Road Following in an Unstructured Desert Environment Based on the EM(Expectation-Maximization) Algorithm

Jaesang Lee and Carl D. Crane III[1]

[1] Center for Intelligent Machines and Robotics, University of Florida, Gainesville, Florida
(Tel : +1-352-392-9461; E-mail: ccrane@ufl.edu)

Abstract: This paper describes the development and performance of a vision system, named PFSS (Path Finder Smart Sensor) for autonomous navigation of an unmanned ground vehicle. A monocular camera and vision processing algorithms were used as the sensor system to identify traversable terrain. Unlike the Bayesian based method which was used by Team CIMAR in the 2005 DARPA Grand Challenge, the expectation-maximization (EM) algorithm is applied. The implementation and performance of this approach are reported here.

Keywords: autonomous vehicle, navigation, vision system, expectation-maximization (EM) algorithm, image segmentation

1. INTRODUCTION

1.1 DARPA Grand Challenge 2005 Event Description

The DARPA Grand Challenge is widely recognized as the largest and most cutting-edge robotics event in the world, offering groups of highly motivated scientists and engineers an opportunity to innovate in developing state-of-the-art autonomous vehicle technologies with significant military and commercial applications. The US Congress has tasked the military with making nearly one-third of all operational ground vehicles unmanned by 2015 and the DARPA Grand Challenge is one in a number of efforts to accelerate this effort. The goal of the event is to stimulate innovation and development in robotics by groups of engineers and scientists outside the normal military procurement channels including academia and private industry [1].

1.2 Vehicle and System Architecture for 2005 Event

Team CIMAR is a collaborative effort of the University of Florida's Center for Intelligent Machines and Robotics (CIMAR), The Eigenpoint Company of High Springs, Florida, and Autonomous Solutions of Young Ward, Utah. The team learned a tremendous amount from the 2004 event and used that experience to develop a new, highly advanced system to reach the finals of the 2005 Grand Challenge (See Figure 1).

Fig. 1: NaviGator vehicle.

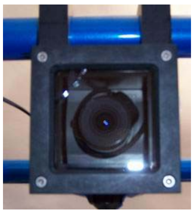
The vehicle architecture embodies four fundamental elements: Planning Element, Control Element, Perception Element and Intelligence Element. The Planning Element consists of components that act as a repository for a priori data, i.e. known roads, trails, or obstacles, as well as acceptable vehicle workspace boundaries. Additionally, these components perform off-line planning based on that data. The Control Element is comprised of components that perform closed-loop control in order to keep the vehicle on a specified path. The Perception Element incorporates components that perform the sensing tasks required to locate obstacles and to evaluate the smoothness of terrain. The Intelligence Element is comprised of components that act to determine the 'best' path segment to be driven based on the sensed information [1,2].

2. COLOR MONOCULAR VISION SYSTEM

2.1 Vision System description

The Pathfinder Smart Sensor (PFSS) is one component of the perception element and consists of a single color camera mounted in the sensor cage at the front of the vehicle and is oriented facing the terrain ahead [1]. Its purpose is to assess the area in the camera's scene for terrain which is similar to that on which the vehicle is currently traveling, and then translate that scene information into traversability information. The PFSS component uses a high-speed frame-grabber to store camera images at 30 Hertz. The camera incorporates an auto-iris lens and is protected from strong lightning conditions by a polarizing film. Fig 2 shows the PFSS system.

The primary feature used for analytical processing is the RGB (Red, Green, and Blue) color space. This is the standard representation in the world of computers and digital cameras and is therefore a natural choice for color representation. Furthermore, RGB is the standard output from CCD-cameras. The importance of being able to characterize the colors of the scene is apparent as, typically, roads have different color content than non-drivable terrain.



(a) Camera Assembly

(b) Sensor Cage

(c) Computer and Electronics Enclosure

Fig. 2: PFSS system

2.2 Training Area

Three subset image regions combined with vehicle pose information are used to define the road and the background training area. This positioning information was made readily available due to the architecture deployed on the NaviGator, specifically JAUS (Joint Architecture for Unmanned Systems) [4]. This architecture defines a set of versatile components and their interfaces, which allow the PFSS to obtain global position and pitch information from the position estimation components. Since the sky portion of the image hinders the classification procedure, the PFSS needs to perceive the sky line. Unlike the PFSS algorithm used for DGC 2005 [3], the NaviGator's pitch information defines the horizontal line for detecting the sky portion of the image as represented by the empirically determined red horizontal line shown in Fig 3. Next, a 160 × 50 sub-image is used to define the drivable area and two 40 × 60 sub-images are used to define the background. The drivable sub-image is placed in the bottom, center of the image (see Fig 3, blue box) while the background sub-images are placed at the middle-right and middle-left of the image(see Fig 3, orange box), which contain background area.

Fig.3 Training area at Nevada

Since the classification algorithm uses statistical information on the two sub-images, defining the correct drivable image and background image is critical. In real-world testing, many background areas were observed to be similar to drivable road, especially when the vehicle is undergoing sharp turns or when the vehicle drives in wide open areas. In such cases, the classification algorithm relies on a single side background image and/or the drivable sub-image's Gaussian information.

3 EM (Expectation-Maximization) Algorithms

3.1 Introduction

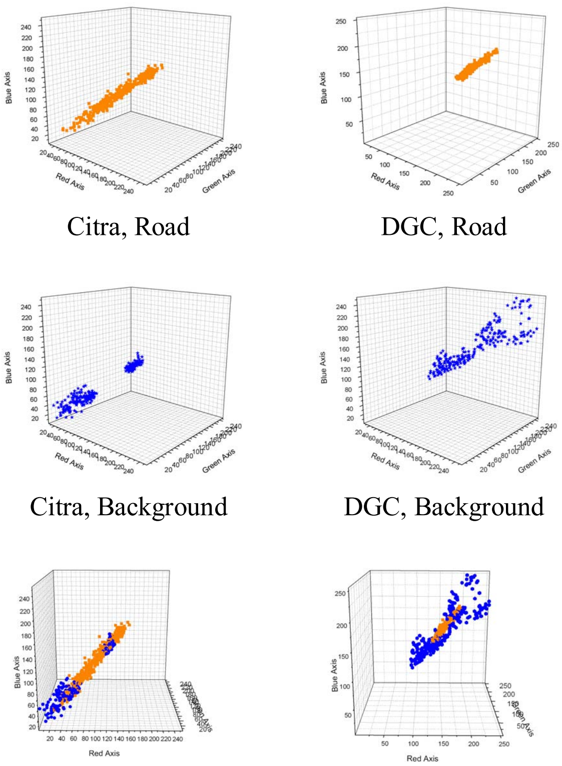
In this system, the EM (Expectation-Maximization) algorithm is implemented for road classification in the unstructured desert environment. The Bayesian based classification algorithm used for the DGC 2005 was limited in many situations because of its basic assumption that training areas have only one Gaussian distribution. In most cases, the properties of the road training area do not change rapidly and its distribution is Gaussian. However background sub-images do change at every scene and it is inappropriate to assume the data distribution is Gaussian. The RGB distribution of the unstructured desert scene shown in Fig 4 is portrayed as a three-dimensional distribution plot in Fig 5. Fig 4 (a) is a test scene at Citra, FL, (b) is a scene from the 2005 DGC course in Nevada. Like most of the outdoor image systems, the NaviGator vision system is susceptible to extreme changes in lighting conditions. Such changes can create shadows in captured images which can result in changes in the image color distribution. Such shadows can be seen in Fig 4(b).

(a) Citra, FL

(b) DGC 2005 course, Nevada

Fig. 4: Sample unstructured desert environment

From the 3D RGB plot in Fig 5, it is clear that most of the road training-area distribution is well clustered and can be evaluated with a single Gaussian distribution. However, for the background case there is no common distribution in the data. Thus, if a single Gaussian distribution is assumed for these areas, a large number of classification errors will be introduced.



Citra, Road — DGC, Road

Citra, Background — DGC, Background

Citra, Road & Background — DGC, Road & Background

Fig. 5: RGB distribution of road training area and background training area

Therefore, Gaussian based classifiers possess limited performance ability in real world scenes. This argument is further evidenced by the statistical model distribution for the background case in which the distribution is poorly defined. Therefore it is clear that a more sophisticated modeling approach is needed, namely a mixture-of-Gaussian model. In the mixture model, a single statistical model is composed of the weighted sum of multiple Gaussians. Therefore a mixture modeling classifier represents more complex decision boundaries between the road training sub-image and background training sub-images. However, computing a complex mixture model requires more processing time than computing a single Gaussian model. Therefore choosing the proper number of mixture models for a real-time application is critical. In this project, a single Gaussian model for the road training sub-image and two mixture Gaussian models for the background sub-image were selected empirically.

3.2 Application of EM Algorithm

The EM algorithm consists of three steps. The first step is deciding on the initial value of $\Theta_i = \{\mu_i, \Sigma_i, P(\omega_i)\}$ (the mean vector, covariance matrix and probability for i-th Gaussian distribution respectively). The second step is the Expectation step which calculates the expected value $E[y_{ij} \mid \Theta]$ for the hidden variable y_{ij}, given the current estimate of the parameter Θ. The third step is calculating a new maximum-likelihood estimate for the parameters Θ^k assuming that the value taken on by each hidden variable y_{ij} is its expected value $E[y_{ij} \mid \Theta]$. The process then continues to iterate on the second and third steps until the convergence condition is satisfied [5, 6].

$$\Theta^k = \arg\max_{\Theta} Q(\Theta, \Theta^{k-1}) \quad (1)$$

where

$$Q(\Theta, \Theta^{k-1}) = E[\log p(x, y \mid \Theta) \mid x, \Theta^{k-1}] \quad (2)$$

It is given that x_j is the known image pixel RGB vector and the labeling of the Gaussian distribution is given in the hidden variable y_i. By completing the datum set for z_j one can let

$$z_j = \{x_j, y_j\} \quad (3)$$

where $j \in \{1, 2, ..., n\}$, and n is the number of background data pixels.

In the mixture modeling, it has to compute the value of the hidden variable vector y_i, where $i \in \{1, 2\}$. The value y_i can be decided by two simple binary random variables

$y_{ij} = 1$ when x_j belongs to Gaussian ω_i

$y_{ij} = 0$ otherwise

so that

$$y_j = \{y_{1j}, y_{2j}\} \quad (4)$$

The vector y_j can only take on two sets of distinct values: {1, 0}, {0, 1}.

By applying the K-mean clustering method for the initial $\{\mu_i, \Sigma_i, P(\omega_i)\}$ for two Gaussian distributions [7] the algorithm clusters objects based on attributes into k partitions. Since it was decided to employ a two mixture Gaussian model for the two background sub-images, the clustering uses two means to compute the covariance and each Gaussian value's probability.

The principal difficulty in estimating the maximum-likelihood parameters of a mixture model is that it is hard to know the labeling y_i of each data pixel. From the initial value by the k-means clustering algorithm, one can compute $E[y_{ij} \mid \Theta]$, where value y_i is given by

$$y_i = \frac{P_i(x \mid \phi)}{\sum_{i=1}^{2} P_i(x \mid \phi_i)} \quad (5)$$

where

$$p(x \mid \phi_i) = p(x \mid \mu_i, \Sigma_i) = \frac{1}{(2\pi)^{3/2} |\Sigma i|^{1/2}} \exp\left[-\frac{1}{2}(x-\mu_i)^T \Sigma_i^{-1} (x-\mu_i) \right] \quad (6)$$

Next, one can compute the new $\{\mu_i, \Sigma_i, P(\omega_i)\}$ in terms of the complete data set $z_j = \{x_j, y_j\}$. Finally, a mixture-of-Gaussian model is computed from the new mean vector

$$\mu_i = \left(\sum_{j=1}^{n} y_{ij} x_j \right) / \left(\sum_{j=1}^{n} y_{ij} \right) \quad (7)$$

Similarly, a new covariance matrix is obtained from:

$$\Sigma_i = \left(\sum_{j=1}^{n} y_{ij} (x_j - \mu_i)(x_j - \mu_i)^T \right) / \left(\sum_{j=1}^{n} y_{ij} \right) \quad (8)$$

Likewise, the probability of a single Gaussian distribution is obtained from:

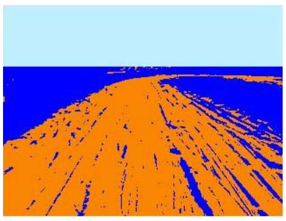
$$P(\omega_i) = \frac{n_i}{n} = \left(\sum_{j=1}^{n} y_{ij} \right) / n \quad (9)$$

Finally, the solution of mixture-of-Gaussian is found as:

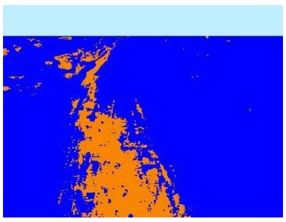
$$\begin{aligned} P(x \mid \Theta) &= \sum_{i=1}^{2} p(x \mid \phi_i) P(\omega_i) \\ &= p(x \mid \phi_1) P(\omega_1) + p(x \mid \phi_2) P(\omega_2) \end{aligned} \quad (10)$$

3.3 Simulation

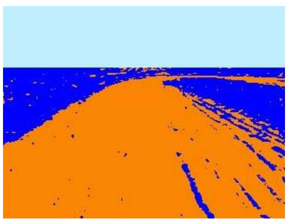
The EM algorithm was simulated on test images to gauge its performance in classifying images. For the purpose of the simulation, a single image was empirically chosen from the test site at Citra Florida as a representative "easy case" and another image from the 2005 DGC as a representative "hard case." The Bayesian based classification result and the EM based classification result are shown in Fig.6. In considering the DGC image (shown in Fig. 4(b)) it is clear that there is a considerable affect of shadow/illumination on the road surface leaving the left portion of the road many shades darker than the right. Since slightly more of the road is covered in shade, the resulting sample contained in the training segment is biased to the left half of the image as shown in Fig.6 (b).



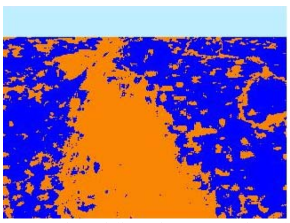
Bayesian classification result of Citra
(a)



Bayesian classification result of DGC 2005
(b)



EM classification result of Citra
(c)



EM classification result of DGC 2005
(d)

Fig 6: Classified road image

Fig. 7 shows the Citra and DGC 2005 scene's EM error where the x-axis shows the number of Gaussian distributions for the road training region and background training region. For the DGC 2005 scene with only one Gaussian distribution an error of 24.02% is obtained. However, if one Gaussian model for the road training region and two Gaussian models for the background training region are used an error of 6.12% is obtained. For the Citra case, the RGB distribution of the road training region and the background training region show that the two distributions do not overlap/intermix. As a result, a single Gaussian distribution for both the road and background training regions yields an error of 9.31%. Similarly, by applying one Gaussian for the road training region and a two Gaussians for the background training regions, the error is 6.42%. From these results, it is clear that the EM classification algorithm provides better classification performance. Furthermore, it is clear that the EM algorithm can dramatically reduce the classification error over the Bayesian, in particular in cases of images obscured by shadow, adverse lighting, or vibration induced motion blur.

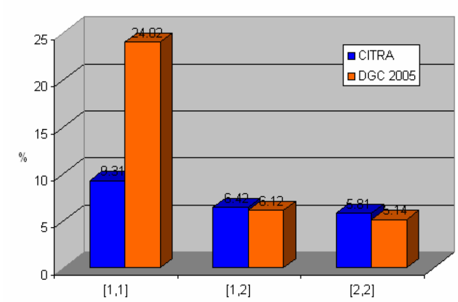


Fig. 7: Error of Citra and DGC 2005 Scene with varying number of mixture-of-Gaussian.

Since the EM algorithm relies on an iterative approach, setting the correct iteration condition is critical in reducing processing time. In this application, the mean RGB value is used to control the iterative process wherein the algorithm will continue to iterate until the difference between the previous mean RGB and the current mean RGB of the image is less than a pre-defined limit.

$$\left|\mu^k{}_i\right| - \left|\mu^{k-1}{}_i\right| < 0.1 \quad (11)$$

It should be noted that the variable k in Eq. (11) represents the current step of the iteration. Fig.8 shows the two Gaussian distribution's absolute mean value over the iteration step for the DGC 2005 image.

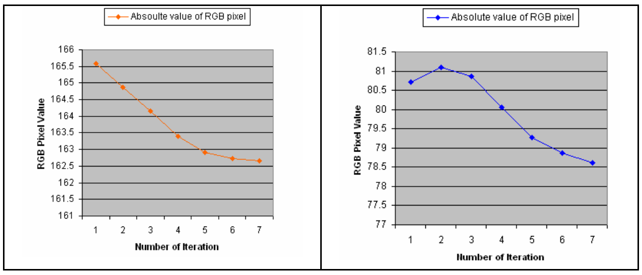
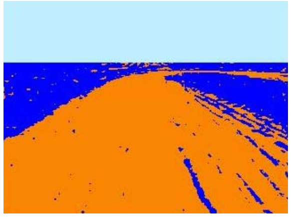


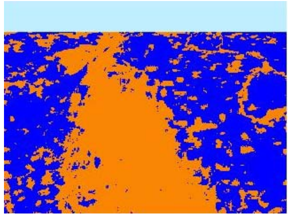
Fig. 8: Absolute value of RGB mean at first and second Gaussian distribution for the DGC 2005 image.

4. Transform to Global Coordinate Map

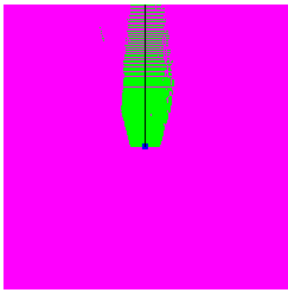
After classification of the image, the areas denoted as drivable road are converted by perspective transformation estimation into global coordinates used for the Traversability Grid [1]. The perspective transformation matrix is calculated based on camera calibration parameters and the instantaneous vehicle heading provided by position estimation components, i.e. GPS and INS. Finally, the PFSS assigns a value of 12 (highly traversable, green) to those cells that correspond to an area that has been classified as drivable [8]. All other cells are given a value of 7 (neutral, pink). Figure 9 shows the three steps of PFSS processing. Fig. 9(a, b) shows the classified image, Fig. 9(c, d) shows the transformed image without pixel interpolation, and Fig. 9(e, f) shows the transformed image with pixel interpolation. In each subfigure (Fig. 9c, d, e, f), the vehicle is located at the center of the image (blue square) with its direction indicated by a thin black line. Since a wide angled lens is used on the camera assembly, a broad swath of the road is captured. However, as a result of the wide angle of view, there is an appreciable distortion in distant regions of the image. This distortion results in only a small amount of pixels representing most of the far away portion of the image. This fact results in the transformation generating a mapped image with "holes" in the distant regions of the map. These holes can then be filled by linear interpolation with respect to the row number of each pixel (see Fig. 9(c, e)). Though this process can possibly introduce some error to the Traversability grid, the un-interpolated data, the horizontal lines or "holes", greatly hinder the Smart Arbiter and the Reactive Driver processing for planning the vehicle path [1]. The purpose of generating the Traversability grid is to generate a standardized sensor output which contains confidence information, location and orientation of obstacles, and can be readily re-oriented to account for changes is vehicle position and orientation. To this end, future work in the area of applying iterative statistical methods to image processing will continue with a focus on the fusion of multiple sensor outputs into a single grid. Additionally, further work is underway on the adaptive training for intensity and saturation adjustment for line finding in urban environments.



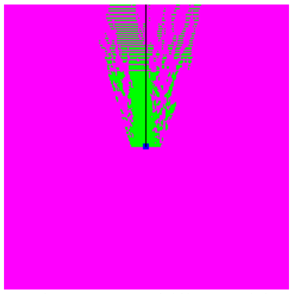
(a) Classified image of Citra



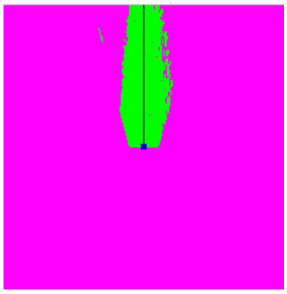
(b) Classified image of DGC 2005 course



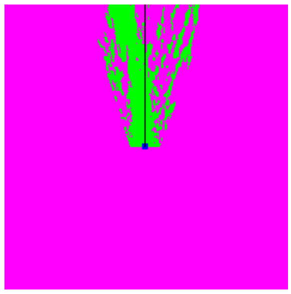
(c) Traversability Grid map image without interpolation of Citra



(d) Traversability Grid map image without interpolation of DGC course



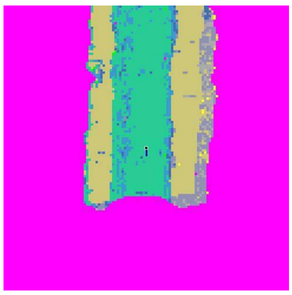
(e) Traversability Grid map image with interpolation



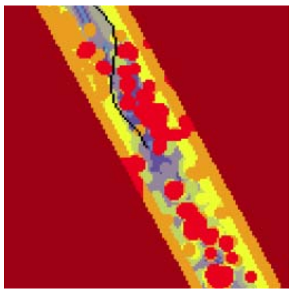
(f) Traversability Grid map image with interpolation

Fig. 9: Transformed image

With the Traversability Grid concept in place to normalize the outputs of a wide variety of sensors, the data fusion task becomes one of arbitrating the matching cells into a single output. This process is based on a method for combining the representative values for each corresponding cell in each traversability grid. Fig. 8(a) shows the LADAR sensor grids map and Fig. 8(b) shows the final path with combined sensor data.



(a) LADAR Sensor Grid Map



(b) Sample planning result through traversability grid

Fig. 10: Other sensor grid map and planning image with whole sensor information

5. SUMMARY

The pathfinder smart sensor (PFSS) was one of the principle components used by Team CIMAR during the 2005 DARPA Grand Challenge. Testing indicated that the EM algorithm based classification result provides a more accurate classification of roads and background than the Bayesian Algorithm.

REFERENCES

[1] Crane, C., Armstrong, D., Touchton, R., Galluzzo, T., Solanki, S., Lee, J., Kent, D., Ahmed, M., Montane, R., Ridgeway, S., Velat, S., Garcia, G., Griffis, M., Gray, S., Washburn, J., "Team CIMAR's NaviGATOR: An Unmanned Ground Vehicle for Application to the 2005 DARPA Grand Challenge," *Journal of Field Robotics,* 2006

[2] Carl D. Crane III, David G. Armstrong, Mel W. Torrie and Sarah A. Gray, "Autonomous Ground Vehicle Technologies Applied to the DARPA Grand Challenge, *ICCAS2004*, 2004.

[3] Jaesang Lee, Carl D. Crane III, Sanggyum Kim and Jungha Kim , "Road Following in an Unstructured Desert Environment using Monocular Color Vision as Applied to the DARPA Grand Challenge", *ICCAS2005*, 2005.

[4] JAUS. (2005). Joint Architecture for Unmanned Systems Reference Architecture, version 3.2: JAUS Working Group (http://www.jauswg.org/).

[5] Moon, T.K., "The expectation-maximization algorithm", *Signal Processing Magazine, IEEE Volume 13, Issue 6,* Nov. 1996 Page(s):47 – 60

[6] "A Gentle Tutorial of the EM Algorithm and its Application to Parameter Estimation for Gaussian Mixture and Hidden Markov Models"

[7] Richard O.Duda, Peter E.Hart, David G.Stork, "Pattern Classification" second edition, Wiley-Interscience

[8] A. Criminisi, I. Reid, A. Zisserman, "A Plane Measuring Device", http: //www.robots.ox.ac.uk/~vgg/presentations/bmvc97/criminispaper/