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

Multi-view traffic sign detection, recognition, and 3D localisation

Radu Timofte · Karel Zimmermann · Luc Van Gool

Received: 29 October 2010 / Revised: 31 July 2011 / Accepted: 15 November 2011

Abstract Several applications require information about street furniture. Part of the task is to survey all traffic signs. This has to be done for millions of km of road, and the exercise needs to be repeated every so often. We used a van with 8 roof-mounted cameras to drive through the streets and took images every meter. The paper proposes a pipeline for the efficient detection and recognition of traffic signs from such images. The task is challenging, as illumination conditions change regularly, occlusions are frequent, sign positions and orientations vary substantially, and the actual signs are far less similar among equal types than one might expect. We combine 2D and 3D techniques to improve results beyond the state-of-the-art, which is still very much preoccupied with single view analysis. For the initial detection in single frames, we use a set of colour- and shape-based criteria. They yield a set of candidate sign patterns. The selection of such candidates allows for a significant speed up over a sliding window approach while keeping similar performance. A speedup is also achieved through a proposed efficient bounded evaluation of AdaBoost detectors. The 2D detections in multiple views are subsequently combined to generate 3D hypotheses. A Minimum Description Length formulation yields the set of 3D traffic signs that best explains the 2D detections. The paper comes with a publicly available database, with more than 13 000 traffic signs annotations.

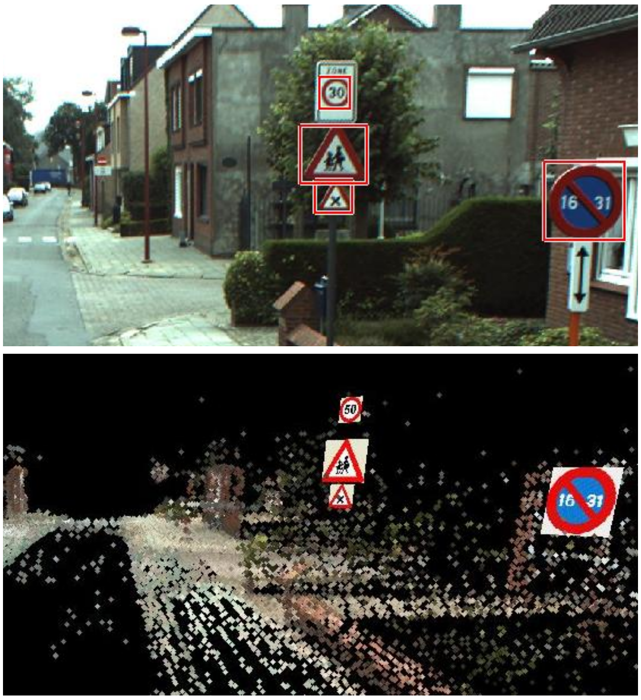


Fig. 1 3D mapped traffic signs in a reconstructed scene.

Keywords Traffic sign recognition · Computer vision-based mobile mapping · Multi-view analysis · Minimum Description Length · Integer Linear Programming

Radu Timofte
ESAT-PSI / IBBT, Katholieke Universiteit Leuven, Belgium
Tel.: +32-16-321704
Fax: +32-16-321723
E-mail: Radu.Timofte@esat.kuleuven.be

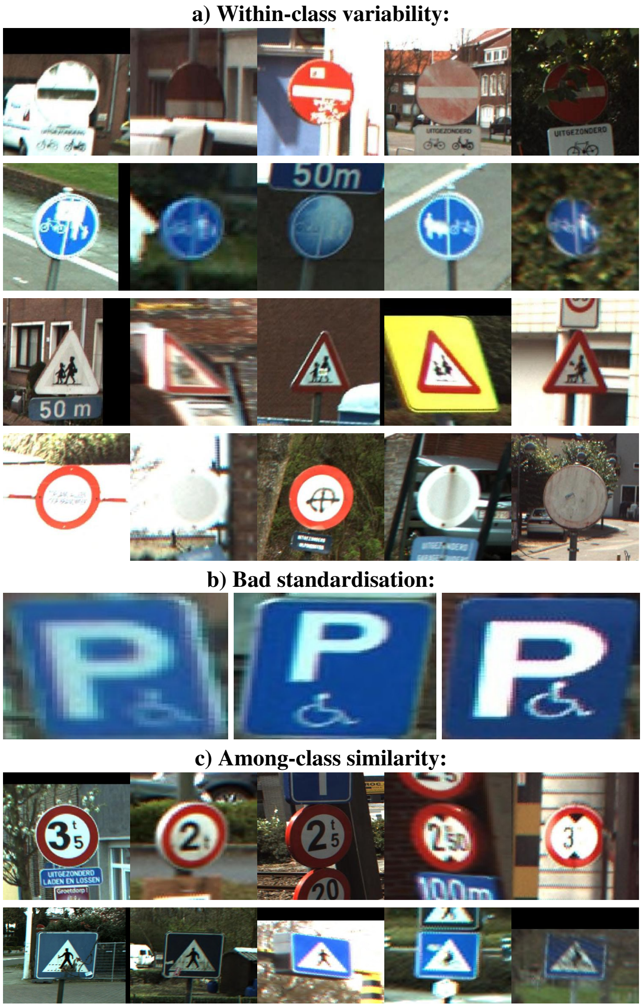
Karel Zimmermann - present address
CMP, Czech Technical University in Prague, Czech Republic
E-mail: zimmerk@cmp.felk.cvut.cz

Luc Van Gool
ESAT-PSI / IBBT, Katholieke Universiteit Leuven, Belgium
E-mail: Luc.VanGool@esat.kuleuven.be

Published online: 30 December 2011
Springer, DOI 10.1007/s00138-011-0391-3

1 Introduction

Mobile mapping is used ever more often, e.g. for the creation of 3D city models for navigation, or to turn old paper maps



a) Within-class variability:

b) Bad standardisation:

c) Among-class similarity:

Fig. 2 The within-class variability and between-class similarity of traffic signs are high. The five first rows show instances of the same class. The last two rows show traffic signs from two distinct classes (first 3 columns vs. last 2 columns).

into digital databases. Several of those applications need the locations and types of the traffic signs along the roads, see Fig. 1. The paper describes an efficient pipeline for the detection and recognition of such signs, from mobile mapping data.

Over the last decade, the computer vision community has largely turned towards the recognition of object classes, rather than specific patterns like traffic signs. However, it would be a mistake to believe that their recognition is not extremely challenging. To be useful, both false positive and false negative rates have to be very low. That is why currently much of this work is still carried out by human operators. There are all the traditional problems of variations in lighting, background, pose, and of occlusions by other objects, see Fig. 2a. In addition, these signs are often not as precisely standardized as one would expect (this also depends on the country; our dataset was acquired in Belgium), see Fig. 2b.

The traffic sign detection problem is traditionally solved by one of the following approaches:

(i) the selective extraction of windows of interest, followed by their classification [14, 17, 19, 3].
(ii) exhaustive sliding window based classification [22, 21, 1].

Approach (i) exploits the saliency traffic signs exhibit by design. A small number of interest regions is selected in the images, through fast and cheap methods. These interest regions are then subjected to a more sophisticated classification. Unfortunately, such approach risks to overlook traffic signs if their assumed saliency has been compromised. See Fig. 12 for some examples.

Approach (ii) considers all regions or 'windows' in the image. As the number of candidate windows is huge, the classification process easily becomes intractable [22]. Additional constraints like minimum and maximum window sizes help to prune that number, at the expense of the number of times the same sign can be detected in image sets of the type we use. Typically, a cascaded classification is applied [1], such that more time is invested in the more promising windows and the vast majority can again be discarded quickly. A single sign often results in multiple detections in overlapping windows, such that a non-maximum suppression is needed as a post-processing step.

In this paper, we contribute to the traffic sign detection problem in the following ways:

Contribution 1: Observing that approaches (i) and (ii) have complementary strengths, we propose their combined use.

Contribution 2: The candidate window selection in approach (i) is usually rather ad-hoc, with thresholds manually chosen. We propose an off-line learning process which automatically selects features and corresponding thresholds.

Contribution 3: We do not stop at single view detection and recognition, but add multi-view 3D localisation. Apart from the value of 3D localisation *per se*, the 3D analysis assists in weeding out false detections while keeping their subset that jointly best explain the observations in the different views.

Contribution 4: An efficient bounded evaluation for linear Discrete AdaBoost-like classifiers [26] is proposed without trading off the performance.

Contribution 5: Since there has been no publicly available database which could serve as a statistically relevant benchmark, we make available such database, as described in Section 7.1 and found at http://homes.esat.kuleuven.be/~rtimofte/traffic_signs/. It contains over 13 000 traffic sign annotations, for more than 145000 images taken on Belgian roads. The image resolution is 1628 × 1236 pixels.

2 State-of-the-art

2.1 Single view detection

The results of traffic sign detection and recognition thus far – often obtained under simpler conditions than in our experiments – testify to the high difficulty of the task.

Lafuente et al. [14] had 26% of false negatives for 3 false positives per image. Maldonado et al. [17] used image thresholding followed by SVM classification. They mention that every traffic sign has been detected at least twice in a total of 5000 video frames, with 22 false alarms. Detection rates per view are not given. In both these methods, thresholds are manually selected. Nunn et al. [21] showed that constraining the search to road borders and an overhanging strip significantly reduces the number of false positives, while false negatives are at 3.8%. In this preselection step, they still found 16494 false positives per image on average using that geometric restriction. All these systems were only tested on highways.

The following systems have also been demonstrated off the highway. Pettersson et al. [22] restricted the detection to speed signs, stop signs and give-way signs. They got 10^{-4} – 10^{-5} false positive rates for 1% false negatives, but fail to mention the number of sub-windows per image. Moutarde et al. [19] reported no false positives at all in a 150 minutes long video, but with 11% of all traffic signs left undetected. Ruta et al. [24] combine image colour thresholding and shape detection, achieving 6.2% false negatives. The number of false positives is not mentioned. Broggi et al. [3] proposed a system similar to [17] where the SVM is replaced by a neural network. No quantitative results are presented.

Although some papers mention the possibility to track the traffic signs, the actual analysis reported in all these papers is based on per-image detection. This is different for the following papers, which consider fused recognition based on multiple detections, as in our case.

In [1] a real-time system for circular traffic signs is proposed that uses a sliding window method. A cascaded AdaBoost detector is trained over Haar-like features defined for each colour channel. The detections are tracked and fused for recognition. A 85% recognition rate is reported for one false positive in every 600 frames (640 × 480 pixel resolution). Ruta et al. in [25] propose a real-time circular traffic sign recognition system that employs colour filtering for red and blue, quad-tree based region of interest extraction, a Hough transform detector with confidence-weighted mean shift refinement, regression tracking based on learning affine distortions over time for specific sign instances, and an AdaBoost variant (SimBoost) for classification. For 720 × 540 pixels videos, they report 12 missclassified signs out of 85 correct detected/tracked traffic signs while not detecting 14 signs and having 10 false detections.

Results so far are not good enough to roll out such methods at a large, urban scale. Both the numbers of false positives and false negatives are too high, or methods are based on assumptions that no longer hold.

Whereas the majority of the previous contributions work with a rather small subset of sign types, our system handles 62 different types of signs. Moreover, the authors usually focus on highway images, whereas our dataset mainly contains images from smaller roads and streets. This poses a more challenging problem as signs tend to be smaller, have more often been smeared with graffiti or stickers, suffer more from occlusions, are often older, and are visible in fewer images. Also, several sign types never appear along highways.

2.2 Multi-view detection

Given the aforementioned limitations with single view methods, it stands to reason to exploit the fact that, typically, a traffic sign is visible in more than one image. Indeed, with the usual mobile mapping vans, multiple, synchronised images are taken a few times per second. This delivers such redundancy and, also, 3D information.

In mainstream computer vision, approaches have recently emerged that try to exploit contextual information. A good example is to use the estimated position of the ground plane, thereby introducing a weak notion of 3D scene layout [11]. This was found to be very beneficial. In a similar vein, Wojek and Schiele [31] went further in coupling object detection and scene labeling approaches. Yet, these approaches still work from a single image. In a mobile mapping setting, a multi-view approach comes natural and can ease such contextual analysis through the explicit 3D information it provides.

As a second strand of relevant research, some recent techniques have focused on detecting and recognizing object related subsets of 3D point clouds [4,20,10]. 3D information is combined with motion, colour, and other data. These systems, which have also been mainly targeting urban scene segmentation and labeling, show remarkable performance. Yet, smaller objects like road signs are among the more difficult ones to handle.

It thus stands to reason to exploit information coming from multiple images. Both the high resolution available in each of those images and the 3D information that can be extracted from them, seem vital inputs. Our method is based on the combination and final selection of detections in a *single 3D space*. Some earlier traffic sign detection methods may have been aggregating detections from multiple views as well, but in different and less exacting ways. e.g. through tracking [1,25,23,27], grouping using GPS information, consistency checks in stereo camera imagery, and/or active vision with high-res regions of interest detected within a low-res camera image [12]. The redundant information coming from the different views is not compiled into a single 3D space, obtained from all views as in our approach.

As a matter of fact, this begs the question what adding additional sensors like laser scanners could do. In [13] such an integrated mobile mapping system is described, but in the automatic mode, the false detection rate is still high and the localisation precision is not better than sub-meter. Adding laser scanning is no miracle cure per se. Before we describe our system in more detail, it is useful to also review literature on the combination of multi-view based detection and tracking.

Fleuret et al. [9] use a multi-view probabilistic occupancy map for people detection and tracking. They globally optimize each individual trajectory separately over long sequences. In contrast, Leibe et al. [15] employ a globally optimal solution for all detections and trajectories at once. The solution is given by a Minimum Description Length (MDL) formulation that inspired also our 3D solution. An important difference lies in the added value of ground plane and space occupancy constraints in their system, however. Neither are of such great help in our traffic sign application. The signs are positioned at varying positions and their volumes are negligible.

Similar challenges are faced by the mobile mapping system in [5], which was designed to find streetlights. Like ours, this system employs multiple cameras mounted on a van, where 2D detections are used to generate 3D hypotheses and their validation is based on back-projection into the images. These authors did use a ground plane constraint and an occupancy map, but at the cost of making strong assumptions about the height above ground and the presence of rather thick poles on which the lamps are fixed. The recent work from [28] uses the same settings as we do, for the 3D mapping of manhole covers. The main assumption is that the manholes are lying on the ground and, thus, the images are projected onto the ground plane and the problem thereby is greatly simplified.

We have to cover cases with signs also fixed to structures like walls or bridges, at rather unpredictable heights. For the aforementioned reasons, we do not make use of these constraints. Also, we formulate criteria for the optimal selection of the basic features (used for detection) and the resulting 3D hypotheses. Moreover, our problem setting imposes the detection of far more object classes, which are typically of a smaller size.

This paper is an extension to our previous work [29]. It contains a more detailed description of the ideas and algorithms, a comparison with a standard sliding window approach as well as with a state-of-the-art part-based approach [8], additional justifications of the design choices made, improved results, as well as the link to the published training and testing datasets.

The structure of the remainder of the paper is as follows. Section 3 first gives an overview of the different steps taken by the system. Then, we focus on the most innovative aspects. Section 4 explains the initial selection of good candidates within the individual images. Section 5 introduces an efficient bounded evaluation of linear AdaBoost-like classifiers, which speeds up the system. Section 6 explains the MDL formulation for 3D traffic sign localisation. Section 7 describes the experimental setup and the results. Section 8 discusses practical issues and comments on the generality of the system. Section 9 draws conclusions.

3 Overview of the system

Before starting with the description of how the traffic signs are detected in the data, it is useful to give a bit more information about our data capturing procedure. Like for most large-scale surveying applications, a van with sensors is driven through the streets. In our case, it had 8 cameras on its roof: two looking ahead, two looking back, two looking to the left, and two to the right. There was an overlap between the fields of view of neighbouring cameras. About every meter, each of the cameras simultaneously takes a 1628×1236 image. The average speed of the van is $\sim$ 35km/h. The cameras are internally calibrated and also their relative positions are known. Structure-from-motion combined with GPS yields the ego-motion of the van.

We do not propose on-line driver assistance but an off-line traffic sign mapping system, performing optimization over the captured views. Only traffic signs captured at a distance of less than 50 meters are considered. The proposed system first processes single images independently, keeping the number of false negatives (FN - the number of missed traffic signs) very low and the number of false positives (FP - the number of accepted background regions) reasonable. Single-view traffic sign detections in conjunction with the multi-view scene geometry subsequently allows for a global optimization. This optimization simultaneously performs a 3D localisation and refinement. Since we deal with hundreds of thousands of high-resolution images the approach is to quickly throw out most of the background, and to then invest increasing amounts of time on whatever patterns survive previous steps.

We now sketch the different steps of the single-view and multi-view processing pipelines. The next two sections then give a more detailed account of these pipelines, resp.

The **single-view** detection phase consists of the following steps:

1) Candidate extraction - very fast preprocessing step, where an optimized combination of simple (i.e. computationally cheap), adjustable extraction methods selects bounding boxes

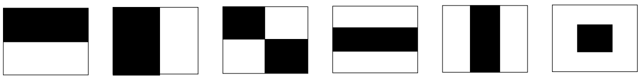


Fig. 3 Haar-like features used in our implementation.

with possible traffic signs. This step requires an automatic off-line learning stage, where an appropriate subset of features and corresponding decision rules is selected. They should yield very high detection rate (FN very low), while keeping the number of false positives in check. This part of the pipeline is described in more detail in Section 4.

2) Detection - Extracted candidates are verified further by a binary classifier which filters out remaining background regions. It is based on the well known Viola and Jones [30] Discrete AdaBoost classifier [26]. The 6 Haar-like patterns used are shown in Fig. 3. Detection is performed by cascades of AdaBoost classifiers, followed by an SVM operating on normalized RGB channels, pyramids of *Histogram of Oriented Gradients*(HOGs) [2] and AdaBoost-selected Haar-like features. The detection time is reduced by using an efficient bounded evaluation of the AdaBoost classifiers, further explained in Section 5.

3) Recognition - Six one-against-all SVM classifiers select one of the six basic traffic sign subclasses (triangle-up, triangle-down, circle-blue, circle-red, rectangle and diamond) for the different candidate traffic signs. They work on the RGB colour channels normalized by the intensity variance.

The **multi-view** phase consists of the following steps:

4) Multi-view hypothesis generation - We search for possible correspondences among the final, single-view candidates in the different views. The search is restricted to a volume with a predefined radius in 3D space. Every geometrically and visually consistent pair is used to create a 3D hypothesis. Geometric consistency amounts to checking the position of the back-projected 3D hypothesis against the 2D image candidates. Visual consistency gives a higher weight to pairs which are more probable to be of the same basic shape.

5) Multi-view MDL hypothesis pruning - The Minimum Description Length principle is used to select the subset of 3D hypotheses which best explains the overall set of 2D (i.e. single-view) candidates. A by-product of the MDL optimization is quite a clean set of 2D candidates corresponding to each particular 3D hypothesis. These candidates allow for 3D hypothesis position refinement. Usually, steps 4) and 5) are iterated. More details are given in Section 6.

6) Multi-view sign type recognition - The collected set of 2D candidates for each 3D hypothesis is classified by an SVM classifier. These classifications then jointly vote on the final type assigned to the hypothesis.

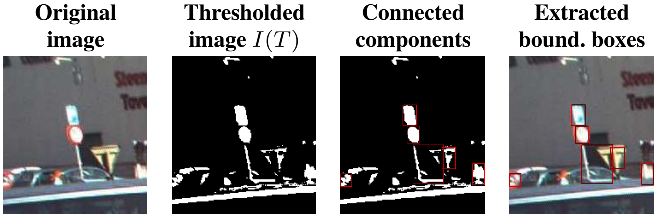


Fig. 4 Colour-based extraction method for threshold $T = (0.5, 0.2, -0.4, 1.0)^\top$

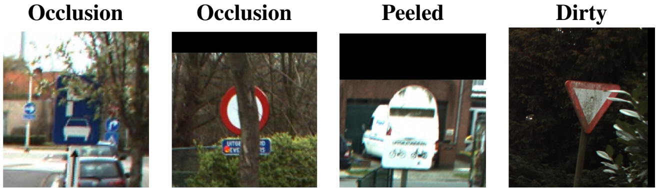


Fig. 5 Not threshold separable traffic signs. There are still traffic signs which are not well locally separable from background; therefore shape-based extraction is used.

4 Single-view candidate extraction

The simplest extraction method often used for traffic sign detection is extraction of connected components from a thresholded image, an idea already used in [17,3]. The principle is outlined in Fig. 4. The *thresholded image* is obtained from a colour image, with colour channels (I_R, I_G, I_B), by application of a colour threshold $T = (t, a, b, c)^\top$:

$$I(T) = \begin{cases} 1 & a \cdot I_R + b \cdot I_G + c \cdot I_B \geq t \\ 0 & \text{otherwise} \end{cases} \qquad (1)$$

Authors often manually select two to five thresholds, which are expected to extract all traffic signs. However, we experimentally observed that under variable illumination conditions and in the presence of a complex background such extraction method is insufficient.

Since there typically is no single threshold performing well by itself, it is necessary to combine regions selected by different thresholds $\mathcal{T} = \{T_1, T_2, \dots\}$, in the sense of adding regions (OR-ing operation). Then, regions passed on by any threshold are going to the next stage, i.e. detection. The more thresholds are used the lower FN can be made but the higher FP risks to get, and the higher the computational cost will be.

Partially occluded, peeled or dirty traffic signs also should pass the colour test. Therefore, this cannot be made too restrictive. Examples are shown in Fig. 5. That is why we also employ shape information to further refine the candidates.

Section 4.1 explains how the set of colour thresholds are learned and how, starting from those, the colour-based candidates are extracted. Section 4.2 then describes a shape-based Hough transform. This takes the borders of the colour-based candidates as input.

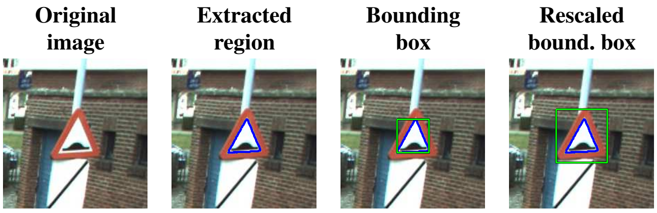


Fig. 6 Demonstration of the extended threshold. The object is not well locally separable from the background, because bricks have a colour similar to that of the red boundary. Therefore the inner white part is extracted and the resulting bounding box is rescaled $\overline{T} = (0.1, -0.433, -0.250, 0.866, 1.6, 1.6)^\top$.

4.1 Colour-based candidate extraction

Given thousands of possible colour thresholds, we search for the optimal subset $\mathcal{T}$ of such thresholds, given some criterion. Since for most interesting such criteria the problem is NP-complete, we formulate our search as an Integer Linear Programming problem. We have experimentally found that finding the real optimum takes several hours, but that ILP, due to the sparsity of the constraints, yields a viable solution within minutes.

The most straightforward criterion is to search for a trade-off between FP and FN.

$$\mathcal{T}^* = \arg\min_{\mathcal{T}} \left(\text{FP}(\mathcal{T}) + \kappa_1 \cdot \text{FN}(\mathcal{T})\right), \tag{2}$$

where FP$(\mathcal{T})$ stands for the number of false positives and FN$(\mathcal{T})$ for the number of false negatives, resp., of the selected subset of thresholding operations $\mathcal{T}$ measured on a training set. The real number κ_1 is a relative weighting factor. In order to avoid overfitting and also to keep the method sufficiently fast, we introduce an additional constraint on the cardinality card$(\mathcal{T})$ of the set of selected thresholds. This can be either a hard constraint card$(\mathcal{T}) < \omega_0$ or a soft constraint as in:

$$\mathcal{T}^* = \arg\min_{\mathcal{T}} \left(\text{FP}(\mathcal{T}) + \kappa_1 \cdot \text{FN}(\mathcal{T}) + \kappa_2 \cdot \text{card}(\mathcal{T})\right) \tag{3}$$

We achieved better results with the soft constraint, but imposing a hard constraint may be necessary if the running time is an issue. Since *accuracy*, defined as the average overlap between ground truth bounding boxes with extracted bounding boxes, is important, we also add a term which increases the penalty for inaccurate extractions:

$$\begin{aligned}\mathcal{T}^* = \arg\min_{\mathcal{T}} & \left(\text{FP}(\mathcal{T}) + \kappa_1 \cdot \text{FN}(\mathcal{T})\right. \\ & \left. + \kappa_2 \cdot \text{card}(\mathcal{T}) - \kappa_3 \cdot \text{accuracy}(\mathcal{T})\right)\end{aligned} \tag{4}$$

Scalars κ_1, κ_2 and κ_3 are learned parameters which we estimate by cross-validation. Reformulations of problems (2,3,4) into the Integer Linear Programming form are described in the Appendix.

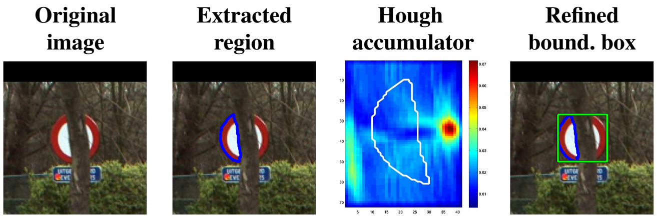


Fig. 7 Shape-based extraction principle. The border of the colour-based extracted region (blue) votes for different shapes in a Hough accumulator. The green bounding box corresponds to the maximum.

Occasionally it happens that the contour of the traffic sign cannot be separated from the background due to colour similarity. See for example Fig. 6, where the rim of the sign is too similar in colour to the background. Fortunately, many traffic signs have also some inner contours (e.g. the white inner part of the sign in Fig. 6, can be separated rather easily). This inner part can often define the traffic sign's outline with sufficient accuracy. We therefore introduce the extended threshold

$$\overline{T} = (\underbrace{t, a, b, c}_{T}, s_r, s_c)^\top \tag{5}$$

which consists of the original threshold T and vertical resp. horizontal scaling factors (s_r, s_c) to be applied to the bounding box which is extracted with the original threshold. Such extended threshold - in the sequel simply referred to as threshold - can reveal a traffic sign, even if its rim poses problems.

Changing illumination poses another problem to thresholding. One could try to adapt the set of thresholds to the illumination conditions, but it is better to add robustness to the thresholding method itself. We adjust the threshold to be *locally stable* in the sense of Maximally Stable Extremal Regions (MSER) [18]. Instead of directly using the bounding box as extracted by the learned threshold (t, a, b, c, s_r, s_c), we use bounding boxes from MSERs detected within the range $[(t-\epsilon, a, b, c, s_r, s_c); (t+\epsilon, a, b, c, s_r, s_c)]$, where ϵ is a parameter of the method. Since MSERs themselves are defined by a stability parameter $\varDelta$, this 'TMSER' method is parametrized by two parameters $(\epsilon, \varDelta)$.

4.2 Shape-based candidate extraction

Traffic signs are meant to be well distinguishable by both their colour and shape. Each of the above thresholds (with scaling and TMSER extensions) let pass a series of connected components, i.e. regions (usually thousands per image). To these regions we now apply an additional shape-filter, akin to the generalized Hough transformation. The principle is outlined in Fig. 7.

In general the image shapes of the signs will be affinely transformed versions of the actual shapes. Using the gener-

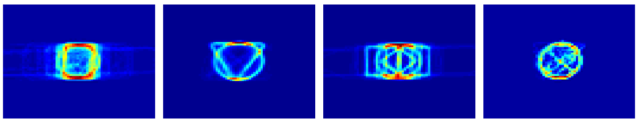


Fig. 8 Threshold-specific fuzzy templates. Selected subset $\{23, 12, 28, 32\}$ from 44 fuzzy-templates.

alized Hough transformation in its traditional form would require to detect every single shape in 5D (or even 6D) Hough accumulator spaces. Apart from the computational load involved, working in such vast spaces is almost guaranteed to fail. Instead, we learn *fuzzy templates* which incorporate small affine transformations and shape variations and we determine explicitly only the position and scale in a 3D Hough accumulator.

The most straightforward fuzzy templates could be learned as a probability distribution of boundaries of colour-based extracted regions for specific signs. Such approach, however, would require as many templates as there are different shapes. A more parsimonious use of templates is possible, however. Since the learned thresholds (Eq. (5)) are usually specialized for some specific basic shapes of traffic signs, we learn threshold-specific fuzzy templates, which allow the system to try only one template per extracted boundary. Fig. 8 gives examples. For each threshold, we first collect boundaries of extracted regions which yield correct bounding boxes. Then the scale is normalized (aspect ratio is preserved) and the probability distribution of the shapes extracted by the threshold is computed. Eventually, the fuzzy template is estimated as the point reflection of the probability distribution, because voting in the Hough accumulator requires the point-reflected shape. For example, the second fuzzy template in Fig. 8 corresponds mainly to traffic signs which are circular or upward-pointing triangular, whence the downward-pointing triangular part of the template (in addition to the circular part).

When a boundary is extracted by a threshold, the threshold-specific fuzzy template is used to compute its generalized Hough transformation. A bounding box corresponding to the maximum in the three dimensional Hough accumulator (2 positions and 1 scale) is reported if the maximum is sufficiently high. The role of the shape selection step mainly consists of selecting a sub-window from a colour-defined bounding box, with the right shape enclosed. In order to avoid replacement of correctly extracted bounding by a bounding box corresponding to a small sub-boundary which has more exact shape than the original one, the original bounding box is also kept.

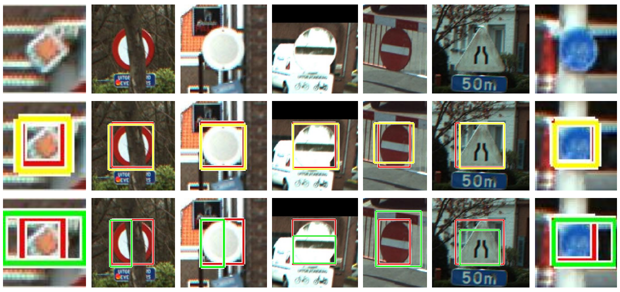


Fig. 9 Shape-based extractable but colour threshold inseparable traffic signs - the ground truth is delineated by a red rectangle, the best shape-based detection is shown in yellow and the best colour-based one in green.

5 Efficient bounded evaluation of AdaBoost classifiers

Here we show a simple way to speed up the evaluation of linear combinations of the form used in our Discrete AdaBoost classifier implementation.

The result of the AdaBoost algorithm is a ‘strong’ classifier constructed as a linear combination

$$f(x) = \sum_{t=1}^{L} \alpha_t h_t(x) \qquad (6)$$

of L ‘simple’‘weak’ binary classifiers/features $h_t(x) : X \to \{-1, +1\}$, where α_t are the weights and X is the space (image) from where x is sampled. The thresholded decision of the final classifier is

$$H(x) = sign\{f(x) - \theta\} \qquad (7)$$

where θ is the threshold.

Since the values of h_t have upper and lower bounds, the partial sums of terms in Eq. (6) are also bounded. Let $\overline{h_t} = +1$ and $\underline{h_t} = -1$ be the upper and lower bounds for h_t. We observe that in order to evaluate $H(x)$, we do not have to compute all h_t, but we can stop after computing s terms if

$$\sum_{t=1}^{s} \alpha_t h_t(x) + \sum_{t=s+1}^{L} \alpha_t \overline{h_t(x)} < \theta \qquad (8)$$

implying that $f(x)$ lies *below* the threshold θ even if all the remaining terms $(s + 1, \ldots, L)$ are at their upper bounds. Also, we can stop after s terms if

$$\sum_{t=1}^{s} \alpha_t h_t(x) + \sum_{t=s+1}^{L} \alpha_t \underline{h_t(x)} > \theta \qquad (9)$$

in which case $f(x)$ would be *above* the threshold θ even if all the remaining terms $(s + 1, \ldots, L)$ are at their lower bounds.

The sums for upper and lower bounds do not depend on the actual value of $h_t(x)$ and are precomputed.

By dropping the evaluation of the whole linear combination and considering the bounded intervals we already get a decrease in computation time for our AdaBoost cascades of 20% up to 30%.

If we first evaluate the terms that contribute the most we get a further computation time reduction. The terms that have the strongest influence on $f(x)$ are those with the largest values for $\alpha_t(\overline{h_t(x)} - \underline{h_t(x)})$.

Since the weights, the upper and lower bounds are known and fixed in our case, we can first sort the terms in descending order according to their $\alpha_t(\overline{h_t(x)} - \underline{h_t(x)})$ values (or just α_t values in our case), and afterwards compute the partial sums for the worst and most favorable cases. By sorting first, we experimentally obtain a decrease in computation time of up to 40%.

Another way of exploiting the linear combinational nature of our classifiers is to employ the training material and to extract frequencies for each term for each particular value or value interval. Thus, the evaluation order will be given by using these frequencies coming from the training material, and the computation time reduction would be obtained along with an estimated probability. This idea is not explored here.

Note that the methods applied here have no impact on the decision values of the considered classifiers but only (in general) improve the computational time. Also, similar methods are applicable to other classifiers based on a linear combination of local, weak decisions.

6 Multiple-view MDL 3D optimization

Single-view detection and recognition is just a preprocessing stage, and the final decision results from global optimization over multiple views, based on the Minimum Description Length principle (MDL). Given the set of images, single-view detections, camera positions and calibrations, MDL searches for the smallest possible set of 3D hypotheses which sufficiently explains all detected bounding boxes. In other words, if a set of detected bounding boxes satisfies some *geometrical and visual constraints*, then all of these bounding boxes are explainable by one 3D traffic sign. Next, we explain how MDL is used for that purpose.

We start by generating an overcomplete set of hypotheses: For every single 2D detection we collect every *geometrically* and *visually* consistent correspondence in another image and use this pair to generate a 3D hypothesis, see Figure 10. Geometrical consistency means that the corresponding detection lies on the epipolar line for the camera pair. Visual consistency means that their recognized subclass types are the same. This step, of course, generates a high number of 3D hypotheses, including false positives and multiple, close but seemingly different 3D reconstructions for the same sign (3D reconstructions are generated from image pairs). The following MDL optimization selects the simplest subset which best explains the 2D detections. For some further explanation, see Fig. 10, right.

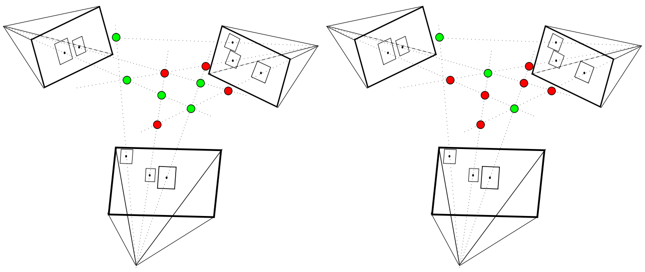


Fig. 10 MDL principle - the corresponding pairs generate 3D hypotheses, from which can be picked up (*green*) a subset (*left*) or the best/smallest subset (*right*) in the MDL sense that explains the 2D detections.

For each 3D hypothesis we will have a 3D position of the centre of the traffic sign, its fitted plane and thus an orientation (and sense), and estimated probabilities to belong to each of basic shapes. For a specific hypothesis h we gather the set of supporting 2D candidates which have a *coverage*[1] with the 2D projection of h above 0.05 and for which the candidate camera and the hypothesis are facing each other (rather than the camera observing the backside of the sign), at less than 50 meters. Let the set of 2D candidates be C_h.

In order to define the MDL optimization problem, we first compute *savings* (in coding length) for every single 3D hypothesis h as follows:

$$S_h \sim S_d - k_1 S_m - k_2 S_e \tag{10}$$

where S_d is the part of the hypothesis which is explained by the supporting candidates (Eq. (12)), i.e. a weighted sum of coverages as explained shortly. S_m is the cost of coding the model itself (a constant penalty in our case), while S_e represents those parts that are not explaining the given hypothesis (Eq. (13)), and k_1, k_2 are weights (as in [15]). For each candidate c we have a 2D projection of h, whence the coverage $O_{c,h}$ of the projected h and the candidate c. The coverage assures independence of the size of supporting candidates. The estimated probability that the candidate explains the hypothesis is taken as the maximum of the probabilities of them sharing a specific basic shape:

$$p(c,h) = \max_{t \in \{\triangle, \triangledown, \circ, \square, \diamond\}} p_t(c) p_t(h) \tag{11}$$

$$S_d = \sum_{c \in C_h} O_{c,h} p(c,h) \tag{12}$$

$$S_e = \sum_{c \in C_h} (1 - O_{c,h}) p(c,h) \tag{13}$$

[1] Coverage is the ratio between the intersection and the union of areas.

Table 1 Belgian Traffic Signs Dataset (BelgiumTS). To the traffic sign (TS) annotations corresponds a number of physically distinct TS. On average we have 3 views/annotations for each physical TS. *3D Testing* contains the TS annotations along with the image/frame sequences where those appeared, and each image is provided with camera parameters and pose. The TS annotations from *3D Testing* form a subset of *2D Testing*. *non-TS* stands for images without traffic sign annotations.

BelgiumTS	TS annot.	Distinct TSs	other images
Training	8851	3020	16045 non-TSs
2D Testing	4593	1545	583 non-TSs
3D Testing	1625	552	121632
Total	13444	4565	16628 non-TSs

We assume that one candidate can explain only one hypothesis. Interaction between any two hypotheses h_i and h_j that get support from shared candidates $C = C_{h_i} \bigcap C_{h_j}$ should be subtracted and is given by

$$S_{h_i,h_j} = \sum_{c \in C} \min_{t \in \{i,j\}} (S_{d_t}(c) - k_2 S_{e_t}(c)) \tag{14}$$

where $S_{d_t}(c)$ and $S_{e_t}(c)$ are constrained to the contribution of c for h_t

Leonardis et al. [16] have shown that if only pairwise interactions are considered, then the Integer Quadratic Problem (IQP) formulation gives the optimal set of models:

$$\max_n n^T S n, \quad S = \begin{bmatrix} s_{11} & \cdots & s_{1M} \\ \vdots & \ddots & \vdots \\ s_{M1} & \cdots & s_{MM} \end{bmatrix} \tag{15}$$

Here, $n = [n_1, n_2, \cdots, n_M]^T$ is a vector of indicator variables, 1 for accepted and 0 otherwise. S is the interaction matrix with s_{ii} being the *savings*, $s_{ii} = S_{h_i}$, while the others are representing the interaction costs between two hypotheses h_i and h_j, $s_{ij} = s_{ji} = -0.5 S_{h_i,h_j}$. The restriction to pairwise interactions does not fully cater for situations where more than 2 hypotheses affect the same image area.

7 Experiments

7.1 Ground truth data

We have collected ground truth data used for this paper. This database, the *Belgian Traffic Signs Dataset* (BelgiumTS), is publicly available at: `http://homes.esat.kuleuven.be/~rtimofte/traffic_signs/`. The dataset contains 13444 traffic sign annotations in 9006 still images corresponding to 4565 physically distinct traffic signs visible at less than 50 meters from the camera. The dataset includes challenging samples as shown in Fig. 2.

Table 1 summarises the most important information about this dataset. It is split into different subsets, corresponding to the rows in the table. For each subset, we indicate the number of traffic sign annotations (2nd column), the number of different signs these correspond to (3rd column), and the number of number of 'background' images without traffic signs (4th column).

The first row describes the Training subset. The annotations therein have been used to train for traffic sign detection, segmentation, and recognition. As negative examples, we use 16045 'background' images, which contain no traffic signs. The 2D Testing subset was used for the validation of the detection. Images were handpicked, so that the majority contains traffic signs. Again, a number of background images without any signs were added. The 3D Testing subset contains continuous sequences of images (8 camera images per meter of road), i.e. with lots more images in between those where traffic signs are visible. This dataset was used for the full pipeline of detection, recognition, and localisation. The 3D Testing traffic sign annotations form a subset to those for 2D Testing. Yet, in total it contains many more images than 2D Testing, i.e. 121632 images from the 8 cameras. All of these come with camera poses and internal calibrations.

To ease the use of the dataset for classification benchmarking, we provide a subset called BelgiumTSC (Belgium TS for Classification) with 4591 cropped training samples and 2534 cropped testing samples. These correspond to the original BelgiumTS Training and 2D Testing parts but restricted to only 62 traffic sign types as used also in this work.

7.2 Single-view evaluation

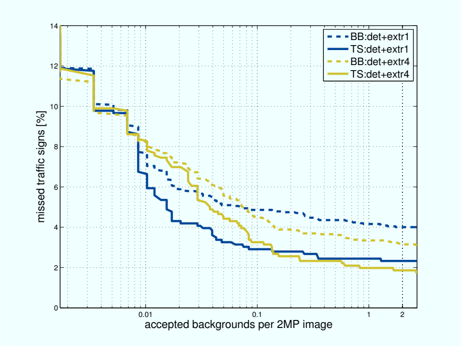
The Training part of BelgiumTS (Table 1) is used for learning the suitable candidate extraction methods as well as for training the AdaBoost cascades and the SVM classifiers. To learn the SVM classifier, Statistical Pattern Recognition Toolbox[2] is used. The 2D Testing part is used for assessing the performance. Our current method has only been trained for 62 traffic signs classes. As a result, the number of used annotations in testing drops to 2571, corresponding to 859 physically distinct traffic signs.

The detection and extraction errors (Table 2) are evaluated according to two criteria: either demanding detection every time a sign appears (FN-BB), or only demanding it is detected at least once (FN-TS). On average, a sign is visible in about 3 views. When False Negatives are mentioned in the literature, it is usually FN-TS which is meant, where the number of views per sign is often even higher (highway conditions). We considered a detection to be successful if the *coverage* ≥ 0.65, which approximately corresponds to the shift of a 20×20 bounding box by 2 pixels in both directions. Note that some of our detected signs are quite

[2] `http://cmp.felk.cvut.cz/cmp/software/stprtool/`

Table 2 Summary of achieved results in single-view detection. Meaning of the above used abbreviations is the following **colour** means method described in Section 4.1, **TMSER** stands for TMSER(ϵ, Δ) $=$ TMSER$(0.1, 0.2)$, **shape** is Section 4.2. **FN-BB** means false negative with respect to bounding boxes, **FN-TS** means false negative with respect to traffic signs. The graph depicts the detection performance for 2 candidate extraction settings: **Extr1** and **Extr4**.

	FN-TS		FN-BB		FP per
	[%]	#/859	[%]	#/2571	2MP img
Extr1 (colour)	0.7%	6	1.1%	29	3 281.8
Extr2 (colour+TMSER)	0.7%	6	1.1%	28	3 741.7
Extr3 (colour+shape)	0.5%	4	0.7%	17	5 206.2
Extr4 (colour+TMSER+shape)	0.5%	4	0.7%	17	5 822.0
Det + Extr1	2.3%	20	4.0%	103	2
Det + Extr4	1.9%	16	3.2%	82	2



small, with the smallest 11×10. Approximately 25% of non-extracted bounding boxes were smaller than 17×17, most of the others were either taken under oblique angles and/or were visually corrupted (e.g. covered by a sticker, heavily occluded, etc.).

Table 2 shows results of both the candidate extraction (still with an appreciable number of FP, see first four rows) and the final detection (i.e. candidate extraction followed by AdaBoost detector and SVM, see last two rows). The ROC curve in Table 2 compares the FN-BB/FN-TS achievable with our pure colour-based extraction method to that with our combined (colour+TMSER+shape)-based extraction method. The shape extraction significantly increases the number of false positives (see for example 4th row in the table). The reason for is that we keep both the original colour bounding boxes and add all bounding boxes that reflect a good shape match. Combined extraction lowers FN, however. Fig. 9 shows traffic signs that could not be detected completely with the colour thresholds, but which could then still be extracted based on their shape.

7.3 Sliding window comparison

We compare the pipeline outlined so far with a sliding window approach. For the latter, we train Discrete AdaBoost cascades directly on sampled subwindows from the Training data (see Table 1). The parameters for sliding window are: 350 pixels minimum window size, 4 aspect ratios - (0.5,0.75,1.0,1.25), 6.67% shift and 1.15 scaling factor. Under these conditions the number of processed windows per 1628×1236 image is higher than 12 million. For testing we use the same 2D Testing dataset (Table 1). For the features, Haar-like masks are computed on HSI channels, as before.

The Matlab/C++ scripts ran weeks for training all the cascades for the sliding window approach. Compared to this, the days of training for the original pipeline looks modest. We trained cascades for each subclass of traffic signs: triangle-up (28 stages cascade), triangle-down (27 stages), circle-blue (26 stages), circle-red (23 stages), rectangle (23 stages) and diamond (25 stages).

The output of the cascades is processed further by a SVM classifier that uses Haar-like features, pyramids of HOGs and pixels in RGB space. All features are variance normalized and mean subtracted and then concatenated into a single feature vector, which serves as input to a linear SVM.

Fig. 11 shows the performance of the sliding window approach, of the **Det+Extr1** pipeline (Table 2), and of their combination. The 2D Testing set has been used. The sliding window approach outperforms the **Det+Extr1** pipeline for low numbers of FN or FP, both for the BB and TS criteria. Nevertheless, if we allow for higher FN or FP (which is the case, as the 3D analysis prunes away most single view errors), then the **Det+Extr1** pipeline is better in terms of TS detections. The performance can be improved by combining the sliding window and the **Det+Extr1** pipelines. Their outputs are combined, all put through a linear SVM, and then selected by thresholding their confidences. Fig. 12 shows cases that could be detected by one pipeline but not the other. Thus, if the computation time is not crucial, running both approaches is advantageous. In our single-core / single-thread implementations, the **Det+Extr1** pipeline is about 50 times faster than the sliding window pipeline. The **Det+Extr1** pipeline achieves about 2 frames per second.

7.4 Part-based model comparison

Here we compare with the state-of-the-art generic object class detector of Felzenszwalb et al. [8]. This discriminatively trained part-based model detector is the top performer of PASCAL VOC Challenge 2009 [6].

The system relies on discriminative training with partially labeled data. The authors combine a margin-sensitive

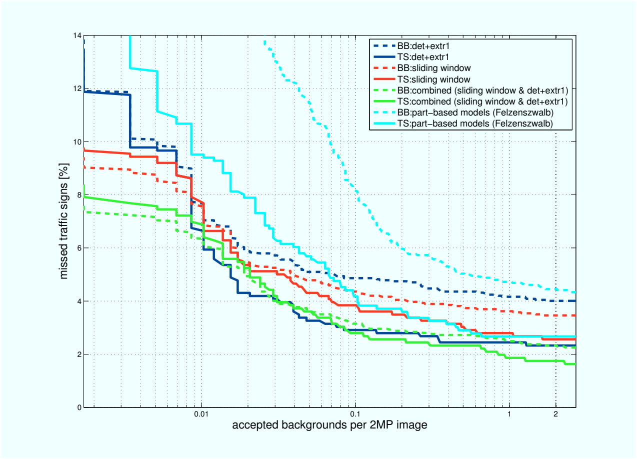
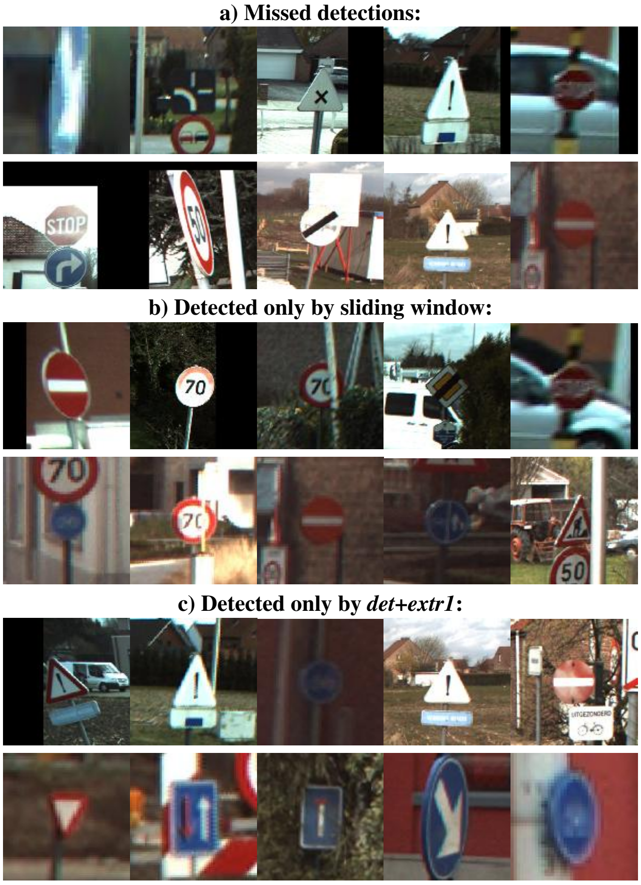


Fig. 11 Comparison with state-of-the-art methods. - Detection plots for every time a sign appears (**BB** - bounding box level), or only demanding it is detected at least once (**TS** - traffic sign level). In blue one finds results for the main pipeline presented in this paper. The results for the alternative sliding window approach are shown in red. Green shows results for their combined use. The combined performance is better than the sliding window approach and our proposed approach with **Extr1** extraction setting taken alone. In cyan is the result of the generic part-based model system from Felzenszwalb et al. [8]

approach for data-mining hard negative examples with a formalism called latent SVM which is a reformulation of MI-SVM in terms of latent variables. A latent SVM is semi-convex and the training problem becomes convex once latent information is specified for the positive examples. This leads to an iterative training algorithm that alternates between fixing latent values for positive examples and optimizing the latent SVM objective function. The HOGs are the basic features employed by this method.

For a fair comparison, we use the publicly available scripts. We train on the Training part of BelgiumTS (see Table 1) a model with 5 components which correspond to the basic shapes of the traffic signs. The 2D Testing material is used for assessing the performance.

Fig. 11 shows how this part-based model detector compares with our proposed systems. The poorer performance when compared with our specialized systems is believed to come from the fact that this approach is a generic one and works on HOG features. The running time compares to the sliding window approach for 2Mpxl images. The cascaded version from the same authors [7] is expected to provide the same accuracy and an up to 20 times speedup, but the system is still far from realtime performance. Thus our **Det+Extr1** pipeline exhibits better accuracy and is much faster than the part-based models variants considered here.


a) Missed detections:

b) Detected only by sliding window:

c) Detected only by *det+extr1*:

Fig. 12 Complementarity of sliding window and the proposed approach. Shown are samples where both methods fail (**a**) and where one method fails but the other one is successful (**b,c**), at the same threshold level.

7.5 Multi-view evaluation

In this section, we report on the multi-view results. Moreover, in the single-view case we only paid attention to the detection of traffic signs, not yet to their recognition or localisation. Here we will also cover these topics. The inclusion of correct localisation within 3 meters in X-Y-Z within the criteria explains why some of the scores go down with respect to the single-view case. Most of the incorrectly 3D localised traffic signs were detected in at least one view.

We evaluate our multi-view pipeline, based on **Det+Extr1** single-view processing, on the 4 image sets of the 3D Testing part of BelgiumTS (see Table 1). The evaluation is restricted to a subset of 62 traffic sign classes. These include all regular signs, but exclude direction indicators with text. A breakdown of the test data per class, along with its achieved performance, is shown in Fig. 13. The results are summarized in Table 3. The operating point was selected to minimize FP at better than 95% correct localisation. This could be shifted towards a better localisation rate at the cost of more FP (see Fig. 14 for false detections). Fig. 15 shows

Table 3 Summary of 3D achieved results. **Localised TS** means correctly located traffic signs in 3D space, **FP** stands for false positives in 3D and **Recognised TS** are the 3D recognition results with respect to the located 3D TS.

#	No.frames	No.TSs	3D Localised TS	FP	Recognised TS
1	8 × 3001	99	94(95.0%)	3	90(95.7%)
2	8 × 6201	87	83(96.5%)	7	81(97.6%)
3	8 × 2001	47	44(93.6%)	2	43(97.7%)
4	8 × 4001	86	83(96.5%)	8	81(97.6%)
∑	8 × 15204	319	304(95.30%)	20	285(97.04%)

samples of missed traffic signs (i.e. not detected, misplaced or wrongly classified). The main causes are occlusions, a weak confidence coming from the detection and/or few views where a sign is visible. The average accuracy of localisation (distance between the 3D position according to the ground truth and the 3D reconstructed traffic sign) is 26 centimeters. 90% of the located traffic signs are reconstructed within 50 centimeters from the ground truth, but we have also 3 traffic signs that are reconstructed at more than 1.5 meters.

The recognition results are summarized in the last column of Table 3. The overall classification rate is 97% with 95.30% accurately 3D-localised traffic signs. In comparison, Ruta et al. [25] achieve 85% classification rate with 86% traffic signs detected, on a smaller dataset but using a real-time system with only a single front camera and exploiting tracking.

8 Discussion

Having introduced our system and the experiments that we perform, we now discuss topics such as trading off performance for speed, practical aspects and the generality of the approach, and driver assistance/real-time applications.

8.1 Performance versus speed

The performance/speed tradeoff is an often returning topic. We considered a processing time of 2fps (operating on 2Mpixel images) to be sufficient if a very high detection rate (> 95%) of close-by traffic signs (within 50 metres) and a very low false detections per image (< 2) could be warranted. Our experiments corroborated that using a multi-view/3D analysis helps a lot in pruning the false detections while getting very high 3D localisation rates. This is another observation to keep in mind when putting together a final system.

Given the above, deploying a **det+extr1** pipeline (see Section 7) makes sense. This proved to yield a speed of 2fps at a level of 96% detection rate and 2 false positives per image (see Table 2), in line with our goals. Combined with the multi-view/3D analysis MDL optimization we exceed 95% accurate 3D localisation rate (see Table 3) with very few false positives (20 in 8 × 15204 recorded frames). These results are quite satisfactory given that our system working without any human supervision.

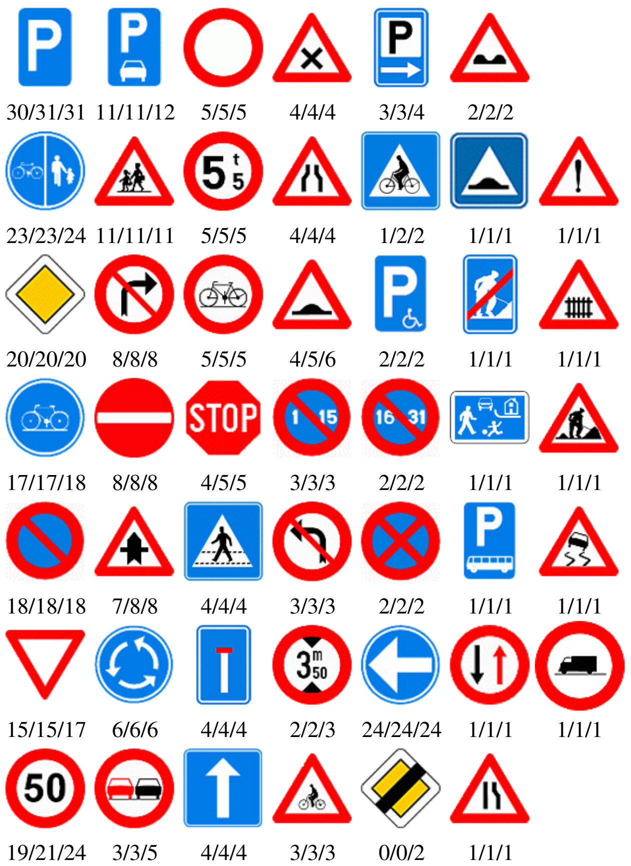


Fig. 13 Breakdown of the traffic sign class occurrence and performance in the test data. The information is provided for each class as *a/b/c*, where *a*–the number of distinct signs from the class correctly 3D localised and recognised, *b*–the number of distinct signs from the class correctly 3D localised and *c*–the number of distinct signs from the class that appears in the test data.

In all steps we could trade speed off for better performance in localisation. This is doable by using more basic threshold methods in the segmentation step or combining with complementary sliding window and/or part-based model approaches (see Fig. 11). Also, if an human operator is post-filtering the results then we could allow more false positives at image level (for per image detection) and at the traffic sign level (for 3D localisation), which would improve the mapping performance.

8.2 Generality

The results in this paper have been presented with a particular application in mind and using a specific setup. Yet, the proposed methods as well as the overall pipeline lend themselves to applications in different contexts and with different

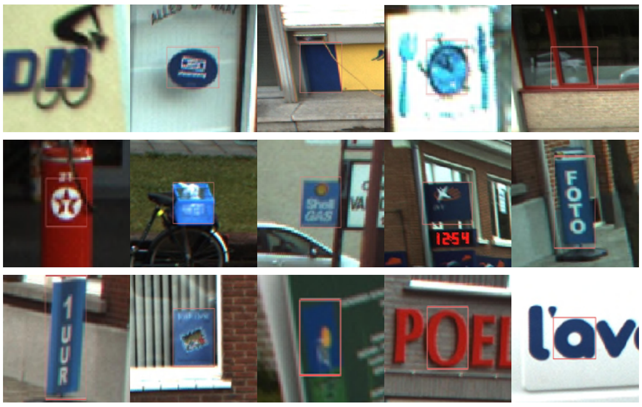


Fig. 14 False positives.

imagery. For instance, the optimisation for picking the best thresholds has a general formulation (see Section 4). Parameters like $\kappa_{1,2,3}$ can be adapted based on training data. The number of annotations, the ratio between false positives and negatives, and the precision of segmentation can be used to set parameters. Thereafter, parameter settings can be fine-tuned further on the basis of cross-validation with sets of parameters in their close vicinity. The initial setting takes a few hours, while cross-validating tens of settings would take a few days, however.

Our cameras yielded substantially different colours. Moreover, the illumination conditions vary a lot (e.g. strong sunlight, shadows). In the absence of a colour normalization and/or illumination compensation of the input images, as in our case, the segmentation thresholding criteria seem capable of largely making up for this. We have also experimented with imagery of lower quality (also taken in Belgium, from a different type of mobile mapping van) and the drop in segmentation performance was less severe than anticipated, with exactly the same thresholding criteria. Note that if a different country would be involved, then certainly the detection and recognition need to be retrained, as the signs will be somewhat different.

8.3 Real-time applications

Our mapping system has always been intended for off-line processing, mainly because our structure-from-motion runs offline and has to be applied prior to the traffic sign part. Indeed, it yields the necessary camera poses, needed for the image fusion and 3D localisation. Otherwise, there clearly is on-line potential. The **det+extr1** pipeline (see Section 7) works at 2fps on 2 Mpixel images and at 16fps on 640×480 pixel images (VGA resolution). The running time increases linearly with the number of pixels. A speed of 16fps is already within the range for driver assistance. On the other hand, for the automated mapping of traffic signs, there is no critical need for on-line processing and it is better to make the most out of the collected data in order to increase the precision (e.g. after driving by the same spot multiple times, which often happens for crossroads where many of the traffic signs are to be found).

This said, we have experimented with driver assitance as well, for which we proposed a slighly modified version of our pipeline [27,23]. Typically only one camera can be used in such case. Yet, still one can combine frame level detection/recognition with 3D pose tracking. We obtained a recognition performance per image level of about 97%, using a linear SVM with pyramidal HOG features LDA-projected to a 61-dimensional subspace. The traffic sign recognition at track level was about 100% in the experiments and we had almost no false detections or missed traffic signs.

9 Conclusions

Traffic sign recognition is a challenging problem. We have proposed a multi-view scheme, which combines 2D and 3D analysis. Following a principle of spending little time on the bulk of the data, and keeping a more refined analysis for the promising parts of the images, the proposed system combines efficiency with good performance. One contribution of the paper is the integer linear optimisation formulation for selecting the optimal candidate extraction methods. The standard sliding window approach is found to be complementary to the proposed detection based on fast extracted candidates, but much slower for similar performance. In case sufficient time is available, it is useful to combine the proposed pipeline with sliding windows. Our experiments show that the state-of-the-art part-based model [8] is slow and performs poorer than our proposed system. Another contribution is the efficient bounded evaluation of linear AdaBoost-like classifiers which brings an important decrease in the computational time. Another novelty is the MDL formulation for best describing the 2D detections with 3D reconstructed traffic signs, without strongly relying on sign positions with respect to the ground plane. Moreover, our task includes accurate 3D localisation of the traffic signs, which prior art did not consider.

In the future, we will research adding further semantic reasoning about traffic signs. They have different probabilities to appear at certain places relative to the road, and also the chances of them co-occurring differ substantially.

Appendix

Appendix details the way of transforming eqs. (2,3,4) into the 0-1 Integer Linear Programming form. Solution of for-

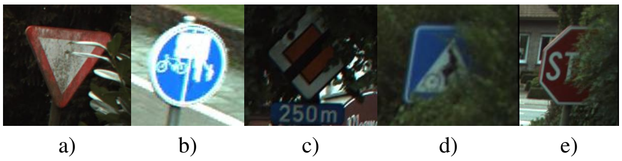


Fig. 15 Not detected(a,b), misplaced(c), or wrongfully classified(d,e) traffic signs.

mulated problems are found via MOSEK optimisation toolbox[3].

Let us suppose we are given n positive samples and m different extraction methods (e.g. colour thresholding with given threshold). Every method correctly extracts (i,e., with sufficient accuracy) some subset of positive samples. Denoting correctly extracted samples by "1" and incorrectly extracted samples by "0", each method is characterized by an n-dimensional extraction vector. We align these vectors row-wise into an $n \times m$ extraction matrix $\mathtt{A}$. Introducing the binary m-dimensional vector $\mathcal{T}$, where selected methods are again denoted by "1" and not selected method by "0", the number of False Negatives from the subset of methods given by $\mathcal{T}$ corresponds to the number of unsatisfied inequalities $\mathtt{A} \cdot \mathcal{T} \geq \mathbf{1}_n$, where $\mathbf{1}_n$ denotes the n-dimensional column vector of ones. Hence, introducing an n-dimensional binary vector of slack variables ξ, the number of False Negatives is

$$\begin{aligned} \mathrm{FN}(\mathcal{T}) = \min_{\xi}\ & \mathbf{1}_n^\top \cdot \xi \\ \text{subj.to: }\ & \mathtt{A} \cdot \mathcal{T} \geq \mathbf{1}_n - \xi, \\ & \xi \in \{0,1\}^n. \end{aligned} \tag{16}$$

Let us be given the m-dimensional real valued vector $\mathbf{b}$ containing the average number of False Positives for every method $1 \dots m$. number of False Positives is estimated on traffic-sign-free images from an urban environment. Then the average number of False Positives obtained using the subset of methods given by $\mathcal{T}$ is

$$\mathrm{FP}(\mathcal{T}) = \mathbf{b}^\top \cdot \mathcal{T} \tag{17}$$

Substituting from Equations (16),(17), yields ILP form of Problem (2):

$$\begin{aligned} \mathcal{T}^* = \arg\min_{\mathcal{T},\xi}\ & \kappa_1 \cdot \mathbf{1}_n^\top \cdot \xi + \mathbf{b}^\top \cdot \mathcal{T} \\ \text{subj.to: }\ & \mathtt{A} \cdot \mathcal{T} \geq \mathbf{1}_n - \xi \\ & \xi \in \{0,1\}^n, \mathcal{T} \in \{0,1\}^m. \end{aligned} \tag{18}$$

Since $\mathrm{card}(\mathcal{T}) = \mathbf{1}_m^\top \cdot \mathcal{T}$, ILP form of Problem (3) is

$$\begin{aligned} \mathcal{T}^* = \arg\min_{\mathcal{T}}\ & \kappa_1 \mathbf{1}_n^\top \cdot \xi + (\mathbf{b}^\top + \kappa_2 \cdot \mathbf{1}_m^\top) \cdot \mathcal{T} \\ \text{subj.to: }\ & \mathtt{A} \cdot \mathcal{T} \geq \mathbf{1}_n - \xi \\ & \xi \in \{0,1\}^n, \mathcal{T} \in \{0,1\}^m. \end{aligned} \tag{19}$$

Finally, introducing the m-dimensional vector $\mathbf{c}$ with average accuracy of every method, ILP form of Problem (4) is:

$$\begin{aligned} \mathcal{T}^* = \arg\min_{\mathcal{T}}\ & \kappa_1 \mathbf{1}_n^\top \cdot \xi + (\mathbf{b}^\top + \kappa_2 \cdot \mathbf{1}_m^\top - \kappa_3 \cdot \mathbf{c}^\top) \cdot \mathcal{T} \\ \text{subj.to: }\ & \mathtt{A} \cdot \mathcal{T} \geq \mathbf{1}_n - \xi \\ & \xi \in \{0,1\}^n, \mathcal{T} \in \{0,1\}^m. \end{aligned} \tag{20}$$

Acknowledgements This work was supported by the Flemish IBBT-URBAN project and the European Commission FP7-231888-EUROPA project. The authors thank GeoAutomation for providing the images.

References

1. 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: IEEE Intelligent Vehicles Symposium (2005)
2. Bosch, A., Zisserman, A., Munoz, X.: Representing shape with a spatial pyramid kernel. In: Proceedings of the 6th ACM international conference on Image and video retrieval, pp. 401–408. ACM Press, New York, NY, USA (2007)
3. Broggi, A., Cerri, P., Medici, P., Porta, P., Ghisio, G.: Real time road signs recognition. In: Intelligent Vehicles Symposium, 2007 IEEE, pp. 981–986 (2007)
4. Brostow, G.J., Shotton, J., Fauqueur, J., Cipolla, R.: Segmentation and recognition using structure from motion point clouds. In: European Conference on Computer Vision (2008)
5. Doubek, P., Perdoch, M., Matas, J., Sochman, J.: Mobile mapping of vertical traffic infrastructure. In: Proceedings of the 13th Computer Vision Winter Workshop, pp. 115–122. Slovenian Pattern Recognition Society, Ljubljana, Slovenia (2008)
6. Everingham, M., Van Gool, L., Williams, C.K.I., Winn, J., Zisserman, A.: The PASCAL Visual Object Classes Challenge 2009 (VOC2009) Results. http://www.pascal-network.org/challenges/VOC/voc2009/workshop/index.html
7. Felzenszwalb, P.F., Girshick, R.B., McAllester, D.: Cascade object detection with deformable part models. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2010)
8. Felzenszwalb, P.F., Girshick, R.B., McAllester, D., Ramanan, D.: Object detection with discriminatively trained part based models. IEEE Transactions on Pattern Analysis and Machine Intelligence **32**(9) (2010)
9. Fleuret, F., Berclaz, J., Fua, P.: Multicamera people tracking with a probabilistic occupancy map. IEEE Transactions on Pattern Analysis and Machine Intelligence **30**(2), 267–282 (2008). DOI http://doi.ieeecomputersociety.org/10.1109/TPAMI.2007.1174
10. Golovinskiy, A., Kim, V.G., Funkhouser, T.: Shape-based recognition of 3d point clouds in urban environments. In: Proceedings of the 12th IEEE International Conference on Computer Vision. Kyoto, Japan (2009)
11. Hoiem, D., Efros, A.A., Hebert, M.: Putting objects in perspective. In: IEEE Conference on Computer Vision and Pattern Recognition, pp 2137–2144, vol. 2 (2006)
12. J. Miura, T.K., Shirai, Y.: An active vision system for real-time traffic sign recognition. In: IEEE Conf. on Intelligent Transportation Systems (ITS), pp. 52–57 (2000)
13. Kingston, T., Gikas, V., Laflamme, C., Larouche, C.: An integrated mobile mapping system for data acquisition and automated asset extraction. In: Proceedings of the 5th Mobile Mapping Technologies Symposium. Padua, Italy (2007)
14. Lafuente, S., Gil, P., Maldonado, R., López, F., Maldonado, S.: Traffic sign shape classification evaluation i: Svm using distance to borders. In: IEEE Intelligent Vehicles Symposium, pp 654–658 (2005)

[3] http://www.mosek.com

15. Leibe, B., Schindler, K., Cornelis, N., Van Gool, L.: Coupled object detection and tracking from static cameras and moving vehicles. IEEE Transactions on Pattern Analysis and Machine Intelligence **30**(10), 1683–1698 (2008)
16. Leonardis, A., Gupta, A., Bajcsy, R.: Segmentation of range images as the search for geometric parametric models. International Journal of Computer Vision, pp 253–277 **14**(3) (1995)
17. Maldonado, S., Lafuente, S., Gil, P., Gómez, H., López, F.: Road-sign detection and recognition based on support vector machines. IEEE Trans. Intelligent Transportation Systems, pp 264–278 **8**(2) (2007)
18. Matas, J., Chum, O., Urban, M., Pajdla, T.: Robust wide baseline stereo from maximally stable extremal regions. In: British Machine Vision Conference, pp 384-393, vol. 1 (2002)
19. Moutarde, F., Bargeton, A., Herbin, A., Chanussot, L.: Robust on-vehicle real-time visual detection of american and european speed limit signs, with a modular traffic signs recognition system. In: IEEE Intelligent Vehicles Symposium, pp 1122–1126 (2007)
20. Munoz, D., Vandapel, N., Hebert, M.: Directional associative markov network for 3-d point cloud classification. In: 3D Data Processing Visualization and Transmission (2008)
21. Nunn, C., Kummert, A., Muller-Schneiders, S.: A novel region of interest selection approach for traffic sign recognition based on 3d modelling. In: IEEE Intelligent Vehicles Symposium, pp 654–658 (2008)
22. Pettersson, N., Petersson, L., Andersson, L.: The histogram feature - a resource-efficient weak classifier. In: IEEE Intelligent Vehicles Symposium, pp 678 - 683 (2008)
23. Prisacariu, V.A., Timofte, R., Zimmermann, K., Reid, I., Van Gool, L.J.: Integrating object detection with 3d tracking towards a better driver assistance system. In: 20th International Conference on Pattern Recognition, ICPR 2010, Istanbul, Turkey, pp. 3344–3347 (2010)
24. Ruta, A., Li, Y., Liu, X.: Towards real-time traffic sign recognition by class-specific discriminative features. In: British Machine Vision Conference (2007)
25. Ruta, A., Porikli, F., Watanabe, S., Li, Y.: In-vehicle camera traffic sign detection and recognition. Mach. Vis. Appl. **22**(2), 359–375 (2011)
26. Schapire, R.E., Freund, Y., Bartlett, P., Lee, W.S.: Boosting the margin: a new explanation for the effectiveness of voting methods. The Annals of Statistics **26**(5), 1651–1686 (1998)
27. Timofte, R., Prisacariu, V.A., Van Gool, L.J., Reid, I.: Chapter 3.5: Combining traffic sign detection with 3d tracking towards better driver assistance. In: C.H. Chen (ed.) Emerging Topics in Computer Vision and its Applications. World Scientific Publishing (2011)
28. Timofte, R., Van Gool, L.: Multi-view manhole detection, recognition, and 3d localisation. In: 1st IEEE/ISPRS Workshop on Computer Vision for Remote Sensing of the Environment (CVRS) in conjunction with the 13th International Conference on Computer Vision (ICCV) (2011)
29. Timofte, R., Zimmermann, K., Van Gool, L.: Multi-view traffic sign detection, recognition, and 3d localisation. In: Proceedings of the IEEE Computer Society's Workshop on Applications of Computer Vision (WACV). IEEE Computer Society, Snowbird, Utah, USA (2009)
30. Viola, P., Jones, M.: Robust real-time face detection. In: IEEE International Conference on Computer Vision, vol. 2, pp. 747–757 (2001)
31. Wojek, C., Schiele, B.: A dynamic conditional random field model for joint labeling of object and scene classes. In: European Conference on Computer Vision, pp 733-747 (2008)