Automatic Roadway Geometry Measurement Algorithm Using Video Images

Yichang (James) Tsai¹, Jianping Wu¹, Yiching Wu¹, and Zhaohua Wang²

¹ Geographic Information System Center, Georgia Institute of Technology,
276 fifth street, Atlanta, GA 30332, United States
{James.tsai@ce, jw248@mail, yw23@mail}.gatech.edu

² School of Civil and Environmental Engineering, Georgia Institute of Technology,
790 Atlantic Drive, Georgia Tech, Atlanta, GA 30332, United States
zw12@ce.gatech.edu

Abstract. The Georgia Department of Transportation (GDOT) collects and maintains an inventory of all public roads within the state. The inventory covers more than 118,000 centerline miles (188,800 km) of roads in 159 counties and over 512 municipalities. The transportation road inventory includes more than 52 items, including roadway geometry, surface type, shoulder type, speed limit signs, etc. Traditional roadway geometric properties, including number of lanes, travel lane, and shoulder widths, are measured in the field. Roadway geometric property measurement is one of the most important and, yet, the most time-consuming and riskiest component of the roadway data inventory. For the past two years, GDOT has sponsored Georgia Tech to develop a GPS/GIS-based road inventory system that re-engineers the existing paper-pencil operations. Georgia Tech has extended the research to develop video image pattern recognition algorithms and a prototype application aimed at automating the roadway geometry measurement to enhance the roadway inventory operations. A highly reliable and effective image extraction algorithm using local thresholding, predictive edge extraction, and geometric optics was developed and is presented in this paper. Preliminary results show it can effectively extract roadway features. A large-scale, experimental study on accuracy and the productivity improvement is under way.

1 Introduction

Collecting roadway infrastructure data, including pavement geometric properties (number of lanes, travel lane and shoulder widths), signs (stop signs, speed limit signs, etc.) at each location (milepost and x, y coordinates) is crucial for state Departments of Transportation (DOTs) in the US for submitting Highway Performance Monitoring System (HPMS) data annually. It is also important for transportation system planning, design, maintenance, and operations, including safety studies and traffic modeling. However, collecting road inventory data is a costly, time-consuming, and sometimes dangerous. The data quality and data update frequency cannot meet the growing transportation needs. Currently, many state DOTs have used video logging as a means for collecting roadway infrastructure data. The roadway video logging images are collected and then displayed on a computer screen, frame by frame, and the various

roadway features, such as number of lanes, pavement width, and type and location of signs, are manually measured/extracted. Although the field data collection method, in terms of data quality and safety, has been improved using this approach, the process of extracting roadway information from the images remains time-consuming and tedious. The current practice of manually viewing and extracting roadway data from millions of video images taken at a typical 25 ft. interval would take an enormous amount of manpower and cost million of dollars to extract roadway data for each state DOT in the US. Thus, there is a need to automate this process. Our research is motivated by this need. We have developed video image sign recognition ([1], [2]) and roadway geometry features to address this need. This paper focuses on introducing algorithms to automate roadway geometry measurement using video images.

In the past, the edge mark recognition is usually developed for the purpose of automatic navigation of vehicles or intelligent drive assistance. The biggest challenge to lane detection methods are bad environmental conditions, such as low contrast, occluded lane markings, vehicles, puddle, ruts, shadows on a road surface, and so on. Weather and lighting are also major factors in bad conditions. There are various research groups that have tackled lane edge detection. The lane detection [3] is performed by finding possible edge features extracted from search windows set in a road image. The search windows handle a part of a whole image, but noise edge features are mistaken of lane markings when noise edge features increase in bad conditions. Some systems try to improve robustness by using a global road shape constraint and by using pixel data extracted from all over the image. Kluge [4] uses a least median of square estimation to determine the curvature and orientation of a road. Lane markings are individually found by a horizontal intensity profile. Though the least median of square estimation is robust for noisy data, real-time response is not assured because of its repetitive computations. Pomerleau [5] accomplishes an adaptively high-speed matching procedure to determine at lane's curvature and lateral offset. It has the advantage of handling any type of cue corresponding to a lane edge on the road surface. Though this algorithm, it reduces computation cost for rapid response and results in low precision due to neglected parameters. Kluge [6] and Kreucher et al. [7] lane detection maximizes the function that estimates the possibility of how well particular lane model parameters match the pixel data in a road image. They used a Metropolis algorithm to maximize the function, which is a random walk like stochastic optimization using an annealing process. Though the Metropolis algorithm achieves robust estimation, it also sometimes fails to find a global maximum. Kreucher [8] presented another algorithm based on the frequency domain features to extract edge information that seems to work well under conditions in which the spatial domain algorithm failed. Takahashi [9] presented an edge detection algorithm based on a global voting process in which the patterns with the greater number of votes are chosen as the best approximations of the left and right lane markings. Using multiple cues and particle filtering, Apostolof [10] provided another novel approach to lane detection. Gern [11] made use of optical flow to successfully extract the lane edges under adverse weather conditions. Xu [12] used a linear lane model to do the lane tracking. Bertozzi [13] introduced a more comprehensive set of various algorithms for lane edge detection. For the transform between image and world coordinates, Bucher [14] presented a method mapping image coordinates with world coordinates that we adopted into our algorithm.

In order to achieve our goal of automatically measuring pavement geometric properties (i.e. number of lanes, widths of travel lane and shoulder widths) using video images, we have developed an algorithm using color segmentation, edge mark extraction, and roadway geometry measurement. We have organized this paper as follows. The need for developing such an image pattern recognition algorithm and application is first introduced along with a brief literature review. An algorithm is proposed and presented in the subsequent section. The strength and limitations of the proposed algorithm are, also, discussed in the final section.

2 Proposed Algorithm

With the need discussed in the previous section, we have developed an image pattern recognition algorithm using local thresholding and predictive edge extraction. We have used pavement markings to identify the travel lane width, and we have used the contrast between grass and pavement to identify the shoulder width. The algorithm consists of three parts: image color segmentation, edge mark extraction, and roadway geometry measurement. The objective of image segmentation is to separate yellow and white edge marks and non-pavement backgrounds, such as grass, woods, or buildings, from the pavement surface background to facilitate the edge mark extraction. Edge mark extraction extracts all the white and yellow edge marks (as regions of interest) and combines broken edge marks if necessary and puts them in order from left to right. In this stage, false edges are also eliminated. The last stage is geometric information collection, which makes use of the result of extraction and calculates the travel lane width, travel lane number, shoulder width, etc. The following are the detailed presentations of the algorithms that we used.

2.1 Image Color Segmentation

In the segmentation, we adopt two types of approaches to work complementally. The first segmentation method makes use of the intensity and color information to segment the white and yellow edge marks from the pavement background. The second method makes use of the gradient to segment the boundary between pavement and background. In some highways, edges are not marked either by yellow or white, in which case the most efficient way to determine edges is by means of gradient because the gradient in the boundary area is usually much higher than internal area. The segmentation of non-white-non-yellow edges and shoulder border requires an algorithm totally different from previous algorithms. The Sobel Gradient to detect non-white and non-yellow edges and shoulder edges is used. Due to the content space limitation, this section presents only the first segmentation method.

Pavement surface is, generally, gray, which is dominated by blue or green but with very little color saturation. The yellow edge marks are dominated by two color components, green and red, and are usually brighter than the pavement surface. The white edge marks have about the same level of intensities in all three colors (red, green, and blue) and are significantly brighter than the pavement surface. A grass or woods background is dominated by green and generally has a much lesser intensity than pavement surfaces. The greatest challenge, however, is that the pavement surface has a widely variable intensity, not only because of the non-uniformity of pavement

surface (existence of black and white spots, also known as pepper and salt noise), but also because of the variation of brightness along the pavement surface due to different reflectivity levels from different angles into the camera. What is more, usually in every travel lane, the middle area is always darker than the area close to a lane border, primarily because the black asphalt at the top of the pavement changes to a lighter color with the increased abrasion by vehicle wheels, and the two sides of the lane always receive far more wheel wear than the middle of the lane. The variation in brightness between the lane middle areas and the side areas can reach up to 30% of average intensity, in which case a global threshold to segment pavement surfaces usually does not work very well. To compound the problem, the white edge lines tend to lose their brightness and become less distinguishable from the pavement surface because of aberration and corrosion. Sometimes, using cameras with different responses (filters) can result in different tints for the same scene, and a weakly blue pavement surface could change. Figure 1 shows the flow chart for the color segmentation. The threshold will have to be dynamic to succeed in separating the background from the edge lines. To make a reliable pavement surface segmentation, we use the following steps:

1. Calculate the average r_{mean} , g_{mean} , b_{mean} and standard deviation r_{std} , g_{std} , and b_{std} of each of the three-color components in two image locations. One is located in the bottom (size of $m \times n$) of the image, and the other is located in the middle area (size of $m' \times n'$). Here m , n , m' , n' are parameters that can be adjusted to result in the best result. In our experiment, m is half of image width, n is one tenth of image height, m' is a quarter of image width, and n' is one tenth of image height. Interpolation of these calculated averages and standard deviations reveal the approximate average and standard deviation distribution in the full range of pavement surface of the image.
2. Classify each pixel based on the following criteria:
 - a. A pixel is regarded as a pavement pixel if

$$r - b < k \cdot |r_{mean} - b_{mean}| \text{ and } g - b < k \cdot |r_{mean} - b_{mean}| \text{ and } |r - r_{mean}| < r_{std}$$
 or

$$|r - r_{mean}| < \alpha \cdot r_{std} \text{ and } |g - g_{mean}| < \alpha \cdot g_{std} \text{ and } |b - b_{mean}| < \alpha \cdot b_{std}$$
 - b. A pixel is regarded as a potential yellow edge pixel if it does not satisfy criterion (a) and if

$$r > \lambda \cdot b \text{ and } g > \lambda \cdot b \text{ and } r > r_{mean}$$
 - c. A pixel is regarded as a white one if it does not satisfy criteria (a) and (b), and it satisfies:

$$r > r_{mean} + \rho \cdot r_{std} \quad \text{and} \quad g > g_{mean} + \rho \cdot g_{std} \quad \text{and} \quad b > b_{mean} + \rho \cdot b_{std}$$
 - d. A pixel that does not meet any of the above criteria is called non-pavement background pixels.

In the above, r , g , and b are the red, green, and blue intensities for a pixel, and r_{mean} , b_{mean} , and g_{mean} are the interpolated red average, blue average, and green average, and r_{std} , g_{std} , and b_{std} are the interpolated standard deviations of the red,

green, and blue averages on the pavement surface. α, λ and ρ are constants that are related with the camera and pavement conditions. The average and standard deviation are used to remove the pavement background. To get the pavement width for the situation where there are no white or yellow edge marks that separate the pavement from non-pavement, the Sobel Gradient is adopted.

Fig. 1. Flow chart of color segmentation

2.2 Edge Extraction and Validation

Figure 2 shows the flow chart for edge extraction. Edge extraction makes use of a depth-first search algorithm to search every white or yellow region of interest (ROI) and to mark every pixel with a certain ROI ID number. Then, we do the linear fit for each ROI and calculate the slope a , intercept b , and the standard deviation STD of the pixels along the line. For a group of points $(x_1, y_1), (x_2, y_2), \dots, (x_n, y_n)$ in a certain ROI, we assume that $x = ay + b$ passes through the middle of this ROI. Then Equation (1) is used to calculate a and b :

Fig. 2. Flow chart for edge mark extraction

$$\left\{ \begin{array}{l} a = \frac{n \sum_{i=1}^n x_i y_i - \sum_{i=1}^n x_i \sum_{i=1}^n y_i}{n \sum_{i=1}^n y_i^2 - \left(\sum_{i=1}^n y_i \right)^2} \\ b = \frac{n \sum_{i=1}^n x_i \sum_{i=1}^n y_i^2 - \sum_{i=1}^n x_i y_i \sum_{i=1}^n y_i}{n \sum_{i=1}^n y_i^2 - \left(\sum_{i=1}^n y_i \right)^2} \end{array} \right. \quad (1)$$

To indicate how close to the line of all points in the ROI are, we use standard deviation (STD) as a benchmark. An ROI that is narrow and very close to a straight line has a smaller STD than an ROI that is wide or curved. The formula to calculate STD is:

$$STD = \sqrt{\frac{\sum x_i^2 - a \sum x_i y_i - b \sum x_i}{n}} \quad (2)$$

After getting the slope, intercept, and STD for all ROIs, they are put in ascendant order according to their slopes. If an ROI is surrounded by a rectangular box described by (x_1, y_1) and (x_2, y_2) , and has the total pixel number of N_{pixels} , then they are validated one by one first using the criteria:

- A valid ROI has the top of the ROI below the vanishing point.
- A valid ROI has the height of $(y_2 - y_1 + 1) > 20$
- A valid ROI satisfies: $STD < 0.3 * \text{height}$
- A valid ROI satisfies: $(y_2 - y_1 + 1) * STD < N_{\text{pixels}}$
- A valid ROI's drawn line will pass through more same color pixels than other color pixels.
- A valid yellow ROI edge's slope is always negative.
- A valid ROI edge always has a distance from the camera's vanishing point smaller than a certain threshold (i.e. 50).

After the above validation process, there may still be redundant edge lines. The following processes are used to remove redundant edges and both lines are then determined:

- First, set those edges whose lengths are longer than a certain threshold as solid yellow or solid white lines.
- If there exist both solid yellow and solid white edge lines and if the highway is divided, then we have already got the traveling lane edges, and we can directly go to shoulder edge detection step.

- c. If at least one side does not have a solid edge, then further analysis is needed to determine whether there exists a partly blocked solid edge. If all edges are found to be lane-separating marks that don't determine the pavement edges, we need to use the Sobel Gradient to detect real pavement edges. If it is a blocked solid edge, then this side is finished. The Sobel Gradient is used to detect the edge, which will be described below, in which case no shoulder detection process is needed.

2.3 Roadway Geometry Measurement

After all edge marks, including those non-white-non-yellow edges, have been extracted, geometric optics are used to compute the pavement geometric information, including pavement width, number of lanes, shoulder width, and possibly the intersection information. To calculate the travel lane width and shoulder width, the pavement area is first identified. In the pavement area, if both left-most edge mark and right-most edge mark are identified, then the travel lane width is calculated by means of geometric optics. Before that, the necessary calibration is needed to determine how many inches a

Fig. 3. Original images and the extracted roadway geometry

pixel stands for in each row. If we assume the vanishing point position in an image is at $P(x_v, y_v)$, then the distance that per pixel stands for is:

$$d = \frac{k}{y_v - y} \quad (4)$$

For a divided highway, travel lane width has a left border of yellow edge line and right border of solid white edge lines. The shoulders are usually situated left from the yellow edge and right of the most-right solid white edge. The widths of the shoulders are usually calculated from the edge line to the borders of pavement surface. For an undivided highway, the travel lane is usually calculated from the leftmost solid white edge line to the rightmost solid edge lines, and shoulder widths are calculated from the pavement boundary to two solid white lines. For the case where no pavement edge lines exist, the derived edges from gradients by means of the Hough Transform are adopted. Here, shoulders are ignored. The number of lanes in the travel lanes is derived by dividing travel lane width by 11 feet and rounding to the nearest integer.

Figure 3 shows the examples that roadway geometry, including travel lane and shoulder lane widths and number of lanes, can be automatically extracted using video images with the developed algorithm. Figures 3.a and 3.b show that the developed algorithm can handle the roadway with and without pavement marks.

3 Summary

Traditional roadway geometric properties, including number of lanes, travel lane and shoulder widths, are measured in the field. Roadway geometric property measurement is one of the most important and, yet, the most time-consuming and riskiest component of the roadway data inventory. This paper presents an automatic roadway geometry feature extraction algorithm using a highly reliable and effective image extraction algorithm using local thresholding, predictive edge extraction, and geometric optics. This methodology makes segmentation very accurate for color images, and it dynamically adjusts the segmentation criteria and is able to deal with widely different lighting conditions. This algorithm combines statistics-based color segmentation, depth-first search edge extraction, geometric optic measurement, and edge detection based on gradient and Hough Transform. It utilizes localized color segmentation to minimize the impact from non-uniform pavement intensity over the image view field. It also uses recursive iteration to get the accurate edge line slope and intercept and result in more accurate measurement of lane width and shoulder width measurement. The sequential images are processed and used to improve the reliability of automatic pavement measurement performance. The preliminary study shows that the developed algorithm can automatically extract roadway geometry effectively and will improve current roadway inventory dramatically.

A large-scale experimental study using video images collected by state DOT is under way. The accuracy of the extracted results will be evaluated, and the improved productivity will be measured. One of the most important features of pavement geometrical property is the curvature. We will implement a reliable highway horizontal

curvature algorithm to measure horizontal curvature. It is also suggested to incorporate and post-process GPS data to assist in roadway geometry validation and calibration.

References

1. Tsai, Y. and Wu, J.: Shape and Texture-based 1-D Image Processing Algorithm for Real-time Stop Sign Road Inventory Data Collection, *Journal of Intelligent Transportation System*, Vol. 7, pp. 213-234.
2. Wu, J. and Tsai, Y. (2005), Speed Limit Extraction and Recognition Algorithm Using Locally Adaptive Thresholding and Depth-First-Search, *Photogrammetric Engineering and Remote Sensing (PE&RS) Journal*. (in press)
3. Behringer, R.: Road Recognition from Multifocal Vision, *Proceedings of IEEE Intelligent Vehicle 1994*, Paris, France, pp.302-307, 1994.
4. Kluge, K.: Extracting Road Curvature and Orientation From Image Edge Points Without Perceptual Grouping Into Features, *Proceedings of IEEE Intelligent Vehicles 1994*, Paris, France, pp. 109- 114
5. Pomerleau, D.: RALPH:Rapidly Adapting Lateral Position Handler, *Proceedings of IEEE Intelligent Vehicle Symposium*, Detroit, MI, USA, September, 1995, pp.506-511.
6. Kluge, K.: A Deformable-template approach to lane detection, *Proceedings of IEEE Intelligent Vehicle Symposium*, Detroit, MI, USA, September, 1995, pp.54-59.
7. Kreucher, C., Lakshmanan, S., and Kluge, K.: A Driver Warning System based on the LOIS Lane Detection Algorithm, *Proceedings of IEEE International Conference on Intelligent Vehicles*, Stuttgart, Octobre 28-30 1998, pp.17-22.
8. Kreucher, C. and Lakshmanan, S.: LANA: A Lane Extraction Algorithm that Uses Frequency Domain Features. *IEEE TRANSACTIONS ON ROBOTICS AND AUTOMATION*, VOL. 15, NO. 2, APRIL 1999, pp343-350.
9. Takahashi, A., Ninomiya, Y., Ohta M., and Tange KA Robust Lane Detection using Real-time Voting Processor, *Proceedings of the IEEE Intelligent Transportation Systems*, Tokyo, Japan, October 5-8, (1999), pp. 76-79.
10. Apostoloff, N., and Zelinsky, A.: Robust Vision based Lane Tracking using Multiple Cues and Particle Filtering, *Proceedings of the IEEE Intelligent Vehicles Symposium 2003*, Columbus, Ohio, USA, June, 9-11, 2003 pp. 558-563.
11. Gern, A., Moebus, R., Franke, U.: Vision-based Lane Recognition under Adverse Weather Conditions Using Optical Flow , *Proceedings of the IEEE Intelligent Vehicles Symposium 2002*, Versailles, France, Jun 17-21, 2002 pp. 652-657.
12. Xu, Y., Wang, R., Li, B., Ji, S. A Vision Navigation Algorithm Based on Linear Lane Model. *Proceedings of the IEEE Intelligent Vehicles Symposium 2000*, Dearborn (MI), USA October 3-5, pp. 240-245. (2000).
13. Bertozzi, M. Broggi, A., Cellario, M., Fascioli, A., Lombardi, P., and Porta, M.:(2002). Artificial Vision in Road Vehicles, *Proceedings of the IEEE - Special issue on Technology and Tools for Visual Perception*, 90(7):1258-1271, July 2002
14. Bucher, T.: Measurement of Distance and Height in Images based on easy attainable Calibration Parameters, *Proceedings of the IEEE Intelligent Vehicles Symposium 2000*, Dearborn(MI), USA. October 3-5, 2000, pp. 314-319.