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

Radu Timofte, Karel Zimmermann, Luc Van Gool
ESAT-PSI / IBBT, Katholieke Universiteit Leuven, Belgium
{Radu.Timofte, Karel.Zimmermann, Luc.VanGool}@esat.kuleuven.be

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. A van with 8 roof-mounted cameras drove through the streets and took images every meter. The paper proposes a pipeline for the efficient detection and recognition of traffic signs. The task is challenging, as illumination conditions change regularly, occlusions are frequent, 3D 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.

1. Introduction

Mobile mapping is used even more often, e.g. for the creation of 3D city models for navigation, or for digital surveying campaigns by public authorities to turn old paper maps into digital databases. Several applications need the locations and types of the traffic signs along the roads, see Fig. 2. The paper describes an efficient pipeline for the detection and recognition of such signs. Over the last years the computer vision community has largely turned towards the recognition of object classes, rather than specific patterns. However, it would be a mistake to believe that the problem at hand is not extremely challenging. Moreover, false positive and false negative rates have to be very low for automated methods to be useful in this case. That is why currently most of this work is still carried out by human operators. There are all the traditional problems of variations in lighting, pose, and background, and of occlusions by other objects, see Fig. 1a. In addition, these signs are often not as precisely standardised as one would expect (this also depends on the country, in our case Belgium), see Fig. 1b.

Whereas the majority of contributions so far work with a rather small subset of sign types, our dataset includes 62 different types of signs. Moreover, the authors usually focus

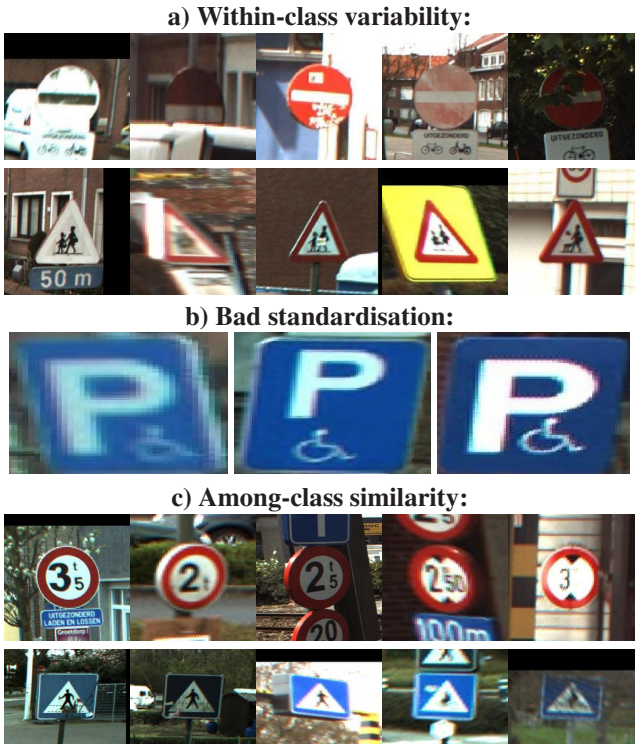

Figure 1. **Within-class variability and between-class similarity are high:** Each of first five rows contains instances of the same class. Each of the last two rows shows traffic signs from two distinct classes.

Figure 2. **3D mapped traffic signs** in a reconstructed scene.

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.

Even under simpler circumstances, the results of traffic sign detection and recognition thus far testify to the complexity of the task. Lafuente et al. [8] had 26% of false negatives for 3 false positives per image. Maldonado et al. [12] used image thresholding followed by SVM classification. They mention that every traffic sign has been detected at least twice in the total number of 5000 video frames, with 22 false alarms. Detection rates per view are not given. Nunn et al. [16] 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%. They still found 16494 false positives. All signs outside the ROI are discarded. Pettersson et al. [17] reported on one of the few results off the highway. But they 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. [14] reported no false positives at all in a 150 minutes long video, but with 11% of all traffic signs left undetected. Ruta et al. [18] combine image thresholding and shape detection achieving 6.2% of false negatives, the number of false positives is not mentioned. Broggi et al. [3] proposed a system similar to [12] 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 limited to a single image and is therefore also purely 2D.

Results so far are not good enough to roll out such methods at a large scale. The numbers of false positives and false negatives are too high. As a matter of fact, this literature is a bit decoupled from mainstream computer vision. There, recent years have witnessed a flurry of activity in object class detection, incl. many classes that are to be found in street scenes. The vast majority works from a single image [6, 1, 9]. Yet, approaches have emerged that try to exploit contextual information like the estimated position of a ground plane, thereby introducing a weak notion of 3D scene layout [7]. This was seen to be very beneficial. In a similar vein, Wojek and Schiele [20] went further in coupling object detection and scene labeling approaches. Also their approach still works from a single image.

As a second strand of research, some recent techniques have focused fully on the annotation of subsets of 3D point clouds [4, 15]. 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 to handle. Moreover, traffic signs are planar. Given that image appearance already yields such strong clues for object recognition, we propose a hybrid strategy.

We do not stop at single view detection and recognition, but include 3D localisation as well. Localisation probably most resembles that of Cornelis et al. [5], who also combined explicit 3D information with 2D car and person detection in a mobile city mapping context. An important difference with this earlier work lies in the far less stringent constraints offered by the 3D scene layout (no longer objects restricted to a ground plane) and the looser spatial arrangements when looking for traffic signs. Moreover, traffic signs have been designed to come as subclasses, and different types of traffic signs within the same subclass have the same shape and colour distributions. The distinction lies in rather small details. These need to be picked up by the system. Hence, the challenge is one of detecting traffic signs irrespective of the typical problems of changing appearances and occlusions, but at the same time recognizing specific sign types based on small differences.

The structure of the remainder of the paper is as follows. Section 2 first gives an overview of the different steps taken by the system. Then, we focus on the most innovative aspects. Section 3 explains the initial selection of good candidates within the individual images. Section 4 explains the MDL formulation for 3D traffic sign localisation. Section 5 discusses experimental results and draws conclusions.

2. 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 has 8 cameras on its roof: two looking ahead, two looking back, two looking to the left, and two to the right. 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 online driver assistance but an offline traffic sign mapping system performing optimization over the captured views. The considered traffic signs are those that are captured at a distance of less than 50 meters The proposed system first processes single images independently, keeping the detection rate very high and the number of false positives (FP) reasonable. Single-view traffic sign detection in conjunction with the use of scene geometry subsequently allows for global optimiza-

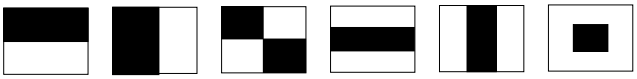

Figure 3. **Haar-like features** used in our implementation.

tion which performs 3D localisation and a refinement simultaneously. Since we deal with hundreds of thousands of high-resolution images the approach is to quickly throw out most of the background, to then invest increasing amounts of time on whatever patterns survive previous steps. We now describe each subsequent step of the single-view and multi-view processing pipelines in more detail.

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

1) Candidate extraction - very fast preprocessing step, where the optimal combination of simple (i.e. computationally cheap), adjustable methods selects bounding boxes with possible traffic signs. This step is designed to yield very few false negatives (FN, i.e. the number of missed traffic signs), while keeping the number of false positives (FP, i.e. the number of accepted background regions) in check. This part of the pipeline is described in more detail in section 3, as most of the novelty in single-view processing is here.

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 Discrete AdaBoost classifier [19]. 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 HOGs [2] and AdaBoost-selected Haar-like features.

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 remaining candidates, within 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 backprojected 3D hypothesis against the 2D image candidates. Visual consistency gives higher weight to pairs with the same basic shape.

5) Multi-view MDL hypothesis pruning - Minimum description length is used to select the subset of 3D hypotheses which best explains the overall set of 2D candidates. A side product of the MDL optimization is quite a clean set of 2D candidates corresponding to each particular 3D hypothesis. These candidates allow for hypothesis position refinement. Usually, steps 4) and 5) are iterated. More details are given in section 4.

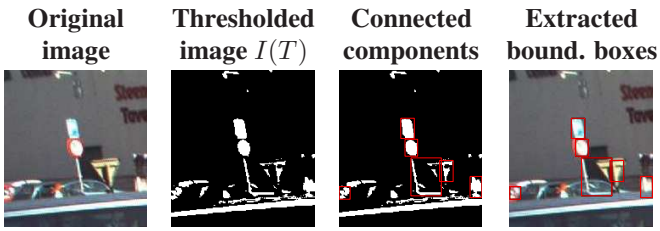

Figure 4. **Colour-based extraction** method for threshold $T = (0.5, 0.2, -0.4, 1.0)^\top$

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.

3. Single-view candidate extraction

For step 1, we start from connected components in a thresholded image, an idea which has already been used in [12, 3]. The principle is demonstrated in Fig. 4, where the *thresholded image* is obtained from a colour image, with colour channels (I_R, I_G, I_B), by application of a colour threshold $T = (t, a, b, c)^\top$

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

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 3.1 explains how the set of colour thresholds are learned and how, starting from those, the colour-based candidates are extracted. Section 3.2 then describes a shape-based Hough transform. This takes the borders of the colour-based candidates as input.

3.1. Colour-based candidate extraction

The task is to find the optimal set $\mathcal{T}$ of colour thresholds, given some criterion. Since for most interesting such criteria the problem is NP-complete, we formulate our search as

Figure 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.

a Boolean Linear Programming problem. We have experimentally found that finding the real optimum takes several hours, but that BLP, 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

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

where $\mathrm{FP}(\mathcal{T})$ stands for the number of false positives and $\mathrm{FN}(\mathcal{T})$ 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 $\mathrm{card}(\mathcal{T})$ of the set of selected thresholds. This can be either a hard constraint $\mathrm{card}(\mathcal{T}) < \epsilon$ or a soft constraint as in

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

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*[1] is usually quite important we add a term assuring that accurate methods are preferred:

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

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

Simple colour thresholding comes out to be insufficient in practice. We therefore introduce a couple of refinements. Many traffic signs have parts that cannot be separated from the background with any threshold. See for example Fig. 6. The rim of the sign is too similar in colour to the background. Yet, the white inner part can be separated rather easily. We therefore introduce the extended threshold

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

[1]Accuracy is the average of overlap between ground truth bounding boxes with extracted bounding boxes.

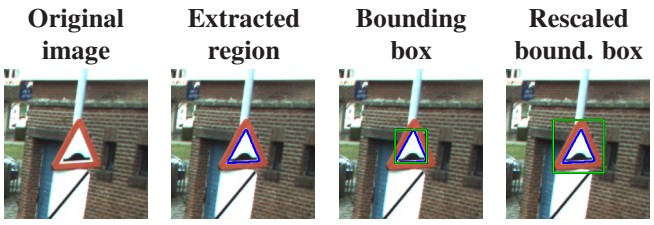

Figure 6. **Demonstration of the extended threshold.** The object is not well locally separable from the background, because bricks have a colour similar to that of the red boundary. Therefore the inner white part is extracted and the resulting bounding box is rescaled $\overline{T} = (0.1, -0.433, -0.250,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

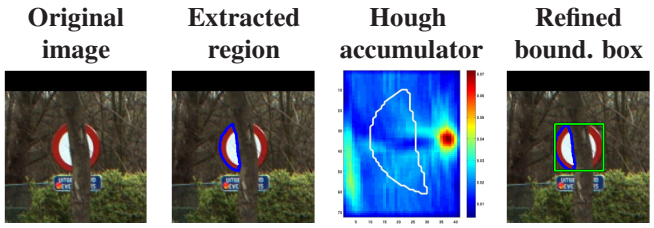

Figure 7. **Shape-based extraction principle.** The border of the colour-based extracted region (blue) votes for different shapes in a Hough accumulator. The green bounding box corresponds to the maximum.

which consists of the original threshold T and vertical resp. horizontal scaling factors (s_r, s_c) to be applied to the extracted bounding box. Such extended threshold - in the sequel simply referred to as threshold - can reveal a traffic sign, even if its rim poses problems. Learning now becomes searching for the set of 6-dimensional thresholds.

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) [13]. Instead of using the bounding box directly 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 Δ, this 'TMSER' method is parametrized by two parameters (ϵ, Δ).

3.2. Shape-based candidate extraction

Traffic signs have characteristic shapes. Each of the above thresholds (with scaling and TMSER extensions) let pass a series of connected components, i.e. regions. To these regions we now apply an additional 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-

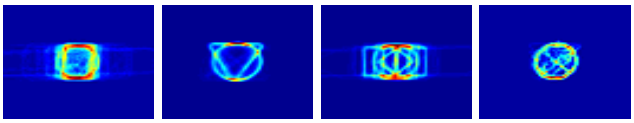

Figure 8. **Threshold-specific fuzzy templates.** Selected subset $\{23, 12, 28, 32\}$ from 44 fuzzy-templates.

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 are usually specialized for some kinds of traffic signs, we learn threshold-specific fuzzy templates. 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 the 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. We always keep the original bounding box as a separate candidate, however.

4. 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.

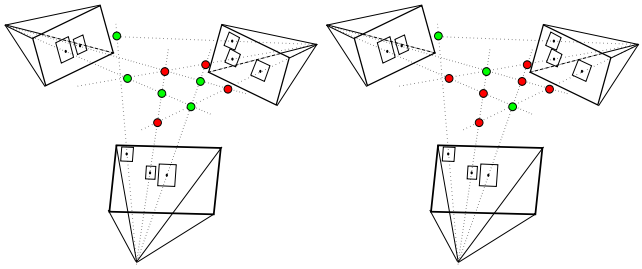

Figure 9. **MDL principle** - the corresponding pairs generate 3D hypotheses, from which one should not pick up the green subset on the *left*, but rather the best/smallest subset explaining the 2D detections, shown in green on the *right*, thus following the MDL principles.

We start by generating an overcomplete set of hypotheses: For every single 2D detection we collect every *geometrically* and *visually* consistent correspondence and use this pair to generate a 3D hypothesis, see Figure 9. 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 responses for real traffic signs. The following MDL optimization selects the simplest subset which best explains 2D detections, see Figure 9, right.

For each 3D hypothesis we will have a 3D position of the centre, a fitted plane and thus an orientation (and sense), and estimated probabilities to belong to the basic shapes. For a specific hypothesis h we gather the set of supporting 2D candidates which have a *coverage*[2] with the 2D projection of h above 0.05 and for which the candidate camera and the hypothesis are facing eachother (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:

$$S_h \sim S_d - k_1 S_m - k_2 S_e \tag{6}$$

S_d is the part of the hypothesis which is explained by the supporting candidates, S_m is the cost of coding the model itself, while S_e represents those parts that are not explaining the given hypothesis, and k_1, k_2 are weights (as in [10]). 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.

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

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:

$$p(c,h) = \max_{t\in\{\triangle,\triangledown,\circ,\square,\Diamond\}} p_t(c)p_t(h) \tag{7}$$

$$S_d = \sum_{c\in C_h} O_{c,h}p(c,h) \tag{8}$$

$$S_e = \sum_{c\in C_h} (1-O_{c,h})p(c,h) \tag{9}$$

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

$$S_{h_i,h_j} = \sum_{c\in C} \min_{t\in\{i,j\}} (S_{d_t}(c) - k_2 S_{e_t}(c)) \tag{10}$$

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

Leonardis et al. [11] have shown that if only pairwise interactions are considered, then the Quadratic Boolean Problem (QBP) formulation gives the optimal set of models:

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

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, $s_{ij} = s_{ji} = -0.5S_{h_i,h_j}$. The restriction to pairwise interactions will not fully capture situations where more than 2 hypotheses affect the same image area.

5. Experiments

5.1. Ground truth data

Our ground truth data consists of 7356 stills (in total 11219 bounding boxes), which correspond to 2459 traffic signs visible at less than 50 meters in at least one view. It includes challenging samples as shown in Fig. 1. The multi-view traffic sign detection, recognition, and localisation are evaluated on 4 sequences, captured by 8 roof-mounted cameras on the van, with a total of 121632 frames and 269 different traffic signs. For each sign the type and 3D location were recorded.

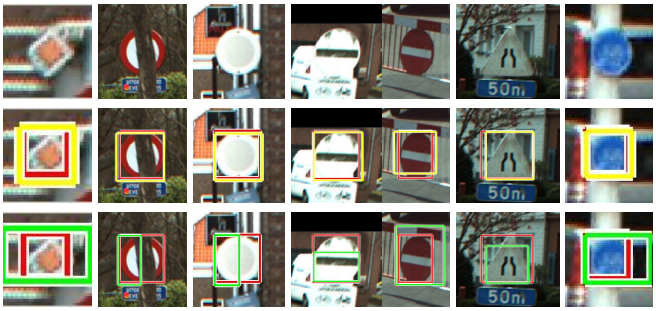

Figure 10. **Shape-based extractable but threshold inseparable traffic signs** - the ground truth is delineated by a red rectangle, the best shape-based detection is shown in yellow and the best colour-based one in green.

5.2. Single-view evaluation

The detection and extraction errors (Table 1) 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, where we typically have visibility in 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 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.

Table 1 shows results of both the candidate extraction method, which, naturally, has a high number of FP (first four rows) and detection (i.e. candidate extraction followed by AdaBoost detector), which is shown in last two rows. The ROC curve in Table 1 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. Shape extraction significantly increases the number of false positives (see for example 4th row in the table). The reason for the increase 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. Traffic signs, not threshold separable as a whole, but which could be extracted based on their shape, are shown in Fig. 10.

5.3. Multi-view evaluation

In this section, we report the multi-view results. Whereas we only reported on detection in the single-view case, we now also pay attention to recognition (the determination of

	FN-TS		FN-BB		FP per
	[%]	#/1274	[%]	#/3756	2MP img
Extr1 (colour)	0.5%	7	1.5%	58	3 442.4
Extr2 (colour+TMSER)	0.4%	5	1.4%	53	4 008.5
Extr3 (colour+shape)	0.2%	2	1.0%	36	6 670.3
Extr4 (colour+TMSER+shape)	0.1%	1	1.0%	36	7 157.3
Det + Extr1	2.4%	31	4.9%	184	2.5
Det + Extr4	2.2%	28	4.3%	163	2.5

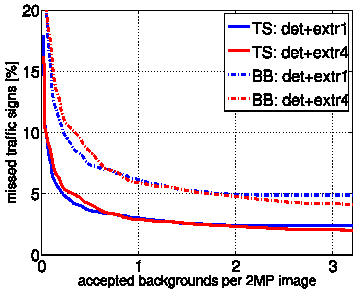

Table 1. **Summary of achieved results in single-view detection**. Meaning of the above used abbreviations is the following **colour** means method described in Section 3.1, **TMSER** stands for TMSER(ϵ, Δ) = TMSER$(0.1, 0.2)$, **shape** is Section 3.2. **FN-BB** means false negative with respect to 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**.

#	No.frames/TSs	Loc.TS	FP	Loc.TS$_r$	FP$_r$	Rec.TS
1	8×3001 /78	75(96.2%)	9	74(94.9%)	5	98.7%
2	8×6201 /71	68(95.8%)	14	68(95.8%)	13	95.6%
3	8×2001 /44	41(93.2%)	5	41(93.2%)	2	100%
4	8×4001 /76	73(96.1%)	9	73(96.1%)	8	97.3%
$\sum$	8×15204 /269	257(95.6%)	37	256(95.2%)	28	97.7%

Table 2. **Summary of 3D achieved results**. Meaning of the above used abbreviations is the following **Loc.TS** means correctly located traffic signs in 3D space, **FP** stands for false positives in 3D and **Rec.TS** are the 3D recognition results with respect to the located 3D TS. **Loc.TS**$_r$ and **FP**$_r$ show the results from the original method after final refinement with template matching.

the specific type of each traffic sign) and localisation performance. Some scores may seem a bit low compared to the single-view ones – here and in the literature – but detection in this section includes localisation (within 3m in X-Y-Z). Note, that most of the incorrectly 3D localised traffic signs were detected in at least one view.

We evaluate our multi-view pipeline on the 4 image sets. The results are summarized in Table 2. 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. Fig. 11 shows samples of missed traffic signs (i.e. not detected or misplaced). 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 24.54 cm. 90% of the located traffic signs are reconstructed within 50 cm from the ground truth, but we have also 3 traffic signs that are reconstructed at more than 1.5 m.

Recognition results are summarized in the last column of Table 2. It is shown that the overall recognition is 97.7%.

Figure 11. **Not detected or misplaced traffic signs.**

6. 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 Boolean linear optimisation formulation for selecting the optimal candidate extraction methods. 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 3D localisation of the signs, which prior art did not consider.

In the future, we will add 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.

Acknowledgments

This work was supported by the Flemish IBBT-URBAN project. The authors thank GeoAutomation for providing the images.

Appendix

It is shown, how to transform the problems of eqs. (2,3,4) into the Boolean Linear programming form.

Let us suppose we are given n positive samples and m different extraction methods (e.g. color thresholding with given threshold). Every method correctly extracts (i,e., with sufficient accuracy) some subset of positive samples. De-

noting 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

$$\begin{aligned} \text{FN}(\mathcal{T}) &= \min_{\xi} \ \mathbf{1}_n^\top \cdot \xi \\ &\text{subj.to: } \mathtt{A} \cdot \mathcal{T} \geq \mathbf{1}_n - \xi, \\ &\qquad \xi \in \{0,1\}^n. \end{aligned} \tag{12}$$

Let us be given the m-dimensional real valued vector $\mathbf{b}$ containing the average number of False Positives for every method $1 \ldots m$, then the average number of False Positives obtained using the subset of methods given by $\mathcal{T}$ is

$$\text{FP}(\mathcal{T}) = \mathbf{b}^\top \cdot \mathcal{T} \tag{13}$$

Substituting from Equations (12,13), Problem (2) is rewritten as:

$$\begin{aligned} \mathcal{T}^* &= \arg\min_{\mathcal{T},\xi} \ \kappa_1 \cdot \mathbf{1}_n^\top \cdot \xi + \mathbf{b}^\top \cdot \mathcal{T} \\ &\text{subj.to: } \mathtt{A} \cdot \mathcal{T} \geq \mathbf{1}_n - \xi \\ &\qquad \xi \in \{0,1\}^n, \mathcal{T} \in \{0,1\}^m. \end{aligned} \tag{14}$$

In addition to that, since $\text{card}(\mathcal{T}) = \mathbf{1}_m^\top \cdot \mathcal{T}$, Problem (3) becomes

$$\begin{aligned} \mathcal{T}^* &= \arg\min_{\mathcal{T}} \ \kappa_1 \mathbf{1}_n^\top \cdot \xi + (\mathbf{b}^\top + \kappa_2 \cdot \mathbf{1}_m^\top) \cdot \mathcal{T} \\ &\text{subj.to: } \mathtt{A} \cdot \mathcal{T} \geq \mathbf{1}_n - \xi \\ &\qquad \xi \in \{0,1\}^n, \mathcal{T} \in \{0,1\}^m. \end{aligned} \tag{15}$$

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

$$\begin{aligned} \mathcal{T}^* &= \arg\min_{\mathcal{T}} \ \kappa_1 \mathbf{1}_n^\top \cdot \xi + (\mathbf{b}^\top + \kappa_2 \cdot \mathbf{1}_m^\top - \kappa_3 \cdot \mathbf{c}^\top) \cdot \mathcal{T} \\ &\text{subj.to: } \mathtt{A} \cdot \mathcal{T} \geq \mathbf{1}_n - \xi \\ &\qquad \xi \in \{0,1\}^n, \mathcal{T} \in \{0,1\}^m. \end{aligned} \tag{16}$$

References

[1] M. Andriluka, S. Roth, and B. Schiele. People-tracking-by-detection and people-detection-by-tracking. In *CVPR*, 2008.

[2] A. Bosch, A. Zisserman, and X. Munoz. Representing shape with a spatial pyramid kernel. In *Proceedings of the 6th ACM international conference on Image and video retrieval*, pages 401–408, New York, NY, USA, 2007. ACM Press.

[3] A. Broggi, P. Cerri, P. Medici, P. Porta, and G. G. Real time road signs recognition. *IVS*, 2007.

[4] G. J. Brostow, J. Shotton, J. Fauqueur, and R. Cipolla. Segmentation and recognition using structure from motion point clouds. In *ECCV*, 2008.

[5] N. Cornelis, B. Leibe, K. Cornelis, and L. Van Gool. 3d urban scene modeling integrating recognition and reconstruction. *IJCV*, 78(2-3):121–141, 2008.

[6] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. *CVPR, 886-893*, 1, 2005.

[7] D. Hoiem, A. A. Efros, and M. Hebert. Putting objects in perspective. In *CVPR, pp 2137–2144*, volume 2, 2006.

[8] S. Lafuente, P. Gil, R. Maldonado, F. López, and S. Maldonado. Traffic sign shape classification evaluation i: Svm using distance to borders. In *IVS, pp 654–658*, 2005.

[9] B. Leibe, A. Leonardis, and B. Schiele. Robust object detection with interleaved categorization and segmentation. *IJCV, pp 259-289*, 77(1-3), 2008.

[10] B. Leibe, K. Schindler, N. Cornelis, and L. Van Gool. Coupled object detection and tracking from static cameras and moving vehicles. *TPAMI*, 30(10):1683–1698, 2008.

[11] A. Leonardis, A. Gupta, and R. Bajcsy. Segmentation of range images as the search for geometric parametric models. *IJCV, pp 253–277*, 14(3), 1995.

[12] S. Maldonado, S. Lafuente, P. Gil, H. Gómez, and F. López. Road-sign detection and recognition based on support vector machines. *ITS, pp 264–278*, 8(2), 2007.

[13] J. Matas, O. Chum, M. Urban, and T. Pajdla. Robust wide baseline stereo from maximally stable extremal regions. In *BMVC, pp 384-393*, volume 1, 2002.

[14] F. Moutarde, A. Bargeton, A. Herbin, and L. Chanussot. Robust on-vehicle real-time visual detection of american and european speed limit signs, with a modular traffic signs recognition system. In *IVS, pp 1122–1126*, 2007.

[15] D. Munoz, N. Vandapel, and M. Hebert. Directional associative markov network for 3-d point cloud classification. In *3DPVT*, 2008.

[16] C. Nunn, A. Kummert, and S. Muller-Schneiders. A novel region of interest selection approach for traffic sign recognition based on 3d modelling. In *IVS, pp 654–658*, 2008.

[17] N. Pettersson, L. Petersson, and L. Andersson. The histogram feature - a resource-efficient weak classifier. In *IVS, pp 678 - 683*, 2008.

[18] A. Ruta, Y. Li, and X. Liu. Towards real-time traffic sign recognition by class-specific discriminative features. In *BMVC*, 2007.

[19] P. Viola and M. Jones. Robust real-time face detection. In *ICCV*, volume 2, pages 747–757, 2001.

[20] C. Wojek and B. Schiele. A dynamic conditional random field model for joint labeling of object and scene classes. In *ECCV, pp 733-747*, 2008.