

# Automatic Detection and Recognition of Symbols and Text on the Road Surface

Jack Greenhalgh<sup>(✉)</sup> and Majid Mirmehdi

Visual Information Laboratory, University of Bristol, Bristol, UK  
{csjhg,m.mirmehdi}@bristol.ac.uk  
<http://www.bris.ac.uk/vi-lab>

**Abstract.** This paper presents a method for the automatic detection and recognition of text and symbols on the road surface, in the form of painted road markings. Candidates for road markings are detected as maximally stable extremal regions (MSER) in an inverse perspective mapping (IPM) transformed version of the image, which shows the road surface with the effects of perspective distortion removed. Separate recognition stages are then used to recognise words and symbols. Recognition of text-based regions is performed using a third-party optical character recognition (OCR) package, after application of a perspective correction stage. Symbol-based road markings are recognised by extracting histogram of oriented gradient (HOG) features, which are then classified using a support vector machine (SVM) classifier. The proposed method is validated using a data-set of videos, and achieves F-measures of 0.85 for text characters and 0.91 for symbols.

**Keywords:** Computer vision · Machine learning · Text recognition · Intelligent transportation systems

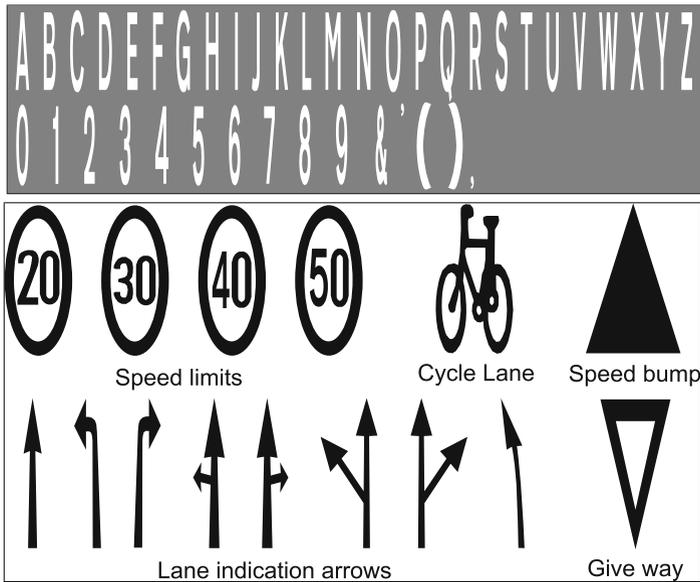
## 1 Introduction

Painted text and symbols which appear on the surface of roads in the UK come in three basic forms: text, symbols, and lane division markers. Text provides information to the driver, such as speed limits, warnings, and directions. Symbol-based road markings are displayed in the form of arrows and other ideograms. We propose a method in this paper that detects and recognises such painted text and symbols using a camera mounted inside a car on the driver's rear-view mirror (looking out front). The possible applications for this work are numerous, and include advanced driver assistance systems (ADAS), autonomous vehicles, and surveying of road markings. While information such as the current speed limit or upcoming turnings are also provided in the form of road signs, there is much information, such as 'road merging' warnings or lane specific directional information which may appear exclusively as road markings.

There are several key issues which make the detection and recognition of painted road text and symbols difficult. Road markings suffer badly from wear

and deterioration due to the fact that vehicles continuously pass over them, and there is also a large amount of variation between different instances of the same symbol or character, as they are often hand-painted. In addition, common issues such as shadowing, occlusion, and variations in lighting also apply. Despite such difficulties, the problem is constrained in some aspects, with a large amount of a priori knowledge which can be exploited. Much of this a priori knowledge relates to the visual appearance of the text characters and symbols themselves. Text and symbol based road markings appear elongated when viewed from above, so as to improve their readability from the view-point of a driver on the road. This offers a constraint on the size and aspect ratio of candidate regions, when detected in an image that has undergone inverse perspective mapping (IPM).

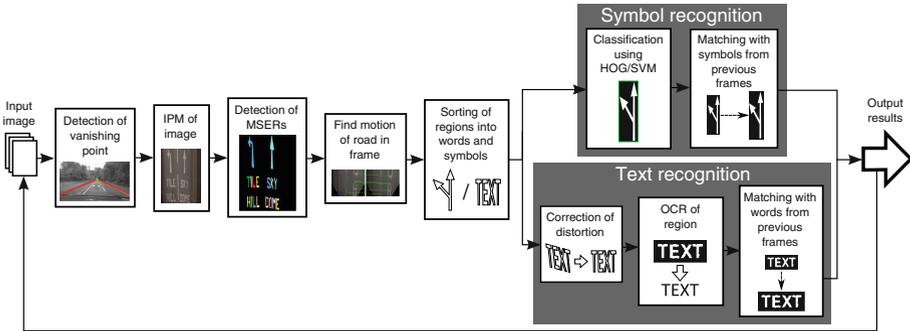
The text-based aspect of road marking recognition is constrained in comparison with more generalised text recognition [1,2]. The total set of text characters is very limited, consisting only of upper-case characters, numbers, and a small number of punctuation marks. Also, text-based road markings are based on a single typeface, as shown in Fig. 1(top). The total number of symbol-based road markings is also fairly limited, with only 15 different symbols appearing in the data-set, as shown in Fig. 1(bottom).



**Fig. 1.** (Top) full character set for typeface on which text road markings are based (bottom) full set of road marking symbols which appear in the data-set.

The first stage of the algorithm removes perspective distortion from the road surface by applying an IPM, after the vanishing point has been detected. Candidate text characters and symbols are then detected as maximally stable extremal regions (MSER) on the road surface. Road marking candidates are sorted into

potential text characters or symbols based on their relative size and proximity, so that they can be classified using separate recognition stages. Symbol-based markings are recognised using histogram of oriented gradients (HOG) and linear support vector machines (SVM). Text words are recognised using an open-source optical character recognition (OCR) engine, Tesseract [3], after a further correction transform has been applied. Recognised words and symbols are matched across consecutive frames so that recognition results can be improved via temporal fusion. The total system pipeline for the algorithm is shown in Fig. 2.



**Fig. 2.** Pipeline for the proposed painted road markings algorithm.

In Sect. 2, an overview of related work is provided. Section 3 describes the stage for the detection and sorting of candidate regions. Sections 4 and 5 focus on the recognition steps for text and symbols, respectively. In Sect. 6, the temporal aspects of the method are described. In Sect. 7, experimental results are presented. Finally, in Sect. 8, conclusions are drawn.

## 2 Related Work

Research on road marking detection can be broadly divided into two categories, one of which focusses on lane division markings, such as [4–8], and the other on symbol or text based markings, such as [9–14], which provide semantic information to the driver.

In cases where symbols painted on the road surface are detected and recognised, the total number of symbol types which are classified is generally very limited, often focussing on just arrows or rectangular elements [9–11, 13]. For symbol detection, several of these works employ an IPM to remove perspective distortion of the road surface, and hence the markings painted on it, such as [9, 11, 12, 14]. The only papers that deal with the recognition of road surface text are [12, 14].

[12] present a method for detecting and recognising both text and symbols on the road surface. An IPM is applied to each frame, after the image vanishing point (VP) has been automatically detected. Regions of interest (ROI) are then

detected in the IPM image by applying an adaptive threshold, and finding CCs in the resulting binary image. After applying some post-processing to the detected shapes, such as orientation normalisation and rejection of complex shapes, the region is classified. The recognition stage involves the extraction of a feature vector from each candidate CC, which includes several shape based features. Each region is then classified using a neural network trained using real road footage. An accumulator of symbols is used to combine results over several frames, and eliminate single frame false positives. The method is limited to recognising only 7 symbols and 16 characters rather than the full alphabet, and is also limited to recognising only 19 unique predefined words. The authors report true positive rates of 85.2% and 80.7% for recognition of arrows and text, respectively, with their method taking 60–90 ms to process a single frame.

[14] propose a method for the detection and classification of text and symbols painted on the road surface. ROIs are detected in each frame as MSERs in an IPM transformed version of the image. The FAST feature detector is then used to extract points of interest (POI) from each ROI. A feature vector is then found for each POI using HOG, and the region is classified through comparison with a set of template images. Although Wu and Ranganathan recognise both text and symbols using template matching, entire words are treated as single classes, and as a result only a small subset of words are recognised. In this respect, their proposed method does not provide ‘true’ text detection, as arbitrary words (such as place names and their abbreviations) are not recognised. The authors report a true positive rate of 90.1% and a false positive rate of 0.9% for the combined recognition of arrows and text, at a processing speed of at least 10 frames per second.

The method proposed in this paper improves upon the current state-of-the-art in several ways. Firstly, the proposed algorithm is able to recognise any arbitrary text word which may appear, rather than the limited set of predefined words recognised by the methods presented by [12, 14]. This aspect is essential for use with an ADAS or autonomous vehicle, given that important directional information may appear in the form of obscure place names, often with abstract abbreviations, which may not appear in the dictionary. For example, the place name ‘Warwick’ is sometimes abbreviated to ‘W’wick’. Secondly, the range of road symbols recognised is expanded upon compared to existing works which classify up to a maximum of 6 symbols only, e.g. [11–14]. This is an advantage of using a synthetically generated data-set, which allows the system to be trained on any number of template images while retaining the robustness provided by machine learning methods.

### 3 Detection and Sorting of Candidate Regions

The first stage of the proposed method detects and sorts candidate regions for road markings. Connected components (CC) representing text characters and symbols are detected in an IPM version of the image, and are then sorted into text characters and symbols based on their attributes, and are passed on to the appropriate classification stage.

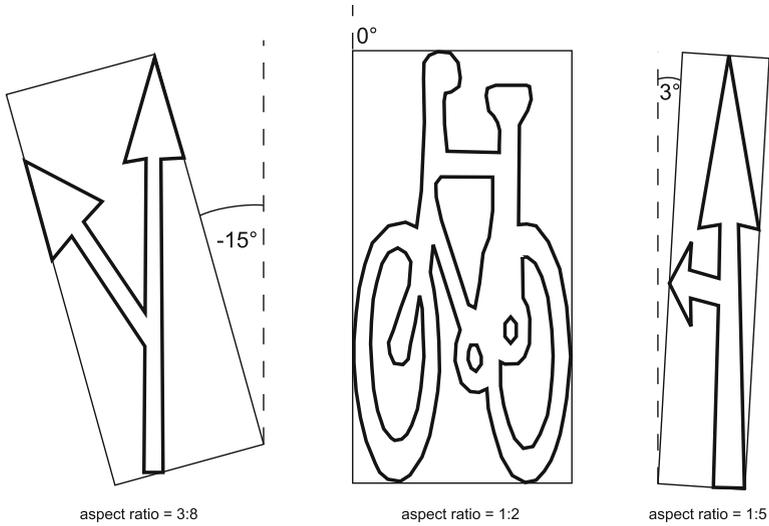
### 3.1 Detection of Candidate Regions

Candidate regions are first detected as MSERs in an IPM transformed version of the input frame. Use of IPM eliminates issues caused by perspective distortion, as text and symbols maintain their shape and scale in this transformed image, regardless of their shape and scale in the original frame. Given that road markings appear as high contrast homogeneous regions on the road surface, MSER is considered to be a suitable method for their detection, especially following its successful application to the detection of road signs in [15,16]. Only light-on-dark MSERs are used, given that road markings are always painted in white paint on dark backgrounds. Figure 3 shows an example frame, IPM image, and detected MSERs, which are individually coloured.



**Fig. 3.** Stages of MSER detection on IPM image, showing original frame (top), IPM image (bottom left), and detected MSERs individually coloured (bottom right) (Color figure online).

A rotated minimum area rectangle (RMAR) is fitted to the CC of each candidate region, the features of which are useful for reducing the total number of candidates. As text and symbols painted on the road are elongated, the angle of their RMAR is expected to be close to  $0^\circ$ , this also places a constraint on the range of aspect ratios which will appear, as illustrated in Fig. 4.



**Fig. 4.** Dimensions of symbol-based road markings.

The features used for the elimination of false positives are shown in Table 1, with all the values determined empirically through analysis of template images.

### 3.2 Sorting of Candidates into Words and Symbols

Once road marking candidates have been detected, they can be divided into words and symbols. Text characters contained within words appear in horizontal lines and in close proximity to each other, whereas symbol-based road markings appear in isolation with a greater amount of space around them. Based on this observation, it is possible to sort road marking candidates into text characters and symbols.

First, each candidate region is compared to each other on the basis of size and location. If two candidates are found to be of a similar height, are approximately vertically aligned, and have a small horizontal distance, they are grouped into a single word. To assess the height similarity, a constraint is applied to the ratio of the two heights. To determine whether two characters are vertically aligned, an empirically determined threshold is applied to an overlap measure,  $\psi$ , which is calculated as follows:

**Table 1.** Features for reduction of candidates.

Feature	Min. value	Max. value
Ratio of area of CC to area to RMAR	0.17	0.7
Aspect ratio of RMAR	0.09	0.68
Angle of RMAR	-20°	20°

$$\psi = \frac{\min\{E_{y2}, F_{y2}\} - \max\{E_{y1}, F_{y1}\}}{\max\{E_{y2}, F_{y2}\} - \min\{E_{y1}, F_{y1}\}}, \quad (1)$$

where  $E$  and  $F$  represent the bounding boxes of the two candidate regions. If there is no vertical overlap between the regions,  $\psi$  will be equal to 0, and if the regions perfectly overlap it will be equal to 1.

The distance between two characters from the same word is expected to be less than approximately one-third of the width of the widest character, which was determined based on examples from the character set shown in Fig. 1(top). Any candidates grouped together are considered to be text characters. The requirements for the matching of text characters are summarised in Table 2, where all of the described features are invariant to scale.

**Table 2.** Scale invariant character grouping constraints.

Feature	Min.	Max.
Character height ratio	0.8	1.25
$\psi$	0.7	1.00
Horizontal character distance	0	(max. character width)*0.35

After the character grouping process, any remaining un-grouped candidate regions are considered to be possible candidates for road symbols. Candidate regions grouped into words are sent to the text recognition stage of the algorithm, and the remaining characters are classified by the symbol recognition stage.



**Fig. 5.** Algorithm output showing candidates for road markings divided into symbols and words (Color figure online).

An example output for this stage of the algorithm is shown in the right-hand image of Fig. 5, where candidate regions are shown in white, bounding boxes for individual words are shown in orange, and bounding boxes for symbols are shown in green.

## 4 Recognition of Words

Next, the proposed method attempts to recognise text contained within candidate word regions. An affine transform is applied to minimise the distortion caused by uneven roads, before the region is interpreted using OCR.

### 4.1 Correction of Perspective Distortion of Words

The IPM stage of the algorithm works on the assumption that the road surface is flat, however, in reality road surfaces often exhibit a slight camber. As a result, words appearing in the IPM road image may sometimes suffer from rotation and shearing, examples of which are shown in the top row of Fig. 6. This perspective distortion reduces the accuracy achieved by the OCR engine, given that it is designed for use with fronto-parallel text.



**Fig. 6.** Examples of distorted words (top row) with corrected versions (bottom row).

To avoid this distortion, a correction stage is applied to candidate words before recognition. First a RMAR is fitted to all the CCs representing the characters in the word. The angle of the RMAR,  $\zeta$ , is then used to correct the rotation of the region, using the following transform:

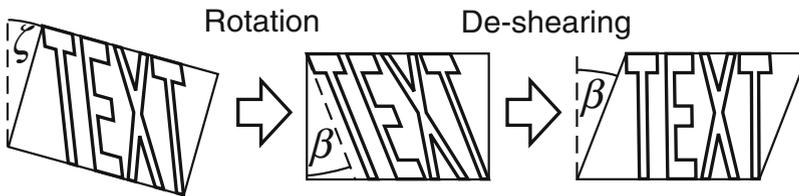
$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} \cos(\zeta) & -\sin(\zeta) \\ \sin(\zeta) & \cos(\zeta) \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}, \quad (2)$$

where  $x$  and  $y$  are pixel coordinates in the original image, and  $x'$  and  $y'$  are pixel coordinates in the rotated image.

In order to fix the shear of the image, the most common edge orientation is found, and used to calculate the shear mapping. The Sobel filter is used to find the horizontal and vertical derivatives of the image, and from these the magnitude and orientation is found at each pixel. A histogram of orientations is then built, with each pixel weighted by its magnitude. The histogram bin with the highest value is taken to be perpendicular to the shear angle,  $\beta$ , and can be used to calculate the transformation as follows:

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} 1 & \tan(\beta) \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}. \quad (3)$$

The stages of this process are shown in Fig. 7. Examples of this transformation applied to several images are shown in Fig. 6.



**Fig. 7.** Diagram illustrating stages of perspective distortion correction.

## 4.2 OCR of Region

Once the correction stage has been applied to the region, the pixel values are normalised and the textual information is interpreted using the open source OCR engine *Tesseract* [3]. The OCR engine has been retrained based on a road marking typeface, which contains only upper-case characters, numbers, and punctuation marks (see Fig. 1(top) for the full character set). When OCR is applied, a value is returned which represents the confidence of recognition, and if this value is below 50%, the entire region is rejected.

## 5 Recognition of Symbols

Once candidate regions have been detected and sorted, the candidates previously selected as possible symbol road markings are classified by a recognition stage. For the classification of symbol-based markings an approach similar to the one described in [15] for the recognition of road signs is used, whereby regions are recognised with HOG descriptors in combination with a classifier. This feature descriptor is deemed to be suitable for encoding the shape of road markings, given that they display clearly defined edges at particular orientations within the image. Each candidate region is resized to  $32 \times 192$ , before a HOG descriptor is calculated. This size is empirically determined to strike a balance between

accuracy and efficiency in computation expense. A synthetically generated data-set is used to train the classifier. This data-set is created by applying distortions to a set of template images, to create images which closely resemble the road markings featured in the data-set. This approach eliminates the necessity for a large training set of real images, and allows the system to be trained on any number of symbol classes.

### 5.1 Selection of Classifier

To select the most efficient classifier for use with the HOG features a comparison between several was made. Classifiers considered included SVM with a linear kernel, SVM with a radial basis function (RBF) kernel, MLP, and Random Forests. Each classifier was trained on a synthetically generated data-set consisting of 1000 examples per symbol class, and a negative class of 5000 examples. The classifiers were then tested on a validation set of real images, the results of which are shown in Table 3. As can be seen from the table, RBF SVM outperforms the other three classification methods, but suffers from a larger classification time. This is likely due to a combination of the mapping process when applied to the data, given its high dimensionality, and the fact that a multi-class SVM classifier is made up of many one-against-one binary classifiers. Although 7.41 ms is not a significant amount of time for a single classification, recognition must be performed on many candidate regions in each frame, therefore making the RBF-kernel SVM potentially unusable for this application. Linear SVM produces the second highest accuracy of all classifiers tested, and also retains a low classification time. Linear SVM is much faster than RBF SVM as the mapping process is removed. Random Forests and MLP have a much lower classification time than RBF SVM, but also suffer from a much lower accuracy. This is largely due to their inability to separate the symbol classes from the background/negative class. The classifier selected was the linear-kernel SVM, due to its high accuracy and fast classification time.

Road markings exhibit a large amount of variation within classes, due to the fact that they are hand painted, and are also subject to greater amounts of deterioration. SVM is able to avoid the problem of over-fitting through careful selection of training parameters. Therefore, for this particular problem SVM proved to be the more suitable classifier.

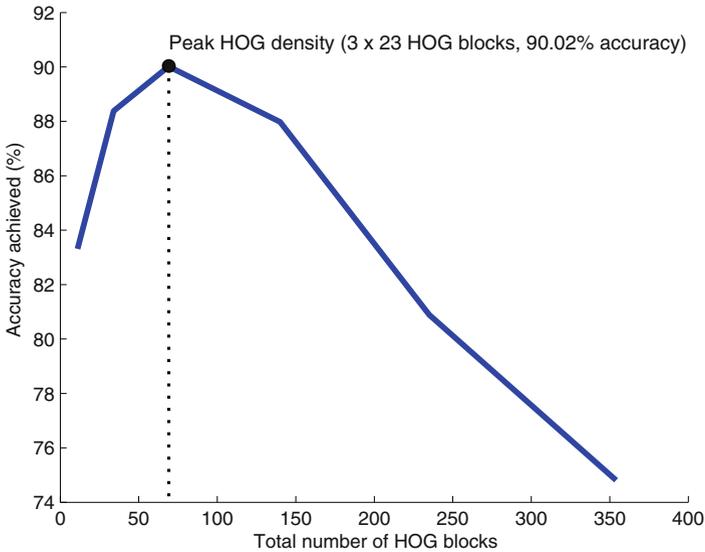
### 5.2 Selection of HOG Block Density

For the calculation of HOG features, the optimal block density varies depending on the type of data which is to be classified. This value can be changed by altering the size of each cell, whereby a smaller cell size provides a higher block density. Lower block densities are preferable in cases where the object to be classified is geometrically varied, such as animals or cars. Higher block densities are more applicable to more constrained objects, such as road markings [17]. A comparison is performed to find the most appropriate HOG block density. An SVM classifier is trained on synthetic data with HOG features calculated at

**Table 3.** Comparison of different classification methods.

Classifier	Accuracy	Average classification time (ms)
SVM (RBF kernel)	93.54 %	7.41
SVM (linear kernel)	90.87 %	0.36
Random forest	75.58 %	0.26
MLP	71.40 %	0.81

number of block density levels, as shown in Fig. 8. The accuracy is then found for each by testing on the same validation test set used in Sect. 5.1. The results of this comparison are shown in the graph in Fig. 8.

**Fig. 8.** Graph showing number of HOG feature blocks against accuracy achieved.

It can be seen from the results in Fig. 8 that the peak block density is found at  $3 \times 23$ . Once the density increases beyond this value, the accuracy begins to decrease, as a result of intraclass variance.

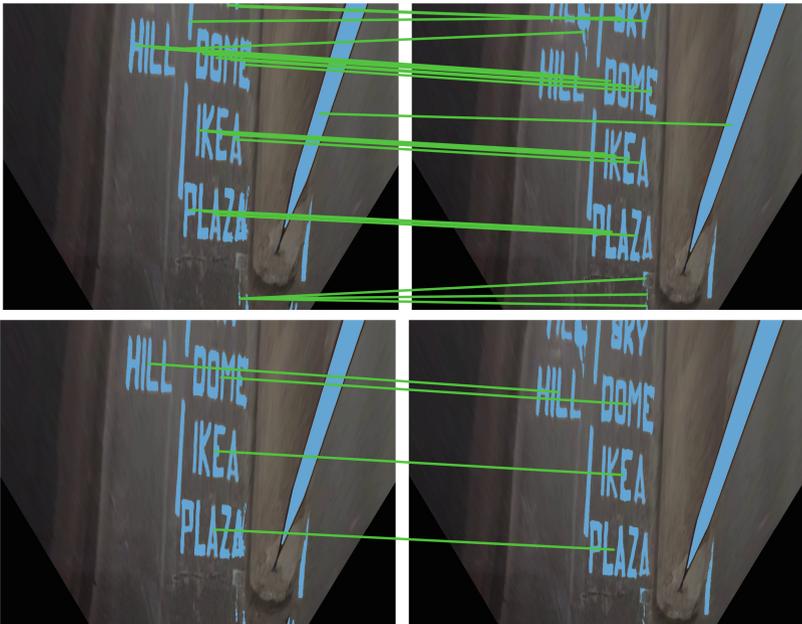
## 6 Temporal Information

Temporal information is exploited in order to improve the accuracy of the method. Candidate words and symbols from consecutive frames are matched, and recognition results from several frames are then fused together.

## 6.1 Calculation of Road Movement

To improve the temporal matching of candidate regions, the motion of the road between consecutive IPM frames is estimated. This movement can be represented by a single 2-D motion vector, which is found by matching MSERs between frames. Use of this 2-D vector allows the current location of previously detected road markings to be estimated, therefore improving the accuracy of matching.

All detected MSERs (symbol, word, or otherwise) from the previous frames are matched to those in the current frame by finding their ‘nearest neighbours’ based on a descriptor vector consisting of a number of simple features, including width, height, and location. Each MSER in the current frame is compared to each one in the previous frame, and matched to the one with smallest Euclidean distance between feature vectors. A 2-D motion vector for the frame is then calculated by finding the mean of the distances between the centroids of each pair of matched regions. MSERs are well suited to image matching applications such as this due to their high repeatability [18], and conveniently have already been detected in the frame during the road-marking detection process. The top row of Fig. 9 shows two consecutive frames with MSERs marked in blue and matches shown with green lines.



**Fig. 9.** (Top row) Consecutive frames of IPM image showing matched MSERs, (Bottom row) Consecutive frames of IPM image showing matched words (Color figure online).

## 6.2 Matching and Temporal Fusion of Words and Symbols

The next stage of the algorithm finds matches between words and symbols in the current frame and previous frame using the road movement vector. Although matching is performed in the previous stage to find the road movement vector, this process does not necessarily match all MSERs correctly. This can be seen in the top-row of Fig. 9 where a text character has been mismatched. Once the road movement vector has been calculated more accurate matching can be performed.

Each current symbol or word is compared to each previously detected symbol or word by computing the Euclidean distance between a set of features. These features include the size and aspect ratio of the RMAR, and the location of the corners of the RMAR once they have been displaced by the calculated road motion vector described above in Sect. 6.1. Each candidate is matched to the previous region with the smallest Euclidean distance between feature vectors. If no match is found, the region is treated as a new candidate. An example of this matching can be seen in the bottom row of Fig. 9 where each text word is matched correctly.

For each series of matched words a histogram is built containing all OCR results, weighted by confidence. The histogram bin with the highest value is taken as the correct result, given that it has been recognised in at least 3 frames. Similarly, a histogram is created for each series of matched symbols, with each bin representing a class of symbol.

**Table 4.** Results for recognition of text characters.

Method	Precision	Recall	F-measure
OCR with no additional processing	0.66	0.79	0.72
OCR with perspective correction	0.66	0.81	0.73
OCR with temporal fusion only	0.79	0.83	0.81
OCR with perspective correction and temporal fusion	0.86	0.87	0.85

## 7 Results

We validate both the symbol-based and text-based aspects of the proposed method via comparative analysis on a data-set, consisting of 42,110 frames at a frame rate of 30 fps and a resolution of  $1920 \times 1088$ . This test data was shot at a number of different vehicle speeds, featuring different scene types which included urban, suburban, and motorway scenes. The results were obtained on a 3.33 GHz Intel Core i5 CPU under OpenCV. The algorithm originally ran at an average rate of 3 fps on a single core. However, this speed was increased to 7.25 fps exploiting the multi-core capability of the processor and implementing the method as a parallel pipeline.

## 7.1 Results for Text Detection and Recognition

In order to validate the various stages of the text recognition aspect of the system, recognition results are provided showing the precision, recall, and F-measure, for the detection and recognition of text characters. We present results which compare standard OCR applied to detected regions, OCR with the perspective correction method described in Sect. 4.1, OCR with the temporal fusion method described in Sect. 6, and OCR with both methods. In this manner, the difference in performance which results from the addition of each of these enhancements can be seen. These results are shown in Table 4.

The results demonstrate that the use of perspective correction and temporal fusion provide a significant improvement compared to the raw OCR approach. It can be seen that the recall result for OCR with perspective correction and temporal fusion is higher than precision, likely due to the variation seen in the text characters to be recognised. While individual characters are similar enough to be reliably detected, intraclass variation between characters is large, causing misclassification, and hence, a lower precision. This intraclass variation is due to the fact that the text characters are largely hand painted. Figure 12 contains example output frames of the algorithm, showing road text correctly detected and recognised (Fig. 10).



Fig. 10. Examples of system output showing correctly detected and recognised text.

## 7.2 Results for Symbol Detection and Recognition

Our proposed algorithm is compared against an existing method proposed by [12] as well as a baseline method in order to validate its performance for the recognition of symbols on the road surface.

The method of [12] was implemented and adapted to detect and recognise symbols on UK roads. An artificial neural network was trained with their proposed feature set, which was extracted from the same synthetic data-set used to train our method.

The baseline method classifies each region using simple template matching. Each detected region is resized to  $32 \times 192$  before its pixels are normalised between 0 and 255. The region is then compared to a number of template images and the template with the smallest difference is selected as the correct class.

From the results shown in Table 5 it can be seen that the proposed method outperforms both the baseline method and the method of [12] (shown in the table as KB2010). The adaptive threshold approach of [12] was unable to detect some of the fainter road markings, causing the recall to be reduced. The recognition stage which used an artificial neural network with shape based features misclassified several non-road marking shapes as road symbols, causing a reduction in precision. The baseline method operates at 1.3 fps, KB2010 at 13.9 fps (average rate), and the proposed method at 7.3 fps.

The baseline method manages a particularly low recall rate, in contrast to the ability of the proposed symbol recognition method in eliminating false positives. Example frames demonstrating the output of the symbol-based stage of the algorithm are shown in Fig. 11.

**Table 5.** Results for recognition of symbols.

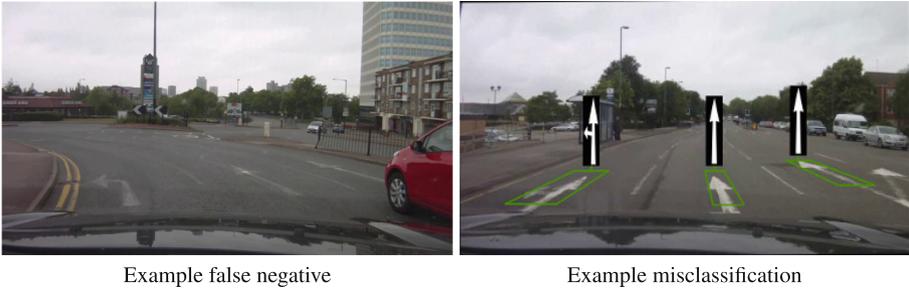
Method	Precision	Recall	F-measure
Baseline method	0.58	0.65	0.61
KB2010	0.74	0.78	0.76
Proposed method	0.91	0.92	0.91



**Fig. 11.** Examples of algorithm output showing correctly detected and recognised symbols.

### 7.3 Failure Cases

There were several cases where the detection and recognition of road markings failed. Some examples of these failure cases are shown in Fig. 12. In the example shown on the left-hand side of Fig. 12 it can be seen that the 3 symbol-based road markings have failed to be detected. This is due to the fact that the road markings are faded, and appear very faint. MSER detects high-contrast regions, therefore, the fact that these regions are so faint has resulted in their missed detection. The road marking shown in the right-hand image of Fig. 12 shows a misclassified symbol on the far right hand side, where part of an arrow pointing in two directions has been falsely recognised as a single arrow pointing forward. In this case, the road marking has deteriorated in such a way that the CC representing the region is split, which has caused a section of the symbol to be classified separately.



**Fig. 12.** Examples where symbol recognition failed.

## 8 Conclusion

A method for the automatic detection and recognition of painted text and symbols on the road surface is proposed. Candidate regions are detected in an IPM transformed image, which is then reduced and sorted into words and symbols before being interpreted using separate recognition stages. Text words are recognised using an open-source OCR engine, after a perspective correction stage. Symbols are classified using HOG features and SVM. Temporal fusion is used to combine recognition results across several frames. Results are provided showing precision, recall, and F-measure for a challenging data-set of videos, for which the system produces F-measures of 0.85 and 0.91 for text characters and symbols, respectively.

## References

1. Clark, P., Mirmehdi, M.: Recognising text in real scenes. *Int. J. Doc. Anal. Recogn.* **4**, 243–257 (2002)
2. Merino-Gracia, C., Lenc, K., Mirmehdi, M.: A head-mounted device for recognizing text in natural scenes. In: Iwamura, M., Shafait, F. (eds.) *CBDAR 2011*. LNCS, vol. 7139, pp. 29–41. Springer, Heidelberg (2012)
3. Google: Tesseract-OCR (2013). <http://code.google.com/p/tesseract-ocr/>. Accessed 8 October 2013
4. Hanwell, D., Mirmehdi, M.: Detection of lane departure on high-speed roads. In: *International Conference on Pattern Recognition Applications and Methods* (2009)
5. Chen, Z., Ellis, T.: Automatic lane detection from vehicle motion trajectories. In: *2013 10th IEEE International Conference on Advanced Video and Signal Based Surveillance*, pp. 466–471. IEEE (2013)
6. Bottazzi, V.S., Borges, P.V.K., Jo, J.: A vision-based lane detection system combining appearance segmentation and tracking of salient points. In: *2013 IEEE Intelligent Vehicles Symposium (IV), Number IV*, pp. 443–448. IEEE (2013)
7. Zhang, F., Stähle, H., Chen, C., Buckl, C., Knoll, A.: A lane marking extraction approach based on random finite set statistics. In: *2013 IEEE Intelligent Vehicles Symposium (IV)*, pp. 1143–1148 (2013)

8. Huang, J., Liang, H., Wang, Z.: Robust lane marking detection under different road conditions. In: 2013 IEEE International Conference on Robotics and Biomimetics (ROBIO), vol. 1, pp. 1753–1758 (2013)
9. Rebut, J., Benschrair, A., Toulminet, G.: Image segmentation and pattern recognition for road marking analysis. In: 2004 IEEE International Symposium on Industrial Electronics, vol. 1, pp. 727–732. IEEE (2004)
10. Vacek, S., Schimmel, C., Dillmann, R.: Road-marking analysis for autonomous vehicle guidance. In: European Conference on Mobile Robots, pp. 1–6 (2007)
11. Li, Y., He, K., Jia, P.: Road markers recognition based on shape information. In: 2007 IEEE Intelligent Vehicles Symposium, Number c, pp. 117–122 (2007)
12. Kheyrollahi, A., Breckon, T.P.: Automatic real-time road marking recognition using a feature driven approach. *Mach. Vis. Appl.* **23**, 123–133 (2010)
13. Danescu, R., Nedevschi, S.: Detection and classification of painted road objects for intersection assistance applications. In: 2010 13th International IEEE Annual Conference on Intelligent Transportation Systems, Number 28, pp. 433–438 (2010)
14. Wu, T., Ranganathan, A.: A practical system for road marking detection and recognition. In: 2012 IEEE Intelligent Vehicles Symposium, pp. 25–30. IEEE (2012)
15. Greenhalgh, J., Mirmehdi, M.: Traffic sign recognition using MSER and random forests. In: European Signal Processing Conference, pp. 1935–1939 (2012)
16. Greenhalgh, J., Mirmehdi, M.: Recognizing text-based traffic signs. *IEEE Trans. Intell. Transp. Syst.* **16**, 1–10 (2014)
17. Bosch, A., Zisserman, A., Munoz, X.: Representing shape with a spatial pyramid kernel. In: CIVR, pp. 401–408. ACM Press, New York (2007)
18. Mikolajczyk, K., Tuytelaars, T., Schmid, C., Zisserman, A., Matas, J., Schaffalitzky, F., Kadir, T., Gool, L.V.: A comparison of affine region detectors. *Int. J. Comput. Vis.* **65**, 43–72 (2005)