In-Vehicle Camera Traffic Sign Detection and Recognition

Andrzej Ruta, Fatih Porikli, Yongmin Li, Shintaro Watanabe

TR2009-081 December 2009

Abstract

In this paper we discuss theoretical foundations and a practical realization of a real-time traffic sign detection, tracking and recognition system operating on board of a vehicle. In the proposed framework a generic detector refinement procedure based on a mean shift clustering is introduced. This technique is shown to improve the detection accuracy and reduce the number of false positives for a broad class of object detectors for which a soft response's confidence can be sensibly measured. Track of an already established candidate is maintained over time using an instance-specific tracking function that encodes the relationship between a unique feature representation of the target object and the affine distortions it is subject to. We show that this function can be learned on-the-fly via regression from random transformations applied to the image of the object in known pose. Secondly, we demonstrate its capability of reconstructing the full-face view of a sign from substantial viewangles. In the classification stage a concept of a similarity measure learned from image pairs is discussed and its realization using SimBoost, a novel version of AdaBoost algorithm, is analyzed. Suitability of the proposed method for solving multi-class traffic sign classification problems is shown experimentally for different image representations. Overall performance of the entire system is evaluated based on a prototype C++ implementation. Illustrative output generated by this demo application is provided as a supplementary material attached to this paper.

Journal of Machine Vision & Applications

Machine Vision and Applications manuscript No.
(will be inserted by the editor)

In-Vehicle Camera Traffic Sign Detection and Recognition

Andrzej Ruta · Fatih Porikli · Yongmin Li · Shintaro Watanabe

Received: dd-mm-yyyy / Accepted: dd-mm-yyyy

Abstract In this paper we discuss theoretical foundations and a practical realization of a real-time traffic sign detection, tracking and recognition system operating on board of a vehicle. In the proposed framework a generic detector refinement procedure based on a mean shift clustering is introduced. This technique is shown to improve the detection accuracy and reduce the number of false positives for a broad class of object detectors for which a soft response's confidence can be sensibly measured. Track of an already established candidate is maintained over time using an instance-specific tracking function that encodes the relationship between a unique feature representation of the target object and the affine distortions it is subject to. We show that this function can be learned on-the-fly via regression from random transformations applied to the image of the object in known pose. Secondly, we demonstrate its capability of reconstructing the full-face view of a sign from substantial viewangles. In the classification stage a concept of a similarity measure learned from image pairs is discussed and its realization using *SimBoost*, a novel version of AdaBoost algorithm, is analyzed. Suitability of the proposed method for solving multi-class traffic sign classification problems is shown experimentally for different image representations. Overall performance of the entire system is evaluated based on a prototype C++ implementation. Illustrative output generated by this demo application is provided as a supplementary material attached to this paper.

Keywords traffic sign recognition · confidence-weighted mean shift · regression tracking · SimBoost

1 Introduction

Road signs are an inherent part of the traffic environment. They are designed to regulate flow of the vehicles, give specific information to the traffic participants, or warn against unexpected road circumstances. Perception and fast interpretation of road signs is crucial for the driver's safety. Public services responsible for the traffic infrastructure maintenance mount the signs on poles on the road sides, over highway lanes, and in other places in a way ensuring that they are easy to spot without distracting the driver's attention from manoeuvring the vehicle. Also the sign pictograms are designed and standarized in accordance with a rule of maximizing simplicity and distinctiveness. However, certain circumstances like high visual clutter, adverse illumination, or bad weather conditions can significantly hamper perception of traffic signs. Purely physiological factors like excitement, irritation or fatigue are known to further reduce the visual concentration of a human and can hence put the the driver's life at risk, while driving at high speeds in particular. For the above reasons, automation of the road sign detection and recognition

Andrzej Ruta · Yongmin Li
School of Information Systems, Computing & Mathematics
Brunel University
Uxbridge, Middlesex UB8 3PH United Kingdom
Tel.: +44-1895-265850
Fax: +44-1895-251686
E-mail: {Andrzej.Ruta,Yongmin.Li}@brunel.ac.uk

Fatih Porikli
Mitsubishi Electric Research Laboratories
201 Broadway
Cambridge, MA 02139
USA
Tel.: +1-617-6217586
E-mail: fatih@merl.com

Shintaro Watanabe
Some institution
Some address

process was found a natural direction to follow as soon as video processing became attainable on a computer machine. Today, it is considered a critical task in the contemporary driver support systems, but their reliability remains still beyond our expectations and a large space for improvement is left.

1.1 Related Work

Different approaches were used in the past for detecting road signs. In the older studies, e.g. [3,4], as well as in many recent ones, e.g. [10,12,19], it was common to employ a heuristic that utilized available prior knowledge about the traffic signs to 1) define how to presegment the scene in order to find the interest regions, and 2) define the acceptable geometrical relationships between the sign parts with respect to color and shape. The major deficiency of these methods were a lack of solid theoretical foundations and high parametrization. A more convincing, parameter-free method for detecting road signs was proposed by Bahlmann et al. [13] who utilized a trainable cascade of boosted classifiers to learn the most discriminative local image descriptors for building a sign detector. Their system demonstrated a good detection rate and was reported to yield very few false alarms at an average processing speed of 10 fps. In several studies, e.g. [3,7,10], the problem of tracking of the observed road signs over time was addressed. However, the proposed frameworks, with the exception of the two-camera system in [7], never went beyond a relatively simple scheme based on a predefined motion model and some sort of geometrical Kalman filtering.

For sign classification, a baseline approach involves a cross-correlation template matching. It was used for example in [3]. This technique is known to be useful only on condition that the object in the tested image can be well aligned with the templates. Other, feature-based methods involved neural networks [10,21] or kernel density estimation [8] and were shown to offer relatively good classification accuracy. Gao et al. [15] employed the biologically-inspired vision models to represent both color and shape features of the traffic signs. They achieved a promising recognition rate for static images of signs affected by substantial noise and perspective transformations. An interesting concept of a trainable, class-specific similarity measure was introduced recently by Paclík et al. [16] who demonstrated a usefulness of this method in solving relatively simple road sign classification problems. A similar approach was further presented by Ruta et al. [19] who adapted it to infer the discriminative sign representations using a single template image per class.

In this paper we present a unified framework for detection, tracking and recognition of traffic signs which alleviates the shortcomings of many previous approaches. At the detection stage we focus on the problem of high sensitivity of the existing object detection techniques. A generic refinement procedure based on a modified mean shift clustering of the detector responses is proposed and evaluated with two different approaches. The best-performing refined detector is selected for the prototype system implementation. For tracking od the existing road sign candidates we employ a trainable regression function that compensates the affine distortions of the target, making our detector pose-invariant and hence more accurate. Ability of the proposed tracker to reconstruct the full-face view of a sign under affine distortions is shown experimentally using synthetic image sequences. Finally, we build a traffic sign classifier based on the concept of a trainable similarity. A novel AdaBoost-like algorithm, called *SimBoost*, is utilized to learn a robust sign similarity measure from image pairs labeled either “same” or “different”. This measure is further directly used within the *winner-takes-all* classification framework to discriminate between multiple road sign classes. The discriminative power of the classifiers trained using SimBoost is demonstrated for different feature representations of the image. Apart from testing the proposed detection, tracking and recognition approaches as standalone algorithms, we also build a demo implementation of a real-time system incorporating all three components. This system is evaluated using real-life video captured from a moving vehicle in urban traffic scenes.

The rest of this paper is divided into five parts. In section 2 our road sign detection method is discussed. In section 3 we develop a pose-invariant sign tracker. Section 4 explains how the concept of trainable similarity is used to construct a robust road sign classifier. In section 5 an extensive experimental evaluation of our algorithms is presented. Finally, in section 6 we conclude our work.

2 Sign Detection

Traffic sign detection is a difficult problem as it involves discriminating a large gamut of diverse objects from a generally unknown background. Taking this diversity into account, we focus in this work on a subset of circular signs that are well-constrained in terms of the size, shape, and contained ideogram. In section 2.1 a fast, application-specific quad-tree focus operator is introduced. We use it to quickly discard the irrelevant fragments of the scene and locate the sparse regions that might contain traffic signs. In section 2.2

we briefly discuss two practically useful sign detection methods. Common limitations of these methods are also analyzed. In section 2.3 a detection refinement scheme is proposed in order to improve the selectivity and accuracy of the over-sensitive detector.

2.1 Quad-tree Focus of Attention

In order to detect the new road sign candidates emerging in the scene, it is first necessary to reduce the search area. Dense scanning of the entire image wastes processor time and is unlikely to work in real time, even using a detector based on the well-known Haar wavelets [11], probably computationally the simplest available image descriptors. One generic method for quick elimination of the irrelevant regions of an image is a rejection classifier cascade introduced by Viola and Jones [11]. Not denying the potential of this technique, we should yet note that it still involves a sequential, pixel-by-pixel processing of the input image and requires complex and time-consuming training. Below we briefly outline a much simpler generic search reduction technique that is tuned to our specific application, and which can be used solely or in chain with other methods.

The proposed quad-tree attention operator associates a scalar feature value $\mathrm{v}(x, y)$ with each pixel of the image I: $\mathbf{V}(I) = \{\mathrm{v}(x, y) : x = 1, \ldots, W, y = 1, \ldots, H\}$, where $W \times H$ is the image size. A region $R(x_1, y_1, x_2, y_2)$ is considered relevant if the sum of the contained pixels' feature values is greater that a predefined threshold t. If an integral feature image is available:

$$\begin{array}{c}\mathbf{\Sigma}(I) = \{\upsilon(x, y) : \upsilon(x, y) = \sum_{i \le x, j \le y} \mathrm{v}(i, j), \\ x = 1, \ldots, W, y = 1, \ldots, H\}\end{array}, \quad (1)$$

then this sum can be computed using only 4 array referencing operations and 4 additions/subtractions:

$$\begin{array}{r}\mathrm{v}(R(x_1, y_1, x_2, y_2)) = \upsilon(x_2, y_2) - \upsilon(x_1, y_2) - \\ \upsilon(x_2, y_1) + \upsilon(x_1, y_1)\end{array}. \quad (2)$$

If the threshold t is set to an appropriately low value that can be used to reliably discriminate between the relevant and irrelevant fragments of the scene at the highest considered resolution, then the RoI-s can be rapidly identified using the following recursive algorithm:

Algorithm 1 is illustrated in Fig. 1. We tailor it to our needs by associating the relevance of a given image region with the amount of contained contrast measured with respect to the appropriate color channels. The traffic signs we focus on always have a distinctive color rim. Therefore, the input image is first filtered using the appropriate set of filters intended to amplify a certain color and suppress any other. Suitable filters used in this work are:

Algorithm 1 Quad-tree RoI selection.

```
input: image $I_{W \times H}$, minimum "amount" of feature contained
    in a RoI, $t$, minimum region size $s$
output: set of RoI-s, $S$
build a feature map $\mathbf{V}(I)$
build an integral feature map $\mathbf{\Sigma}(I)$
initialize an empty set of relevant smallest-scale regions $\mathbf{C} = \emptyset$
call $ProcessRegion(R(1, 1, W, H), t, s, \mathbf{C})$
cluster regions in $\mathbf{C}$
populate $S$ with bounding rectangles of found clusters
```

Algorithm 2 Procedure *ProcessRegion*.

```
input: region $R_{w \times h}$, minimum "amount" of feature contained
    in a RoI, $t$, minimum region size $s$, a set of relevant smallest-
    scale regions $\mathbf{C}$
compute the amount of feature in $R$
if $\min\{w, h\} \ge s$ then
  if $v(R) > t$ then
    set $w = w/2$, $h = h/2$
    for each quarter $Q_j$ of $R$ do
      call $ProcessRegion(Q_j, t, s, \mathbf{C})$
    end for
  end if
else
  add $R$ to $\mathbf{C}$
end if
```

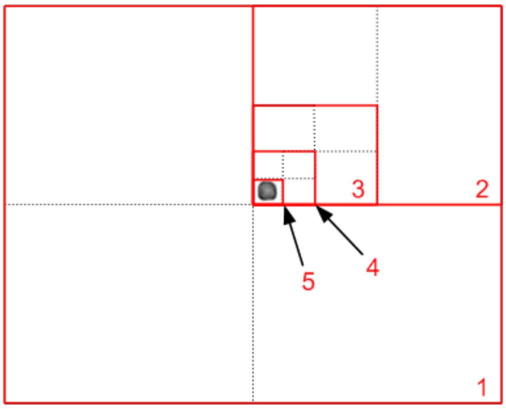


Fig. 1 Quad-tree interest region finding algorithm. The consecutive numbers correspond to the order of quarters being processed.

$$\begin{array}{l}f_R(\mathbf{x}) = \max(0, \min(\frac{x_R - x_G}{s}, \frac{x_R - x_B}{s})) \\ f_B(\mathbf{x}) = \max(0, \min(\frac{x_B - x_R}{s}, \frac{x_B - x_G)}{s}))\end{array}, \quad (3)$$

where x_R, x_G, x_B denote the red, green and blue components of an input pixel and $s = x_R + x_G + x_B$. The above filters effectively extract the red and blue fragments of the image, which is shown in Fig. 2.

The RoI selection algorithm starts with applying filters (3) to the input image. Then, two feature images $\mathbf{V}_R(I)$ and $\mathbf{V}_B(I)$ are constructed as gradient magnitude maps for each color. Similarly, two integral images $\mathbf{\Sigma}_R(I)$ and $\mathbf{\Sigma}_B(I)$ are build from $\mathbf{V}_R(I)$ and $\mathbf{V}_B(I)$ respectively. A region corresponding to the entire image is

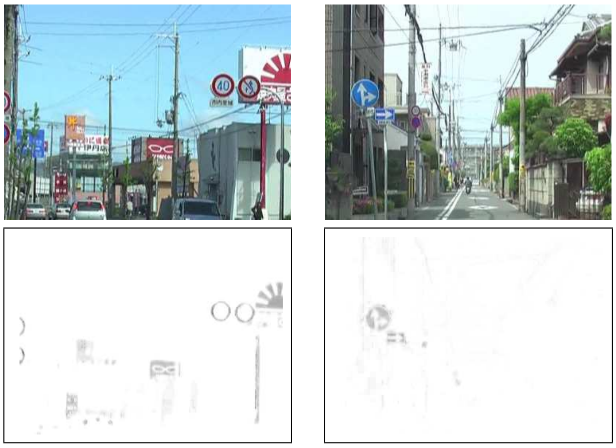


Fig. 2 The effect of applying the color filters (3) to the example RGB images (top row): red color filter (bottom-left), blue color filter (bottom-right). This figure is best viewed in color.

now checked against the total color gradient contained using a maximum of the values picked from both integral images. As it is typically far above the predefined threshold, the image is subdivided into four quarters and each quarter is recursively processed in the same way. The process is stopped either when the current input region contains less gradient than the threshold or upon reaching the minimum region size. The above-threshold lowest-level regions are further clustered and the ultimate RoI-s are constructed as bounding rectangles of the found clusters. This way we can very quickly discard the irrelevant fragments of the scene, e.g. sky or asphalt, which either do not contain the interest colors and/or are too low-contrasting. Note that the total amount of color-specific gradient constitutes a much stronger filter than simply the total amount of characteristic color, which fails in presence or uniform reddish or blueish regions, e.g. sky or large color billboards.

Further processing is done only in the found interest regions. Below, the two useful techniques for traffic sign detection are presented.

2.2 Sign Detectors

Traffic sign detector must be both sufficiently discriminative and computationally inexpensive so that it is able to work in real time even in the worst-case scenario, when a large part of the scene has to be scanned. We evaluate here two detection techniques which seem particularly useful for road sign detection: Haar rejection cascade and the Hough transform.

Haar cascade of boosted classifiers for object detection has been thoroughly discussed in [11]. This technique revolves around an idea of building a multi-stage classifier in which at each new layer the layer-specific binary classifier is trained in a supervised way using all available true positive images and only these negative, i.e. background images that were misclassified in the previous layer. This way the cascade is arranged in such a way that in runtime the most top-level classifier can quickly reject most of the irrelevant parts of the scene, leaving the more ambiguous regions to process by the classifier in the next layer. This recursive process is further continued for the increasingly hard regions and only the regions successfully passing the last layer are retained. AdaBoost algorithm [5] is used to train the classifier in each layer and the expected performance specifications are given as the training parameters. For example, the boosted classifier in each layer might be set to grow until it can correctly classify 99% of the true positives form the previous layer and not less than 50% of the previous layer's false positives. The third parameter, maximum overall false positive rate of the cascade is provided to determine when to stop the training process. Robustness of the cascade setup in combination with using simple Haar wavelet filters underlying each weak classifier make the cascade relatively inexpensive in terms of the computation involved.

Although there is a common agreement on the usefulness of the rejection cascade for general object detection, this approach has also many disadvantages. First and foremost, it may be insufficiently discriminative if the intra-class variability is too high. Secondly, it involves a very expensive training and requires large volumes of data. In addition, many implementation details are technically demanding. For example, it is unclear how to generate negative images to populate the training pool of the classifiers located deep in the cascade, say, at the n-th level. An overall false positive rate of the cascade up to the level $n-1$ might already be very low. This implies that random selection of the background regions from the images not containing the target object might be extremely time-consuming.

The second detection technique we evaluate is based on the Hough transform (HT) [2]. The purpose of this method is to find the imperfect instances of objects within a certain class of shapes by a voting procedure carried out in a parameter space. The simpler the parametric description of a shape, the more suitable this approach is in real-time vision. In our case, most of the popular road signs are either circles or equiangular polygons: equilateral triangles, squares, or octagons (STOP sign), depending on the country. To detect circular structures in an image, a well-known circular Hough transform can be used which involves voting in a three-parameter space. For regular polygons a generalized method has been proposed by Loy et al. [12]. A desirable property of these HT variants is their accuracy and tolerance to noise and partial occlusions. Among major disadvantages is their sensitivity to the quality of the input edge map, which in turn depends on the external factors, like scene illumination.

Both techniques are known to suffer from the problem of producing multiple, mostly redundant, positive hypotheses around the true target object candidates. As processing of each such hypothesis separately is impractical, the output of an over-sensitive detector is typically subject to some sort of postprocessing intended to produce a single accurately fit shape per object instance. Below we propose such a postprocessing technique.

2.3 Confidence-Weighted Mean Shift Refinement

Accuracy of an over-sensitive detector that produces redundant positive hypotheses around the true object candidates must necessarily be improved to make it useful for real-time operation. One possible way of doing that is to consider the detector's response space a probability distribution with modes to be found. Mean shift algorithm [9] is a well-established kernel density estimation technique that can be used to find the modes of the underlying distribution. However, the original mean shift formulation does not account for the possibly varying relevance of the input points. Below we propose a simple modification, called *Confidence-Weighted Mean Shift*, which alleviates this shortcoming by incorporating the confidence of the detector's responses into the mode finding procedure. It is shown that such a refinement procedure can be applied to the output of any detector that yields a soft decision or can be modified to do so.

We first characterize each positive hypothesis of the detector with a vector, $\mathbf{x}_j = [x_j, y_j, s_j]$, encoding the object's centroid position and its scale. In addition, $\mathbf{x}_j$ is assigned a confidence value, q_j which is related to the soft response of the detector. In the case of a single boosted classifier in each layer of the Haar cascade, such a confidence measure can naturally be related to the distance of the response from the linear decision boundary:

$$q_j = q(\mathbf{x}_j) = \sum_{t=1}^{T} \alpha_t h_t(\mathbf{x}_j) \ , \tag{4}$$

where $h_t(\mathbf{x}_j)$ denote the weak classifier responses, $\alpha_t = \log(\frac{1-e_t}{e_t})$, and e_t are the error rates of the weak classifiers. In the case of a cascade, the confidence formula can no longer be treated as a distance from the decision boundary, which is now non-linear. However, it can be approximated by a sum of $q_j^{(k)}$ terms over all K cascade layers, taking the modified thresholds t_k in each layer into account:

$$q_j = \sum_{k=1}^{K} q_j^{(k)} = \sum_{k=1}^{K} \sum_{t=1}^{T_k} (\alpha_t h_t(\mathbf{x}_j) - t_k) \ . \tag{5}$$

In the case of a Hough detector, the confidence of each above-threshold circle picked from the accumulator array can simply be measured with the normalized number of votes cast for this circle. In general, confidence q_j can be expressed with any quantity that evaluates to a numerical, comparable value, and is indicative of the likelihood of the response.

Assuming that $f(\mathbf{x})$ is the underlying distribution of $\mathbf{x}$, the mean shift algorithm iteratively finds the stationary points of the estimated density via alternate computation of the mean-shift vector, and translation of the current kernel window by this vector, until convergence (for details, refer to [9]). Our modified mean-shift vector is made sensitive to the confidence of the input points in the following way:

$$\mathbf{m}_{h,G} = \frac{\sum_{j=1}^{n} \mathbf{x}_j q_j g \left\| \frac{\mathbf{x}-\mathbf{x}_j}{h} \right\|^2}{\sum_{j=1}^{n} q_j g \left\| \frac{\mathbf{x}-\mathbf{x}_j}{h} \right\|^2} - \mathbf{x} \ , \tag{6}$$

where $g(\cdot)$ is the underlying gradient density estimator and h is the bandwidth parameter determining the scale of the estimated density. Incorporating the confidence terms q_j in (6) is equivalent to amplifying the density gradients pointing towards the more reliably detected circle locations. The found modes of $\mathbf{x}$ correspond to the new road sign candidates which we need to track in the consecutive frames of the input video.

3 Tracking

To recognize traffic signs from a moving vehicle, it is crucial to have a view-independent object detector. Training such a detector directly exhibits serious difficulties as it requires feature descriptors to be both: highly discriminative and pose-invariant. Our method of solving such a detection problem follows a different strategy and has been shown successful in several studies, e.g. [18,20]. Instead of devising a pose-independent feature representation of the target, we learn an application-specific motion model from the random affine transformations applied to the full-face view of a detected sign. This model is learned via regression using the Lie algebra of the motion group, and encodes the correlations between a unique feature representation of a sign and the affine transformations it is subject to while being approached by a camera. In section 3.1 we provide the

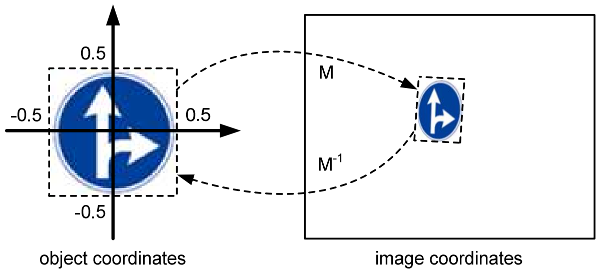


Fig. 3 Affine transformation matrix and its inverse.

theoretical foundations of our regression tracking algorithm. Section 3.2 describes a concrete realization of this method.

3.1 Tracking as a Regression Problem

Let $\mathbf{M}$ be an affine matrix that transforms a unit square at the origin in the object coordinates to the affine region enclosing the target object in the image coordinates:

$$\mathbf{M} = \begin{pmatrix} \mathbf{A} & \mathbf{t} \\ 0 & 1 \end{pmatrix} \quad , \tag{7}$$

where $\mathbf{A}$ is a 2×2 nonsingular matrix and $\mathbf{t} \in \mathbb{R}^2$. Let $\mathbf{M}^{-1}$ be an inverse transform, that maps the object region from image coordinates back to the object coordinates, as shown in Fig. 3. Our goal is to estimate the transformation matrix $\mathbf{M}_t$ at time t, given the observed images up to that point, $I_{0,\ldots,t}$, and the initial transformation $\mathbf{M}_0$. $\mathbf{M}_t$ is modeled recursively:

$$\mathbf{M}_t = \mathbf{M}_{t-1} \Delta \mathbf{M}_t \quad , \tag{8}$$

which means that it is sufficient to estimate only the increment $\Delta \mathbf{M}_t$ corresponding to the motion of the target from time $t-1$ to t in object coordinates. It is determined by the regression function:

$$\Delta \mathbf{M}_t = f\left(\mathbf{o}_t(\mathbf{M}_{t-1}^{-1})\right) \quad , \tag{9}$$

where $\mathbf{o}_t(\mathbf{M}_{t-1}^{-1})$ denotes an image descriptor applied to the previously observed image, after mapping it to the unit rectangle.

The regression function $f : \mathbb{R}^m \longmapsto A(2)$ is an affine matrix-valued function. To learn its parameters, it is necessary to know the initial pose of an object, $\mathbf{M}_0$, and the image I_0 at time t_0. Training examples are generated as pairs $(\mathbf{o}_0^i, \Delta \mathbf{M}_i)$, where $\Delta \mathbf{M}_i$ are random deformation matrices around identity and $\mathbf{o}_0^i = \mathbf{o}_0(\Delta \mathbf{M}_i^{-1} \mathbf{M}_0^{-1})$. The optimal parameters of f are derived on the grounds of the Lie group theory by minimizing the sum of the squared geodesic distances between the pairs of motion matrices: estimated $f(\mathbf{o}_0^i)$, and known $\Delta \mathbf{M}_i$. Details of this method can be found in [20].

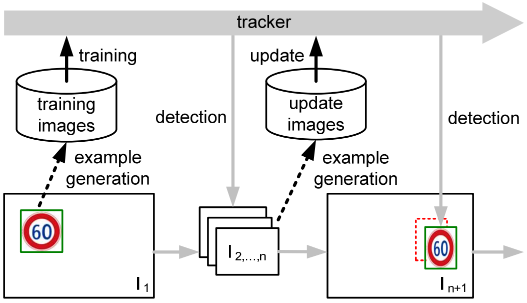


Fig. 4 Operation of a road sign tracker over time. The period between the initial candidate detection and the first tracker update is depicted.

3.2 Tracker Architecture

The regression tracker introduced in section 3.1 is utilized in our traffic sign recognition system as shown in Fig. 4. Once a candidate sign has been detected for the first time, a new tracker is initialized with the region corresponding to the bounding rectangle of the found circle instance, assuming no distortion [1]. At this point a small number of random deformations are generated from the observed image and used for instant training. A map of 6×6 regularly spaced 6-bin gradient orientation histograms is used as an object descriptor. The trained tracker is employed to detect the sign in n subsequent frames, each being used to generate and enqueue m new random deformations.

In a realistic traffic scenario the scene is often difficult and changes fast. Therefore, the accuracy of the tracker is likely to deteriorate very quickly as a result of the cumulated reconstruction errors caused by: 1) contaminating the training examples with the unwanted background fragments, and 2) changing appearance of the target. To deal with this problem, we update the tracking function after each n frames by re-training it on the collected portion of $n \cdot m$ random training transformations. The updated function f is trained in a similar way as is done after the initial sign detection, i.e. by minimizing the sum of the squared geodesic distances between the estimated and the known motion matrices, but another constraint is introduced on the difference between the current and the previous regression coefficients (refer to [20] for more details). The updated tracker is used to re-estimate the pose of the observed sign and the space is allocated for a new portion. Such a periodic update scheme allows us to recover from the misalignments likely to occur during the sign tracking.

[1] This assumption is valid as the road signs are detected for the first time at a considerable distance from the camera, where this distance if much greater than the distance of the sign from its optical axis.

Finally, the track is assumed to be lost when the sign either gets out of the field of view or when the normalized cross-correlation between its current warped image and the warped image recorded at the last update drops below a predefined threshold. The latter condition prevents the track from running out of control due to the cumulated errors of the tracker.

4 Recognition

Recognition of traffic signs is a hard multi-class problem with an additional difficulty caused by the fact of certain signs being very similar to one another, e.g. speed limits. The approach we have adopted in this work is centered around the concept of trainable similarity that can be inferred from the pairs of examples. Once the similarity between any two images has been estimated, any multi-class classification problem can be solved by comparing the similarities between the unknown example and each class's prototype. A tested example belongs to the class to which it is the most similar. For robust similarity assessment we use a novel variant of AdaBoost algorithm, called *SimBoost*. It is derived in section 4.1. In section 4.2 we outline how the classifier trained via SimBoost is used to recognize objects in image sequences.

4.1 SimBoost Algorithm

Formally, our classifier, $F(\mathbf{x})$, is designed to recognize only two classes: "same" and "different", and is trained using pairs of images, i.e. $\mathbf{x} = (i_1, i_2)$. The pairs representing the same class of sign are labeled $y = 1$ (positive), and the pairs representing two different classes are labeled $y = -1$ (negative). Real-valued discriminant function F is learned using a modified AdaBoost algorithm [5] which we call *SimBoost*. We define F as a sum of image features f_j:

$$F(i_1, i_2) = \sum_{j=1}^{N} f_j(i_1, i_2) \ . \tag{10}$$

Each feature evaluates to:

$$f_j(i_1, i_2) = \begin{cases} \alpha \text{ if } \ d(\phi_j(i_1), \phi_j(i_2)) < t_j \\ \beta \text{ otherwise} \end{cases} \ , \tag{11}$$

where ϕ_j is a filter defined over a chosen class of image descriptors, d is a generic distance metric that makes sense for such descriptors, and t_j is a feature threshold. In other words, each feature quantifies a local similarity between the input images and responds to this similarity depending on whether or not it is sufficient to consider the images as representing the same class.

Let $h_j(i_1, i_2) = d(\phi_j(i_1), \phi_j(i_2))$. Let us also denote by W_+^+ the total weight of these positive examples that are labeled positive by this weak classifier (true positives), and by W_+^- the total weight of those that are labeled negative (false negatives). By analogy, let W_-^- and W_-^+ be the total weight of true negatives and false positives respectively. In other words:

$$\begin{array}{ll} W_+^+ = \sum\limits_{\substack{k:y_k=1 \\ \wedge\, h(\mathbf{x}_k)<t}} w_i & W_+^- = \sum\limits_{\substack{k:y_k=1 \\ \wedge\, h(\mathbf{x}_k)\geq t}} w_i \\ W_-^+ = \sum\limits_{\substack{k:y_k=-1 \\ \wedge\, h(\mathbf{x}_k)<t}} w_i & W_-^- = \sum\limits_{\substack{k:y_k=-1 \\ \wedge\, h(\mathbf{x}_k)\geq t}} w_i \end{array} \ . \tag{12}$$

In each boosting round the filter ϕ_j and the threshold t_j are selected so as to minimize the weighted error of the training examples:

$$e_j = \sum_{\substack{k:y_k=1 \\ \wedge\, h(\mathbf{x}_k)\geq t}} w_i + \sum_{\substack{k:y_k=-1 \\ \wedge\, h(\mathbf{x}_k)<t}} w_i = W_+^- + W_-^+ \ . \tag{13}$$

Secondly, the optimal values of α and β are found based on the Schapire and Singer's criterion [6] of minimizing:

$$Z = \sum_{k=1}^{M} w_k e^{-y_k f(x_k)} \ , \tag{14}$$

where M is the total number of training examples. First, the sum is split as follows:

$$\begin{aligned} Z &= \sum_{k:y_k=1} w_k e^{-f(x_k)} + \sum_{k:y_k=-1} w_k e^{f(x_k)} = \\ &= \sum_{\substack{k:y_k=1 \\ \wedge\, h(\mathbf{x}_k)<t}} w_i e^{-\alpha} + \sum_{\substack{k:y_k=1 \\ \wedge\, h(\mathbf{x}_k)\geq t}} w_i e^{-\beta} + \\ &+ \sum_{\substack{k:y_k=-1 \\ \wedge\, h(\mathbf{x}_k)<t}} w_i e^{\alpha} + \sum_{\substack{k:y_k=-1 \\ \wedge\, h(\mathbf{x}_k)\geq t}} w_i e^{\beta} = \\ &= W_+^+ e^{-\alpha} + W_+^- e^{-\beta} + W_-^+ e^{\alpha} + W_-^- e^{\beta} \end{aligned} \ . \tag{15}$$

Taking partial derivatives of Z with respect to α and β and setting each to zero determines the optimal values of each parameter to be set in a given boosting round:

$$\alpha = \tfrac{1}{2} \log \left(\frac{W_+^+}{W_-^+} \right) \qquad \beta = \tfrac{1}{2} \log \left(\frac{W_+^-}{W_-^-} \right) \ . \tag{16}$$

A classifier trained using SimBoost algorithm yields a decision which is a linear combination the weak classifiers' responses:

$$l(i_1, i_2) = \text{sign} F(i_1, i_2) = \text{sign} \left(\sum_{j=1}^{N} f_t(i_1, i_2) \right) \ . \tag{17}$$

4.2 Temporal Classification

In order to be able to use the binary classifier discussed in section 4.1 for solving a multi-class problem, the classifier's response must be made soft. This can be done in a straightforward way by omitting the sign in the right-hand-side expression of equation (17), i.e. considering the bare value of function F. This value can be treated as a degree of similarity of the two input images. Let $p_1, \ldots, p_K$ be the prototype images of K targeted classes. If one of the images passed on input of our road sign classifier, say i_1, is a prototype of known class k ($i_1 = p_k$), the classifier assigns such a label to the other, unknown image, that satisfies:

$$l(i) = \arg\max_k F(p_k, i) \ . \tag{18}$$

In other words, $l(i)$ is determined from the prototype to which the tested image is the most similar.

To classify a sequence of images, $i_{1,\ldots,T}$, the maximum rule in (18) is applied to the sum of $F(p_k, i_t)$ terms over all images i_t, $t = 1, \ldots, T$. Each i_t denotes a warped image of a sign obtained by applying the inverse of the transformation matrix $\mathbf{M}_t$ to the frame at time t. Additionally, the contribution of the most recent observations is emphasized to reflect the fact that the image of a sign becomes generally clearer as the vehicle approaches the target. The ultimate classifier's decision at time T is determined from:

$$l(i_{1,\ldots,T}) = \arg\max_{k=1} \sum_{t=1}^{T} q(t) F(p_k, i_t) \ , \tag{19}$$

where $q(t) = b^{T-t}$, $b \in (0, 1]$, is a relevance of the observation i_t.

5 Experimental Results

In this section we present the experimental evaluation of the road sign detection, tracking and recognition algorithms proposed in our study. Each of the three core modules of the intended system are first tested as standalone components. The two considered detection methods and the detection refinement algorithm are evaluated on the static road sign images in section 5.1. The better-performing refined detector is chosen to be incorporated into the prototype system and the justification of this choice is provided. In section 5.2 we concentrate on the proposed regression tracker and measure its capability of modeling the affine distortions that the traffic signs are subject to while being approached by a car-mounted camera. A small set of synthetic image sequences are generated to facilitate this experiment. Performance of a classifier trained via SimBoost is measured in section 5.3, again using the static images of traffic signs. The feature representation guaranteeing the most reliable assessment of the similarity between the two images is determined based on the obtained experimental results. Finally, in section 6 we assemble the entire sign detection, tracking, and recognition system and test it on a number of real-life video sequences captured from a moving vehicle. Computational aspects are also discussed.

5.1 Evaluation of Road Sign Detectors

In order to make a right choice of a sign detector to be used in our system, we first tested the Haar cascade and the Hough circle detector discussed in 2.2 without considering the video context. The test image sequences we possess were acquired in the urban areas in Japan, where most of the traffic signs captured were circular. We have therefore evaluated the capacity of both abovementioned techniques for detecting 14 types of circular signs. We ran each detector in the small regions of the input images around the known ground truth sign locations. Specifically, the size of each analysis region was set to 1.5 diameter of a sign located in the region center. A few example images used are shown in Fig. 5. The experiment was performed using a total of 8175 challenging images and repeated for 1) varying threshold of the classifier in the last cascade layer, and 2) varying threshold in the Hough vote space. To increase the discriminative power of both detectors, we transformed each input image using the color filters (3). In the case of Hough transform, the color-specific edge maps were computed and the HT was run on each of them, pixel by pixel. When evaluating the Haar cascade, filters (3), along with a gray-scale transformation, were used to parametrize the Haar wavelets, as proposed by Bahlmann et al. [13]. In order to train the classifier, a set of another 4218 images was first clustered to reduce the intra-class variability. Then, a separate cascade was trained for each cluster. The test images were scanned by the cascaded classifier with a 2-pixel step to reduce computation.

For each image a ground truth center position and the radius of a sign was given, (x_c, y_c, r). Quantities measured were: 1) the mean number of candidates per image detected, 2) mean distance between a detected circle and the ground truth circle expressed with a Euclidean metric over the abovementioned triples, and 3) miss rate, i.e. the percentage of images where no sign was detected. Relationship between the miss rate and the two other quantities are illustrated in Fig. 6. The experiment showed that the Haar cascade is a slightly more

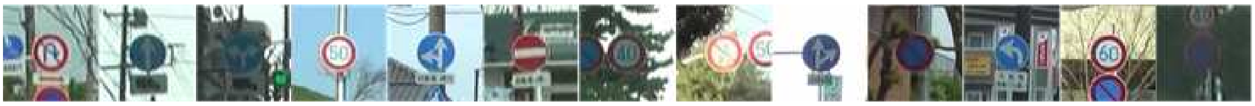


Fig. 5 Example images used in the experimental evaluation of the traffic sign detectors.

accurate road sign detector than the circular Hough transform in the entire range of practically useful operating points. However, this advantage was achieved at the cost of higher sensitivity, and hence more computation. While the average processing time of a single image was approximately 10ms for a Hough detector, this time increased to over 20ms for a Haar cascade [2]. The difference in the accuracy of both detectors can partly be attributed to the nature of the voting in Hough space. As it is generally unknown whether the road sign is darker or lighter than the background, the votes coming from the contour pixels are cast on both sides of the circle. At times, the number of votes cumulated outside the true signs can be sufficiently high to produce false candidates that are adjacent to the true one. Besides, the circular Hough transform is relatively insensitive to scale when the input image contains thick edges. In that case it often yields above-threshold responses for a whole range of circle radii. Regardless of the results of this comparison, both techniques appeared to be impractical when used alone, i.e. without an appropriate postprocessing of the detector responses.

We have repeated the above experiment, but applying the proposed *Confidence-Weighted Mean Shift* refinement algorithm to the output generated by each detector. Obtained results are shown in Fig. 7. It can easily be noticed that in the case of the circular Hough detector, the mean number of detected candidates per image roughly corresponds to the percentage of the images where any candidate was detected. This implies that the proposed detection refinement scheme most likely collapses the multiple positive responses of the detector into a single candidate, which is an intended outcome. The same effect is achieved by a refined cascade of classifiers only for a relatively high threshold of the last layer, when the miss rate of the detector is considerable. The mean error of both detectors appears to be lower with the refinement procedure enabled, with the Haar cascade being by 25-50% more accurate. Interestingly, the improvement in the accuracy of the Hough detector is dramatic, while the refined Haar cascade merely eliminates redundancies, but does not reduce the error significantly. Moreover, the difference between the average processing time of a single image became even more apparent as it stayed at the 10ms level for the Hough detector, but increased to 50-100ms for a cascade. The latter observation suggests that the cascade of boosted classifiers does not seem to be an adequate method for detecting many types of signs at once. Decomposition of the problem via clustering of the training data makes this method more discriminative, but also unnecessarily expensive. No or little advantage over the much simpler Hough-based detector, which requires no training, makes us adopt the latter method for further experiments presented in this paper. Figure 8 illustrates example output of the Hough circle detector before and after applying the *Confidence-Weighted Mean Shift* refinement algorithm.

5.2 Evaluation of the Regressor Tracker

We have conducted a separate experiment aimed at evaluating the ability of our road sign tracker to retrieve the full-face view of a sign under affine transformations. This experiment was done in the following way. Five synthetic image sequences were prepared using the OpenGL framework [22]. In each sequence a template image of one sign is shown in an empty 3D scene. The consecutive images depict the sign getting closer to the camera and hence increasingly distorted. This simulates a realistic scenario of a car approaching a road sign mounted on the side of the road or above the road lane. The rendered scenes were deliberately constructed without any background and with constant illumination to minimize the effects of possible contamination of the image regions enclosing the target object and to ensure its consistent appearance. For each image sequence the refined circular Hough detector was set to capture the circle instances of radius 12-24 pixels. The tracker was triggered at the time of initial detection of a sign by the HT and updated every 15 frames. Upon the initial detection, the nearly undistorted image of a sign in gray-scale was recorded to serve as a reference image.

Robustness of the on-line learned tracking function to the affine distortions was measured by recording a normalized cross-correlation (NCC) between the reconstructed full-face view of a sign in each frame and the reference image. The changes of this correlation over time for all five sequences are shown in Fig. 9 [3]. In each

[2] For a pixel-by-pixel scanning, the cascaded classifier was approximately 7 times slower than the Hough detector.

[3] The sequences used in this experiment are provided in the supplementary material accompanying this paper.

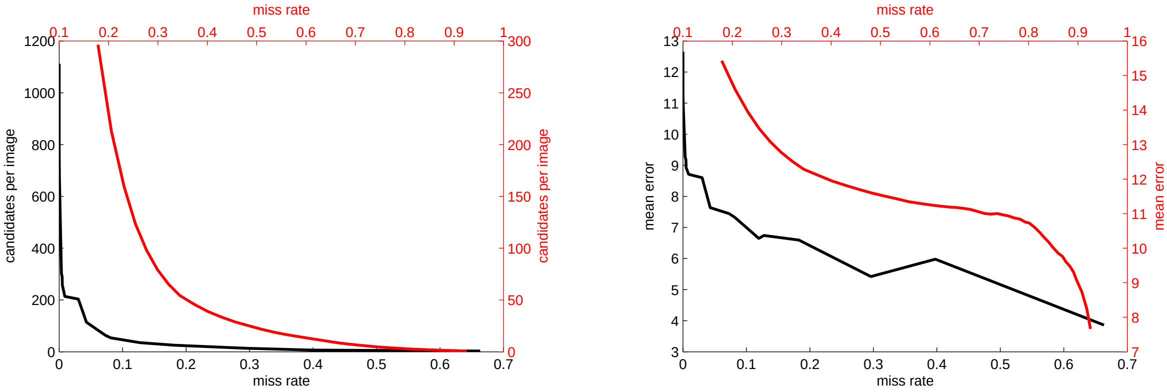


Fig. 6 Relationship between the mean number of candidates per image detected and the miss rate (left), and between the mean distance of the detected candidates from the ground truth circles and the miss rate (right). Black lines and axes correspond to the Haar cascade and the red lines and axes correspond to the circular Hough detector. This figure is best viewed in color.

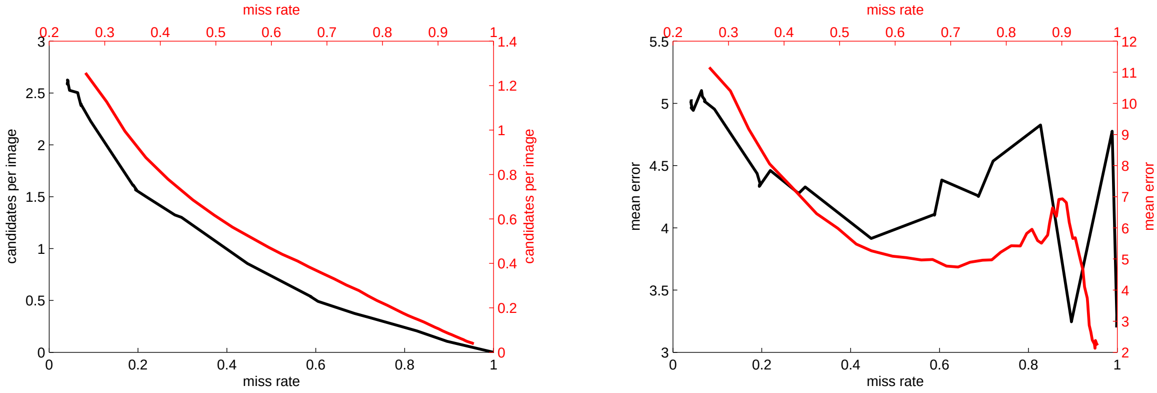


Fig. 7 Relationship between the mean number of candidates per image detected and the miss rate (left), and between the mean distance of the detected candidates from the ground truth circles and the miss rate (right). Black lines and axes correspond to the refined Haar cascade and the red lines and axes correspond to the refined circular Hough detector. This figure is best viewed in color.

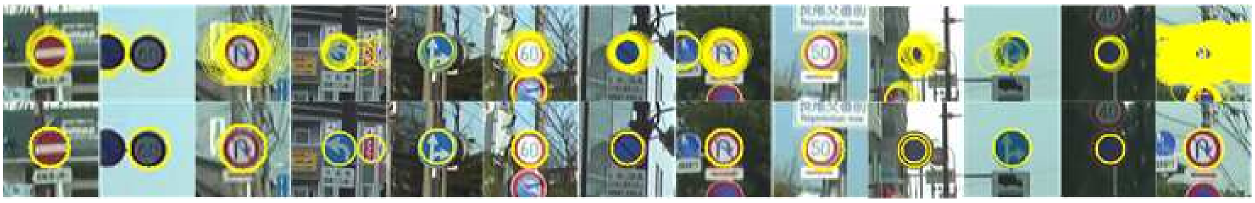


Fig. 8 Output of the Hough circle detector before (upper row) and after (lower row) applying the refinement procedure. The transparency of the detected circles in the upper row images correspond to their confidence expressed with the scaled number of votes picked from the Hough voting space.

plot the behavior of NCC for a 6D regression function encoding all six 2D affine transform parameters is compared to the behavior of NCC observed using three other trackers. These are: 1) a 4D regression function encoding only two rotation-shift parameters and both translation parameters, 2) a 3D regression function encoding only one isotropic scaling-rotation parameter and both translation parameters, and 3) a simple tracker which makes individual circle detections in each frame, but uses a Kalman filter (KF) [1] to predict the position and scale of a sign. During the on-line training of the regression trackers, all non-translation parameters were randomly generated within the range $[-0.2, 0.2]$ and the translation parameters were randomly generated within the range $[-0.4, 0.4]$.

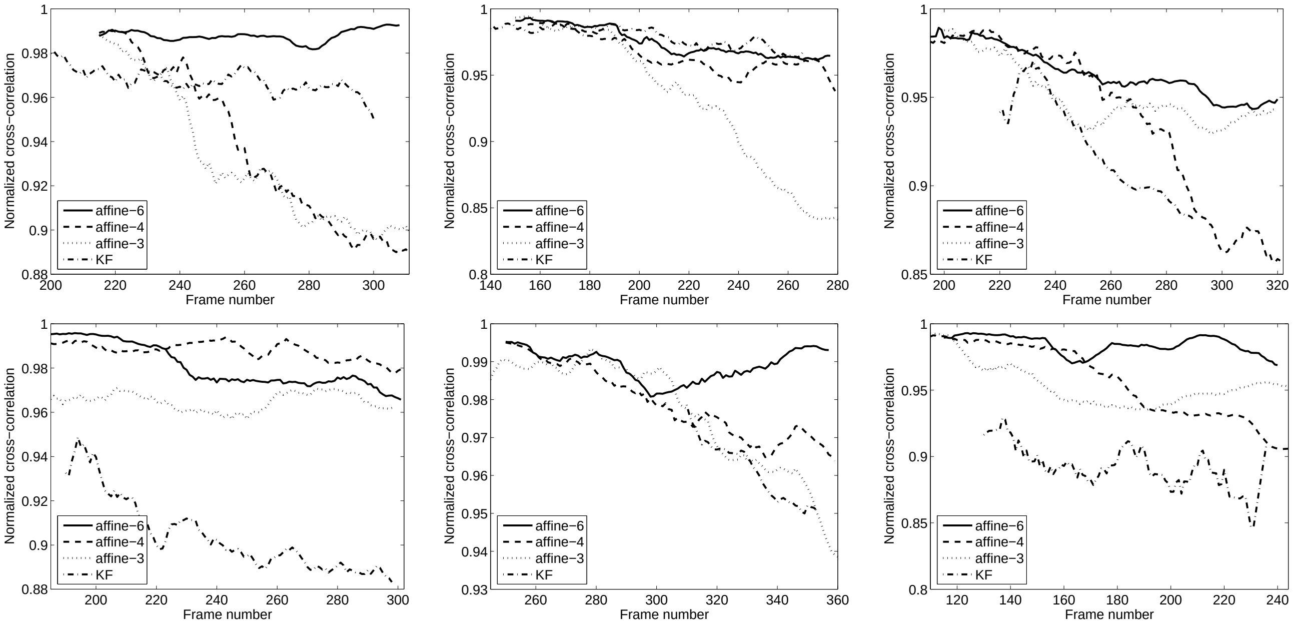


Fig. 9 Normalized cross-correlation (NCC) between the reference image recorded at the time of initial sign detection and the reconstructed full-face view of a sign in each frame of the input sequences. Each sequence was generated in a synthetic empty 3D scene and simulates what is typically observed from a vehicle approach a traffic sign.

Based on the results of the above experiment, we conclude that learning of the motion model based on the Lie algebra enables construction of a robust object tracker which is invariant to the affine transformations. In Fig. 9 the 6D affine tracker outperforms the two other regression trackers and the KF-based tracker which do not model the full structure of the motion. The correlation between the original frontal view of a sign and a view inferred from the current transformation parameter estimates remains high for the entire duration of the sequences. In the case of the 4D and 3D affine trackers, as well as the KF-based tracker, this correlation drops more quickly, particularly in the second part of each sequence. In addition, the behavior of the KF-based tracker is less stable, as no temporal dependency between the consecutive frame observations is modeled. In other words, as long as the sign remains relatively unaffected by the affine distortion, all methods provide a relatively accurate track of the target. However, when the sign gets closer to the camera and thus becomes more substantially distorted in the image plane, only the fully-affine regression tracker remains able to restore the full-face view of the target with low error. From the point of view of the entire system this is a particularly useful property because the most informative frames of the input video, when the appearance of a sign is the most unambiguous, can be efficiently used for recognition.

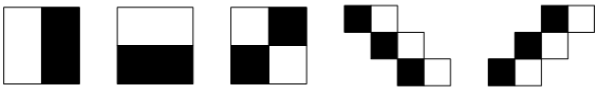


Fig. 10 Haar wavelet features used in the experimental evaluation of the traffic sign classifier trained using the SimBoost algorithm.

5.3 Evaluation of SimBoost

Performance of the road sign classifier trained with the SimBoost algorithm introduced in section 4.1 has been estimated using the similar dataset as the one used in section 5.1. 7757 static images of 14 circular Japanese road signs were cropped from the test video sequences such that each sign filled the entire image, and used to train a 100-feature classifier. Another 8434 images were used for testing. The quality and the illumination in all images varied significantly. When constructing the test input pairs, the prototype images of each class were chosen randomly out of all images available for this class. Exploiting flexibility of the distance measure in (11), three different image descriptors and the associated distance metrics were used within the SimBoost framework to populate the pool of input features. They are listed in table 1.

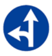
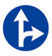
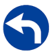
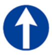
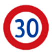
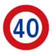
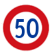
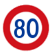
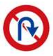
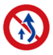
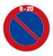
Results of the experiment are shown in the confusion matrices in Fig. 11. As seen, the histograms of oriented gradients and the color-parametrized Haar wavelet filters are the most useful image descriptors for classification of the traffic signs. Interestingly, both types of

Table 1 Image descriptors and the associated distance metrics used in the experimental evaluation of the traffic sign classifier trained using the SimBoost algorithm.

Image feature	Description	Associated distance metric
Color-parametrized Haar wavelet [11]	Rectangular filters shown in Fig. 10, parametrized with color, as described in section 5.1. Only the filters of scale $w, h = \{4, 8\}$px, shifted by $\frac{1}{4}w$, $\frac{1}{4}h$ along each dimension were used, where by scale we refer to the width and height of a single rectangular block of a filter.	$d(\phi_j(i_1), \phi_j(i_2)) = \lvert v_1 - v_2)\rvert$, where $v_1 = \phi_j(i_1)$, $v_2 = \phi_j(i_2)$
Histogram of oriented gradients (HOG) [14]	6-bin gradient orientation histograms computed at all possible image regions satisfying: $w, h = \{10, 15, 20\}$px, $d_x = \frac{1}{2}w$, $d_y = \frac{1}{2}h$, where w, h are the width and height of the analysis region, and d_x, d_y are the shifts along each axis.	$d(\phi_j(i_1), \phi_j(i_2)) = \sqrt{\sum_{k=1}^{n}(v_{1,k} - v_{2,k})^2}$, where $\mathbf{v}_1 = \phi_j(i_1)$, $\mathbf{v}_2 = \phi_j(i_2)$, $\mathbf{v}_1, \mathbf{v}_2 \in \mathbb{R}^n$, and n is the number of histogram bins
Region covariance [17]	4×4 covariance matrices encoding x and y coordinates and the first-order image derivatives. Only the regions of scale $w, h = \{10, 15, 20\}$px, shifted by $\frac{1}{2}w$, $\frac{1}{2}h$ along each dimension were considered.	$d(\phi_j(i_1), \phi_j(i_2)) = \sqrt{\sum_{k=1}^{n} \ln^2 \lambda_k(\mathbf{C}_1, \mathbf{C}_2)}$, where $\{\lambda_k(\mathbf{C}_1, \mathbf{C}_2)\}_{k=1,\ldots,n}$ are the generalized eigenvalues of $\mathbf{C_1}$ and $\mathbf{C_2}$, computed from $\lambda_k \mathbf{C}_1 \mathbf{x}_k = \mathbf{C}_2 \mathbf{x}_k$

features carry non-overlapping pieces of discriminative information, and thanks to the flexibility of the distance formulation in SimBoost, these cues can efficiently be combined. The classifier trained with both types of descriptors available in the input feature pool achieved a superior correct classification rate of nearly 76%. In Fig. 12 we have visualized the first 10 features selected by SimBoost for this best-performing classifier.

Table 2 Classification rates obtained in the dynamic experiment. The numbers of correctly classified signs of each class are given against the total numbers of such signs detected in the input sequences.

					
7/7	4/5	1/1	4/5	1/2	10/10
					
6/9	2/2	3/3	9/10	26/31	

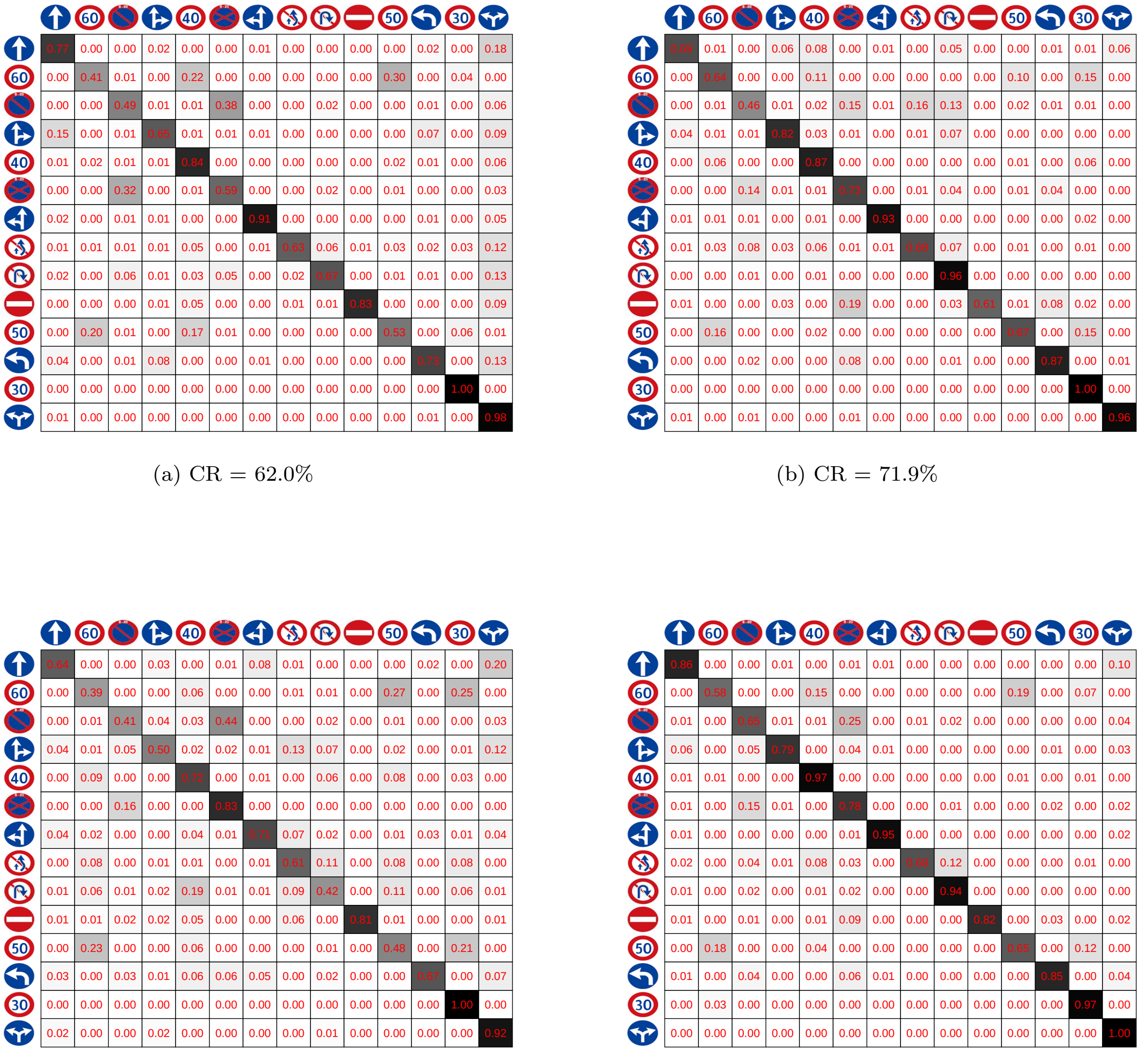
6 Performance of the Entire System

To evaluate the proposed traffic sign detection, tracking and recognition algorithms altogether, we built a prototype system incorporating all three components with their optimal settings. A demo application was implemented in C++ and part of the computationally demanding image processing operations were handled using the OpenCV library [23]. The system allows manual modification of several parameters, among others the frequency of detection [4], the scale of the signs to be detected, and the frequency of the tracker update.

We have obtained a number of realistic video sequences to test an overall performance of the implemented system. Each sequence was captured with a front-looking wide-angle camera mounted on board of a vehicle, in various, usually crowded street scenes in Japan. The illumination of the scene is roughly constant in all test videos. In system runtime, a 720×540 pixels portion of the scene was cropped from the upper-central region of each frame of the input video, and further downscaled by 50%. The range of radii of the circles captured by the detector was set to 12-24 pixels and the tracker updated itself every $n = 15$ frames, generating $m = 6$ new random affine transformations in each frame. During the on-line training of the regression tracker, the affine matrix parameters were generated randomly within the same ranges as defined in the experiment from section 5.2, i.e. $[-0.2, 0.2]$ for all non-translation parameters, and $[-0.4, 0.4]$ for both translation parameters. Table 2 illustrates the numbers of traffic signs of each class that occurred in the videos and were detected, together with the numbers of these signs that were correctly classified.

As seen, an overall error rate of the classifier did not exceed 15%. Misclassifications were mainly caused by the motion blur erasing the relevant image gradients, and by the cumulated reconstruction errors of the tracker. These errors can partly be attributed to the background fragments which contaminate the corners of the image regions enclosing the target circular signs. Regarding the other system components, the refined Hough circle detector appeared to be relatively accurate and resistant to clutter. Overall, it missed 14 true signs, mostly due to the insufficient figure-background contrast, and yielded less than ten false sign candidates. Figure 13 shows several examples of signs our detector was not able to detect. Finally, the tracker demonstrated its ability to rapidly correct small affine sign

[4] Exploration of the entire scene in search of the new road sign candidates in each frame of the input video is unnecessary and can be performed every k frames without the increase in the miss rate.



	↑	60	⦸	↱	40	⊗	↰	🚫	⊘	⛔	50	↶	30	⇆
↑	0.77	0.00	0.00	0.02	0.00	0.00	0.01	0.00	0.00	0.00	0.00	0.02	0.00	0.18
60	0.00	0.41	0.01	0.00	0.22	0.00	0.00	0.00	0.00	0.00	0.30	0.00	0.04	0.00
⦸	0.00	0.00	0.49	0.01	0.01	0.38	0.00	0.00	0.02	0.00	0.00	0.01	0.00	0.06
↱	0.15	0.00	0.01	0.65	0.01	0.01	0.01	0.00	0.00	0.00	0.00	0.07	0.00	0.09
40	0.01	0.02	0.01	0.01	0.84	0.00	0.00	0.00	0.00	0.00	0.02	0.01	0.00	0.06
⊗	0.00	0.00	0.32	0.00	0.01	0.59	0.00	0.00	0.02	0.00	0.01	0.00	0.00	0.03
↰	0.02	0.00	0.01	0.01	0.00	0.00	0.91	0.00	0.00	0.00	0.00	0.01	0.00	0.05
🚫	0.01	0.01	0.01	0.01	0.05	0.00	0.01	0.63	0.06	0.01	0.03	0.02	0.03	0.12
⊘	0.02	0.00	0.06	0.01	0.03	0.05	0.00	0.02	0.67	0.00	0.01	0.01	0.00	0.13
⛔	0.00	0.00	0.00	0.01	0.05	0.00	0.00	0.01	0.01	0.83	0.00	0.00	0.00	0.09
50	0.00	0.20	0.01	0.00	0.17	0.01	0.00	0.00	0.00	0.00	0.53	0.00	0.06	0.01
↶	0.04	0.00	0.01	0.08	0.00	0.00	0.01	0.00	0.00	0.00	0.00	0.73	0.00	0.13
30	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	1.00	0.00
⇆	0.01	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.01	0.00	0.98

(a) CR = 62.0%

	↑	60	⦸	↱	40	⊗	↰	🚫	⊘	⛔	50	↶	30	⇆
↑	0.69	0.01	0.00	0.06	0.08	0.00	0.01	0.00	0.05	0.00	0.00	0.01	0.01	0.06
60	0.00	0.64	0.00	0.00	0.11	0.00	0.00	0.00	0.00	0.00	0.10	0.00	0.15	0.00
⦸	0.00	0.01	0.46	0.01	0.02	0.15	0.01	0.16	0.13	0.00	0.02	0.01	0.01	0.00
↱	0.04	0.01	0.01	0.82	0.03	0.01	0.00	0.01	0.07	0.00	0.00	0.00	0.00	0.00
40	0.00	0.06	0.00	0.00	0.87	0.00	0.00	0.00	0.00	0.00	0.01	0.00	0.06	0.00
⊗	0.00	0.00	0.14	0.01	0.01	0.73	0.00	0.01	0.04	0.00	0.01	0.04	0.00	0.00
↰	0.01	0.01	0.01	0.00	0.01	0.00	0.93	0.00	0.00	0.00	0.00	0.00	0.02	0.00
🚫	0.01	0.03	0.08	0.03	0.06	0.01	0.01	0.68	0.07	0.00	0.01	0.00	0.01	0.00
⊘	0.00	0.00	0.01	0.00	0.01	0.00	0.00	0.00	0.96	0.00	0.00	0.00	0.00	0.00
⛔	0.01	0.00	0.00	0.03	0.00	0.19	0.00	0.00	0.03	0.61	0.01	0.08	0.02	0.00
50	0.00	0.16	0.00	0.00	0.02	0.00	0.00	0.00	0.00	0.00	0.67	0.00	0.15	0.00
↶	0.00	0.00	0.02	0.00	0.00	0.08	0.00	0.00	0.01	0.00	0.00	0.87	0.00	0.01
30	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	1.00	0.00
⇆	0.01	0.00	0.01	0.00	0.00	0.01	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.96

(b) CR = 71.9%

	↑	60	⦸	↱	40	⊗	↰	🚫	⊘	⛔	50	↶	30	⇆
↑	0.64	0.00	0.00	0.03	0.00	0.01	0.08	0.01	0.00	0.00	0.00	0.02	0.00	0.20
60	0.00	0.39	0.00	0.00	0.06	0.00	0.00	0.01	0.01	0.00	0.27	0.00	0.25	0.00
⦸	0.00	0.01	0.41	0.04	0.03	0.44	0.00	0.00	0.02	0.00	0.01	0.00	0.00	0.03
↱	0.04	0.01	0.05	0.50	0.02	0.02	0.01	0.13	0.07	0.00	0.02	0.00	0.01	0.12
40	0.00	0.09	0.00	0.00	0.72	0.00	0.01	0.00	0.06	0.00	0.08	0.00	0.03	0.00
⊗	0.00	0.00	0.16	0.00	0.00	0.83	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
↰	0.04	0.02	0.00	0.00	0.04	0.01	0.71	0.07	0.02	0.00	0.01	0.03	0.01	0.04
🚫	0.00	0.08	0.00	0.01	0.01	0.00	0.01	0.61	0.11	0.00	0.08	0.00	0.08	0.00
⊘	0.01	0.06	0.01	0.02	0.19	0.01	0.01	0.09	0.42	0.00	0.11	0.00	0.06	0.01
⛔	0.01	0.01	0.02	0.02	0.05	0.00	0.00	0.06	0.00	0.81	0.01	0.00	0.00	0.01
50	0.00	0.23	0.00	0.00	0.06	0.00	0.00	0.00	0.01	0.00	0.48	0.00	0.21	0.00
↶	0.03	0.00	0.03	0.01	0.06	0.06	0.05	0.00	0.02	0.00	0.00	0.67	0.00	0.07
30	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	1.00	0.00
⇆	0.02	0.00	0.00	0.02	0.00	0.00	0.00	0.00	0.01	0.00	0.00	0.00	0.00	0.92

(c) CR = 54.1%

	↑	60	⦸	↱	40	⊗	↰	🚫	⊘	⛔	50	↶	30	⇆
↑	0.86	0.00	0.00	0.01	0.00	0.01	0.01	0.00	0.00	0.00	0.00	0.00	0.00	0.10
60	0.00	0.58	0.00	0.00	0.15	0.00	0.00	0.00	0.00	0.00	0.19	0.00	0.07	0.00
⦸	0.01	0.00	0.65	0.01	0.01	0.25	0.00	0.01	0.02	0.00	0.00	0.00	0.00	0.04
↱	0.06	0.00	0.05	0.79	0.00	0.04	0.01	0.00	0.00	0.00	0.00	0.01	0.00	0.03
40	0.01	0.01	0.00	0.00	0.97	0.00	0.00	0.00	0.00	0.00	0.01	0.00	0.01	0.00
⊗	0.01	0.00	0.15	0.01	0.00	0.78	0.00	0.00	0.01	0.00	0.00	0.02	0.00	0.02
↰	0.01	0.00	0.00	0.00	0.00	0.01	0.95	0.00	0.00	0.00	0.00	0.00	0.00	0.02
🚫	0.02	0.00	0.04	0.01	0.08	0.03	0.00	0.68	0.12	0.00	0.00	0.00	0.01	0.00
⊘	0.01	0.00	0.02	0.00	0.01	0.02	0.00	0.00	0.94	0.00	0.00	0.00	0.00	0.00
⛔	0.01	0.00	0.01	0.00	0.01	0.09	0.00	0.00	0.00	0.82	0.00	0.03	0.00	0.02
50	0.00	0.18	0.00	0.00	0.04	0.00	0.00	0.00	0.00	0.00	0.65	0.00	0.12	0.00
↶	0.01	0.00	0.04	0.00	0.00	0.06	0.01	0.00	0.00	0.00	0.00	0.85	0.00	0.04
30	0.00	0.03	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.97	0.00
⇆	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	1.00

(d) CR = 75.8%

Fig. 11 Classification accuracy of a 100-feature classifier trained using the SimBoost algorithm and different image descriptors: (a) color-parametrized Haar wavelets [13], (b) histograms of oriented gradients (HOG), (c) 4 × 4 covariance matrices encoding x and y coordinates and the first-order image derivatives [17], (d) Haar and HOG features jointly.

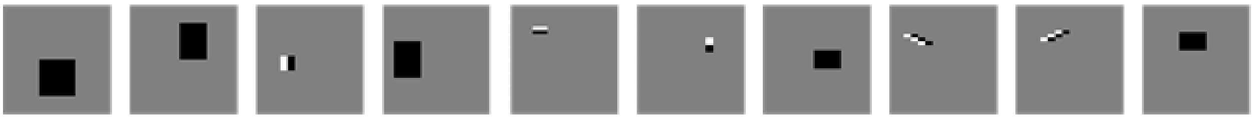


Fig. 12 10 best features selected by the SimBoost algorithm while training the 14-class road sign classifier using jointly the color-parametrized Haar wavelet filters and the histograms of oriented gradients. Both kinds of image descriptors are present.

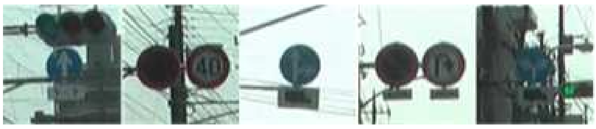


Fig. 13 Examples of road signs the refined Hough detector could not capture.

distortions, which enabled real-time system operation. Example videos demonstrating this ability are available at: `http://people.brunel.ac.uk/~cspgaar/MVA2009/`.

7 Conclusions

In this study we have presented a comprehensive approach to detection, tracking and recognition of traffic signs from a moving vehicle. Our system is comprised of three components. The detector utilizes a state-of-the-art object detection technique, but features a *Confidence-Weighted Mean Shift* mode-finding algorithm to improve its accuracy and cope with multiple redundant hypotheses in the detector's response space. The main contribution of our work are the novel tracking and recognition algorithms. The proposed tracker models the motion of the target through an instance-specific tracking function. It encodes correlations between the unique feature representation of a candidate sign and the affine distortions it is subject to while being approached by a camera. It is shown that based on the Lie group theory such a tracking function can be learned and updated instantly from random transformations applied to the image of the target in known pose. A detected and tracked sign candidate is classified by maximizing its similarity to the class's prototype image. This similarity is estimated by a linear combination of local image descriptor differences and is learned from image pairs using a novel variant of AdaBoost algorithm, called *SimBoost*.

The proposed algorithms have been evaluated in a number of experiments involving static road sign images, synthetic image sequences, and real-life video captured from a car-mounted camera. These experiments were aimed at: 1) evaluation of the detection refinement algorithm with two different object detection techniques and determining the most suitable refined detector, 2) demonstrating the ability of the tracker to model the affine motion of the signs and reconstruct their frontal views under significant deformations, and 3) estimating the error rate of a classifier trained with different low-level image descriptors using the SimBoost algorithm so as to determine the most suitable image representation. The overall performance of the system was estimated based on the prototype implementation we built in C++. The obtained results prove the efficiency of the proposed algorithms and confirm that we have chosen the right direction to tackle the challenging traffic sign recognition problem.

References

1. Kalman, R. E.: "A New Approach to Linear Filtering and Prediction Problems", *Transactions of the ASME - Journal of Basic Engineering*, vol.82, Series D, pp.35-45 (1960)
2. Duda, R.O. Hart, P.E.: "Use of the Hough transformation to detect lines and curves in pictures", *Communications of the ACM*, vol.15, no.1, pp.11-15 (1972)
3. Piccioli, G. De Micheli, E. Parodi, P. Campani, M.: "A robust method for road sign detection and recognition", *Image and Vision Computing*, vol.14, no.3, pp.209-223 (1996)
4. de la Escalera, A. and Moreno, L. E. and Salichs, M. A. and Armingol, J. M.: "Road traffic sign detection and classification", *IEEE Transactions on Industrial Electronics*, vol.44, no.6, pp.848-859, 1997.
5. Freund, Y. Schapire, R.E.: "A short introduction to boosting", *Journal of Japanese Society for Artificial Intelligence*, vol.14, no.5, pp.771-780 (1999)
6. Schapire, R. E. Singer, Y.: "Improved boosting algorithms using confidence-rated predictions", *Machine Learning*, pp.80-91 (1999)
7. Miura, J. Kanda, T. Shirai, Y.: "An active vision system for real-time traffic sign recognition", In *Proc. of the IEEE Conference on Intelligent Transportation Systems*, pp.52-57 (2000)
8. Paclík, P. Novovicova, J. Pudil, P. Somol, P.: "Road Sign Classification using the Laplace Kernel Classifer", *Pattern Recognition Letters*, vol.21, no.13-14, pp.1165-1173 (2000)
9. Comaniciu, D. Meer, P.: "Mean shift: a robust approach towards feature space analysis", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol.24, no.5, pp.603-619 (2002)
10. Fang, C-Y. Chen, S-W. Fuh, C-S.: "Road-Sign Detection and Tracking", *IEEE Transactions on Vehicular Technology*, vol.52, no.5, pp.1329-1341 (2003)
11. Viola, P. Jones, M.: "Robust Real-time Object Detection", *International Journal of Computer Vision*, vol.57, no.2, pp.137-154 (2004)
12. Loy, G. Barnes, N. Shaw, D. Robles-Kelly, A.: "Regular Polygon Detection", In *Proc. of the 10th IEEE International Conference on Computer Vision*, vol.1, pp.778-785 (2005)
13. Bahlmann, C. Zhu, Y. Ramesh, V. Pellkofer, M. Koehler, T.: "A System for Traffic Sign Detection, Tracking and Recognition Using Color, Shape, and Motion Information", In *Proc. of the IEEE Intelligent Vehicles Symposium*, pp.255-260 (2005)
14. Dalal, N. Triggs, B.: "Histograms of Oriented Gradients for Human Detection", In *Proc. of the 2005 IEEE International Conference on Computer Vision and Pattern Recognition*, vol.1, pp.886–893 (2005)
15. Gao, X.W. Podladchikova, L. Shaposhnikov, Hong, K. Shevtsova, N.: "Recognition of traffic signs based on their colour and shape features extracted using human vision models", *Journal of Visual Communication and Image Representation*, vol.17, no.4, pp.675-685 (2006)
16. Paclík, P. Novovicová, J. Duin, R. P. W.: "Building Road-Sign Classifiers Using a Trainable Similarity Measure", *IEEE Transactions on Intelligent Transportation Systems*, vol.7, no.3, pp.309-321 (2006)
17. Tuzel, O. Porikli, F. Meer, P.: "Region covariance: A fast descriptor for detection and classification", In *Proc. of the 9th European Conference on Computer Vision*, pp.589-600, 2006.

18. Bayro-Corrochano, E. Ortegón-Aguilar, J.: "Lie algebra approach for tracking and 3D motion estimation using monocular vision", *Image and Vision Computing*, vol.25, no.6, pp.907-921 (2007)
19. Ruta, A. Li, Y. Liu, X.: "Towards Real-time Traffic Sign Recognition by Class-specific Discriminative Features", In *Proc. of the 18th British Machine Vision Conference*, vol.1, pp.399-408 (2007)
20. Tuzel, O. Porikli, F. Meer, P.: "Learning on Lie Groups for Invariant Detection and Tracking", In *Proc. of the 2008 IEEE Conference on Computer Vision and Pattern Recognition*, pp.1-8 (2008)
21. Nguwi, Y-Y. Kouzani, A.Z.: "Detection and classification of road signs in natural environments", *Neural Computing and Applications*, vol.17, no.3, pp.265-289 (2008)
22. Open Graphics Library, `www.opengl.org/`
23. Open Computer Vision Library , `http://sourceforge.net/projects/opencvlibrary/`