Detection of Text Marks on Moving Vehicles

Vladimir Y. Mariano Rangachar Kasturi
Department of Computer Science and Engineering
Pennsylvania State University, University Park, PA 16802 USA
{mariano,kasturi}@cse.psu.edu

Abstract

Vehicle text marks are unique features which are useful for identifying vehicles in video surveillance applications. We propose a method for finding such text marks. An existing text detection algorithm is modified such that detection is increased and made more robust to outdoor conditions. False alarm is reduced by introducing a binary image test which remove detections that are not likely to be caused by text. The method is tested on a captured video of a typical street scene.

1. Introduction

The need for increased security has spurred a great deal of research to develop intelligent tools for automatic analysis of video. Advances in scene classification, object detection, object classification and activity recognition have proved the eventual feasibility of video-based security systems. Nowadays, inexpensive security cameras can be easily mounted above a street scene to observe activity. A particular object of interest is the vehicle. Vision systems have been envisioned to constantly monitor traffic and observe passing vehicles, extracting important features such as vehicle type, color and distinct marks. Text is a distinct mark that can be found in many vehicles, especially commercial vehicles. In this paper we present an approach for finding text marks on vehicles in a large street scene as captured by a video camera. It uses a typical wide view of a street scene (Figure 1) where vehicles can be observed in full length as well as moving people. Our plan to later use this method as a part of a larger system that creates an index and identifies vehicles by their type, color and distinct marks. In a surveillance application, this could be used for recognizing suspicious activity. An example of suspicious activity is when a certain car repeatedly passes in front of a building (e.g. embassy), or a "Ryder" truck that unusually slows down in front of a federal building.

Another application is in law enforcement, where police can mount a system for finding vehicles of a certain description. One such example is when authorities in the Washington, D.C. area were looking for a box truck that is linked to a sniper. This truck is described as having text marks on its side (Figure 3). The system's ability to find text marks would be useful in this case.

Finding text in video has been a challenging problem. Factors like poor resolution, noise and compression artifacts degrade the video images. In the imaged scene, text can have different fonts, size, colors and background. The most common constraint among previous text detection methods is the assumption that text lines are oriented horizontally. Limited success has been achieved without this assumption.

On vehicles, text marks are very small compared to the imaged scene (Figure 1) and are typically aligned with the vehicle's body or its direction of travel. Motion blur causes character edges to degrade, although the transition between character pixels and background are still visible (Figure 2).

In [6], we have developed a method for detecting text where the only assumptions are that it is horizontally aligned and the stroke pixels have uniform color. It was proven to be invariant to scale and performed well on caption text and horizontal scene text. We modified this method to increase detection and make it more robust to small off-horizontal angles and greater color variation. To reduce the false alarm caused by the increased detection, a binary image test is added which favors detections having text-like properties. The method is applied to the vehicle image. This image of the vehicle is obtained by segmenting moving objects from the scene and rotating the object image such that the vehicle body (and the text marks, if any) are oriented close to horizontal.

2. Related Work

The problem of detecting text in video has been approached using three kinds of methods: edge-based, texture-based, and color-based methods. Edge-based methods [1, 5] use the observation that text has strong edges between the character and background pixels. Texture-based

Proceedings of the Seventh International Conference on Document Analysis and Recognition (ICDAR 2003)
0-7695-1960-1/03 $17.00 © 2003 IEEE

Figure 1. A typical street scene with vehicles and moving people. Our goal is to find any text marks on passing vehicles.

Figure 2. A magnified view of the red text mark of the bus in Figure 1.

methods [4, 8] rely on the texture feature uniformity across the text object.

Color-based methods [3, 2, 6] assume that the text pixels are similar in color or intensity. The three cited methods particularly have the advantage of making no assumption about the color and intensity of the text pixels, whereas others assume a white or bright text foreground color. Color features are first extracted and spatial rules are applied to find the spatial boundaries. The major difference between the three methods is the amount of information used in extracting the color features. Jain and Yu [3] used the information in the whole image in obtaining a multi-valued image decomposition. Connected components is used to determine the spatial boundary. This method would work well if either the text occupies a large portion of the image or if the image has a few distinct colors, such as a magazine. In contrast, Gargi et al [2] examined very small features – small segments of the same color are fused together and grown to become text regions. The work of Mariano and Kasturi [6] strikes a balance between the two by examining a single row of pixels at a time. For every third row of the image, color clusters are computed, and a local search is used to find text boundaries in the row's vicinity. This method work well for horizontal, uniform-color text such as captions.

3 Detection of Vehicle Text

The scene is a typical wide view of a street where both vehicles and people are observed. A video camera is mounted overhead in such a way that the sides of passing vehicles are visible and rarely occluded.

3.1 Vehicle Segmentation and Orientation

The first step is to extract the vehicle image and normalize its orientation. Given an image sequence, moving objects are extracted using frame differencing followed by a series of morphological operations. The vehicle image is then rotated such that it is as close to being horizontal as possible. The assumption is that most text found on vehicles are aligned horizontally with the vehicle's body. The angle of rotation can be computed using the angle of the vehicle's parallel edges (as in [7]), the direction of travel, or prior knowledge of the scene. For our experiments, we use the angle of the vehicle's parallel edges.

3.2 Color Feature Extraction

Text is detected using the method in [6] which is briefly described below. Section 3.3 describes how the method is modified. The basic assumption is that the pixels within the characters' strokes are similar in color, regardless of the background. The rectified vehicle image is scanned every second line of pixels. We use an ellisoid in the $L^*a^*b^*$ color space to model the color variation of stroke pixels. A maximum distance (between ellipsoid points and the center) limits the size of each ellisoid. Candidate ellipsoids are computed from the clusters resulting from a hierarchical clustering of row pixels. For each candidate ellipsoid, a local spatial search finds other pixels that fall in the ellipsoids range of colors, and heuristics are defined to test whether the pixels form a text-like structure.

The algorithm visits every row in the extracted vehicle image. Given a row R on the image, we want to find out whether or not R passes through the middle of a text region.

Figure 3. A box truck with text marks. Washington, D.C. police gave this image as a description of the vehicle linked to the area sniper.

The pixels of R are transformed and clustered in the perceptually uniform $L^*a^*b^*$ color space using hierarchical clustering and the weighted Euclidean norm as the distance function. The weighted norm was used to achieve a slight invariance to lightness (weights: $L^* = 0.8, a^* = 1.1, b^* = 1.1$). Later, this weighted norm will be used to test membership into the formed clusters. Ellipsoids in color space are defined as a result.

Each cluster C is tested to see if it contains pixels belonging to text. Locating the bounding rows (top and bottom rows of text) is the first step (Fig. 4). The cluster points are marked back on row R to create streaks $S_i, i = 1 \ldots N_s (number of streaks)$ of pixels in the row R. Then pixels above and below R are examined and each pixel that falls within the ellipsoid of cluster C are colored with a value of T. All other pixels are marked T'.

We now try to find out if there are bounding rows above and below R which may contain horizontal text. Given a pair of adjacent streaks S_i and S_{i+1}, we find R_a – the first row above R in which the segment covering S_i and S_{i+1} is colored T'. We also find R_b – the first row below R in which the segment covering under S_i and S_{i+1} is colored T'. The R_a of each pair of adjacent streaks is computed and collected in an alignment histogram H_a, where the bins are the rows of the image. H_b is computed in the same way by taking all the R_b's. We declare the existence of a bounding row B_a if at least 60% of the elements in H_a are contained in a window of three or fewer adjacent histogram bins. B_b's existence is computed in the same way from H_b. If B_a and B_b exists, $height$ is defined as their difference.

If the cluster C contains text pixels, then B_a and B_b would mark the text block's upper and lower row boundaries , and $height$ would define its vertical dimension. Figure 4 illustrates the computation of B_a, B_b and $height$.

We look for text blocks using heuristics on $height$ and the short streaks' lengths and spacings. Streaks longer than $height$ are discarded. Spacings that are longer than $height$ are considered not part of a text block. The remaining regions are now smaller blocks with short streaks. If a block's width is greater than $1.5 * height$ and the number of short streaks inside is greater than 3, then it is considered a text block, otherwise it is discarded. Finally, the text block is expanded a few pixels to the left and right to ensure full coverage of the characters at the ends.

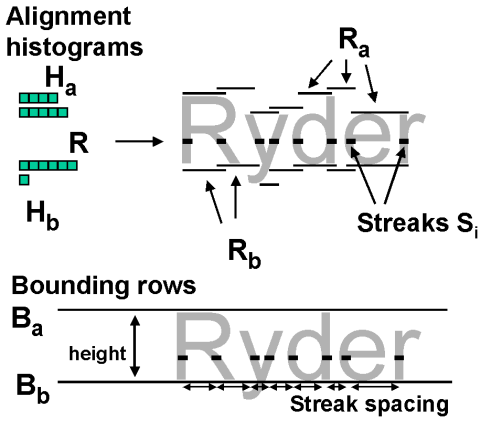

Figure 4. Computing the bounding rows. One of the color clusters in row R are marked as short streaks and pixels of the text "Ryder" lie within the ellipsoid of the cluster. The spacings between the streaks are used to compute the left and right boundaries of the text bounding box.

3.3 Detection Increase and Binary Image Test

The detection algorithm assigns the boundaries of a text box based on an extracted color, a local spatial search above and below the analyzed row R, and the spacings between the streaks. This has worked quite well in previous work on artificially-placed caption text [6] where there is little variation in color across text pixels. Furthermore, caption text is almost perfectly horizontal, which benefits localization. On vehicle text, however, conditions are less favorable. Even if the vehicle image is rotated to be horizontal, the text can still be off-horizontal by a few degrees. On the vehicle surface, the colors of the text pixels still appear similar but there is greater variation compared to caption text. The reason is that this is on an object surface that is imaged outdoors.

To make the detection more robust to these problems, the

parameters of the detection algorithm are relaxed in order to increase detection. To relax the horizontal constraint, we increase the number of adjacent bins in the the alignment histograms which are used to find the bounding rows B_a and B_b. To account for the greater variation in text pixel color, the maximum distance (between cluster elements and cluster center) is increased in the clustering algorithm.

These modifications would result in better recall but would also increase the number of false alarm. A binary image test is used to reduce the false alarm by distinguishing between boxes that are likely caused by text and those that are not. The binary image of each text box is computed by testing each pixel whether it is inside the color ellipsoid which was computed in the clustering step. Figure 5 shows binary images of boxes caused by the "Balfurd" text in Figure 7. Consider any of these binary images. If we observe most of the pixel rows, there is repeated transition between consecutive zeros and consecutive ones as you go from left to right. This is caused by the characters' vertical and diagonal strokes. Now comparing these to the binary image of a typical false alarm in Figure 6, this is not the case.

The binary test examines each row of the text box's binary image and counts the number of segments (countiguous groups of ones). If enough of the rows have segments greater than a threshold then the text box is accepted. The threshold should be dynamic because the number of segments is directly related to the number of characters covered by the box. Since the number of characters is not known, we use as an alternative the text box's aspect ratio (width/height) multiplied by a factor to compute the threshold. We define the test as follows: Let

$$MedSeg = Median(NumSeg[])$$

$$NumSeg[] = [NumSeg_1, NumSeg_2, \ldots, NumSeg_m]$$

where $NumSeg_j$ is the number of segments across row j of the binary image and m is the height of the binary image. A segment is defined as a group of row-contiguous pixels with binary value equal to 1.

The text box is accepted if its binary image satisfies:

$$MedSeg > (Width_{binary}/Height_{binary}) * RatioFactor$$

This is a comparison of $MedSeg$ with a dynamic threshold. It is a function of the box's aspect ratio and controlled by a fixed $RatioFactor$ parameter.

Finally, all the detected boxes that passed the binary test are fused together to form the text regions.

In summary, the detection algorithm extracts multiple hypotheses on the color of the text, and applies rules to test each hypothesis and find the spatial boundaries. The ability to generate good hypotheses makes it possible to create the binary image, have the binary test and weed out false alarm. This ability sets it apart from the other color-based detection methods in the literature and obviously from edge-based and texture-based methods.

Figure 5. Binary images generated by color ellipsoid membership. These overlapping text boxes are detected on the "Balfurd" text in Figure 7. For each text box, a single color cluster from the pixels of a single row is used to compute an ellipsoid in color space and spatially find the bounding box. The color ellipsoid is then used to generate the binary image (black = 1).

3.4 Tests on Image Sequences

The method was tested on a 20-minute video of a street scene with pedestrians and vehicles. Of the 285 vehicles that appeared on the scene, 24 had text that is large enough to be read of which 23 had their text marks detected. Figure 8 shows an example of a detected text mark on a segmented vehicle. Of the vehicle which didn't have text in them, there were 33 false alarm.

4 Conclusion and Future Direction

We have proposed a method for detecting text marks on vehicles using color features. A modified existing method

Figure 6. Binary image of a false alarm. In this example, the analyzed row passed through the vehicle windows whose sides were mistaken for character strokes.

Figure 7. Most text are parallel to the vehicle body. The moving vehicle is extracted from the scene and oriented horizontally before applying the text detection algorithm. The angle of rotation can be derived using the angle of the vehicle's parallel edges, the direction of travel, or prior knowledge of the scene. The "Balfurd" image shows what the text looks like after rotation.

was applied to extracted vehicle images and a binary image test is used reduce the false alarm.

When a text mark is covered by multiple detected boxes, the binary images (like those in Figure 5) seem to suggest that the hypothesized colors are accurate even if we are not sure which ones are foreground. Perhaps a method can be formulated for grouping and fusing these images in order to create a binary which can be passed to an OCR for recognition.

Figure 8. The test street scene, a moving truck segmented from the scene, and the detected text mark.

References

[1] L. Agnihotri and N. Dimitrova. Text detection for video analysis. In *Proc. IEEE Workshop on Content-Based Access of Image and Video Libraries*, pages 109–113, June 1999.

[2] U. Gargi, S. Antani, and R. Kasturi. Indexing text events in digital video databases. In *Proc. International Conference on Pattern Recognition*, pages 916–918, 1998.

[3] A. K. Jain and B. Yu. Automatic text location in images and video frames. *Pattern Recognition*, pages 2055–2076, December 1998.

[4] H. Li, D. Doermann, and O. Kia. Automatic text detection and tracking in digital video. *IEEE Transactions on Image Processing*, 9(1):147–156, January 2000.

[5] R. Lienhart and A. Wernicke. Localizing and segmenting text in images and video. *IEEE Transactions on Circuits and Systems for Video Technology*, 12(4):256–268, 2002.

[6] V. Y. Mariano and R. Kasturi. Locating uniform-colored text in video frames. In *Proc. International Conference on Pattern Recognition*, volume 4, pages 539–542, 2000.

[7] T. N. Tan and K. D. Baker. Efficient image gradient based vehicle localization. *IEEE Transactions on Image Processing*, 9(8):1343–1356, 2000.

[8] Yu Zhong, Hongjiang Zhang, and Anil Jain. Automatic caption localization in compressed video. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(4):385–392, 2000.