An Algorithm to Estimate Vehicle Speed Using Un-Calibrated Cameras

D.J. Dailey
Dept. of Electrical Engineering
University of Washington
Seattle, WA 98195
dailey@ee.washington.edu

L. Li
Dept. of Electrical Engineering
University of Washington
Seattle, WA 98195

Abstract

In this paper we present a new algorithm to estimate speed using a sequence of video images from an un-calibrated camera. The algorithm uses frame differencing to isolate moving edges and track vehicles between frames. The algorithm uses a known vehicle length distribution with image information to estimate speed.

1 Introduction

Image processing techniques have been applied to traffic scenes for a variety of purposes including: queue detection, incident detection, vehicle classification, and vehicle counting [1, 2, 3, 4, 5]. In this paper, we present a new algorithm to estimate speed using a sequence of video images from an un-calibrated camera. This work is motivated by the large number of roadside cameras installed by DOT’s to observe traffic. The cameras are typically not installed in a manner that they can easily be calibrated, and they are typically used by operators who can tilt, pan, and zoom using a joystick to change the camera calibration. The combination of movable cameras and lack of calibration makes estimating speed for un-calibrated cameras a challenge.

Relatively few efforts have been made to measure speed using video images from un-calibrated cameras. Some preliminary research on pixel speed estimation in images appears in [4]. In previous work few efforts were made to map pixel speed to ground truth speed. A review of the literature on speed estimation using cameras indicates that most algorithms either reference information in the scene or create such references interactively. For example, Worrall [6] reports an interactive tool to perform camera calibration in which an operator uses parallel road marks to identify vanishing points and then places a rectangular calibration grid on the image. Further, in [7] and [8], speed measurements are made using the known physical distance between two detection windows placed on the road image by an operator. Similarly, several other authors [9, 10] suggest estimating speed by placing two detection lines, of known separation, in the image and measuring travel times between the lines. In addition, Houkes [11] suggest the selection of 4 reference points forming a rectangle and performing off-line measurements. All these methods require the operator to perform a calibration procedure before speed estimation can be undertaken.

In this paper, it is assumed that we have no control over camera movements, and thus cannot directly obtain information such as camera focus, tilt, or angle. It is further assumed that the camera parameters can change with time. In the work presented here, we are monitoring congested freeways and have neither the ability nor the authority to set permanent marks on the road. Given this scenario, we believe on-line calibration is a necessary step to enable the use of the large, installed base of TMS cameras.

We assert that exact calibration is not necessary to estimate speed. Instead, we use: (1) geometric relationships inherently available in the image, (2) some common sense assumptions (listed below) that reduce the problem to a 1-D geometry, and (3) the distribution of vehicle lengths, to propose a novel method that extracts scale information and estimates speed.

To describe and demonstrate our speed estimation scheme, we first review the assumptions made in formulating the algorithm. We then enumerate the steps of the algorithm, followed by a discussion of the individual steps. Finally we present some preliminary quantitative results of the algorithm.

2 Underlying Assumptions

To create an algorithm to estimate speed from video images we make several assumptions to simplify the problem:

1. The speed of the vehicles is finite. The speed of a vehicle has both physical and legal limits.

2. The vehicle movement is smooth. There are no sudden changes of direction in the time interval (330ms) between frames in the image sequence.

3. Motion is constrained to the road plane. Tracking of vehicles in the image sequence is a one dimensional problem.

4. The scale factor (feet per pixel) varies linearly along the direction of vehicle travel. This assumption constrains the vehicles to be moving generally toward or generally away from the camera.
5. The lengths of the vehicles in the images are realizations from a known vehicle length distribution.

With these assumptions, the vehicles are treated as though they travel in one dimension along a straight line in the image. The vehicles are tracked across these images to obtain scale factors that estimate the real-world distance represented by pixels at various locations in the image. Using a linear function to fit to the empirical scale factors it is possible to estimate the real-world distance traveled. Combining the distance traveled with the known frame rate allows us to estimate speed. An algorithm to perform this estimation is presented in the next section.

3 The Algorithm

The algorithm operates on a series of at least five sequential images. The inner loop operates on sequential groups of three images to create one enhanced image. The outer loop uses a sequence of enhanced images to estimate speed.

**Outer Loop**

1. Obtain five or more sequential images (320x240), gray scale at three frames per second (e.g. $[I_1, I_{i+1}, I_{i+2}, I_{i+3}, I_{i+4}, ..., I_{i+N}]$ where $N \geq 5$)

2. Create sets of three sequential images
   
   (e.g. $[I_i, I_{i+1}, I_{i+2}]$ is the $i$th set of $(N-2)$ sets)

**Inner Loop**

   For each of the sets of three sequential video images, perform the following:

   (a) Median filter each image.

   (b) Difference the first and second images ($I_i - I_{i+1}$) as well as the third and second images ($I_{i+2} - I_{i+1}$) to get two difference images.

   (c) Apply a Sobel edge detector to the difference images to obtain edge images $Sobel(I_i - I_{i+1})$ and $Sobel(I_{i+2} - I_{i+1})$.

   (d) Threshold the edge images to create binary images.

   (e) Intersect the two binary images to obtain the moving edge image ($ME(I_{i+1})$) for the $I_{i+1}$ image:

   $\begin{align*}
   ME(I_{i+1}) &= \text{Threshold}(Sobel(I_i - I_{i+1})) \\
   &\cap \text{Threshold}(Sobel(I_{i+2} - I_{i+1})).
   \end{align*}$

   (f) Apply dilation to the moving edge image.

   (g) Apply erosion to the moving edge image.

   (h) Identify the set of points $C_j(I_{i+1})$ for the $j$ convex hulls in the moving edge image $ME(I_{i+1})$.

   (i) Calculate centroid $p(i + 1, j) = (x, y)$ for the $j$th convex hull in image $I_{i+1}$.

   (j) Calculate the set of points for the bounding boxes $B_j(I_{i+1})$ for the $j$ convex hulls.

**End of the inner loop**

3. Select sets of co-linear centroids $\{p(i + 1, j), p(i + 2, j), p(i + 3, j), ..., p(N - 2, j)\}$ in sequential images and estimate a best fit line through these points. The slope of this line is the tangent of the angle of motion $\alpha$ for the $j$th centroid in the series of images. This is used to establish a new coordinate, $z$, along the direction of motion such that $z^2 = x^2 + y^2$ and $\tan(\alpha) = \frac{dy}{dx}$.

4. For each of the collinear bounding boxes in sequential images, estimate the pixel length $L(i + 1, j)$ along the direction $\alpha$ using $\frac{\sup(y : y \in B_j(I_{i+1})) - \inf(y : y \in B_j(I_{i+1}))}{\sin(\alpha)}$.

5. Estimate the scale factor $q$ (feet/pixel) for the $z$ location of the centroid $p(i + 1, j)$ using the mean vehicle length $l$,

   \[ q(i + 1, z) = \frac{l}{L_p(i + 1, j)}. \]

6. Using a series of two or more scale factor estimates, estimate the slope ($m$) and intersection ($b$) of the scale factor function $q(z|m, b)$ using

   \[ \min_{(m, b)} \|q(z|m, b) - q(i + 1, z)\| \quad \forall z, \]

   where $q(z|m, b) = mz + b$, and $z$ is the distance along a line at an angle $\alpha$ in the images.

7. Estimate the interframe distances,

   \[ d_k = \int_{z_k}^{z_{k+1}} q(z) dz \quad \forall \quad k \in (i + 1, N - 2). \]

8. Estimate the mean of interframe distances, $E[d_k]$, and use the ratio of the interframe mean and the frame rate ($\Delta t$) to estimate speed,

   \[ \hat{S} = \frac{E[d_k]}{\Delta t}. \]

**End Outer Loop**

4 Algorithm Operation

To explain the operation of the algorithm just enumerated, we identify the basic tasks necessary to estimate speed from sequential un-calibrated images and map these tasks into the steps in the algorithm. The tasks necessary to obtain speed from un-calibrated images are: (1) obtain sequential images, (2) identify the moving vehicles in the sequential images, (3) track the
vehicles between images, (4) dynamically estimate the scale factor in feet per pixel, and (5) estimate speed from distance traveled and the interframe delay.

As mentioned earlier, the algorithm operates on sets of sequential images taken from DOT CCTV cameras. The images used in this work are, grey scale, 320 by 240 pixels, sampled three times per-second. The resolution and sample rate are selected to provide sufficient detail in the image to identify individual vehicles and to capture sequential images rapidly enough so that individual vehicles can be tracked between images without pattern recognition techniques (e.g. the vehicles moves no more that about one vehicle length between images).

The images used in our algorithm are taken from roadside CCTV cameras installed by Washington State DOT in their traffic management role. The DOT transports the video from the roadside cameras to the control center using a dedicated fiber system. In the control center, operators can pan, tilt, and zoom the cameras using a joystick. The cameras are actively being used for traffic management activities. No camera calibration information is available for these cameras, and it is the purpose of this work to demonstrate that the images from such cameras can be used as an alternative speed estimate.

The video is digitized at a rate of three-frames per second and stored in files using the Jpeg image format. These Jpeg files are the sequential images required for step one in the outer loop of the algorithm. Sets of three sequential images are used in the inner loop of the algorithm. The left side of Figure 1 shows two example images.

In step (b) of the algorithm, a median filter, using a 3x3 kernel, is applied to each image to remove high frequency noise in the images [13, 14].

To identify the moving vehicles in the images, the non-moving background must be removed. Two basic techniques to remove the static background information appear in the literature. The first is to obtain a frame with only the background that can be subtracted from the frames in which there are vehicles [15]. This frame is then updated to match the current lighting levels [2]. This method is not only computationally expensive, but it may be impossible, on congested freeways, to obtain an image with the correct lighting level and with no vehicles present. The second technique uses sequential frames to perform forward and backward differences between the frames [16, 15, 17]. Viren [17] suggests using interframe differences with a differential operator to extract moving edges.

The algorithm presented here uses interframe differences and then applies a Sobel edge detector to the resulting image. Step (d) of the algorithm creates two difference images, and step (e) applies the Sobel edge detector to those images. The resulting images are thresholded to obtain a binary images. The upper right image in Figure 1 is the binary image that results of applying the Sobel edge detector to the difference of the two images in the left column of Figure 1. The two binary images are intersected in step (f) of the algorithm to obtain a moving edge image. The resulting moving edge image for the example sequence appears in the lower right of Figure 1.

Examining the lower right image in Figure 1 shows that while we have identified the moving edges, those
edges do not make closed polygons identifiable as individual vehicles. To overcome this problem and create closed curves, we use two morphological operations. We enhance the moving edge image by sequentially applying dilation and erosion. [18] Dilation of an object is the translation of all of its points with regard to a structural element followed by a union operation. Dilation is applied to the binary image to close the curves in the moving edge image; it also expands the overall size of the area enclosed. Erosion is then used to shrink the object back to the original size. In the algorithm presented, a 3x3 structural element is used in step (f) to perform dilation and in step (g) to perform erosion.

After the application of the morphological operators, the moving edges are filled in to create moving blobs. These moving blobs represent the vehicle motion in the images. Past work has asserted that the convex hull surrounding a vehicle in an image is a good approximation of the projection of a vehicle in the image [19]. To characterize the moving blobs, we first calculate the convex hull for the blobs in step (h) of the algorithm. We also calculate the centroid of the convex hulls in step (i). The centroids of the convex hulls are used as the effective location of the vehicle in the image.

Having located a vehicle in one image, the vehicle is tracked across images by enforcing co-linearity of the centroids of the convex hulls. The left side of Figure 2 presents a representation of three convex hulls with centroids \( x_1, y_1 \), \( x_2, y_2 \), \( x_3, y_3 \). In step (j) of the algorithm, the vehicle is tracked as moving along the line at an angle \( \alpha \) relative to the horizontal scan lines in the image. In the work presented here, a minimum value of 0.90 of the linear regression correlation coefficient,

\[
\tau = \frac{n \sum_{i=1}^{n} x_i y_i - \sum_{i=1}^{n} x_i \sum_{i=1}^{n} y_i}{(n \sum_{i=1}^{n} x_i^2 - (\sum_{i=1}^{n} x_i)^2)^{1/2} (n \sum_{i=1}^{n} y_i^2 - (\sum_{i=1}^{n} y_i)^2)^{1/2}},
\]

is used to identify co-linear centroids and track a vehicle. This completes tasks one through four necessary to estimate speed.

The fifth task necessary to estimate speed is to make an estimate of the scale factor that maps distance traveled in the image to distance traveled on the road. We have assumed the vehicles are taken from a known distribution [20], and we can use the properties of that distribution to estimate the length of the vehicles in the images. We make individual estimates of the scale factor \( q \) as the ratio of the mean vehicle length \( \bar{l} \), taken from the known distribution, and the estimate of vehicle length from the image. This latter length is approximated by the length of the line, in the direction of travel, crossing the bounding box surrounding the convex hull of the vehicle. The right side of Figure 2 illustrates this approximation, and step 4 in the algorithm provides this estimate.

We assume that the scale factor \( q \) changes linearly along the path that the vehicle travels,

\[
q(z|\mu, \beta) = mz + b,
\]

where \( z \) is the distance along a line at an angle \( \alpha \) in the images. The parameters of this scale factor function are estimated in step 6 of the algorithm using the set of scale factor estimates from step 5. The distance traveled is the integral of this function along the \( z \) direction,

\[
d = \int_{z_1}^{z_2} q(z)dz.
\]

This distance is estimated in step 7 of the algorithm.

Finally, having an estimate of the distance traveled, we use the interframe sample time in step 8 to estimate the vehicle speed,

\[
\hat{S} = \frac{E[\Delta s]}{\Delta t}.
\]

This provides an estimate of speed from un-calibrated cameras.

5 Empirical Results

This paper presents preliminary results of applying the algorithm to images from a variety of lighting conditions. Figure 3 is a histogram of the error between the individual speed estimates and the ground truth. These results are for 60 tests of the algorithm without regard to lighting effects. There are errors as large as 30% in one of the tests. On examining the relationship between lighting conditions and the error in the estimates it has become clear that the shadow effects may account for the errors of over 10% in the speed estimate. Reconciling the algorithms against lighting conditions is an ongoing effort.

This paper presents a new algorithm to estimate speed from un-calibrated cameras. Un-calibrated cameras are widely available to DOT operators and can provide a valuable, additional quantitative measure for traffic operations and traveler information.

Acknowledgments

This work was supported in part by a series of contracts with the Washington State Department of Transportation and by the U.S. Department of Transportation.

References


Figure 2: Use of centroids to establish the travel direction $\alpha$ (left), Use of bounding box to estimate vehicle length (right).

Figure 3: Histogram of percent error.


