

A new approach in road sign recognition based on fast fractal coding

Hossein Pazhoumand-dar · Mahdi Yaghoobi

Received: 1 February 2011 / Accepted: 3 August 2011 / Published online: 28 September 2011
© Springer-Verlag London Limited 2011

Abstract The tasks of traffic signs are notifying drivers about the current state of the road and giving them other important information for navigation. In this paper, a new approach for detection, tracking, and recognition such objects is presented. Road signs are detected using color thresholding, after that candidate blobs that have specific criteria are classified based on their geometrical shape and are tracked through successive frames based on a new similarity measure. Candidate blobs that successfully pass the tracking module are processed for extracting their fractal features, and final recognition is done based on support vector machines with kernel function. Results validate effectiveness of newly employed fractal feature and show high accuracy with a low false hit rate of this method and its robustness to illumination changes and road sign occlusion or scale changes. Also results indicate that compared to the other pictogram feature representation techniques, this approach shows a more proper description of road signs.

Keywords Road sign recognition ·
Computer vision-based driver assistance ·
Fractal feature extraction · Support vector machines ·
Road sign tracking

1 Introduction

The tasks of traffic signs are notifying drivers about the current state of the road and giving them other important information for navigation. The information provided by the road signs (RS) is encoded in their visual traits: shape, color, and pictogram.

Close to two decades, great amount of research in the field of automatic road sign detection and recognition systems (RSR) has been done. Because drivers may not notice the presence of road signs, one of the main objects of such systems is to warn them of presence of traffic signs in different ways. However, it is interesting to pay attention to some of the common problems that are involved in traffic-sign detection. The first problem to be overcome is caused by the variable lighting conditions of a scene in a natural environment. Another problem to surpass is the possible rotation of the signs. Although the perfect position for a road sign is perpendicular to the trajectory of the vehicle, many times, the sign is not positioned that way. Therefore, an automatic system must be able to detect signs in many positions and hence must be invariant to rotation and translation. The next problem is related to traffic-sign size because we find signs of different dimensions, although officially there are only three normal sizes for nonurban environments [1].

The first paper in the field of road sign recognition emerged in Japan in 1984. The goal was to try different methods for the detection of objects in natural scenes. Since that time, many research groups are paying attention to the field and great amount of works have been published. Most efforts have focused on the development of computational algorithms to detect road signs in each independent frame, and many works separate the algorithms into two stages: detection and recognition.

H. Pazhoumand-dar (✉) · M. Yaghoobi
Department of Computer Science, Mashhad Branch,
Islamic Azad University, Mashhad 9187147578, Iran
e-mail: pazhoumand@mshdiau.ac.ir

M. Yaghoobi
e-mail: yaghoobi@mshdiau.ac.ir

RS detection are mainly based on color criteria [1–4] followed by the geometrical edge analysis or shape information. As a result, in such systems, the color space plays an important role. In [3, 4], the popular RGB color space, or variations based on it such as relations between the color coefficients are used. However, the RGB space is not optimized for problems such as RSR, because it is susceptible to lighting changes. Thus, color spaces that are more immune to such changes are preferred: for instance, the hue saturation intensity (HSI) family spaces is used in [2] that applied a threshold over a hue, saturation, value representation of the image to find regions with a high probability of having a traffic sign. A more sophisticated method of sign detection based on genetic algorithms, which also has used HSI, is proposed in [5]. After color segmentation, road signs are detected based on using circular and triangular shapes. In addition to HIS, the YUV color space was used in [6] to detect blue rectangular signs. However, as [7] demonstrated, there is not an algorithm robust enough to all difficulties from outdoor environments.

Once the color segmentation process has been done, some researches have gone into classifying the candidate signs according to their shape. For example in [2], shape discrimination is done by the FFT to deal with change in rotations and scale. In [4], different methods using the extracted corners are applied to the traffic-sign shapes, and in [8], color extraction is complemented with shape features using two NNs. Other algorithms used structural information based on edge detection, rather than color information. For example, a Laplacian filter with previous smoothing was used in [9] for grayscale images. Grayscale images were also used with a Canny edge detector in [10], and a color image gradient was used in [11]. Recently, a two-camera system is used in [12]. In this case, two identical cameras are used to get stereo pairs of images. Using the epipolar geometry and the predefined calibration parameters, the region of a detected object on one source is estimated on its stereo object. With a similarity criterion based on normalized correlation, the position of the target is found. Using this dual redundancy, only the object that is not observable by either of the cameras is lost. In their work, fast radial symmetry was used for the detection of circular signs, and for the case of triangular signs, the method is based on the Hough transform. However, detection of traffic signs in only a single image has three problems: (a) information about positions and sizes of traffic signs, which is useful to reduce the computational time, is not available; (b) it is difficult to detect a traffic sign correctly when a temporary occlusion occurs, and (c) correctness of the detection is hard to verify [13]. To maintain high classification accuracy in real-time implementation, it is advisable to utilize the temporal dimension and track the traffic sign candidates over time.

After a traffic sign has been detected, it should be recognized from a wide set of possible classes using some kind of classification technique. Normally, the pictogram of detected sign must be suitably represented for a specific classification method that is usually done by either extracting some features from the image and using these features as inputs to the classifier or presenting the part of the image itself through the sub-sampled pixel intensities. Examples of methods in the first approach are histograms [14], FFT computed after a complex log-mapping of exterior borders [15], and wavelets [16]. Furthermore, authors in [17] used a method based on a modified AdaBoost algorithm that is called SimBoost.

After the choosing of appropriate features, many classification methods have been proposed. For example, different kinds of neural networks, such as ART1, ART2, SOM, Hopfield and Back-Propagation Neuronal Network (NN) as well as different types of template matching methods, were applied in many RSR researches for pictogram recognition stage [3, 16, 18]. In [19], authors used neural network (NN) and decomposed the multiclass classification problem into a series of one-to-one problems, each assigned to a separate network and a similar problem decomposition but using an error-correcting output code framework was done in [12] and authors proposed a novel binary classifier through an evolutionary version of AdaBoost. A multiclass learning technique based on embedding a forest of optimal tree structures in the error-correcting output code was used for recognition.

Recently, support vector machines (SVMs) have been reported as a good method to sign recognition due to their ability to provide good accuracy as well as being sparse methods. In [2, 4, 20, 21], authors propose a road sign recognition system based on SVMs, and extracted sign candidate was classified by two stages: shape classification and pictogram recognition using Gaussian kernel SVMs. The authors compare different feature representation of candidate sign pictogram and usage of other kernel functions for SVM models. Nevertheless, for complete data sets of traffic signs, the number of operations needed in exiting methods is still large, whereas the accuracy needs to be improved. Since feature representation of pictogram play essential rule in RS recognition accuracy, many other researches try to evaluate robust methods to rotation, occlusion, and scale changes for this purpose.

In this paper, we have proposed a new method for traffic sign tracking to improve precision with a low false negative rate in detection results, and also we have applied a method based on fractal feature extraction for road sign pictogram representation in order to make the traffic sign recognition task less expensive computationally and increase its accuracy.

The next section presents an overview of the proposed system. Sections 3, 4, and 5 discuss the segmentation

procedure, shape classification, and tracking methods, respectively. Section 6 discusses about compute feature vector of inner image based on fractal features, and Sects. 7 and 8 show the representative experimental results and conclusions.

2 System overview

In this paper, we present a new method for detection, tracking, and recognition of traffic signs that has been applied to Iranian traffic signs. An overview of the proposed method is shown in Fig. 1.

This method consists of the following steps: (1) Color segmentation is performed on HSI space to obtain candidate blobs; (2) Features based on the distance from the outer borders of the blob's bounding box along with SVM classifiers are used to classify the candidate blobs into true and false candidates; (3) Ring projection is applied to match blobs across frames for the sake of tracking; (4) Fractal feature extraction is performed on tracked blobs following binarization, and then object classification is conducted with SVM classifiers with kernel function. In the following, each step of the proposed method is presented in detail.

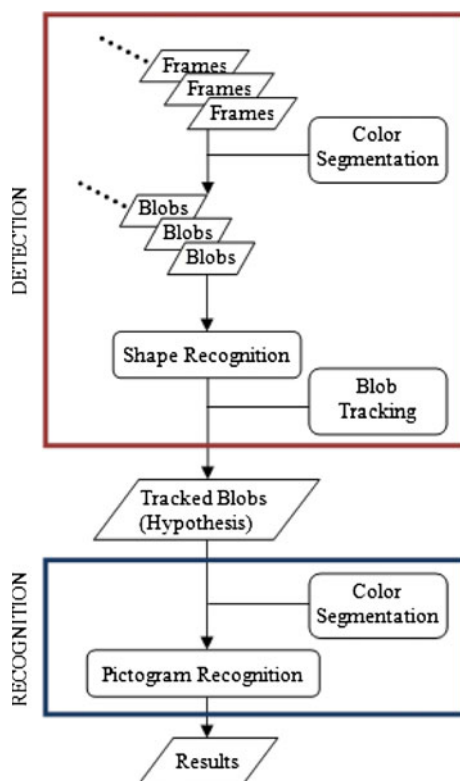


Fig. 1 Illustration of the proposed method

3 Sign detection

Color segmentation is a critical step, since errors at this stage might have fatal consequences for the rest of the system [18]. Color information is used in the detection algorithm and segmentation of the image based on two color used in road signs (red and blue). Thresholding in HSI color space is an efficient way to find the areas of interest, which may contain a road sign because of its similarity to the human perception of colors and besides two components (Hue and Saturation) encode the color information, being strongly robust against lighting condition variations [4].

Thresholds were setup using a set of real traffic scenes with variable illumination conditions. Thus, the segmentation algorithm is pixel-based classification into three classes (red, blue, and noise). Consequently, even road signs with different illumination are correctly processed and the algorithm is fast. However, negative alarm of color classification has severe consequences to the classification result. A set of rules applied to blobs in resulting binary image for each color to obtain Regions of Interests (ROI's). A ROI succeed to next level if it meets the following criteria:

$$1. \quad 0.7 * Width(blob) \leq height(blob) \leq 1.3 * width(blob)$$

OR

$$0.7 * Height(blob) \leq width(blob) \leq 1.3 * height(blob)$$

$$2. \quad Area_{min} \leq area(blob) \leq area_{max}$$

where *Height* and *Width* are the height and width of blob's bonding box. In Fig. 2, some results of detection stage are shown. Figure 2a and d contains the original pictures of road scenes. Figure 2b and e represents results for color segmentation, and Fig. 2c and f shows their corresponding image after removing blobs without criteria mentioned above.

4 Shape classification

Once areas in an image that may contain a traffic sign have been detected; morphological dilation is applied to join part of signs that may have become separated; and then the shape of each ROI is recognized using linear SVMs. Support vector machines are the classifiers that are formulated for a two-class problem and are known to have high generalization. So an N-class problem must be converted into N two-class problems. In the *i*th two-class problem, class *i* is separated from the remaining classes. Namely, in the *i*th problem, the class labels of input vector *x* that belongs to class *i* are set as $y = 1$, and the class labels of remaining data are set as $y = -1$.

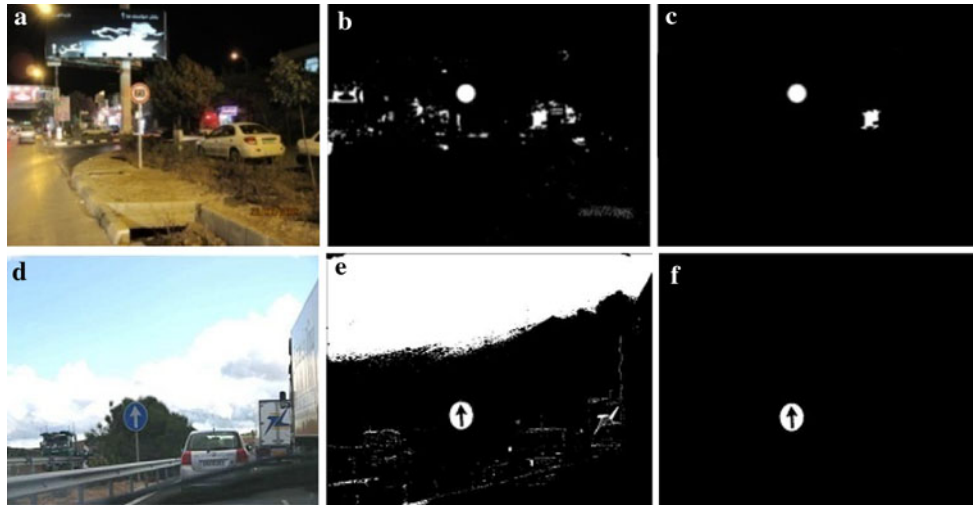


Fig. 2 a, d The original pictures of road scenes. b, e Results for color segmentation. c and f Corresponding images after removing blobs without criteria mentioned in Sect. 3

The basic concept of SVMs is to transform the input vectors to a higher dimensional space Z by a nonlinear transform, and then an optimal hyper-plane that separates the data can be found. This hyperplane should have the best generalization capability. As shown in Fig. 3, the black dots and the white dots are the training dataset that belong to two classes. If a hyper-plane $\{w, b\}$ separates the two classes, the points that lie on it satisfy $x \cdot w^T + b = 0$, where w is normal to the hyperplane, $|b|/\|w\|$ is the perpendicular distance from the hyper-plane to the origin, and $\|w\|$ is the Euclidean norm of w . The optimal plane H is found by maximizing the margin value $2/\|w\|$. Hyperplanes H_1 and H_2 are the planes on the border of each class and also parallel to the optimal hyper-plane H . The data located on H_1 and H_2 are called support vectors. The function of hyper-plane is as follow:

$$f(x) = \text{sgn}(x \cdot w^T + b) \tag{1}$$

The points for which the equality in (1) holds give us the scale factor for w and b or, equivalently, a constant difference of the unity. These points lay on both hyper-planes H_1 and H_2 .

In this paper, we use distance to border as input feature vector to the linear SVMs, as introduced in [22]. In their

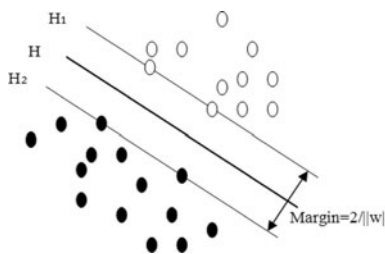


Fig. 3 The SVM binary classification

Fig. 4 DtBs for a triangular shape



work, DtBs are the distances from the external edge of the blob to its bounding box. Figure 4 shows four DtB vectors of 6 components that have been used for a triangular shape based on the previous color detection. DtBs are feed to specific SVMs to diagnose the possible geometric shapes. Since we seek for blue sign and signs with red ribbon, we consider 2 possible shapes and eight SVMs are needed to classify each DtB vector. Therefore, four favorable votes are gathered for each shape. A majority voting method has been used in order to get the classification with a threshold; if the result does not relate to desired shapes, blob is rejected as a noisy shape. Figure 5 shows some results for shape classification of signs with different lighting conditions. As pointed out in [22], due to normalization of ROI scale to specific size, this method is invariant to scale changes. Since four feature vectors are used to characterize blobs, and linear SVMs have high robustness to noisy data, shape classification results are invariant to rotation and occlusion of the sign.

5 Tracking

In suggested system, to have precision with a low false negative rate in detection results, a tracking algorithm is applied to assign blobs that appear in the frame (t) to the



Fig. 5 Some results for shape classification of signs with different lighting conditions

blobs in the previous frames ($t - T$), $T \geq 1$ which are candidate as a traffic sign. Two blob tracking method were merged, which were (a) Multi-Resolution Blob Tracking [23] and (b) Two-Layered Blob Tracking [13]. In [23], tracking is performed by matching blobs between the frames. The sizes of blobs are expected to increase from frame to frame and the positions might change according to the movement of the vehicle. Due to the temporal occlusion and/or missed sign detection, blobs in frame (t) may not have a corresponding blob in some of the previous frames. To overcome this problem, the blob identification needs to be maintained to be able to track the exact blob that may occur in future frames. Once a ROI is resulted by shape classification module, matching hypotheses are generated, resulting in the instantiation of links between known tracked blobs in the previous time slice and potential blobs in the current time slice. Hypotheses are given a measure of confidence derived from a blob dissimilarity measure. Hence by specifying a threshold on the measure of confidence, only possible hypotheses are formed. The similarity measure used here is an approximation, based on distance function [24], normalized area difference and centroid distance.

The matching distance between the two corresponding blobs in the current frame and the corresponding one is conducted by the Ring Partitioned Matching method [13], which provides a fast and robust matching of the images under problems of the scale varying, rotation, and partial occlusion. In this method, the image is partitioned into several ring-shaped areas as shown in Fig. 6.

So the matching distance between two target instances (blobs) x and y in two consecutive time slices (t) and ($t - \Delta t$) is defined as:

$$d_{(x,y)}^{NO} = \sqrt{\sum_i w_i \left(\frac{pN_i}{pN_i + qN_i} - \frac{pO_i}{pO_i + qO_i} \right)^2} \quad (2)$$

where pN_i and qN_i is the number of white and black pixels in the ring i of the blob x , pO_i and qO_i is the number of white and black pixels in the ring i of the corresponding blob y and w_i is the associated weight for that ring. Normalized area difference and centroid distance between two blobs in successive frames is calculated by (3).

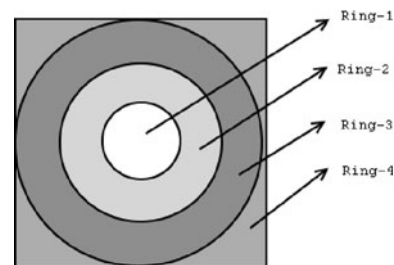


Fig. 6 Ring partition

$$K(x, y) = 1 - \frac{D(x, y)}{D_m} S(x, y) \quad (3)$$

where $D(x, y) \leq D_m$ and $K(x, y) = 0$ otherwise. In (3), $S(x, y)$ is the difference between the area of two blobs, $D(x, y)$ is the centroid distance for two blobs (x and y), and D_m is maximum distance threshold proportional to the image dimensions given as in (4).

$$D_m = \frac{\text{blob}_{\text{width}} + \text{blob}_{\text{height}}}{2} \quad (4)$$

Finally, the similarity measure for matching the two blobs x and y is given by (5), respectively:

$$CM_{(x,y)} = \alpha \left(1 - \frac{d_{(x,y)}^{NO}}{\sqrt{\sum_i w_i}} \right) + (1 - \alpha)k_{(x,y)} \quad (5)$$

where $d_{(x,y)}^{NO}$ and $k_{(x,y)}$ for these two blobs is calculated using (2) and (3), respectively. There are three relationships between the blobs in the successive frames, i.e., *tracked*, *leaving* and *entering* blobs. *Leaving* blobs are blobs in the frame ($t - 1$) that are not identified as a match pair in frame (t), i.e., the blobs in the frame ($t - 1$) that disappear in the frame (t); blobs that have a match pair in frames ($t - T$), $1 \leq T \leq T_{\text{total}}$ are *tracked* blobs, respectively. Here, threshold variables α and T_{total} are determined experimentally.

A road sign is successfully tracked if it labeled *tracked* at least in specific number of $T \leq T_{\text{total}}$ recent frames and the variable S in (6) becomes above certain value.

$$S = \sum_{m=t-T}^t CM_{(x,y)}^m \quad (6)$$

In (6), $CM_{(x,y)}^m$ is the blob (x) similarity measure to its pair (y), in frame m that is calculated using (5) and

t indicates present time. Blobs that successfully tracked are processed for pictogram feature extraction and final recognition.

6 Compute feature vector of inner image based on XFF

In our study, the area of signs where the pictogram is placed follows certain standards. With these standards in mind, the pictogram has been extracted from its frame and after binarization processed for classification. The method of binarization has great impact on correctness of traffic sign recognition. We used self-adaptive image segmentation [25] to extract binary inner pictogram of detected traffic signs. Equation (7) describe this method's formula as follows:

$$P(i,j) = \begin{cases} 1, & \text{if } I(i + H_h, j + L_h) < w \\ 0, & \text{otherwise} \end{cases} \quad (7)$$

and

$$w = a \times \text{avg}_{\text{Rgn}} + b \times \min_{\text{Rgn}}$$

where H_h and L_h represent row and column coordinates of top and left corner of the detected traffic sign region, $I(i + H_h, j + L_h)$ shows pixel's illumination channel value on location $(i + H_h, j + L_h)$ of original natural scene image. avg_{Rgn} and \min_{Rgn} , respectively, represent average and minimal value of all pixel's gray value in traffic sign region. a and b are definite value coefficients in the interval $[0, 1]$. Since each ROI is normalized to 50×50 pixel, this method is robust to scale variation. In Fig. 7, some examples of road sign pictogram binarization using self-adaptive image segmentation method in adverse lighting condition are shown. After binarization of detected sign's pictogram, feature extraction and classification procedures begin.

In pattern recognition, features are used to distinguish one pattern from the others. To design a good sign recognizer, many parameters should be taken into consideration for feature extraction. Firstly, it should present a good discriminative power and low computational cost.



Fig. 7 Results for pictogram binarization with Self-adaptive image segmentation method in different lighting condition

Secondly, it should be robust to the geometrical status of sign, such as the vertical or horizontal orientation and the size or the position variation. Thirdly, it should be robust to noise. We use automatic extraction of fractal features (XFF) method [26] for pictogram feature extraction as novel contribution in RSR field.

In XFF method, the Collage Theorem was applied to extract the meaningful aspects of a 2-D black and white bi-level or gray-level image discriminate fractal encodings that are encoded by iterated function system (IFS). If we consider the bidimensional space R^2 and let \vec{x} be a point in R^2 , an affine transformation has the general form of (8):

$$M(x) = \begin{bmatrix} a & b \\ c & d \end{bmatrix} \vec{x} + \begin{bmatrix} e \\ f \end{bmatrix} \quad (8)$$

where $a, b, c,$ and d express a combination of rotations, scaling, and shears while the values e and f encode the translations. Given (9) an example of such transformation is shown in Fig. 8.

$$M \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} \frac{1}{2} & -\frac{1}{2} \\ \frac{1}{2} & \frac{1}{2} \end{bmatrix} \vec{x} + \begin{bmatrix} 4 \\ 0 \end{bmatrix} \quad (9)$$

The principle underlying XFF is the same as that underlying fractal compression system, looking for a set of transformations that encode the given image. The main difference stands in the fact that it is inspired by the Collage Theorem as originally stated by Barnsley and not by the Collage Theorem for Local IFS. In particular, while the compression systems encode a whole image by searching for a set of transformations that turn parts of an image into other parts of it, XFF focuses only on the shape that is represented; i.e., it looks for IFS that describes the shape discarding the background. As a consequence, the representations obtained are more concise than those obtained in the case of compression [27]. In XFF, image is uniquely identified by the parameters of the IFS whose fixed point is the image itself. The parameters of such approximation, IFS_{appr} , will be used as descriptive features for the processed image. Since XFF focuses on representing isolated objects, in this case, we used this method to represent the features of K objects with the biggest area in each

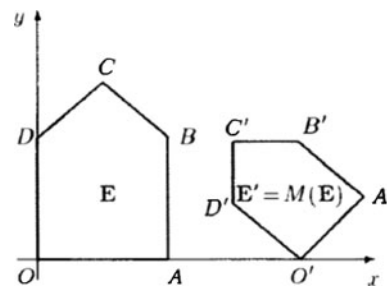


Fig. 8 E is mapped on E' by means of (9)

pictogram In Fig. 9, the flowchart for extracting XFF features from a detected sign’s pictogram has been described.

IFS_{appr} for each object in the pictogram of detected sign is obtained by covering the object with a number of contractions of it. In XFF first, a set of transformations that are parametric is built that is called proto-transformations. The proto-transformations are obtained either by rotating a

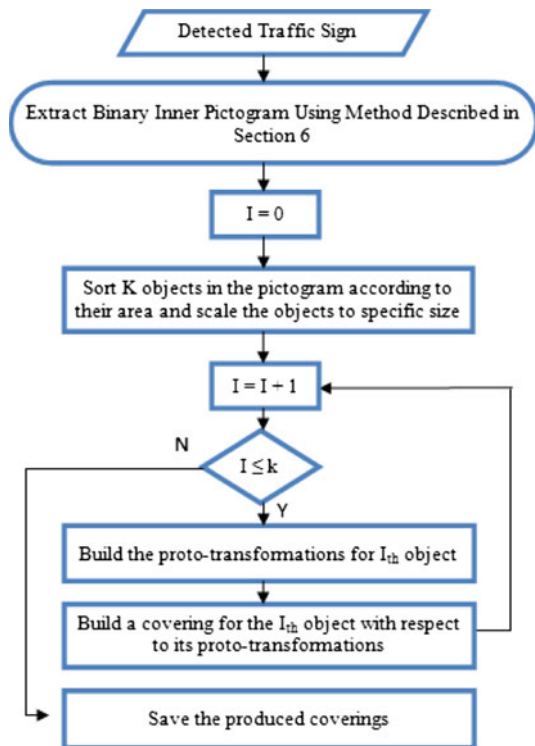


Fig. 9 Flowchart for extracting XFF features from pictogram of detected sign

Fig. 10 The pseudo code for extracting XFF features for each object in pictogram of detected sign

1. **begin**
2. $IFS_{appr} := \emptyset;$
3. $currBest := null;$
4. **while** the *stop criterion* is not verified **do**
5. **for** $x \in [1, width]$ and $y \in [1, height]$ **do**
6. instantiate each proto-transformation by setting parameter e to x and parameter f to y , thus obtaining a set of candidate transformations; evaluate each candidate transformation and select the best one;
7. **if** it scores better than the current best candidate transformation
8. **then** $currBest :=$ selected transformation;
9. **end if**
10. **end for**
11. increment x and y of a predefined step;
12. **end while**
13. $IFS_{appr} := IFS_{appr} \cup \{currBest\};$
14. **end while**
15. sort IFS_{appr} according to a *polar ordering*
16. **end.**

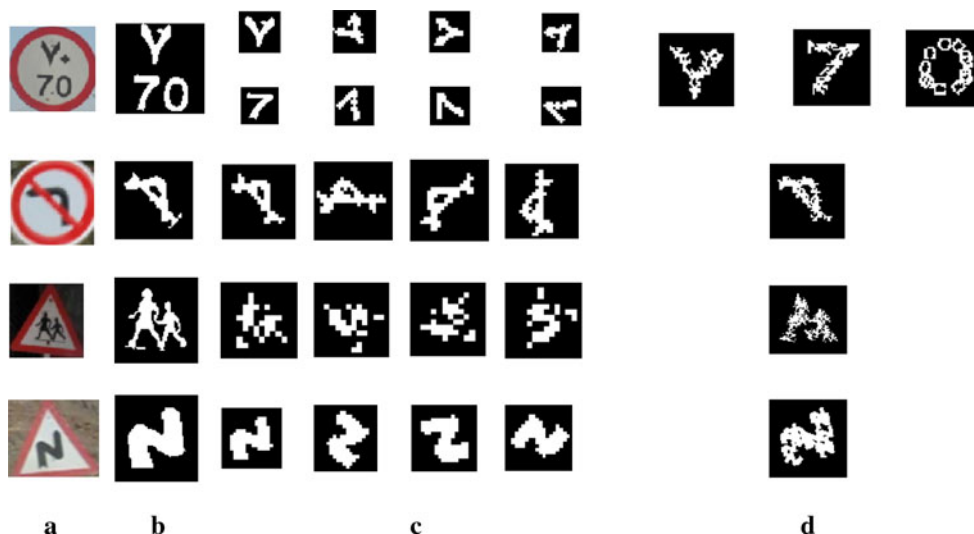
scaled copy of the original image or by flipping a previously built proto-transformation around the X or around the Y axes. The scaling factor, the number of proto-transformations, and the angles that define the rotations are decided experimentally. The same scaling factor is used to reduce both the width and the height of the image. Once the proto-transformations are ready, the encoding phase for the object can start. In Fig. 10, the pseudo code for extracting XFF features for each object in pictogram of detected sign has been described.

In Fig. 10, *width* and *height* are the width and the height of the original object in pictogram. In words, at each iteration, a set of candidate transformations is produced on a regular grid of points, then a scoring criterion is applied to each candidate transformation, the best one is selected and added to IFS_{appr} , and the pixels that are covered by the selected transformation are marked. Note that IFS_{appr} can contain different instantiations of the same proto-transformation.

The stop criterion is, alternatively, a test on the number of iterations (go on until N transformations have been selected) or a test on the amount of foreground that has been covered (go on until $\%N$ of the foreground pixels have been covered).

The outcome of XFF is an IFS $\{M_1, \dots, M_m\}$. Each transformation M_k is coded by three numbers, an identifier of the proto-transformation it derives from (this is possible because the number of proto-transformations is finite) and parameters e and f . Thus, the number of features required to encode K object in pictogram is $K \times m \times 3$. Figure 11a contains the original pictures of some detected signs in real road scenes with different conditions. Figure 11b represents binaries instances of the pictogram and in Fig. 11c and d four self-affine transformations and fractal reconstruction of objects in

Fig. 11 **a** Original pictures of the detected sign. **b** Binaries instances of the each sign's pictogram. **c** Four self-affine transformations of each object in pictogram. **d** Fractal reconstruction of objects in pictogram using 16 contracted copies of itself



pictogram using 16 contracted copies of it is shown, respectively.

7 Sign classification

In the proposed method, after successfully extracting the pictogram features, sign classification module relate these features to one of the target sign classes. For sign recognition, we used C type SVM with kernel function. A solution in cases the data cannot be separated by a linear function is to map the input data into a different space $\varphi(x)$ by using kernel function. Due to the fact that the training data are used through a dot product, if there is a “kernel function,” so that we satisfy $K(x_i, x_j) = \langle \varphi(x_i), \varphi(x_j) \rangle$, we can avoid computing $\varphi(x)$ explicitly and use the kernel function $K(x_i, x_j)$.

In this work, we used a radial basis function (RBF) kernel since other kernels gave worse results and also for its excellent performance, as follows:

$$K(x_i, x_j) = e^{-\gamma |x_i - x_j|^2} \quad (10)$$

Given input vector x , C type SVM find the minimum value of $\varphi(x)$ in (11) for binary classification:

$$\varphi(w) = \frac{1}{2} w^T w + C \sum_{i=1}^l \xi_i \quad (11)$$

Subject to

$$y_i (w^T \varphi(x_i) + b) + \xi_i \geq 1, \quad \xi_i \geq 0, i=1, \dots, l \quad (12)$$

Whereas ξ_i can be understood as the error of the classification, C is the penalty parameter for this term, and l is the number of training samples. Then, the decision function (1) can be revised as (13):

$$f(x) = \text{sgn} \left[\sum_{i=1}^{N_s} \lambda_i y_i K(x_i, X) + b \right] \quad (13)$$

where N_s is the number of support vectors, X is input vector, x_i are the support vectors, and $K(x_i, X)$ is the kernel function. To search for the decision region, all feature vectors of a specific class are grouped together against all vectors corresponding to the rest of classes (including noisy objects), following the one versus all classification algorithm. The following procedure is performed to obtain the best performance in the training procedure:

1. Calculate the feature vector from candidate signs.
2. Use cross-validation to find the best parameter C and γ
3. Use the best parameter C and γ to train the whole training set

8 Results

In order to assess the effectiveness of proposed method and comparison to other methods for road sign detection and recognition, experiments were run and conducted results are described here.

In general, the quality of the results obtained by any study on RSR varies from one research group to another mainly due to the lack of a standard database of road images [18]. For example, it is impossible to know how well one system is robust to variation of illumination of the images, because in the different studies, usage of images with low illumination in experiments is usually not specified. Also, some other researches are based on a small set of images, as the task of collection of a set of road scene images with different condition takes much time. For example, authors in [16] and [28] used little more than a

hundred sign images. To avoid these problems in our study, a large road sign image dataset is produced and used to train and test the detection and shape classification modules. Also for testing our tracking algorithm, several video sequences have been captured with fluctuating speed over a stretch of approximately 15 km and during both the day and at night using PAL non-interlaced video (frames of 640×480 pixels). In addition, these images and video sequences cover a wide sort of weather conditions including sunny, cloudy, and rainy with signs that have temporal occlusion, rotation, and other problems mentioned in Sect. 1. This database has been created in association with Dr. Mahdi Yaghoobi from the Artificial Intelligence group in Islamic Azad University of Mashhad [29] and is soon to be made freely available to the scientific community.

Using 38 test sequences containing 438 test candidates, the results shown in Table 1 obtained for accuracy of the detection, shape classification, and tracking modules. Tracking of signs was performed by means of (6), and after testing several values for parameters, the best output occurred when the α and T_{total} were set to 0.55 and 13, respectively. In addition, in our experiments, the value of each w_i in (2) was set to 1, 1, 0.5, and 0.25 for $i = 1, 2, 3,$ and 4. According to the results, almost all road signs have been correctly detected. Table 1 also shows the total number of traffic signs that appear in each lighting

condition and the number of correctly tracked signs (each sign is detected multiple times in the sequence) and shows the accuracy of 96% in detection process. Also the confused shape classification can be attributed to objects with color and shape similar to traffic signs that are appear in road scenes and have been rejected by the recognition module. As Table 1 indicates, in night lighting condition due to absence of noisy objects in road scenes and because of high reflectivity of light in road signs, the false alarm value becomes less compare to other lighting conditions.

As Fig. 12 shows, there are three types of sign we consider for classification here: type I and type II represent circular giving orders and triangular with white background signs and type III represents circular mandatory signs. The experimental data used to training and testing the SVM models for pictogram classification were selected from our image database of manually cropped road signs that consist of 1,650 data samples. Data samples contained 10 categories of type I, 15 categories of type II, and 8 categories of type III road signs. Each category had 50 samples and divided randomly into two data sets, 30 for training and 20 for testing in each experiment.

Using these data samples, in order to determine the proper scaling factor in XFF coding procedure, which concludes satisfactory approximation of whole pictogram, we repeat our experiment with different values and the accuracy of pictogram classification for each one is shown in Fig. 13. Obviously for values less than 0.2, the XFF method becomes more sensitive to noise in objects and may give deferent fractal code for little changes in their shape. Furthermore, we evaluate the performance of the XFF feature representation method in comparison with existent methods such as the geometric and Zernike moments, principal component analysis (PCA), and binary representation as used in [20] and [4, 21], respectively in accuracy for pictogram classification. All experiments were run on a Pentium 4 3.2 GHz PC with 1,024 MB DRAM,

Table 1 The accuracy rate for detection and tracking modules

Weather conditions	Sunny (daylight)	Cloudy (rainy)	Night
Traffic signs	153	145	140
Successfully tracked	150	141	138
False alarm	31	23	12
Confused shape classification	11	8	5

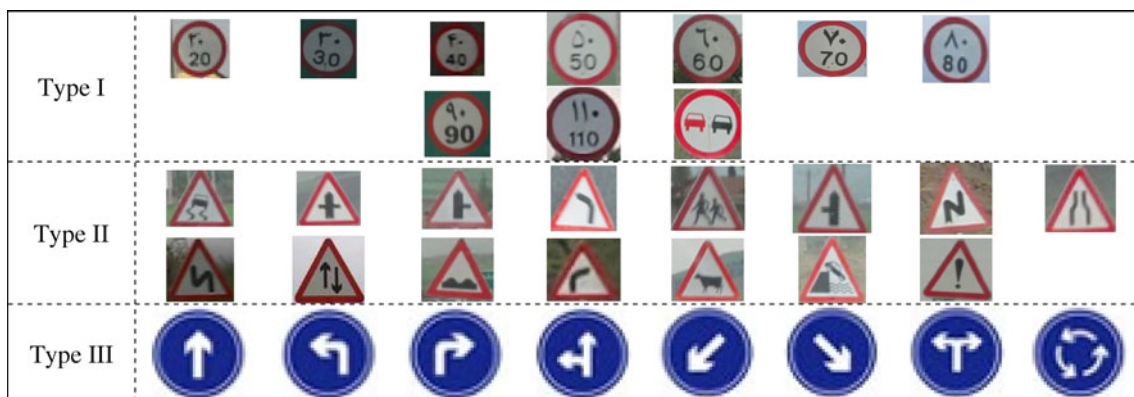


Fig. 12 Types of road sign considered to classification: Type I and Type II represent circular giving orders and triangular with white background signs and Type III represents circular mandatory signs

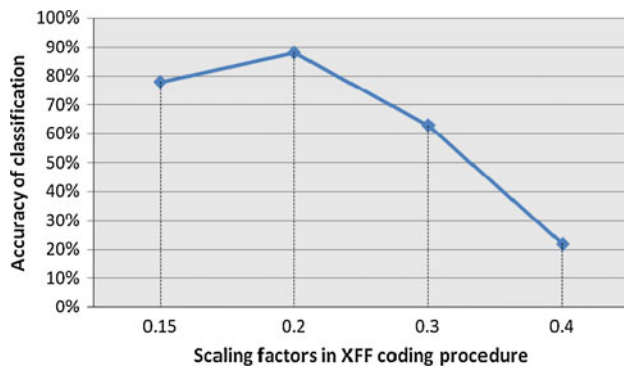


Fig. 13 Accuracy of pictogram classification for each value of scaling factors in XFF coding procedure

Table 2 The performance of the XFF feature representation method in comparison with other methods

Feature extraction method	Type of signs	Recognition accuracy (%)	Overall accuracy (%)
Haar features	Type I	75	72
	Type II	68	
	Type III	73	
Zernike moments	Type I	82	83
	Type II	87.8	
	Type III	78	
Binary	Type I	84	81
	Type II	79	
	Type III	80	
PCA	Type I	82	85
	Type II	88	
	Type III	86	
XFF	Type I	90	88
	Type II	86	
	Type III	88	

and the SVMs software used is SVM_V0.51 [30]. Results of these experiments are shown in Table 2.

Results in Table 2 lead us to a number of conclusions: PCA does not seem to be a satisfactory method for classifying similar classes because it only takes into account the total variance of the data instead of class membership. Also, the AdaBoost classifier with Haar Features, despite being more complex, did not outperform XFF method. Binary representation method is not invariant to rotation and occlusion, so its performance becomes low for large datasets with such samples.

With regard to the execution speed, the recognition module with an average single-image processing time lower than 75 ms can easily be integrated in a real-time road sign recognition system.

9 Conclusion

This paper has presented a new method to detect, track, and recognize traffic signs in a video sequence, considering all the difficulties for this field. A new similarity measure has also been introduced to track detected blobs, and pictogram classification has been based on fractal features, which is a novel contribution in this field. For comparative examination of XFF feature selection algorithm, a number of separate experiments by using other popular methods for representation of road sign's pictogram have been performed. In the experiments, the proposed method has been compared with the other methods and the result is promising.

In addition for tracking the sign, the system presents information about its state and so the confidence of detection process for each sign is known. This will help the recognition module in a constant mode when the same sign appears in several images. Due to this awareness of the sign state, the system can be useful for other applications such as maintenance and inventories of traffic sign in highways and or cities. We consider using other shape classification techniques and SVM classifier structures as extension of this work.

References

- Gomez-Moreno H, Lopez-Ferreras F (2007) Road-sign detection and recognition based on support vector machines. *IEEE Trans Intell Transportation Syst* 8(2):264–278
- Maldonado-Bascón S, Acevedo-Rodríguez J, Lafuente-Arroyo S, Fernández-Caballero A, López-Ferreras F (2010) An optimization on pictogram identification for the road-sign recognition task using SVMs. *Comput Vis Image Underst* 114(3):373–383
- Bahlmann C, Zhu Y, Ramesh V, Pellkofer M, Koehler T (2005) A system for traffic sign detection, tracking, and recognition using color, shape, and motion information. In: *IEEE intelligent vehicles symposium (IV 2005)*, Las Vegas, NV, June 2005
- de la Escalera A, Moreno L, Salichs MA, Armingol JM (1997) Road traffic sign detection and classification. *IEEE Trans Ind Electron* 44(6):848–859
- de la Escalera A, Armingol JM, Mata M (2003) Traffic sign recognition and analysis for intelligent vehicles. *Image Vis Comput* 21:247–258
- Miura J, Kanda T, Shirai Y (2000) An active vision system for real-time traffic sign recognition. In: *Proceedings of IEEE intelligent transportation system*, 1–3 Oct 2000, pp 52–57
- Gomez-Moreno H, Maldonado-Bascon S, Gil-Jimenez P, Lafuente-Arroyo S (2010) Goal evaluation of segmentation algorithms for traffic sign recognition. *IEEE Trans Intell Transport Syst* 11(4):917–930
- Fang C, Chen S, Fuh C (2003) Road sign detection and tracking. *IEEE Trans Veh Technol* 52(5):1329–1341
- Aoyagi Y, Asakura T (1996) A study on traffic sign recognition in scene image using genetic algorithms and neural networks. In: *Proceedings of 22nd IEEE international conference on industrial electronics, control instrumentation*, Taipei, Taiwan, Aug 1996, 3, pp 1838–1843

10. Garcia-Garrido M, Sotelo M, Martin-Gorostiza E (2006) Fast traffic sign detection and recognition under changing lighting conditions. In: Sotelo M (ed) Proceedings of the IEEE ITSC pp 811–816
11. Liu Y, Ikenaga T, Goto S (2006) Geometrical, physical and text/symbol analysis based approach of traffic sign detection system. In: Proceedings of IEEE intelligent vehicle symposium, Tokyo, Japan, June 2006, pp 238–243
12. Baro X, Escalera S, Vitria J, Pujol O, Radeva P (2009) Traffic sign recognition using evolutionary adaboost detection and forest-ECOC classification. *IEEE Trans Intell Transport Syst* 10(1):113. doi:12610.1109/TITS.2008.2011702
13. Soetedjo A (2007) Improving the performance of traffic sign detection using blob tracking. *IEICE Electron Express* 4(21):684–689
14. Vicen Bueno R, Gil-Pita R, Rosa-Zurera M, Utrilla-Manso M, Lopez-Ferreras F (2005) Multilayer perceptrons applied to traffic sign recognition tasks. In: Proceedings of the 8th international work-conference on artificial neural networks, IWANN, Vilanova i la Geltru, Barcelona, Spain, pp 865–872
15. Hibi T (1996) Vision based extraction and recognition of road sign region from natural color image, by using HSL and coordinates transformation. In: Proceedings of the 29th international symposium automotive technology and automation, Florence, Italy, 1996
16. Hsu SH, Huang CL (2001) Road sign detection and recognition using matching pursuit method. *Image Vis Comput* 19:119–129
17. Ruta A, Li Y, Liu X (2010) Robust class similarity measure for traffic sign recognition. *IEEE Trans Intell Transport Syst* 11(4):846–855
18. Prietoa MS, Allen AR (2009) Using self-organising maps in the detection and recognition of road signs. *Image Vis Comput* 27(6):673–683
19. Nguwi Y–Y, Kouzani AZ (2008) Detection and classification of road signs in natural environments. *Neural Comput Appl* 17(3):265–289
20. Gilani SH (2007) Road sign recognition based on invariant features using support vector machine. Master Thesis, Faculty of Electrical and Computer Engineering, Dalarna University, Sweden
21. Fleyeh H, Shi M, Wu H (2008) Support vector machines for traffic signs recognition. In: IEEE international joint conference on neural networks, 2008. IJCNN 2008 (IEEE World Congress on Computational Intelligence), pp 3820–3827
22. Lafuente-Arroyo S, Gil-Jiménez P, Maldonado-Bascón R, López-Ferreras F, Maldonado-Bascón S (2005) Traffic sign shape classification evaluation I: SVM using distance to borders. In: Proceedings of the IEEE intelligent vehicle symposium, Las Vegas, NV, June 2005, pp 557–562
23. Franc¸is A (2004) Real-time multi-resolution blob tracking. IRIS Technical Report, 2004
24. Soetedjo A, Yamada K (2005) Traffic sign classification using ring partitioned method. *IEICE Trans Fundam Electron Commun Comput Sci* E88(9):2419–2426
25. Zhang K, Sheng YH, Gong ZhJ (2007) A self-adaptive algorithm for traffic sign detection in motion image based on colour and shape features. In: Proceedings of SPIE, geoinformatics 2007: remotely sensed data and information, Nanjing, China, 6752, pp 6752G_1–6752G_13
26. Baldoni M, Baroglio C, Cavagnino D (2000) Use of IFS codes for learning 2D isolated-object classification systems. *J Comput Vis Image Underst* 77(3):371–387
27. Baldoni M, Baroglio C, Cavagnino D (1998) XFF: a simple method to eXtract fractal features for 2D object recognition. In: Amin A, Dori D, Pudil P, Freeman H (eds) Advances in pattern recognition, joint iapr international workshops SSPR'98 and SPR'98, Springer, Berlin/New York, pp 382–389
28. Piccioli G, Micheli ED, Parodi P, Campani M (1996) Robust method for road sign detection and recognition. *Image Vis Comput* 14:209–223
29. Artificial Intelligence Group in Islamic Azad University of Mashhad, available from World Wide Web. Accessed April 2010. <http://en.mshdiau.ac.ir/index.php>
30. Platt JC. Fast Training of support vector machines using sequential minimal optimization (Online). Available: <http://research.microsoft.com/~jplatt>