

3.1.3 F-measure

The recall of the classification methods is defined by the following formula:

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

where $precision$ and $recall$ are defined by Equations 9 and 10 respectively. This measure is the harmonic mean of precision and recall.

3.2 Results Test Runs

This subsection describes the results obtained performing the tests described in the previous section.

3.2.1 Gaussian Mixture Model

Figures 5(a) to 5(d) display three-dimensional plots of measured F-measure F set against the logarithmic scale of the threshold, $log(\Theta)$ and the logarithmic scale of the number of Mixtures/Bins, $log(n)$. The four images display results obtained in both maps and using both color spaces.

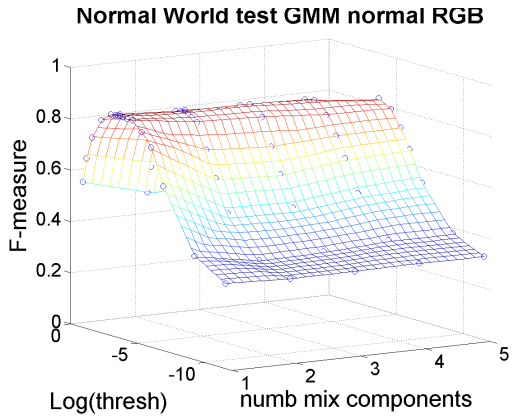
$\mathbf{F}\pm\sigma$ $\{No.Mix, \Theta\}$	Standard RGB	Normalized RGB
Map 1 Normal Maze	$\mathbf{0.867}\pm1.33e^{-2}$ $\{1, 0.025\}$	$\mathbf{0.896}\pm6.15e^{-3}$ $\{1, 0.03\}$
Map 2 Colored Maze	$\mathbf{0.44}5\pm9.22e^{-2}$ $\{6, 0.008\}$	$\mathbf{0.291}\pm7.70e^{-3}$ $\{7, 0.00004\}$

Table 1: GMM: Overview of the highest average F-measure scores F in the 4 setups.

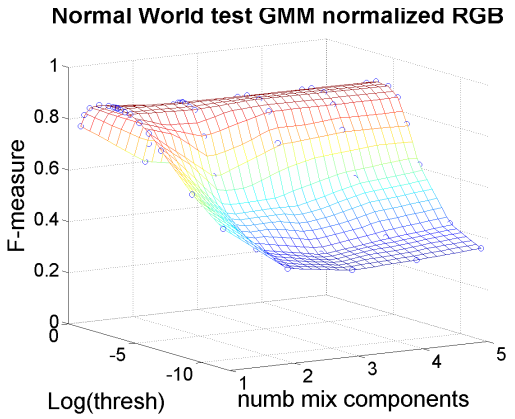
The optimal F-measure measurements obtained in all four settings have been recorded in Table 1. For the Normal maze, the GMM performed very well, with F-measure scores of nearly 0.9 for both the standard and normalized RGB.

3.2.2 Histogram Method

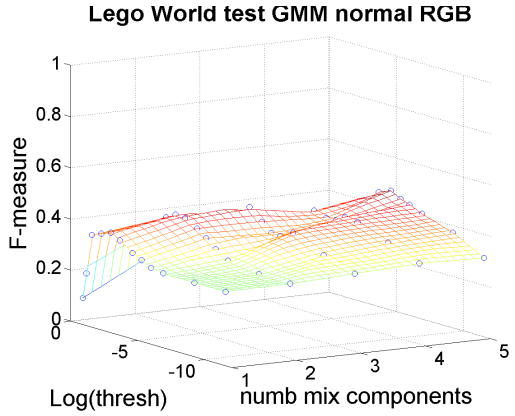
Figures 6(a) to 6(d) display three-dimensional plots of measured F-measure F set against the logarithmic scale of the threshold, $log(\Theta)$ and the logarithmic scale of the number of Mixtures/Bins, $log(n)$.



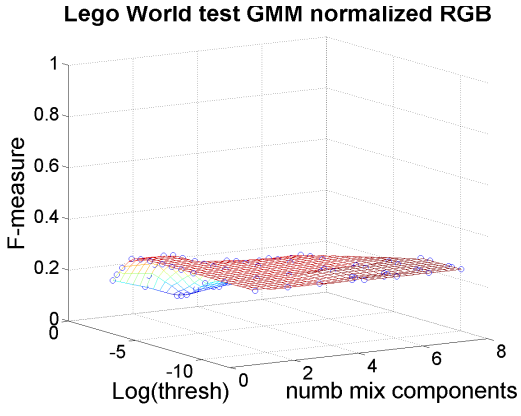
(a) Map 1, Standard RGB



(b) Map 1, Normalized RGB



(c) Map 2, Standard RGB



(d) Map 2, Normalized RGB

Figure 5: GMM Test Results. Note that $numberofMixtureComponents$ and $Threshold$ values are represented on a logarithmic scale.

$\mathbf{F}\pm\sigma$ $\{Binsize, \Theta\}$	Standard RGB	Normalized RGB
Map 1 Normal Maze	$\mathbf{0.85}8\pm1.67e^{-2}$ $\{12^3, 0.003\}$	$\mathbf{0.890}\pm3.53e^{-3}$ $\{7^3, 0.007\}$
Map 2 Colored Maze	$\mathbf{0.70}4\pm3.02e^{-2}$ $\{22^3, 0.001\}$	$\mathbf{0.323}\pm1.04e^{-3}$ $\{40^3, 0.007\}$

Table 2: HIST: Highest average F-measure scores F in the 4 setups.

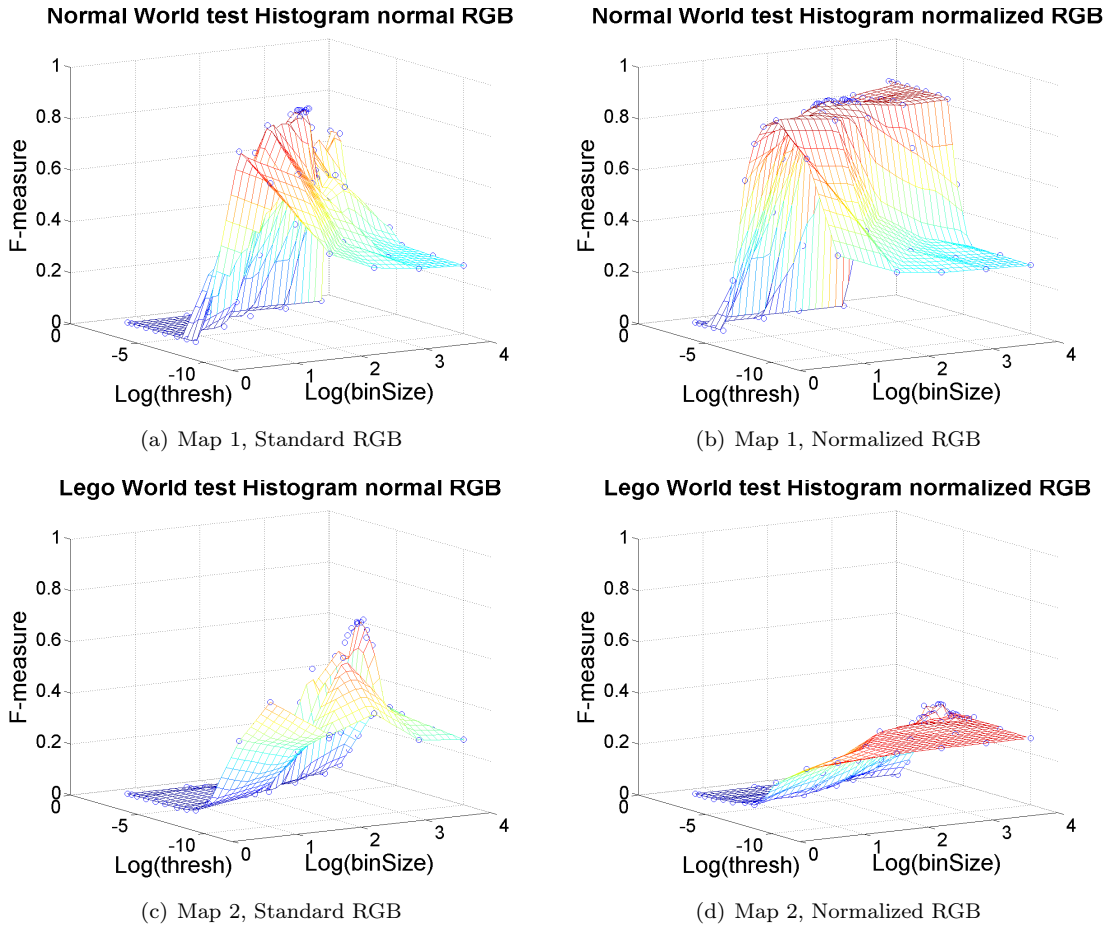
The four images display results obtained in both maps and using both color spaces.

The optimal F-measure measurements obtained in all four settings have been recorded in Table 2. For the Normal maze, the HIST method performed nearly as good as GMM. The HIST method applied in the standard RGB space was the only method capable of obtaining a reasonable F-measure of the Colored Maze.

4 Discussion

The test results reveal the succes of using both methods to detect empty space. They also reveal the importance of performing adequate preprocessing of the data, and choosing the right model parameters and thesholds.

The tests results obtained in the normal world show that the Gaussian mixture model classifier using only one component has the highest performance, with an F-score of 0.896. The color histogram classifier has an almost equally high per-



(a) Map 1, Standard RGB

(b) Map 1, Normalized RGB

(c) Map 2, Standard RGB

(d) Map 2, Normalized RGB

Figure 6: HIST Test Results. Note that $numberofBins$ and $Threshold$ values are represented on a logarithmic scale.

formance with an F-score of 0.89 using normalised images and 7 bins per color channel.

The Gaussian mixture classifier seems most preferable to use in this world, as a color histogram classifier with a bin size of 7 has approximateley $(256/7)^3 \approx 50000$ bins and as much parameters and in comparison, the Gaussian mixture model classifier with only one mixture component has only 12 parameters, which are the variables that denote the three dimensional component mean and covariance matrix. The color histogram classifier requires more memory and has a higher likeliness to overfit the data. Given that the Gaussian Mixture Model classifier achieves a higher accuracy using a far lower amount of parameters, we believe it is the preferred classifier in the normal world.

The Colored maze was especially designed to stress test the methods. The results obtained in the colored world show greater differences in classifier performance. In this world, color intensity is essential for separating the empty space from obstacles and, hence, color normalization decreases the performance both classifers dramatically. However, using original RGB image data, the color histogram classifier achieves an F-measure score of 0.7039, where the Gaussian mixture model classifier only achieves a an F-measure score of 0.4448.

Given that free-space rgb values will lie in between obstacle rgb values in the standard RGB space (see Section 2.3.1), this difference in performance can be explained by the fact that the color histogram classifier is less distracted with the

proximity of obstacle rgb color values. This allows the classifier to correctly classifiy a large amount of non-similar free-space rgb values in the RGB space, even when most lie relatively close to obstacle rgb values. In comparison, the Gaussian mixture classifier defines probability based on variation(s) and mean(s) and is hence less capable of creating discrete separations between the two pixel classes. Also, considering the distribution of the two classes in the RGB space, the K-Mean initialisation of the Gaussian mixture model classifier will more likely result in the EM-algorithm reaching a local optimum, dramatically influencing classifier performance. Hence, based on these results, we must conclude that the color histogram classifer is the preferred classifier when free-space colors lie dispersed in the RGB space but are relatively similar to non-free colors.

5 Further Work and 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. Notice that a constant texture on the floor is not uncommon in many applications. The simulated world has a constant but realistic lighting, and the same texture is seen in varity of shades. This explains why both tested methods work better for the Normalized color space in the Normal Maze.

Color constancy and color normalization play an important role in the correct detection of empty space under changing lightning conditions. The normalization method used for this research is unable to deal with changing illuminant color. More advanced color consistancy methods such as normalisation based on the *gray world assumption* or the *white patch assumption* [13] are able to remove the effect of a changing illuminance, so they could particularly improve performance in real world scenarios. Otherwise the free space model has to be dynamically retrained when the illuminant color changes. This requires the proximity of another robot equiped with both an camera and laser scanner.

An equally important improvement of our approach would be the automatic discovery of the best parameter settings for our classifiers. For the selection of the best number of components for a Gaussian mixture classifier, there exists some well established criteria such as the Baysian Information Criteria (BIC) and the Akaike information criterion (AIC) [9]. These methods attempt to balance performance gain against increase in model. A new Gaussian mixture model training algorithm that includes selection of the number of components as part of the algorithm is introduced in [5].

lxplicitly learning a statistical model of the appearance of the obstacles gives another opportunity for improvement. Instead of the current tuple (*free space*, *non-free space*), the tuple (*free space*, *obstacle*, *unknown*) is learned. The obstacle information is now ignored since it is typically more difficult to collect accurate obstacle color information than to get good empty space color information, while the latter is sufficient in principle. However, including obstacle information might make it easier to automate the optimal selection of thresholds, and improve general performance as well.

To summarize, we implemented and compared two methods for vision-based free space detection for robots in the rescue domain. Both the color histogram classifier and the Gaussian mixture model classifier method performed very well in the normal world with F-scores on free space recovery of nearly 0.90. The Gaussian mixture Model classifier performs better in scenarios with a relatively small amount of different colors defining free space, particulary given its better performance with a much lower amount of model parameters. In case of a large amount of non-similar free-space rgb values in the RGB space, the color histogram classifer proved to achieve better performance.

This vision based free based detection can be learned by robots equipped with both a camera and a laser sensor and distributed wirelessly to other robots equiped with only a camera.

References

[1] S. Carpin, M. Lewis, J. Wang, S. Balakirsky, and C. Scrapper. Bridging the Gap Between

Simulation and Reality in Urban Search and Rescue. In G. Lakemeyer, E. Sklar, D.G. Sorrenti, and T. Takahashi, editors, *RoboCup 2006: Robot Soccer World Cup X*, volume 4434 of *Lecture Notes on Artificial Intelligence*, pages 1–12, Berlin Heidelberg New York, October 2007. Springer.

[2] F. W. Rauskolb et al. Caroline: An autonomously driving vehicle for urban environments. *Journal of Field Robotics*, 25(9):674–724, 2008.

[3] J. Sun et al. Learning from examples in unstructured, outdoor environments. *Journal of Field Robotics*, 23(11-12):1019 – 1036, 2007.

[4] S. Thrun et al. Stanley: The robot that won the darpa grand challenge. *Journal of Field Robotics*, 23(9):661–692, 2006.

[5] M.A.T. Figueiredo and A.K. Jain. Unsupervised learning of finite mixture models. *IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE*, 24(3):381–396, March 2002.

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

[7] Joshua Gluckman and Shree K. Nayar. Ego-motion and omnidirectional cameras. In *ICCV '98: Proceedings of the Sixth International Conference on Computer Vision*, page 999, Washington, DC, USA, 1998. IEEE Computer Society.

[8] H.-M. Gross, H.-J. Boehme, and T. Wilhelm. Contribution to vision-based localization, tracking and navigation methods for an interactive mobile service-robot. volume 2, pages 672–677 vol.2, 2001.

[9] G. J. McLachlan and D. Peel. *Finite Mixture Models*. John Wiley and Sons, 2000.

[10] 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*, pages 593 – 600, 2005.

[11] Shree K. Nayar. Catadioptric Omnidirectional Camera. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 482–488, Jun 1997.

[12] Nils J. Nilsson. Shakey the robot. Technical Report 323, AI Center, SRI International, 333 Ravenswood Ave., Menlo Park, CA 94025, Apr 1984.

[13] A. Rizzi, C. Gatta, and D. Marini. Color correction between gray world and white patch. In *Proceedings of SPIE - The International Society for Optical Engineering*, volume 4662, pages 367–375, 2002.

[14] Tijn Schmits and Arnoud 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.

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

[16] Iwan Ulrich and Illah Nourbakhsh. Appearance-based obstacle detection with monocular color vision. In *Proceedings of AAAI 2000*, 2000.