A Robust Boundary Line Extraction Algorithm

Donald MacArthur\textsuperscript{1}
Erica Zawodny\textsuperscript{1}
Michael Nechyba\textsuperscript{2}
Carl D. Crane III\textsuperscript{1}

\textsuperscript{1}Center for Intelligent Machines and Robotics
Department of Mechanical and Aerospace Engineering
P.O. Box 116300

\textsuperscript{2}Machine Intelligence Laboratory
Department of Electrical and Computer Engineering
P.O. Box 116200

University of Florida
Gainesville, Florida 32611

email: ccrane@ufl.edu

Abstract

This paper describes the development of a new line extraction algorithm that is to be used on an autonomous vehicle. The objective of the algorithm is to locate lines painted on the travel surface. The autonomously navigating vehicle will use this information to navigate in its environment. The algorithm will be described and results presented.

Introduction

Computer vision is an area of growing interest in the field of autonomous ground vehicles. Much research has been performed in color and pattern recognition for autonomous navigation. This paper investigates the problem of lane detection/following for a fully autonomous ground vehicle. This paper will discuss the algorithms that were implemented on real world data sets and provide corresponding analyses.

One place where the algorithm will be applied is at the annual Intelligent Ground Vehicle Competition that is sponsored by the Association for Unmanned Vehicle Systems International (AUVSI). Each team designs, builds, and programs a ground vehicle to traverse several obstacle courses. One of the obstacle courses involves lane following, where the vehicle must travel through the obstacle course bounded by painted lines. The images in Figure 1 show a sample of the painted lines and the obstacles encountered.

As automobiles become more sophisticated, autonomous driving is becoming more of a possibility. This technology can be applied to commercial, military, and consumer markets.

\textbf{Figure 1:} Sample Obstacle Course Images
Effective detection of road boundaries is essential to any type of autonomous driving controls. Obstacle detection and collision avoidance can be combined with boundary detection algorithms to provide a complete control solution for an autonomous system. Figure 2 depicts typical highway markings.

The field of autonomous robotics is rapidly expanding and much research is being performed in robot cooperation, sensing capabilities, and artificial intelligence. Computer vision has been applied to assembling of parts in a factory, a variety of sports applications, feature/object tracking, and road following and horizon tracking [1] for different types of autonomous robots. The particular application of interest for this project is road/lane following using computer vision.

Previous work has been performed for the ground vehicle competition mentioned earlier. In particular, a team from Virginia Tech used an intensity-based line finding approach to determine where the painted lines occur in the image. The assumption is made that the boundary lines will be the most intense features in the image and that the line occupies only a small portion of the image. The algorithm isolates pixels that have intensity greater than the threshold value. In addition, they have developed a dynamic method of selecting the threshold based on an image histogram. The highest peak (i.e. greatest intensity) in the histogram should represent lines in the image. [2]

Machine vision has been applied to the task of automated vehicle steering. Carnegie Mellon University has developed a vision system that helps drivers steer by determining the road curvature and then calculates the lateral offset of the vehicle relative to the center of the lane. This method does not rely on particular features in an image; instead a sampled image is analyzed and several hypotheses are applied. The algorithm can hypothesize a possible road curvature and then transform the image based on that hypothesis. The algorithm is seeking the “straightest” feature in the image and this is quantified by a scan-line intensity profile. In short, visible image features are indicated by sharp discontinuities in the scan-line intensity profile, which indicates a sudden change in intensity in the image. [3]

Each of these methods was considered when formulating this particular project. For this effort, it was desired to try to extract features from the image. In this case the lines are the patterns we are trying to recognize and classify.

Data Collection / Initial Data Analysis

An obstacle course was created similar to that expected at the intelligent ground vehicle competition. Data was collected by remotely driving a robot through the obstacle course while simultaneously taking pictures using a USB Internet camera (see Figure 3). Several data sets were obtained which include a series of approximately 245 pictures taken at a resolution of 320×240. All of the images were of the portable pixmap (PPM) form because this format is easier to manipulate in software and contains very little header information.

Additional data was collected on an asphalt road with painted white boundary lines. The data set consisted of approximately 147 sample images at a resolution of 320×240. The image format was also PPM format.

Upon analyzing the data it was found that there was a considerable amount of high frequency noise present. This could disturb the clustering of similar regions. Therefore, each image was pre-processed using an averaging filter to establish more continuity between regions of similar color properties.

As stated above the images being processed are of 320×240 resolution and processing each image can require processing of each pixel. This becomes computationally intensive and expensive in terms of

![Figure 2: Typical Highway Markings](image)

![Figure 3: Robot Platform and USB Camera](image)
processing power. It would be beneficial if the dimensionality of the input data could be reduced. Principal component analysis serves to reduce the dimensionality of the data without any loss of the desired features in the image.

A raw camera image was analyzed using a Principal Component Analysis (PCA) algorithm to determine the mean and covariance of each image. Every pixel in the image was projected along each of the eigenvectors of the covariance matrix. Plots of the data revealed that the 2nd principal component distinguished the line features from the environment in the image (see Figure 4). While the 1st principal component extracts the line features there is still a considerable amount of high frequency noise occurring in the image. The 3rd principal component did not provide any useful information.

Vector Quantization (VQ) software that was provided by M. Nechyba of the University of Florida was then used to segment the image into regions of similar color and spatial properties. For this algorithm the objective was to find boundary lines. This assumes that the lines are composed of a continuous color and geometry.

The vector quantization algorithm uses data that was pre-processed by the PCA algorithm. For each image a codebook was created using the PCA and the position data for each pixel. A total of $2^7$ vectors were created using the VQ algorithm, which allowed for each pixel to be classified as one of 128 classes. This information was used to group the pixels for further processing. Figure 5 shows a sample of the VQ classification of the image pixels.

**Algorithm Development**

**A. Maximum Likelihood (ML) Estimation**

The parameters for the maximum likelihood classifier were determined by segmenting several training images using the provided segmentation program. The experimenter extracted the white lines visually using the segmentation program [1]. The
mean and covariance for the Gaussian model were determined using the extracted data. The validation images were processed on a pixel-by-pixel basis by calculating the probability of the pixel color information given the model parameters.

The mean ($\mu$) and covariance ($\Sigma$) were both determined for each color channel: R, G, and B using equations (1) and (2) shown below. The probability of whether a specific pixel was part of the line was determined using equation (3). This algorithm was implemented in C as well as in MATLAB.

$$\mu = \frac{1}{n} \sum_{j=1}^{n} x_j \quad (1)$$

$$\Sigma = \frac{1}{n} \sum_{j=1}^{n} (x_j - \mu)(x_j - \mu)^T \quad (2)$$

$$p(x) = \frac{1}{(2\pi)^{d/2}|\Sigma|^{1/2}} \exp \left[ -\frac{1}{2} (x - \mu)^T \Sigma^{-1} (x - \mu) \right] \quad (3)$$

Preliminary results using ML are shown in Figure 7.

B. Artificial Neural Network (ANN)

A classifier was developed using a single hidden layer multilayer perceptron network. The architecture that was used included three inputs including a bias for the input layer, ten hidden layer perceptrons, and one output layer perceptron. The non-linear function that was used was an exponential sigmoid which was used for every perceptron. The ANN was created and trained using an original program written in C by the authors.

The training data was obtained using a method similar to that used for the ML algorithm. For the Neural Network, data was collected for the white lines and for the background/non-white line pixels using the segmentation code provided [1]. Three sample images were used for training the network. The network was trained using gradient descent and backpropagation. Shown in Figure 8 is the corresponding ANN classified image where the sample image shown in Figure 7 was used.

C. Spatial Statistics Classifier (SS)

This algorithm was developed so that classification could be performed without the need for a color model. The idea was to develop a classifier based solely on spatial properties of regions of similar color and proximity. In an attempt to reduce the order of the input space, PCA was performed to analyze the underlying properties of the image.

After analyzing the PCA data it was found that the main principal component for the image lied close to the gray axis. The data from the 2nd principal component showed very distinct differences between background pixels and line pixels. This projection also was not as noisy as that of the 1st component. The 2nd principal component was chosen as the sole color input for the vector quantization portion of the algorithm.
The vector quantization was performed in order to separate the input space into regions with similar color and spatial properties. The input vector for the VQ algorithm was composed of a scaled projection of the pixel color vector with the 2nd principal component, and the pixel location. This allowed for the VQ algorithm to be performed using only a 3D input vector instead of a 5D vector should all of the color information be used.

A classifier was then developed to utilize the geometry of the regions. The ratio of the eigenvalues calculated from the x and y spread over each region provided a basis for line classification. Due to the inherent nature of the white lines, the regions containing the white lines had a high eigenvalue ratio. By establishing a simple threshold for the eigenvalue ratio, the classification was performed on each region in the image. It should be noted that this algorithm required very little to no training except for the selection of the eigenvalue ratio threshold. The algorithm is depicted in Figure 9. Figure 10 shows the respective PCA/VQ/Spatial Statistics Classified image for the same sample image shown in Figure 7.

**Initial Investigation Discussion**

The ML, ANN, and SS methods provided promising results. The results for the ML algorithm showed that classification of the boundary lines performed well with a small amount of misclassification. The probability threshold was arbitrarily set using the classification performance of the initial training images. The ANN algorithm provided similar results compared with the ML algorithm. The ANN results showed a significantly greater amount of noise compared with the ML results. Misclassification can be attributed to lack of training data or premature network convergence while learning. The algorithm performed classification of the white lines well but had a considerable amount of false detects for the background. Due to the static nature of each algorithm it was observed that classification performance suffered under changing environmental conditions.

The SS algorithm differed from the ML and ANN algorithms by utilizing information other than color for classification and did not require a rigid color model. This algorithm uses the shape of the distribution of regions composed of pixels of similar color. The free parameter for this algorithm consisted of the eigenvalue ratio threshold. This threshold was set using the classification performance of the initial training images used for the ML and ANN algorithms.

**Formulation of Final Algorithms**

After observing the performance of the individual algorithms it would be beneficial to combine the properties of each algorithm to improve overall classification performance. By using a “mixture of experts” technique the algorithms can be combined to produce an algorithm that outperforms each contributing algorithm. Two “mixture of experts” algorithms were developed to evaluate the performance of this technique. These algorithms were developed to provide classification using an unsupervised learning technique (SS) combined with a priori color information (ML/ANN).

A. SS/ML

This algorithm combines the ML algorithm with the spatial statistics classifier (SS). The algorithm first evaluates the image using the SS algorithm and associated pre-processing techniques (PCA, VQ). The algorithm then calculates the average color
values for each classified region and calculates the associated probability using the ML algorithm. This allows for the classified regions derived from the SS algorithm to be further classified using the ML algorithm. Results are shown below in Figure 11 where the same sample image in Figure 7 was used. While the ML algorithm extracts the lines well there was a fair amount of noise and false detects in the classified image. Introduction of the SS classifier helps to reduce noise and false detects while maintaining high levels of line extraction.

B. SS/ANN

This algorithm combines the ANN algorithm with the spatial statistics classifier (SS). The algorithm first evaluates the image using the SS algorithm and associated pre-processing techniques (PCA, VQ). The algorithm then calculates the average color values for each classified region and passes the information to the ANN algorithm. This allows for the classified regions derived from the SS algorithm to be further classified using the ANN algorithm. Results are shown in figure 12. The performance of the SS/ANN algorithm was slightly less than that of the SS/ML algorithm. This can be attributed to the discrete nature of the neural network architecture and the fact that the mean values for the regions were used versus the RGB values of each pixel.

C. Evaluation of Real World Data

Due to the difficulty in quantifying the actual performance of the SS/ML and SS/ANN algorithms, two movies were created, which allows for visualization of the performance of the algorithm. The movies consist of a sequence of the raw images, plots of the RGB values of each pixel, classified SS/ML images, and classified SS/ANN images. These algorithms were applied to the Intelligent Ground Vehicle Competition data set and the paved road data set. Each data set was processed and independent color models and ANNs were created using the first few images in the sequences. Figures 13 and 14 show the first image of both movies.

Conclusion

This paper has presented several algorithms that were developed to create a robust boundary line extraction algorithm. The paper described the process by which these algorithms were implemented and evaluated. The final algorithms combined the elements of various classifiers, which utilized color and spatial properties of the image. It was found that by combining the various classifiers, the performance was improved beyond that of the performance of each individual classifier.

The algorithms that were developed greatly improved the existing methods by which boundary lines were extracted. Future work can include incorporation of adaptation of the color models, modifications to produce a real time algorithm, and multiple color model incorporation to detect and classify not only white lines but lines of different colors.

Acknowledgements

The authors would like to gratefully acknowledge the support of the Air Force Research Laboratory at Tyndall Air Force Base, Florida.
References


Figure 13: Evaluation of IGVC Data Set

Figure 14: Evaluation of Road Data Set