Recognizing Text-Based Traffic Signs

Jack Greenhalgh and Majid Mirmehdi

***Abstract*—We propose a novel system for the automatic detection and recognition of text in traffic signs. Scene structure is used to define search regions within the image, in which traffic sign candidates are then found. Maximally stable extremal regions (MSERs) and hue, saturation, and value color thresholding are used to locate a large number of candidates, which are then reduced by applying constraints based on temporal and structural information. A recognition stage interprets the text contained within detected candidate regions. Individual text characters are detected as MSERs and are grouped into lines, before being interpreted using optical character recognition (OCR). Recognition accuracy is vastly improved through the temporal fusion of text results across consecutive frames. The method is comparatively evaluated and achieves an overall F_{measure} of 0.87.**

***Index Terms*—Maximally stable extremal region (MSER), scene structure, text detection, traffic text sign recognition.**

I. Introduction

The automatic detection and recognition of traffic signs is a challenging problem, with a number of important application areas, including advanced driver assistance systems, road surveying, and autonomous vehicles.

While much research exists on both the automatic detection and recognition of symbol-based traffic signs, e.g., [1]–[6], and the recognition of text in real scenes, e.g., [7]–[14], there is far less research focused specifically on the recognition of text on traffic information signs [15]–[17]. This could be partly due to the difficulty of the task caused by problems, such as illumination and shadows, blurring, occlusion, and sign deterioration.

Without the use of additional temporal or contextual information, there is few information to determine traffic signs from nontraffic signs on the fly, while driving, other than basic features, such as shape or color. On this basis, the number of false positives (FPs) likely to occur in a cluttered image, such as a road scene, is high. This is demonstrated in the example in Fig. 1, where although the traffic sign present in both images is successfully detected, more FPs are detected by the system (in the top scene) when additional structural and temporal information is not deployed.

We approach this problem by detecting large numbers of text-based traffic sign candidates using basic shape and color information. This overdetection is important to ensure that no true positives (TPs) are missed. We then reduce the large number of detected candidate regions by making use of the structure of the scene, as well as its temporal information, to eliminate unlikely candidates.

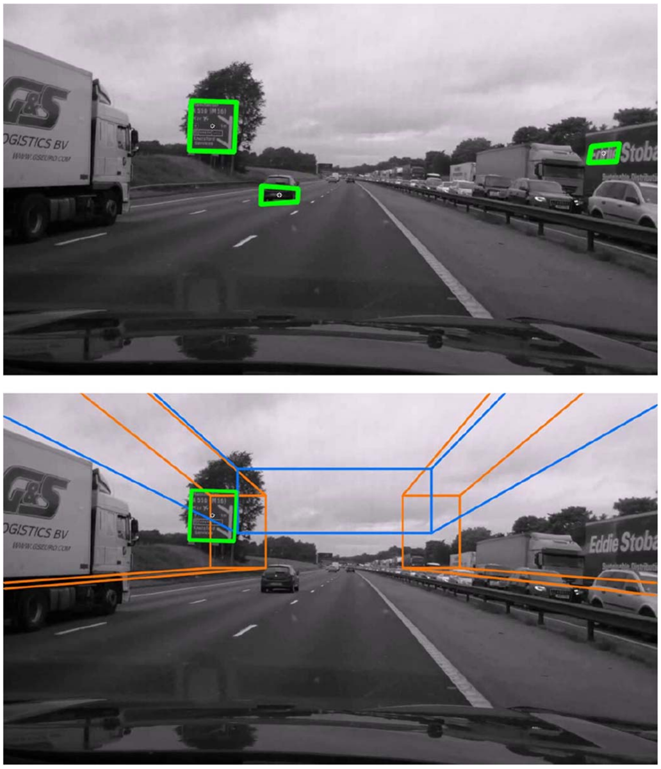


Fig. 1. System output showing detection of traffic signs (top) without and (bottom) with the use of structural and temporal information.

The proposed system comprises two main stages: detection and recognition. The detection stage exploits knowledge of the structure of the scene, i.e., the size and location of the road in the frame, to determine the regions in the scene that it should search for traffic text signs. These regions are defined once the vanishing point (VP) of the scene and, hence, the ground plane are determined.

Potential candidate regions for traffic signs are then located only within these scene search regions, using a combination of MSERs and hue, saturation, and value (HSV) color thresholding. By matching these regions through consecutive frames, temporal information is used to further eliminate FP detected regions, based on the motion of regions with respect to the camera and the structure of the scene.

Once a potential traffic sign has been located, the next stage of the algorithm attempts to recognize text within the region. First, an approximate perspective transform is applied to the region, in order to vertically align text characters. Candidate components for text characters are then located within the region and sorted into potential text lines, before being interpreted

Manuscript received May 2, 2014; revised August 14, 2014; accepted September 29, 2014. Date of publication December 8, 2014; date of current version May 29, 2015. This work was supported in part by the Engineering and Physical Sciences Council and in part by Jaguar Cars Limited. The Associate Editor for this paper was B. Morris.

The authors are with the Department of Computer Science, University of Bristol, BS8 1TR Bristol, U.K.

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TITS.2014.2363167

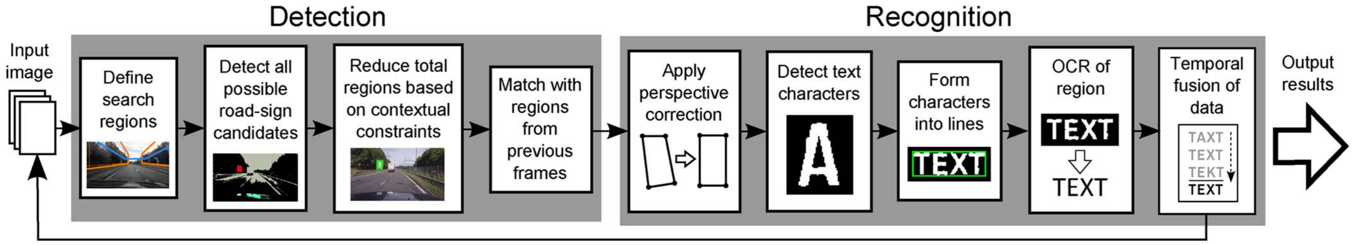

Detection
Recognition
Input image
Define search regions
Detect all possible road-sign candidates
Reduce total regions based on contextual constraints
Match with regions from previous frames
Apply perspective correction
Detect text characters
A
Form characters into lines
TEXT
OCR of region
TEXT
TEXT
Temporal fusion of data
TAXT
TEXT
TEKT
TEXT
Output results

Fig. 2. Pipeline for detection and recognition stages of the proposed approach.

using an off-the-shelf optical character recognition (OCR) package. To improve the accuracy of recognition, OCR results from several frames are combined together by matching individual words through frames and using a weighted histogram of results. The entire system pipeline is shown in Fig. 2.

In Section II we review past work and state the improvements that we make against those that are of most significant relevance. Then, in Section III, we outline the methodology used for the detection of text-based traffic signs. In Section IV we describe the methodology used for extracting and interpreting text from the detected traffic signs. In Section V we present comparative results to illustrate the performance of the system. Finally, the conclusion is drawn in Section VI.

II. Related Work

Much research exists on the detection and recognition of text in natural scenes. Approaches to this problem can be broadly divided into two groups: region-based methods, e.g., [9], [12], and [18], and connected component (CC)-based methods, e.g., [10], [11], [13], [14], and [19]. Region-based text detection methods use local features, such as texture, to locate text regions, whereas CC-based methods attempt to segment text characters individually by using information such as intensity, color distribution, and edges. They usually consist of three phases: a first stage to detect CCs within the image, a second stage to eliminate unlikely CCs based on their features, and a final stage that attempts to group the remaining CCs into words or lines.

More relevant to the context of this paper, the amount of research focused specifically *on the detection of text within traffic signs* is fairly limited, perhaps due to the difficulty of the task. The existing state-of-the-art methods all consist of two stages: detection and recognition, e.g., [15]–[17].

Wu *et al.* [15] found candidate regions, using a combination of Shi and Tomasi features [20], Gaussian mixture models, and geometric analysis. The authors assumed that traffic sign text appeared on a vertical plane with respect to the motion and optical axis of the camera. However, in reality, it is likely that text signs will appear from a viewpoint that is not quite fronto-parallel. Therefore, a perspective transform is necessary to give OCR a better chance of text recognition, as performed by our proposed method. Candidate regions were matched through consecutive frames, and were interpreted using an OCR system once they were of an adequate size. The authors reported a detection rate of 88.9% and a false detection rate of 9.2%, based on a data set of 22 video sequences, each around 30 s long.

Reina *et al.* [16] segmented regions of interest based on color information, by applying a threshold to the chrominance and luminance channels in the Lab color space. Rectangular regions were found by comparing the fast Fourier transform (FFT) signature of each blob to the FFT signature of a rectangular-shaped reference. The four points representing the corners of the rectangular region were then found by taking the peaks of the FFT signature; using these points, the regions were rotated in an attempt to vertically align text characters. This is again an insufficient approach to deal with the perspective recovery of the text panel from the vehicle viewpoint, and the perspective correction in Section IV-A establishes a more robust solution. No quantitative results were provided by the authors.

The method presented by González *et al.* made use of MSERs for the detection of both traffic signs and text characters [17]. White and blue traffic panels were detected in each frame, using a combination of color segmentation and bag of visual words. These regions were then classified using both support vector machines and Naïve Bayes classifiers. The method was applied to single images, with no use of temporal information, and the emphasis placed on the geolocalization of traffic signs using Global Positioning System information. The height of the text itself was used to approximate the real-world size, and hence distance from the camera to the traffic signs. All results were based on 10 763 images taken from Google Street View. The authors provided individual detection and recognition rates for words, numbers, and symbols at short, medium, and long distances. These rates ranged between 13.09% and 90.18% for detection and between 8.51% and 87.50% for recognition.

The methods in [15]–[17] suffer from several limitations, which are improved upon by our proposed method, where assumptions made for the detection of text-based traffic signs are general enough to ensure a high recall rate. Our proposed method uses structural and temporal information to eliminate the additional FPs. Results are provided in Section V-A to validate this claim. In addition, our method offers improvements on the raw OCR approach, which was used in works such as [15], by using perspective recovery and temporal fusion methods. The performance of this approach is validated in Section V-B.

While use of temporal information in the context of symbol-based traffic sign detection has been explored before [21], the work described in this paper expands on the idea, by also incorporating structural information from the scene.

Other example works that have considered text detection with a mobile sensor are those that rely on wearable devices, e.g., Goto and Tanaka [22] and Merino-Gracia *et al.* [19], and those applied to mobile robotics in [23]–[25].

Several data sets have been proposed for the validation of traffic sign recognition systems, including the German traffic sign detection benchmark [26], the German traffic sign recognition benchmark [27], and the Belgian traffic sign data set

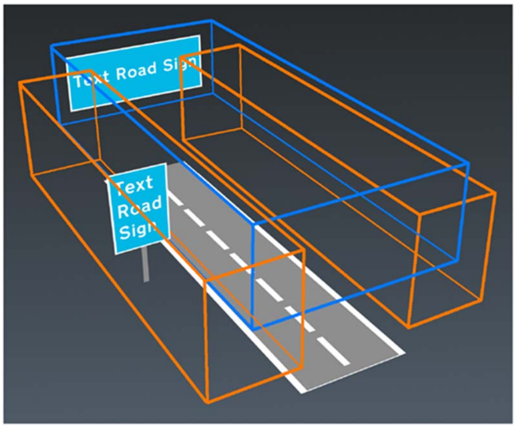


Fig. 3. Illustration of search regions projected into the 2-D image.

[28]. It should be noted that the focus of these data sets is on the detection of symbol-based traffic signs, and they are therefore not applicable to the validation of our method, which focuses on text. The traffic text sign data that we used were obtained from Jaguar Land Rover Research, and these are available to other researchers at http://www.bris.ac.uk/vi-lab/projects/roadsign/index.html. These data were captured with a camera, for which the full calibration parameters are known.

III. Detection of Text-Based Traffic Signs

The first stage of the proposed system detects candidates for text-based traffic signs. This consists of three phases: determination of search regions (regions of interest where the text sign is expected to be found), detection of all possible candidates within these regions, and reduction of candidates using contextual constraints.

Search regions of interest for traffic signs are found within the image, by first locating the sides of the road in the image and then defining 3-D search boxes, which are projected back onto the original 2-D frame. These search regions are shown in Fig. 3, where the orange region is for traffic signs on either side of the road, and the blue box is for overhead gantries.

A. *Finding Sides of Road and VP*

In order to determine search regions for traffic signs in each frame, the sides of the road and the road VP must be detected. Our approach to VP detection is traditional and popularly used in other works, e.g., [29]. First, the Canny edge detector is used to detect edges in the image, which is followed by the Hough transform to locate straight lines. The total number of Hough lines is then reduced by eliminating lines that are too short, that do not approximately pass through the center of the frame, or (for the purposes of our application) that appear near the top of the image; an example frame is shown in Fig. 5. An "accumulator" of line intersections is then created from the intersections between the remaining Hough lines, the peak of which is taken to be the VP of the road. The parameters for Canny and Hough were determined empirically on a subset of our data set and fixed throughout our experiments.

Once the VP is found, the camera yaw γ and pitch θ can be computed as

$$\gamma = \tan^{-1}\left(\frac{C_x - VP_x}{f}\right) \tag{1}$$

$$\theta = \tan^{-1}\left(\frac{C_y - VP_y}{f}\right) \tag{2}$$

where f is the camera focal length, and C is the camera's center of projection. Using γ and θ, an inverse perspective mapping (IPM) can be performed on both the original frame and the detected Hough lines using

$$u(x, 0, z) = \frac{\left(\tan^{-1}\left(\frac{z-d}{x-l}\right) - (\gamma - \alpha_u)\right) \cdot (m-1)}{2\alpha_u} \tag{3}$$

$$v(x, 0, z) = \frac{\left(\tan^{-1}\left(\frac{h\sin\gamma}{z-d}\right) - (\theta - \alpha_v)\right) \cdot (n-1)}{2\alpha_v} \tag{4}$$

where u and v represent coordinates in the original frame, m and n are the dimensions of the original frame, α_u and α_v are the angular aperture, and x and z represent the coordinates in the IPM image. The values h, l, and d represent the position of the camera with respect to the ground plane, as shown in Fig. 4. These values can be estimated and adjusted, in order to shift and scale the IPM image.

From the reduced set of Hough lines, it is possible to approximate the sides of the road in the IPM image. If we assume that the camera is located on the center of the vehicle facing forward and that the vehicle is in the middle of the lane, it follows that the center of the current lane will be in the center of the IPM image. An example of the IPM image and transformed Hough lines is shown in Fig. 5.

The set of IPM Hough lines are then further reduced by eliminating lines that are not approximately vertical and lines that are below a certain length. This set of lines is then divided into two groups, for the left- and right-hand sides of the image. The mean of all lines, weighted by line length, is then calculated for each side of the image. These average lines are taken as approximations of the sides of the road. Estimates for the VP and sides of the road are detected in every frame, and they are then tracked throughout subsequent frames using the Kalman filter, following the work in [29].

B. *Defining Search Regions Within the Original Frame*

Once the sides of the road are detected, the size and location of the search regions can be defined. Three search regions are used, i.e., one to the left-hand side of the road, one to the right, and one above. The dimensions of these regions are determined empirically through analysis of the validation data set and kept constant throughout our experiments. The top search region is defined to be the width of the road, given that overhead gantries, which appear in this region, never extend beyond the sides of the road. Therefore, the width of this region is determined dynamically based on the detected positions for the sides of the road. The dimensions and height of the roadside regions are fixed, but their horizontal positions change dynamically to position them by either side of the road.

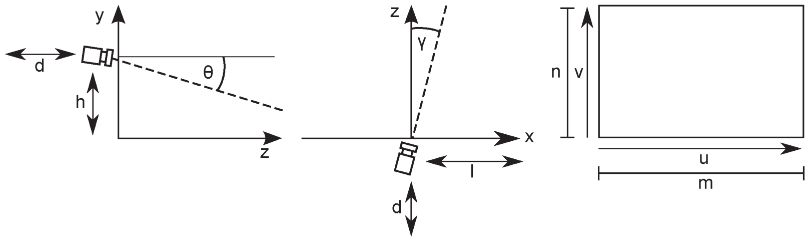


Fig. 4. Camera position and captured frame.

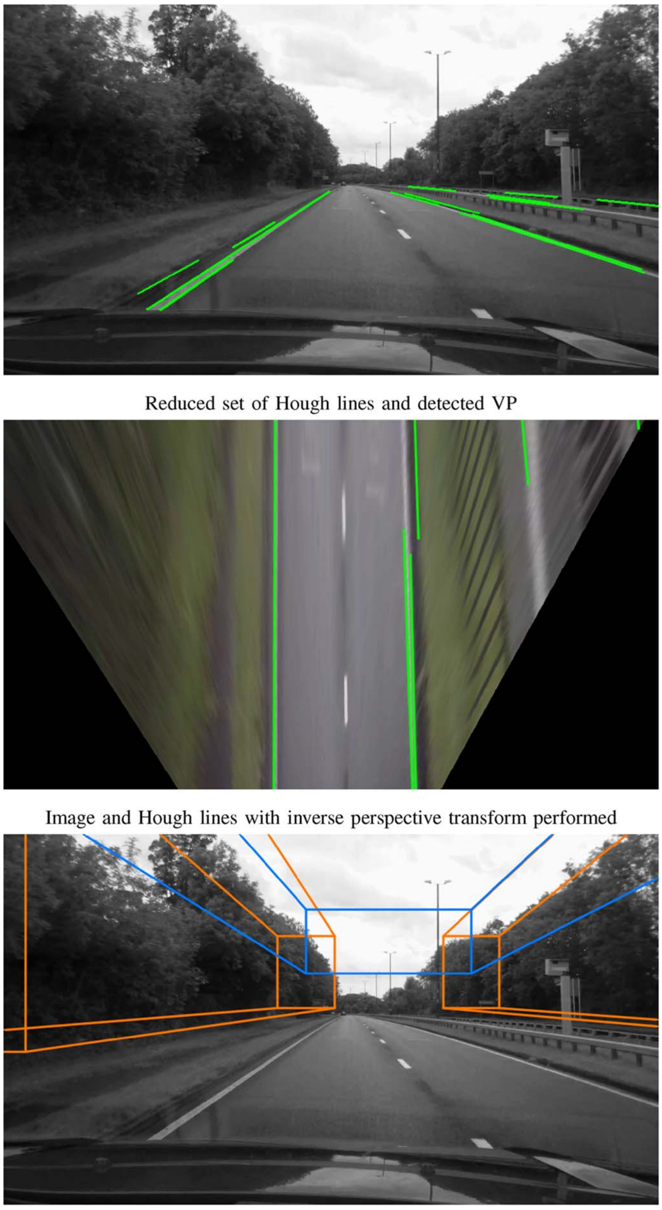

Reduced set of Hough lines and detected VP

Image and Hough lines with inverse perspective transform performed

Algorithm output showing search regions

Fig. 5. Output of various stages of algorithm to define search regions.

The real-world dimensions of these regions can be roughly estimated, by assuming that the distance between the overhead gantry and the ground is approximately 5.1 m, i.e., the minimum legal unmarked gantry height in the U.K., and using this as a reference. These dimensions are stated in Table I.

TABLE I
DIMENSIONS OF SEARCH REGIONS

Region	Width (m)	Height (m)	Length (m)	Height from ground (m)
Road-side	6.4	8.6	30.6	0.95
Overhead	road width	7.8	30.6	5.0

TABLE II
VALUES USED FOR HSV THRESHOLDING

Colour	Hue range	Min sat.	Max sat.
Brown	12°- 52°	50%	100%
Green	136°- 176°	20%	100%
Blue	184°- 224°	24%	100%

These 3-D regions are then projected back into the original 2-D frame, as shown in Fig. 5, with the search regions for signs by the sides of the road in orange and the search region for overhead gantries in blue.

C. Detection of Text Traffic Sign Candidates

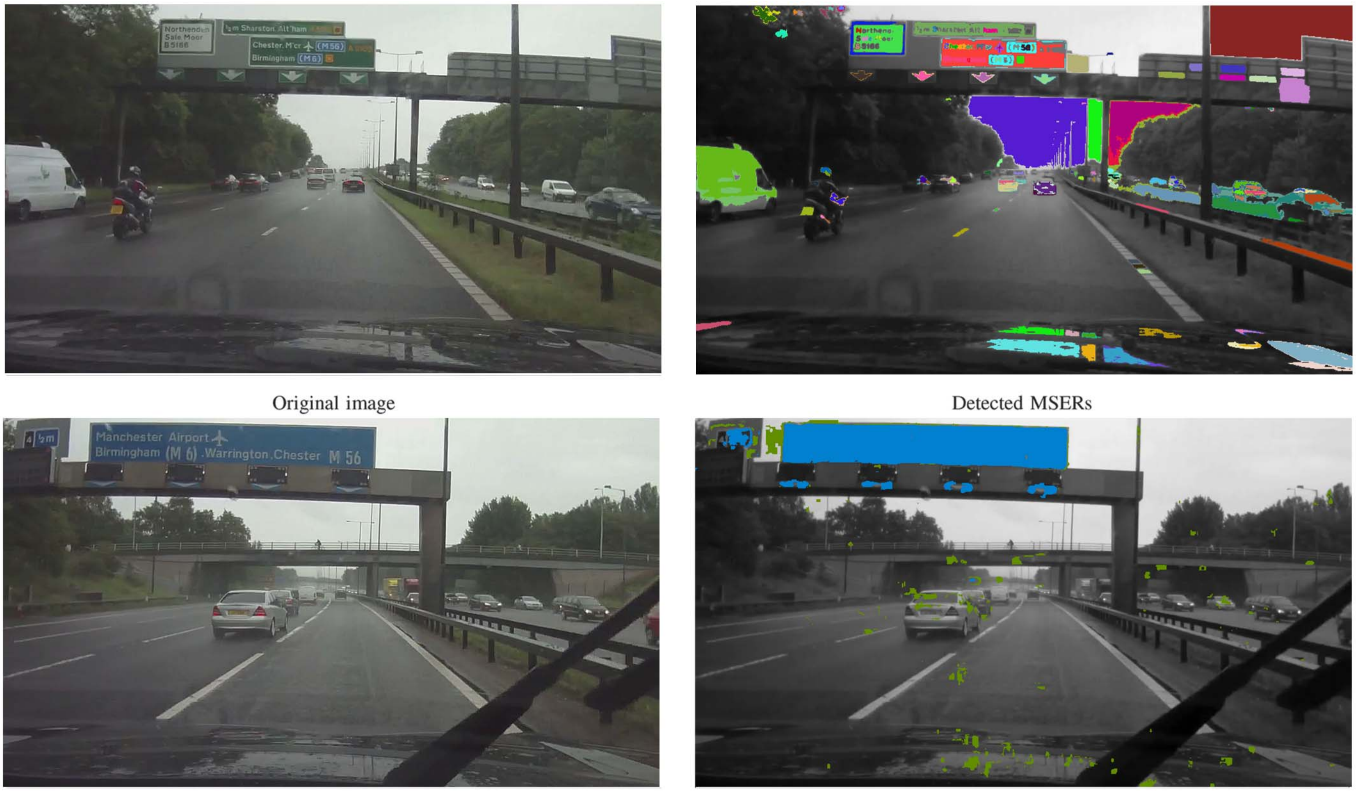
The next stage of the algorithm involves the detection of candidates for text-based traffic signs within our defined scene search regions. We follow from our previous work on the detection of symbol-based traffic signs [4] and detect text traffic sign candidates using both MSER and HSV color thresholding. These two kinds of region detector are used, in order to gain as high a recall as possible and ensure that all possible traffic signs are detected in all conditions.

MSERs are defined to be regions that maintain their shape approximately through several image threshold levels. This region detector is robust to lighting and contrast variations and detects high-contrast regions, which make it suitable for the detection of traffic signs. An example frame with detected MSERs is shown in Fig. 6.

Additional traffic text sign candidates are detected using HSV thresholding. Each frame is first transformed into the HSV color space, before a threshold is applied to both hue and saturation channels. The value channel is ignored to help the system remain invariant to changes in brightness. Threshold values are determined using the template images provided in the U.K. Department of Transport Traffic Sign Manual [30]. These values are provided in Table II and are also illustrated in Fig. 7, for green, blue, and brown traffic signs. They remain the same for all our experiments. Two sets of CCs are thus found by HSV thresholding to detect candidate regions for blue and green traffic signs. Example candidate regions are shown in Fig. 6.

D. Reduction of Candidate Regions Based on Contextual Constraints

Next, we reduce the total number of candidates, by using both temporal and contextual information. Assuming that the



Original image

Detected MSERs

Original frame

Detected HSV thresholded regions

Fig. 6. Examples of MSERs and HSV-thresholded regions.

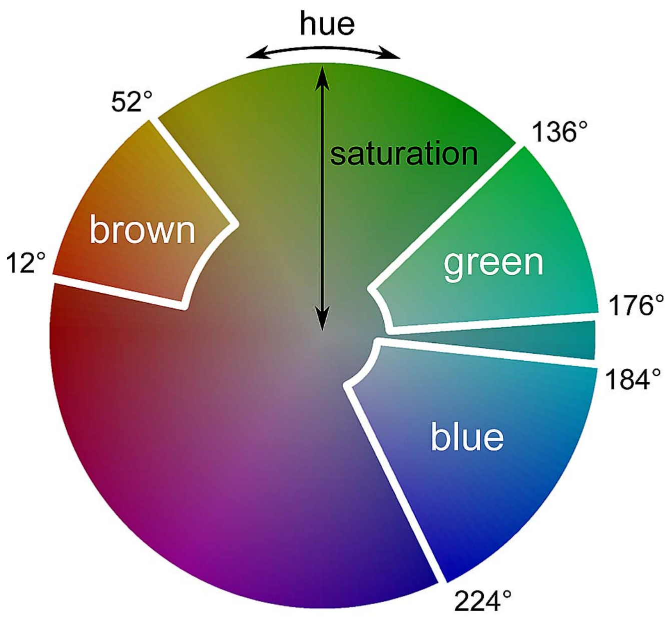


Fig. 7. Hue against saturation, with values for color traffic signs marked.

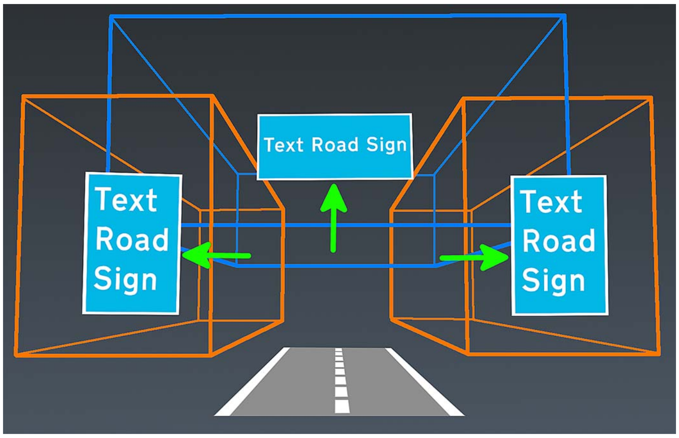


Fig. 8. Motion of traffic sign with respect to camera.

vehicle is moving forward and that the traffic sign appears within the defined search regions, we can expect the motion of tracked regions within the frame to be as illustrated by the green arrows in Fig. 8. Temporal information about candidate regions is then easily gained by matching each candidate between frames using its size, aspect ratio, and location features. It is assumed that between consecutive frames, the Euclidean distance between matching regions will remain small, despite their temporal motion within the frame. Based on this assumption, each traffic sign candidate from the current frame is compared with each candidate from the previous frame. A match is made between the regions with the smallest Euclidean distance, given that this distance is below a defined threshold and that their aspect ratios are suitably similar. If no match is found, the detected candidate is treated as a new traffic sign.

For a pinhole camera model, it is given that candidates will grow in size through consecutive frames. This is shown in Fig. 9, where C represents the camera center, T_1 and T_2 represent a traffic sign at different distances with respect to the camera, and T_1' and T_2' represent the 2-D projection of T_1 and

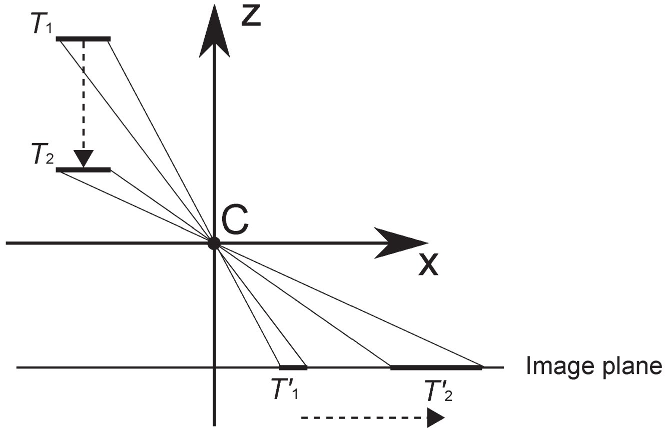


Fig. 9. Pinhole camera model representing motion of traffic sign with respect to the camera.

T_2 on the image plane. As the camera moves forward, the traffic sign in its view will move away from the VP in the image plane and increase in size. Any tracked candidates that violate these conditions are assumed to be FPs and are rejected.

The constraints applied to the region size vary based on the location of the region within the frame. The maximum and minimum values for region width and height increase, the further the region is from the VP. Overhead gantries tend to be far wider than traffic signs found by the side of the road; therefore, the maximum and minimum values for aspect ratio and width depend on which search region the traffic sign is detected in. The maximum sizes of candidate traffic signs are represented by the size of the 3-D search boxes, where the maximum size of roadside candidates depends on their horizontal distance from the VP, and the maximum size of gantry candidates depends on their vertical distance from the VP.

IV. Recognition of Text

The second stage of the system recognizes text contained within the detected candidate regions. To increase the chances of OCR in recognizing our noisy text regions, we first apply an approximate perspective transform to the rectangular candidate regions to vertically align them and their text characters. Individual text characters are then segmented, formed into words, and then sent to OCR. Results from several instances of each traffic sign are then combined, in order to further improve recognition. These steps are detailed next.

A. Correction of Detected Candidate Regions

Before text is read from the detected region, an approximate perspective transform is applied to vertically align the text characters and reduce perspective distortion. The correction is performed by first fitting a quadrilateral to the CC representing the traffic sign; example traffic sign shapes are shown in Fig. 10. The method is required to be robust to noisy rectangular candidates, such as those in Fig. 11.

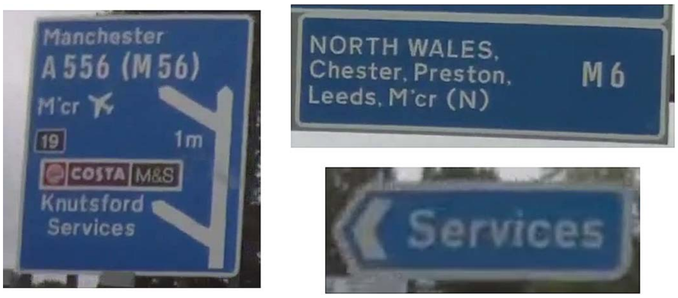


Fig. 10. Example traffic sign shapes.

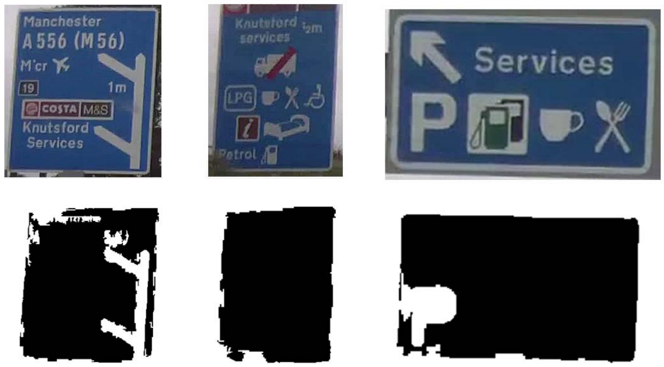


Fig. 11. Example noisy traffic sign candidates shown with their corresponding HSV threshold CCs.

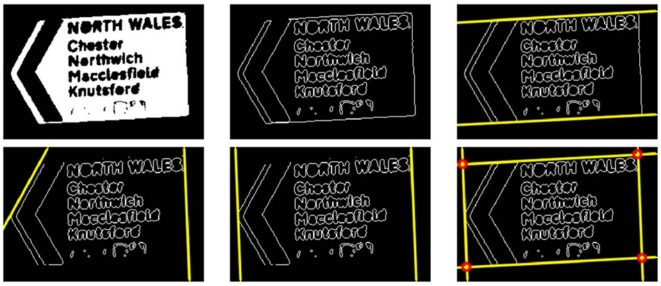


Fig. 12. Stages of quadrilateral detection, showing (top left) original CC, (top center) edge image, (top right) fitted horizontal line, (bottom left) fitted vertical lines, (bottom center) corrected vertical lines, and (bottom right) detected quadrilateral.

First, the CC is filtered down to just the points representing edge pixels, and the well-known random sampling consensus (RANSAC) algorithm is then applied to estimate parameters for lines representing the top and bottom edges [31]. It is assumed that either the left or right side of the CC can be approximately fitted to a single straight line, if not both sides. RANSAC is again applied to fit a straight line to both the leftmost pixels and rightmost pixels, ignoring any points associated with the fitted top and bottom lines.

The line that best fits its edge pixels is then selected, and the other is rejected and replaced with a line of equal gradient, intersecting the outer most pixel. For example, in the bottom left image in Fig. 12, the right-hand side would be selected and the other rejected.

The left and right sides of the quadrilateral representing the candidate are assumed to be parallel, as rotation around the x-axis is minimal. The quadrilateral representing the region is then found from the points at which these four lines intersect. Each stage of this method is shown in Fig. 12.

A homography H can be calculated from this set of points (x) and a set of points representing a regular rectangle x' (see

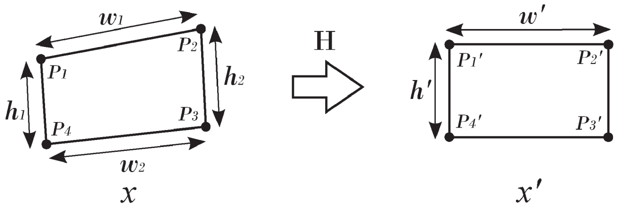
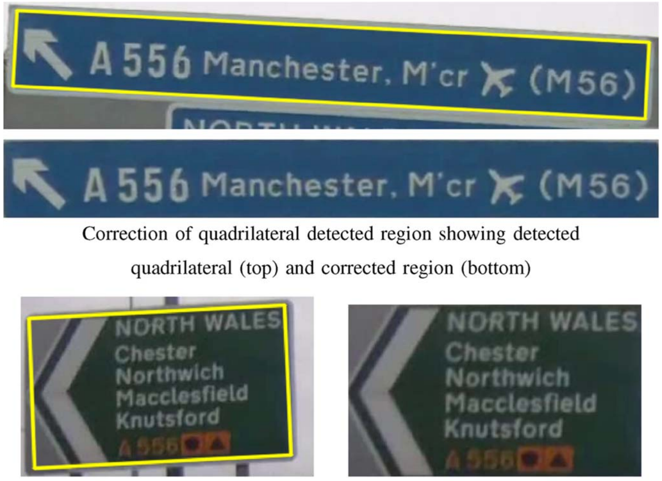


Fig. 13. Approximate perspective transform using homography H.



Correction of quadrilateral detected region showing detected quadrilateral (top) and corrected region (bottom)

Correction of non-quadrilateral detected region, showing detected quadrilateral (left) and corrected region (right)

Fig. 14. Example detected regions with quadrilaterals and resulting corrected regions.

Fig. 13) [32]. The dimensions of the corrected rectangle x' are defined as

$$w' = \max\{|P_1 - P_2|, |P_4 - P_3|\} \tag{5}$$

$$h' = \max\{|P_1 - P_4|, |P_2 - P_3|\} \tag{6}$$

where w' and h' represent the width and height of the corrected region, and P_1, P_2, P_3, and P_4 are points that represent the corners of the detected quadrilateral.

Perspective mapping can now be performed using homography H, which will cause text characters to be vertically aligned. Example results of this transformation are shown in Fig. 14.

B. Detection of Text Lines

The next stage of the algorithm locates lines of text within the detected candidate regions. This allows the total number of CCs to be reduced, removing noncharacter CCs and hence improving the chances for higher OCR accuracy. Text characters are first located as MSERs within the region, which are then reduced based on thresholds applied to features of the candidate characters and their bounding boxes (BBs). These thresholds were determined empirically based on a validation data set, and these are recorded in Table III. All remaining character regions are then grouped into text lines. As the region has been transformed with the approximate perspective transform, the text lines are assumed to be vertically aligned.

TABLE III
CHARACTER FEATURES

Features	Min value	Max value
Aspect ratio	0.18	1.8
CC area / BB area	0.33	1.0
CC perimeter / BB perimeter	0.7	1.94

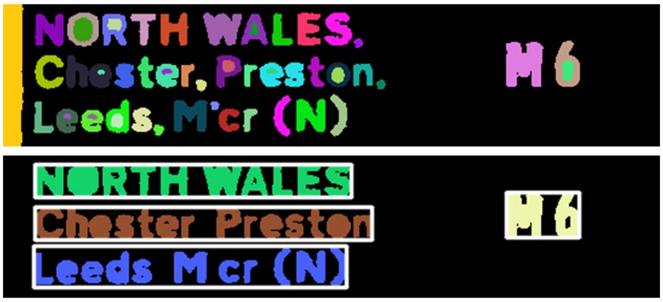


Fig. 15. Stages of line detection, showing (top) detected MSERs and (bottom) detected text lines.

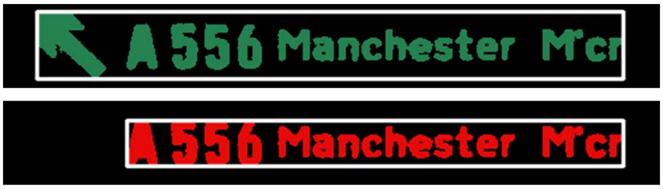


Fig. 16. Line detection in two passes, showing (top) detected text line after first pass and (bottom) detected text line after second pass.

Each character is compared with other characters and labeled based on simple perceptual similarity rules, i.e., similarity of component heights, vertical distance, horizontal distance, and ratio of component areas. Fig. 15 shows the detected MSER components and initially detected text lines.

Once text lines are detected, a second pass removes unlikely characters from each line. This stage is necessary because symbols and other noncharacter components can get grouped with text characters, causing OCR errors. For each text line, the median character height is found and then used to define a stricter set of size constraints. For example, as shown in Fig. 16, the "arrow" symbol has been incorrectly grouped as a text character in the first pass but is then removed in the second pass.

C. OCR for Individual Candidates

The set of detected text lines (in grayscale) are passed on to the open-source OCR engine "Tesseract" [33] for recognition. Given that U.K. text-based traffic signs contain only two typefaces (see Fig. 17), i.e., motorway and transport, the OCR engine was retrained using only these typefaces [34]. Tesseract was also trained on other symbols, which may appear to avoid their misclassification as characters, e.g., an "airport" symbol may be incorrectly classified as a letter "X."

D. Temporal Fusing of OCR Results

To improve the accuracy of OCR, results are combined across several frames. Individual words are compared from frame to frame based on size, with word BBs normalized by the region size. The results of the ten most recent detections

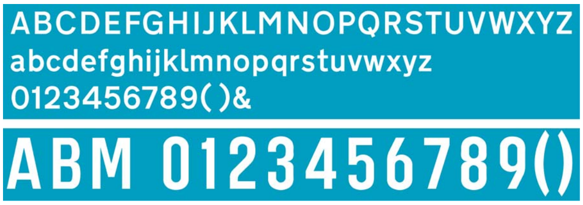


Fig. 17. U.K. traffic sign typefaces, showing (top) transport typeface and (bottom) motorway typeface.

TABLE IV
OCR Results for Individual Frames and Final Result

Frame	OCR result
n-7	TirodnesS can kill Take a break
n-6	Tjrednoss can kill T&ko a broak
n-5	Tiredness can kill Tak& a broak
n-4	Tiredness can kill Tako a break
n-3	Tiredngss can kill T&ke a break
n-2	Tiredness can kill T8ke a break
n-1	Tirednesis can kill Take a break
n	Tiredness can kill T8k8 break
Fused data	**Tiredness can kill Take a break**

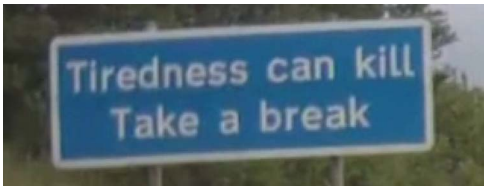


Fig. 18. Text on traffic sign associated with Table IV.

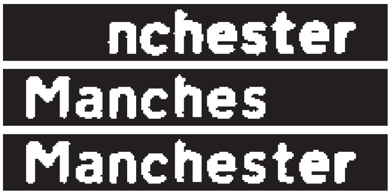


Fig. 19. Combination of word fragments, showing (top and middle) two word fragments and (bottom) the resulting combination of those word fragments.

are combined. A histogram of OCR results is created for each tracked word, with each word weighted by the recognition confidence rate returned by the OCR. At each frame, the result in the histogram of words, with the highest value, is taken to be the word for that frame. If the word is only recognized in a single frame, then it is ignored.

An example of our OCR result fusion method is shown in Table IV, with text read from the traffic sign shown in Fig. 18. It is worth noting in this example that, despite no single frame producing a perfectly accurate OCR result, the fused result is entirely correct. In addition to combining OCR results for exactly matching words, fragments of words are also combined. Occasionally, sections of words become temporarily unreadable due to occlusion or blurring. This is overcome by attempting to match together fragments of words, which overlap over successive frames.

If two words are found to overlap and have a similar height relative to the region size, an attempt is made to match the two words together. The two word fragments are overlapped iteratively, until a match is found between more than two of the characters, whereupon a new word is created from a combination of the existing words. An example of combined words is shown in Fig. 19.

V. Experimental Results

The proposed method currently runs at an average frame rate of 14 frames/s, under Open Source Computer Vision (OpenCV), on a 3.33-GHz Intel Core i5 CPU. A considerable increase in speed was gained by running the algorithm in parallel as a pipeline, although the system retains a latency of around 140 ms. Example outputs of the algorithm are shown in Fig. 20.

A. Comparative Analysis for the Detection Stage

To evaluate the performance of the detection stage of our system, comparative analysis was performed against two existing algorithms. These were the methods proposed by Reina *et al.* [16] and González *et al.* [35]. Since both methods were designed to recognize Spanish road signs, which are blue and white, it was necessary to adapt the algorithms to detect U.K. road signs, which also feature green and brown backgrounds. The method of González *et al.* [35] detected blue road signs as MSERs in the blue channel of a normalized red, green, and blue (RGB) image. Therefore, to extend this to green road signs, MSERs were also detected in the green channel. Brown road signs were detected as dark-on-light MSERs in a grayscale frame. The method of Reina *et al.* [16] uses hue, saturation, and intensity (HSI) thresholding to find candidate blobs for blue road signs. Therefore, additional thresholds were added to their method for the detection of brown and green road signs. These algorithms were optimized using the same validation data set used to develop our proposed method.

These data comprised nine video sequences, with a total of 23 130 frames, at a resolution of 1920 × 1088 pixels. The ground truth for detection was based on human observation; therefore, distant or heavily blurred traffic signs, which were unreadable by the eye, were ignored. The data set used for testing was entirely separate from the validation set used for the development and parameter tuning of the proposed and implemented systems.

Regions were determined to be candidate traffic signs, if detected for at least five subsequent frames. The results of this comparison are shown in Table V, where values for $Precision$, $Recall$, and F_{measure} were computed as

$$Precision = \frac{TP}{TP + FP} \tag{7}$$

$$Recall = \frac{TP}{TP + FN} \tag{8}$$

$$F_{\text{measure}} = 2 * \frac{Precision * Recall}{Precision + Recall} \tag{9}$$

where TPs are correctly detected text-based traffic signs, FPs are regions incorrectly identified as text-based traffic signs, and false negatives (FNs) are text-based traffic signs that the system failed to detect.

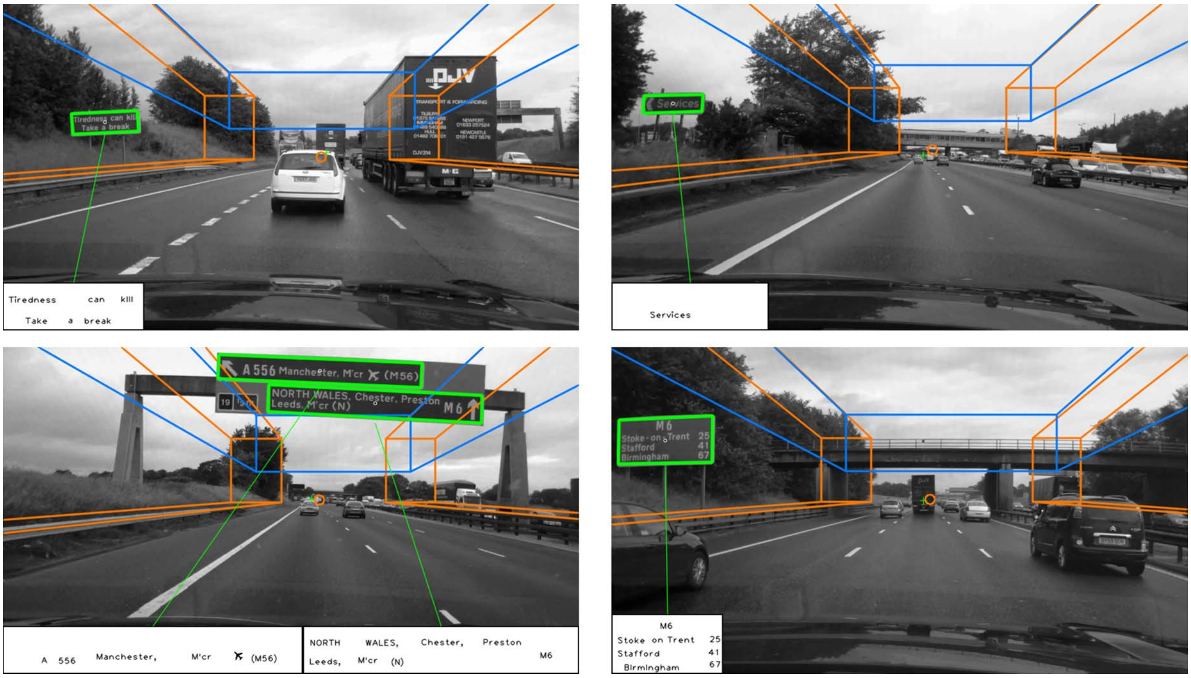


Fig. 20. Example outputs of the algorithm.

TABLE V
RESULTS OF COMPARATIVE ANALYSIS FOR THE DETECTION STAGE

Method	$Precision$	$Recall$	$F_{measure}$
Reina et al. [16]	0.58	0.64	0.61
González et al. [35]	0.54	0.68	0.60
Our method without use of structural information	0.19	0.90	0.31
Our method	0.96	0.90	0.93

TABLE VI
RESULTS FOR THE RECOGNITION STAGE

Method	$Precision$	$Recall$	$F_{measure}$
Standard Tesseract OCR	0.48	0.34	0.40
OCR with shape correction only	0.69	0.75	0.72
OCR with temporal fusing only	0.83	0.33	0.47
Full proposed method	0.87	0.91	0.89

It can be seen from these results that the proposed method achieves an F_{measure} of 0.93, whereas Reina *et al.* [16] and González *et al.* [35] reach the considerably lower values of 0.61 and 0.60, respectively. The use of geometrical, contextual, and temporal information in our system allows the total number of FPs to be reduced, thus increasing its precision and F_{measure}.

Also included in Table V are results for our text-based traffic sign detection method without the use of structural information. As expected, the resulting increase in FPs causes a huge reduction in precision, whereas the recall stays the same. In addition to reduced precision, the computational expense means that the frame rate drops from 14 frames/s to 6 frames/s. This is due to both the increased search area for candidate traffic signs and the slow down created by the increased number of detected FPs that it is necessary to process.

B. Performance of the Recognition Stage

To evaluate the performance of the recognition stage, $Precision$, $Recall$, and F_{measure} were computed based on the number of individual words correctly classified. For a word to be considered a TP, all characters must be correctly recognized in the correct letter case. If a single character is recognized incorrectly, then the entire word is considered to be an FP. Symbols such as "airport" were included in the training set merely to avoid their misclassification as characters, and are therefore classified as true negatives (TNs) when recognized, and have no effect on the result. There are 15 of these symbols in total, examples of which include directional arrows and the airport symbol.

We compare the results for OCR applied to a single instance of each detected traffic sign when the region was largest and most visible in the frame without any preprocessing, against OCR after application of our perspective correction method described in Section IV-A, OCR after application of our temporal fusion method described in Section IV-D, and then against OCR after the application of both perspective correction and the temporal fusion method. The results are presented in Table VI and show that use of our perspective recovery and temporal fusing methods vastly improve the recognition accuracy. It can be seen that temporal fusing improves the precision but not the recall; this is due to the rejection of low-confidence words by the system. The total system performance, i.e., for both detection and recognition stages, based on the number of individual words recognized in all traffic signs, in all video sequences, gave a precision value of 0.97, a recall value of 0.80, and an F_{measure} value of 0.87.

VI. CONCLUSION

A novel system for the automatic detection and recognition of text in traffic signs based on MSERs and HSV thresholding has been proposed. The search area for traffic signs was reduced using structural information from the scene, which aided in reducing the total number of FPs. Perspective rectification and temporal fusion of candidate regions of text were used to improve OCR results. Both the detection and recognition stages of the system were validated through comparative analysis, achieving the F_{measure} of 0.93 for detection, 0.89 for recognition, and 0.87 for the entire system.

ACKNOWLEDGMENT

The authors would like to thank Dr. T. Popham of Jaguar Land Rover Research for his support.

REFERENCES

[1] S. Maldonado-Bascón, S. Lafuente-Arroyo, P. Gil-Jimenez, H. Gomez-Moreno, and F. Lopez-Ferreras, "Road-sign detection and recognition based on support vector machines," *IEEE Trans. Intell. Transp. Syst.*, vol. 8, no. 2, pp. 264–278, Jun. 2007.

[2] F. Zaklouta and B. Stanciulescu, "Real-time traffic-sign recognition using tree classifiers," *IEEE Trans. Intell. Transp. Syst.*, vol. 13, no. 4, pp. 1507–1514, Dec. 2012.

[3] J. Greenhalgh and M. Mirmehdi, "Traffic sign recognition using MSER and random forests," in *Proc. EUSIPCO*, Aug. 2012, pp. 1935–1939.

[4] J. Greenhalgh and M. Mirmehdi, "Real-time detection and recognition of road traffic signs," *IEEE Trans. Intell. Transp. Syst.*, vol. 13, no. 4, pp. 1498–1506, Dec. 2012.

[5] A. Møgelmose, M. M. Trivedi, and T. B. Moeslund, "Vision-based traffic sign detection and analysis for intelligent driver assistance systems: Perspectives and survey," *IEEE Trans. Intell. Transp. Syst.*, vol. 13, no. 4, pp. 1484–1497, Dec. 2012.

[6] M. A. García-Garrido *et al.*, "Complete vision-based traffic sign recognition supported by an I2V communication system," *Sensors*, vol. 12, no. 2, pp. 1148–1169, Jan. 2012.

[7] P. Clark and M. Mirmehdi, "Recognising text in real scenes," *Int. J. Document Anal. Recog.*, vol. 4, no. 4, pp. 243–257, Jul. 2002.

[8] C. Merino and M. Mirmehdi, "A framework towards real-time detection and tracking of text," in *Proc. CBDAR*, 2007, pp. 10–17.

[9] S. Hanif and L. Prevost, "Text detection and localization in complex scene images using constrained AdaBoost algorithm," in *Proc. ICDAR*, Jul. 2009, pp. 1–5.

[10] L. Neumann and J. Matas, "A method for text localization and recognition in real-world images," in *Proc. ACCV*, 2010, pp. 9–11.

[11] B. Epshtein, E. Ofek, and Y. Wexler, "Detecting text in natural scenes with stroke width transform," in *Proc. CVPR*, 2010, pp. 2963–2970.

[12] J.-J. Lee, P.-H. Lee, S.-W. Lee, A. Yuille, and C. Koch, "AdaBoost for text detection in natural scene," in *Proc. ICDAR*, Sep. 2011, pp. 429–434.

[13] C. Yi and Y. Tian, "Text string detection from natural scenes by structure-based partition and grouping," *IEEE Trans. Image Process.*, vol. 20, no. 9, pp. 2594–2605, Sep. 2011.

[14] J. Zhang and R. Kasturi, "Character energy and link energy-based text extraction in scene images," in *Proc. ACCV*, 2010, no. 2, pp. 308–320, Springer.

[15] W. Wu, X. Chen, and J. Yang, "Detection of text on road signs from video," *IEEE Trans. Intell. Transp. Syst.*, vol. 6, no. 4, pp. 378–390, Dec. 2005.

[16] A. Reina, R. Sastre, S. Arroyo, and P. Jiménez, "Adaptive traffic road sign panels text extraction," in *Proc. WSEAS ICSPRA*, 2006, pp. 295–300.

[17] A. González, L. Bergasa, and J. Yebes, "Text detection and recognition on traffic panels from street-level imagery using visual appearance," *IEEE Trans. Intell. Transp. Syst.*, vol. 15, no. 1, pp. 228–238, Feb. 2014.

[18] X. Chen, J. Yang, J. Zhang, and A. Waibel, "Automatic detection and recognition of signs from natural scenes," *Trans. Image Process.*, vol. 13, no. 1, pp. 87–99, Jan. 2004.

[19] C. Merino-Gracia, K. Lenc, and M. Mirmehdi, "A head-mounted device for recognizing text in natural scenes," in *Proc. CBDAR*, 2011, vol. 7139, pp. 29–41.

[20] J. Shi and C. Tomasi, "Good features to track," in *Proc. CVPR*, 1994, pp. 593–600.

[21] S. Lafuente-Arroyo, S. Maldonado-Bascon, P. Gil-Jimenez, H. Gomez-Moreno, and F. Lopez-Ferreras, "Road sign tracking with a predictive filter solution," in *Proc. IECON*, Nov. 2006, pp. 3314–3319.

[22] H. Goto and M. Tanaka, "Text-tracking wearable camera system for the blind," in *Proc. ICDAR*, 2009, pp. 141–145.

[23] H. Shiratori, H. Goto, and H. Kobayashi, "An efficient text capture method for moving robots using DCT feature and text tracking," in *Proc. ICPR*, 2006, pp. 1050–1053.

[24] M. Tanaka and H. Goto, "Autonomous text capturing robot using improved DCT feature and text tracking," in *Proc. ICDAR*, Sep. 2007, pp. 1178–1182.

[25] S. Scherer, D. Dube, P. Komma, and A. Masselli, "Robust real-time number sign detection on a mobile outdoor robot," in *Proc. ECMR*, 2011, pp. 1–7.

[26] S. Houben, J. Stallkamp, J. Salmen, M. Schlipsing, and C. Igel, "Detection of traffic signs in real-world images: The German traffic sign detection benchmark," in *Proc. IJCNN*, Aug. 2013, pp. 1–8, IEEE.

[27] J. Stallkamp, M. Schlipsing, J. Salmen, and C. Igel, "The German Traffic Sign Recognition Benchmark: A multi-class classification competition," in *Proc. IJCNN*, Jul. 2011, pp. 1453–1460.

[28] R. Timofte, K. Zimmermann, and L. V. Gool, "Multi-view traffic sign detection, recognition, 3D localisation," in *Proc. WACV*, Dec. 2009, pp. 1–8.

[29] D. Hanwell and M. Mirmehdi, "Detection of lane departure on high-speed roads," in *Proc. ICPRAM*, 2009, pp. 529–536.

[30] "Traffic Signs Manual," London, U.K., 2013.

[31] M. Fischler and R. Bolles, "Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography," *Commun. ACM*, vol. 24, no. 6, pp. 381–395, Jun. 1981.

[32] R. I. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision,* 2nd ed. Cambridge, U.K.: Cambridge Univ. Press, 2004.

[33] Google, Tesseract-OCR 2013, [Online; accessed 8-October-2013]. [Online]. Available: http://code.google.com/p/tesseract-ocr/

[34] The Department for Transport, "Working drawings for traffic signs," London, U.K., Oct. 2013. [Online]. Available: https://www.gov.uk/working-drawings-for-traffic-signs

[35] A. Gonzalez, L. M. Bergasa, J. J. Yebes, and J. Almazan, "Text recognition on traffic panels from street-level imagery," in *Proc. IVS*, Jun. 2012, pp. 340–345.

Jack Greenhalgh received the M.Eng. degree in computer systems engineering from University of Sussex, Brighton, U.K., in 2010. He is currently working toward the Ph.D. degree on the subject of "driver assistance using automated sign and text recognition" with the Department of Computer Science, University of Bristol, Bristol, U.K.

His research interests include image processing, computer vision, machine learning, and intelligent transportation systems.

Majid Mirmehdi received the B.Sc. (Hons) and Ph.D. degrees in computer science from City University, London, U.K., in 1985 and 1991, respectively.

He is a Professor of computer vision with the Department of Computer Science, University of Bristol, Bristol, U.K., where he is also the Graduate Dean, Faculty of Engineering. His research interests include natural scene analysis and medical imaging, and he has more than 150 refereed conference and journal publications in these and other areas.

Dr. Mirmehdi is a Fellow of the International Association for Pattern Recognition. He is a member of The Institution of Engineering and Technology (IET) and serves on the Executive Committee of the British Machine Vision Association. He is the Editor-in-Chief of *IET Computer Vision Journal* and an Associate Editor of *Pattern Analysis and Applications Journal*.