Machine Vision and Applications
DOI 10.1007/s00138-010-0289-5

ORIGINAL PAPER

Automatic real-time road marking recognition using a feature driven approach

Alireza Kheyrollahi · Toby P. Breckon

Received: 1 December 2009 / Revised: 10 May 2010 / Accepted: 20 June 2010

Abstract Automatic road marking recognition is a key problem within the domain of automotive vision that lends support to both autonomous urban driving and augmented driver assistance such as situationally aware navigation systems. Here we propose an approach to this problem based on the extraction of robust road marking features via a novel pipeline of inverse perspective mapping and multi-level binarisation. A trained classifier combined with additional rule-based post-processing then facilitates the real-time delivery of road marking information as required. The approach is shown to operate successfully over a range of lighting, weather and road surface conditions.

Keywords Computer vision · Mobile robotics · Road marking recognition · Vanishing point detection · Intelligent vehicles

1 Introduction

Autonomous driving and road intelligence have been the focus of attention for many computer vision researchers over the last 20 years [1]. Although significant achievement has been made in developing a vehicle that can perform some form of autonomous guided driving, progress has been slow because of the problems of speed, safety and the real-time complexity of the on-road situation.

A human driver gathers constant and numerous visual information from the road and the surroundings. Our brain is quite efficient in analysing this information and responding quickly by an appropriate course of action. For a computer vision system to be able to display a similar ability, it must encompass various detection abilities, each of which has been subject of significant research activity [2–4].

Whilst work on lane detection and tracking is significant [5,22], the literature on road marking recognition is limited with no reported work for real-time on-road text recognition. While road marking (including arrows) and text recognition is a relatively simple task for human drivers, its automatic detection would be very useful—and perhaps essential in some cases—for an autonomous vehicle or as an aid to driver situational awareness in an increasingly complex road environment.

Here we propose a multi-step processing pipeline for robust recognition of road markings and text. First image frames from an on-board camera are captured and pre-processed to remove the perspective effect via an inverse perspective mapping (IPM) driven by automatic vanishing point (VP) detection. After removing the effects of perspective a multi-level thresholding approach is applied to extract bright on-road objects that contrast against the road surface. These objects are then simplified to a contour representation that is passed to an artificial neural network (ANN) classifier for recognition. The results of this per-symbol (i.e. glyph level) classification are post-processed for either driver display or potential use by an autonomous driving decision engine. This approach is shown to operate in real-time under a variety of driving, lighting and road conditions.

2 Previous work

Prior work in this area is limited and we briefly review the main seminal works in this area [6,7,29,30].

A. Kheyrollahi · T. P. Breckon (✉)
Applied Mathematics and Computing Group,
School of Engineering, Cranfield University, Bedfordshire, UK
e-mail: toby.breckon@cranfield.ac.uk

A. Kheyrollahi
e-mail: a.kheyrollahi@cranfield.ac.uk

Published online: 13 August 2010

Charbonnier et al. [6] report a marking recognition process, which relies on finding stretches of lines using a horizontal scan line and then using Radon transform to find the most probable two lines which make up the start and the end of a rectilinear marking. Recognition of an arrow is based on comparing projection of the left and right of the identified rectilinear marking. In this work no perspective correction is done prior to recognition process and resulting performance is not real-time.

Rebut et al. [7] describe an extensive method recognising four classes of arrows and linear markings. They initially use Hough transform for linear marking detection and an arrow pointer template to find the location of the arrow symbols. When marking objects are located on the road surface a Fourier descriptor is then used to extract key features from which a k-Nearest Neighbour (k-NN) classifier is used for final recognition. Training is achieved using a database of sample arrow marking images with further samples created by adding noise to the limited initial data set. A Fourier feature descriptor of degree 34 resulted in an overall global detection error of 6% but a significant false alarm rate of 30%. Again real-time performance was not achieved as processing was carried out off-line in a post-analysis application setting.

More recent work on this topic [29] follows a similar shape-based methodology to that proposed here but is limited to on-road arrow recognition and uses a limited feature set that drives a classifier poorly suited to the complex alpha-numeric character sequences, under degraded quality conditions, considered here. In other recent work [30] an Eigenspace recognition approach is proposed but is reliant on good automated road glyph extraction and sample alignment (as per seminal Eigenspace recognition approaches [31]). Unlike the methodology proposed here to deal with such variation under in-situ on-vehicle operation under varying environmental conditions, [30] does not address either of these issues. Detection rates (for a small set of isolated on-road glyphs only) are comparable to those achieved here but [29,30] do not consider complex sequence recognition in the presence of glyph extraction and road position related noise.

By contrast our method uses a range of features that include invariant spatial moments, histogram projections and normalised angular measurements that are input to a trained neural network for real-time symbol recognition. This approach offers a significantly lower false alarm rate, within the bounds of real-time performance over a much larger set of symbol classes (6 on-road arrow types and 17 alpha-numeric characters) than prior work in the field [6,7,29,30]. In addition it facilitates the recognition of complex multi-glyph sequences under both varying road, lighting and marking quality conditions.

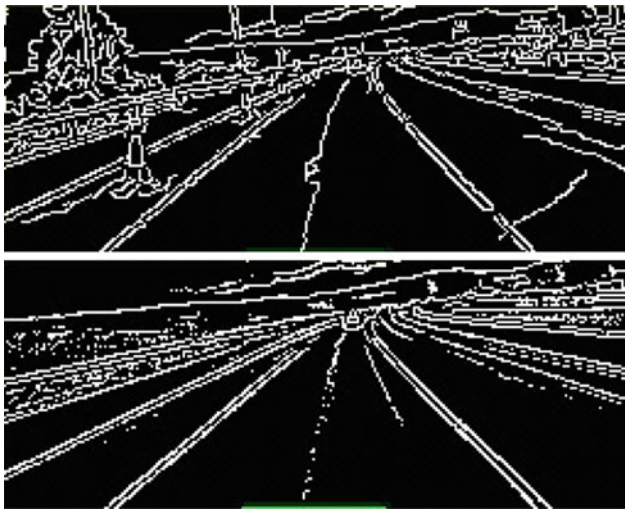

Fig. 1 Temporal filtering of Canny edge detector output. *Upper* standard Canny edge detection output. *Lower* temporal filtering result of Canny edge image sequence

3 Perspective image correction

As a pre-processing stage to our feature extraction approaches we first perform perspective correction on the images obtained from the vehicle mounted, forward facing camera (e.g. Fig. 2). This is performed via a one-time calibration process of vanishing point (VP) detection and subsequent inverse perspective mapping (IPM).

3.1 Vanishing point detection

A vanishing point (VP) is a point in perspective images to which parallel lines converge. Conventional 2D images are essentially a transformation of the 3D world onto 2D plane (image plane). Following a classical pinhole camera model [23] parallel lines (e.g. road edges) within the 3D scene appear to meet at a point within the 2D image-dependent camera angle and lens characteristics [8,23]. An example is shown in the road edges shown in Fig. 1 (upper).

In general, images that illustrate such a perspective (i.e. perspective images) can have up to three such vanishing points located on the image boundary, outside the boundary (external) or at infinity (i.e. far distance within the image, denoted as the infinite vanishing point—e.g. Fig. 2 left). The vanishing point closest to the centre of the image, the dominant vanishing point, is commonly used for perspective correction in road scene images via camera calibration. The first stage in this process is the detection of the VPs within the image.

Classical VP detection is based on mapping edge lines detected within the image onto a unit Gaussian sphere as first described by Barnard [8]. Each line creates a circum-circle on the sphere with the maximal accumulated

Fig. 2 IPM Transform applied to example road image

intersecting region of these circum-circles defining the vanishing point locations. Further developed by various authors [9–11] and the capability for boundary, external and infinite VP detection makes this a popular approach.

However, recent studies show that such Gaussian sphere techniques, although simplifying the unbounded problem to a bounded search space, can produce spurious and false results especially in the presence of noise or texture [12,13]. An alternative, less prevalent approach is the use of a polar space accumulator as originally described by Nakatani [14]. As each point can be represented in polar space by a sinusoid, this improvement uses error minimising of the sinusoids to find the convergence [15]. Clustering of lines in Hough space has also been proposed as an alternative method whereby regular Hough line detection is then followed by clustering line segments to candidate vanishing points [16].

In this work we have used a variant on this latter approach [11,12] which also uses the classical Hough transform *as in the initial stages of VP detection*. The output of Canny edge detection [32] performed on a down-scaled (320 × 240) and Gaussian smoothed version of the image is fed into a temporal filter defined as follows:

$$T_t = \sum_{i=t-n+1}^{t} f_i \tag{1}$$

From Eq. 1, this temporal filter T_t at time t operates over a number of images i in a given calibration sequence. f_i is the processed image frame at time i and n is the number of cumulative frames used to generate an accumulated edge output, T_t. This output, T_t, is then normalised before further processing for Hough-based line detection. As shown in Fig. 1 this temporal filtering will attenuate edge fluctuations that are associated with noise (trees, shadows, etc., shown in Fig. 1 upper) in any given frame and will enhance edges that are constant over n frames such as the road markings and boundaries we desire for VP detection (Fig. 1 lower). A short sequence of n frames, readily obtainable at 25 Hz from a modern video source, over a short distance of roadway gives a suitably stable scene for such a multi-frame temporal approach to be applicable.

This output of T_t is then used to find linear edge features, for VP detection, using a classic Hough transform method [11] within each frame t based on the previous n frames. The maximally detected l lines are extracted from each frame based on their Hough space accumulator value after the exclusion of lines falling within orientation threshold l_t of the vertical or horizontal [11,12].

From this set of l lines (here using $l = 60$), we then find the intersection points of all possible line pairings. These points are then clustered based upon a k-NN clustering approach in 2D image space (here using $k = 3$ for maximal presence of 3 VP in image). Each resulting cluster is then given a suitability score as follows:

$$\Psi(U) = \sum_{i=1}^{n} (|x_c - x_i| + |y_c - y_i|) \tag{2}$$

The score for a given cluster U is calculated as the sum of Manhattan distances of all intersection points, (x_i, y_i), within the cluster from the vanishing point of the previous frame (x_c, y_c),calculated using the same process with $T_t$for frame $t - 1$ (Eq. 2). The Manhattan distance was found to empirically offer more stable results at reduced computational cost than the standard Euclidean approach. Where no previous VP is available, an arbitrary point (such as (0, 0)) is used. A simple averaging on the points from the winning cluster of this scoring method is used to determine the final candidate vanishing point for the frame. This resulting VP is then further averaged with VP of previous frame $t - 1$.

This overall VP detection process converges to the correct vanishing point in approximately 100 frames (i.e. 4 s video @ 25 fps) and acts as a one-time computationally expensive calibration process for a given vehicle camera installation. The detected VP is then used to drive the inverse perspective mapping (IPM) of the on-vehicle camera image.

3.2 Inverse perspective mapping

As previously mentioned, the perspective effect within the 2D images introduces artefacts which could interfere with successful feature extraction and recognition. This issue is particular prevalent in the example of ground plane objects that appear at a ∼45–90° angle to the image plane of the camera. This is illustrated in Fig. 2 (left) with regard to the speed limit symbol (40) on the roadway in front of the vehicle camera.

This effective of perspective can be overcome, although not entirely, by applying an inverse perspective transform using a technique known as inverse perspective mapping (IPM) [24]. The application of IPM requires six parameters [5]: (1) focal length of camera α, (2) height of camera from ground plane h, (3) distance of the camera from the middle of the road d, (4) vertical position of the camera along the road axis l, (5) pitch angle θ, (6) yaw angle γ.

Whilst the first four parameters are obtainable empirically from the vehicle camera installation, the final yaw and pitch angle parameters are retrieved via the vanishing point (x_{vp}, y_{vp}) identified in the prior detection exercise via Eq. 3 [5].

$$\theta = \arctan\left(1 - \left(\frac{2y_{vp}}{N}\right)\right)$$
$$\gamma = \arctan\left(1 - \left(\frac{2x_{vp}}{N}\right)\right) \quad (3)$$

The IPM transform [24] then maps point (u, v) in the vehicle camera image (dimension $M \times N$) to point (x, y) onto the road plane (Eq. 4). This mapping to the road plane of the image pixels (flattened as z is always zero, Eq. 4) can then be extracted as an image of the ground plane with perspective distortion effects removed [24].

The required mapping (Eq. 4) although computationally expensive only requires calculation once for a given set of calibrated vanishing points. It can then be stored for use, as a real-time mapping, for all subsequent image frames from the on-vehicle camera.

$$k_m = \frac{2\alpha}{M-1} \quad k_n = \frac{2\alpha}{N-1}$$
$$x\,(u, v) = \frac{\mu - (\gamma - \alpha)}{k_n} \quad (4)$$
$$y\,(u, v) = \frac{\frac{a\tan(h\sin(\mu))}{x-d} - (\theta - \alpha)}{k_m}$$

An example of this inverse mapping transform is shown in Fig. 2 where we see an image frame from an on-vehicle camera (Fig. 2, left) transformed to an inverse perspective mapping image of the roadway ground plane (Fig. 2, right) based on the detection of vanishing points as outlined previously. As is apparent in Fig. 2, the on-road marking in the transformed image have had the effects of perspective apparent in the original partially removed. This mapped image is significantly more viable as an input for constructing a robust method of road-marking extraction.

4 Road-marking extraction

Road-marking extraction involves the binarisation (i.e. threshold based extraction) of the road scene resulting from the application of the IPM transform to facilitate road surface glyph isolation using a contour-based approach.

Achieving robust thresholding in the presence of extreme light and shadow variations is a classical challenge in image processing that has plagued earlier work [5,6]. Numerous noise sources (shadows, sun/headlight/streetlight reflection, road surface debris and decay) interfere with the process. Broggi [5] proposed an image enhancement method using custom localised adaptive thresholding which produced successful results albeit with a significant, non-real-time computational cost.

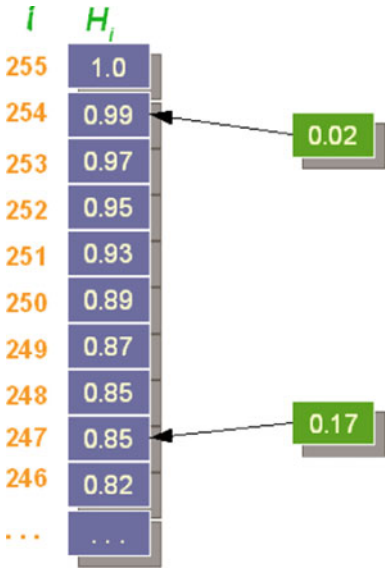

Fig. 3 Choosing four thresholds with $p = 0.02$ and $q = 0.17$ and a 256 bin cumulative histogram

In this work we propose a related adaptive global thresholding approach driven from global histogram information on a per image basis within any given road image sequence.

4.1 Adaptive image thresholding

In general an N-value adaptive global threshold approach is employed to create N-separate binary images for subsequent shape isolation. The normalised cumulative histogram [17] of the resulting IPM transformed image (in grey scale) is used to establish these thresholds. An upper and lower border, p and q, for the range of interest in this histogram are established as percentile offsets of the normalised cumulative histogram maximum value (1.0). This range is within the histogram is then equally sub-divided into $N-1$ sub-ranges via the creation of N thresholds. Here, for road scenes, we use $N = 4$ and create four thresholds from three sub-ranges. For example if $p = 0.02$ and $q = 0.17$ (2nd and 17th percentile) then we then choose cumulative threshold value $k = \{0.02, 0.07, 0.12, 0.17\}$ (for $N = 4$) and then find the corresponding image pixel value thresholds as the lowest index (i) cumulative histogram (H_i) bin with a value greater than or equal to $1.0 - k$.

As illustrated in Fig. 3 for $p = 0.02$ and $q = 0.17$ (2nd and 17th percentile) the corresponding upper and lower thresholds fall at 254 and 247 with the two intermediate thresholds (7th and 12th percentile) falling at the equidistance index positions of 252 and 250. Overall, this algorithmic approach isolates N-boundaries based on the cumulative

Fig. 4 Adaptive thresholding under extreme lighting variations

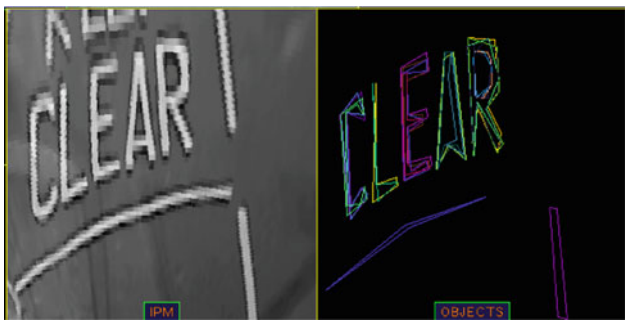

Fig. 5 Examples of shape isolation in the post-adaptive threshold images

distribution of the pixel values within the IPM transformed image from which N binary images, corresponding to differing shape isolation characteristics, can thus be extracted.

A remaining problem is that the distribution of the IPM image will vary substantially depending on the presence/absence and size of any road markings in the image frame. Using fixed values for p and q thus leads to spurious false-positive glyph detection due to poor threshold selection under certain conditions. This is dealt with by reference to the overall mean intensity of the grey scale image frame, $avg(Image)$, and the scaling of p and q on a per frame basis as $i' = C(i/avg(Image))$ where constant C is set empirically to $C = 128$ for 8-bit grey scale images and $i = \{p, q\}$.

The most challenging glyph extraction conditions are generally found in bright sunlight conditions. In Fig. 4 (right) we see an example of thresholding using the proposed approach under such conditions. Of the four binary images produced (Fig. 4, left) we see the arrow glyph easily becomes disconnected in all but one (Fig. 4, top leftmost). This use of a multi-level adaptive threshold approach facilitates robust connected glyph extraction even in the presence of extreme lighting variations and noise (e.g. shadows of Fig. 4, right).

Overall the approach performs well as a robust, real-time methodology for glyph extraction from the road surface that operates successfully both under daylight and night driving conditions over a wide range of illumination conditions.

4.2 Shape isolation

From these binary images, a set of connected image contours are extracted using a backtracking approach [17] prior to simplification into a closed polygon shape representation using the Douglas–Peuker derivative of [18]. This is performed on over all four versions of the road surface IPM transformed image that result from earlier multi-level adaptive thresholding.

Figure 5 shows some examples of the closed contour configurations extracted from these images for differing types of road-marking glyph. On the left (in Fig. 5), we see the IPM input image whilst on the right we show the simplified polygon shapes extracted from the four levels of binary thresholding applied to the IPM input. Notably the complexity of extracted contours does vary significantly.

4.3 Shape post-processing

In order to simplify the later recognition task, and also as an initial method of false-positive glyph filtering we perform two additional stages of shape post-processing: *complexity rejection* and *orientation normalisation*.

Complexity rejection considers the complexity of the resulting simplified polygon representation extracted from the image contours [18] with a view to excluding overly simple or complex shape contours from further processing. At present this is performed using explicit minimal and maximal bounds on the number of segments each polygon contains. Currently, polygons with less than three segments or more than 35 segments are excluded (by empirical selection). Segments below this complexity are commonly found to be the rectilinear lane marks on the road surface (e.g. Fig. 2, left/right) after the contour smoothing applied by [18]. Those above this complexity are almost invariably blocks of foreground segmentation noise originating from the binarisation process (e.g. Fig. 4, top left) rather than genuine road-marking glyphs.

In addition we also perform *orientation normalisation* of the extracted shape segments in order to compensate for both distortion noise in the IPM transform (Sect. 3, e.g. Fig. 2) and any misalignment of the vehicle camera to the roadway (e.g. due to vehicle turning/over-taking/lane transition or camber in the road itself).

This is performed by identification of the object axis of the closed contour itself via reference to the central moments of shape, μ_{ij} [19]. As shown in Eq. 5 axis angle Θ can be derived as the arctangent of the first and second order central moments.

$$\Theta = \frac{1}{2} \arctan\left(\frac{2\mu'_{11}}{\mu'_{20} - \mu'_{02}}\right) \tag{5}$$

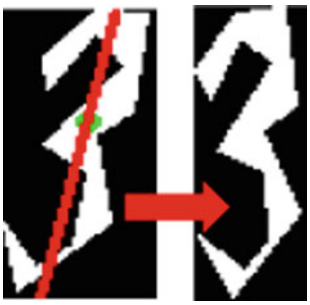

Fig. 6 Orientation normalisation

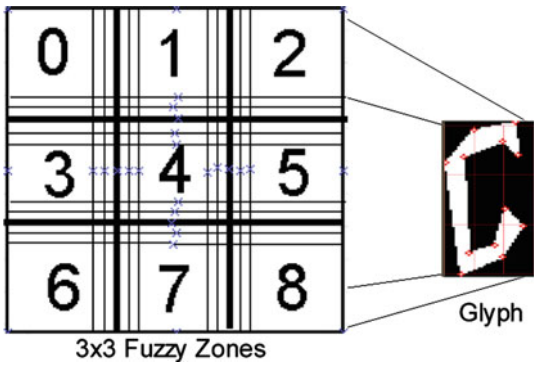

Fig. 7 Fuzzy angle zones mapped onto an example extracted glyph

As shown in Fig. 6, the rotation of the extracted shape can thus be normalised to a standard position.

5 Road marking recognition

The final stage of our pipeline is road marking recognition based on feature extraction, neural network classification and final post-processing. This is performed on the remaining extracted road marking objects with overly complex shapes rejected and the remaining shapes normalised to a common axis orientation as previously defined.

5.1 Feature extraction

Feature extraction extracts a feature vector representation for each potential glyph extracted and retained in earlier shape isolation and post-processing. In feature extraction for character recognition tasks such as this, it is generally recommended to use a ratio of 5–10 times the number of features per class as to the number of classes in the recognition problem [20]. Here with defined 23 classes (one per glyph) we employ a feature vector of 118 elements (ratio ≈ 9.5 features to classes).

This feature vector is primarily constructed from the aspect ratio of the glyph (i.e. height divided by width), normalised central moment measurements of shape [20], Hu spatial moments [25] and horizontal and vertical projection of the glyph. In addition a novel set of glyph features is also introduced based on the fuzzy zoning of angles extracted from the simplified polygon representation extracted from the glyph contour.

For fuzzy zoning of the angles, we calculate the angle of the contour at each edge connection on the contour outline, normalise them (range 0°–179°) and thus create two sets of 9-element features (one each for acute and obtuse angles) via fuzzy set grouping based on spatial position (fuzzy zoning). These zones are defined as a 9-way (3 × 3) grid division mapped onto the extracted area of the glyph (as shown in Fig. 7). The exact area (in pixels) of each zone within the grid is adapted to be an equal division of the overall glyph area. Membership of a given pixel to a given zone is not exclusively disjunctive with boundary pixels having membership of multiple zones via the definition of overlapping fuzzy zone boundaries (Fig. 7).

Accumulations are made on a per zone basis for angular features within the glyph contour. Within the fuzzy border regions such features only partially contribute to the overall zone accumulation relative to its position in the border region. Features within a zone contribute 1.0, those outside 0 and a range 0–1.0 for fuzzy zone contributions. Over all zones the sum of contributions made by a given feature to all accumulators will always be 1.0. This is illustrated in Fig. 7 where we see an example glyph mapped to a set of nine (3×3) fuzzy zones with overlapping boundaries that correspond to these varying feature contributions over multiple zones.

One element in the feature vector will thus represent the acute and obtuse accumulator for each of the nine angle zones covering the spatial layout of the glyph. Formally, we can define each such accumulator follows *for a glyph contour* with N angular features:

$$\Phi_i = \sum_{j=0}^{N-1} \psi\,(i,\, j) \tag{6}$$

In Eq. 6 Φ_i is the resulting accumulation for zone i, and $\psi(i, j)$ is fuzzy membership relating to the position of angular feature j to zone i.

Finally, in addition to this concept of fuzzy spatial angular accumulator measures we also employ a secondary linear angular profile measure over the length of the external boundary contour of the glyph (orientated clockwise from the topmost point). This essentially constructs a fuzzy contour orientation profile for the simplified object. Given the near scale-constancy of road-markings, we consider the external boundary contour as N equilength segments over its length (here we empirically use $N = 25$ for UK road-marking scale). Each segment is represented by the last (polygon joint) angle along its length from the common contour origin

(topmost contour point) for each glyph shape. Where a given segment has no angle over its length it is represented by the angle from the previous segment (initialised from 0 for first segment).

This angular profile along the length of the exterior glyph contour is then encoded as a normalised four-element vector ω_n. This proportionally assigns weights to multiple vector elements to create a novel multi-variate representation for segment angle α (degrees) and vector elements ω_i for $i = \{0 \ldots 3\}$ (see Eq. 7).

$$\omega_i = \frac{90 - \min\left(|\alpha - 90i|, 90\right)}{90} \tag{7}$$

Overall feature extraction results in an 118-element feature vector descriptor of each glyph comprising of aspect ratio (one element), normalised central moments (seven elements up to order 3), Hu spatial moments (seven elements up to spatial order 3), horizontal/vertical histogram projections (50 vertical and 35 horizontal) and fuzzy zoning of angles (nine for acute and nine for obtuse). This feature vector description forms the input representation to a trained neural network classifier used for recognition.

Overall these features are selected to give an overall descriptor that combines established global measures of glyph shape (aspect ration, moments, histogram projections) and relative geometric feature placement (fuzzy zoning of angles). Here multiple spatial/geometric measures are used in a "rich" descriptor to facilitate (a) differentiation over a larger glyph alphabet and (b) increased robustness to noise over a large test evaluation set when compared to earlier (feature-sparse or reduced-feature) works [6,7,29,30]. Empirical results show a further increase in feature vector complexity reduces overall performance of the approach due to the prevalence of noise (e.g. increased granularity of fuzzy angle zones) whilst earlier work illustrates lesser results over a smaller feature set [6,7,29,30]. The use of a larger multi-faceted feature set in this work mitigates the overall effect of isolated feature stability under varying road conditions (Sects. 4/6) when combined with a noise-tolerant classification approach (Sect. 5.2). Future work will investigate feature stability under such conditions for the derivation of an optimal subset where applicable.

5.2 Neural network classifier

Here we use a single hidden layer artificial neural network classifier [26] with the sigmoid activation based on training with resilient back propagation (RPROP) [27]. This operates on a 118-node input layer (same as feature vector), 69-node hidden layer and 23-node output layer (size of glyph alphabet). The number of hidden nodes was set empirically based on minimising the generalisation error over unseen test data sample in order to avoid overfitting [21].

Classifier training/testing was performed based on 1022 sample glyphs manually extracted from real road footage sequences with randomly selected 80–20% ratio subsets used for training and testing respectfully [26]. Training takes order of $20s$. for the current glyph alphabet in use. Parameter variations are considered and presented in terms of those selected for the final ANN configuration. Other parameter choices made are selected empirically and shown to be robust over varying marking quality, weather conditions, scene cluttered and illumination conditions.

The current alphabet includes 6 arrow classes (straight, toLeft, toRight, doubleToLeft, doubleToRight and bifurcation) and 17 characters. Individual glyphs were only included on the basis of suitable available training data and the current set could be expanded given sufficient training examples for further glyph types. Due to the similarity of three pairs of glyphs (S and 5, O and 0, A and 4) these were defined as three single glyphs (S/5, O/0, A/4) with further differentiation performed in post-processing with relation to word context.

The neural network implementation in use [26] was trained on the basis of an output vector of floating point values (range -1 to $+1$) where the maximal vector entry indicates the classification of the glyph. This notion of output classification requires two additional thresholds: (a) *a maximal score* above which the output vector entry must be to be considered a valid classification and (b) *a maximal separation distance* between the maximal output vector entry and the second maxima in the output. Imperially a *maximal score threshold* of 0.7 and a *maximal separation distance* threshold of 0.2 were selected to with output results not satisfying these criteria resulting in a null classification for the given glyph character.

5.3 Post-processing

After individual glyph classification via neural network a secondary stage of post-processing is employed to identify coherent and consistent glyph patterns (words and symbol structures) on the roadway. In terms of the overall system-level results (Sect. 6) this stage is as important as its predecessor in false positive elimination. The post-processing strategy has been implemented separately for the arrow and character subsets of the overall glyph alphabet.

Arrows Here we have used a simple post-processing method based on a temporal accumulator value over multiple image frames in the road sequence. This assists in the elimination of "single-frame" false-positive glyph classifications by providing temporal consistency filtering. For every arrow shape recognised in a frame t, via the previous neural network classifier, value s is added to the accumulator of that class, $Acc_t(C_i)$, as depicted in Eq. 8 where $X_i$is a glyph instance classified as class C_i.

$$Acc_t\left(C_i\right) = \min\left(\tau, Acc_{t-1}\left(C_i\right) + \sum_{j=0}^{n-1}\left(s, X_j \rightarrow C_i\right)\right) \tag{8}$$

The use of the min() function in Eq. 8 ensures the accumulator for a given class $Acc_t(C_i)$, will not exceed a threshold τ. Every class, C_i,for which a glyph instance is not found (i.e. classified) in frame t has its accumulator, $Acc_t(C_i)$ decremented by decay factor s' to a minimal value bound of 0. Using such a decay factor ensures that after a few frames, without a classification for class C_i the associated accumulator, $Acc_t(C_i)$, will suitably decay below a defined display threshold.

Text patterns The post-processing of text patterns to provide a simplified form of *"word recognition"* is achieved in two stages: word coherence and matching. The first stage, word coherence, attempts to find lines of characters that construct individual words whilst the second stage of matching applies a fuzzy match against a set of pre-defined dictionary words/patterns.

Given the context of on-road marking recognition and the associated limited vocabulary this is deemed more appropriate than a full classical HMM-based text pattern recognition approach [28]. An empirical study of the vocabulary of words allowed on UK roads shows at most a set of 10–15 regular, non-geographic "words" that are encountered within this context.

Our dictionary defines 19 exemplar words (road numbers, speeds, directions, labels): {A5, 20, 30, 40, CAR, WASH, SOUTH, LANE, SLOW, NO, ENTER, BUS, ONLY, KEEP, CLEAR, M1, A421, HOTEL, A509}. The potential performance degradation of expanding this set to sub-100 such words is considered to be minimal.

Initial word search spatial orders the glyph characters detected by earlier processing (left to right), identifies spatially distinct lines of text on the roadway scene and groups those lines into words based on relative spatial location of the glyphs. In the secondary matching step we make a fuzzy match against the dictionary list as follows.

For defined dictionary word (glyph sequence) C_p, we match the detected word (glyph sequence) W_q from the roadway by maintaining an accumulator value, C_{jp}, for each character j of $C_{p,}$.In doing so, we assume that C_p has m characters and W_q has n characters and $n \neq m$ is possible. The value of this accumulator, $f()$, is calculated as shown in Eq. 9 for each character, C_{jp}, which matches to character i of W_q, denoted W_{iq}.

$$f\left(C_p^j, W_q\right) = \sum_{i-0}^{n-1} s\left(1 - \left|\varphi\left(j, m\right) - \varphi\left(i, n\right)\right|\right) \tag{9}$$

In Eq. 9, s is an accumulation constant and and $\phi()$ is defined as in Eq. 10 for a given i and n (as exemplars).

Fig. 8 Successful recognition of the on-road word "SLOW" in night video footage

$$\varphi\left(i, n\right) = \frac{i}{n-1} \tag{10}$$

This function is designed to penalise the score of the detected word if position of each detected character is not close to the position of a similar character in the dictionary word that is being matched.

An example of successful recognition of the glyph sequence corresponding to the word "SLOW" is shown in Fig. 8. Here we see the original roadway camera view (top left view), the IPM transformed images and thresholded glyph text (bottom left view) and the positional overlay of the detected and recognized glyph pattern (e.g. "SLOW") overlain onto the IPM transform of the roadway scene (bottom right view). The top right view (Fig. 8) shows the current edge pattern of the roadway for reference. A label of the recognised text is also shown in the main top left view (Fig. 8). Notably this result has been obtained at night based on vehicle headlight illumination of the road markings.

6 Results

We present a series of the results of the proposed approach operating on UK rural and urban roads under varying weather and lighting conditions. All of the video sequences used were captured at 15 fps using a 640 × 480 resolution digital camera, mounted behind the windscreen, with the speed of the vehicle varying between 30–70 mph (as per UK traffic regulations). An example of the camera to road perspective is shown in Fig. 9 (top left).

Image frames retrieved from the capture device are initially pre-processed via the VP detection module (Sect. 3) for one-time initiation of the IPM transform parameters. These are down-sampled to a 320 × 240 resolution for processing

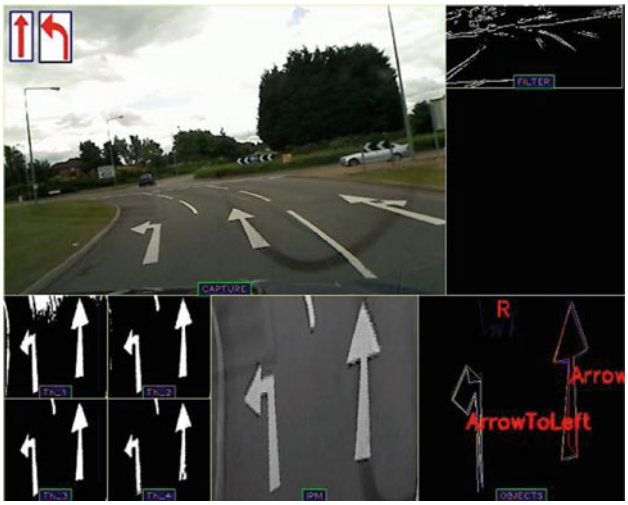

Fig. 9 Recognition of two arrows in the same frame

Fig. 10 Recognition of two words "CAR" and "WASH"

Fig. 11 Erroneous recognition of "40" speed limit in place of "KEEP CLEAR"

from which a road-way sub-region of interest defined in initial calibration (e.g. Fig. 9, bottom left). Post-initialisation all video frames were processed in real-time at approximately 11–15 fps (1.5 GHz CPU, single core in use) with each video frame taking approximately 60–90 ms to process. As expected processing time varies slightly with road marking complexity due to the number of glyphs to be processed in a given frame.

The approach generally performed very well under varying lighting and weather conditions. Figures 9 and 10 show the successful recognition of multiple arrow and multiple line word markings under daylight conditions. Figure 8 shows a similar successful recognition under night conditions. In all of the examples (Figs. 8, 9, 10, 11) we see the original roadway camera view (top left view), the IPM transformed images and thresholded glyph text (bottom left view) and the positional overlay of the detected and recognized glyph pattern overlain onto the IPM transform of the roadway scene (bottom right view).

By contrast Fig. 11 shows the erroneous detection of the "A40" road marking due to multiple failures in the detection of the "KEEP CLEAR" marking pattern. This is largely attributable to broken glyph characters caused by road noise (poor/worn markings), light reflection via the windscreen in front of the camera and the elongated nature of this particular marking with regard to the camera field of view (FoV). Improved camera placement and FoV engineering could look to overcome this issue in place of adjustments to the processing methodology itself.

Direct comparison with prior work in this area [6,7,29,30] is practically difficult and thus not presented here. Empirical studies on simple non-feature based approaches (e.g. template matching [23], also [6,7]) yield poor performance due to variation in the geometric skew of the glyphs and deal with a significantly lesser glyph alphabet that considered here. Future work could consider further comparative evaluation of [6,7,29,30] over the glyph alphabet used here.

In order to quantify the performance of the system we present isolated glyph recognition rates for the neural network classifier over a set of extracted and pre-processed (as per Sect. 4) glyph examples in Table 1. In addition we present overall system performance over a set of test sequences in varying conditions in Table 2. As we see from Table 1, isolated glyph recognition over the test set generally performs very well based on the current feature selection used for classification (~75–100%). Sub-optimal recognition on the bifurcation arrow is encountered in some cases due to its similarity (at a given scale) to the glyph character "U". Alphanumeric character recognition was in general very good but suffered in some instances from limited variation in

Table 1 ANN recognition result on test samples (overall = 92%)

1	100%
2	100%
3	80%
A4	86.9%
Arrow	96%
ArrrowToLeft	100%
ArrrowToRight	85.7%
Bifurcation	75.5%
C	100%
DoubleArrowLeft	100%
DoubleArrowRight	100%
E	100%
H	83.3%
K	77.8%
L	100%
M	75%
N	88.8%
O	89.5%
R	75.9%
S5	100%
T	80%
U	83.7%
W	100%

training examples. Improved training, over a larger set of training examples would potentially address both of these issues.

From Table 2, we can see the overall combined recognition result for the initial glyph classification and the post-processing arrow/glyph sequence consistency checks. These results are shown for a variety of weather, illumination and environmental conditions. Overall we see 85% successful recognition for arrows and 81% recognition for the 19 dictionary text patterns/words. False detection are minimal and this is largely aided by the post-processing employed after initial glyph classification. For reference Table 2 refers to the UK location of the sequence and the environment/weather/illumination conditions under which the testing was performed.

7 Summary

A methodology for the automatic recognition of on road markings is presented that operates, under varying test conditions, with approximately 81–85% success based on the data sets available for classifier training. This approach has been shown to work within a viable real-time constraint (11–15 fps) both in daylight and at night. Prior work in this area is very limited [6,7,29,30] and no prior work on the recognition of on-road textual words has been carried out. Extensive results are shown under varying conditions, vehicle speeds and environmental conditions.

The approach is based on the construction of an IPM transformed image of the roadway derived from an initial calibration of automatic vanishing point detection. This top down roadway view forms the input to robust feature extraction supported by a unique illumination resistant multi-threshold. A trained single layer neural network classifier is then used for individual glyph classification prior to glyph word/pattern construction via a simple post-processing matching methodology.

Future work will investigate the use of alternative shape features for glyph description, the extension of system glyph

Table 2 Arrow and text recognition results over six different video sequences

Sequence (# frames)	Arrows (total/recog.)	Arrows (False detect.)	Text (total/recog.)	Text (false detect.)	Conditions
Night (8370)	15/12	2	4/2	0	Night, Poor capture: 30–60 mph
London City (13094)	23/21	2	22/18	5	Cloudy, sunny, rainy: 30–60 mph
Leighton Buzzard Town/Rural 1 (9421)	16/14	2	9/7	1	Sunny, extreme shadow condition: 30–60 mph
Leighton Buzzard Town/Rural 2 (8709)	33/29	1	7/5	3	Rainy, wet: 30–60 mph
Milton Keynes City/Rural (9723)	27/24	0	2/2	1	Clear, cloudy: 30–60 mph
Cranfield Rural (4130)	10/7	1	8/8	0	Cloudy, Poor arrow quality: 20–60 mph
Motorway/Highway (6208)	5/3	0	0	0	Partly cloudy: ≈70 mph
Overall	129/110 (85.2%)	8	52/42 (80.7%)	10	

vocabulary and word/pattern dictionary and the potential use of alternative classifiers for the core glyph classification task.

References

1. Bishop, R.: Intelligent vehicle applications worldwide. Intell. Syst. Appl. IEEE **15**(1), 78–81 (2000)
2. Campbell, N.W., Pout, M.R., Priestly, M.D.J., Dagless, E.L., Thomas, B.T.: Autonomous road vehicle navigation. Eng. Appl. Artif. Intell. **7**(2), 177–190 (1994)
3. Maurer, M.. Dickmanns, E.D.: A system architecture for autonomous visual road vehicle guidance. ITSC 97. IEEE Conference on Intelligent Transportation System, pp. 578–583 (1997)
4. Hoffmann, G.M., Tomlin, C.J., Montemerlo, F., Thrun, S.: Autonomous Automobile Trajectory Tracking for Off-Road Driving: Controller Design, Experimental Validation and Racing. American Control Conference, 2007 ACC'07, pp. 2296–2301 (2007)
5. Broggi, A.: Robust real-time lane and road detection in critical shadow conditions. In: Proceedings of International Symposium on Computer Vision, pp. 353–358 (1995)
6. Charbonnier, P., Diebolt, F., Guillard, Y., Peyret, F.: Road markings recognition using image processing. In: IEEE Conference on Intelligent Transportation System, ITSC 97, pp. 912–917 (1997)
7. Rebut, J., Bensrhair, A., Toulminet, G.: Image segmentation and pattern recognition for road marking analysis. IEEE Int. Symp. Ind. Electron. **1**(4–7), 727–732 (2004)
8. Barnard, S.T.: Interpreting perspective images. Artif. Intell. **21**, 435–462 (1983)
9. Magee, M.J., Aggmal, J.K.: Determining vanishing points from perspective images. Comput. Vis. Graphics Image Processing **26**, 256–267 (1984)
10. Quan, L., Mohr, R.: Determining perspective structures using hierarchical Hough transform. Pattern Recognit. Lett. **9**(44), 279–286 Elsevier (1989)
11. Lutton, E., Maitre, H., Lopez-Krahe, J.: Contribution to the determination of vanishing points using Hough transform. Pattern Anal. Mach. Intell. **16**(4), 430–438 (1994)
12. Cantoni, V., Lombardi, L., Porta, M., Sicard, N.: Vanishing point detection: representation analysis and new approaches. In: Proceedings of 11th International Conference on Image Analysis and Processing Volume, Issue, pp. 90–94 (2001)
13. Almansa, A., Desolneux, A., Vamech, S.: Vanishing point detection without any a priori information. Pattern Anal. Mach. Intell. **25**(4), 502–507 (2003)
14. Nakatani, H., Kimura, S., Saito, O.: Extraction of vanishing point and its application to scene analysis based on image sequence. In: Proceedings of the 5th International Pattern Recognition Conference (1980)
15. Matessi, A., Lombardi, L.: Vanishing point detection in the hough transform space. In: Proceedings of the Fifth International Euro-Par Conference, Tolouse, France, pp. 987–994 (1999)
16. McLean, G.f., Kotturi, D.: Vanishing point detection by line clustering. IEEE Trans. Pattern Anal. Mach. Intell. **17**(11), 1090–1095 (1995)
17. Pratt, W.K.: Digital Image Processing, 3rd edn. John, New York (2001)
18. Wu, S.T., Marquez, M.R.G.: A non-self-intersection Douglas-Peucker algorithm. In: Proc. XVI Brazilian Symposium on Computer Graphics and Image Processing, pp. 60–66 (2003)
19. Sebe, N., Lew, M.S.: Robust Computer Vision: Theory and Applications. Kluwer, Norwell (2003)
20. Trier, Ø.D., Jain, A.K., Taxt, T.: Feature extraction methods for character recognition—a survey. Pattern Recognit. **29**(4), 641–662 (1996)
21. Bishop, C.M.: Neural Networks for Pattern Recognition. Oxford University Press, Oxford (1996)
22. Kastrinaki, V., Zervakis, M., Kalaitzakis, K.: A survey of video processing techniques for traffic applications. Image Vis. Comput. **21**, 359–381 (2003)
23. Forsyth, D., Ponce, J.: Computer Vision—A Modern Approach. Prentice-Hall, New Jersey (2003)
24. Bertozzi, M., Broggi, A., Fascioli, A.: Stereo inverse perspective mapping: theory and applications. Image Vis. Comput. J. **8**(16), 585–590 (1998)
25. Hu, M.-K.: Visual pattern recognition by moment invatiants. IRE. Tran. Inf. Theory **8**, 179–187 (1962)
26. Bishop, C.M.: Neural Networks for Pattern Recognition. Oxford University Press, Oxford (1996)
27. Riedmiller, M., Braun, H.: A direct adaptive method for faster backpropagation learning: the RPROP algorithm. In: Proceedings of the ICNN, San Francisco, pp. 586–591 (1993)
28. Kundu, A., He, Y., Bahl, P.: Recognition of hand-written word: first and second order hidden Markov model based approach. Pattern Recognit. **22**(3), 283–297 (1989)
29. Li, Y., He, K., Jia, P.: Road markers recognition based on shape information. In: Proc. IEEE Int. Symp. on Intelligent Vehicles Symposium, pp. 117–122 (2007)
30. Noda, M., Takahashi, T., Deguchi, D., Ide, I., Murase, H., Kojima, Y., Naito, T.: Recognition of road markings from in-vehicle camera images by a generative learning method. In: Proc. IAPR Conf. on Machine Vision Applications, 15–5 (2009)
31. Turk, M., Pentland, A.: Eigenfaces for recognition. J. Cogn. Neurosci. **3**(1), 71–86 (1991)
32. Canny, J.: A computational approach to edge detection. IEEE Trans. Pattern Anal. Mach. Intell. **8**(6), 679–698 (1986)