Determination of Minimum Gap in Congested Traffic

Serena Blacoe[1], Colin Caprani[2], Eugene OBrien[1], Alessandro Lipari[1]
[1] School of Civil, Structural and Environmental Engineering, University College Dublin, Belfield, Dublin 4, Ireland
[2] Department of Civil and Structural Engineering, Dublin Institute of Technology, Bolton Street, Dublin 1, Ireland
email: serena.blacoe@ucdconnect.ie, colin.caprani@dit.ie, eugene.obrien@ucd.ie, alessandro.lipari@ucd.ie

ABSTRACT: Accurate evaluation of site-specific loading can lead to cost and material savings in rehabilitation and replacement of bridges. Currently, bridge traffic load assessment is carried out using long run traffic simulations based on weigh-in-motion (WIM) data obtained at the site. Congestion is the governing load condition for long-span bridges. To correctly model congestion, a minimum gap between vehicles is usually assumed. Where the gap is overestimated, the calculated characteristic load is smaller than the actual characteristic load leading to an unsafe assessment. If the gap is underestimated, the safety assessment is too conservative, which is both costly and wasteful of finite resources. This paper outlines the development of an optical method to measure parameters required to model driver behaviour in congestion. Images are obtained using a camera with a wide angle, aspherical lens. Edge detection and Hough transforms are used to location wheels and bumpers. The resulting data can increase the accuracy of traffic microsimulation and hence, the assessment of long span bridge traffic loading.

KEY WORDS: Bridge; Long-Span; Traffic; Loading; Image Processing; Hough Transform.

1 INTRODUCTION

1.1 Background

In 2007, road transport of freight was worth over €300 billion. Freight transport grows year-on-year, roughly in-line with economic growth [1]. The continued serviceability of bridges is essential to this trade growth. There are about 1 million bridges in Europe. The annual expenditure required to sustain the existing stock of bridges is estimated at €6.6 billion, with the replacement cost for bridges about €400 billion [2].

To minimize resources, more accurate assessment of the safety of bridges may yield substantial savings: over 50% of the European bridge stock is estimated to have been built pre-1970 [2]. Whilst it is important to be able to quantify the capacity of the bridge, since loading is far more variable, there is more scope for improving accuracy by determining the actual load to which it is subjected.

Site specific bridge assessment can offer accurate load prediction [3]. Data from a WIM system installed near the bridge can be used to calibrate a site-specific load model. These models are typically free-flow models for short-to-medium span bridges, or simple congestion models for long-span bridges [4]. Simulations of 1000s of years of traffic are possible with these models, giving accurate predictions.

1.2 Traffic load modelling

WIM data is usually collected for free-flowing traffic, and usually only contains trucks. This data is appropriate for bridges of short- to medium-spans [3], [5]. In the assessment of long span bridges (over about 40 m), congested traffic governs and the presence of cars must be considered. Recently traffic microsimulation has been used to model congested traffic streams for bridge loading calculation [6], [7]. A critical input parameter in such a model is the distribution of gaps between vehicles in congested traffic. As WIM data records the time between axles passing a location, from which the distance can be extrapolated if the traffic is free flowing, it is unsuitable for determining gaps in congested traffic. This is due to the fact that there may be long time gaps between axles passing a location, whilst they are physically close. Previously, minimum gaps have been simply assumed. Long-span bridge load effects are sensitive to the gap assumed and the variation can be as much as 20% to 30% [8]. This paper proposes a camera system to measure these gaps accurately, without impacting on traffic flow.

2 CAMERA SYSTEM

2.1 Previous use of cameras in similar applications

Cameras have been used previously for traffic applications such as surveillance, vehicle detection, and automatic vehicle guidance, but have not been used to calibrate a microsimulation traffic model. Frenze [8] developed a system that could identify the presence of a HGV and determine the axle configuration. Footage of traffic was captured by a camera placed roadside at an oblique angle to the direction of traffic. A region of interest (ROI) was specified, within which wheels should be found. Frames from the video footage were converted to binary colour using thresholding, after which the Sobel edge operator [10] was used to identify edges in the binary image. The circular Hough transform [11] was then used to identify likely locations of circle centres. Results were not perfect, with an over-identification of background points as wheels and only 94 out of 100 trucks detected. False positive identification of wheels was a more significant problem, which was attributed to the presence of too much information in the edge image. Fung et al [12] used a similar system. Instead of carrying out edge detection on a binary image, they carry out edge detection directly on the image obtained from the camera. In this instance, Gaussian filtering was used before applying the Sobel edge filtering to reduce

noise in the image. To speed up the processing, the images were compressed using the Haar wavelet [13], [14], which reduced the size of the image without loss of detail.

Aerial cameras have also been used. Gupte et al [15] used a pole mounted camera with an aerial view of traffic for the purpose of conducting vehicle counts, classification of vehicles, and to study lane changing behaviour. Unlike the other references, vehicles were detected and classified using a template matching system. During a 20 minute sequence, their algorithm correctly detected, tracked, and classified 63% of vehicles. Errors that occurred were explained by occlusion of parts of vehicles and/or poor image segmentation. Coifman et al. [16] address partial occlusion in congested traffic. Features such as corners are tracked, rather than attempting to track the entire vehicle. As features exit the tracking region, they are grouped using a common motion constraint.

Other approaches, such as that by Achler & Trivedi [17], train an algorithm to detect wheels by manually identifying wheels in a large training set of images. From this, Gaussian mixture models of wheels and of non-wheels are developed. Images can then be compared with these models and likely candidates for wheels identified. The foreground candidate wheels are tracked.

2.2 Site arrangement

For this work, two arrangements are considered for obtaining information on gaps in congested traffic. Firstly, a single camera with a wide-angle lens placed adjacent to the slow lane is considered. The benefits are that synchronization of footage is not needed; installation is easy; and the system is portable. Drawbacks of this arrangement are the required wide-angle lens, and potential interference from fast lane traffic on the identification of vehicles in the slow lane.

The second considered arrangement involves two cameras. One camera is placed beside the slow lane to obtain the internal vehicle dimensions (such as bumper to axle, axle to axle and axle to bumper distances). The second camera is aerial-mounted to determine the gaps between vehicles. An advantage is that a wide-angle field of view is not needed at the roadside, reducing possible interference from fast lane traffic. As a result, a standard lens can be used, and the camera can be placed close to the slow lane, reducing possible interference from fast lane traffic. Disadvantages of this arrangement include the footage synchronization, increased installation apparatus, and reduced portability.

The single camera arrangement is selected for the remainder of this work. Wide-angle lenses are easily obtained and interference from the fast lane can be removed through a common motion constraint, as was done in [16].

2.3 Camera and lens description

The camera used in this work is the IDS uEye LE USB camera [18]. This camera can accept a wide-angle lens, and can take more than 10 frames per second – a minimal specification identified for this application.

Wide-angle lenses are prone to aberrations which can affect shape and/or sharpness. For this application, shape distortion is the most critical aberration to avoid, as this would create an error in the calculated distances between objects in the image. Aspherical lenses do not suffer from these shape distortions, and so a Theia SY125M lens is used this work [19].

3 IMAGE PROCESSING

3.1 Overall approach

A similar approach to Fung et al. [12] and Frenze [8] is used for this work. The image is segmented and the Hough transform used to identify the location of wheels and axles in the image. Wheels are identified in lieu of actual axle identification. The images are reduced in dimension using wavelet compression without loss of resolution. This significantly reduces processing time for the Hough transform.

3.2 Edge detection for bumper identification

Truck bumpers are often vertical (see Figure 1). The Hough transform can identify linear features in a binary (black or white) edges-only image. Conversion of colour or grey scale images to black and white is straightforward, but edge detection is not. Edge detection algorithms work by determining the location of sudden luminosity intensity changes. Both the Sobel and Canny edge detection algorithms are considered for this work [10].

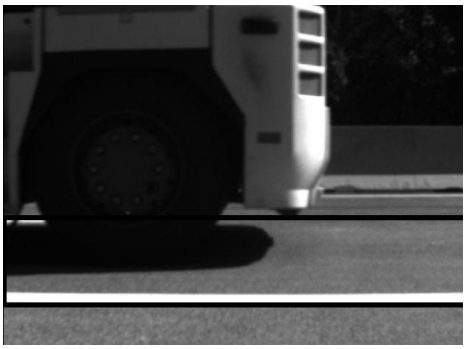
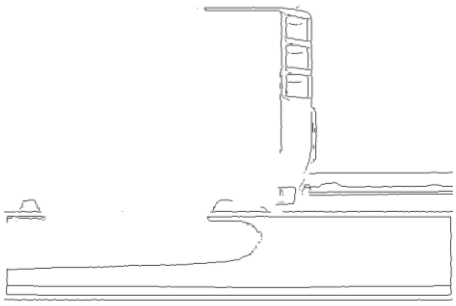


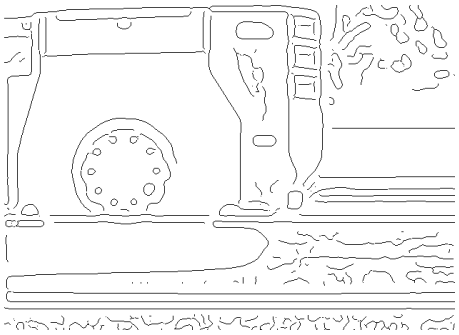
Figure 1. Sample truck bumper.

Sobel edge detection is based on a central difference approximation to the image gradient in the horizontal and vertical directions. A threshold is specified and any point with a gradient above this threshold is designated as an edge point [10]. The Canny edge detector first smooths the image using a Gaussian filter, and then determines gradients in the horizontal, vertical, and diagonal directions. The magnitude and direction of the edges are determined. Edge points are those points whose magnitudes are locally maximum in the direction of their gradients. The algorithm retains only pixels along the top of the ridges from the gradient calculation (non-maximal suppression). Two thresholds – an upper and a lower - are specified for edge points. Those points with magnitudes above the upper threshold are automatically retained as edge points. The points with magnitudes below the lower threshold are immediately discarded. Hysteresis thresholding is then carried out on the points with magnitudes that lie between the two thresholds. If a strong edge point exists in one of the eight pixels around it (8-connected to a strong edge), it is retained as an edge point; otherwise it is discarded [10]. In both the Canny and Sobel edge detector algorithms, the thresholds specify the proportion of non-zero points that are to be discarded (i.e. a threshold of 0.1 implies that 10% of non-zero points should be discarded).

The Sobel and Canny edge detection algorithms are applied to the sample truck bumper of Figure 1. The results are shown in Figure 2. As can be seen, the Sobel method fails to detect any element of the wheels (Figure 2(a)), whereas the Canny method has correctly identified the majority of the wheel outline (Figure 2(b)). It is possible to include more detail of the wheel outline by lowering the threshold in the Sobel method, but this would result in the inclusion of large amounts of noise. The Canny algorithm detects more of the wheel outline due to its inclusion of filters that detect diagonal edges. Although the algorithm does need to identify the location of the wheels, the bumper and wheel detection occur independently of each other. Consequently, the Sobel algorithm is used to create the edge image for the Hough line detection.



(a) Sobel method



(b) Canny method

Figure 2. Edge Detection applied to Figure 1.

3.3 Hough transform for bumper identification

A Hough transform is used on the established edge image. The Hough transform is an exhaustive search method commonly used in image segmentation tasks [10]. The algorithm cycles through every edge point in the image and determines the equations of the lines that could pass through each point.

The use of Cartesian coordinates is not suitable since the slope is infinity for vertical lines. Consequently polar coordinates are used in this work instead. Considering the polar equation of a line, the set of lines that pass through a point *A* at (x_0, y_0) are:

$$R(\theta) = x_0 \cos\theta + y_0 \sin\theta \qquad (1)$$

This process is illustrated in Figure 3, where example combinations of radius and angle that describe lines passing through point *A*, such as (R_1, θ_1) and (R_2, θ_2), are shown.

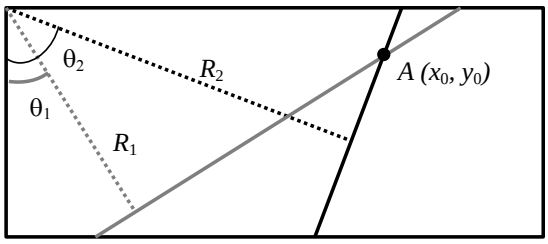


Figure 3. Polar coordinates of lines.

A two-dimensional parameter space of radius and angle can be constructed having dimensions of the discretized number of angles and radii. For each edge point, all possible radius and angle combinations are calculated. The weights of the points in parameter space corresponding to each of those combinations are incremented by +1. This is repeated for all edge points. These weights corresponding to the edge image in Figure 2(a) are shown in Figure 4. Strong linear features are evident at approximately 90 degrees at radii of about 400 and 600 pixels. These peaks in the parameter space correspond to lines in the original image as can be seen in Figure 5.

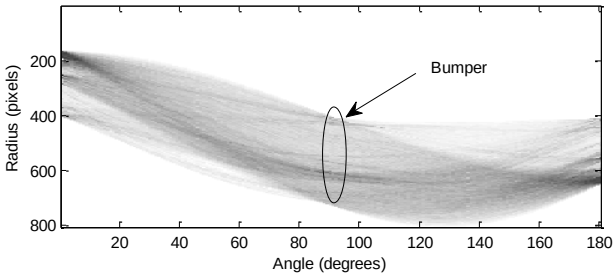


Figure 4. Hough transform parameter space.

Figure 5. Detected truck bumper.

This algorithm is applied to the more difficult problem posed by a car bumper, and the result is shown in Figure 6. The failure to detect the bumper is attributed to the lack of verticality of the bumper. However, this problem can be overcome by subtracting the background from the acquired images [17], but this is not further considered for the present work.

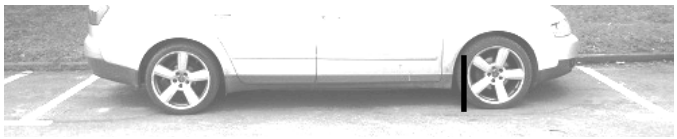


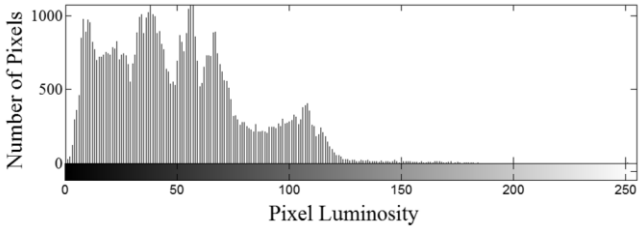
Figure 6. Incorrect detection of car bumper.

3.4 Wheel detection

The Hough transform has been successfully generalized to allow for the detection of shapes such as circles and ellipses [11]. However, the addition of the third variable – radius – to the Hough transform results in a computation of order $O(n^3)$, where n is the number of pixels. Hence the execution time increases significantly. Therefore in this work, several pre-processing steps are taken before carrying out the circular Hough transform.

A region of interest is specified around the band in the image where wheels are likely to be found. An example image is shown in Figure 7(a). As is evident from this figure, and the intensity histogram (Figure 7(b)) there can be very little contrast between the tyres, car, and road.

(a). Example image;

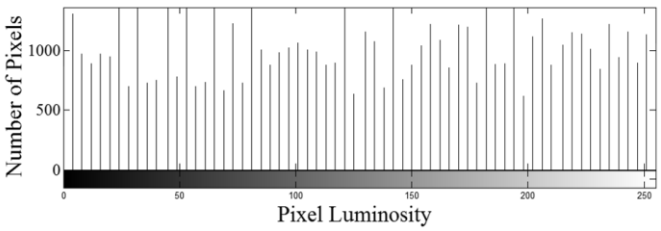


(b) Histogram of pixel luminosities;

Figure 7. Pre-processing of image with poor contrast.

The contrast in an image with poor contrast can be improved by altering the luminosity of pixels so that there are similar numbers of pixels with each value of luminosity within the image [10]. This is termed "equalizing the histogram". Figure 8 shows the same image after the histogram has been equalized and the equalized histogram.

(a) Equalized image;



(b) Equalized histogram;

Figure 8. Image after histogram equalization.

Minimization of the number of edge points in the edges-only image is important in reducing computation. To this end, the image is morphologically opened, then closed, using a disk shaped structuring element [10]. This removes small areas of detail from the image, as shown in Figure 9.

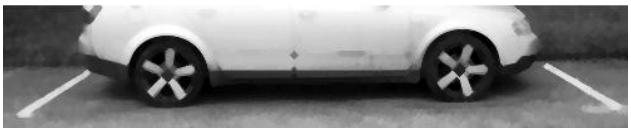


Figure 9. Example image after grayscale opening and closing morphology.

Next, thresholding is used to convert the image to binary: pixels with luminosity values above 25 in 8-bit grayscale representation (intensities range from 0 to 255) are converted to white; pixels at or below 25 are converted to black. The threshold value is an algorithm input variable that needs to be set for a particular site and particular conditions. Any areas of small detail are removed from this image through the process of erosion [10]. This results in the image of Figure 10. Edge detection using the previously-described Sobel operator is then carried out and the result shown in Figure 11.

Figure 10. Binary image after erosion.

Figure 11. Wheel edge image.

The circular Hough transform is carried out on the edge image shown in Figure 11. Rearrangement of the Cartesian parametric equations of a circle allows all possible centres for a given point (x_i, y_i) to be calculated:

$$y_c = y_i - r \cdot \sin\theta \tag{2}$$

$$x_c = x_i - r \cdot \cos\theta \tag{3}$$

Each edge point (x_i, y_i) is considered to lie on a circle. Using Equations (2) and (3), every possible centre for an edge point is calculated for a given radius. The weights in the parameter space corresponding to all possible centres for the edge point are incremented by +1. This is illustrated in Figure 12. As can be seen, the true centre of the circle will be incremented more than any of the other proposed centres. This calculation is carried out for every edge point and repeated for every radius to be considered. As the equations are in Cartesian coordinates, the parameter space will have the same height and width as the image. It will consist of several layers, or pages, with each page corresponding to a discretized radius.

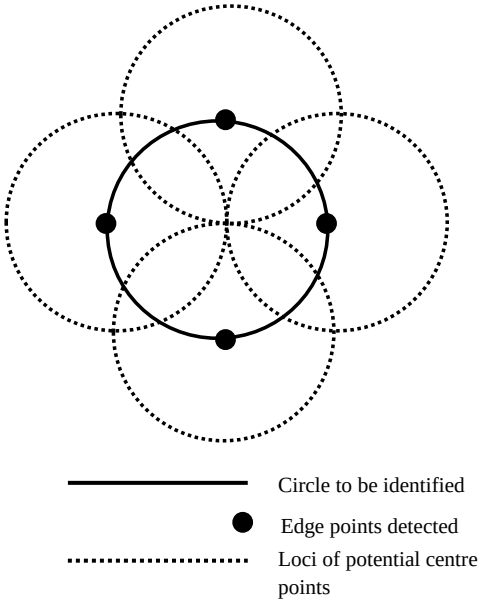


Figure 12. Circular Hough transform voting.

To reduce noise interference, all weights below a threshold are set to zero. The centre of a wheel is the centre of numerous circles, so high values should be expected at the location of a centre for a number of radii. The parameter space is 'flattened' by adding the weights recorded at each edge point from each layer corresponding to each radius considered. Within this space, peaks can then be identified.

Figure 13 shows the flattened parameter space corresponding to the image of Figure 7(a). It can be seen that a large number of potential circles are identified at the wheel regions. However, the most likely wheel location and radius must be identified by detecting the peak value in the parameter space.

Figure 13. Circular Hough transform parameter space.

The peak detection algorithm creates a list of weights above an appropriate threshold. This threshold is an input variable to the algorithm. It specifies an area the size of a wheel of the minimum radius considered around the first weight and identifies the maximum weight within that area. All other weights in the area are set to zero and the list of weights above the threshold updated. The algorithm cycles through the entire list until no further updates can be made. This is described in Table 1. By using a minimum radius search area, and only allowing one peak to be identified in this search area, only one wheel is identified in each wheel-sized area. Any remaining weights above the threshold are designated as wheels and their location noted. Good accuracy has been experienced with this method of peak detection. Figure 14 shows the location of the detected wheels for Figure 7(a). Figure 15 shows the results when applied to Figure 1. As can be seen in Figure 17, these techniques can be applied even to dark vehicles where contrast between the car and wheel is low. This is as a result of the morphological operations earlier described.

Table 1. Peak detection algorithm pseudo-code.

```
[peaksy, peaksx] = parameter_space ≥ peak_threshold

for i = 1 to length(peaksy)
   search_area = parameter_space(peaksy(i)±min(radius),
                                 peaksx(i)±min(radius))
   find max(search_area)
   parameter_space(search_area( ≠max(search_area))) = 0
   [peaksy, peaksx] = parameter_space ≥ peak_threshold
end
```

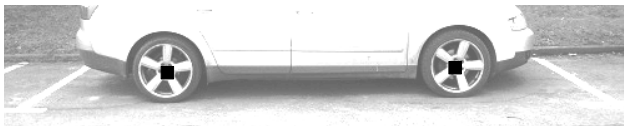


Figure 14. Detected wheels.

Figure 15. Wheel detection on HGV example.

Figure 16. Low contrast example.

3.5 Calculation of distances between objects

Detected wheels and bumpers are converted to Cartesian coordinates and the distance in pixels between the wheel peaks and bumper peaks can be calculated. An example is shown in Figure 17 for the vehicle of Figure 1.

Figure 17. Distance between HGV bumper and wheel.

This pixel distance can be related to physical space using a site-calibrated transform. Depending on the lateral position of the vehicle, it may be necessary to have a reference calibration scale in each image. Trigonometry can be used to determine the physical distances from the pixel distances, once the site arrangement, lens, and camera properties are known.

4 SUMMARY & CONCLUSIONS

4.1 Summary

This paper describes the development of an optical system to provide statistical information on the minimum gap in congested traffic. This information is critical to improve the accuracy of bridge safety assessment.

Images of vehicles are obtained from a camera with a wide-angle, aspherical lens. Morphological processing is used to reduce noise. Hough transforms are then used to segment the image and determine the location of the vehicle wheels and bumpers. This allows for the calculation of the distances required to calibrate a traffic microsimulation model.

As this research is a work in progress, results have yet to be obtained.

4.2 Conclusions

An important aspect of this work is edge detection. It has been shown that, although the Canny edge detection algorithm is generally better at detecting the edges of wheels, appropriate pre-processing and use of the Sobel edge detection algorithm produces adequate results for the present purpose.

The pre-processing undertaken prior to edge detection is sensitive to lighting conditions as appropriate pixel luminosity thresholds must be set. Once appropriate thresholds are set, good success has been shown in extracting circular edges and minimizing the number of edge points detected. It therefore represents an appropriate method for reducing the computation time for wheel detection. However, the prior establishment of the appropriate threshold is the cost of this efficiency.

Using the Hough transform to detect bumpers is successful where the bumper is near to vertical. However, the Hough transform is not suited to the detection of car bumpers, as they tend not to be vertical. Here, other means of bumper detection are required. Although computationally demanding, the circular Hough transform is accurate in the detection of wheel centres.

ACKNOWLEDGMENTS

The sponsorship of Science Foundation Ireland, under which this research is being conducted, is gratefully acknowledged.

REFERENCES

[1] European Commission (2010), *EU energy and transport in figures 2010*, Luxembourg, Publications Office of the European Union,2010.

[2] COST 345 (2004), 'Procedures Required for Assessing Highway Structures'. *Report of Working Group 1: Report on the Current Stock of Highway Structures in European Countries, the Cost of Their Replacement and the Annual Cost of Maintaining, Repairing and Renewing Them*, available online at http://cost345.zag.si/.

[3] COST 345 (2004), 'Procedures Required for Assessing Highway Structures'. *Report of Working Groups 4 and 5: Numerical Techniques for Safety and Serviceability Assessment for Highway Structures*, available online at http://cost345.zag.si/.

[4] Crespo-Minguillon, C. & Casas, J. R. (1997), 'A comprehensive traffic load model for bridge safety checking'. *Structural Safety*, 19(4), 339-359.

[5] Sustainable and Advanced Materials for Road Infrastructure (SAMARIS) (2006), 'Guidance for the Optimal Assessment of Highway Structures'. *Competitive and Sustainable Growth Programme Deliverable D30*, available online at http://samaris.zag.si/.

[6] Caprani, C.C., Lipari, A. and OBrien, E.J. (2012), 'Load effect of multi-lane traffic simulations on long-span bridges', *International Conference on Bridge Maintenance, Safety and Management*, IABMAS, Lake Como, Italy, July.

[7] Caprani, C.C. and OBrien, E.J. (2008), 'The governing form of traffic for highway bridge loading', *Proceedings of 4th Symposium on Bridge and Infrastructure Research in Ireland*, eds. E. Cannon, R. West and P. Fanning, National University of Ireland, Galway, pp. 53-60.

[8] Caprani, C. (2005), *Probabilistic analysis of highway bridge traffic loading. Ph.D., University College Dublin*, PhD Thesis, University College Dublin, Ireland.

[9] Frenze, J.F. (2002), 'A video-based method for the detection of truck axles', *National Institute for Advanced Transportation Technology, University of Idaho*, Document Number N02-05.

[10] Gonzalez, R. C., Woods, R. E. & Eddins, S. L., *Digital Image Processing Using MATLAB*, Gatesmark Publishing, USA, second edition, 2009.

[11] Ballard, D. H. (1981), 'Generalizing the Hough transform to detect arbitrary shapes'. *Pattern Recognition*, 13(2), 111-122.

[12] Fung, Y.-F., Lee, H. & Ercan, M. F. (2006), 'Image Processing Application in Toll Collection'. *IAENG International Journal of Computer Science*, 32(4).

[13] Kaiser, G., *A friendly guide to wavelets*, Birkhaeuser, Boston, 1994.

[14] Meyer, Y., *Wavelets: algorithms and applications*, Society for Industrial and Applied Mathematics, translated by Ryan, R. D., USA, 1993.

[15] Gupte S., Masoud, O., Martin, R.F.K. and Papanikolopoulos, N.P. (2002), 'Detection and Classification of Vehicles'. *IEEE Transactions on Intelligent Transportation Systems*, 3(1), 37-47.

[16] Coifman, B., Beymer, D., McLauchlan, P. and Malik, J. (1998), 'A real-time computer vision system for vehicle tracking and traffic surveillance'. *Transportation Research Part C*, 6(4), 271-288.

[17] Achler, O. & Trivedi, M. M. (2004). 'Camera Based Vehicle Detection, Tracking, and Wheel Baseline Estimation Approach', *IEEE Intelligent Transportation Systems Conference*, Washington, D.C., USA.

[18] http://www.altavision.com.br/Datasheets/IDS_EN/UI-1245LE.html

[19] http://www.theiatech.com/products.php?lens=SY125