DE GRUYTER DOI 10.1515/jisys-2013-0026 — Journal of Intelligent Systems 2013; 22(4): 417–435

Siril Yella* and Mark Dougherty

Automatically Detecting the Number of Logs on a Timber Truck

Abstract: This article describes a method of automatically detecting, counting and classifying logs on a timber truck using a photograph (taken by the driver). An image-processing algorithm is developed to process the photograph to calculate an estimate of the number of logs present and their respective diameters. The algorithm uses color information in multiple color spaces as well as geometrical operators to segment the image and extract the relevant information. This information enables the sawmill to better plan internal logistics and production in advance of the truck's arrival time. The algorithm is robust with respect to external factors such as varying lighting conditions and camera angle, but some inaccuracies remain, mainly caused by logs being occluded or covered in mud or snow.

Keywords: Log detection, color segmentation, circular object detection, forestry, Sawmill.

***Corresponding author: Siril Yella,** Department of Computer Engineering, Dalarna University, 781 70 Borlänge, Sweden, e-mail: sye@du.se
Mark Dougherty: Department of Computer Engineering, Dalarna University, 781 70 Borlänge, Sweden

1 Introduction

The work described in this article is dealt in the context of Bergkvist Insjön AB,[1] the largest privately owned sawmill in Sweden. Various issues effecting the efficiency of internal operations within the sawmill yard have been identified. The sawmill yard operations consist of unloading logs from trucks, sorting the logs, transferring to and from storage bins and other storage areas and loading logs into the sawmill itself. The timber arriving in trucks has to pass from one phase to another in an integrated supply chain fashion, where all activities are associated

1 http://www.bergkvist-insjon.se/.

with the flow and transformation of timber from raw material stage to the end use stage [7]. Given the huge production volume, it is obvious that the efficient handling of resources and avoiding production delays are high priority. A particular challenge confronting the Bergkvist-Insjön sawmill concerns the unexpected arrival of several trucks (loaded with timber of unknown quality) at the same time. This is not an issue that can be easily regulated by the sawmill because of a number of actors (such as small enterprises making delivery of timber, truck drivers). A typical routine at the Bergkvist-Insjön sawmill yard is explained as follows for the sake of clarity.

Trucks loaded with logs arrive at the sawmill, form a queue and await their turn to get unloaded. Under normal circumstances, the logs are unloaded onto a measurement buffer; they then pass through a scanner that sorts the logs into classes depending on their quality, length and diameter. On exiting the measurement station, the logs are placed in bins according to their class. An issue worth mentioning at this point concerns the limited capacity of the measurement buffer. The limited capacity of the measurement buffer makes it a bottleneck in the routine, restricting the number of logs that could be unloaded onto the buffer. In such cases, logs are unloaded onto the ground. Unloading logs onto the ground is not a preferred strategy. This is because unloading logs onto the ground implies that the same load has to be handled yet another time, increasing the handling costs. Nevertheless, logs are unloaded onto the ground on a regular basis to avoid long queues of trucks waiting to be unloaded. Another circumstance that increases handling costs is if the sorting bins become full, which can lead to production stop. As a bin approaches full capacity, a log stacker has to empty it and move the logs to another storage area. How critical this operation is depends on what quality/size of logs are entering the system and also whether another bin is currently empty and could be used as a temporary overflow.

Unexpected arrival of several trucks at the same time affects the overall efficiency at the Bergkvist-Insjön sawmill. Achieving better routines to carry out the internal operations is therefore essential. Advance information providing an estimate of the amount and the size of logs that will arrive together with an approximate arrival time would be helpful in scheduling and prioritizing operations within the sawmill yard. Having the driver photograph the rear end of the timber truck after the logs have been loaded at the roadside is a possible solution. The photograph can then be sent by wireless Internet to the sawmill, and a machine vision approach is used to estimate the number of logs present on the truck together with their respective diameters (see Section 2). The roadside location is known by GPS positioning, and thus, an estimate of arrival time can also be calculated using a navigation system.

A preliminary work reporting the detection of the number of logs on timber trucks was reported earlier by the authors [10]. Such work mainly reported the usage of color information to detect logs. Although this algorithm worked for the data set available (just 10 images), this method was found not to be robust when applied in less than ideal conditions. Typical issues are mud or snow obscuring some of the logs, poor lighting and so on. In such situations, color alone is not sufficient.

More recently, relevant work aimed at counting the number of logs was also reported by Ying and Wei [13]. Counting the number of logs was aimed at improving the production efficiency in a log yard. The aforementioned work has mainly used Otsu thresholding and distance transforms to count the number of logs in the image. Whilst this work is interesting, the authors did not make it clear how the data were collected. Moreover, it was also not clear how the approach was validated.

In the current work, the authors have tried to integrate the advantages of both of the above methods by developing an algorithm that uses both color and shape information. Further data collection was prioritized, and 38 images were acquired to create a larger data set.

The detection of logs in the current work is divided into two stages. In the first stage, color information is used to be able to segment the region containing the logs from the entire image. A *K*-means algorithm with three cluster centers is used for segmenting the logs. The shape of the logs is then further analyzed to detect and count the total number of logs present in the image. A circular Hough transform (CHT) is used to detect logs. Results achieved in this work are validated against two human observers. Such results indicate that a detection accuracy of >90% as compared with a human observer can be achieved when the logs on the truck are clearly visible.

The rest of the article is organized as follows. Section 2 provides a brief overview of the image acquisition and also outlines algorithm design. Section 3 explains log detection and the counting routine used in this work. Section 4 further presents results and discussion. The article finally presents concluding remarks.

2 Image Acquisition and Design

The first stage of any automatic vision system is the image acquisition stage. Digital images of logs on timber trucks were acquired. A Nikon DSLR D200 has been used to acquire high-quality VIS images (2896×1944 pixels; ~380–780 nm). A total of 38 images capturing only the rear end of the truck were acquired. A typical timber truck trailer contains three carriages, and it is quite difficult to acquire images of individual carriages; hence, only the rear end was photographed. Further, upon determining the number of logs present in the image, the number is multiplied by 3, and an overall sum was calculated to be the average number of logs in a truck. This obviously relies on the assumption that a given truck has a homogenous load; in general, this assumption is reasonable because the timber harvested from a particular forest site tends to be reasonably uniform in quality [7]. At this point, it is worth mentioning that the images were collected under what can be best described as “field”, as opposed to “laboratory” conditions, i.e., no control of illumination, no standard distance between the camera and the truck and so on. As an example, one might observe that in some cases the logs are clearly visible, whilst in others, they are obscured (see Figures 2–5). A collection of relevant data without any standard measures is chosen because of the application’s pragmatic usage in reality; the images are expected to be taken by the truck drivers (see Section 1). Apart from the objective in the current work, it is felt that a high-quality digital photograph of the rear end of the truck could also be interpreted manually in case the algorithm is not conclusive.

Application of appropriate image analysis techniques has been carried out using the image processing toolbox, and clustering using K-means was carried out using the neural networks toolbox in MATLAB.

3 Detecting the Number of Logs on a Timber Truck

To be able to report the log detection routine used in this work, the whole process was segmented into six stages, as described below (see Figure 1).

3.1 Preprocessing

After acquiring the necessary images, they were resized to 400×600 pixels using bilinear interpolation to reduce the computational burden for any further analysis (see Figure 2A). The lack of control over lighting resulted in low contrast in some images. The contrast in each image was enhanced by mapping the pixel intensities of the image to new values such that 1% of the data is saturated at low and high intensities [8].

Before proceeding any further, it is worth mentioning that two different human observers were asked to manually count the number of logs present in each image separately. Such observations were duly documented, for two reasons first, to be able to validate the log detection algorithm, and second, to check the

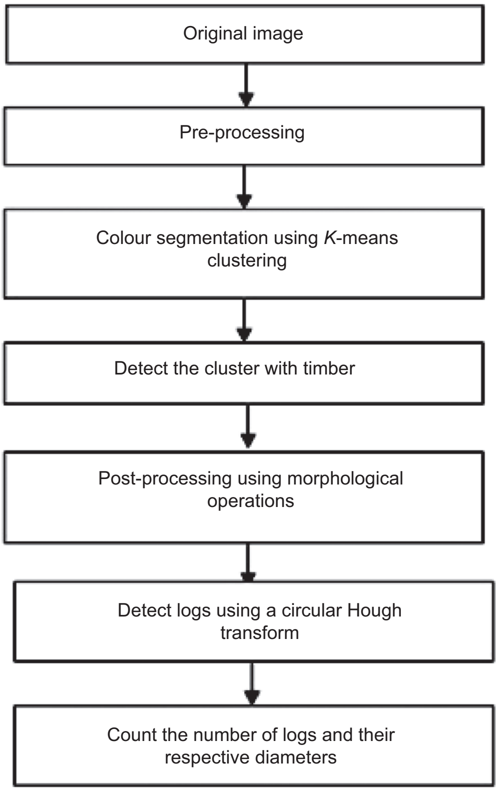

Figure 1. Design.

level of disagreement between human observers. Practical observations showed that the observers have reported different number of logs in about 35 cases out of a total of 38 cases. The prime reason for the difference in observations reported is mainly attributed to an inability to precisely decide on the presence or absence of logs. This is because when logs of varying lengths are loaded onto a truck, logs that are not relatively long enough are obscured between other longer logs. The color (due to relatively low contrast) and, to some extent, shape of such obscured logs are not very obvious, making it hard for different humans to perceive and determine their presence, thereby leading to a difference in the overall count. Bearing in mind the difference in color perception by different observers, a perceptually uniform color space was chosen. The color image (RGB) was converted into $L^*a^*b^*$ color space (more commonly known as CIELAB) for color segmentation. Another well-known disadvantage of RGB as a color space lies in the difficulty it poses whilst attempting to separate color information from intensity. Within the CIELAB space, such a separation is made possible. This is because of the fact that luminance and chromaticity (a and b) are obtained as a nonlinear mapping of the CIE XYZ coordinates. In the current case, L^* represents the luminance (lightness), and color is represented by a^* (red-green) and b^* (green-blue). The CIELAB space facilitates the identical encoding of colors when similarly perceived colors are identified. The color differences amongst various hues are also perceived uniformly, making it a robust choice [3, 8].

3.2 Color-Based Image Segmentation Using *K*-Means

As indicated earlier, color information was initially used to segment the region containing the logs from the entire image (see Section 1). Logs in the current case possessed a distinct color as compared with the remaining parts of the image and was used a feature during image segmentation, i.e., separating an image into regions based on homogeneity or heterogeneity.

A K-means clustering algorithm with three cluster centers (K=3) and Euclidean as a distance function has been used. K-means is an unsupervised learning algorithm that is iterative in nature. The K-means clustering algorithm arbitrarily assigns objects in the image to k groups (or means). Each object in the image is assigned to the nearest cluster, and the assignment is based upon the distance between the object and the cluster center. As objects are assigned, the cluster center is recomputed as the average of the points in that cluster. This procedure is repeated until the clusters converge. The K-means clustering algorithm is simple to implement and is able to cluster the image without having any idea about the relationships between the various objects. A slight disadvantage of K-means lies in the fact that the algorithm is unable to determine the number of cluster centers. Hence, the performance of the K-means clustering algorithm mainly depends on the number of clusters supplied as input by the user. The choice of K in the current work is based on the visual analysis where objects in the image can be grouped into three categories, namely background, logs and other remaining objects in the image. A complete discussion concerning K-means clustering algorithm is out of the scope of this article but could be founds elsewhere [2, 4, 6 and 12].

3.3 Detecting the Cluster Containing the Logs

Once an image is segmented, all the different objects in the image are assigned an index corresponding to the cluster, i.e., each cluster is a separate image containing different objects. As a result, the logs could be found in one of the clusters. However, often, one does not know which cluster contains the logs. This is because of the dynamic nature of the K-means clustering algorithm; which tends to allocate objects to different clusters every time, making it very hard to automatically determine the cluster containing the logs. The clusters are converted back into RGB color space for further processing, and all the images are appropriately analyzed (see Section 3.2) as follows. Initially, the images were converted into the hue, saturation and value (HSV) color space. The fact that the HSV color space allows usage of color descriptions (adding white or black to a certain color to produce different tints and shades) as opposed to RGB's usage of color primaries makes it a suitable choice. In the HSV color space, segmentation is normally performed on the hue component and is mainly aimed at distinguishing objects with different hues. The fact that hue is not entirely reliable for discrimination when saturation is low was also taken into account. The hue and saturation channels of all the three images were processed separately. Initially, holes in each image were filled morphologically to create more uniform regions. Further, the standard deviation of hue and saturation channels were calculated and compared separately for every image. After a reasonable amount of experimentation, it has been found that the image with the relatively low standard deviation in the hue channel contained the logs. Further, it was observed that the same image exhibited relatively a high standard deviation in the saturation channel. This is because the color distribution amongst the logs exhibited a relatively low degree of variance when compared with other image regions. The logs were relatively darker (reddish colored) compared with other lighter regions such as the sky. Finally, the cluster containing the logs was selected for further processing (see Figure 2B).

3.4 Postprocessing

Upon investigating the result of clustering, one can observe some residual irrelevant objects (see Figure 2B), i.e., timber lying on the ground, etc. The fact that the timber lying on the ground exhibits the same color as the logs resulted in the inclusion of such irrelevant objects. Hence, a postprocessing routine comprising of a series of morphological operations was used to eliminate all such irrelevant objects.

In the first stage, objects that are smaller than 150 pixels were removed. Such objects are often timber lying on the ground and are seldom logs on the truck (see Figure 2C). In the second stage, the morphological operation of erosion was applied. The erosion operation changes any white pixel adjacent to a black pixel to black, i.e., pixels are removed from the boundaries of objects in an image. The extent to which pixels are removed from boundaries of object depends on the shape and size of the structuring element that is being used in carrying out the morphological operations. Typical shapes of structuring elements used to carry out morphological operations include ball, disk, square, line, etc. [3, 8]. A disk-shaped structuring element 4 pixels in diameter was used in the current case to further reduce the presence of unnecessary objects (see Figure 2D).

The result of erosion in the above stage has affected the structural integrity of the logs. Hence, in the third stage, the morphological operation dilation was applied. The dilation operation changes any black pixels adjacent to a white pixel to white, i.e., pixels are added to the boundaries of objects in an image [3, 8]. Dilation in the current case has been carried out by using a disk-shaped structuring element 8 pixels in diameter to preserve the necessary details (see Figure 2E).

In the fourth stage, the connected components in the binary image were computed to be able to further study regional properties of the remaining objects. A component-labeling algorithm finds all connected components in an image and assigns a unique level to all points in the same component. Points in a connected component form a candidate region for representing an object. Labeling could be done in either 4 or 8, where 4 specifies 4-connected objects and 8 specifies 8-connected objects (i.e., when connections are also possible diagonally). The elements of the label are integer values ≥ 0. The pixels labeled “0” are the background. The pixels labeled “1” make up one object, the pixels labeled “2” make up a second object and so on [3, 8]. At this point, the area of all the remaining objects in the labeled image are computed, and only the object with the relatively largest area, logs in the current case, is retained for further processing. Finally, the resultant image is used as a mask and multiplied with the image in stage 1 to be able to acquire a region of interest that is more subtle and suitable for further processing (see Figure 2F).

3.5 Detecting Logs Using CHT

At this stage, the horizontal and vertical gradient images were computed on the region of interest. Next, the gradient magnitude was computed, and a thresholding on the gradient magnitude of the image was performed, i.e., pixels with gradient magnitudes smaller than the input threshold were removed from further computation. A gradient threshold of 14 pixels has been in the current case. Further, a CHT algorithm was used with the aim of detecting circles (logs in the current case). CHT proposed by Peng [9] was used in the current work. The fact that the above-mentioned CHT algorithm is able to detect circles with multiple radii makes it a suitable choice where the radii of the logs vary greatly. CHT is an extension of the standard Hough transform (HT), which in itself has been heavily used for the detection of parametric lines in images. HT implements a voting process that maps image edge points into manifolds in an appropriately defined parameter space. The peaks in the space correspond to the parameters of detected lines [9, 11]. The CHT algorithm uses the following parametric representation for the identification of circles:

$$x = x_0 + r * \cos(\theta)$$
$$y = y_0 + r * \sin(\theta), \qquad (1)$$

where (x_0, y_0) are the coordinates of the center of the circle in the X and Y directions and r is the radius. Detection of circles using a CHT is based on the idea that perpendiculars to the edge points of a circle cross in the center of the circle. Circles are drawn over various edge points, and their respective coordinates are accumulated (and stored in an accumulator array). The number of circles that pass through coordinates of each edge point is noted. Note that the accumulator is a three-dimensional Hough array where the first two dimensions represent the coordinates of a circle (x_0, y_0), and the third dimension represents the radii r. To determine a circle, votes in a circular Hough space are accumulated and analyzed to estimate the three parameters of a circle, and the coordinates with the highest count are finally selected.

Bearing in mind the computational complexity involved in dealing with three-dimensional parameter spaces, the radii of detection were set within a range. After a reasonable amount of experimentation, 6 and 26 were chosen as the minimum and maximum radii. A complete discussion concerning HT and CHT is out of the scope of this article but could be found elsewhere [1, 5, 9, 11]. At this point, all the logs in the image are detected; however, further processing is necessary before finally marking the logs (see Figure 2G).

3.6 Counting the Number of Logs

As indicated above, circles are marked on top of objects that are circular in shape – usually logs (see Section 3.5). However, in some cases, circles were marked on objects that are not logs (see Figure 2G). As an example, Figure 2G contains three circles numbered 3, 68 and 72. Upon carefully observing the picture, one could notice the absence of logs within the aforementioned circular regions detected by the algorithm. This is because in the current case CHT produces false positives by accumulating votes that are consistent in location but inconsistent in other properties such as color and shape. As a result, most of the extra (and usually empty) circles are spatially close to the logs. Therefore, after finding the relevant coordinates and their radii, a search in the (x_0, y_0, r) space was conducted to be able to reduce the number of false-positive circles. Often, false positives are detected due to the presence of circular arcs belonging to the walls of the logs and other irrelevant objects present in the image and are seldom logs. Hence, the neighborhood of every circle was individually processed, and their mean color value (in CIELAB space) was computed. After reasonable amount of experimentation, it was observed that the circled regions containing logs had a mean color value of >75. Circles with a mean color value of <75 were excluded from further processing. The remaining logs were relabelled, and their respective radii were updated. Finally, the number of logs and their respective radii are reported for further decision making (see Figure 2H).

4 Results and Discussion

The results of the new hybrid machine vision algorithm for detecting logs on timber trucks are promising.

As indicated earlier, two different human observers were asked to manually count the number of logs present in each image separately (see Section 3.1). A difference of opinion in counting the number of logs in each image was observed in 35 instances out of a total 38 instances. Bearing in mind the difference in opinion, the numbers of logs reported by the observers were compared against the number reported by the machine vision algorithm separately to be able to validate the results. Such results have been tabulated to promote the readability of the article (see Table 1). Results reported by the machine vision algorithm individually were further grouped and organized into four classes on the basis of detection accuracy (see Table 2). The fact that the results are relatively comparable to each other also justifies grouping. A few example images (one per class) were selected to demonstrate log detection accuracy (see Figures 2–5). The example images are linked to their corresponding image numbers in Table 1 for the sake of clarity (see Table 3). Further, the mean and standard deviation of detection accuracies of the images in each class are also presented (see Figure 6) to help understand the intraclass differences. Upon observation, one could notice that all except one class exhibited relatively low standard deviation, i.e., relative differences in detection accuracies between the images are low. By comparison, the class where the detection accuracy is <70% has indicated a high standard deviation. A deeper examination

Table 1. Results Achieved by the Machine Vision Algorithm.

Image	Number of logs reported by the first-time observer	Number of logs reported by the second-time observer	% of logs agreed by both the observers	Number of logs detected using machine vision	% of logs accurately detected when compared with the first observer	% of logs accurately detected when compared with the second observer
1.jpg	87	90	96.66	80	91.95	88.88
2.jpg	70	59	98.57	34	48.57	49.27
3.jpg	73	76	96.05	51	69.86	67.10
4.jpg	79	80	98.75	55	69.62	68.75
5.jpg	76	79	96.20	60	78.9	75.94
6.jpg	59	60	98.33	56	94.91	93.33
7.jpg	52	90	91.11	45	54.87	50.00
8.jpg	96	97	98.96	82	85.41	84.53
9.jpg	69	69	100	59	85.50	85.50
10.jpg	62	62	96.87	61	98.38	95.31
11.jpg	59	60	98.33	51	86.44	85.00
12.jpg	63	65	96.92	60	95.23	92.30
13.jpg	51	55	92.72	46	90.19	83.63
14.jpg	61	61	100	60	98.36	98.36
15.jpg	45	45	100	40	88.88	88.88
16.jpg	79	80	98.75	72	91.13	90.00
17.jpg	65	73	89.04	35	53.84	47.94
18.jpg	59	62	95.16	37	62.71	59.67
19.jpg	64	65	98.46	63	98.43	96.92
20.jpg	51	60	85.00	48	94.11	80.00
21.jpg	46	47	97.87	39	84.78	82.97
22.jpg	64	66	96.96	60	93.75	90.90
23.jpg	69	77	89.61	52	75.36	67.53
24.jpg	60	62	96.77	52	86.66	83.87
25.jpg	45	47	95.74	22	48.8	46.80
26.jpg	61	54	88.52	49	80.32	90.74
27.jpg	76	74	97.36	70	92.10	94.59
28.jpg	90	91	98.90	74	82.22	81.31
29.jpg	56	60	93.33	43	76.78	71.66
30.jpg	74	84	88.09	59	79.72	70.23
31.jpg	98	102	96.07	74	75.51	72.54
32.jpg	61	62	98.38	54	88.52	87.09
33.jpg	52	53	98.11	48	92.30	90.56
34.jpg	62	71	87.32	50	80.64	70.42
35.jpg	52	58	89.65	31	59.61	53.44
36.jpg	82	83	98.79	77	93.90	92.77
37.jpg	45	46	97.82	34	75.55	73.91
38.jpg	72	74	97.29	62	86.11	83.78

Table 2. Individual Result of Detection Grouped into Four Classes.

Class	Number of samples in each class when compared with the first observer	Number of samples in each class when compared with the second observer
Detection accuracy >90%	13	11
Detection accuracy between 80% and 90%	11	12
Detection accuracy between 70% and 80%	6	6
Detection accuracy <70%	8	9
Total	38	38

of the images in each class has helped understand the reasons where the performance deteriorates. The two main reasons behind the deterioration in performance were identified and discussed as follows.

First, poor detection rates were observed in cases where some of the logs in the image were obscured. Log obscurity was also the main cause behind the disagreement between the observers. Disagreements were largely caused when relatively short logs were obscured between their longer counterparts, making it very hard for a human being to determine their presence. The fact that an ordinary timber truck is loaded with logs of uneven lengths coupled with improper camera position and lack of controlled lightning conditions made such a situation unavoidable (see Section 2). Acquiring an image that captures a good view of most of the logs on the truck demands that the camera is positioned high (relative to the height of the truck) above the ground. The absence of any such care whilst acquiring images has obscured some logs often appearing as gaps (dark regions) in the image (see Figures 2, 5 and 6). It is worth mentioning that image acquisition was carried out intentionally in an uncontrolled manner to be able to report a robust algorithm that will work in reality and where the truck drivers will photograph the rear end of their trucks using their mobile phones (see Section 1).

Second, the occlusion of logs has also contributed to poor detection rates. Figure 5 presents an example of an image where several logs are covered in snow. Snow on the logs was a result of travel on the local roads during the winter months in Sweden. Logs covered in snow have presented a major challenge in the current case. This is due to the effect of color segmentation where logs covered in snow were allocated to the same cluster as the background due to the similarity in color. Logs covered in snow were thus excluded from the region of interest during the color segmentation phase, which has led to a substantial drop in detection rates. Although the algorithm reports failure in cases where logs are covered in snow, it is not particularly seen as a weakness in the current case. This is because,

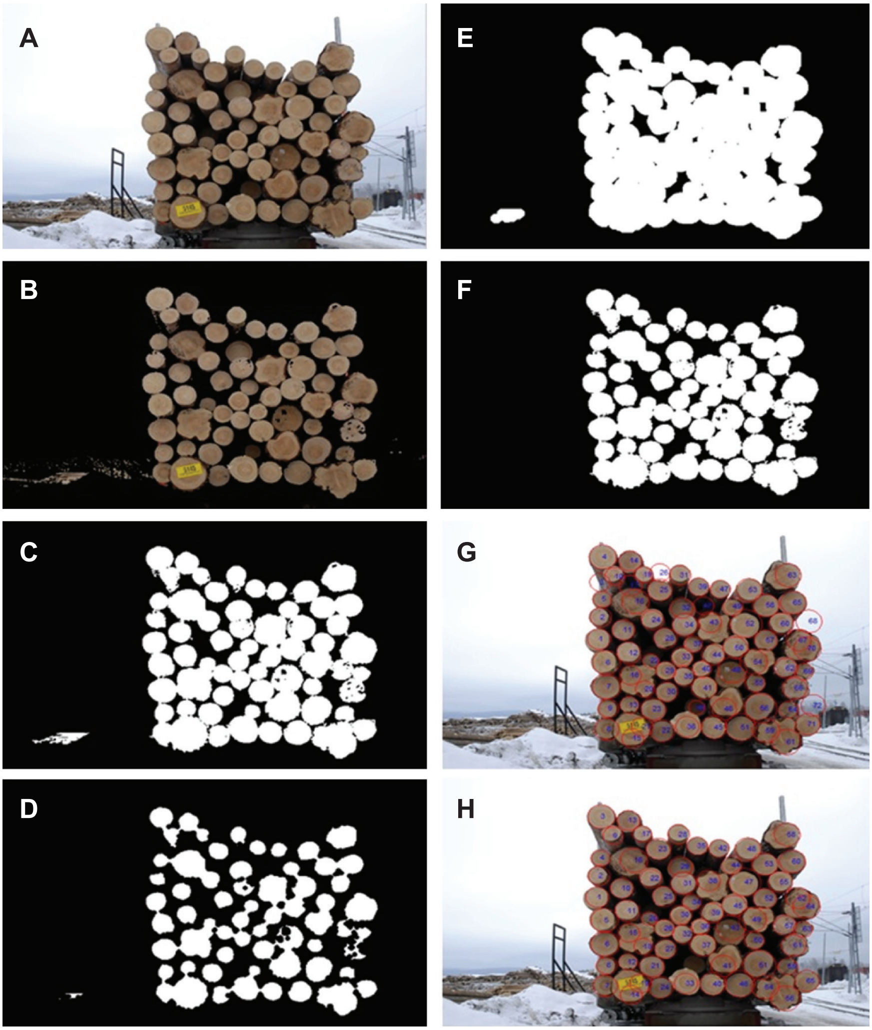

Figure 2. Various Stages of Processing Aimed at Detecting and Counting the Logs. Notice that the Log Detection Accuracy is >90%.

in reality, images of the rear end of the truck would be acquired on site before departing from the harvesting site, and logs at the harvesting site are mostly clean, especially when the loading process is just completed. The fact that the algorithm exhibits robust performance when the logs are clean confirms its practical utility. Limitation in the number of images we could collect at the roadside just after loading is impractical within a small-scale project. Hence, in the current case, images were acquired after the trucks have arrived at the sawmill. Acquiring

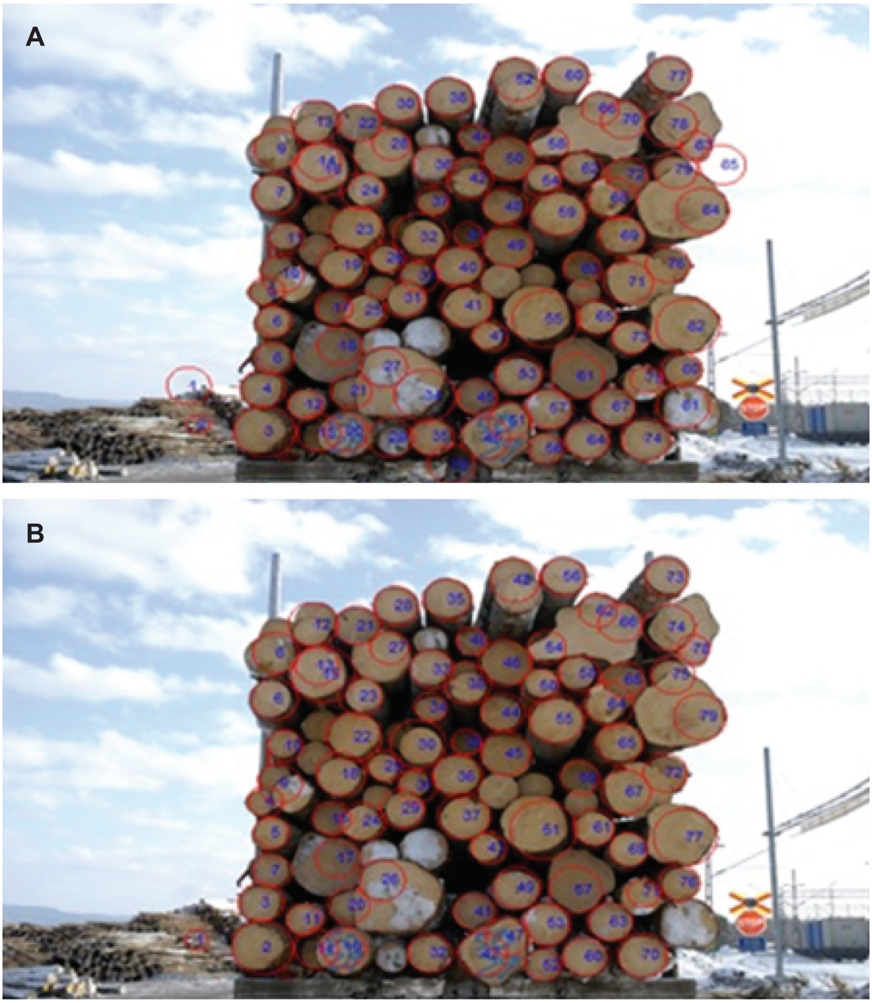

Figure 3. Image Where Log Detection Accuracy is Between 80% and 90%.

images at the sawmill has facilitated the collection of a heterogeneous (timber from different harvests, sizes, etc.) image set without much inconvenience and has thus allowed rigorous testing.

Apart from the presence of snow, a few other less significant factors have been identified. Batch labels marking the type of the harvest pasted directly onto the logs, remains of tree branches, etc. are typical examples of such reasons.

As indicated earlier, the CHT algorithm is dependent on a predefined range of the circle radius for detection. The choice of radii is derived through

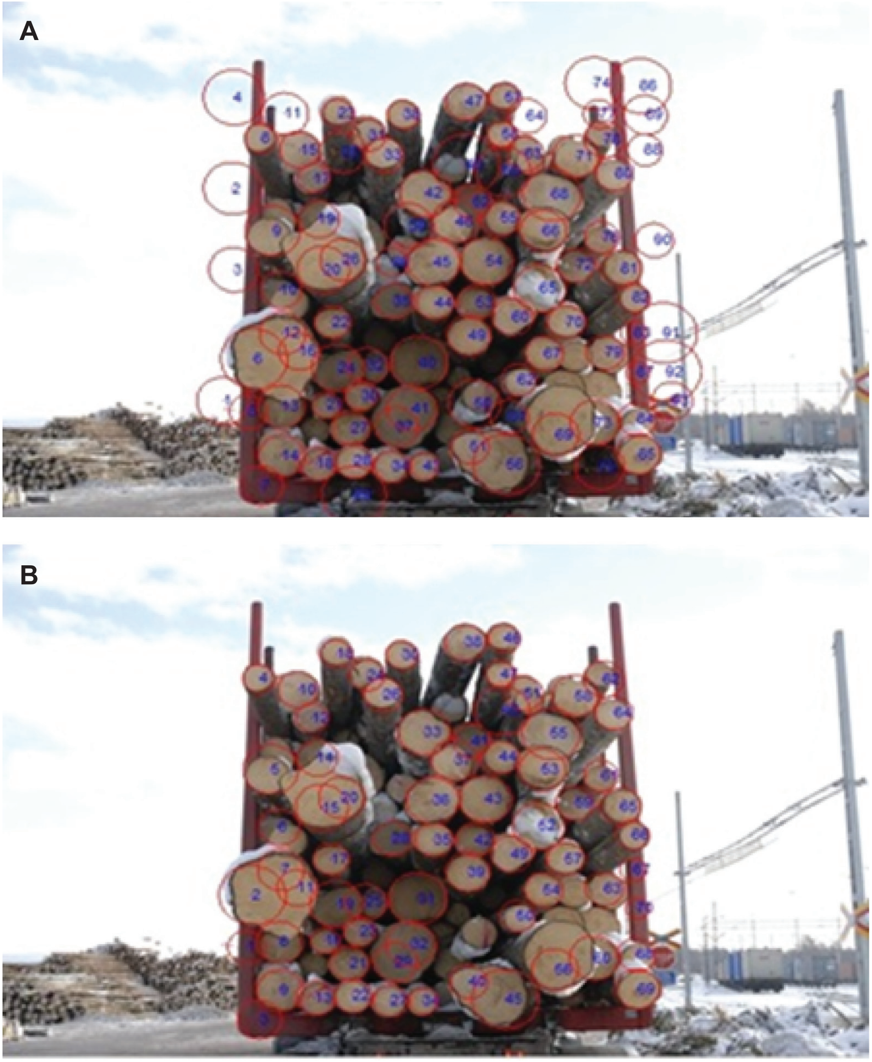

Figure 4. Image Where Log Detection Accuracy is Between 70% and 80%.

experimentation and is mainly intended to reduce the computational burden whilst executing the CHT (see Section 3.5). The downside of such an approach is that logs whose diameters fall outside the range are sometimes excluded from detection. However, this was not perceived as a severe problem because, in most cases, the CHT algorithm has detected an out-of-range circle using two or more smaller circles, resulting in a situation where the same log is detected more than

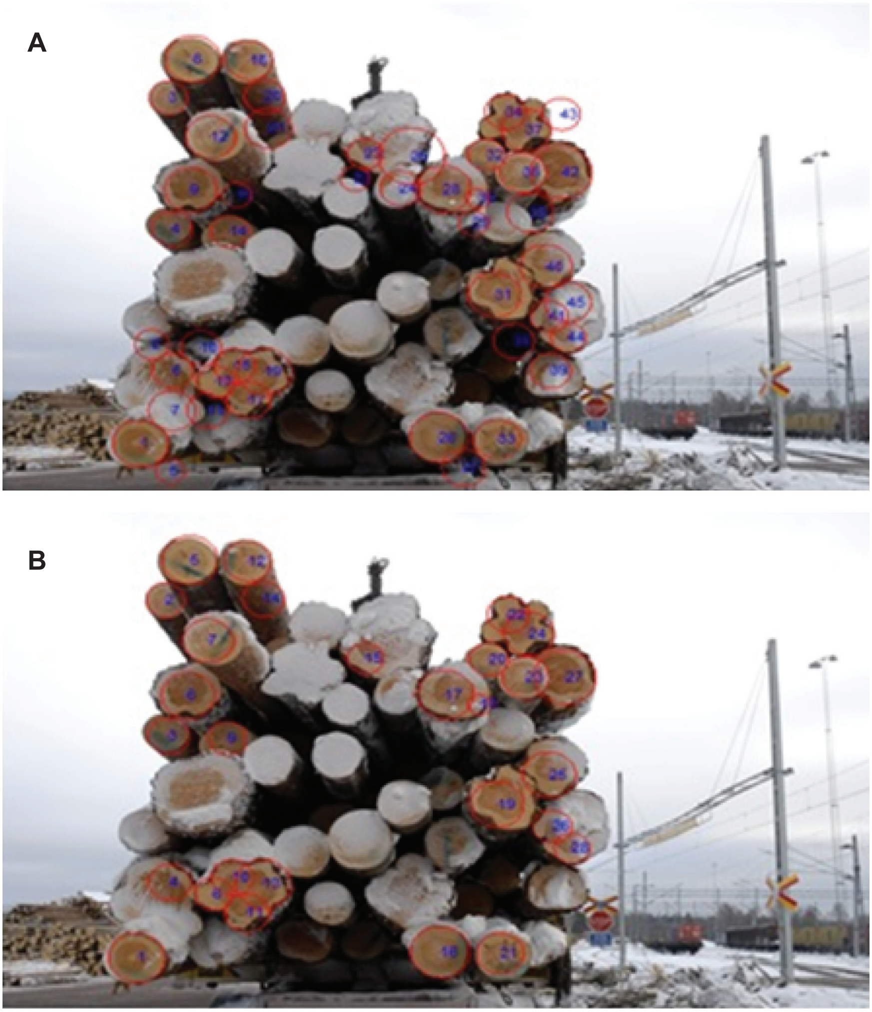

Figure 5. Image Where Log Detection Accuracy is <70%.

Table 3. Example Images and their Corresponding Classes.

Figure title	Corresponding image in Table 1	Class
Figure 2	10.jpg	Accuracy >90%
Figure 3	8.jpg	Accuracy between 80% and 90%
Figure 4	30.jpg	Accuracy between 70% and 80%
Figure 5	25.jpg	Accuracy <70%

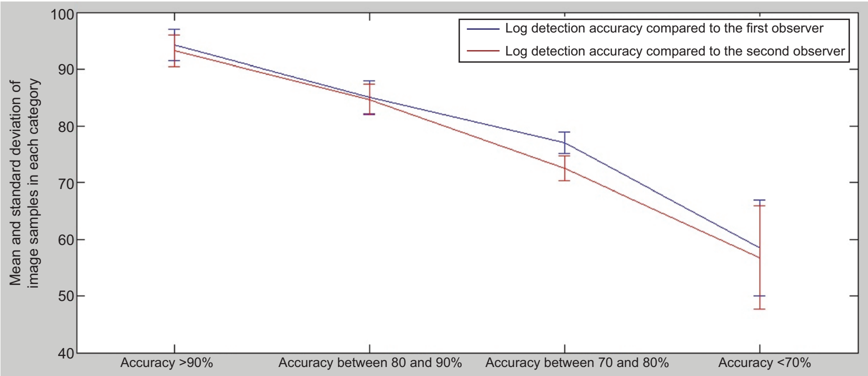

Figure 6. Mean and Standard Deviation of the Image Samples Grouped into Four Classes.

once (see Figures 2–4). Further, the irregular shape of the logs was yet another issue (see Figures 2–6). It has been experimentally observed that logs with irregular shapes contained more than one circular edge, thereby leading to a situation where one single log is detected more than once. A routine aimed at handling the over detection of logs has already been explained earlier (see Section 3.6). An excess detection of (the same) logs for reasons explained above could not be dealt with the routine explained in Section 3.6. Circles marked on the same log exhibit similar color properties, and hence, color information is deemed unsuitable for distinguishing circles marked on the same log. Other features such as circle proximity, overlap, etc. were investigated but were not robust enough in actually achieving any distinction. Hence, excess detection of logs in this stage was handled by manual intervention. As an example, Figure 2 presents a case where the machine vision algorithm has marked 65 circles within the image. Upon carefully looking at Figure 2, one may observe that four logs have been marked using more than one circle. Cases as such were manually observed, and the number of logs that were marked more than once was removed from the total number before presenting the final count to avoid ambiguity in the final number. As a result, one may observe a slight mismatch in the number reported in the table as opposed to the number of logs marked within the image.

5 Conclusions and Future Work

The current article has presented a promising approach for segmenting and counting the number of logs and their respective diameters on board timber trucks. The overall aim is to help plan and optimize work routines inside a large privately owned sawmill in Sweden by providing information about the load and approximate arrival times of timber trucks before their arrival. Images of the rear end of the timber trucks were photographed for the purpose. A total of 38 images under different conditions were acquired to verify proof of concept. The images were processed appropriately using relevant techniques to be able to report the number of logs present on the truck. Results achieved in the current work have shown that the use of image data for detecting and counting logs is indeed possible and that the results are very reliable and accurate with a clear view of the logs in the photograph. Such results were compared with that of human observers for the sake of validation.

Certain specific situations where the algorithm fails have been identified, and it is hoped that development of the technique can further improve performance. In the future, it is recommended that images of the rear end of the trailer are acquired immediately after loading the timber on to the truck as opposed to the current work where the images were acquired after the trucks have arrived at the sawmill. The problem of calculating the approximate arrival time has not been addressed and solved in this work. Hence, it would be interesting to extend the work further using the geo-tagged images to work out the approximate arrival time of the trucks. It is also proposed that the algorithm be developed as a smartphone application to realize its pragmatic applicability. Finally, it would be interesting to customize the reported algorithm to solve problems within other immediately relevant areas such as forestry, etc.

Acknowledgments: The authors wish to thank Shaik Asif Rahman, a research assistant at the Dalarna University, and other members of the staff at Bergkvist-Insjön AB for providing substantive support whilst collecting the data necessary to carry out this work.

Received May 6, 2013; previously published online June 21, 2013.

Bibliography

[1] R. O. Duda and P. E. Hart, Use of the hough transformation to detect lines and curves in pictures, *Commun. ACM* **15** (1972), 11–15.
[2] R. O. Duda and P. E. Hart, *Pattern Classification*, 2nd ed., Wiley, New York, 2001.
[3] R. C. Gonzalez and R. E. Woods, *Digital Image Processing*, 3rd ed., Prentice Hall, New Jersey, 2007.
[4] J. A. Hartigan and M. A. Wong, Algorithm AS136: a K-means clustering algorithm, *Appl. Stat.* **28** (1979), 100–108.

[5] P. V. C. Hough, Methods and means for recognising complex patterns, US Patent 3, 069; 654, 1962.
[6] R. Jain, R. Kasturi and B. G. Schunck, *Machine Vision*, McGraw-Hill, New York, 1995.
[7] C. G. Lundahl, *Optimised processes in sawmills*, licentiate thesis, Luleå Tekniska Universitet, 2007.
[8] MATLAB, *Image Processing Tool Box*. www.mathworks.se/. Accessed January 20, 2013.
[9] T. Peng, Circular Hough transform. www.mathworks.com/matlabcentral/fileexchange/9168-detect-circles-with-various-radii-in-grayscale-image-via-hough-transform. Accessed January 20, 2013.
[10] S. A. Rahman, S. Yella and M. Dougherty, Image processing technique to count the number of logs in a timber truck, in: Proceedings of the IASTED Conference on Signal and Image Processing, USA, 2011.
[11] M. Smereka, I. Duleba, Circular object detection using a modified Hough transform, *Int. J. Appl. Math Comput. Sci.* 18 (2008), 85–91.
[12] R. H. Turi, Clustering-based color image segmentation, Ph.D. thesis, Monash University, Australia, 2001.
[13] X. Ying and X. Wei, Application of some valid methods in logs counting system based on digital image processing, in: IEEE 17th International Conference on Industrial Engineering and Engineering Management, Xiamen, China, 2010.