Presented at the 4th DLSU Innovation and Technology Fair 2016
De La Salle University, Manila, Philippines
November 24 & 25, 2016

Philippine Bus Number Location Identification

Jedidiah L. Caychingco
College of Computer Studies
De La Salle University
j.caychingco@yahoo.com

Abstract: In this paper, the author discusses a way to isolate the area containing the bus number from an image of a bus using edge detection, morphological operations, and image segmentation. The training set consists of 15 images of Jac Liner buses that are tilted by 30 degrees at most, and contains 3 to 4-digit bus numbers. The images are of different sizes: the smallest being 500x375 pixels while the largest is 1024x768 pixels. Furthermore, the images contain ample illumination. The created program produced an accuracy of 20%.

Key Words: Bus number identification; Sobel edge detection; morphological operations; image segmentation; variable-sized input.

1. INTRODUCTION

There are several hurdles we need to overcome to achieve a green society. It is the author's desire to help further this cause. However, we heavily rely on a substance that is one of the leading causes of the destruction of our planet: greenhouse gases. This paper aims to reduce the greenhouse gases produced by buses by tackling the problem of smoke belching.

The Environmental Management Bureau (n.d.) is responsible for monitoring the pollution produced. Manually keeping track of problematic vehicles is a tedious process and there are also times wherein no one is available to monitor them. Therefore, there is a need to streamline this process and make it easier for the authorities.

To the author's knowledge there are no studies concerning themselves with locating the bus number. Therefore, this paper presents a novel approach in that area. To aid in this research, the author made use of the abundance of literature present concerning the identification of a vehicle by its license plate(LP) and adapted their use to suit the task at hand.

Bus numbers are an effective medium to identify buses due to their uniqueness. However, it is also a challenging task to achieve due to variations in: illumination, angles, obstructions, background, and other factors. The task's difficulty is further compounded due to an absence of a standard for bus numbers. To standardize the data, the sample used consists only of Jac Liner buses.

The following sections of the paper are organized as follows: Section 2 presents the review of related literature, Section 3 describes the algorithm used, Section 4 shows the results, Section 5 is where the author details the conclusions and his recommendations to improve the accuracy of the program, Section 6 contains the acknowledgements, and Section 7 contains the sources used.

2. REVIEW OF RELATED LITERATURE

To filter out the unneeded information from the input image, Rashid, A.E. (2013) made use of the vertical Sobel edge operator to get the edges of the image and a threshold to only get the long edges. He noted that the long edges contain relevant information about the image such as the dimensions of the bus, and are a different entity from the image's background. Afterwards, he created a convex hull encompassing the long edges, and thereby encompassing the bus to an extent, and removed the area outside it, thereby removing the small edges. His rationale with removing the small edges is that they correspond to elements in the background.

Sun, M., Qiu, Y., & Zhou, W. (2009) made use of vertical edge detection combined with morphological operations and region of interest analysis to get the location of the LP of a car.

To extract the edges of an image without biases for angles, Chen, R., & Luo, Y. (2012) created an improved Prewitt operator. They concluded that the location of a vehicle's LP in the image contains more texture information compared to other parts of the image. With this, they used horizontal and vertical projection to determine the areas containing the greatest concentration of textures. However, this operation results in multiple candidate areas. To remedy this, they made use of a height to width ratio so that only the candidate areas that pass this requirement are accepted.

Rabee, A., & Barhumi, I. (2014) used morphological operations and the machine learning classifier Support Vector Machines (SVM) to locate the LP and identify its contents. His approach combines rule-based methods to get the LP's location and the SVM to identify the characters in the LP.

3. ALGORITHM DESCRIPTION

The program made use of a variety of methods to process its input. In order, they are: grayscale conversion, area of interest isolation, image pre-processing, and bus number candidate selection.

3.1. *Grayscale conversion*

A colored image is in the RGB (red, green, blue) format wherein each pixel is comprised of red, green, and blue components at different intensities. In order to get the image's luminance level, or its grayscale equivalent, the RGB value in each pixel must be multiplied by a certain value as specified by equation 1. Chen, R., & Luo, Y. (2012) noted that processing images in grayscale will result in faster processing and offers a uniform range of values. These steps can be seen in Fig. 2 to 3. Each pixel in the grayscale image will then have a value of 0 to 255 wherein 0 represents the color black, 255 represents the color white, and the values between them represent different shades of gray.

$$I_{(x,y)} = 0.2989*R + 0.587*G + 0.114*B \qquad \text{(Eq. 1)}$$

Where:

$I_{(x,y)}$ = The pixel located at row x and column y
R = The value of the red pixel of $I_{(x,y)}$
G = The value of the green pixel of $I_{(x,y)}$
B = The value of the blue pixel of $I_{(x,y)}$

3.2. *Area of interest isolation*

After converting the image into grayscale, its irrelevant parts are removed. Irrelevant parts are defined to be the background of the bus as well as the upper half of the bus.

In order to separate the bus from its background, the grayscale image is first binarized, a process that causes each pixel in the image to be either black or white, denoted by 0 or 1 respectively. The default threshold of 0.5 is kept for the binarization process. This states that pixels whose value is less than the half way point between black and white are turned to black while those whose values fall above the half way point are turned to white. Specifically, if a pixel in the grayscale image has a value within 0 to 127, it is turned to 0; while values within 128 to 255 are turned to 1.

The vertical Sobel edge operator in Fig. 1 is then applied to the image. The process results in Fig. 4.

Afterwards, the small edges are removed. This is what Rashid, A.E. (2013) has done. Small edges are defined to be the edges whose area in pixels is less than 1/8 of the image's height. This step is performed so that the location of the bus can be approximated since the edges in the background are normally smaller than the that of the bus and correspond to objects such as windows, pedestrians, and other objects. The output is shown in Fig. 5. Ideally, this step will result in the outline of the bus. But it is also possible for some long-edged items in the background to remain such as lamp posts and buildings.

A convex hull encompassing the remaining long edges is then created and is shown in Fig. 6. Afterwards, areas outside of the convex hull are removed and the resulting image is multiplied with the greyscale image Fig 7.

Rashid, A.E. (2013) further noted that to further isolate a car's LP, the top half of the car is removed since the LP is found at the lower half of the car. The same logic is applied in this case to locate the bus number since it is located on the lower half of the bus. The resulting image is found in Fig. 8.

1	2	1
0	0	0
-1	-2	-1

Fig. 1 The vertical Sobel edge operator

3.3. Image preprocessing

The code by Qadir, R. (2013) was applied to further preprocess the image: the morphological operations opening and dilation is then applied to the image so that the areas of its components can be increased. The output is shown in Fig. 9. As a disclaimer, his code is suboptimal and contains hard-coded variables.

Areas in the image that are enclosed in borders are then filled in as was performed by Chen, R., & Luo, Y. (2012). This results in the blobs in Fig. 10. The blobs that cannot be filled in are removed since they correspond to non-enclosed portions and are thus, not candidates for bus numbers. The remaining blobs are then thinned. Ideally at this stage, the bus number will appear to be a blob. However, many other areas in the image will look like blobs as well.

3.4. Bus number candidate selection

To disambiguate the location of the blob containing the bus number compared to the other blobs, we will remove candidates whose dimensions are not that of a bus number. This was step was also performed by Rabee, A., & Barhumi, I. (2014). The result is shown in Fig. 11. Their rules for determining the eligibility of an area are rules 1 to 3 listed below. Rule 4 is an additional rule created by the author to further aid in isolating the bus number.

1. A width to height ratio of 1.5 to 2.5
2. A minimum width of 30 pixels
3. A minimum height of 11 pixels
4. Number of segments is within 4 to 9

However, the author opted to not use rules 2 and 3 because they are not effective when used with images with different resolutions.

The current image containing the remaining candidate areas are then multiplied with the initial scale image to give content to the blobs. This is shown in Fig. 12.

The processes detailed above have significantly reduced the number of candidates since it removes areas such as: the bus body's design, bus name, wheels, and other elements. However, there are still incorrectly identified areas present.

In order to rectify this, we made use of rule 4 that requires the candidate area to have a specific number of segments. A segment is defined to be an enclosed region within the image. To count the number of segments a candidate region has, it is once again binarized with the same threshold of 0.5. The rationale for the specified segment count is that it encompasses the required segments for a 3 to 4-digit bus number. Specifically, the candidate region's bounding box corresponds to a segment; the number 8 contains two segments since the program identifies it as two circles put together, one segment for its top circle, and another for its bottom half; and the other numbers have only one segment. Thus, the lowest possible segment count of a 3-digit bus number would contain 4 segments: one for the bounding box, and three for digits that are not 8. The highest segment count of a 4-digit bus number would be 9: one segment for the bounding box, and eight for four instances of the number 8.

A problem with this approach is that one to two-digit bus numbers cannot be captured by the program since they will not pass the width to height ratio required to be considered as a license plate due to their smaller width when compared to their three to four-digit counterparts. Moreover, if the width to height ratio requirement was removed, the program will result in more false positives since it will identify areas that do not have the dimensions of a bus number to be an area containing a bus number.

Fig. 2. RGB image

Fig. 3. Gray scale image

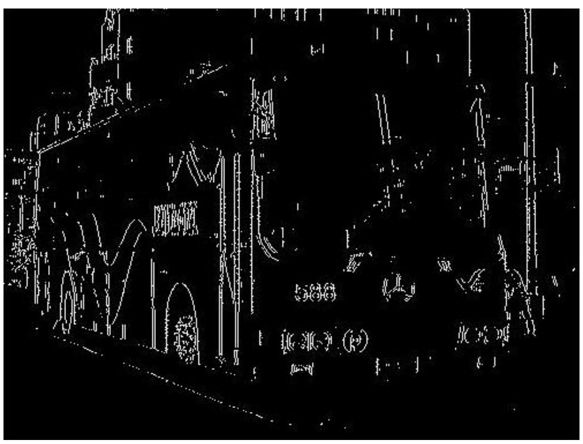

Fig. 4. After applying the vertical Sobel edge operator

Fig. 5. Removed the small edges

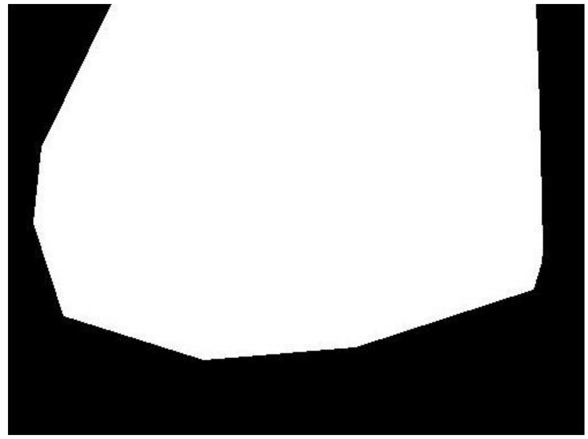

Fig. 6. The convex hull encompassing the long edges

Fig. 7. Removed the area outside of the convex hull. and multiplied it with the gray scale image.

Fig. 8. Area of interest

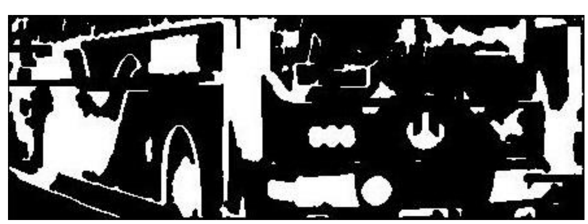

Fig. 9. Blobs

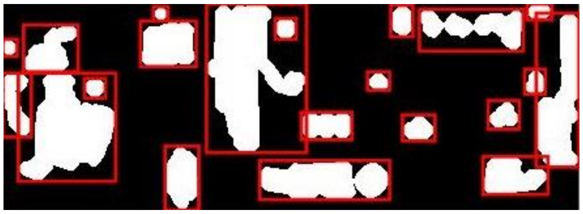

Fig. 10. Candidate regions blobs

Fig. 11. Blobs with bus number-like dimensions

Fig 12. Multiplied image

Fig. 13. Segment count filtered candidate region

Fig. 14. Final output

4. RESULTS

Among the 15 test images, only three(3) were successfully captured by the program. This results in an accuracy rating of 20%. There also exists six(6) separate instances that successfully capture the bus number but unfortunately also captures other invalid regions as is shown in Fig. 15.

Fig. 15. The bus number with invalid results.

5. CONCLUSIONS AND RECOMMENDATIONS

An interesting thing to note with the valid results is that one(1) of them contains a bus that was taken from its side, whereas the other images were taken mostly from their front. Aside from this, there are little similarities among the three(3) valid results that will allow them to stand out from the rest. Therefore, it is likely that it is only by chance in which the accuracy was obtained.

In order to minimize occurrences that non-bus number areas are obtained such as is shown in Fig. 15, additional pre-processing must be done. However, this should be done with care as to not overfit the program to the dataset given. Overfitting would cause the program to have a bias towards the dataset it was used with and thus, make it inaccurate when used with other datasets.

The invalid areas produced by the program are: the LP of the bus, its destination placard, and its bumper, among other areas. The low accuracy can be attributed to the problems encountered during the image processing wherein even the bus number is removed. A way to improve this is to consider additional variables in removing invalid candidate areas, as well as, to conduct more experiments regarding changing the weights of the program.

Applying machine learning classifiers such as the SVM, as was used by Rabee, A., & Barhumi, I. (2014), can aid in properly configuring the weights used in the program.

A larger sample size would also benefit the future researchers in improving the application. The small sample size used in this paper is brought upon by the difficulties experienced in getting the bus images. This resulted in the researcher relying on images found online for the dataset.

A problem with identifying buses using their bus number is that there is no standard to them; some bus numbers are located below the wind shield whereas some are at the upper portion of the windshield. Therefore, for this approach to be more effective, it is necessary to create a means of identifying bus numbers located at a variety of places.

6. ACKNOWLEDGEMENT

The author thanks Sir Joel Ilao of De La Salle University for his guidance with the topic.

7. REFERENCES

Chen, R., & Luo, Y. (2012). An improved license plate location method based on edge detection. Physics Procedia, 24, 1350-1356.

Environmental Management Bureau. (n.d.). Retrieved September 1, 2016, from http://emb.gov.ph/

Qadir, R. (2013, February 20). Vehicle number plate recognition. Retrieved August 22, 2016, from https://www.mathworks.com/matlabcentral/fileexchange/40426-vehicle-number-plate-recognition

Rabee, A., & Barhumi, I. (2014, May). License plate detection and recognition in complex scenes using mathematical morphology and support vector machines. In IWSSIP 2014 Proceedings (pp. 59-62). IEEE.

Rashid, A. E. (2013, February). A fast algorithm for license plate detection. In Signal Processing Image Processing & Pattern Recognition (ICSIPR), 2013 International Conference on (pp. 44-48). IEEE.

Sun, M., Qiu, Y., & Zhou, W. (2009, December). License plate extraction based on vertical edge detection and mathematical morphology. In Computational Intelligence and Software Engineering, 2009. CiSE 2009. International Conference on (pp. 1-5). IEEE.