Color road segmentation and video obstacle detection

Matthew A. Turk and Martin Marra

Martin Marietta Denver Aerospace
P.O. Box 179, M.S. H0427, Denver, Colorado 80201

Abstract

The primary vision task in road-following for a mobile robot is to provide a description of the road environment, including possible obstacles on the road. Techniques are presented for road segmentation and obstacle detection based on color video data. Using constraints on road characteristics in the image space and in 3D color space, the road is extracted and represented by its edges. Assuming vehicle movement, obstacles are detected at a distance and an obstacle avoidance mode is entered.

Introduction

To perform in a real outdoor road environment, a mobile robot requires the ability to sense and perceive relevant environmental features. For the particular task of road-following, the vision system must model the road in the face of changing weather and lighting conditions, dirt on the road, ill-defined road edges, etc. The system must also be able to quickly detect and model obstacles in the road so that a navigation system can adequately slow down and steer around them.

Road-following and obstacle avoidance for a mobile robot have been demonstrated under the Autonomous Land Vehicle (ALV) program at Martin Marietta Denver Aerospace, beginning with a public road-following demonstration in May of 1985. The initial demonstration and ALV system are described by Lowrie, et al. [1]. In subsequent demonstrations, the ALV (affectionately referred to as "Alvin") has traveled faster and farther, demonstrating more robust vision and navigation systems. A program update is given in [2]. Recently, Alvin has reliably performed obstacle avoidance at slow speeds using both video and range imagery as input. Dunlay [3][4] describes road-following and obstacle detection and avoidance using a laser range scanner. A detailed description of Alvin's vision system, concentrating on video road-following, is presented in [5].

A laser range scanner is a very useful device for describing shape and therefore good for modeling obstacles in the road. However range scanners presently have at least two drawbacks when compared to passive, video sensing. Range scanning based on a dual mirror configuration is inherently slow, providing images about an order of magnitude slower than video cameras. This may cause image smearing due to sensor movement, as well as reducing available processing time. Also, detecting range via phase shift measurement causes ambiguity intervals in the image, since phase difference can only be measured modulus one complete cycle. Ambiguity intervals can be disambiguated to a fair degree of accuracy, but this again adds time to the image acquisition stage. These drawbacks are presently limiting the use of the laser scanner in obstacle avoidance to relatively low speeds, because of the latency in image acquisition and the limited sensing distance and ambiguity.

We therefore would like to detect the presence of obstacles using a video sensor, which can see far down the road and acquire images quickly. This will allow us to travel at a relatively fast speed (e.g. 20 km/hr) keeping an eye out for the presence of obstacles, and then to slow down when an obstacle is detected and use laser range data to model the obstacles for navigation.

This paper describes techniques used for color road segmentation, the first step in video road-following, and the methods currently being investigated and tested for video obstacle detection.

This work was performed under the Autonomous Land Vehicle Program sponsored by the Defense Advanced Research Projects Agency (DARPA) as part of its Strategic Computing Program and contracted through the Army Engineer Topographic Laboratories (Contract # DACA76-84-C-0005).

Road segmentation

Alvin currently has a single color RGB camera mounted on a pan/tilt mechanism. Vehicle position and orientation are available to the vision system indirectly from a land navigation system (LNS). The vision system resides on a Vicom image processing computer, a 68000-based host with a high-speed image bus and frame-rate convolution and point operation boards. The initial road-following algorithm developed for autonomous navigation was motivated largely by the available hardware.

Each pixel in an RGB image describes a point (or a vector) in the three dimensional image space. A gray-scale image may be obtained by projecting the RGB point onto the line through the origin and perpendicular to the plane R + G + B = 0. In general, a single-band image may be obtained by projecting RGB points onto the line through the origin and perpendicular to the plane rR + gG + bB = 0 - this is equivalent to a tricolor operation, a linear combination of the red, green, and blue images. A tricolor operation also describes a dot product between pixel values (R,G,B) and the plane unit normal (r,g,b). Figure 1 shows this relationship.

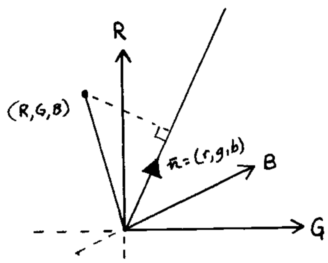

Fig 1. Tricolor operation in RGB space.

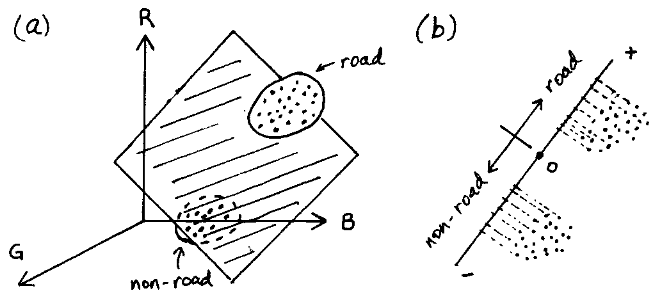

Fig 2. (a) RGB clusters (b) Projection onto line

If a plane in RGB space separates two distinct clusters of pixel values (e.g. corresponding to road and non-road regions of the image), it is simple to produce a binary image describing the two regions by performing a tricolor operation and thresholding the result. Figure 2(a) shows a plane separating two RGB clusters. As in Figure 2(b), each point projects onto a line perpendicular to the separating plane; the intersection of the plane with the line defines a threshold which separates the two regions. A tricolor operation with weights (r,g,b) equal to the normal of this discriminating plane performs this projection, and a threshold operation distinguishes cluster #1 from cluster #2. This describes a simple clustering or classification segmentation [6][7] using a linear discriminant function. In this example, the tricolor weights or parameters (r,g,b) define the orientation of the plane, while the threshold determines the plane's translation.

2D scattergrams and binary roads

The preceding discussion may be applied to a two dimensional feature space as well. In this case, the discriminant is a line, rather than a plane. The parameters of the tricolor (or in this case "bicolor") operation describe the orientation or slope of the line, and the threshold determines the perpendicular translation of the line (or the intercept). Figure 3(b) shows a Red/Blue scattergram, a two dimensional histogram, from the original RGB image of Figure 3(a). The line separating the elliptical clusters in the Red/Blue space defines a road/non-road boundary as seen in the binary image of Figure 3(d). Figure 3(c) outlines the various scene components in the scattergram.

We have noticed that the green band gives very little information helpful in direct road/non-road segmentation, at least in the road areas and conditions which we have experienced. Because of this we are able to reduce the three dimensional segmentation problem, to a large degree, to a two dimensional segmentation. The techniques are simply generalized to 3D, but some aspects are much less computationally expensive in Red/Blue space rather than RGB space.

Assuming there is a reasonable road/non-road separation in the Red/Blue space, the question remains of how to choose appropriate tricolor parameters and the threshold that produces the binary road image, both initially and dynamically as the scene changes. For a mobile robot, images may vary greatly with time as a result of sensor motion, changing weather and illumination conditions, unexpected road conditions such as shadows or dirt on the road, etc.

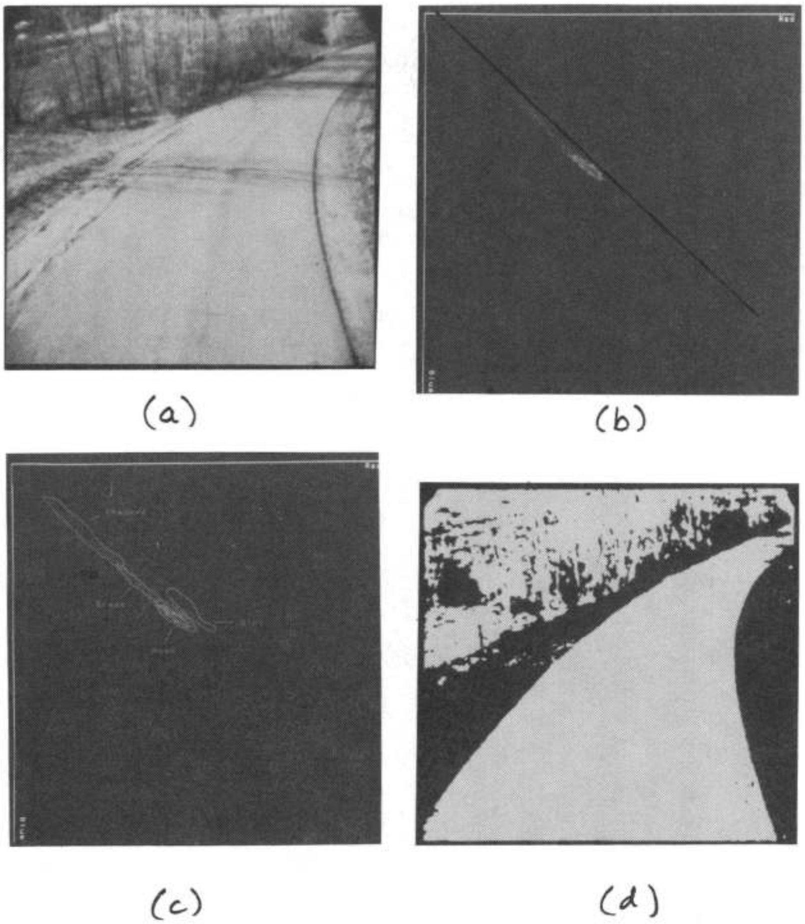

Fig 3. (a) Road scene. (b) Corresponding Red/Blue scattergram. (c) Outlines of individual scene components. (d) Binary "road" image.

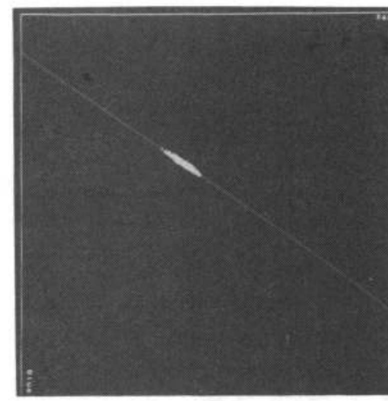

Fig 4. Road cluster and calculated orientation.

Dynamic parameter selection

The tricolor parameters may be chosen dynamically by noting that the orientation of the Red/Blue line is equal to that of the road "ellipse" or cluster. If we can sample road pixels exclusively, we can calculate the orientation θ of the road cluster. The angle the principal axis of the cluster makes with respect to the Red axis is defined by the equation [8]

$$\theta = 0.5 \tan^{-1} \left(\frac{b}{a-c}\right) \qquad (1)$$

where

$$a = \sum_i (r-\bar{r})^2$$

$$b = 2 \sum_i (r-\bar{r})(b-\bar{b}) \qquad (2)$$

$$c = \sum_i (b-\bar{b})^2$$

and ($\underline{r}$, $\underline{b}$) is the mean of the cluster. From θ the red and blue color parameters are calculated as (r, b) = (cos θ, sin θ). Figure 4 shows the line calculated from the road cluster.

Performing a tricolor operation on the RGB image produces a gray-scale image where each pixel represents the perpendicular distance from the scattergram point to the line with orientation θ that passes through the origin. This line is effectively translated by adding a constant to the gray-scale image - however, thresholding the image accomplishes both this translation and the resulting road/non-road segmentation in one step. In the general case, performing a tricolor with parameters (r,g,b) and a threshold operation with a threshold of λ is described by the equation

$$I(i,j) = \begin{cases} 1 & \text{if } rR(i,j) + gG(i,j) + bB(i,j) + \lambda < 0 \\ 0 & \text{otherwise} \end{cases} \tag{3}$$

In selecting the threshold λ we want to translate the separating line in Red/Blue space to the border between the road and non-road clusters. The original method involved calculating the standard deviation of the road cluster and moving some multiple of σ away from the mean. At present we look for the points in the road cluster most in the direction of the road/non-road boundary, then move a small constant distance further. This is more reliable in the presence of "false" road sample points, e.g. dirt on the road.

Video obstacle detection

As pointed out in Section 1, the current range scanner is not sufficient for obstacle modeling (and therefore obstacle avoidance) at fast vehicle speeds, because of the image acquisition time and the limited range of view. Because of the increased distance needed to decelerate to a stop in the worst case (if the obstacle is a brick wall!), doubling vehicle speed requires a more than doubling of processing speed. An alternative method of detecting obstacles must be used while traveling at fast speeds, so the speed can be decreased immediately to allow enough time for range-based obstacle avoidance. Rather than using a specialized radar, sonar, or laser device for the purpose of detecting obstacles in the road in front of the mobile robot, we would like to use the more general video sensor. The video camera provides more than adequate image acquisition rates, depth of view, and resolution at a distance.

Rather than attempting to completely model the shape of objects in the road scene, we can use assumptions about vehicle movement, obstacles, and the road to form constraints to quickly detect obstacles in the road image. Possible assumptions include:

Lateral movement is small at high speeds compared with forward movement.

Obstacles look different from the road.

The road surface is fairly consistent in color and intensity.

At a distance, road "texture" is minimal (particularly after blurring).

Obstacles produce significant edges when a simple edge operator is applied.

Obstacles are segmented distinct from the road.

Using the first four constraints, we have developed a simple video obstacle detection algorithm. The algorithm is an addition to the existing video road-following algorithm [5], since it uses partial results from the road segmentation. The algorithm is described by the following steps:

(1) Digitize two successive images, so that any obstacles in the road will have "moved" a small number of rows down the image.

(2) Perform a tricolor operation on the images, creating a weighted combination of red, green, and blue components in one image.

(3) Subtract the images and take the absolute value of the resulting image.

(4) Search in a region of interest, defined by a trapezoid representing the upper section of the segmented road, for values above a preset threshold. If there are enough pixels above the threshold, signal that an obstacle is present.

Figure 5 shows each step of the algorithm.

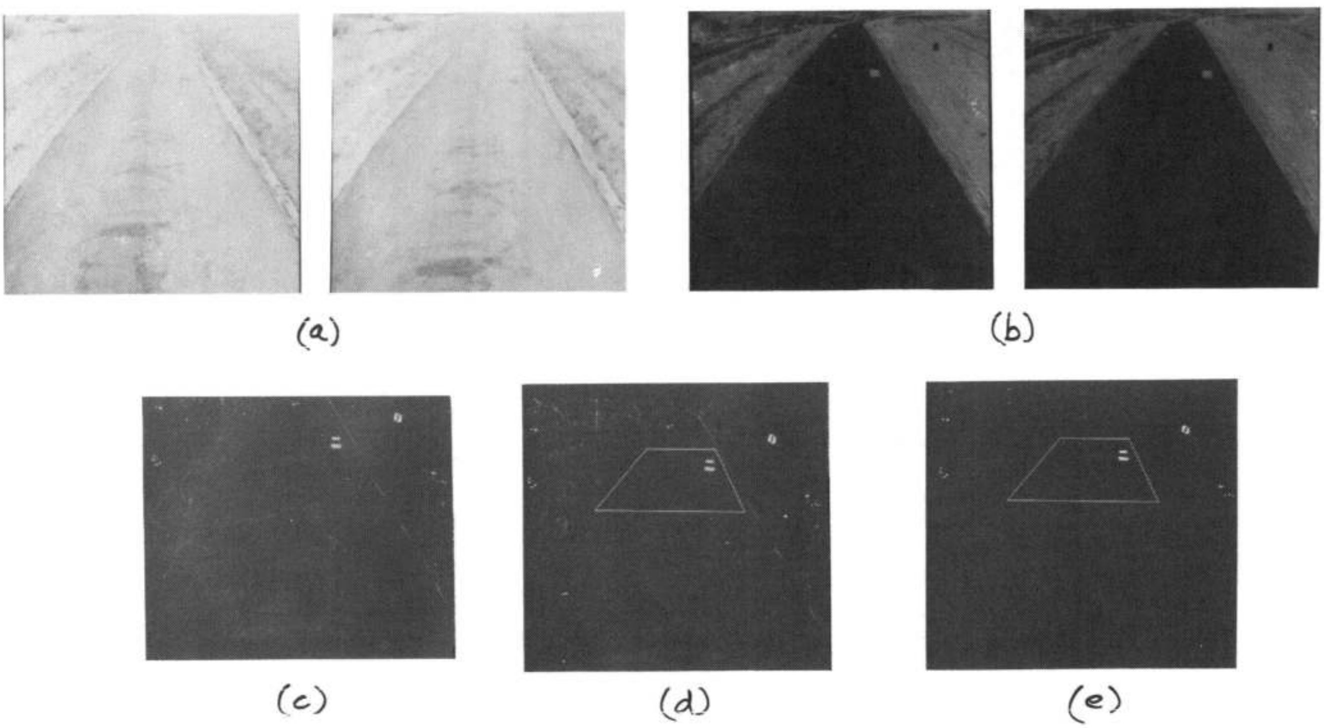

Figure 5. (a) Images taken about 2 meters apart. (b) Tricolor results from images in (a). (c) The result of subtracting the two and taking the magnitude. (d) The region of interest superposed on (c). (e) Pixels in the ROI labeled as an obstacle.

In the first step, the critical parameter is the time between image acquisitions. Taken too close in time, the images do not differ enough to note apparent obstacle movement. Waiting too long between image acquisition may allow excessive vehicle movement in the lateral direction, which violates a basic assumption of the algorithm and would allow non-road pixels to fall inside the region of interest. The best solution is to space the image acquisition according to vehicle speed so that any obstacles would be expected to move 4-10 rows between images. Using our default camera model and vehicle parameters, at a speed of 20 km/hr this translates to approximately a 350 msec separation between images. Figure 5(a) shows such an image pair.

A tricolor operation produces a weighted linear combination of the red, green, and blue components of an RGB image. Using the assumption that obstacles look different from the road in a color image, we want to perform a tricolor operation that enhances this difference. Conceptually, the tricolor operation projects points in RGB space onto a line whose direction is defined by the red, green, and blue parameters (weights) of the tricolor. The maximum separation in the resulting image occurs when these parameters describe the direction between "road" and "obstacle" clusters in the RGB space.

Assuming a small lateral motion in between successive image acquisitions, as well as a reasonably constant road curvature, subtracting the enhanced images gives significant non-zero values where there has been movement in the scene between the two images, as seen in Figure 5(c). Since the road is basically a homogeneous region in the image after blurring, especially at a distance where the look angle is small, any significant non-zero values in the road are most likely to be caused by obstacles. Figure 5(d) shows the region of interest within the road boundary in which the presence of an obstacle may be detected. This region of interest is bounded on the top and sides by the binary road segmentation and on the bottom by the range "down the road" where obstacles must be detected to avoid a collision. Since only the initial detection is important to the algorithm, the road region near the vehicle does not need to be searched.

At present, an object is detected if there are at least N pixels within the region of interest of the subtracted magnitude image above some threshold value T. This simple scheme may also detect "objects" in the road that are not obstacles, e.g. dirt patches and tarmac for road repair. More sophisticated detection methods are being considered, taking into account shape as well as number of points.

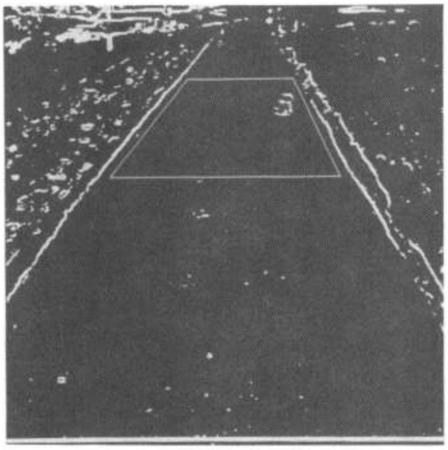

Figure 6. Edge image with region of interest outlined.

Other images may serve as input to the detection phase as well or better than a subtraction of successive images. Edge images produce good highlights of obstacles within the region of interest. Figure 6 shows an edge image with the region of interest outlined. A Sobel operator returns very light edges from shadows and tarmac patches and strong edges on most non-road colored obstacles when applied to the tricolored feature image. Advantages to using an edge image are that it requires no restriction on vehicle motion and it is faster to produce than the subtracted image; however, there are fewer pixels describing the object. Both methods are currently being implemented and have achieved promising results.

Discussion

Road segmentation is a first step in modeling the road for navigation. After finding the road outline in the image, the edges are sampled down into a small number of edge points which are then converted to three dimensional road edge points and sent to the navigator. Various techniques have been proposed for the 2D to 3D conversion, making use of assumptions of road geometry [5]. When an obstacle is detected, obstacles are also outlined, converted to a 3D description, and sent along with the road description to the navigator.

The video road segmentation algorithms have been largely motivated by the available hardware and real-time speed considerations. Alvin has traveled at speeds up to 20 km/hr with the current hardware and software configurations. There are still situations which cause trouble for the algorithms, however: dirt on the road, significant shadows, a partially wet road, very low sun angles, and spectral reflection, to name a few. Such demands for robustness have guided the development of the road-following system, as much is learned through test failures.

Video obstacle detection is currently being tested and developed. Initial testing has proven successful, and it looks more attractive than incorporating a special (perhaps radar) sensor for detecting obstacles. Autonomous mobility through passive, rather than active, sensing is a long-term goal of the ALV project.

Summary

We have presented techniques used for road segmentation and video obstacle avoidance for Alvin, the Autonomous Land Vehicle. Road segmentation uses knowledge of the relationship in color space between road and non-road scene elements. Obstacle detection uses constraints on motion, obstacle appearance, and road homogeneity in the image to detect the presence of an obstacle so that Alvin can slow down and enter an obstacle avoidance mode.

References

[1] J. M. Lowrie, M. Thomas, K. Gremban, and M. Turk, "The autonomous land vehicle (ALV) preliminary road-following demonstration," Intelligent Robots and Computer Vision, David P. Casasent, Ed., Proc. SPIE 579, pp. 336-350, Sept. 1985.

[2] J. M. Lowrie, "Autonomous land vehicle program update: 1987 update," Proc. SPIE Mobile Robot Conference, Cambridge, MA, October 1986.

[3] R. T. Dunlay, "Robot road following using range imagery," Proc. SME Second World Conference on Robotics Research, August 1986.

[4] R. T. Dunlay, "Obstacle avoidance on roadways using range data," Proc. SPIE Mobile Robot Conference, Cambridge, MA, October 1986.

[5] M. Turk, D. Morgenthaler, K. Gremban, and M. Marra, "Video road-following for the autonomous land vehicle," Submitted to the 1987 IEEE International Conference on Robotics and Automation.

[6] G. B. Coleman and H. C. Andrews, "Image segmentation by clustering," Proc. IEEE, Vol. 67, Number 5, pp. 872-884, May 1979.

[7] R. O. Duda and P. E. Hart, Pattern Classification and Scene Analysis, New York, Wiley-Interscience, 1973.

[8] B. K. P. Horn, Robot Vision, MIT Press, Cambridge, MA, 1986.

[9] R. Gershon, A. D. Jepson, and J. K. Tsotsos, "The effects of ambient illumination on the structure of shadows in chromatic images," Technical Report RBCV-TR-86-9, Dept. of Computer Science, Univ. of Toronto, 1986.