A VIDEO-BASED METHOD FOR THE DETECTION OF TRUCK AXLES

FINAL REPORT

OCTOBER 2002

Budget Number. KLK474
N02-05

Prepared for

IDAHO DEPARTMENT OF TRANSPORTATION

Prepared by

NATIONAL INSTITUTE FOR ADVANCED TRANSPORTATION TECHNOLOGY

UNIVERSITY OF IDAHO

James F. Frenze

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

I. INTRODUCTION

To ensure longevity and minimize maintenance costs of roads and bridges it is important for the trucking industry to comply with load regulations. Maximum permissible load is based upon vehicle axle configuration. The Idaho Transportation Department was interested in developing an advisory system that promotes voluntary compliance by informing drivers of the maximum permissible load. Current systems used to enforce load regulations include weigh bridges, weigh-in-motion systems and road tubes. Weigh bridges use static scales, requiring the truck to be stationary, to measure gross weight and weight per axle and are not viable due to the volume of truck traffic on highways. Weigh-in-motion systems can be used to weigh trucks in motion at highway speeds. They also measure speed, height of the truck, number of axles and inter-axle distance [1]. However, such systems are expensive, non-portable, and pavement intrusive. Road tubes, used to count the number of axles, provide good accuracy but have associated maintenance costs and are often pavement intrusive.

In this paper we discuss an alternate approach using image analysis. This method ispromising as it offers high flexibility without pavement intrusion. It provides inter-axle spacing and can be extended without requiring additional hardware resources to collect other traffic parameters such as speed and body type, without interference to traffic. Furthermore, a video-based method can detect the presence of a variable load suspension (VLS) system, an auxiliary axle that can be raised or lowered. Road protection requires proper use of such systems; thus, the Idaho Transportation Department is interested in collecting data on the usage of VLS systems. Current methods in use cannot gather this information.

A number of studies involving image processing for collecting traffic parameters, such as traffic volume, vehicle type and queue parameters have been reported in the literature. In [2], image windows are placed on lanes to detect vehicles by applying edge detection within the window and processing the histogram of each window. Speed is computed by measuring the time taken for the vehicle to travel between the windows. Motion detection and vehicle detection are applied to detect queue parameters. In [3], the use of imaging techniques to identify vehicle make and model is described. Infrared images are used to reduce sensitivity to body color and lighting conditions. Local features are extracted and matched against a training set to recognize vehicles. In [4], a system for the detection and classification of vehicles is described. The location, width, length and velocity of regions that are a part of the vehicle are detected to classify them. This scheme, while collecting most vehicle characteristics, does not detect axles and VLS systems.

An optical system was developed to count axles as explained in [5]. It uses a laser diode to scan the lane, while a position detector detects reflected light from the road surface, axle and vehicle body. A range image is formed by collecting reflected light positions and used to distinguish between reflecting surfaces such as the road, vehicle body and wheels. A peak in the range image indicates the presence of an axle. This method, while offering good accuracy, does not provide the advantage of extending the system to collect other traffic parameters and provides only binary information on the presence of axles.

The Hough transform is a pattern recognition technique used to identify shapes that can be parameterized, such as straight lines, circles, and ellipses. The Hough transform algorithm and its variations have been used in numerous pattern recognition applications as listed in [6]. In [7], the Hough transform is used to detect circular and sub-circular features from aerial photographs, representing archaeological monuments. We use the generalized Hough transform to detect the presence of circular patterns representing wheels, as the transform is robust to missing data and noise in the image. A survey of Hough transform methods is presented in [8].

This paper is organized as follows. Section II describes the experimental setup. Section III is an introduction to the Hough transform algorithm. Section IV details the methods employed. Section V is a tabulation of the results obtained. Section VI provides an estimate of the computational complexity and Section VII is a description of a prototype system. Section VIII presents the conclusions from this research and identifies areas for future research.

II. EXPERIMENTAL SETUP

We used a collection of pre-existing videotapes provided to us by the Idaho Transportation Department. These tapes were recorded in NTSC format from cameras that were mounted on the side of two-lane highways. The cameras were mounted at an angle to the road facing the on-coming traffic as shown in Figures 1 and 2. The camera field of vision was such that long trucks spanned multiple frames. Furthermore, measurements regarding camera position

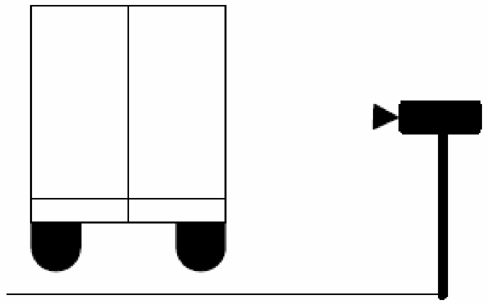


Figure 1. Side view of the camera mounting.

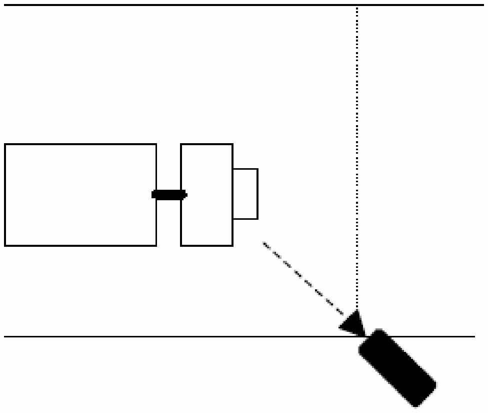


Figure 2. Top view of camera mounting showing the approximate mounting angle to the truck.

relative to the road were not available. Image sequences from a VCR were acquired into image buffers using a Matrox Orion framegrabber, as shown in Figure 3. The Matrox imaging software library routines were used to grab and display interlaced images with a resolution of 640 x 480 pixels from the videotape. The processing algorithms were implemented in the C programming language and executed on a 1 GHz Pentium III PC with 512MB of memory.

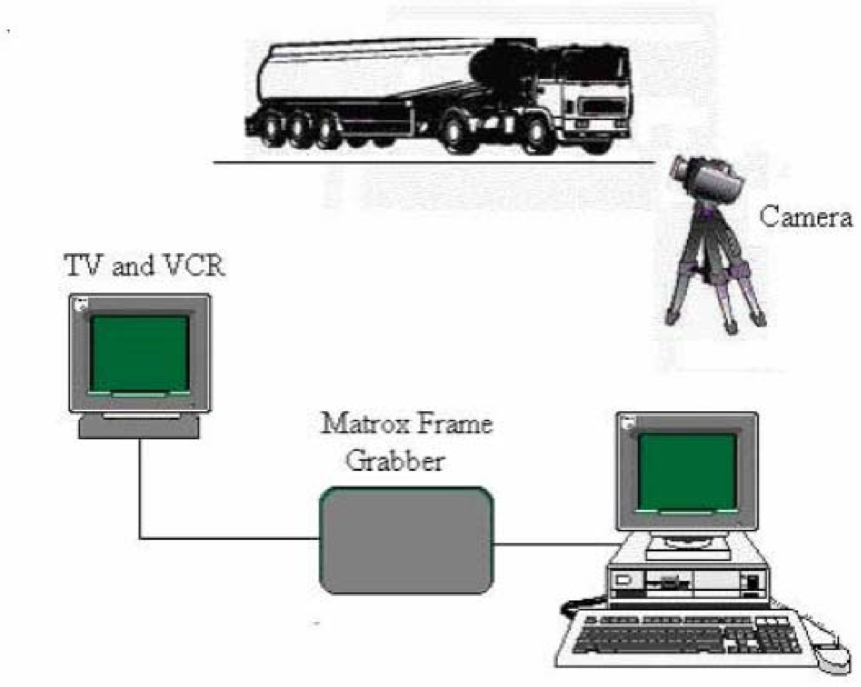


Figure 3. Experimental set-up showing the VCR, Matrox Orion frame grabber and the camera.

III. HOUGH TRANSFORMATION

Circular features of the wheels of the truck are detected using the Hough transformation. This algorithm is capable of recognizing shapes in images through a voting procedure, using a "one to many" mapping of a feature space point to the possible parameter space points. The voting is constrained by the parametric equation. The parameter space is an accumulator array, with its dimension corresponding to the feature space. Each cell in the parameter space accumulates evidence of possible circle centers. The circle is parameterized as

$$(x-cx)^2 + (y-cy)^2 = r^2, \quad (1)$$

where (cx,cy) are the coordinates of the center of the circle and 'r' is the radius of the circle. The accumulator array collects votes for possible values of center locations (cx,cy) for each edge point (x,y) in the feature space for a specific radius r. Each edge point (x,y) on the circle in the feature space contributes votes in a circular shape on the parameter space, as shown in Figure 4. The intersection of these circles forms a local maximum in the parameter space, indicating the presence of a circle in the feature space. Multi-dimensional accumulator arrays are required for detection of circles with differing radii, with each accumulator array representing specific radius value [9]. The Hough transform is generally robust to noise, since random noise does not contribute uniformly to a single accumulator cell. However, noise due to poor image quality and segmentation causes an increase in the background level in the Hough accumulator array, while missing object edges or occluded shapes produce lower peaks.

Wheel locations are restricted to a region close to the road, as shown by the band in Figure 5. A generalized Hough transformation, as explained above, was used to detect the circular feature of the wheels. Votes corresponding to a complete circle in the parameter space must be recorded for each edge point in the feature space. This was done by testing points within a square region of the parameter space, with a side dimension equal to the diameter of the circle, for satisfaction of equation 1.

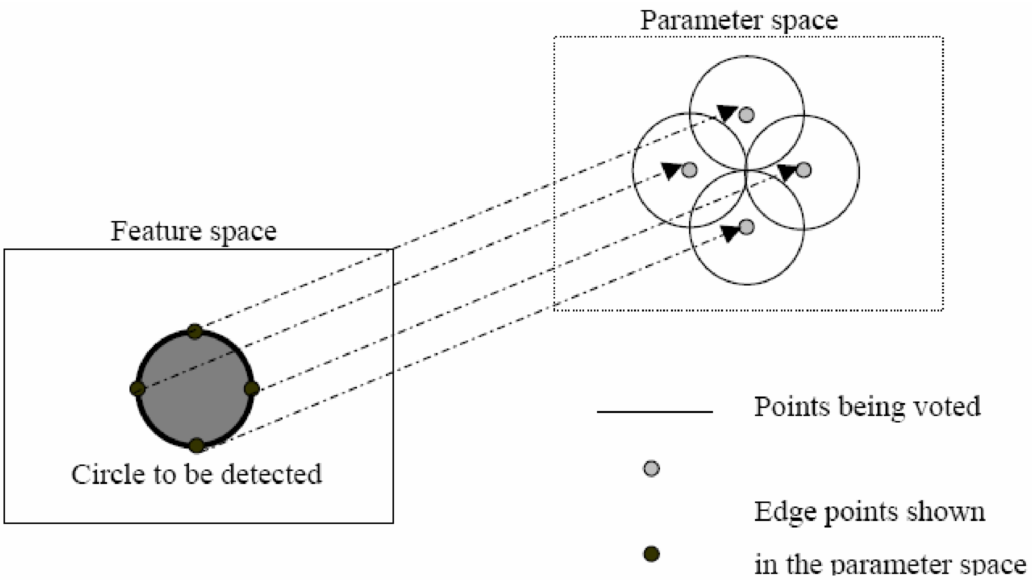


Figure 4. The feature space shows the circle to be detected and the points on the circumference for which votes are collected in the parameter space. The circle drawn with thin lines indicates the points in the parameter space that are voted for the points at its center.

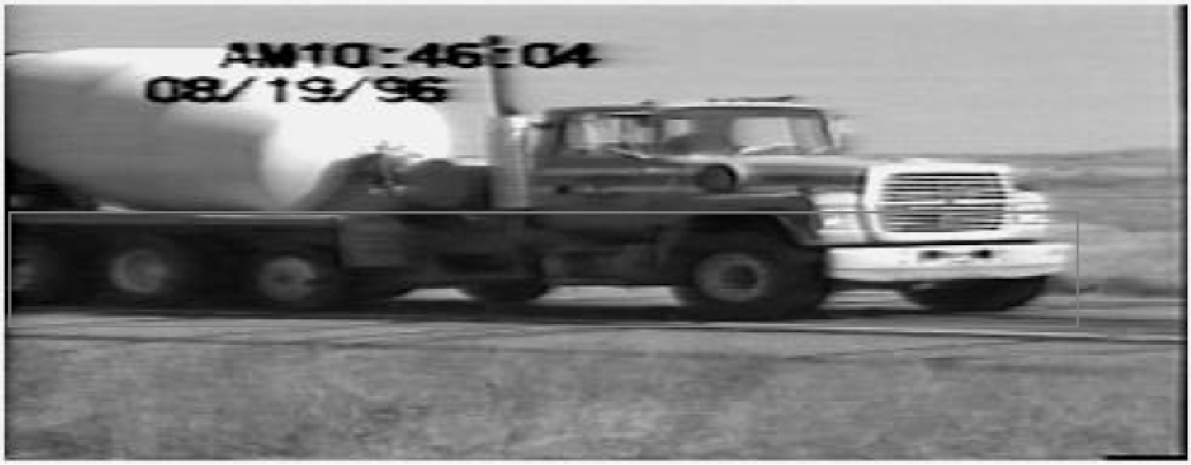


Figure 5. The outlined area indicates the region of possible wheel locations, restricting the wheel search area.

IV. METHOD

Images from the camera using the setup described above were recorded on videotape and were used for testing the system. Images were captured in grayscale to reduce computation time and buffer space requirements. The processing sequence is shown in Figure 6. The frames were captured in a synchronous mode using MIL library functions, at a rate of 15 frames/sec. This frame rate was sufficient to extract vehicle features.

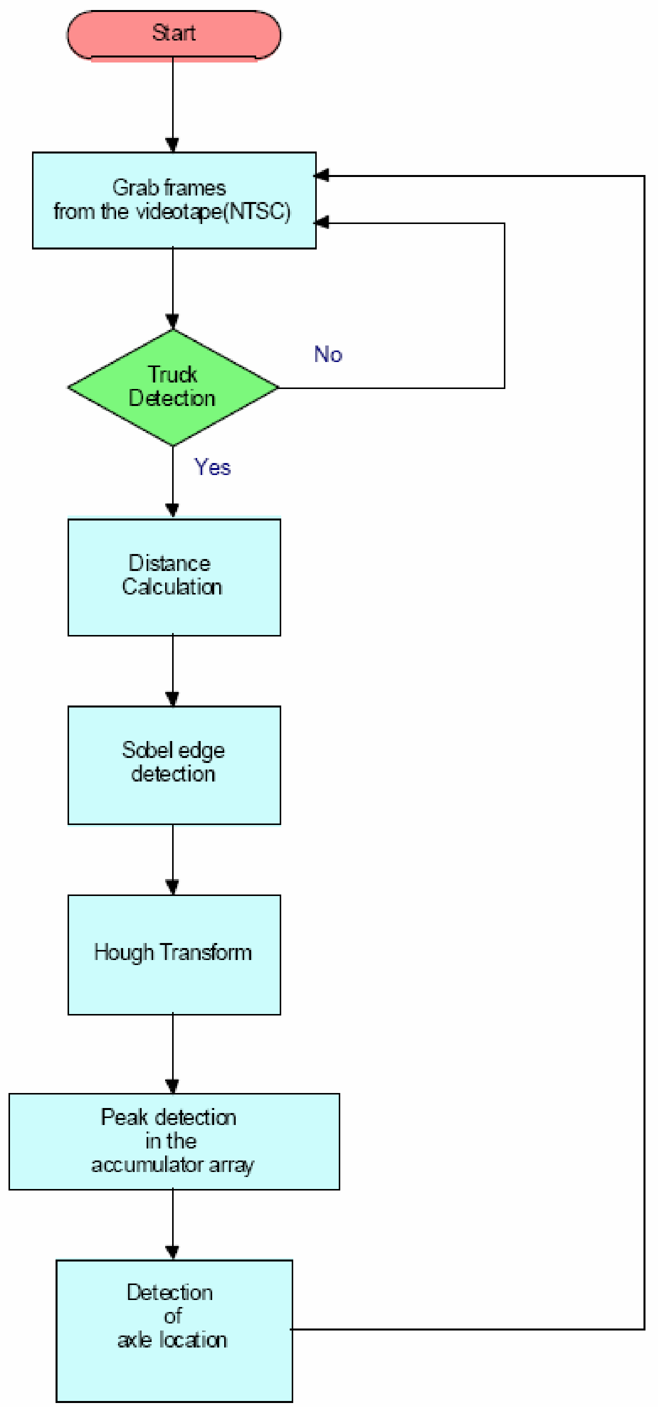

Start
Grab frames from the videotape(NTSC)
Truck Detection
No
Yes
Distance Calculation
Sobel edge detection
Hough Transform
Peak detection in the accumulator array
Detection of axle location

Figure 6. Flowchart outlining the methods used for axle detection.

A. Truck Detection

In the first stage of processing, frames containing trucks were captured. This was done using a trigger at the left hand side of the frame, as shown in Figure 7. A small window of size 5 x 5 pixels was used as a virtual detector, since processing needs to be performed on this region for all the frames at the frame rate of 15 frames/sec. Detectors were positioned for individual camera views to avoid triggering by cars, SUV, and traffic in the far lane. Trucks were detected by subtracting the background image from the current frame in the window area and summing the magnitude of the difference of individual pixels. A suitable threshold was set as a trigger for the window area processed by examining the images in the videotapes. Once a truck enters the field of view, it triggers the detector. The same method was used to determine when a truck exited the field of view. The video frame rate exceeds the axle-processing rate and hence all of the captured frames were buffered. Further processing of the video sequence is suspended while the buffered frames are processed; this can result in missed vehicles under heavy traffic flow.

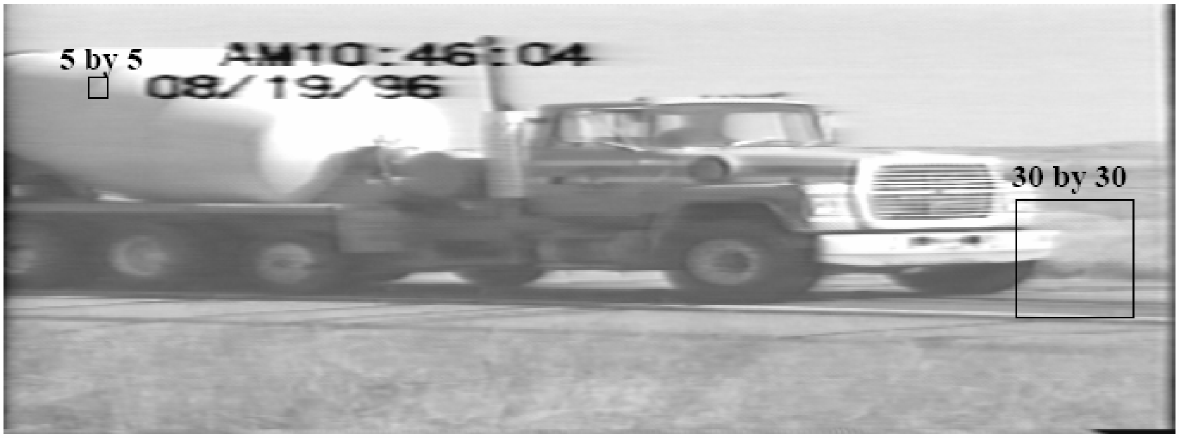


Figure 7. Virtual detector windows are shown at the left and the right sides of the frame to detect the truck entering and exiting the field of view.

B. Distance Estimation

In order to reduce processing time, the Hough transform is performed over a small region, as shown in Figure 8. As a result, axle detection will span multiple frames and the distance traveled by the truck between frames is needed to determine axle spacing. The distance in pixels is added to each frame processed to obtain the position of wheels with respect to the first detected wheel. A template window of size 60 x 60 pixels, positioned to capture a portion of the wheel and the truck body, was used to determine the distance between subsequent frames. A moving correlation is performed, with the best correlation indicating the template displacement. Computing the distance between alternate frames instead of adjacent frames reduces the error in distance estimation. The actual distance traveled by the truck in inches can be mapped to the pixel distance by calibrating the camera setting. This could be done using a placard of known dimensions on the side of the road or using markings on the road, as explained in [4]. Speed of the truck was obtained by using the distance estimate between subsequent frames and the frame rate.

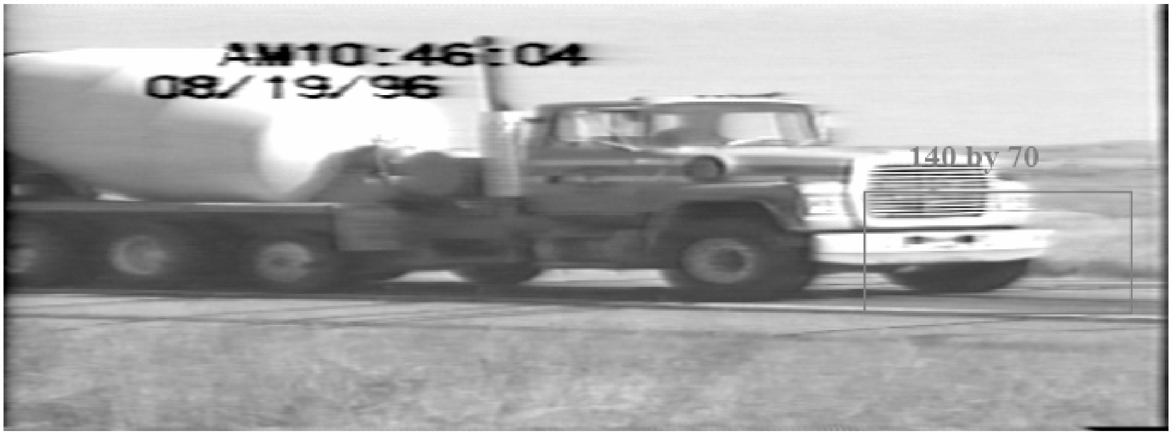


Figure 8. Hough Transform processing window is shown at the right corner of the frame. Edge detection and the Hough Transform are applied to this region and the axle position is recorded.

C. Edge Detection

The interlaced fields produced ghosting due to the motion of the vehicle between fields. To overcome this problem, odd lines from the frame were discarded, reducing the vertical resolution by a factor of two. The grayscale images thus obtained were converted into black and white images using thresholding. The Sobel gradient-based edge operator was applied to the binary image to extract the edge between the rim and the tire. The edge detection was performed on a small region equal to the size of the Hough processing window (140 x 70 pixels) to reduce computation and improve the efficiency of the system. The gradient threshold to identify an edge was set based on a study of a series of images. Since the camera was mounted at an angle, the edge image obtained is slightly elliptical along the y-axis. As the odd fields of the frame are discarded the shape becomes more circular.

D. Hough Transform

As mentioned above, the Hough transform was performed over a small portion of the band, in order to reduce processing time and the number of duplicate counts. The window size for processing was set, as shown in Figure 8, based on the maximum distance traveled by the truck in pixels between frames; this ensures that no wheels are missed.

The accumulator array collects votes for a specific radius and was processed for the presence of distinct peaks as an indicator of the location of the wheels. A suitable threshold is fixed to indicate the presence of a wheel in the accumulator array. The distance traveled by the truck between frames was used to obtain inter-axle distance. The peaks in each of the accumulator arrays from individual frames are assembled into a composite to obtain the axle locations. This axle location information could be used to access a database of known axle configurations mapped to the load limit for the specific configuration.

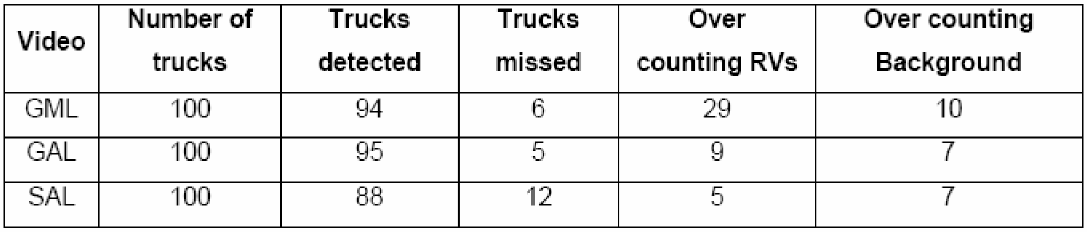
V. RESULTS

Videotapes recorded under various lighting conditions were tested. The criteria for performance evaluation were the accuracies of truck detection, distance estimation and axle detection, including variable load suspension systems.

A. Truck detection result

A large number of trucks were tested for vehicle detection and the results verified manually. The results are tabulated in Table 1 for various lighting conditions and indicate that the number of missed trucks is low. False positives are due to recreation vehicles detected as trucks and background variations due to changes in ambient lighting.

TABLE 1 **Truck detection results indicating over-counting due to RVs and background variations**



Video	Number of trucks	Trucks detected	Trucks missed	Over counting RVs	Over counting Background
GML	100	94	6	29	10
GAL	100	95	5	9	7
SAL	100	88	12	5	7

B. Distance estimation result

The distance traveled in pixels by the truck between frames computed by the algorithm was compared with a manual estimate of distance traveled. The manual estimate was calculated by inspecting pixel reference points between alternate frames of the truck. These results are tabulated in Table 2. Inter-axle distance was obtained by comparing pixel distance to known feature dimensions. Error in the distance estimate is caused by inaccuracies in the template matching.

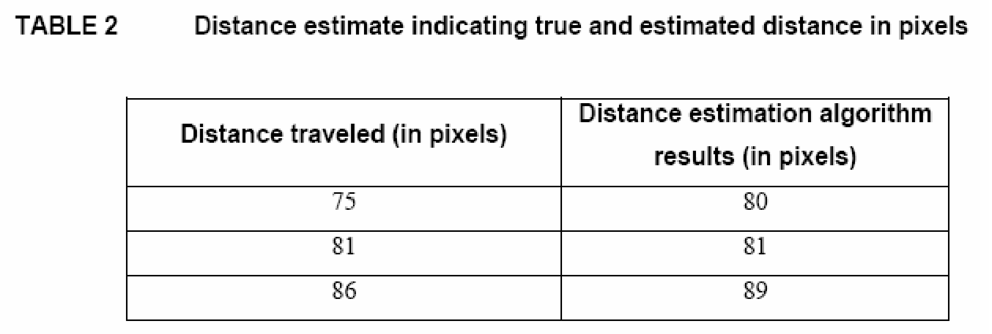


TABLE 2 Distance estimate indicating true and estimated distance in pixels

Distance traveled (in pixels)	Distance estimation algorithm results (in pixels)
75	80
81	81
86	89

C. Axle counting result

The results of the wheel position detection algorithm are tabulated in Table 3. These results, as shown in Figure 9, indicate that the false positive count is high. This is due in part to the

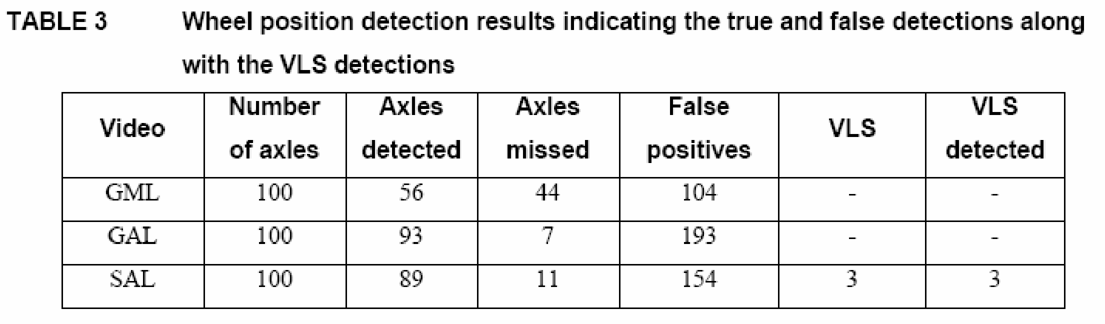
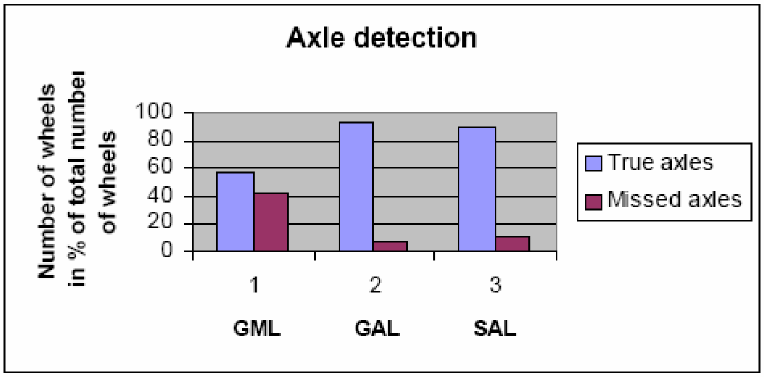


TABLE 3 Wheel position detection results indicating the true and false detections along with the VLS detections

Video	Number of axles	Axles detected	Axles missed	False positives	VLS	VLS detected
GML	100	56	44	104	-	-
GAL	100	93	7	193	-	-
SAL	100	89	11	154	3	3

wires and chains hanging from the body of the truck and grill features on the front of the truck. These features result in additional edges in the edge image and subsequent peaks in the Hough accumulator array. Furthermore, the wheel image is not perfectly circular, due to the camera angle and processing only one field of the interlaced images. This causes a lower peak in the Hough accumulator array. Combined, these factors result in multiple peak values that are close to those caused by the wheels, resulting in false positives. Strong light conditions yield better contrast between the rim and tire resulting in more continuous edges. Axle detection hence performs better under strong light and is sensitive to lighting conditions.



GML: Good morning light

GAL: Good afternoon light

SAL: Strong afternoon light

Figure 9. Graph showing the performance of the axle detection algorithm under three different lighting conditions.

VI. COMPUTATIONAL COMPLEXITY

The computational complexity of the algorithms was studied to identify areas for improvement. Certain frames undergo more processing compared to others, depending on the presence of trucks in the frame. As shown in the Figure 6, the MIL grab function is called at the rate of 15 frames/sec, with the left window trigger function performed on every grabbed frame. Frame grabbing stops after a truck is captured and resumes only after it has been processed. Hence, the number of times this function is called over an interval of time depends on the traffic distribution.

The code was profiled using function-profiling in Visual C++ to determine the computational load distribution of the MIL functions and the algorithms over a fixed time, as shown in Table 4. This gives an estimate of the time spent, including looping overheads in the code. The distance calculation is performed once per captured truck and is shown in the table. The MIL timer function was used to obtain an accurate estimate of the time distribution in algorithms per frame after the truck has been detected; the results are shown in Figure 10. Processing time for the virtual detectors occupies a very small percentage of the total processing time and has not been included.

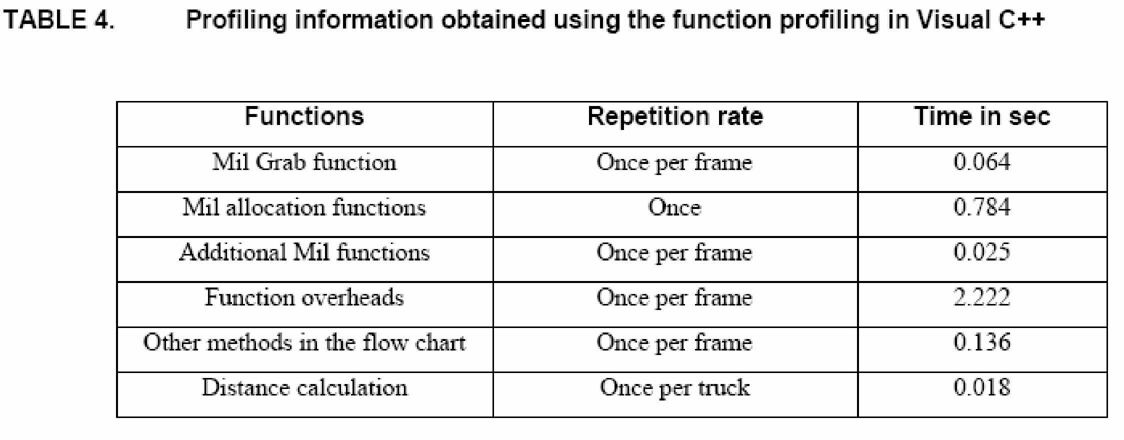


TABLE 4. Profiling information obtained using the function profiling in Visual C++

Functions	Repetition rate	Time in sec
Mil Grab function	Once per frame	0.064
Mil allocation functions	Once	0.784
Additional Mil functions	Once per frame	0.025
Function overheads	Once per frame	2.222
Other methods in the flow chart	Once per frame	0.136
Distance calculation	Once per truck	0.018

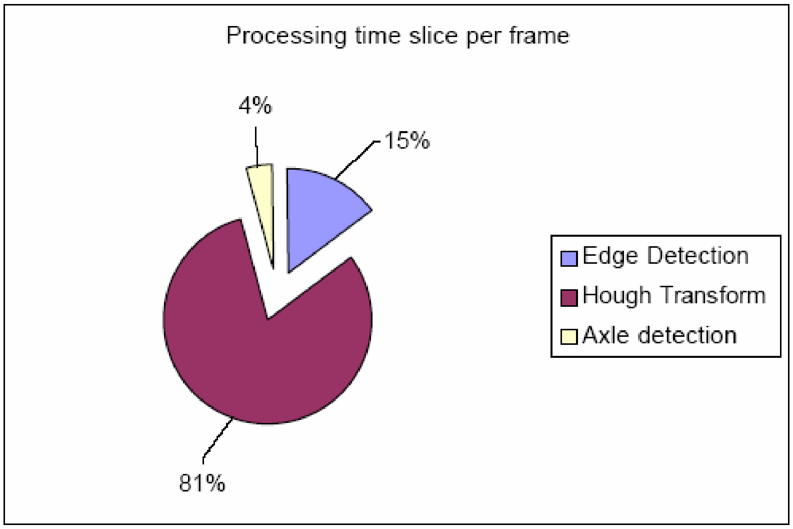


Figure 10. A pie chart showing the percentage of processing time for processing steps performed per frame after the truck has been captured. The percentages are in terms of the total time per frame.

It is clear that the majority of the time was spent in performing the Hough transform. The reason for this is that we used the generalized Hough transform with a minor variation as explained in Section III. Further reduction in computation could be achieved by using edge orientation information as described in [10]. The edge direction on the circle circumference points towards the circle center, thus reducing the search area for each edge point. This reduces the computation load, since only an arc at a distance equal to the radius in the direction of the edge needs to be tested for possible center location. The computational time of the Hough transform could be reduced by a factor of six using this method [10]. If this scheme were adapted in our method, the computational time for the Hough transform would be 25 ms and it would be possible to realize this system in real-time. Other optimization techniques for storage and computation requirements are presented in [11] and [12].

VII. HARDWARE IMPLEMENTATION

The computational complexity in the previous section indicates that a major portion of the time is spent in the MIL library functions. This overhead could be reduced and the processing implemented in real-time if a dedicated processor was used. A possible stand-alone system that could be installed by the side of the road, based on the Clarity ASIC from Sound Vision, Inc.[1] is shown in Figure 11. Clarity includes an ARM RISC processor, built-in memory controller, memory bus interface, USB interface, LCD interface, programmable CCD and CMOS sensor interfaces, and various other peripherals. Source code can be written in Visual C.

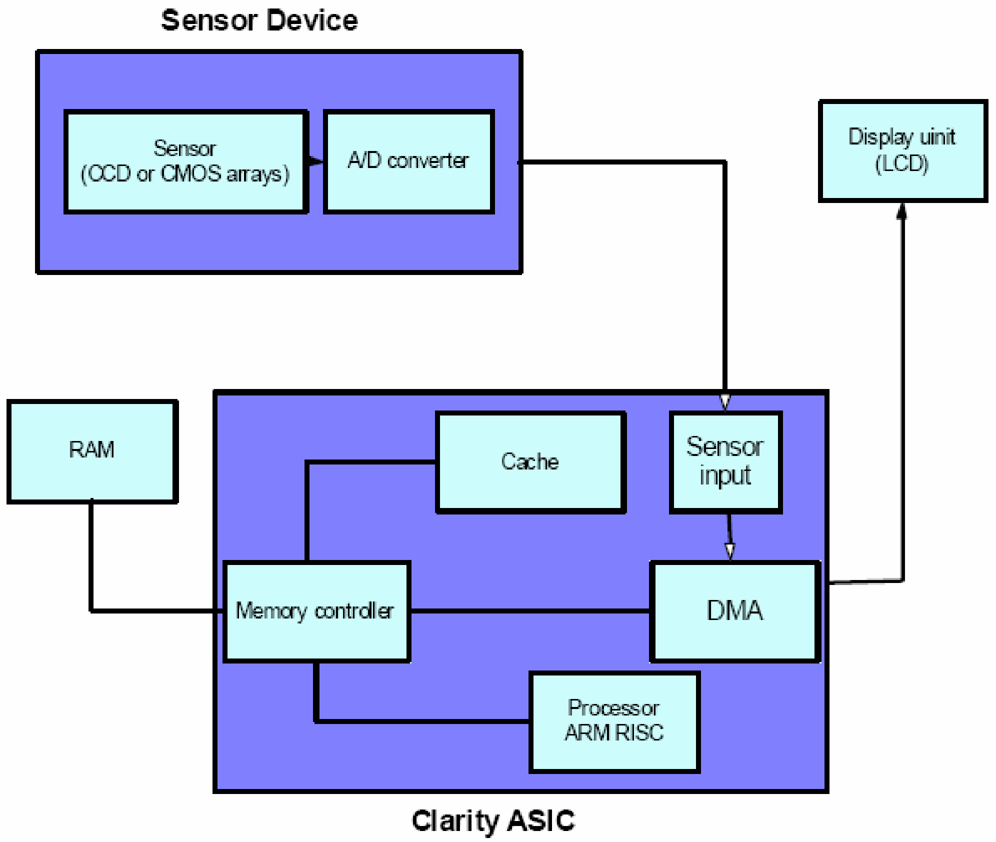


FIGURE 11. Block diagram of the proposed stand alone system using the Clarity ASIC.

The image sensor could be either a CCD array or CMOS sensor. The choice depends primarily on the cost and the image quality required. CCDs cost about $120, while CMOS sensors cost about $15. However, CMOS sensors have low sensitivity, yielding poor image quality under low light conditions due to the pixel fill factor, the ratio of the photodiode area to photosensitive area. A portion of the photo site in CMOS sensors is taken by additional circuitry required to filter noise. The CCD sensors offer column and row accessibility, which facilitates window-of-interest readout. This could prove useful in our application, eliminating the need to buffer the entire frame. These sensors have various configurations, such as color vs. monochrome, different resolutions, and data rate (frames/sec). A monochrome sensor with a resolution of 640 x 480 pixels and data rate of 15 frames/sec would be suitable for our application.

Clarity is not compatible with all types of sensors and LCD displays. Compatible CCD devices are manufactured by Sony, Sharp and IBM. CMOS sensor manufacturers include Motorola, Hyundai and Agilent. Supported color LCDs are available from Seiko, Epson, Unipac and Kopin. The operating temperature of this stand-alone unit is 32º F to 149º F. The Clarity’s operating temperature limits the lower operating temperature of the system. This would restrict operation to seasons in which the daytime temperature remained above freezing. Estimates of the system component costs are shown in Table 5. These costs include

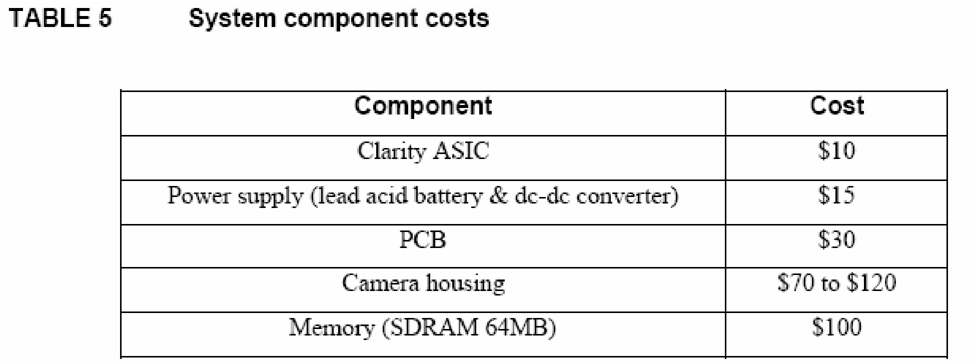


TABLE 5 System component costs

Component	Cost
Clarity ASIC	$10
Power supply (lead acid battery & dc-dc converter)	$15
PCB	$30
Camera housing	$70 to $120
Memory (SDRAM 64MB)	$100

only the basic hardware components shown in the block diagram in Figure 11. The approximate cost of building this system with just the basic components and the CMOS sensor option is $270, while it increases to $375 if CCD arrays are used.

[1] Sound Vision Inc. , 432 Boston Post Road, Wayland, MA 01778. (www.soundvision.com)

VIII. CONCLUSIONS

In this paper we have presented a video-based method for the detection of axles. Computational complexity and the computation time were estimated. These indicate that this method is computationally intense and requires further research to implement as a real-time system. Edge detection produces noisy image due to objects in the wheel vicinity, such as the grill and wires hanging from the truck. The axle detection results indicate a high false positive count and sensitivity to lighting conditions. Results obtained indicate good accuracies with truck detection and speed estimation, while altering the camera mounting would yield better axle detection.

Further investigation into edge detection is required to eliminate noise due to objects other than the wheel while still preserving the edges at the wheel in order to obtain better performance. The camera-mounting angle plays an important role in the correct detection of axles, as the wheel shape is dependent on the mounting angle. This video-based system would benefit from the use of adaptive thresholding to accommodate varied lighting conditions.

REFERENCES

[1] D. Cebon and C. B. Winkler, “Multiple-Sensor Weigh-in-Motion: Theory and Experiments,” *Transportation Research Record*, Vol. 1311, pp. 70-78, 1991.

[2] M. Fathy and M. Y. Siyal, “A window-based processing technique for quantitative and qualitative analysis of road traffic parameters,” *IEEE Trans. Vehicular Technology*, Vol. 47, pp. 1342-1349, Nov. 1998.

[3] M. Kagesawa, S. Ueno, K. Ikeuchi and H. Kashiwagi, “Recognizing vehicles in infrared images using IMAP parallel vision board,” *IEEE Trans. Intelligent Transportation Systems*, Vol. 2, pp. 10-17, Mar. 2001.

[4] S. Gupte, O. Masoud, R. F. K. Martin, N. P. Papanikolopoulos, “Detection and classification of vehicles,” *IEEE Trans. Intelligent Transportation Systems*, Vol. 3, pp. 37-47, Mar. 2002.

[5] T. Ueda, K. Takemura, S. Ogata and T. Yamashita, “Optical axle counting equipment,” in *Proc. ITSC*, 1997, pp. 326-331.

[6] J. Illingworth and J. Kittler, “A survey of the Hough Transform,” *Computer Vision, Graphics, and Image Processing*, 1988, pp. 87-116.

[7] R. Redfern and G. Lyons, “The application of digital techniques to the detection and extraction of archaeological earthwork monuments from aerial photographs,” *Information Technology Center Journal*, National University of Ireland 1998.

[8] J. Illingworth and J. Kittler, “A survey of efficient Hough Transform methods,” in *Proc. 3rd Alvey Vision Conf.*, 1987, pp. 319-326.

[9] E. R. Davies. *Machine Vision: Theory, Algorithms, Practicalities*. 2nd Ed., CA: Academic Press, 1997, pp. 212-214.

[10] C. Kimme, D. H. Ballard, and J. Sklansky, "Finding circles by an array of accumulators," *Communication of the ACM*, Vol. 18, pp. 120-122, Jan. 1975.

[11] T. J. Atherton and D. J. Kerbyson, "Size invariant circle detection," *Image and Vision Computing*, Vol. 17, pp. 795-803, Sept. 1999.

[12] H. K. Yuen, J. Princen, J. Illingworth and J. Kittler, "Comparative study of Hough Transform methods for circle finding," *Image and Vision Computing*, Vol. 8, pp. 71- 77, 1989.